Welcome to Game Warden

Game Warden is a fully managed, secure DevSecOps platform that supports the compliant delivery of SaaS applications to a variety of environments, including FedRAMP, classified (SIPR), unclassified (NIPR), and commercial networks.

The platform is designed to reduce the complexity of achieving and maintaining an **Authority to Operate (ATO)**. It provides pre-approved hosting environments, seamless CI/CD integration, and built-in security tooling aligned with the Department of Defense (DoD) and federal compliance frameworks.

Platform capabilities

Game Warden provides development and operations teams with the ability to:

- Deploy applications and services across multiple environment types with inherited compliance.
- Integrate with CI/CD pipelines to automate container scanning, hardening, and deployment.
- Monitor and manage application artifacts, deployment status, and vulnerability findings.
- Operate securely across multiple cloud providers and DoD networks.

The platform enables organization to build applications for a wide range of missions and classifications.

Managing applications and services

The Game Warden App provides a centralized interface for:

- Managing deployments across staging and production environments.
- Tracking service-level activity such as pipeline runs, image pushes, and container scans.
- Monitoring and responding to security findings.
- Accessing compliance artifacts such as **Body of Evidence (BoE)** and **Certificates to Field (CtFs)/Software Approval**.

Each deployed application is linked to a dedicated BoE to ensure traceability and compliance across its lifecycle.

Supported environments

Game Warden supports deployments to:

- FedRAMP-authorized environments
- DoD Impact Levels (IL2–IL6), which are defined by the DoD Cloud Computing Security Requirements Guide (CC SRG):
  - **IL2** – Public or non-critical mission information (typically deployed to unclassified environments such as NIPRNet).
  - **IL4** – Includes Controlled Unclassified Information (CUI), such as For Official Use Only (FOUO), Personally Identifiable Information (PII), and Protected Health Information (PHI), along with non-critical mission information and non-National Security Systems (non-NSS).
  - **IL5** – Includes higher sensitivity CUI, mission-critical information, and National Security Systems (NSS). IL5 occupies a narrow space between IL4 and IL6 and is distinguished by its authorization to process NSS.
  - **IL6** – Classified information systems and data classified up to SECRET (typically deployed to classified environments such as SIPRNet or JWICS).
- Commercial environments

Game Warden is designed to support mission owners and software teams in delivering secure, compliant applications across government and commercial domains with reduced operational overhead.

View Game Warden Help Center offline

Want to view our doc site offline?

1. Click this link to download the archive.
2. Install Python if you don't already have it.
3. Extract the `.tar.gz` file.
4. Open your terminal (or Command Prompt on a Windows machine), navigate to the extracted folder, then run: `python3 -m http.server 8080`.
5. In your browser, go to http://localhost:8080/ to view the site.

Got a question? Reach out to Second Front today!

Understanding Impact Levels

The Department of Defense (DoD) uses Impact Levels (IL) to classify information systems based on the potential consequences if their data is compromised. This classification ensures that each system has appropriate security measures.

Each Impact Level considers three main security aspects:

- **Confidentiality**: Ensuring only authorized individuals access the information.
- **Integrity**: Maintaining the accuracy and trustworthiness of the information.
- **Availability**: Guaranteeing reliable access to information for authorized users.

The Defense Information Systems Agency (DISA) developed the **DoD Cloud Computing Security Requirements Guide** (CC SRG) to provide security guidelines for cloud computing. This guide incorporates standards from:

- Federal Information Security Management Act (FISMA)
- National Institute of Standards and Technology (NIST) Special Publication 800-37

The CC SRG extends the FedRAMP framework with enhanced, DoD-specific security controls designed to meet the distinct demands of military and national security systems.

Detailed DoD Impact Levels (IL)

Impact Level	Information Sensitivity	Security Controls	Location	Off-Premises Connectivity	Separation	Personnel Requirements
2	Public or non-critical Mission Information	FedRAMP Moderate	US / US outlying areas Or DoD on-premises	Internet	Virtual / Logical PUBLIC COMMUNITY	National Agency Check and Inquiries (NACI)
4	CUI or Non-CUI Non-Critical Mission Information Non-National Security Systems	Level 2 + CUI-Specific Tailored Set	US / US outlying areas Or DoD on-premises	NIPRNet via CAP	Virtual / Logical Limited "Public" Community Strong virtual separation between tenant systems & information	US Persons ADP-1 Single scope Background Investigation (SSBI)
5	Higher Sensitivity CUI Mission Critical Information National Security Systems	Level 4 + NSS- Specific Tailored Set	US / US outlying areas Or DoD on-premises	NIPRNet via CAP	Virtual / Logical Federal Gov. Community Dedicated Multi-Tenant Infrastructure Physically Separate from Non-Federal Systems Strong virtual separation between tenant systems & information	ADP -2 National Agency Check with Law and Credit (NACLC) Non-Disclosure Agreement (NDA)
6	Classified SECRET National Security Systems	Level 5 + Classified Overlay	US / US outlying areas Or DoD on-premises CLEARED /CLASSIFIED FACILITIES	SIPRNET DIRECT With DoD SIPRNET Enclave Connection approval	Virtual / Logical Federal Gov. Community Dedicated Multi-Tenant Infrastructure Physically Separate from Non-Federal and Unclassified Systems Strong virtual separation between tenant systems & information	US Citizens w/ Favorably adjudicated SSBI & SECRET Clearance NDA

Overview of Impact Levels

The DoD defines several Impact Levels, each corresponding to the sensitivity of the data:

Impact Level 2 (IL2)

- **Data Type**: Publicly releasable information.
- **Use Case**: Suitable for systems handling non-sensitive data.

Impact Level 4 (IL4)

- **Data Type**: Controlled Unclassified Information (CUI).
- **Use Case**: Systems requiring protection for sensitive but unclassified data.

Impact Level 5 (IL5)

- **Data Type**: Controlled Unclassified Information (CUI) and National Security Systems (NSS) information.
- **Use Case**: Systems that handle higher sensitivity data needing stricter controls.

Impact Level 6 (IL6)

- **Data Type**: Classified information up to SECRET level.
- **Use Case**: Systems managing classified data with the highest security requirements.

Why Impact Levels matter

If your organization wants to work with the DoD, understanding Impact Levels is essential due to the following reasons:

1. You must match the right security level:

Each DoD project or system requires a certain level of protection. The Impact Level tells you:

- What kind of data you're dealing with (e.g., public, sensitive, or classified).
- What security controls are needed to protect that data.

If your solution doesn't meet the required Impact Level, it can't be used in that environment.

2. It impacts your cloud hosting choices:

Not all cloud platforms are authorized for every Impact Level. For example:

- IL2 can use commercial cloud environments.
- IL5 and IL6 require specially authorized environments.

This affects how and where you can deploy your software.

3. It shapes your compliance path:

The higher the Impact Level, the more security and documentation you need. Knowing your target level early helps you:

- Plan your compliance and certification strategy.
- Avoid surprises later in the authorization process.

4. It builds trust with DoD partners:

Demonstrating that you understand and meet the right Impact Level shows:

- You're serious about cybersecurity.
- You're a credible partner for defense work.

Game Warden environments

Game Warden environments are aligned with DoD Impact Levels and follow a structured deployment lifecycle. Each deployment stage corresponds to a designated Impact Level aligned with the data sensitivity and operational purpose of that environment.

Development (DEV)

Your application is first deployed to the Game Warden Commercial environment, which serves as our DEV for early configuration, code validation, and foundational functional testing. This environment is authorized for non-classified data only and provides broad access for both your team and Game Warden customer engineering staff.

During this phase, our engineers work closely with your team to validate configurations, resolve issues, and confirm that your application operates effectively within the Game Warden ecosystem before progressing to higher security boundaries.

Staging (STG)

After initial onboarding, your application is promoted to the Staging (STG) environment. The Impact Level of the STG environment matches the intended Production (PRD) IL. For example, if your application is destined for IL5 production, your staging environment will also operate at IL5.

This is where extensive testing and verification take place under real-world conditions. Access to STG is more tightly controlled and may be limited depending on classification and mission sensitivity.

Production (PRD)

Once staging tests are successfully completed, your application is deployed to Production (PRD) at your selected IL (IL4 or IL5). This live environment is accessible to authorized end users and is protected by the full set of controls appropriate to its Impact Level.

Game Warden supports production deployments at IL4 and IL5, with IL6 support on the roadmap. Additional security controls are applied as the Impact Level increases to ensure compliance and operational integrity.

> **Warning**
>
> Access to each environment is restricted based on the user's assigned IL. Only personnel with the appropriate credentials and authorizations can access higher-level environments such as IL5.

Impact Level 5 Compliance Requirements

Impact Level 5 (IL5) is designed for systems that manage Controlled Unclassified Information (CUI) with a mission-critical or national security context. These systems require stronger safeguards compared to IL4 environments because they support DoD operations and warfighter missions.

IL5 compliance requirements

To achieve IL5 authorization, cloud systems must meet stringent criteria, including but not limited to:

- **Hosting in a U.S. facility** and using a cloud service operated by U.S. citizens.
- **Physical and logical separation** from non-DoD customers.
- **Access controls and auditing** that meet or exceed NIST SP 800-53 security controls at the High baseline.
- **Continuous monitoring** and vulnerability management compliant with DoD standards.
- **Incident response and reporting** procedures aligned with DoD cyber operations.

These requirements ensure that systems operating at IL5 can protect mission-sensitive data while maintaining availability and resilience. Most commercial cloud offerings require significant customization and investment to meet these demands.

Obtain access to IL5

To access an IL5 environment, you must complete a series of prerequisites to ensure your identity, system, and access controls meet DoD security standards.

1. Begin with Platform One onboarding

1. Visit the Platform One site to provision a Platform One (P1) Single Sign-On (SSO) account. See Platform One for detailed instructions.
2. Install and configure Appgate SDP, which is required for secure network access. See Install Appgate SDP for detailed instructions.
3. Configure your government-issued access card—such as a CAC (Common Access Card), ECA (External Certification Authority), or PIV (Personal Identity Verification)—to map to your P1 SSO account. See Platform One (P1) SSO Configuration for detailed instructions.
4. Run compliance hardening scripts on your device to achieve **at least 80% compliant** before proceeding. See IL5 Access Setup for detailed instructions.

2. Access and review IL4 documentation

Before gaining IL5 access, you must review critical documentation hosted in an IL4 environment, which outlines compliance expectations for IL5 systems.

As IL4 documentation is hosted within a secure environment, it requires the same access setup completed during Platform One onboarding:

- P1 SSO account
- Appgate SDP
- Mapped government-issued access card (CAC, ECA, or PIV)

If you've completed the onboarding steps above, you already have the necessary credentials to access the IL4 documentation.

3. Additional configuration

After satisfying the above requirements:

- Our engineering team will ensure you are assigned to the correct application groups.
- Additional access configurations may be required based on your role and project scope.

4. Need help?

If you have any questions or encounter issues during the process, reach out to your Technical Implementation Manager (TIM) for assistance.

Understand Authorization to Operate & Deployment Passport

Deploying secure applications into Department of Defense (DoD) environments requires strict compliance with cybersecurity and authorization standards. Game Warden simplifies this process through its existing Authorization to Operate (ATO), allowing your organization to inherit this authorization through a structured pathway.

This guide explains how Game Warden's ATO, Deployment Passport, and Certificate to Field (CtF)/Software Approval work together to enable compliant deployments.

Key terms at a glance

Term	What It Means	Why It Matters
ATO	A government-issued authorization that allows your system to operate on DoD networks.	Required to deploy to any DoD environment. Game Warden holds an ATO for Impact Levels (IL) 2, 4, and 5, authorized through AFWERX. Applications hosted on Game Warden may inherit this ATO.
ATO Inheritance	Your app runs under Game Warden's existing ATO instead of undergoing its own full authorization process.	Reduces compliance burden and accelerates deployment timelines.
CtF/Software Approval	An approval memo signed by an authorizing official or delegated authority that confirms your app can inherit Game Warden's ATO for a specific Impact Level.	Required before deploying your app to staging (STG) or production (PRD) environments.
Deployment Passport	A bundle of compliance documents, including the CtF/Software Approval, demonstrates that your app meets all required security standards.	Reviewed by authorizing official or delegated authority.

How the process works

1. You build your app and prepare it for deployment.
2. Game Warden helps you harden and scan the app, generating evidence of security posture.
3. The Game Warden team uploads this body of evidence into the *Deployment Passports* section of the **Documents** page in the Game Warden App. Optionally, you can submit additional compliance evidence or external ATOs.
4. The Deployment Passport package is sent to an authorizing official or delegated authority for reviewing and approval.
5. Once approved, you are issued a CtF/Software Approval. This approval allows your app to **inherit Game Warden's ATO** and be deployed to staging (STG) and production (PRD) environments.

   Danger

   The ATO is non-transferable and only valid for Game Warden-hosted applications.

Required components for the Deployment Passport

The following components must be compiled into your Deployment Passport package to request a CtF/Software Approval:

- **Authorization Boundary Diagram** – A visual representation of your application architecture, showing all components, data flows, containers, and external services. It must indicate the direction of data movement (ingress, egress, bidirectional), as well as the ports and protocols used. This diagram ensures your deployment aligns with required ATO specifications. For more information, see Authorization Boundary Diagram.
- **Body of Evidence (BoE)** – A detailed form completed within the Game Warden app that outlines how your application meets ATO controls. It includes system architecture, data handling practices, external approvals (if applicable), and proof of an active government contract. The Game Warden security team reviews and approves the BoE. For more information, see Body of Evidence.
- **Security Findings Summary** – A document containing hyperlinks to vulnerability scan results, exported from Findings, providing a snapshot of your application's security posture.
- **ISSM Critical/High/Stop Security Findings Memo** (if applicable) – A waiver memo addressing critical or high Common Vulnerabilities and Exposures (CVEs) that cannot be immediately remediated. It must justify risk mitigation strategies and is subject to higher scrutiny by Game Warden and government reviewers.
- **Game Warden Authority to Operate Documentation** – A copy of Game Warden's signed ATO, issued by an Authorizing Official (AO), specific to the Impact Level (IL) targeted by your deployment.
- **Optional Documentation** – You may include additional supporting materials, such as an existing ATO from another AO or further compliance evidence relevant to your system.

  Info

  Game Warden engineers help generate many of these artifacts as part of the security hardening and image scanning process.

When are updates required?

You should update your Deployment Passport and BoE under the following conditions:

- Annually, or
- Whenever a major release occurs, such as:
  - Adding or removing containers
  - Introducing new services or external connections
  - Modifying your Authorization Boundary Diagram

After updating your BoE, the Game Warden Security team will generate the remaining components, including updated scan results.

Note

A new Deployment Passport is not required for minor releases, such as:

- Updating existing containers
- Changing internal networking
- Applying non-breaking patches

FAQs

How to define a major vs. minor release?

A **major release** involves changes to your application that require review and approval by the Authorizing Official (AO) and an updated Deployment Passport. Examples include:

- Changes to the Authorization Boundary Diagram, such as adding or removing containers
- Adding new services
- Modifying external communications

A **minor release** includes updates that do not require a new Deployment Passport, such as:

- Changes to internal container networking
- Updating existing containers

How does the CtF/Software Approval process work for new releases?

If your new release stays within the scope of your approved Deployment Passport and follows the Game Warden Software Development Life Cycle (SDLC), the existing CtF/Software Approval can often be amended rather than reissued.

To qualify, your application must still meet the compliance requirements defined by the Director of Security and the Authorizing Official (AO).

If these conditions are met, the updated version of your application can be added to the existing CtF/Software Approval without requiring a full re-approval.

The Game Warden security team will review the release and determine whether a new CtF/Software Approval is necessary based on the scope and impact of your changes.

Game Warden Account Setup for DoD Deployments

Game Warden enables you to deploy your application to the Department of Defense (DoD) network using a secure, managed deployment environment with built-in compliance support.

To access and deploy applications to DoD networks through Game Warden, you must first establish a **Platform One SSO (P1 SSO) account** for secure authentication. This guide walks you through setting up your account and user access for environments authorized at specific DoD Impact Levels (ILs).

> **AFWERX Access Requirements by IL**
>
> - **IL2**: Requires only a P1 SSO account.
> - **IL4 & IL5**: Require both a P1 SSO account **and** a government access card (CAC/PIV/ECA) linked to that P1 account.
> - **IL4 access from outside the NIPRNet**: Requires Appgate SDP.
> - **IL5 access from outside the NIPRNet**: Requires Appgate SDP and may also require running additional compliance scripts.

What is P1 SSO?

P1 SSO is the secure authentication gateway that enables users to sign in once and access multiple P1-hosted services and applications. By leveraging P1 SSO, we can streamline user login processes across various tools within the Platform One ecosystem, ensuring both convenience and compliance with stringent security protocols.

P1 SS0 also supports:

- Integration with government-issued access cards (CAC, ECA, PIV)
- Multi-factor authentication (MFA)
- Centralized user identity management (via Keycloak)

After completing the Game Warden technical screening process, all users in your organization must create a P1 account to access applications within DoD environments through the Game Warden app.

> **Info**
>
> P1 SSO is one of the many tools offered by Platform One to support secure, scalable, and efficient software delivery across the DoD. See Platform One for more information.

Create a P1 SSO account

> **Tip**
>
> We recommend completing the following steps using a desktop browser such as Firefox or Google Chrome for optimal security and compatibility.

Step 1 - Create your P1 account

1. Go to the P1 Registration page and complete the user registration form with the following info:
   - Your first name and last name.
   - Select the organization your affiliation belongs to; otherwise, select **Other**.
   - Select your rank or pay grade. Military ranks are grouped by service branch, and civil service grades range from AA to SCS. Select **N/A** if none apply.
   - Enter your organization name.
   - Enter your organization's location (at minimum, the city and state).
   - Enter a username and your work email address.

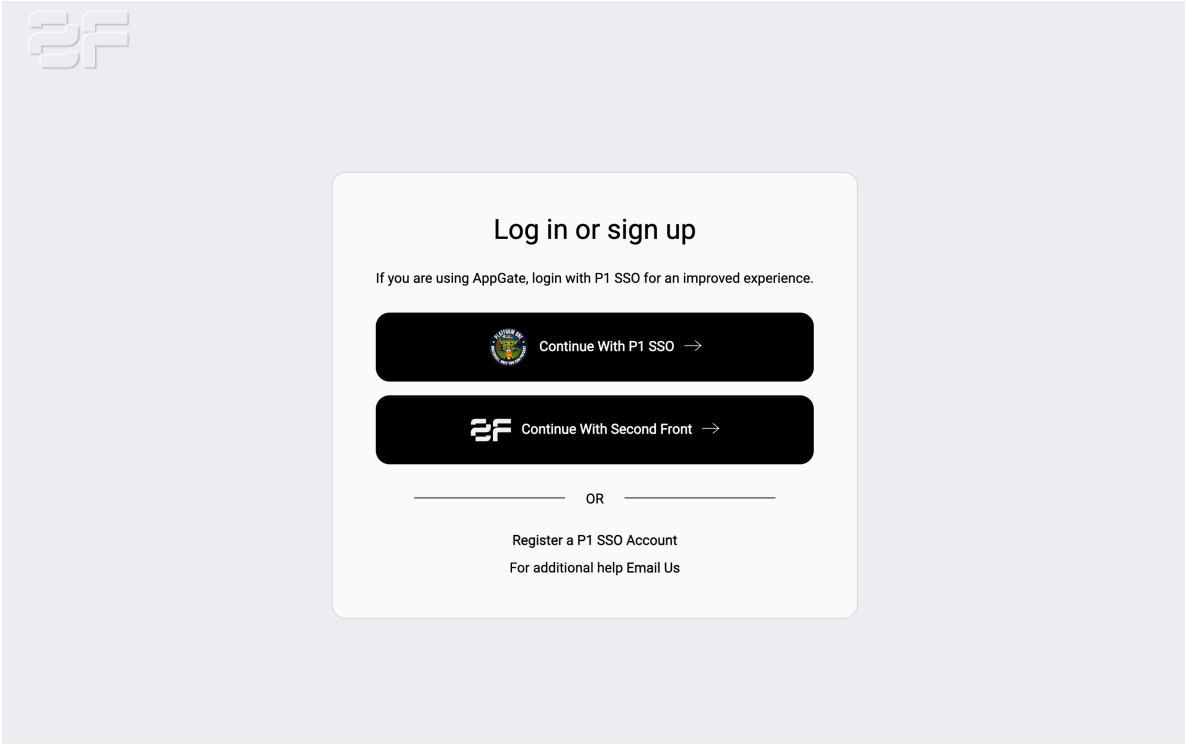
- Optionally, provide an access note. This helps administrators assign the appropriate access.
- Create and confirm your password. **Note**: If you are registering with a Common Access Card (CAC), External Certification Authority (ECA), or Federal Personal Identity Verification (PIV), creating a password is optional—especially if you plan to access your account exclusively with one of these government-issued cards. For more details, see Government Access Cards and Link Access Card with Platform One Account.

2. Click **Register**.
3. When prompted, complete the multi-factor authentication process:
    1. Install an authentication app (e.g., FreeOTP, Microsoft Authenticator, or Google Authenticator) on your mobile device.
    2. Open the app and scan the QR code displayed on the screen.
    3. Enter the current code from your authenticator app into the **Six digit code** field, and optionally provide the device name to help manage your OTP devices.
    4. Click **Submit**.
4. Review the consent form and click **Accept** to continue.
5. **Check your work email for a verification link. Click the link or paste it into your browser to verify your email. Note that the link will expire in three days. If the verification link has expired, please email P1 support at AFLCMC.HNCX.Helpdesk@us.af.mil to request a new verification email.**
6. On the **Edit Account** page, confirm your account's information, then click **Save**.

Step 2 - Verify access vis Keycloak

Keycloak is the Identity and Access Management (IAM) system used by Game Warden. Follow below steps to verify your access:

1. Visit login.afwerx.dso.mil, then click **Log in with P1 SSO**.

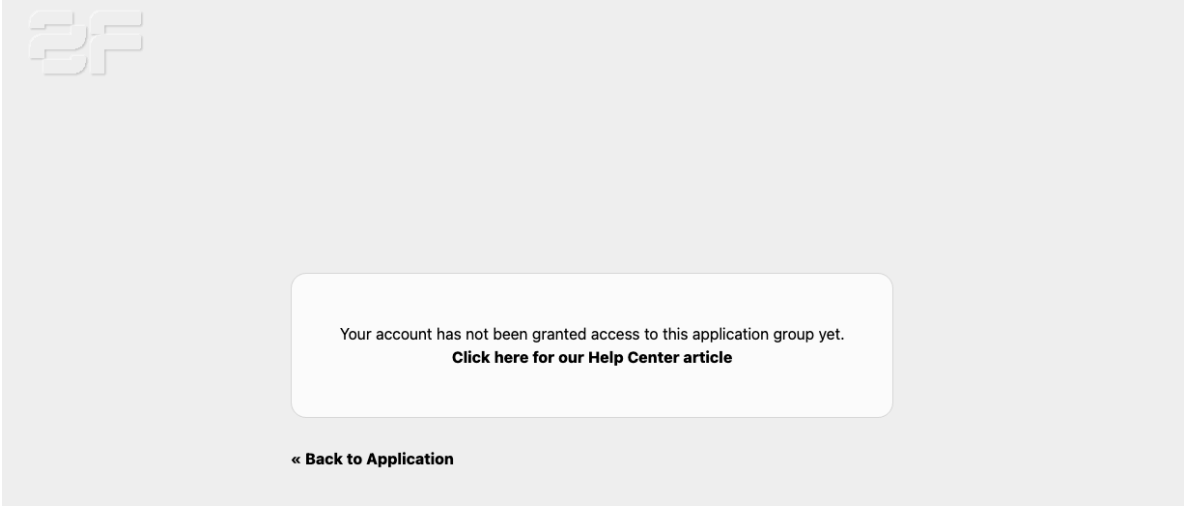


2. Enter your P1 credentials, then click **MFA Log In** .
3. Enter your MFA code, then click **MFA Log In** .

4. Review the consent form and click **Accept** to continue.
5. On the **Keycloak Personal Info** page, confirm your information and click **Save**.

3. Join your company profile on Game Warden

1. Go to Game Warden app and log in with your P1 SSO credentials.
2. After successfully login, you should see the below screen:

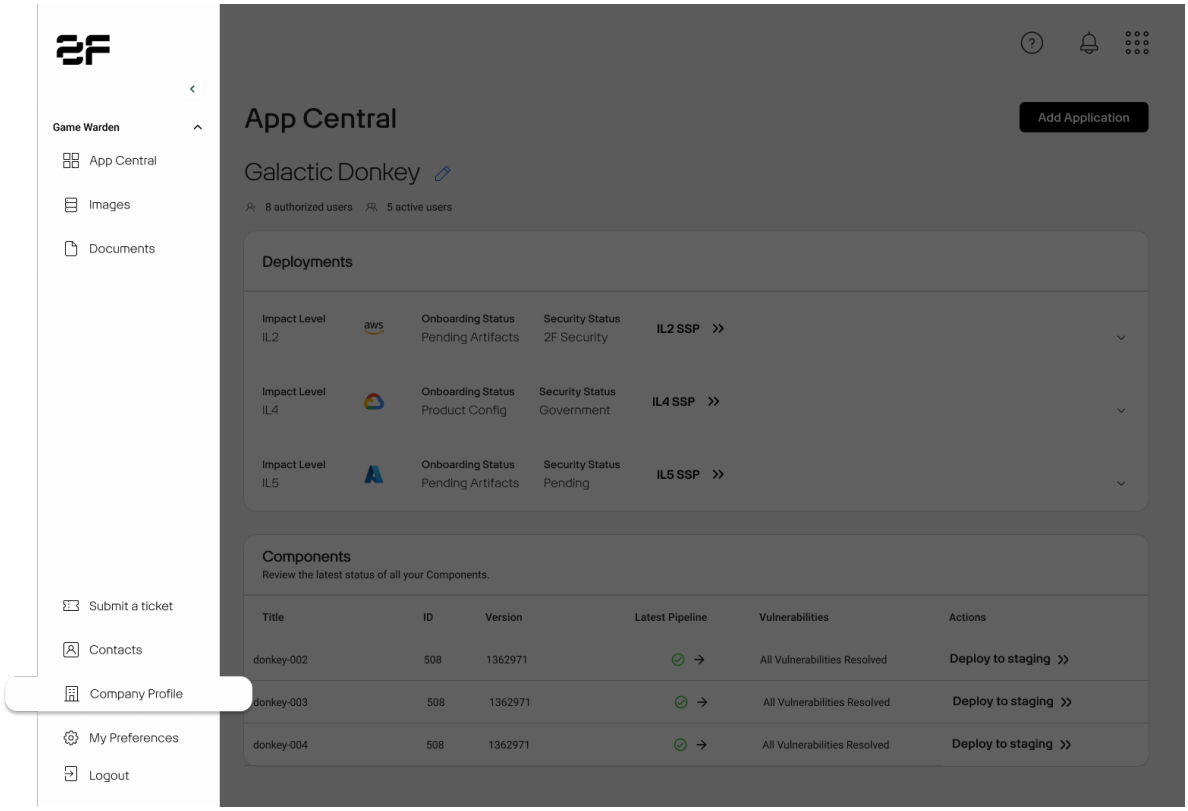


Notify Second Front so we can add you to your organization profile.

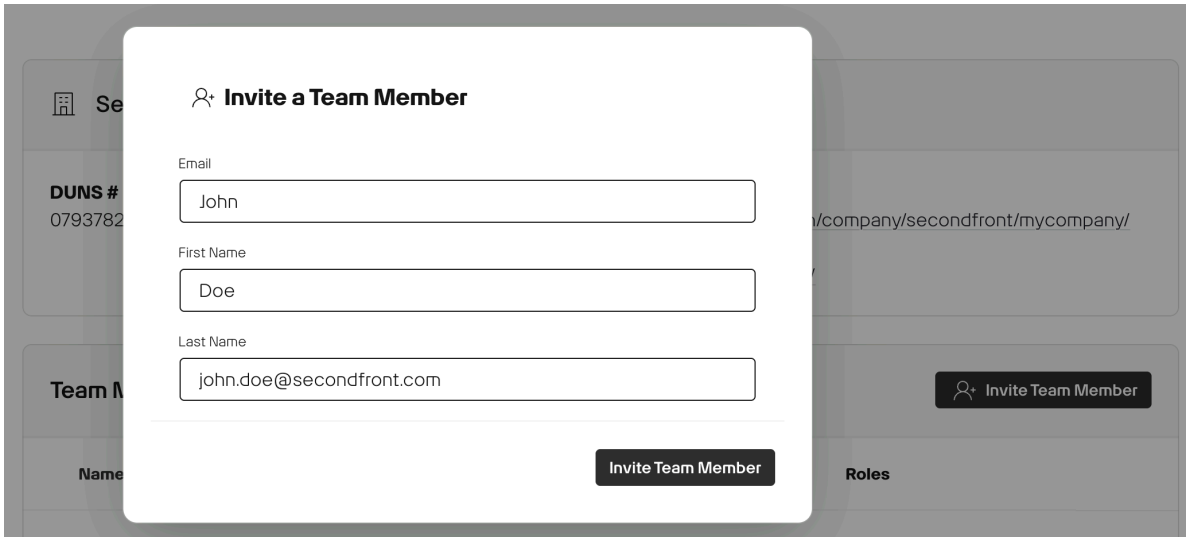
Add your team to Game Warden

Once your user account is set up, you can invite your team to Game Warden from within the app.

1. Go to the **Company Profile** tab in the left sidebar.



2. Click **Invite Team Member**.
3. Fill in each team member's email and name, then click **Invite Team Member**.



Your team member will receive an email invitation to join the Game Warden app. All team members you invite to Game Warden **must** first establish a P1 account before logging in.

Warning

If your team member's email domain is different from yours, contact your Technical Implementation Manager (TIM) for assistance.

Add Customer Admin privileges to team members

If you have Customer Admin privileges, you can grant the same access level to other team members. Follow these steps:

1. Locate the team member in your user list, then click the **kebab menu** next to their name.
2. Select **Edit User** from the dropdown.
3. In the **Edit a Team Member** modal, navigate to the Roles section and click **Make user a Customer Admin**.
4. Click **Submit** to save the changes.

The team member will immediately receive Customer Admin permissions and can perform all administrative functions within your organization.

Customer and Customer Admin permissions

Capability	Customer	Customer Admin
Submit Tickets	Yes	Yes
Access ScanLab	Yes	Yes
Download ScanLab artifacts	Yes	Yes
Delete deployments	No	Yes
Set up Harbor credentials	No	Yes
Fill out Body of Evidence forms	Yes	Yes
Resolve vulnerabilities	Yes	Yes
Deploy images	No	Yes

Troubleshooting common P1 access issues

Lost device or need to reset MFA?

Re-scan a new QR code during sign-in.

MFA code not accepted?

The code may have expired. Re-scan the QR code to generate a new one.

Password reset emails

- Didn't receive one? Try resending and check your spam folder.
- Still not working? Email Platform One at aflcmc.hncx.p1cst@us.af.mil.

Disabled P1 account

Email Platform One at aflcmc.hncx.p1cst@us.af.mil with the words "unlock", "disable", or "reactivate" in the message body. Your account will unlock automatically, and you'll get an email confirmation.

How do I change the email address associated with my P1 SSO account?

To update your email address, go to the P1 Account page, modify the email field, and click **Save** to confirm the change.

Your account has not been granted access

This message usually means your account is missing the necessary group permissions to access Game Warden applications.

If you've registered with a Common Access Card (CAC), External Certification Authority (ECA), or Federal Personal Identity Verification (PIV), follow these steps to resolve the issue and clear any cached credentials:

1. Open Google Chrome in Incognito Mode with your card inserted.
2. Go to https://login.dso.mil. When prompted, associate your card with your Platform One (P1) account.
3. Go to http://login.afwerx.dso.mil. When prompted, associate your card with your Game Warden account.
4. Return to the original Game Warden application URL and try logging in again.

Oops, your session has expired. Please try again.

This error occurs when a previous login session is cached in your browser. To resolve it, go to your browser's address bar, delete the `.mil` portion of the URL, retype it, and press **Enter**. This should reload the login page and prompt you to sign in with your Platform One credentials.

If the error persists, repeat this process up to two more times. If you're still unable to log in, reach out by submitting a support ticket.

My attempt to log into Game Warden was halted by the "Verify your email to start using" requirement screen

This message indicates that you have not yet verified the email address associated with your P1 account creation. If the verification link has expired, please email P1 support at AFLCMC.HNCX.Helpdesk@us.af.mil to request a new verification email.

Game Warden account disabled

If you haven't logged in for **30 days or more**, your Game Warden account may be automatically disabled for inactivity. You'll receive an email with instructions on how to restore your account.

If you need help reactivating your account, contact Second Front vis Slack.

Platform One Useful Links

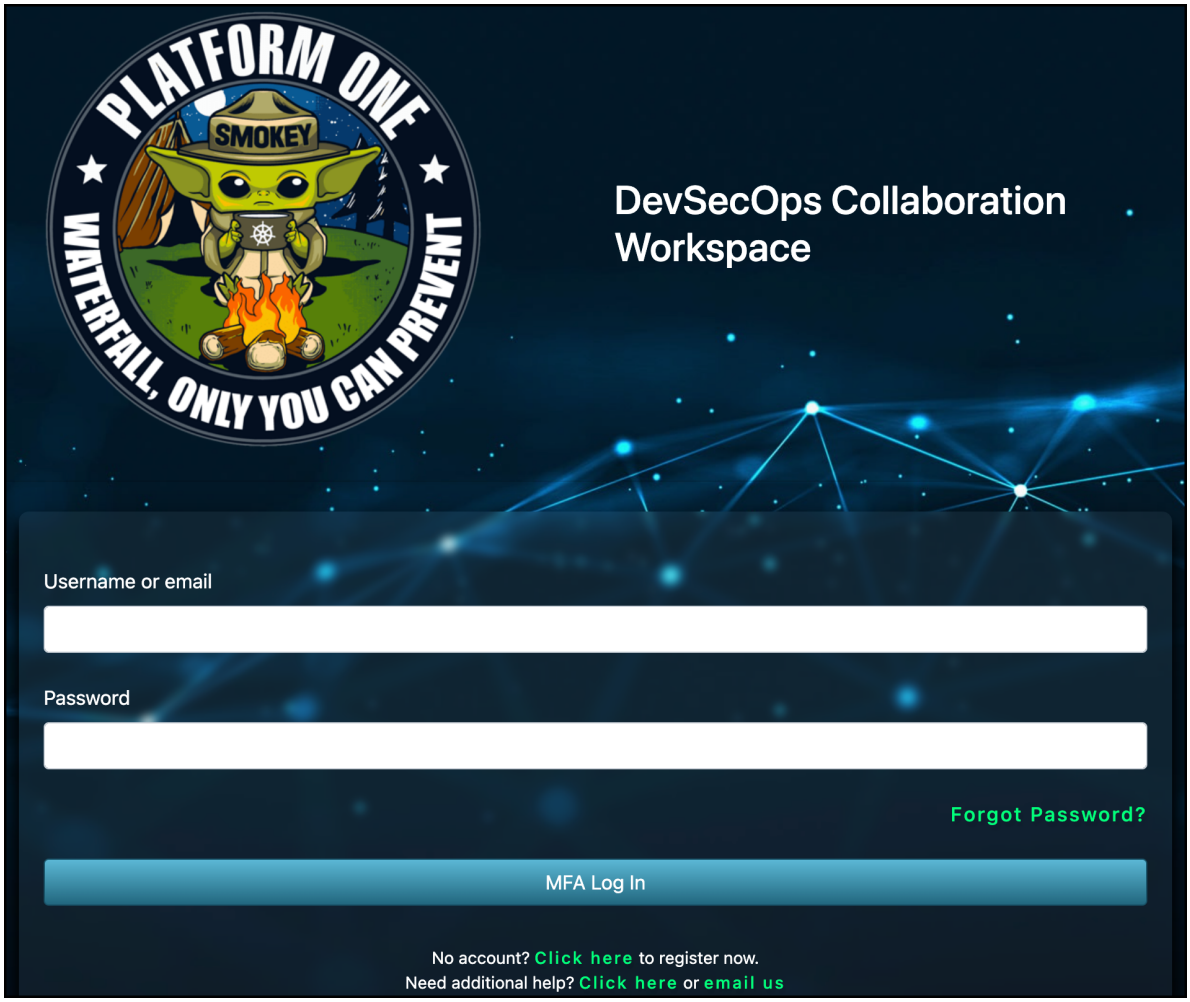
- Support

Reset Password and Email Update

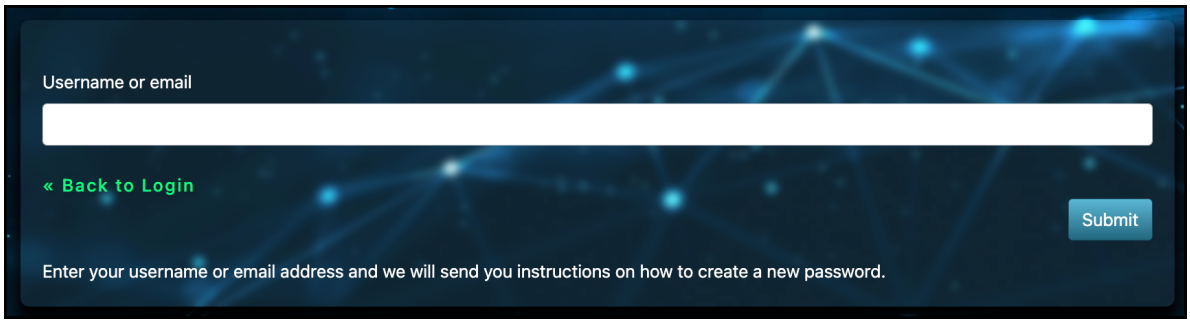
If you've forgotten your Game Warden password or recently changed your email address, this guide will walk you through how to restore access to your account.

Reset password

1. From the Platform One login page, click **Forgot password?**.



2. A new page will appear asking for the email address linked to your Game Warden account. Enter your email address and click **Submit**.



3. Check your email inbox for a password reset message. Follow the instructions in the email to create a new password.

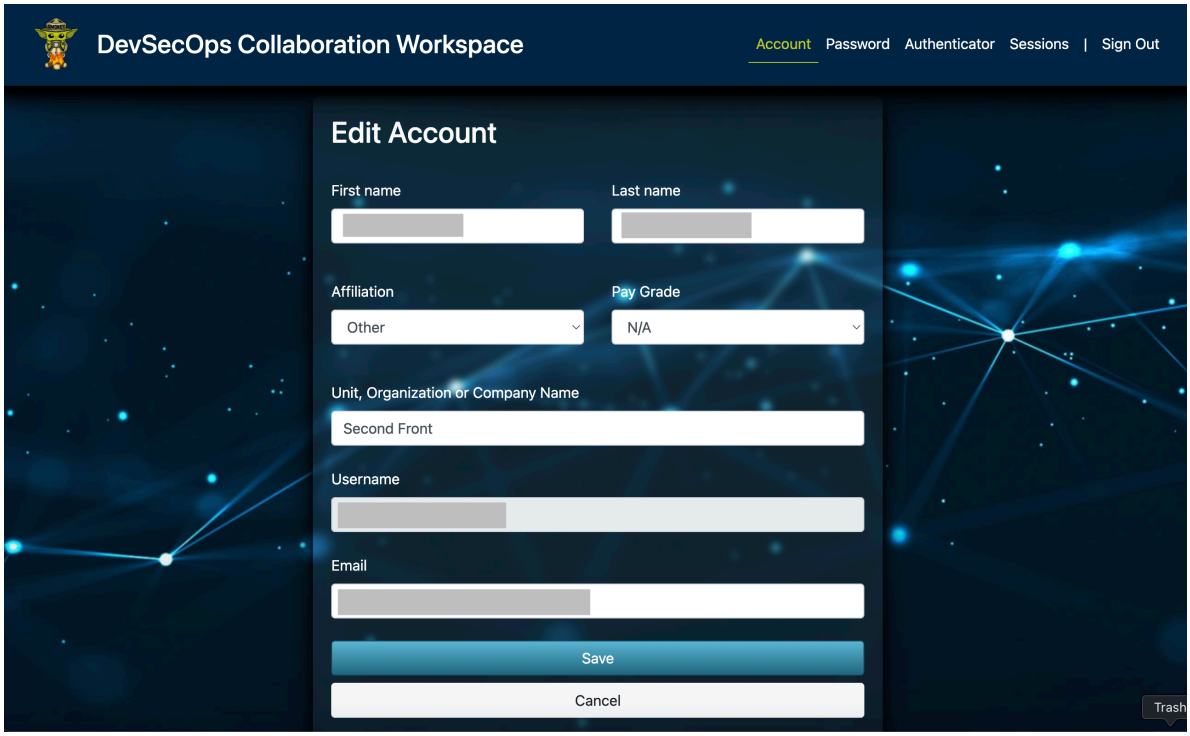
   Tip

   - If you don't see the message, check your Spam or Junk folder.
   - If you don't receive a password reset email, it may mean the email you entered is not linked to a Game Warden account. For assistance, contact Second Front via Slack.

Update your email address

If your email address has changed (e.g., due to a company domain migration), follow these steps to update your P1 account:

1. Log in to your Platform One (P1) account.
2. On the **Edit Account** page, update the **Email** field with your new address.



3. Click **Save**. Your updated email will automatically sync across Game Warden systems.

Troubleshoot Access to Game Warden App from NIPRNet

This guide helps diagnose and resolve issues when a user on NIPRNet is unable to access Game Warden-hosted applications. These steps are especially helpful if a user reports timeouts or sees a blank/white screen when attempting to reach an application endpoint.

NIPRNet is a secure network used by the DoD to handle unclassified information.

Pre-requisites

Before beginning any troubleshooting, ensure the user has:

- Created a **Platform One SSO (P1-SSO)** account.
- Configured their **CAC/ECA/PIV** credentials for access to IL4+ environments.
- Successfully associated their **access card with the Game Warden Keycloak** instance.

Step-by-step network and access tests

Local Firewall test

Navigate to: https://login.dso.mil

Results:

- *Success*: Outbound connection is working. Proceed to the next test.
- *Failure*: Contact your local network support to allow access to this domain.

AFWERX test

Navigate to: https://grafana.il4.afwerx.dso.mil

Results:

- *Success*: Game Warden services are online and accessible.
- *Failure*: There may be an issue with AFWERX or Game Warden availability. Contact support.

P1 / CNAP test

Navigate to: https://code.il4.dso.mil

Results:

- *Success*: You should see a message similar to: "Your account has not been granted access to this application group yet." This indicates that CNAP pass-through is confirmed. Proceed to the AFWERX test.
- *Failure*: CNAP access is likely not whitelisted.

If CNAP access fails, you'll need to gather your agency's IP address information for whitelisting. Work with your **network administrator** to obtain all applicable IP ranges in **CIDR format**.

How to find your IP address

Depending on your system and security settings, use one of the following:

Use Command Prompt:

1. Press **Windows Key + R**, type `cmd`, and press **Enter**.
2. Type `ipconfig` and press **Enter**.

3. Locate and copy your **IPv4 Address**.

Use Powershell:

1. Press **Windows Key + R**, type `powershell`, and press **Enter**.
2. Type `ipconfig` and press **Enter**.
3. Locate and copy your **IPv4 Address**.

Need further help? Reach out by submitting a support ticket.

Platform One

Platform One (P1) is a DevSecOps platform managed by the U.S. Department of Defense (DoD). It provides a set of standardized tools, services, and infrastructure to support the secure development and deployment of software within DoD environments. P1 is commonly used by DoD teams and approved contractors to meet compliance requirements across various classification levels.

Game Warden integrates with P1, enabling access to DoD-approved tooling for secure software development, including key solutions such as Big Bang (a pre-configured Kubernetes platform) and Iron Bank (a hardened container image repository).

P1 components

Big Bang

Big Bang is the DoD-approved Kubernetes-based architecture that underpins both the Game Warden platform and the applications it hosts. It embodies Infrastructure as Code (IaC) principles, using immutable code to provision and manage DoD-compliant infrastructure.

Because Big Bang adheres to strict security and compliance standards, applications deployed within it are eligible for an associated Authority to Operate (ATO). All containers deployed by the Game Warden team into the Kubernetes cluster must conform to Big Bang protocols and configurations.

Iron Bank

Iron Bank is Platform One's hardened container image repository. It stores containerized images that have been security-hardened and scanned to align with DoD cybersecurity standards.

Containers sourced from Iron Bank are approved for deployment within Game Warden and other DoD environments, supporting secure, consistent, and compliant software delivery.

Party Bus

Party Bus is another secure DevSecOps platform offered within the Platform One ecosystem. It is similar in scope to Game Warden, with comparable capabilities for deploying software to DoD-authorized environments. Both platforms leverage P1 infrastructure, though they may serve different mission partners or deployment workflows.

Access P1

To access P1 and its associated tooling, you must first establish a **P1 SSO account** for secure authentication. See Create a P1 SSO account for setup instructions.

FAQs

How are Game Warden and P1 related from an infrastructure perspective?

Although Game Warden's infrastructure and platform are independently managed and distinct from P1, we maintain several key integration points through P1 services:

1. **Shared Tech Stack** – Game Warden incorporates components of the P1-approved Big Bang stack within our own platform architecture, aligning with DoD security and deployment standards.
2. **Network Access** – We connect to the DoD network via the P1-managed Cloud Native Access Point (CNAP), which provides our IP space and enables the use of a `.mil` domain.

3. **Federated Authentication** – Game Warden federates with the P1-managed Single Sign-On (SSO) solution to support Common Access Card (CAC) authentication for secure user access.

These integration points allow Game Warden to maintain DoD compliance while operating as a standalone platform.

When transitioning from P1 to Game Warden, can we perform QA testing during the migration?

Yes. Customers performing a direct lift-and-shift from P1 to Game Warden will have the opportunity to validate functionality within both the developing (DEV) and staging (STG) environments prior to production (PRD) deployment.

Access Control in Game Warden for DoD Deployment

Game Warden leverages **Keycloak** for identity and access management, integrated with **Platform One Single Sign-On (P1 SSO)** to authenticate users across its platform. While Game Warden centrally handles authentication, application owners are responsible for implementing authorization—ensuring only the right users can access specific resources within their application.

This guide walks you through how authentication and authorization work across the platform, and what steps your organization must take to ensure secure access control.

Keycloak and P1 SSO

Keycloak is integrated with P1 SSO, allowing users to sign in once using their government-issued credentials (e.g., CAC, ECA, or PIV) and access all Game Warden services. When users complete the onboarding process with P1, their Keycloak account is created automatically.

Default access behavior and application-level control

By default, any user with a valid P1 SSO account and knowledge of your application's environment URL (DEV, STG, PRD) can access your application unless explicit restrictions are implemented. **Game Warden does not automatically restrict user access based on roles or affiliation**. Access control must be implemented by the application using the JWT token provided after authentication.

> **Note**
>
> For apps hosted at IL4 or IL5, end-users must have a valid Government Access Card linked to their P1 SSO account.

How JWT authorization works

After authenticating through P1 SSO, Keycloak issues a JSON Web Token (JWT) containing verified user identity claims.

When integrating your applications with Game Warden, two primary actions occur:

- **Authentication (AuthN)** – Keycloak authenticates the user.
- **Authorization (AuthZ)** – Keycloak determines access level based on group membership and PKI credentials.

After Authservice validates a request, it injects a **Bearer token** in the Authorization header of the user's request. This JWT is signed by Keycloak and includes identity and authorization claims. Any pre-existing Authorization header will be replaced.

Your application is responsible for:

1. Receiving the JWT from Keycloak.

This JWT is injected by Authservice into the `Authorization: Bearer` header of each request.

2. Validating the JWT signature and expiration.

This step is optional if you prefer Authservice to handle JWT validation and token expiration. If so, reach out to your Second Front representative to request this setup.

JWTs are signed by Keycloak and should be verified using a trusted public key.

Example encoded JWT:

```
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJ6IElmYlVwWlBfWlhlWnlhWlhWbkZWWlBWUkl3QUllZjJhWjJ
eyJleHAiOjE2NjIwNTYzMTUsImlhdCI6MTY2MjA1NjAxNSwiYXV0aF90aW1lIjowLCJqdGkiOiIxMTRiYTIwOS1kMjBmLTRmYTI
A048UC_EFnA8ZC_nQL-I8IF5Xm9V6bb9JNQAUX4Ubg
```

3. Extracting claims (e.g., user roles, groups)

Example decoded JWT with payload data:


```
{
    "exp": 1662056315,
    "iat": 1662056015,
    "auth_time": 0,
    "jti": "114ba209-d20f-4fa2-ad33-84119ae12bf8",
    "iss": "https://keycloak.dev.il2.gamewarden.io/auth/realms/gamewarden",
    "aud": [
        "112_a2941a91-a86d-4440-b4b8-55c9feb30573_rain_drop"
    ],
    "sub": "fb435032-4fe7-4dab-879d-0021afa6a02e",
    "typ": "ID",
    "azp": "112_a2941a91-a86d-4440-b4b8-55c9feb30573_rain_drop",
    "session_state": "ab53d0a9-09b4-426b-840b-de3c34605d58",
    "sid": "ab53d0a9-09b4-426b-840b-de3c34605d58",
    "email_verified": true,
    "name": "JOHN DOE",
    "groups": [
        "/EXAMPLE-ORG/IL4/ROLES/ADMIN",
        "/EXAMPLE-ORG/IL4/ROLES/ADMIN",
        "Impact Levels/IL2 Authorized",
        "Impact Levels/IL4 Authorized",
        "/EXAMPLE-ORG/IL4/ROLES/USER"
    ],
    "preferred_username": "john.doe",
    "given_name": "JOHN",
    "family_name": "DOE",
    "email": "john.doe@example.com"
}
```


4. Applying authorization logic to control access, ensuring users are only given access to appropriate views or APIs.

5. Enforcing session expiration and re-authentication policies.

This enables you to build role-based or attribute-based access controls directly into your application.

Note

In a typical OpenID Connect (OIDC) flow, your application would initiate login by redirecting the user to Keycloak. Once authenticated, Keycloak redirects the user back to your app with a token. Your app would then use this token to manage sessions and authorization headers. In Game Warden, this flow is handled automatically behind the scenes. Your app receives requests only after the user is authenticated, eliminating the need for manual OIDC session setup.

When your app receives a request from a logged-in user, Authservice automatically injects the **Authorization: Bearer header** into that request. This JWT is signed by Game Warden's Keycloak and contains the user's identity and role/group claims.

How to Validate JWTs

To validate the JWT on your application side:

1. Extract the token from the Authorization header (e.g., Authorization: Bearer eyJ…).
2. Decode and validate it using a standard JWT library for your language (e.g., PyJWT, jose, etc.).
3. Verify the signature using the public key from Keycloak.

If you're using Python, review below resources for details on how to complete the validation:

- PyJWT usage documentation
- Video walkthrough of JWT validation in Python

Tip

Use the `sub` (subject) field in the JWT to identify users. This is a unique Keycloak subject ID (in GUID format) that remains consistent even if the user's email, name, or affiliation changes. It ensures

Integrate OIDC Clients with Keycloak on Game Warden

This guide walks you through integrating directly with our Identity Provider (IdP), **Keycloak**, deployed via **Platform One's Big Bang**. It covers how to request credentials, configure your application, map claims/roles, and implement login, refresh, and logout flows using OpenID Connect (OIDC).

> **Who should use this guide?**
>
> Customers building their own OIDC client (instead of using Authservice)

Terminology & environment

Terminology

Term	Meaning
IdP (Keycloak)	The identity provider that issues authentication tokens.
Realm	An authentication namespace in Keycloak (e.g., `gamewarden`).
Client	An application registered in Keycloak that consumes OIDC.
Confidential Client	Used by backends/services that can safely store a client secret.
Public Client	Used by SPAs/browser-based apps; does **not** store a secret and must use **PKCE**.

Base URLs

When configuring OIDC, your application needs to know *where* to retrieve Keycloak's public configuration and *which issuer to trust* when validating tokens.

Keycloak exposes two important URLs:

1. OIDC Discovery (Well-Known) URL: This is the endpoint your application calls to automatically retrieve token endpoints, signing keys, and supported claims. Your OIDC library will typically use this URL directly or indirectly.

```
https://<idp-host>/auth/realms/<YOUR_REALM>/.well-known/openid-configuration
```

> **Example Commercial environment**
>
> `https://login.secondfront.com/auth/realms/gamewarden/.well-known/openid-configuration`

2. Issuer URL (must match exactly): The issuer is the value embedded inside the tokens issued by Keycloak. Your application must trust this value exactly, or token validation will fail.

```
https://<idp-host>/auth/realms/<YOUR_REALM>
```

> **Important**
>
> Keycloak **always includes** `/auth` in the issuer path. If your OIDC library expects an issuer without `/auth`, tokens will be rejected with errors such as:
>
> - *Issuer mismatch*
> - *JWKS not found*
> - *Token signature validation failed*

Integration steps

Step 1 - Requesting access

Submit a request to Second Front with the following details:

- **Client name** (as you want it to appear in Keycloak)
- **Redirect URIs** for authentication (e.g., Production and Staging)
- **Application type** (Web server, SPA, Native, or Machine-to-Machine)
- Whether the client should be **Public** (no secret, uses PKCE) or **Confidential** (requires client secret)
- Any required **claims** (e.g., `email`, `groups`, custom attributes)
- If needed, **token lifetime adjustments** (access, refresh, offline tokens)

Second Front will provision a **Keycloak Client** in the appropriate shared realm and provide you with:

- **Client ID** (and **Client Secret**, if applicable)
- **OIDC Discovery URL** (well-known configuration endpoint)
- Any **scopes**, **mappers**, or **group-based access controls** that were configured for the client

Note: Access can be restricted to your organization's group space (e.g., `/Customers//**`). See Claims, scopes, and group/role mapping for more details.

Step 2 - Choosing your Client Type

App type	Client type	Auth flow
Server-rendered Web	Confidential	Authorization Code **(+ PKCE recommended)**
SPA (React/Vue/etc.)	Public	Authorization Code + **PKCE (required)**

Step 3 - Configure your Client (Keycloak Admin steps)

> **Note**
>
> These steps are performed by Second Front during client provisioning. They are included here for visibility and review.

1. **Create Client** (Keycloak → *Clients* → *Create*):
   - **Client Type:** OpenID Connect
   - **Client ID:** `us_<UUID>_<CUSTOMER_NAME>`
   - **Client Protocol:** `openid-connect`
   - **Access Type: Confidential** (or **Public** for SPA / native)
   - **Valid Redirect URIs:** Provide exact callback URLs (use wildcards sparingly)
   - **Post-Logout Redirect URIs:** Provide exact return URLs after logout
2. **Credentials** (Confidential Clients Only): Generate a **Client Secret** (stored server-side only).
3. **Login Settings**:
   - **Standard Flow (Authorization Code):** ON
   - **Implicit Flow:** OFF (deprecated)
   - **Direct Access Grants (Password Grant):** OFF
   - **Service Accounts:** ON (only if client-credentials flow required)
   - **PKCE: Required** for Public clients (SPA, mobile, desktop)
4. **Client Mappers**:
   - Add claim mappers to include attributes such as `email`, `roles`, or organizational groups.
   - See Claims, scopes, and group/role mapping for mapping guidance.

5. **Advanced Settings**: Token lifetimes may be adjusted if needed; otherwise realm defaults apply.

   Note: In certain government environments, token lifetimes may be fixed and cannot be altered.

Step 4 - Application-side configuration

Use the **Discovery URL** to auto-configure endpoints.

```
{
    "issuer": "https://<idp-host>/auth/realms/gamewarden",
    "client_id": "us_<UUID>_<CUSTOMER_NAME>",
    "client_secret": "<SECRET-IF-CONFIDENTIAL>",
    "redirect_uri": "https://<your-app>/oidc/callback",
    "post_logout_redirect_uri": "https://<your-app>/",
    "response_type": "code",
    "scope": "openid profile email",
    "use_pkce": true
}
```

Auth Code Flow (with PKCE)

1. Redirect the user to the `authorization_endpoint` with the following parameters:
   - `client_id`
   - `redirect_uri`
   - `response_type=code`
   - `scope`
   - **PKCE** parameters:
     - `code_challenge`
     - `code_challenge_method=S256`
2. Exchange the authorization `code` at the `token_endpoint` with your `code_verifier` to receive:
   - `id_token`
   - `access_token`
   - `refresh_token` (optional)
3. Validate the **ID Token**:
   - Verify its signature using the realm's **JWKS** endpoint from OIDC discovery.
   - Confirm that the `aud` (audience) and `iss` (issuer) claims match your client configuration.

Client Credentials Flow

```
POST token_endpoint
  grant_type=client_credentials
  client_id=...&client_secret=...
```

Claims, scopes, and group/role mapping

- **Standard OIDC**: `sub`, `iss`, `aud`, `exp`, `iat`
- **Profile/Email**: `email`, `email_verified`, `given_name`, `family_name`, `preferred_username`
- **Groups**: `groups` (e.g., `/Customers/<CUSTOMER_NAME>/developer`)
- **Custom attributes**: e.g., `customer_id`, `employee_number` (may be added at request of customer)

Example ID Token (truncated)

```
{
  "exp": 1762360748,
  "iat": 1762360448,
```

```
  "jti": "12345678-1234-1234-1234-123456789123",
  "iss": "https://login.secondfront.com/auth/realms/gamewarden",
  "aud": "us_98fhdje8-kld4-8d23-aa11-mgja92jfh456_example-customer",
  "sub": "87654321-4321-4321-4321-321987654321",
  "typ": "ID",
  "azp": "us_98fhdje8-kld4-8d23-aa11-mgja92jfh456_example-customer",
  "sid": "dc23c6a8-280c-4168-aa92-452c235b27a3",
  "acr": "1",
  "email_verified": false,
  "name": "Example User",
  "groups": [
    "/Gamewarden/Users",
    "/Customers/example-customer/developers",
  ],
  "preferred_username": "example.user",
  "given_name": "Example",
  "family_name": "User",
  "email": "example.user@example.com"
}
```


Restricting group visibility to a customer

We can restrict the groups included in issued tokens so that users only receive groups under: `/Customers/<CUSTOMER_NAME>/...`.

This ensures your application only sees roles and permissions relevant to your organization.

> **Note**
>
> If your application requires additional custom claims (e.g., user role, tenant ID, or feature flags), please provide the claim names and expected data types and our team will configure the appropriate mappers.

Token lifetimes & refresh strategy

Game Warden environments follow security-first token lifetimes aligned with DoD and industry best practices:

- **Access Token**: Short-lived (**5 minutes**) — limits the impact of token leakage.
- **Refresh Token**: Longer-lived (**30 minutes**) and **rotates on each use** — reduces replay risk.
- **SSO Session Duration**: Typically **up to 10 hours** before requiring re-authentication.

These defaults balance strong security controls with a smooth user experience. If your application requires different lifetimes, discuss with our team—adjustments may be possible depending on your deployment environment and compliance level.

> **Best practice**
>
> Use **short-lived access tokens** paired with **rotating refresh tokens** to minimize risk if a token is compromised.

Logout & session management

When users sign out, the application should trigger **RP-Initiated Logout**, which logs the user out of both your application **and** the Keycloak SSO session.

RP-initiated logout (recommended)

Call the realm's `end_session_endpoint` with:

- `id_token_hint` — the **ID Token** issued during login
- `post_logout_redirect_uri` — a **pre-registered** URL where the user should be redirected after logout

Example call

```
GET https://<idp-host>/auth/realms/<YOUR_REALM>/protocol/openid-connect/logout
  ?id_token_hint=eyJ...
  &post_logout_redirect_uri=https%3A%2F%2F<your-app>%2F
```

> **Important**
>
> The value of `post_logout_redirect_uri` **must** exactly match one of the URIs listed in the client configuration.

Front-channel vs back-channel logout

Keycloak supports multiple logout notification mechanisms:

Mode	How it Works	Use When
Front-channel logout	Keycloak loads each client's logout URL in the browser.	Your app can handle logout via browser redirect.
Back-channel logout	Keycloak sends a server-to-server logout request.	Your app has a backend endpoint for logout processing.

If your application expects to be notified when the user logs out **from another session**, provide:

- A **logout endpoint URL**
- Expected authentication/validation behavior for the logout request

> **Troubleshooting**
>
> If you see a **Keycloak confirmation page** during logout, check:
>
> - `id_token_hint` is **missing or expired**
> - `post_logout_redirect_uri` is **not in the allowed list**
>
> Adding both parameters correctly will produce a silent, seamless logout experience.

Security guidance

- **Do not store client secrets** in browser-based or mobile applications. Use **Public** clients with **PKCE** for SPAs and native apps.
- Always validate **Issuer** and **JWKS**:
  - Issuer **must** match realm URL *including* `/auth`
  - Validate token signatures against the realm's JWKS
  - Verify `aud`, `iss`, `nonce`, and token expiration
- Use **state** + **nonce** with Authorization Code Flow and verify both during the callback to prevent CSRF and replay attacks.
- Apply **least-privilege** access:

  - Request only the scopes and claims your application actually needs.
  - Avoid broad group claims unless necessary.
- Consider **refresh token rotation** and server-side session enforcement for added protection.

Troubleshooting

Symptom	Likely cause	Fix
`could_not_discover_issuer` / *"Not found"*	Wrong issuer URL (missing `/auth`)	Use `https://<idp-host>/auth/realms/<`
`invalid_grant` / code exchange fails	Missing/wrong `code_verifier`	Ensure PKCE end-to-end
`redirect_uri_mismatch`	Callback not whitelisted	Add exact redirect URI in client
`invalid_audience`	Token `aud` doesn't include your client	Add audience mapper / check `resource_access`
Logout shows confirmation page	No `id_token_hint`	Pass `id_token_hint` and whitelist `post_logout_redirect_uri`

Samples

Minimal auth code (server) using discovery

```
# 1) Discover endpoints
curl -s https://<idp-host>/auth/realms/<YOUR_REALM>/.well-known/openid-configuration | jq .

# 2) Browser to authorization endpoint (build URL)
# https://<idp-host>/auth/realms/<YOUR_REALM>/protocol/openid-connect/auth?response_type=code&client_id

# 3) Exchange code for tokens
curl -s -X POST \
  -d grant_type=authorization_code \
  -d code=<CODE_FROM_CALLBACK> \
  -d client_id=cust-<CUSTOMER_NAME>-<APP> \
  -d client_secret=<SECRET> \
  -d redirect_uri=https://<your-app>/oidc/callback \
  https://<idp-host>/auth/realms/<YOUR_REALM>/protocol/openid-connect/token
```

Client credentials

```
curl -s -X POST \
  -d grant_type=client_credentials \
  -d client_id=cust-<CUSTOMER_NAME>-svc \
  -d client_secret=<SECRET> \
  https://<idp-host>/auth/realms/<YOUR_REALM>/protocol/openid-connect/token
```

Support & changes

If you need to update your OIDC client configuration (e.g., add redirect URIs, rotate secrets, adjust claims), please open a support request with the following information:

- Client ID
- Environment (e.g., Staging, Production)
- Requested change
- Reason / justification

For urgent items such as incident response, credential compromise, or access revocation, contact Support and mark the request as High Priority so our team can respond immediately.

Using Government Access Cards with Game Warden

This guide provides essential information about government-issued access cards and when they're required for accessing Game Warden.

Once you have obtained a government access card, refer to Link Access Card with Platform One (P1) Account for instructions on associating your card with your P1 account for future authentication to Game Warden.

What are government access cards?

Government-issued access cards are secure credentials that use certificate-based authentication to verify a user's identity when accessing government systems and environments. These cards embed cryptographic certificates that support Private Key Infrastructure (PKI) standards for secure access.

Game Warden supports the following types of government-issued access cards:

Common Access Card (CAC)

Issued by the U.S. Department of Defense (DoD) to military personnel, government employees, and eligible contractors.

Process to acquire CAC To obtain a CAC, you must go through the DoD vetting process with the support of a government sponsor. This process can take several months to complete.

The DoD issues CACs to eligible individuals, including:

- Active-duty military personnel
- Reservists
- Federal civilian employees
- Authorized contractors

Acquisition process:

You must work directly with your government sponsor to complete the following steps:

1. Sponsorship and Eligibility Verification
2. Registration and Enrollment
3. Background Investigation
4. Card Issuance

For detailed guidance, see the Process for Acquiring or Renewing a CAC or refer to the General Information section.

> **Note**
>
> You may receive a CAC based on fingerprint results; however, final approval depends on passing the **National Agency Check with Inquiries (NACI)**. If the background check is not approved, the CAC will be revoked.

External Certification Authority (ECA)

Issued by DoD-approved commercial vendors to contractors and vendors who work with DoD systems but are not eligible for CACs.

Process to acquire ECA The ECA program provides digital certificates to eligible individuals affiliated with companies that require access to DoD systems. ECAs serve as an alternative to CACs for contractors and partners who are not eligible for a CAC.

To obtain an ECA certificate, you must complete the DoD vetting process through a DoD-approved vendor such as IdenTrust or WidePoint. The typical processing time is approximately 30 days.

Acquisition process with IdenTrust:

1. Complete the required forms and have them notarized. You must submit your notarized forms within **30 calendar days** of notarization — late submissions are invalid.
2. Submit your application and documentation to the vendor within 30 days of notarization.
3. Undergo identity verification by the vendor (typically 3–5 business days).
4. Receive your certificate by mail based on the delivery option you select.

For compatibility recommendations and important usage notes, see the ECA Certificate Compatibility Guide.

Warning

- Game Warden recommends using IdenTrust, a DoD-approved provider, for obtaining ECA certificates.
- For best compatibility with Game Warden's Keycloak Single Sign-On (SSO), we recommend selecting the **ECA Medium Token Assurance** option from IdenTrust. This token type has been validated to work with Game Warden's identity and access management system.
- Game Warden is **not affiliated with IdenTrust**. Please contact IdenTrust directly for questions regarding certificate issuance or support.
- Known compatibility considerations:
- **Mac users with M1 chips must use Firefox** when accessing government systems with an IdenTrust ECA. IdenTrust is aware of this limitation and is working to address it.
- **Linux/Ubuntu is not supported** for retrieving digital certificates from IdenTrust.
- Recommended browsers for certificate setup:
  - **Windows:** MS Edge or Google Chrome (latest versions)
  - **Mac:** Mozilla Firefox (latest version)

For complete system requirements and supported configurations, refer to the IdenTrust Certificate Compatibility Guide.

Personal Identity Verification (PIV)

Issued by U.S. federal agencies for access to federal systems.

Process to acquire PIV Federal agencies issue PIV cards to eligible individuals to provide secure access to government systems. To obtain:

- Work directly with your government sponsor to request and complete the PIV issuance process.
- Confirm that your PIV card is issued with the required certificate policies to enable access to IL4+ environments via Platform One SSO.
- Coordinate with the Game Warden team to verify your access after issuance.

Note

- Game Warden does not issue or manage government access cards. You **must** obtain them directly from a government sponsor or a DoD-approved provider.
- Users with a **CAC**, **ECA**, or **PIV** can access Game Warden environments at **IL2**, **IL4**, and **IL5** through **Platform One (P1) Single Sign-On (SSO)** - provided their access card includes one of the required certificate policies.
- Contact the Game Warden Platform team for proactive guidance before attempting access. We can assist with validating your card's configuration and ensuring it works with IL2, IL4, and IL5 environments.

Government access card comparison

The table below summarizes key requirements and acquisition processes for each supported government access card.

- CAC and PIV cards are issued directly by government sponsors.
- ECA certificates must be obtained from a **DoD-approved vendor**, such as IdenTrust, Inc.

Card Type	Estimated Wait Time	U.S. Citizenship Requirement	Cost
CAC	Up to 18 months (based on background investigation)	Not required	Consult with your government sponsor
ECA	~30 days after submitting notarized forms	Not required	View IdenTrust pricing - prices may vary for non-U.S. citizens
PIV	2-6 weeks	Must be a U.S. National†	Consult with your government sponsor

(†)**U.S. National:** An individual who owes permanent allegiance to the United States. This includes U.S. citizens and certain non-citizens, such as individuals born in American Samoa or Guam.

When is a government access card required?

Access to Game Warden-hosted applications is governed by Impact Level (IL) requirements and access control policies. A government-issued access card is **required** for certain environments and strongly recommended for others, depending on the deployment stage and access method.

The table below summarizes where a government access card is required:

Resource	Requires Government Access Card
Customer Application at **IL2 STG**	No
Customer Application at **IL2 PRD**	No
Customer Application at **IL4**	Yes
Customer Application at **IL5**	Yes
Game Warden Application at **IL2**	No
Game Warden Application at **IL5**	Yes
Harbor Registry at **IL2**	No
Harbor Registry at **IL5**	Yes
ArgoCD at **IL2**	No
ArgoCD at **IL4**	Yes
ArgoCD at **IL5**	Yes
Grafana at **IL2**	No
Grafana at **IL4**	Yes
Grafana at **IL5**	Yes

Certificate policies and Public Key Infrastructure (PKI)

What is PKI?

PKI is a security framework used to issue and manage digital certificates for secure authentication, encryption, and digital signatures. In government systems, PKI ensures that users accessing sensitive environments—such as Game Warden at **IL4** and **IL5**—are authenticated with trusted, cryptographically validated credentials.

PKI works by using a pair of cryptographic keys (public and private) linked to a user's identity, managed through a **Certificate Authority (CA)**. These keys are embedded in a **digital certificate** that users present when authenticating to a system.

What are Certificate policies?

A certificate policy is a defined set of rules embedded in a digital certificate that dictates how the certificate may be used and the level of assurance it provides.

Certificate policies are represented by **Object Identifiers (OIDs)** — standardized numeric codes uniquely assigned to each policy. Systems such as P1 and Game Warden validate both the certificate and its associated policy before granting access.

These policies allow relying parties to enforce usage restrictions and ensure compliance with security standards.

Approved certificate policies

Certificate Policy OID	Policy Identifier
2.16.840.1.101.2.1.11.5	id-US-dod-medium
2.16.840.1.101.2.1.11.9	id-US-dod-mediumhardware
2.16.840.1.101.2.1.11.10	id-US-dod-PIV-Auth
2.16.840.1.101.2.1.11.17	id-US-dod-mediumNPE
2.16.840.1.101.2.1.11.18	id-US-dod-medium-2048
2.16.840.1.101.2.1.11.19	id-US-dod-mediumHardware-2048
2.16.840.1.101.2.1.11.20	id-US-dod-PIV-Auth-2048
2.16.840.1.101.2.1.11.31	id-US-dod-peerInterop
2.16.840.1.101.2.1.11.36	id-US-dod-mediumNPE-112
2.16.840.1.101.2.1.11.37	id-US-dod-mediumNPE-128
2.16.840.1.101.2.1.11.38	id-US-dod-mediumNPE-192
2.16.840.1.101.2.1.11.39	id-US-dod-medium-112
2.16.840.1.101.2.1.11.40	id-US-dod-medium-128
2.16.840.1.101.2.1.11.41	id-US-dod-medium-192
2.16.840.1.101.2.1.11.42	id-US-dod-mediumHardware-112
2.16.840.1.101.2.1.11.43	id-US-dod-mediumHardware-128
2.16.840.1.101.2.1.11.44	id-US-dod-mediumHardware-192
2.16.840.1.101.2.1.11.59	id-US-dod-admin
2.16.840.1.101.2.1.11.60	id-US-dod-internalNPE-112
2.16.840.1.101.2.1.11.61	id-US-dod-internalNPE-128
2.16.840.1.101.2.1.11.62	id-US-dod-internalNPE-192
2.16.840.1.101.3.2.1.12.1	id-eca-medium
2.16.840.1.101.3.2.1.12.2	id-eca-medium-hardware
2.16.840.1.101.3.2.1.12.3	id-eca-medium-token
2.16.840.1.101.3.2.1.12.4	id-eca-medium-sha256
2.16.840.1.101.3.2.1.12.5	id-eca-medium-token-sha256
2.16.840.1.101.3.2.1.12.6	id-eca-medium-hardware-pivi
2.16.840.1.101.3.2.1.12.8	id-eca-contentsigning-pivi
2.16.840.1.101.3.2.1.12.9	id-eca-medium-device-sha256
2.16.840.1.101.3.2.1.12.10	id-eca-medium-hardware-sha256
2.16.840.1.101.3.2.1.3.4	id-fpki-certpcy-highAssurance
2.16.840.1.101.3.2.1.3.7	id-fpki-common-hardware
2.16.840.1.101.3.2.1.3.12	id-fpki-certpcy-mediumHardware
2.16.840.1.101.3.2.1.3.13	id-fpki-common-authentication
2.16.840.1.101.3.2.1.3.16	id-fpki-common-High

Certificate Policy OID	Policy Identifier
2.16.840.1.101.3.2.1.3.18	id-fpki-certpcy-pivi-hardware
2.16.840.1.101.3.2.1.3.20	id-fpki-certpcy-pivi-contentSigning
2.16.840.1.101.3.2.1.3.24	id-fpki-SHA1-hardware
2.16.840.1.101.3.2.1.3.36	id-fpki-common-devicesHardware
2.16.840.1.101.3.2.1.3.38	id-fpki-certpcy-mediumDeviceHardware
2.16.840.1.101.3.2.1.3.39	id-fpki-common-pivi-contentSigning

FAQs

Will it affect my app's deployment to IL4+ environments if I don't have a government access card?

Game Warden can deploy your application to **IL4+ staging (STG)** and **production (PRD)** environments even if you do not have a government access card (CAC, PIV, or ECA). However, you will **not** be able to access your environment, logs, or application endpoints without an approved government access card.

Can I hire a service member with a government access card to perform DevSecOps work for my company?

No. Hiring a Reservist or National Guard member to use their government access card for company work is strictly prohibited. This constitutes misuse of government credentials and can result in serious consequences for the service member.

Link Access Card with Platform One Account

To access Impact Level 4 (IL4) and higher environments in Game Warden, you must have a valid Platform One Single Sign-On (P1 SSO) account linked to your government-issued access card (CAC, PIV, or ECA). If you have not yet created a P1 SSO account, follow the instructions in Create a P1 SSO Account before proceeding.

Linking your access card to your P1 SSO account is required **before** you can access Game Warden-hosted staging (STG) and production (PRD) environments. This applies to both Game Warden customers and their end users.

This guide explains how to link your government access card to your P1 SSO account.

Equipment configuration

Before linking your card, ensure your machine is configured for government access card use. This may require downloading and installing the necessary **DoD root certificates**.

- Follow the Remote Access: Initial Setup Guide provided by the DoD Cyber Security Center for certificate installation.
- For Mac users, refer to this setup guide for platform-specific configuration and recommended card readers.

How to link the access card with your P1 SSO account

Before you begin, make sure you have a compatible card reader.

1. Insert your government access card into the reader. Wait until the card reader lights stop flashing before proceeding.
2. Go to the P1 login page. You will be prompted to authenticate with your government access card. The first time you authenticate, you will see a message acknowledging detection of a new access card.
3. Enter your P1 username and password.
4. Enter your Multi-Factor Authentication (MFA) code.
5. Confirm the association request when prompted. This confirmation links your access card to your P1 SSO account. Future logins will automatically authenticate with your government access card.

What to expect after linking your card

Once your government access card is successfully linked:

- You can authenticate into **IL4+ environments** via P1.
- The system will automatically attempt to use your access card when logging in.
- If the card is not detected, you may still enter your **P1 SSO username, password,** and **MFA code**, but a government access card is always required for IL4+ access.

Access requirements for users outside the NIPRNet

The Non-classified Internet Protocol Router Network (NIPRNet) is the DoD's private network for transmitting unclassified but sensitive information.

If you are **accessing IL4+ environments outside of the NIPRNet** or its VPN (e.g., Air Force Desktop Anywhere), you must use Appgate SDP - a DoD-approved authentication solution - for secure access. Appgate must remain open and active for the duration of your session.

To set up Appgate, see Install Appgate SDP.

FAQs

Why don't I have access after setting up my P1 account?

This is often because you haven't yet attempted to log in to Game Warden app.

After creating your P1 SSO account and being added to Game Warden, you must log in at least once. This initial login triggers the import of your user data from P1 into Game Warden's Keycloak (our identity management system).

Once your account is imported, a Game Warden Keycloak administrator can assign the appropriate group memberships associated with your company.

Where can I find more information or get support for P1?

Refer to the Platform One FAQ for common questions and login troubleshooting.

For additional support, contact **aflcmc.hncx.p1cst@us.af.mil**.

Install Appgate SDP

Appgate Software-Defined Perimeter (SDP) is a DoD-approved authentication service that enforces **Zero Trust** principles for secure network access. Managed by the **Platform One (P1) Cloud Native Access Point (CNAP)** team, Appgate SDP helps protect DoD networks by:

- Denying implicit trust for users, devices, and applications
- Enforcing a "verify-then-trust" approach
- Granting least-privilege access based on verified credentials

Appgate SDP is required to securely access Impact Level 4 (IL4) and Impact Level 5 (IL5) environments when connecting from a non-NIPRNet network — such as a commercial or public internet connection.

- **IL4 environments**: Require Appgate SDP
- **IL5 environments**: Require Appgate SDP and may also require running compliance scripts

  Why Appgate SDP matters for Game Warden access

  Game Warden relies on P1 CNAP and Appgate SDP to enforce access controls for IL4 and IL5 environments.

  - Appgate, combined with DoD-approved security measures, ensures compliance with the Cloud Computing Security Requirements Guide (SRG) — particularly sections 5.10.1.x of SRG v1, Release 4.
  - Without Appgate SDP (or a direct NIPRNet connection), you cannot reach Game Warden-hosted IL4+ environments.

How to install Appgate SDP

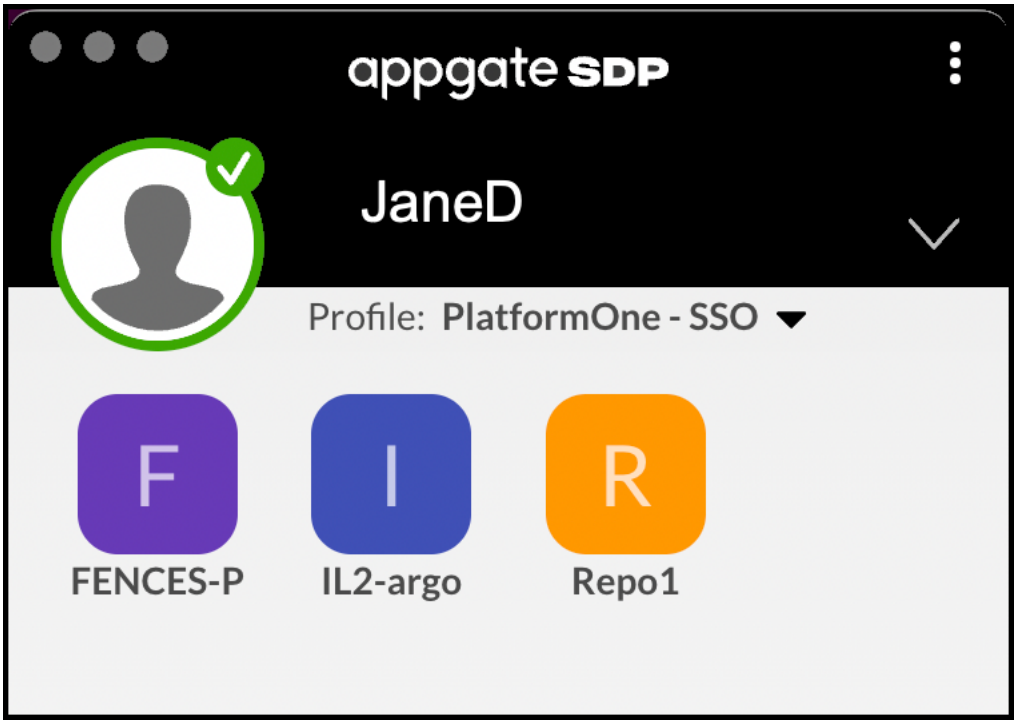
Prerequisites

- Multi-Factor Authentication (MFA) app
- Active P1 SSO account
- Government access card linked to your P1 account
- Compatible operating system (Windows, macOS, or Linux)

1. Download Appgate SDP Client for your operating system.
2. Install Appgate SDP on your machine, following the on-screen instructions.
   - During installation, when prompted in the **Create Profile** window, select **Use Profile Link**.
   - Copy and paste the following profile link into the **Profile Link** field:

     `appgate://connect.cnap.dso.mil/eyJwcm9maWxlTmFtZSI6IlBsYXRmb3JtT25lICOgU1NPIiwic3BhIjp7Im1vZGU:`
3. Click **Submit** to finalize the profile creation.
4. Connect to Appgate SDP:
   a. After profile creation, click **Connect**. A browser window will open to the **P1 Login page**.
   b. Authenticate using your CAC/ECA or P1 credentials and enter your MFA verification code when prompted.
   c. Accept the DoD Information System usage conditions.

Once authenticated, your Appgate SDP session becomes active, and your authorized environments will be displayed. You can now close the P1 window.



Maintain the latest Appgate SDP version

Ensure you are running the latest version of Appgate SDP to guarantee an uninterrupted connection and access to the latest security updates. See Software Availability section for version details and support lifecycles.

Manage your Appgate session

- **To sign out:**
  - Click the **kebab menu**
  - Select **Sign Out** — this will deactivate the session but leave the client open
- **To quit Appgate:**
  - Select **Quit** from the **Options menu** — this will close the client entirely
- **To reconnect:**
  - Open Appgate SDP and click **Sign in with Provider** to launch the P1 login page

Important

You must keep **Appgate SDP open and active** while accessing IL4+ environments when connected via a non-NIPRNet network.

Support

If you experience issues installing or connecting with Appgate SDP, contact **Platform One Support:**

- **Email:** aflcmc.hncx.p1cst@us.af.mil
- **P1 Customer Portal:** Access Portal

Install External Certification Authority Tokens

This guide walks you through obtaining a **DoD-approved External Certification Authority (ECA) hardware token** from IdenTrust and configuring it for use with Platform One (P1) to access Impact Level 4 (IL4) and Impact Level 5 (IL5) resources.

Tested Setup

- MacBook Pro (M2)
- IOGEAR GSR203 card reader

Step 1 — Acquire an ECA hardware token

Warning

- You must obtain and install the ECA token before you can use the card to access IL4 and IL5 environments.
- You must complete the token acquisition process **within 30 days of purchase**, or you will need to start over.

1. Submit ECA request through IdenTrust

1. Visit IdenTrust ECA Certificates for DoD Access.
2. Click **BUY NOW** and follow these steps:
   a. Select **My Federal Program is not Listed**.
   b. Confirm that you live in the U.S.
   c. Select the certificate as **ECA Medium Token Assurance | Hardware Storage**.
   d. Select **1 year** validity period.
   e. Select **HID Smart Card** (with or without a card reader, as needed).
3. Complete the checkout process.
4. Download and print the **Certificate Forms Packet** when provided.

2. Complete Authorization forms

1. Open the **Certificate Forms Packet** and fill out **Page 2** with applicant and organization officer information.
2. Have your **Organization Officer** inked sign and date the form.
3. Complete **Page 4** in front of a licensed notary, presenting two valid IDs based on the following requirements:
   - **U.S. Citizens**: Provide **one ID from List A + one from List B or C**, **OR** provide **one from List B + one from List C**.
   - **Non-U.S. Citizens**: Provide a **valid passport + one ID from List B**.
   - If you declared multiple citizenship in your certificate request, you must present a valid passport for each.

   Tip: Confirm with your notary ahead of time which forms of ID they will accept, as this may vary by jurisdiction.

   Accepted Forms of Identification:

List A — Photo ID establishing identity and citizenship	List B — Photo ID establishing identity	List C — Document establishing U.S. citizenship
- Passport from country of citizenship	- Military ID with photo	- Consular Report of Birth from a U.S. Consulate (Form FS-240)
- Certificate of U.S. Citizenship issued by USCIS (formerly INS)	- Driver's license or government-issued ID card with photo	- Certificate of Birth Abroad issued by the U.S. Department of State (Form DS-1350)
- Certificate of Naturalization issued by a court of competent jurisdiction (pre- or post-1991)	- Permanent or Unexpired Temporary Resident Card issued by USCIS with photo	- Original or certified copy of birth certificate issued by a county, state, or government authority bearing an official seal

4. Mail the completed forms with tracking to IdenTrust HQ. IdenTrust will call your Organization Officer to verify submission. Ensure they are available.

3. Receive and activate your ECA token

After approval (typically 3–5 business days), you will receive an installation email. **Do not proceed** until you have your token and card reader in hand.

Step 2 — Provision your ECA card

1. Install OpenSC

1. Download the latest version from OpenSC Releases.
2. Install using the `.dmg` installer. If you are familiar with Homebrew, you may use it — but note that all library paths will differ from those used in this guide.

2. Reboot and initialize card reader

1. Reboot your computer with the card reader **disconnected**.
2. After logging back in, insert your ECA card. You should see a prompt to pair the card with your account — this confirms the card is working.

(Optional) 3. Test OpenSC with your card

Open Terminal and run:

```
/Library/OpenSC/bin/pkcs11-tool --login --test \`\`\` If successful, the output will resemble the follo
```

4. Retrieve and install IdenTrust certificates

1. Open the email from Registration@identrust.com.
2. Follow the link to www.identrust.com/install. Enter your activation code and password you created during the checkout process in Submit ECA request through IdenTrust.
3. Download and run the retrieval application as instructed.

Step 3 — Install required certificates

1. Install DoD Certificates

Follow the guide on MilitaryCAC.com. to download and install the DoD certificates for your Mac. Ensure the certificates are installed in both the macOS keychain and Firefox (step 5a).

2. Install IdenTrust ECA Root Certificates

Download the IdenTrust ECA Root Certificates from this link. Install them using the same method as the DoD certificates.

Step 4 — Configure applications for smart card use

For applications supporting PKCS11 libraries, use:

`/Library/OpenSC/lib/onepin-opensc-pkcs11.so`

If you are familiar with Homebrew, you may use it — but note that all library paths will differ from those used in this guide.

Cloud Native Access Point Whitelist

The Cloud Native Access Point (CNAP), a service managed by Platform One (P1), provides secure access to Game Warden-hosted environments at Impact Levels 4 and 5 (IL4 and IL5).

The CNAP Whitelist is a security mechanism that restricts access based on an approved list of IP addresses within the Department of Defense (DoD) Non-classified Internet Protocol Router Network (NIPRNet) boundary. Although most NIPRNet-assigned IPs are already included, specific users, data connections, or devices may require whitelisting to resolve access or connectivity issues with Game Warden-hosted IL4/IL5 environments.

When you need CNAP whitelisting

You **must** request CNAP whitelisting if:

- Your application requires external data connections (ingress, egress, or bidirectional).
- The connection originates **outside of the Game Warden Authorization Boundary** but **inside the NIPRNet boundary** (accredited at IL4 or IL5).

When CNAP whitelisting is not required

You **do not need** CNAP whitelisting if:

- Your application is deployed to **IL2 Staging (STG)** or **Production (PRD)** — these environments connect via the internet, but must still be reflected in your Authorization Boundary Diagram.
- Your application is deployed to **IL6 STG/PRD** — these classified environments are segregated from IL4/IL5 for security.
- Your application **does not require external data connections** — no data leaves or enters the Game Warden Authorization Boundary.

  Warning

  - CNAP will not approve IPs from commercial ISPs (e.g., Verizon, AT&T, Comcast) or home users, even on Government Furnished Equipment (GFE).
  - You can verify if an IP is DoD-registered at ARIN Whois.

How to verify if you need whitelisting

Test your connection by attempting to load: https://code.il4.dso.mil

- **If the page loads**: Your IP is already whitelisted.
- **If the page times out**: You must submit a whitelist request to P1.

Why you may not have CNAP access

The USAF's Zero Trust Architecture requires explicit approval for CNAP access. The default is to deny access to unfamiliar IP ranges.

Agencies may request access proactively or after encountering issues. Agencies with frequently changing IP ranges are responsible for keeping their listings current.

Note

Game Warden does not manage or have visibility into CNAP's allowlisted IPs.

Submit a CNAP whitelist request

Information required

Before submitting a CNAP whitelist request, you must gather detailed information about your external data connections. If you are unable to obtain this information, contact your government contract sponsor or Mission Owner for assistance.

The following information is required for **each external connection**:

Information	Example
IP Addresses with Port/Protocol	IP: 192.168.1.1 Port: 443 TCP
IP Address Range (if applicable)	192.168.1.1-192.168.1.254 Port: 443 TCP

How to submit your request

Requests must be submitted by a government user. Contractors must route requests through their government sponsor.

All requests must be submitted directly to P1 using the P1 General Help Form. If you need help submitting the form, contact the P1 Help Center.

Tip

You may need to submit a separate whitelist request for each government installation or environment.

What to include in your whitelist request

When submitting a CNAP whitelist request, you must provide the following:

- Justification explaining why Appgate SDP cannot be used.
- Confirmation that the IP addresses are registered to the DoD.
- The smallest possible scope of IP addresses necessary for your use case.
- Only the egress IP addresses — the IPs visible to external systems. If your traffic is routed through NAT or proxies, only the publicly exposed IPs are required.
- IPs listed in CIDR notation. For example, `192.168.1.0/24`.
- The physical location or military installation associated with the IP addresses.
- A valid point of contact (POC) for the request.
- The CNAP-hosted site(s) your request pertains to.

Access Control for DISA IL4 & IL5

Applications hosted on DISA infrastructure (IL4/IL5) via the 2F.mil DNS configuration use specific access methods. While CNAP and AFWERX deployments rely on Appgate, DISA-hosted environments sit behind a Boundary Cloud Access Point (BCAP) and require a different connection protocol.

Access requirements

Review the following requirements for network access and authentication.

Network access

Users must connect from a **NIPRNet-based IP address** using one of the following methods:

Option 1: Government Furnished Equipment (GFE)

- Use GFE directly from a military installation
- Use GFE remotely via your service-specific VPN

Option 2: Virtual Desktop Infrastructure (VDI)

- Access through your service's VDI solution
- VDI presents your connection as originating from a NIPRNet IP address to DISA

Authentication

- **CAC/ECA certificate required** for all application access
- Authentication is managed through Second Front's Keycloak instance

Access methods summary

Method	Requirements	Use Case
On-site GFE	GFE + NIPRNet connection	Working from military base
Remote GFE	GFE + Service VPN	Working remotely with GFE
Service VDI	Service VDI access + CAC/ECA	Working from non-GFE device

Access Grafana dashboard

To access the Grafana dashboard, follow the same network (NIPRNet) and authentication (CAC/ECA) protocols described previously. See Access Grafana dashboard for specific endpoints.

Troubleshooting

Cannot access application?

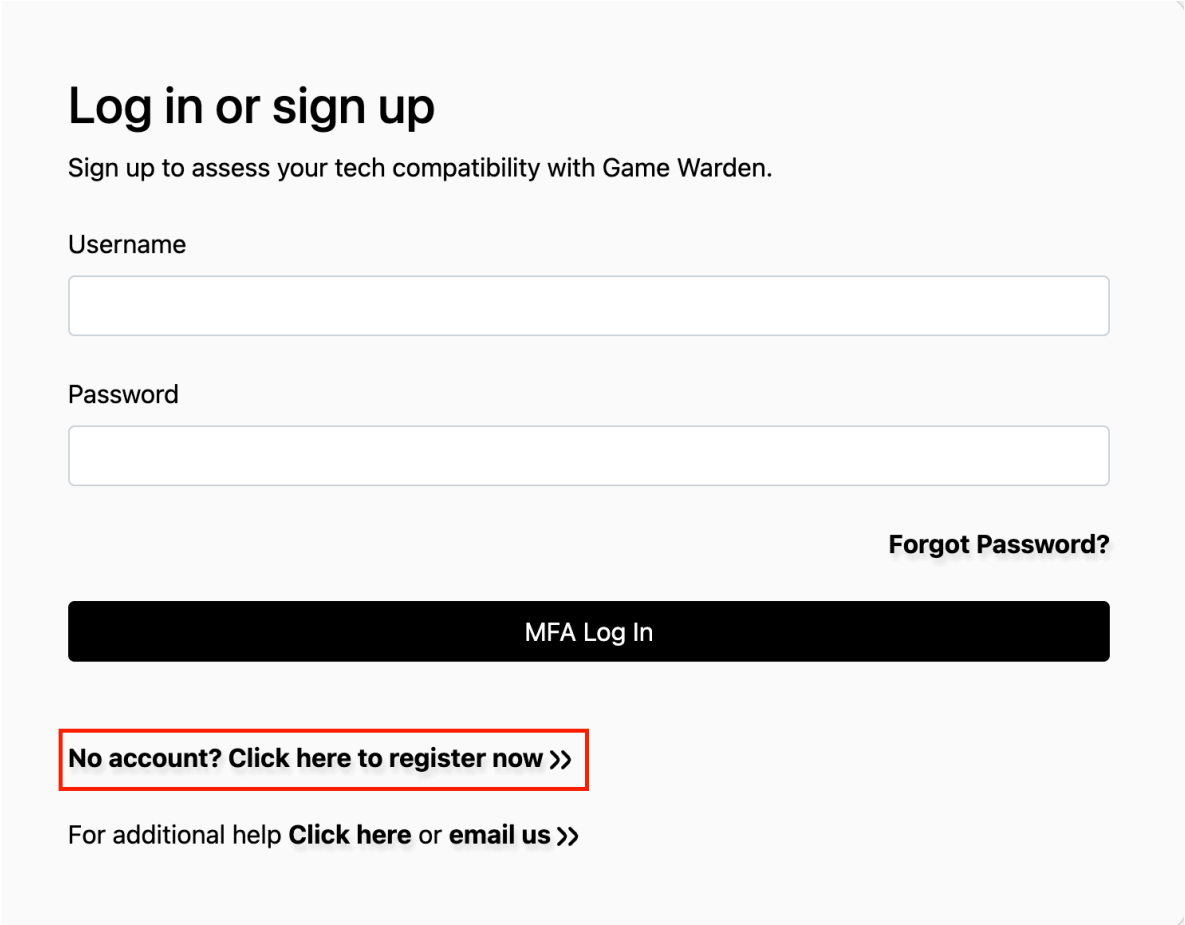
1. Verify you're connecting from a NIPRNet IP address (via GFE or VDI).
2. Confirm your CAC/ECA certificate is properly installed and valid.
3. Ensure you're accessing the correct `2F.mil` URL.
4. Check that your browser is configured to present CAC/ECA certificates.

Game Warden Account Setup Guide for FedRAMP Deployments

Game Warden enables you to deploy your application to a FedRAMP network—a cloud environment that has been authorized under the Federal Risk and Authorization Management Program to meet the U.S. government's security standards for handling federal data (e.g., AWS GovCloud, Microsoft Azure Government, or Google Cloud Platform for Government)—using a secure, managed deployment environment with built-in compliance support. This guide walks you through setting up your account and getting started.

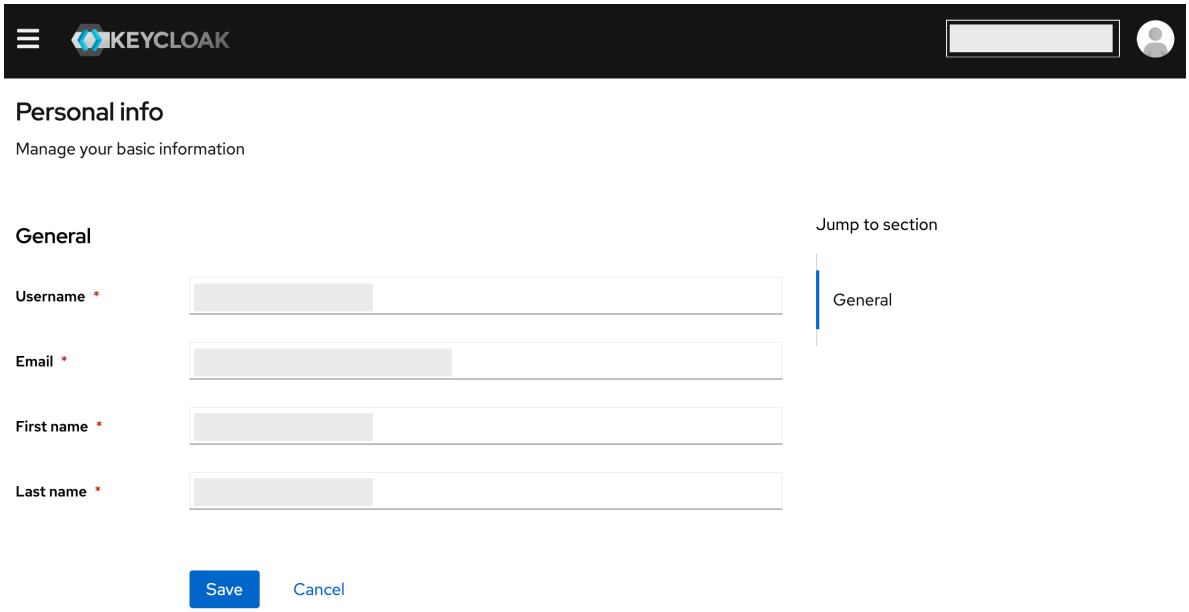
Create your account

1. Visit Game Warden for FedRAMP Deployment, then click **No account? Click here to register now**.



2. On the **Regular User Registration** page, enter the following:
   - Your first name and last name.
   - Select the organization your affiliation belongs to; otherwise, select **Other**.
   - Select your rank or pay grade. Military ranks are grouped by service branch, and civil service grades range from AA to SCS. Select **N/A** if none apply.
   - Enter your organization name.
   - Enter your organization's location (at minimum, the city and state).
   - Enter a username and your work email address.
   - Optionally, provide an access note. This helps administrators assign the appropriate access.
   - Create and confirm your password.

3. Click **Register**.
4. When prompted, complete the multi-factor authentication process:
    1. Install an authentication app (e.g., FreeOTP, Microsoft Authenticator, or Google Authenticator) on your mobile device.
    2. Open the app and scan the QR code displayed on the screen.
    3. Enter the current code from your authenticator app into the **Six digit code** field, and optionally provide the device name to help manage your OTP devices.
    4. Click **Submit**.
5. Review the consent form and click **Accept** to continue.
6. On the **Keycloak Personal Info** page, confirm your information and click **Save**.



Log in to your account

1. Once your account has been set up by Second Front, visit Game Warden for FedRAMP Deployment.
2. Enter the username and password you created during registration.
3. Click **MFA Log In**.
4. Enter the current code from your authenticator app and click **MFA Log In**.
5. Review the consent form and click **Accept** to access the Game Warden app.

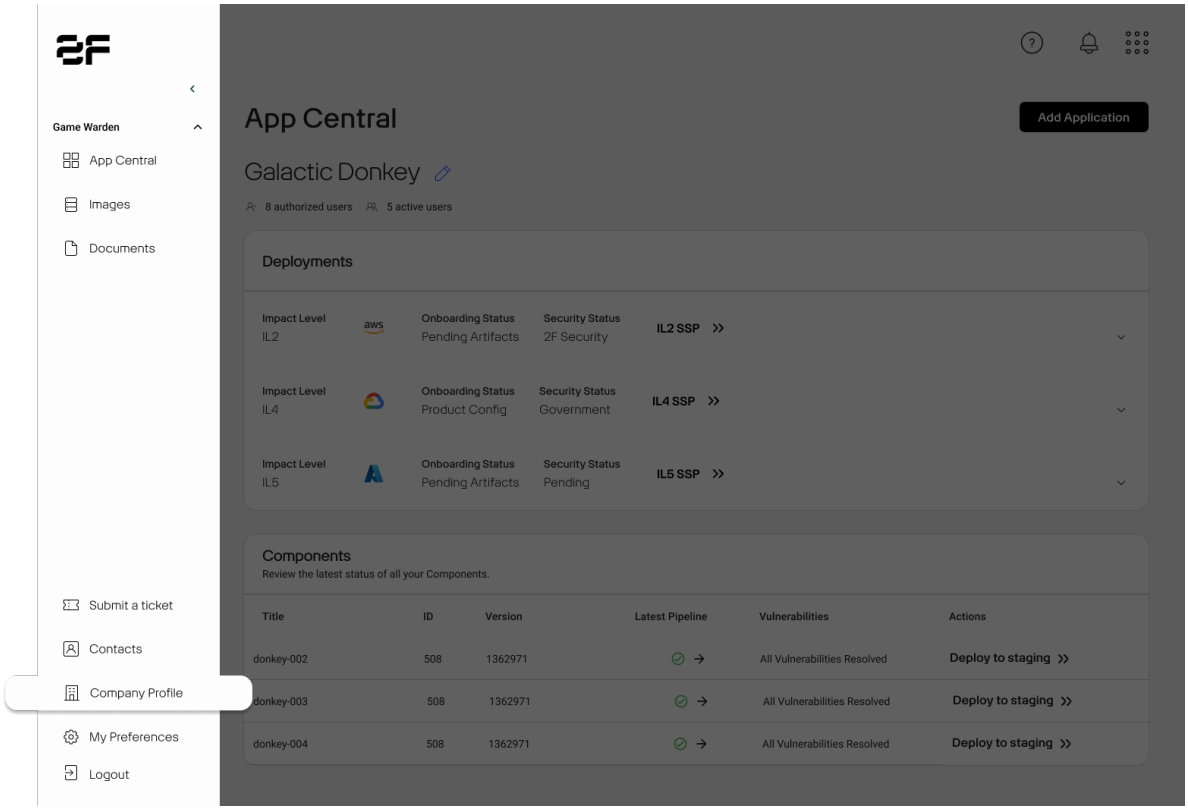
Troubleshooting & best practices

- Forgot your password? Click **Forgot Password?** on the login page to reset your password.
- Need more help? Use the **Click here** or **email us** links at the bottom of the page.
- Keep your MFA method private and secure.
- Always log out when you're away from your computer or working on a shared device.

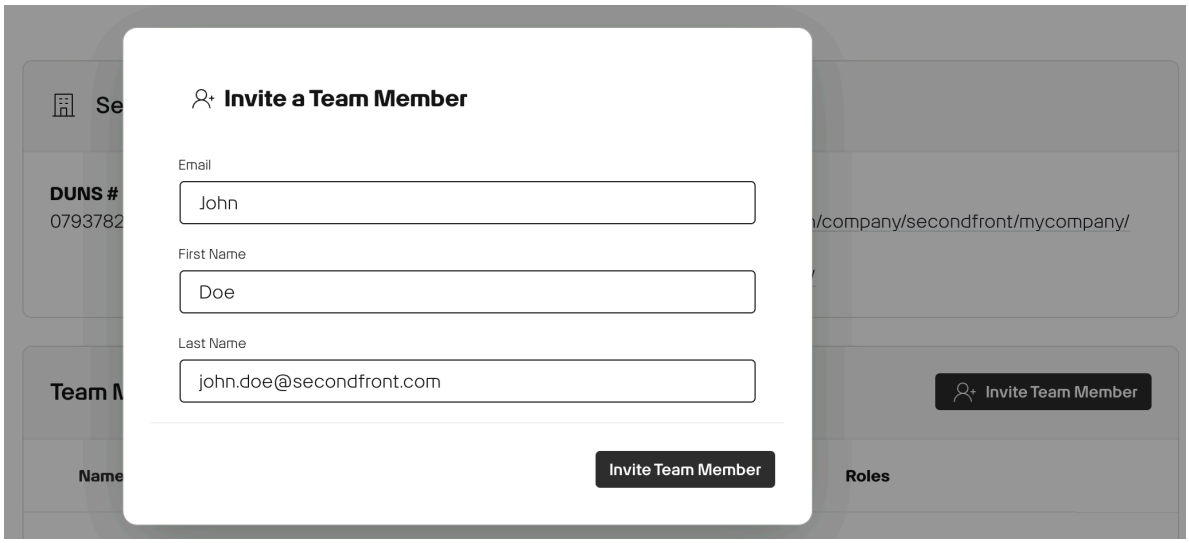
Add your team to Game Warden

Once your user account is set up, you can invite your team to Game Warden from within the app.

1. Go to the **Company Profile** tab in the left sidebar.



2. Click **Invite Team Member**.
3. Fill in each team member's email and name, then click **Invite Team Member**.



Your team member will receive an email invitation to join the Game Warden app.

Warning

If your team member's email domain is different from yours, contact your Technical Implementation Manager (TIM) for assistance.

Add Customer Admin privileges to team members

If you have Customer Admin privileges, you can grant the same access level to other team members. Follow these steps:

1. Locate the team member in your user list, then click the **kebab menu** next to their name.
2. Select **Edit User** from the dropdown.
3. In the **Edit a Team Member** modal, navigate to the Roles section and click **Make user a Customer Admin**.
4. Click **Submit** to save the changes.

The team member will immediately receive Customer Admin permissions and can perform all administrative functions within your organization.

Customer and Customer Admin permissions

Capability	Customer	Customer Admin
Submit Tickets	Yes	Yes
Access ScanLab	Yes	Yes
Download ScanLab artifacts	Yes	Yes
Delete deployments	No	Yes
Set up Harbor credentials	No	Yes
Fill out Body of Evidence forms	Yes	Yes
Resolve vulnerabilities	Yes	Yes
Deploy images	No	Yes

Security Processes for FedRAMP

This guide outlines key security responsibilities and processes customers must follow when deploying their applications into Game Warden's FedRAMP-authorized environment.

Incident response

While some of the Incident Response controls are partially inheritable, the responsibility of Incident Response falls heavily on the customer.

If the customer has an incident, notify Second Front Security immediately at security@secondfront.com. The Second Front Security Team will conduct an investigation in accordance with our Incident Response Plan.

If the incident requires intervention or communication with Second Front Security Team, the team will reach out to the customer via Slack or email to help resolve the issue.

Application security requirements

This section outlines the key security activities customers must complete and maintain to remain compliant with FedRAMP and agency expectations.

Common Vulnerabilities and Exposures (CVEs) remediation

> **Important**
>
> The customer's sponsor reserves the right to approve any CVEs normally not accepted by Second Front. The approval must be obtained by the customer in writing from their sponsor. Anything that is approved by the sponsor will need to be an operational requirement and a deviation request must be submitted to Agency Security and the 3PAO.

All CVEs must comply with FedRAMP's established Acceptance Baseline Criteria. All CVEs will need to be remediated or justified and accepted by Second Front Security Team. CVEs that are not able to be remediated must be added to the customer Plan of Action and Milestones (POA&M) documentation with real and actionable incremental milestones to reduce risk over time until the CVE is resolved.

Any Critical or High CVEs that cannot be remediated (removed from the application before deployment) will need to be submitted as a deviation request or a POA&M must be created with actionable milestones.

Second Front Security Team will review each image's remediation and justifications before allowing them to be deployed to staging and/or production after FedRAMP Authorization is obtained.

All Moderate and Low vulnerabilities exceeding FedRAMP remediation timelines must be resolved or justified prior to Production (PRD) deployment. These items will be transitioned to a POA&M once Continuous Monitoring (ConMon) begins.

For Moderate and Low vulnerabilities still within their remediation window, customers must exercise due diligence to resolve or POA&M them before their respective deadlines.

> **Remediation Timeline**
>
> For required timeframes to remediate or justify findings, see **Tables A–C** in the Acceptance Baseline Criteria.

External Data Connections (EDCs)

All customers will be required to list out all EDCs according to Agency Security's requirements for the Body of Evidence (BoE).

Authorization Boundary Diagram (ABD)

All customers will be expected to complete an ABD in accordance with our current FedRAMP processes found here.

Any changes to the ABD will require a new diagram to be added to the customer's account and the changes should be communicated to Agency Security, 3PAO, and the customer's sponsor.

Image hardening

All images must be hardened using Game Warden's hardening script. No unhardened images may be promoted to staging and/or production with the exception of images without a shell which will not be hardened.

Second Front's engineers will be responsible for applying hardening scripts to each image. Second Front Security Team will be responsible for ensuring hardening scripts are applied to images during the Security Review process.

Static Application Security Testing (SAST)

All customers will be required to perform SAST scans on all codebases included for all images deployed onto Game Warden. SAST should be uploaded into the Game Warden application for the initial deployment.

Dynamic Application Security Testing (DAST)

All customers are responsible for running DAST on their applications in accordance with FedRAMP's ConMon requirements. Second Front offers DAST scanning in house for customers that can be run monthly.

ConMon

Once your application is deployed into Game Warden's FedRAMP-authorized environment, it enters an ongoing, monthly security reporting and review process required by FedRAMP to ensure platform and application security over time.

The sections below outline Second Front's responsibilities and yours during ConMon.

Game Warden's FedRAMP Authorization ConMon

The Second Front Security Team will submit the Game Warden relevant information on a monthly basis, in accordance with FedRAMP standards. The Game Warden relevant information will encompass the Platform as a Service and will not include the applications hosted on Game Warden. Second Front **will not** provide the platform monthly continuous monitoring information to customers.

Second Front submits Asset Inventory Lists, Database Scans, Operating Systems Scans, and Web Application Scans (**only** for images that Second Front uses), and Second Front's POA&M. Second Front is not responsible for any of the customer's recurring submission requirements except for Database Scans.

Image scanning

The Game Warden application is configured to scan all images for new CVEs/Vulnerabilities in production once a month. Agency Security will assist the customer in pulling down those scans monthly to allow the customer to submit the images for ConMon.

Understanding Commercial Deployment

Second Front's Commercial Deployment environment offers a secure, government-grade platform for software vendors to deploy production-ready applications—without the delays of full DoD authorization. Deploying into this environment allows your organization to serve regulated industries, engage defense stakeholders, and prepare for future government deployment.

This guide outlines the Commercial Deployment environment, highlighting its key benefits, strategic use cases, core platform features, security expectations, and the process for transitioning to Impact Level (IL) 2-5, or FedRAMP environments.

High-value use cases

The Commercial Deployment environment is ideal for organizations that need to:

- **Serve defense and regulated markets** with a secure environment that emulates DoD-grade compliance practices.
- **Validate product-market fit** in a secure, government-grade environment—**without needing a DoD contract**. Ideal for early-stage companies to build credibility, gather feedback, and demonstrate value before pursuing formal government agreements.
- **Demonstrate mission fit** to government stakeholders, accelerating sales and partnership cycles.
- **Prepare for government authorization** by refining application security posture in a pre-ATO setting.
- **Deploy production applications** to commercial end users, including defense contractors and primes, without crossing into FedRAMP or IL4/5 boundaries.

Key benefits

Benefits	Details
Government-equivalent security	Applications are hosted in a secure AWS US East environment that mimics IL2/IL4 practices, including: CVE scanning; SAST/DAST attestation from the security team (*); Body of Evidence (BoE); all without requiring a formal Certificate to Field (CtF)/Software Approval.
Faster time to market	No FedRAMP or IL4/IL5 authorization required to onboard, deploy, or launch. Start engaging customers and iterating on your product sooner.
Flexible onboarding and growth path	Start in Commercial and migrate seamlessly to IL2, IL4, IL5, or FedRAMP environments when you're ready—no need to start over.
Self-service access with managed support	Self-register and begin setup with help from Second Front implementation engineers and support teams.

(*) As part of our security screening, Second Front (2F) Systems will perform the DAST scan, while your organization is responsible for conducting and providing the SAST artifacts.

Platform features

Feature	Description
Isolated Commercial Infrastructure	Hosted in AWS us-east-1; distinct from GovCloud environments.
Secure Identity Management	Uses Keycloak with support for SAML and OAuth 2.0.
Automated DevSecOps	CI/CD, container scanning, and infrastructure automation via GitLab, ArgoCD, and container registry.
Integrated Monitoring	Access built-in observability tools like Grafana for logs and metrics.
Dedicated Ticketing System	Manage deployments, onboarding, and support via a dedicated support portal.
BoE	Required for each app; DoD-specific sections omitted for Commercial deployments.

Security expectations

Although the Commercial environment resides outside formal government boundaries (ATO/FedRAMP), it upholds rigorous security protocols:

- Completion of a BoE within the Game Warden platform.
- Identification and remediation of all CVE vulnerabilities. If remediation is not possible, a written justification or mitigation plan must be provided, along with a proposed remediation timeline.
- Review and confirmation of SAST and DAST artifacts.
- Approval of External Data Connections (EDCs).
- Optional migration to higher assurance environments with minimal disruption.

Second Front Security Operations Center participates in onboarding and monitoring setup for each deployment.

Customer journey

Launch and stay in Commercial

1. Register for access at https://login.gamewarden.io.
2. Complete onboarding with help from our Implementation Engineers.
3. Submit your BoE.
4. Pass a security review.
5. Deploy your application to production.

Migrate to DoD or FedRAMP environment

1. Start in Commercial and mature your application.
2. Seamlessly migrate to IL2, IL4, IL5, or FedRAMP using a shuttle service developed by Second Front.
3. Undergo a full security review and receive AO approval in your new environment.

FAQs

How do I get started?

Create an account at https://login.gamewarden.io. You will be prompted to register your organization and users.

Can I deploy to cloud regions other than AWS US East (us-east-1)?

Currently, no. However, you may submit region requests to the product team for future consideration.

Do I need to complete a BoE?

Yes. While some DoD-specific sections are optional, the majority of the BoE must be completed before deploying to production.

Will my app be reviewed for security before going live?

Yes. All production deployments require a security review, even in the Commercial environment.

Is this a FedRAMP-authorized environment?

No. The Commercial environment is separate from FedRAMP and ATO boundaries, though it mimics IL2/IL4 practices for development and security posture.

What's next

As adoption grows, Second Front is continuing to enhance the Commercial Deployment offering with:

- A one-click migration path to higher ILs
- Enhanced environment labeling and navigation
- Dynamic login methods based on region
- Improved onboarding automation and observability

Got a question? Reach out to Second Front System today!

Commercial Deployment Security Policies

This guide outlines the security processes and standards required for applications deployed in Second Front's Commercial Deployment environment. Although the Commercial Deployment environment is outside formal Authorization to Operate (ATO) and FedRAMP boundaries, our goal is to uphold the same rigorous security practices used in Department of Defense (DoD) environments, including:

- Body of Evidence (BoE)
- CVE Management
- External Data Connections (EDCs)
- Authorization Boundary Diagram (ABD)
- Image Hardening
- DAST & SAST Requirements

BoE

All applications must have a completed BoE within the Game Warden platform. The following sections **may be omitted**:

- Role Identification
- Information Security Plan
- CAC Personnel
- Secure Software Development Framework (SSDF) Attestation
- Certificate to Field (CtF) Recommendation Memo

CVE management

All applications must comply with Second Front's Acceptance Baseline Criteria for vulnerabilities (CVEs):

- All **CVE vulnerabilities** must be remediated. If remediation is not possible, a written justification or mitigation plan must be provided, along with a proposed remediation timeline.
- Justifications must be reviewed and approved by Second Front's security team before deployment to development or production.
- If a Critical or High CVE **cannot be remediated**, the security team must sign a risk acceptance memo before deployment.

External Data Connections (EDCs)

All external data connections must be documented in the BoE and included in the Authorization Boundary Diagram; and each EDC must be reviewed and approved by Second Front's security team before use in the application.

Authorization Boundary Diagram (ABD)

An Authorization Boundary Diagram visually maps your system's components, data flows, and security boundaries. It ensures your system integrates securely with Game Warden and meets DoD deployment standards.

All customers must submit an Authorization Boundary Diagram in accordance with DoD deployment standards. This diagram should include:

- All containers
- Communication flows
- Ports and protocols
- External data connections

For submission requirements, see Authorization Boundary Diagram.

Image hardening

All container images are hardened by Second Front engineers using Game Warden's official image hardening scripts. This process is verified by the security team as part of the security review. Only hardened images are eligible for promotion to staging or production environments.

DAST and SAST requirements

DAST Scan

As part of our security screening, Second Front performs Dynamic Application Security Testing (DAST) scan on all container images.

- DAST scans must meet DoD-level standards.
- Third-party reviewers are **not required** to review DAST results for commercial deployments.

SAST Scan

Customers must perform Static Application Security Testing (SAST) on all container images.

- These scans will be held to the same standards as DoD deployments.
- Results must be submitted as part of the security review.

Second Front is committed to maintaining strong security practices across all deployment environments. Adhering to the policies in this document ensures your applications are production-ready, defensible, and aligned with federal cybersecurity expectations—even without a formal government authorization boundary.

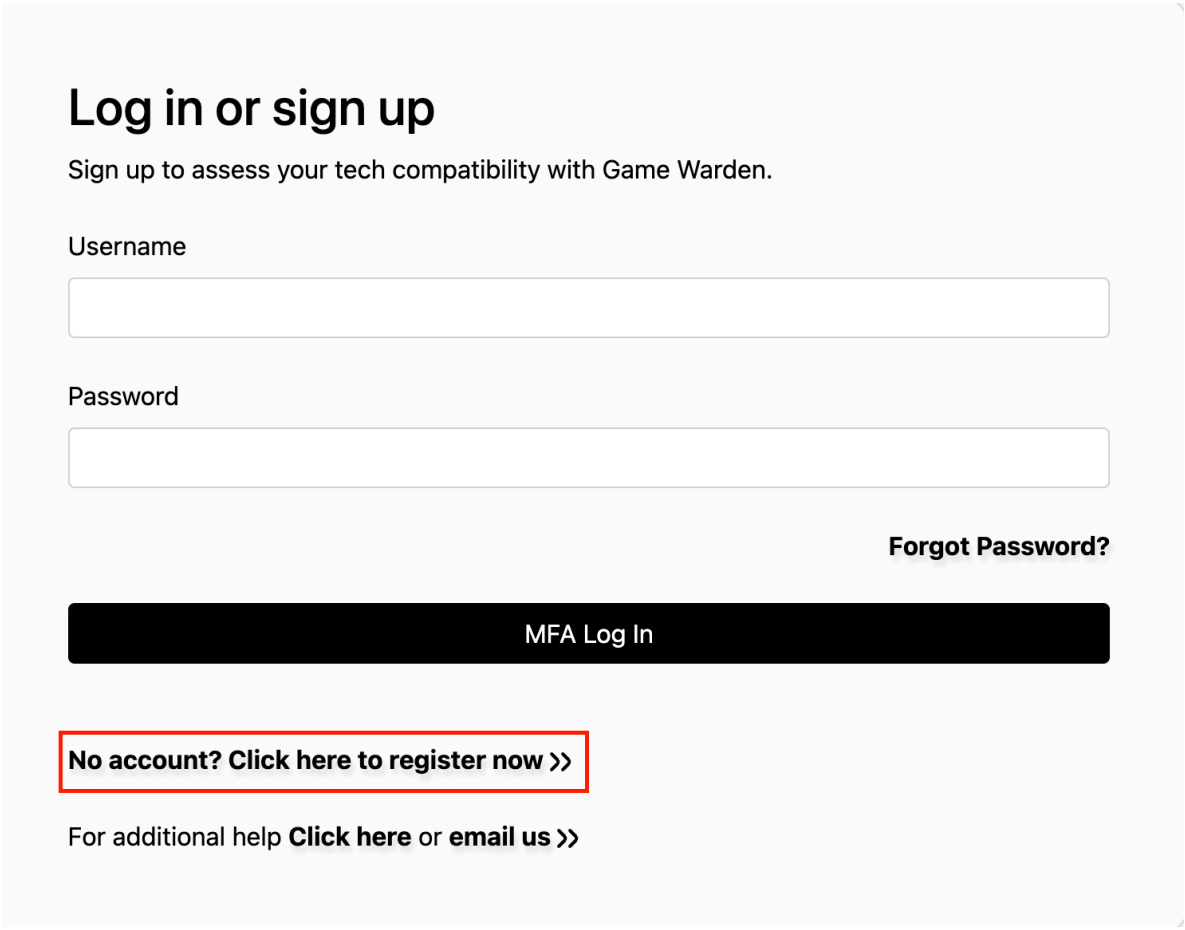
Got a question? Submit a support ticket today!

Game Warden Account Setup Guide for Commercial Deployments

Game Warden enables you to deploy your application to a commercial network using a secure, managed deployment environment with built-in compliance support. This guide walks you through setting up your account and getting started.

Create your account

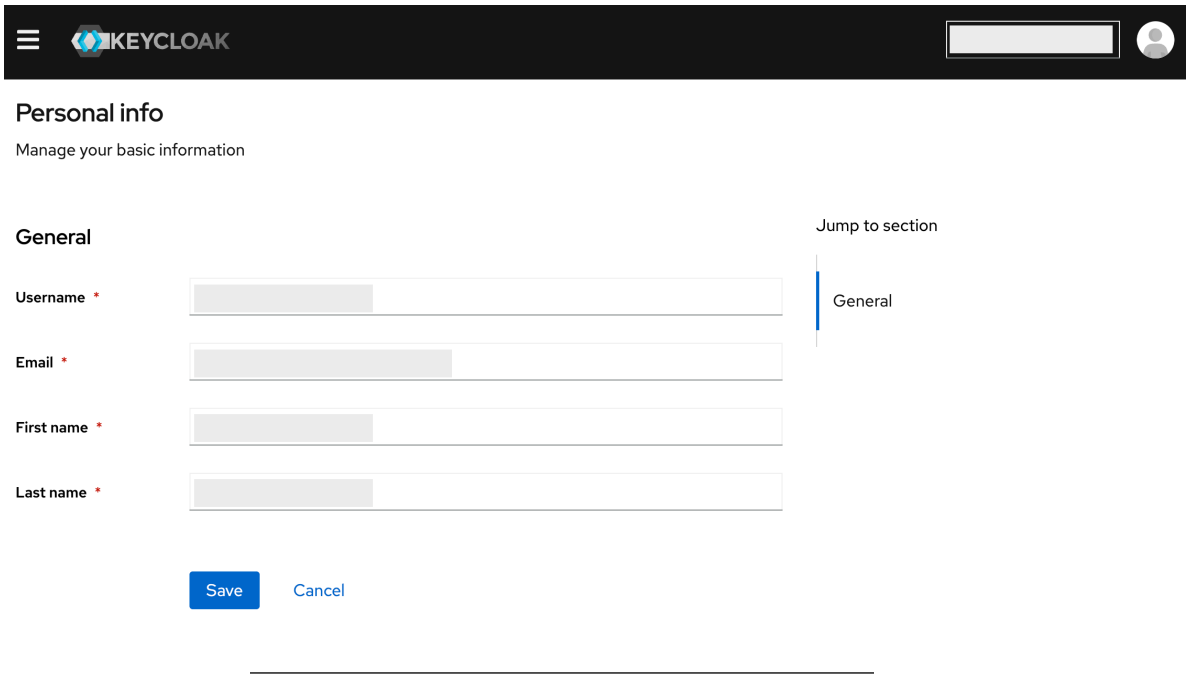
1. Visit Game Warden for Commercial Deployment, then click **No account? Click here to register now**.



2. On the **Regular User Registration** page, enter the following:
   - Your first name and last name.
   - Select the organization your affiliation belongs to; otherwise, select **Other**.
   - Select your rank or pay grade. Military ranks are grouped by service branch, and civil service grades range from AA to SCS. Select **N/A** if none apply.
   - Enter your organization name.
   - Enter a username and your work email address.
   - Optionally, provide an access note. This helps administrators assign the appropriate access.
   - Create and confirm your password.
3. Click **Register**.
4. When prompted, complete the multi-factor authentication process:

1. Install an authentication app (e.g., FreeOTP, Microsoft Authenticator, or Google Authenticator) on your mobile device.
2. Open the app and scan the QR code displayed on the screen.
3. Enter the current code from your authenticator app into the **Six digit code** field, and optionally provide the device name to help manage your OTP devices.
4. Click **Submit**.

5. Review the consent form and click **Accept** to continue.
6. On the **Keycloak Personal Info** page, confirm your information and click **Save**.



Log in to your account

1. Once your account has been set up by Second Front, visit Game Warden for Commercial Deployment.
2. Enter the username and password you created during registration.
3. Click **MFA Log In**.
4. Enter the current code from your authenticator app and click **MFA Log In**.
5. Review the consent form and click **Accept** to access the Game Warden app.

Troubleshooting & best practices

- Forgot your password? Click **Forgot Password?** on the login page to reset your password.
- Need more help? Use the **Click here** or **email us** links at the bottom of the page.
- Keep your MFA method private and secure.
- Always log out when you're away from your computer or working on a shared device.

General Requirements

Before contacting the Game Warden team, ensure your deployment meets the following general requirements. These apply based on whether you are deploying to **DoD**, **FedRAMP**, or **Commercial** environments.

Requirements comparison

DoD requirements

Category	Requirements
Account Requirements	Must support **P1 SSO authentication** only.
Authentication Requirements	**Appgate SDP** is required for IL4+ access unless connected via **NIPRNet** or **NIPRNet VPN** (e.g., Air Force Desktop Anywhere). Appgate must remain active during the session. A **P1 SSO account** is required to use Appgate SDP.
Hardware & Compatibility Requirements	If using External Certification Authority (ECA) card, you must use **Windows** or **macOS** to download your digital certificate. **Linux/Ubuntu** is not supported and may cause compatibility issues. Recommended browsers: **Microsoft Edge** or **Google Chrome** on Windows, **Mozilla Firefox** on macOS. Always use the latest browser versions. Read more about IdenTrust certificate compatibility.
Government Contract Requirement	Must maintain an active **DoD contract** for deployment eligibility. Only Common Access Card (CAC), ECA, or Personal Identity Verification (PIV) card holders may access Game Warden for sensitive classifications.
Domain & Environment Requirements	Applications must be deployed under the **afwerx.dso.mil** domain.

FedRAMP requirements

Category	Requirements
Account Requirements	Must support **OAuth 2.0** or **SAML** Identity Providers.
Government Contract Requirement	Government sponsor required.
Domain & Environment Requirements	Applications must be deployed in the **Game Warden FedRAMP environment**.

Commercial requirements

Category	Requirements
Account Requirements	Must support **OAuth 2.0** or **SAML** Identity Providers.
Government Contract Requirement	No contract required.

Category	Requirements
Domain & Environment Requirements	Applications must be deployed in the **Game Warden Commercial environment**.

Need help?

The Game Warden team is available to help ensure your product aligns with these requirements and to provide guidance throughout your deployment process. Contact the Growth team for assistance or additional information.

Technical Requirements

To deploy an application on Game Warden, your solution must satisfy the following architecture and security specifications **before** you engage the Game Warden onboarding team.

Architecture

- **Containerization (OCI-compliant)** - The application **must run in containers** that conform to the Open Container Initiative (OCI) specification.
- **Kubernetes compatibility** - The application **must be deployable on Kubernetes**, using standard Kubernetes primitives (Deployments, StatefulSets, Services, ConfigMaps, etc.) and **must not rely on host-level access or non-Kubernetes runtimes**.
- **Database seeding**:
  - Provide automated seed services or SQL/DDL scripts for the Game Warden team to execute.
  - At **IL4** you **will not** have direct write access to production databases.
- **CPU architecture** - Workloads **must target AMD64/x86_64**. ARM architectures are not currently supported.

Security

Requirement	Details
Meeting ATO vulnerability baseline	Game Warden performs continuous security scans. All findings must be remediated in accordance with the Acceptance Baseline Criteria. Components must be patched regularly to maintain ATO compliance.
Continuous CVE remediation	New CVEs discovered post-deployment **must be resolved promptly** by the application team.
DoD-approved authentication	Applications must integrate with a DoD-approved identity provider—**Game Warden SSO** or **Platform One SSO**.
Credentialed access (IL4+)	Personnel accessing IL4+ environments require a valid government access card credential obtained through standard DoD vetting.
Data classification limits	Permitted data classifications: **CUI, PII, IL2, IL4, IL5, ITAR**. Contact Game Warden before processing **IL6, Special Access Programs (SAP), or Sensitive Compartmented Information (SCI)** data.

AWS GPU support by environment

AWS GovCloud East (IL2 - IL5)

For a list of Amazon EC2 instance types available in AWS GovCloud (US-East), see the AWS documentation.

AWS Secret Partition (IL6)

EC2 Instance	Instance Name	GPU Supported
g3	g3.4xlarge	1 NVIDIA Tesla M60 GPU, with 2048 parallel processing cores and 8 GiB of video memory
g3	g3.8xlarge	2 NVIDIA Tesla M60 GPUs, each with 2048 parallel processing cores and 8 GiB of video memory
g3	g3.16xlarge	4 NVIDIA Tesla M60 GPUs, each with 2048 parallel processing cores and 8 GiB of video memory
g4dn	g4dn.xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.2xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.4xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.8xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.12xlarge	4 NVIDIA T4 Tensor Core GPUs
g4dn	g4dn.16xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.metal	8 NVIDIA T4 Tensor Core GPUs
g5	g5.xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.2xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.4xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.8xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.16xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.12xlarge	4 NVIDIA A10G Tensor Core GPUs
g5	g5.24xlarge	4 NVIDIA A10G Tensor Core GPUs
g5	g5.48xlarge	8 NVIDIA A10G Tensor Core GPUs
p3	p3.2xlarge	1 NVIDIA Tesla V100 GPU, pairing 5,120 CUDA Cores and 640 Tensor Cores
p3	p3.8xlarge	4 NVIDIA Tesla V100 GPUs, each pairing 5,120 CUDA Cores and 640 Tensor Cores
p3	p3.16xlarge	8 NVIDIA Tesla V100 GPUs, each pairing 5,120 CUDA Cores and 640 Tensor Cores
p3dn	p3dn.24xlarge	8 NVIDIA Tesla V100 GPUs
p5	p5.48xlarge	8 NVIDIA H100 Tensor Core GPUs

EC2 Instance	Instance Name	GPU Supported
p5en	p5en.48xlarge	8 NVIDIA H100 Tensor Core GPUs

AWS Top Secret Partition

EC2 Instance	Instance Name	GPU Supported
g3	g3.4xlarge	1 NVIDIA Tesla M60 GPU, with 2048 parallel processing cores and 8 GiB of video memory
g3	g3.8xlarge	2 NVIDIA Tesla M60 GPUs, each with 2048 parallel processing cores and 8 GiB of video memory
g3	g3.16xlarge	4 NVIDIA Tesla M60 GPUs, each with 2048 parallel processing cores and 8 GiB of video memory
g4dn	g4dn.xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.2xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.4xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.8xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.12xlarge	4 NVIDIA T4 Tensor Core GPUs
g4dn	g4dn.16xlarge	1 NVIDIA T4 Tensor Core GPU
g4dn	g4dn.metal	8 NVIDIA T4 Tensor Core GPUs
g5	g5.xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.2xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.4xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.8xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.16xlarge	1 NVIDIA A10G Tensor Core GPU
g5	g5.12xlarge	4 NVIDIA A10G Tensor Core GPUs
g5	g5.24xlarge	4 NVIDIA A10G Tensor Core GPUs
g5	g5.48xlarge	8 NVIDIA A10G Tensor Core GPUs
g6	g6.xlarge	1 NVIDIA L4 Tensor Core GPU
g6	g6.2xlarge	1 NVIDIA L4 Tensor Core GPU
g6	g6.4xlarge	1 NVIDIA L4 Tensor Core GPU
g6	g6.8xlarge	1 NVIDIA L4 Tensor Core GPU
g6	g6.16xlarge	1 NVIDIA L4 Tensor Core GPU
g6	g6.12xlarge	4 NVIDIA L4 Tensor Core GPUs
g6	g6.24xlarge	4 NVIDIA L4 Tensor Core GPUs
g6	g6.48xlarge	8 NVIDIA L4 Tensor Core GPUs
p3	p3.2xlarge	1 NVIDIA Tesla V100 GPU, pairing 5,120 CUDA Cores and 640 Tensor Cores

EC2 Instance	Instance Name	GPU Supported
p3	p3.8xlarge	4 NVIDIA Tesla V100 GPUs, each pairing 5,120 CUDA Cores and 640 Tensor Cores
p3	p3.16xlarge	8 NVIDIA Tesla V100 GPUs, each pairing 5,120 CUDA Cores and 640 Tensor Cores
p3dn	p3dn.24xlarge	8 NVIDIA Tesla V100 GPUs
p4d	p4d.24xlarge	8 NVIDIA A100 Tensor Core GPUs
p5	p5.48xlarge	8 NVIDIA H100 Tensor Core GPUs

Next steps

Confirm your architecture and security posture meet the requirements above, then contact the Game Warden team. We'll help ensure alignment and support your application launch. Reach out to the Growth team at growth@secondfront.com for details.

Supported Design Patterns

Supported design patterns define technical and security requirements for integrating customer applications with the Game Warden platform. These patterns guide customers on architectural expectations, security compliance, and operational considerations to streamline onboarding and deployment.

Architecture

Must-Have

Specification	Details
Containerized applications	Applications must be containerized and comply with Open Container Initiative (OCI) standards.
Database seeding services or scripts	Provide database seeding services or scripts for Game Warden to execute, especially at IL4 where production database write access is restricted.
Complete applications	Applications must be fully functional before starting the onboarding process.

Nice-to-Have

Specification	Details
Built-in database migration	Your organization should handle your own data and schema migrations.
Microservice architecture	Single service per container is preferred to promote reliability.
Dummy data for testing	Provide dummy data or data scrubbing methods for Staging (STG) environment testing.
End-to-end automated testing	Implement automated testing to validate application functionality.
Structured logs	Use structured logging to simplify observability and troubleshooting.
Configuration documentation	Provide documentation for configuration variables and APIs.
Simple Helm charts	Prefer Helm charts designed for Helminator, supported by detailed Authorization Boundary Diagrams.
Logs to stdout	Log to stdout for seamless collection and analysis with observability tools.
Detailed diagrams	Submit comprehensive Authorization Boundary Diagrams showing ports, protocols, and system architecture.

Acceptable Use

Specification	Details
Single points of failure	Must function reliably; Game Warden will assist with resilience improvements.
Monolithic applications	Accepted but increase operational complexity.

Specification	Details
S3-hosted frontends	Supported with conversion to NGX containers as needed.
Own Helm charts or Kustomize manifests	Supported but Helm (via Helminator) is preferred. Knowledge of Kubernetes is required.
Mobile applications	Supported.

Deal Breaker

Specification	Details
Moving high Impact Level (IL) data to lower ILs	Prohibited to prevent data spillage.
External connections outside NIPRNet	Not supported; feature under future consideration.
IL4/IL5 access without Appgate SDP	For AFWERX deployment, must use Appgate SDP or approved VPN when not on NIPRNet.

Security

Must-Have

Specification	Details
Data classification adherence	Only approved classifications (CUI, PII, IL2, IL4, IL5, ITAR (International Traffic in Arms Regulations)) are allowed. Contact Game Warden for IL6, SAP (Special Access Program), or SCI(Sensitive Compartmented Information) data.
Keycloak integration	All applications must use Game Warden's Keycloak for access control.
ATO compliance	Applications must meet Authority to Operate (ATO) standards and address security vulnerabilities promptly.
Integration with DoD-approved authentication	Only Game Warden and Platform One authentication services are supported.
Government-issued access credentials	Common Access Card (CAC), External Certification Authority (ECA), or Personal Identity Verification (PIV) credentials are required for IL4+ access.
Regular updates	Applications must be maintained and updated to mitigate known vulnerabilities to ensure alignment with the Acceptance Baseline Criteria.
Authorization Boundary Diagram	Provide a detailed diagram showing components, data flow, ports, protocols, and external connections.
Active DoD contract	Required for IL4+ deployment.

Nice-to-Have

Specification	Details
Case-by-case external data transit	Requires Game Warden review and government approval.
Pre-scan with Anchore open source tools	Recommended before submitting images.
Data classification tagging	Use data tags (e.g., CUI) to aid in proper handling.
Known base containers	We recommend using Game Warden-provided or commonly supported base images, such as Universal Base Images (UBIs), which streamline security review and troubleshooting.

Acceptable Use

Specification	Details
Data pull operations	Allowed when initiated within the Game Warden boundary.
NIPRNet external connections	Permitted with NIPRNet IP whitelisting.
Iron Bank-based containers	Acceptable if they meet Iron Bank baseline standards.

Deal Breaker

Specification	Details
Hosting classified data	Prohibited on Game Warden; deployment to classified networks requires ODIN.
Data movement from higher to lower ILs	Strictly prohibited.
No Game Warden SSO for IL4/IL5	Mandatory to use Game Warden SSO.
Non-accredited external connections	External IL4/IL5 connections must be to accredited systems.
Unvetted commercial data streaming	Requires pre-approval discussion with Game Warden.

General

Must-Have

Specification	Details
DoD contract for IL4/IL5	An active DoD contract is required to deploy to the IL4 and IL5 Staging (STG) and Production (PRD) environments.
U.S. Citizenship	Required for engineers involved in deployment.
`.mil` domain association	For AFWERX, applications must deploy under `afwerx.dso.mil`.
`2F.mil` domain association	For DISA, applications must deploy under `2F.mil`.

Nice-to-Have

Specification	Details
Knowledge of Kubernetes and microservices	Facilitates collaboration and troubleshooting.
Allow Game Warden to host/mirror code	Preferred; planned feature on the roadmap.
Proactive application testing	Recommended prior to deployment.

Acceptable Use

Specification	Details
IL2 deployment without DoD contract	Supported.
Use of AWS GovCloud East	Supported with future plans to expand to GovCloud West.

Deal Breaker

Specification	Details
No DoD contract for IL4+	Cannot deploy beyond IL2 without an active DoD contract.

Feature requests

Have ideas for new features or improvements in Game Warden? We want to hear from you! Reach out to our Product team at product@secondfront.com.

Recommended Containers

Your application must be containerized to deploy on the Game Warden platform. Although you are free to use any container tools, Game Warden recommends the following options to simplify onboarding and increase the likelihood of security approval:

- Chainguard
- Iron Bank
- Other container options

Chainguard containers

Chainguard provides minimal, security-focused container images designed to reduce vulnerabilities and simplify compliance.

These images are typically smaller and less complex, often resulting in **zero Common Vulnerabilities and Exposures (CVEs)**.

Why choose Chainguard

- Minimal footprint with a strong security posture.
- Often CVE-free, reducing your remediation burden.
- Free when using images tagged `latest`.

Usage guidelines

- You may build upon Chainguard base images and **push your derived images to the Game Warden Harbor registry**.
- The `latest` tag always pulls the newest image version, which may introduce unexpected changes. To avoid unexpected changes, always:
  - Pull the `latest` tag.
  - Retag the image with your specific application version.
  - Push the tagged version to Game Warden.
- Chainguard images with tags other than `latest` may require a paid subscription. Contact your Technical Implementation Manager for partnership details.

Iron Bank containers

Iron Bank hosts hardened container images vetted by Platform One (P1) to meet Department of Defense (DoD) standards. These images can be pushed to Harbor registry, and eventually deployed to Game Warden.

When selecting images from Iron Bank, pay close attention to two key ratings — they help assess the security posture of each image:

- **Acceptance Baseline Criteria (ABC)** - Indicates whether the image meets Iron Bank's security compliance standards.
- **Overall Risk Assessment (ORA)** - Scored between 0% and 100%, where higher scores reflect better security posture.

We recommend choosing images that are **ABC Compliant** with an **ORA of 80% or higher** to improve your chances of approval on the Game Warden platform.

Responsibilities when using Iron Bank

- Understand the ABC and ORA scores of your selected images.
- Notify the Game Warden team when you plan to use Iron Bank images, especially those currently marked as compliant with strong ORA scores.
- Monitor image compliance regularly and stay informed on End of Life (EoL) status, as image compliance can change over time.
- If a container's compliance status changes:
  - Coordinate with the image owner through P1 to resolve issues.
  - Migrate to a compliant image if necessary.
- Always use the latest, compliant version of your selected image.

If an Iron Bank container does not initially meet security requirements, you may be asked to select a different image. However, containers below the recommended thresholds may still be approved after review by the Game Warden security team:

- **Low and Medium CVEs** exceeding thresholds may be justified.
- **High and Critical CVEs** are assessed individually for security impact.

To minimize approval delays, notify the Game Warden team early when selecting images for deployment.

Images with EoL status

- **EoL images are unsupported** and likely won't pass security review.
- Typically, you'll have **6–12 months of overlap** between new and deprecated images.
- Check for updates **monthly** to maintain compliance.

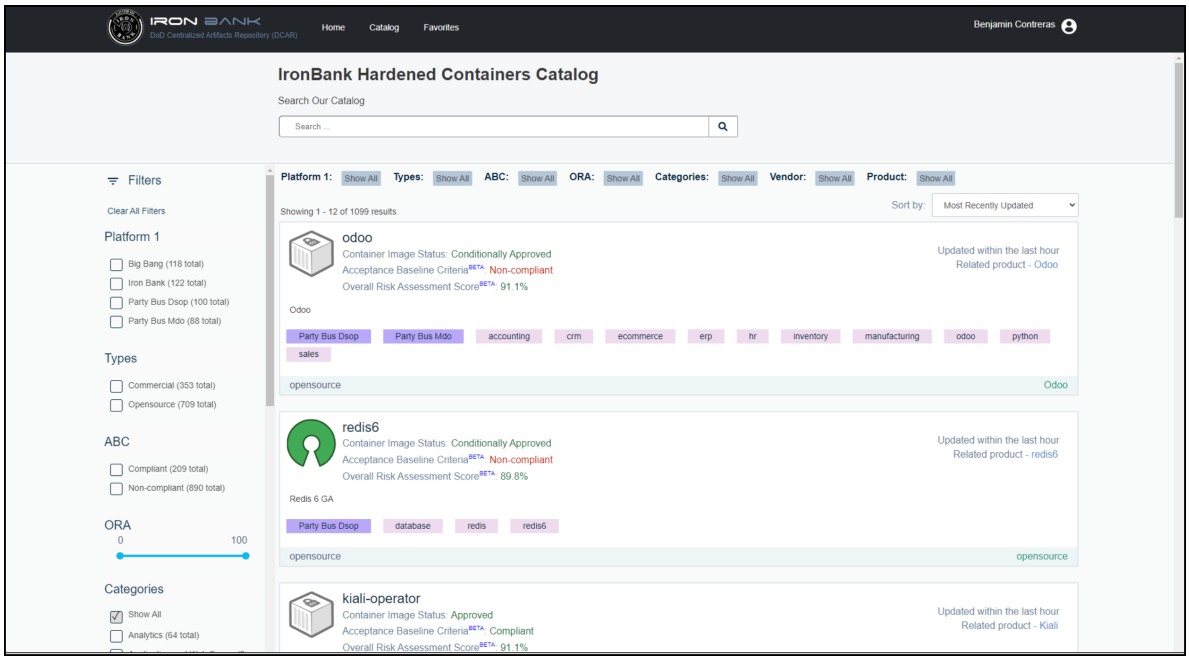
Approval process for Iron Bank images in Game Warden

- The **2F security team** may approve unmodified Iron Bank images that meet compliance or acceptable risk thresholds.
- Once approved, any inherited vulnerabilities from the base image appear in:
  - Game Warden Findings
  - Deployment Passport

This eliminates the need for additional justification from your team for those base vulnerabilities.

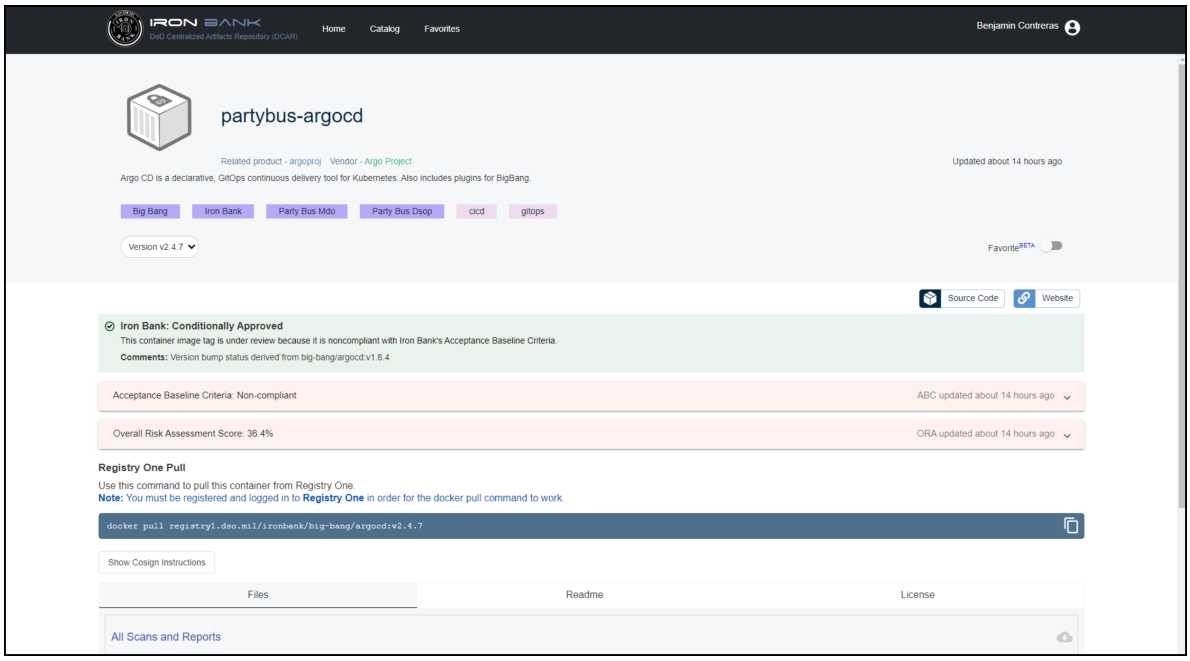
How to pull from Iron Bank

1. Log into the Iron Bank Catalog using your **P1 SSO**.
2. Search for your preferred image.



3. From the search results, select the image to view the Registry One Docker pull command.
4. Locate the pull command for your image, for example:

   `docker pull registry1.dso.mil/ironbank/big-bang/argocd:v2.4.7`



Other container options

You may choose to build your own containers or select images outside of Chainguard or Iron Bank. However, these images will undergo more rigorous security scrutiny.

If you do, we recommend scanning your images with open-source tools such as Trivy or Grype to identify and address Common Vulnerabilities and Exposures (CVEs) before submitting to Game Warden.

Recommended practice: Distroless containers

Distroless containers are minimal container images that include only the application and its runtime dependencies — they exclude the operating system package manager and unnecessary libraries. This approach improves security by reducing the container's attack surface and limits the potential for vulnerabilities.

Benefits of distroless containers

- Smaller image size, leading to faster downloads and deployments.
- Fewer included libraries and tools, reducing the attack surface.
- Lower CVE exposure due to minimal dependencies.
- Easier to maintain and deploy consistently.

Mobile Access Guidance

This guide outlines how Game Warden supports mobile access and sets expectations for customers considering mobile solutions.

Recommended: Progressive Web Applications (PWAs)

Game Warden recommends customers build responsive Progressive Web Applications for mobile use.

Why PWAs?

- Work across any standards-compliant browser
- Do not require app store distribution
- Compatible with Appgate SDP and Keycloak
- Simplify deployment and maintenance

PWAs are the best way to offer mobile access while meeting Game Warden's security and compliance requirements.

Not Supported: Personal or Unmanaged Devices

Game Warden does not allow mobile applications on personal or unmanaged devices at any Impact Level. These devices are not authorized to handle Controlled Unclassified Information (CUI) or sensitive data. This restriction is enforced by federal policy and is not expected to change.

Possible with Conditions

If you have a government partner and a signed contract, you may explore the feasibility of deploying a mobile app on government-issued devices.

To begin, you must provide:

- A sponsoring government partner with a signed contract
- Design specifications for the mobile application
- Specifications for the edge device
- Details on the intended use case

  Customer responsibilities

  - You and your government partners are responsible for mobile app security, design, and compliance.
  - Second Front will assist only after feasibility is established with the government sponsor.

Securing External Data Connections

Securing data exchange is essential for applications deploying in both Commercial and DoD Impact Level (IL) environments. Every External Data Connection (EDC) must be carefully planned, documented, and reviewed to ensure compliance with DoD security standards.

This guide outlines the requirements and security practices for safely establishing EDCs in the Game Warden platform.

Why securing EDCs matters

Improperly configured external connections can lead to **data spills**, **unauthorized access**, and **CtF/Software Approval delays**. For example:

- An IL4 application cannot send Controlled Unclassified Information (CUI) to an IL2 system or public internet endpoint.
- All data moving over external connections must be encrypted using FIPS-compliant standards.
- Game Warden must verify that connections don't bypass IL segmentation rules.

Key security requirements

- **Data Flow Documentation** – Diagrams and written descriptions of all inbound, outbound, and bidirectional data flows.
- **Encryption In Transit** – All data must be protected with TLS 1.2+ (e.g., mTLS).
- **Encryption At Rest** – External systems must use FIPS 140-2 validated encryption modules or equivalent.
- **Impact Level Segmentation** – Data must remain within its assigned IL unless explicitly approved.

Security guidelines for Impact Levels

IL2 – Public or Non-Critical Mission Data

- Enforce basic access controls and authentication
- Use TLS 1.2+ encryption for all data in transit
- Maintain activity logging and monitoring to detect misuse
- Avoid exposing non-public DoD data despite lower sensitivity

IL4 – Controlled Unclassified Information (CUI)

- Strictly enforce IL segmentation—CUI must remain within IL4 or higher
- **Do not process or store National Security System (NSS) CUI**
- Apply encryption at rest using FIPS 140-2 validated modules
- Maintain role-based access control (RBAC) and audit logging
- Mission Owner Attestation may be required before approving EDCs

IL5 – Higher Sensitivity CUI / Mission-Critical

- Enforce least privilege for all users and services
- Segment workloads to isolate sensitive functions
- Apply continuous monitoring (e.g., IDS/IPS, SIEM)
- Encrypt all external data at rest and in transit
- Document and justify all boundary-crossing data flows

- Cross-IL data movement requires formal security review

IL6 – Classified Systems

- Continuous monitoring and active threat detection
- Apply the strongest possible encryption for all data
- Enforce restricted physical and logical access
- Follow NSA/DoD-defined cross-domain control protocols
- External connections must use approved cross-domain solutions and are rarely permitted

---

EDC and Sponsoring Agency approval

Second Front customers can leverage an authorizing agency sponsor to extend their application's capabilities beyond the Game Warden boundary. This allows for the integration of external services and data sources while maintaining a robust security posture.

Key guidance

- **Authorizing Official (AO) Authority**: Your application's risk-owning AO has the authority to approve External Data Connections (EDCs) residing outside the approved Game Warden boundary. This includes connections to other ATO'd environments or API-extensible managed services, provided they meet your application's required IL.
- **Documentation & Compliance**: To facilitate the approval process, all EDCs-including external managed services-must provide the information outlined in the EDC Checklist. This documentation is vital for the AO's risk assessment and ensures the connection adheres to platform security standards.
- **Customer Licensing & Access**: For all EDCs to external APIs, customers are responsible for procuring necessary licenses and managing API credentials as required for their specific mission use case.

Commercial connection rules

Commercial data connections entering a DoD-IL environment must follow a strict **ingress-only** policy—this includes IL2 environments. Outbound or bidirectional communication is not allowed unless explicitly approved through the EDC request process.

Following these security requirements—and taking a proactive approach to risk management—helps teams maintain secure data flows, protect critical systems, and ensure mission continuity. A **documented incident response plan** is also essential for addressing potential security events swiftly and effectively.

EDCs checklist

If your application includes any EDCs, you must submit an EDC request package to Second Front for review and approval. Use the checklist below to validate your setup before assembling the required artifacts for your submission.

Impact Level Segmentation

Checklist items:

- Are data flows confined to their respective ILs?
- Do systems only communicate with others at the same IL?

Data Protection

Checklist items:

- Is data encrypted in transit (mTLS 1.2+)?
- Is data encrypted at rest (FIPS 140-2 validated modules)?

Connection Documentation

Checklist items:

- Have you included a complete network diagram?
- Have you described the data types, purpose, and justification for each connection?

IL-Specific Requirements

Checklist items:

- **IL2:** Are external connections limited to essential, public-facing services?
- **IL4:** Has a Mission Owner Attestation been signed (if applicable)? No NSS CUI processed?
- **IL5:** Are encryption, logging, and access controls in place?
- **IL6:** Are strongest encryption and continuous monitoring in place?

CTI Containment

Checklist items:

- Have you ensured that no Controlled Technical Information (CTI) is transmitted outside the DoD environment?

Data Traversal Between ILs

Checklist items:

- If data must traverse between different Impact Levels (ILs), has a formal security review and approval been completed?

Commercial-to-DoD Connections

Checklist items:

- Does your commercial connection follow the **ingress-only** rule (unless explicitly approved)?

Security Posture

Checklist items:

- Are vulnerability scans conducted regularly?
- Are IDS/IPS systems in place?
- Is the principle of least privilege enforced?
- Is continuous monitoring active?
- Is there an incident response plan?

Required artifacts

Include the following with your EDC request:

- List of all IPs, ports, and protocols
- Direction of data flow for each connection

- Description of transmitted data types
- Diagram showing data flow and boundary crossings, similar to the already existing Authorization Boundary Diagram provided as part of the Body of Evidence (BoE)
- Justification and purpose for each connection

Need help?

For questions or to submit your request package, contact your Technical Implementation Manager or Mission Success Manager.

External Messaging Services

External messaging services allow your Game Warden-deployed application to send outbound, no-reply messages to users through supported cloud-native tools. These services can deliver emails, pub/sub events, or webhooks using providers like AWS, GCP, or Azure.

The information in this guide helps you build secure, compliant, and effective user messaging into your application, without violating DoD or Game Warden requirements.

Supported messaging types

Game Warden supports the following message types:

Message Type	Supported Providers	Description
Email	AWS SES, SendGrid, Mailgun	Send outbound, no-reply emails
Pub/Sub	GCP Pub/Sub, Azure SB	Broadcast internal or external events
Webhooks	Custom HTTPS endpoints	Connect with third-party systems

---

Template configuration

You can define custom message templates that originate from within your application and serve specific use cases. Below are some common use cases:

Use Case	Example
User Notification	"You have new documents available for download."
Credential Expiry Alert	"Your password will expire in 3 days."
System Maintenance Notice	"System maintenance is scheduled for 22:00 UTC."
Event Trigger Notification	"Your request has been processed successfully."

Template design guidelines

To ensure compliance and prevent security issues, follow these rules:

- Use **generic, non-personalized** language (e.g., "You have a new message. Log in to view.")
- **Do not include Controlled Unclassified Information (CUI)** or personal identifiable information
- Avoid **CUI markings or classification labels**
- Carefully validate any **dynamic template variables** to avoid leaking sensitive content

  Warning

  If CUI data is sent through these services, it may cause data spillage into unauthorized networks or storage systems.

Customer responsibilities

Game Warden cannot determine which data is CUI across all customer applications. Your team must:

- Work with your **Mission Owner** to identify what qualifies as CUI
- Ensure message templates do not expose sensitive data
- Prevent unauthorized distribution of CUI through external services

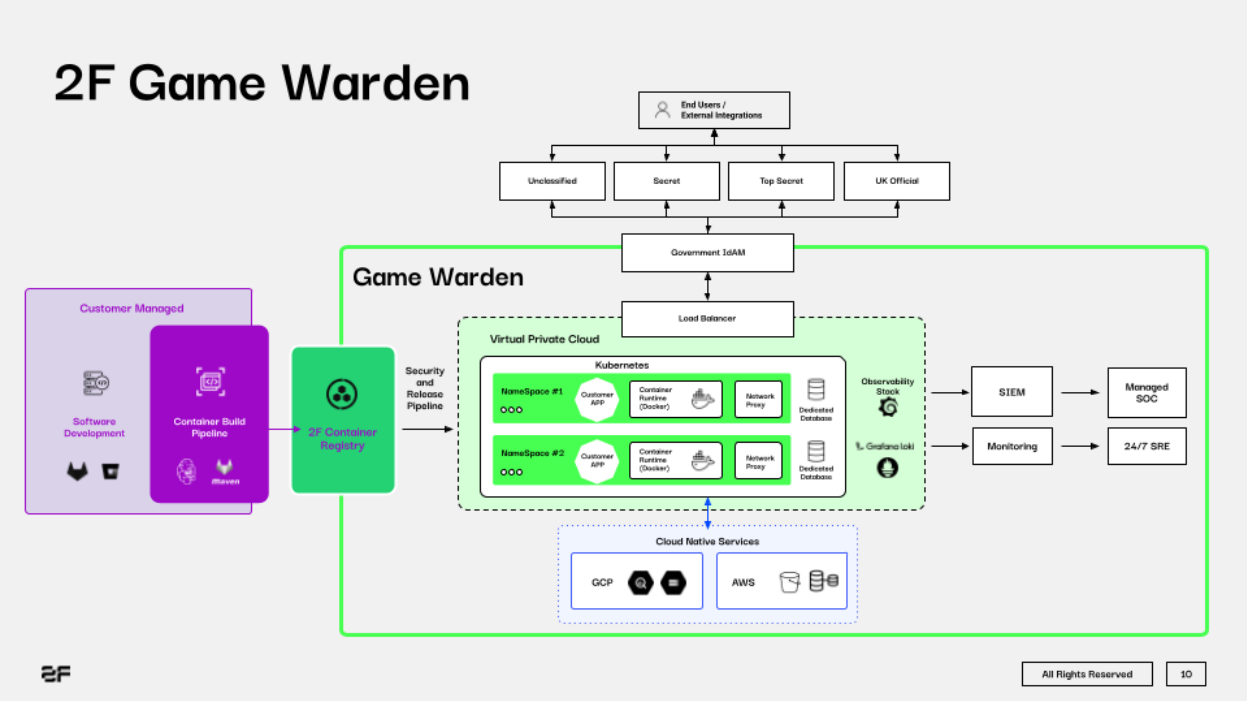
> **Info**
>
> If you suspect a **CUI data spillage**, report the incident immediately by following the procedure outlined in the Customer Incident Reporting Procedure.

Need help?

If you're unsure about your messaging integration or compliance boundaries, contact your implementation engineer.

Game Warden Platform Architecture

Game Warden Platform provides a secure, scalable, and compliant environment for deploying SaaS applications to government networks. With built-in Authority to Operate (ATO) inheritance, observability, and CI/CD integration, the platform minimizes operational overhead while maintaining strict security and compliance standards.



The following sections provide an overview of Game Warden's architectural components.

Customer-managed components

Your organization owns and operates the following components:

- **Software development**: Application code is developed using standard development tools and frameworks (e.g., GitHub, GitLab, Bitbucket).
- **Container build pipeline**: Use a CI/CD pipeline managed by your organization to generate a Docker container image of your application's code.

  Meet 2F Workshop

  Second Front offers a secure, pre-integrated development environment called Workshop, designed to help you build and deploy apps faster within Game Warden. Workshop includes hardened containers, built-in security scans, and ready-to-use CI/CD pipelines—reducing setup time and helping you meet compliance requirements from day one.

Second Front container registry

This is a secure, Game Warden-managed registry where validated application containers are stored. These containers are processed through a security and release pipeline before being deployed into the Game Warden Platform.

Game Warden Platform

The core platform is responsible for securely hosting and managing the lifecycle of SaaS applications in a government-approved environment.

- **Virtual Private Cloud (VPC)**: The VPC serves as the isolated hosting environment where customer workloads run, ensuring network segmentation and policy enforcement.
- **Kubernetes Orchestration**: Applications are deployed in dedicated namespaces, each containing:
  - **Customer App**: The containerized workload provided by the application team.
  - **Container Proxy**: Handles secure ingress/egress and traffic inspection.
  - **Network Proxy**: Applies additional network policies or segmentation controls.
  - **Dedicated Database (if needed)**: Managed databases for persistent storage.
- **Cloud Native Services**: Game Warden is cloud-agnostic and can run on multiple providers, including:
  - GCP (Google Cloud Platform)
  - AWS (Amazon Web Services)

Security and compliance layer

Game Warden enforces strict access and compliance controls, including:

- **Government IdAM & Load Balancer**: All access to applications passes through a government-validated identity and access management system and a centralized load balancer.
- **End User Access Tiers**: Supports varying sensitivity levels such as Unclassified, Secret, Top Secret, UK Official, etc., ensuring appropriate data segregation.

Observability and operations

Game Warden includes a built-in observability stack for operational insight and compliance monitoring:

- **Grafana Loki**: Used for application and system log aggregation.
- **Monitoring Tools**: Tracks performance, availability, and resource usage.
- **SIEM Integration**: Platform-level security event and information management for infrastructure monitoring and incident response.
- **Managed SOC and 24/7 SRE**: Real-time security response and platform reliability support.

Platform Processes & Security Controls

Game Warden is a Department of Defense (DoD)-authorized DevSecOps platform designed to securely and efficiently deploy SaaS applications into government networks. The platform integrates security tooling, continuous integration/continuous deployment (CI/CD) pipelines, and governance processes to streamline software delivery while ensuring compliance with federal requirements.

Integrated tooling and CI/CD pipelines

Game Warden provides a managed CI/CD ecosystem powered by GitLab, which supports source code management, issue tracking, and pipeline automation. Pipelines enable automated code testing, container image scanning, hardening, and deployment to secured environments.

Image scanning and security tools

- **Anchore Enterprise and Checkmarx supported Zed Attack Proxy** - Used for container image security scanning, vulnerability identification, and compliance enforcement.
- **ClamAV** - Performs malware detection on container images.

Image scan results, including Common Vulnerabilities and Exposures (CVEs), are visible in Findings within the Game Warden web app, providing transparency into your security posture.

Image management with Harbor Registry

Harbor is the secure container registry used by Game Warden. All container images pushed into Game Warden go through Harbor, where status tags are applied to indicate their scanning and hardening states.

Tags help track images as they progress through security checks and readiness for deployment.

Understanding images and containers

A container is a lightweight, portable package that contains an application along with all its required dependencies, ensuring consistent execution across different environments.

A container image is a static specification of this package, composed of multiple layers. The foundational layer is the base image, which provides core operating system components. Additional layers include required software packages, such as application binaries, runtime environments (e.g., nginx), or other dependencies.

Base image

Base images must **remain unmodified after selection**. Any customization must occur in layers built on top of the base image.

Game Warden supports applications structured with multiple containers, where each container is responsible for a distinct service or function within the overall application architecture.

Game Warden pipelines and deployment processes

Automated pipelines

Pipelines are sets of automated tasks that move your applications through scanning, hardening, and deployment stages. Game Warden engineers provision infrastructure, configure pipelines, and develop bootstrap scripts to automate CI/CD workflows.

You can monitor pipeline progress and security reviews via Findings.

Deployment workflow

1. **Development:**
   - No Certificate to Field (CtF)/Software Approval required.
   - Engineers configure the environment and validate basic functionality.

2. **Staging:**
   - Requires a signed CtF/Software Approval.
   - Extensive application and security testing.
   - Requires Government Access Cards for environment access at IL4 and above.

3. **Production:**
   - Live deployment environment with customer access.
   - Monitored by Site Reliability Engineers (SREs) for stability and support.
   - Requires Government Access Cards for environment access at IL4 and above.

How pipelines support Deployment Passport creation

The Game Warden pipelines do more than move your application through CI/CD stages, they also generate critical artifacts that make up your Deployment Passport.

As your container images progress through scanning, hardening, and security validation, the results and evidence gathered become part of the documentation package submitted for security approval.

These automated security checks, along with your application architecture and compliance data, are compiled by the Game Warden Security team to build a complete Deployment Passport, a required component for deployment into IL4+ environments.

Security hardening and binary reduction

Before deployment, all images undergo hardening to reduce security risks. The process includes:

- Restricting user permissions and removing unnecessary users/groups.
- Locking down file system permissions.
- Cleaning temporary files and broken links.

Hardening script example

```
#!/bin/sh

set -x #trace on
set -e #break on error
user_id=$1
hardening_platform=$2

echo "Using user " $user_id
echo "Hardening for " $hardening_platform

# Create the 'appuser' user and group used to launch processes
# The uid and gid will be set to 950 to avoid conflicts with offical users on the docker host.
# groupadd -r appuser -g 950 && \
#     useradd -u 950 -m -d /home/appuser -r -g appuser -s /sbin/nologin -c "restricted docker account" a
```

```
# mkdir /app
# chown -R appuser:appuser /app

#nginx specific commands
case $hardening_platform in
  nginx)
    # NGINX files for nonroot
    chown -R $user_id:$user_id /etc/nginx
    mkdir -p /var/cache/nginx
    chown -R $user_id:$user_id /var/cache/nginx
    mkdir -p  /var/log/nginx
    chown -R $user_id:$user_id /var/log/nginx

    # NGINX directory for data
    mkdir -p /var/www
    chown -R $user_id:$user_id /var/www
    mkdir -p /var/run
    touch /var/run/nginx.pid
    chown -R $user_id:$user_id /var/run/nginx.pid
    ;;
esac

sed -i -r "s/$user_id:!:/$user_id:x:/" /etc/shadow

# Remove unnecessary user accounts.
sed -i -r "/($user_id|root)/!d" /etc/group
sed -i -r "/($user_id|root)/!d" /etc/passwd
sed -i -r "/($user_id|root)/!d" /etc/shadow

# Remove interactive login shell for everybody but $user_id.
sed -i -r "/$user_id:/! s#^(.*):[^:]*$#\1:/sbin/nologin#" /etc/passwd

# Removing files generated by sed commands above (group-, passwd- and shadow-)
find $sysdirs -xdev -type f -regex '.*-$' -exec rm -f {} +

# Initial list of system directories
all_sysdirs="
  /bin
  /etc
  /lib
  /lib64
  /sbin
  /usr
"

# Filter out only existing directories
sysdirs=""
for dir in $all_sysdirs; do
  if [ -d "$dir" ]; then
    sysdirs="$sysdirs $dir"
  else
    echo "Directory $dir not found. Skipping."
  fi
done
```

```
# Ensure system dirs are owned by root and not writable by anybody else.
find $sysdirs -xdev -type d \
  -exec chown root:root {} \; \
  -exec chmod 0755 {} \;

# Remove existing crontabs, if any.
rm -fr /var/spool/cron

# Remove kernel tunables since we do not need them.
rm -fr /etc/sysctl*
rm -fr /etc/modprobe.d
rm -fr /etc/modules

# Remove fstab since we do not need them.
rm -f /etc/fstab

# Remove all but a handful of admin commands.
[ -d /usr/sbin ] && find /usr/sbin ! -type d \
  -a ! -name nologin \
  -delete

# Remove all but a handful of executable commands.

##executabe commands to remove
case $hardening_platform in
  dotnet)
    find /usr/bin ! -type d \
    -a ! -name cd \
    -a ! -name ls \
    -a ! -name sh \
    -a ! -name bash \
    -a ! -name dash \
    -a ! -name dir \
    -a ! -name env \
    -a ! -name rm \
    -a ! -name find \
    -a ! -name coreutils \
    -a ! -name test \
    -a ! -name dotnet \
    -delete
    ;;

  go)
    find /usr/bin ! -type d \
    -a ! -name cd \
    -a ! -name ls \
    -a ! -name sh \
    -a ! -name bash \
    -a ! -name dir \
    -a ! -name env \
    -a ! -name rm \
    -a ! -name find \
    -a ! -name coreutils \
```

```
  -a ! -name test \
  -delete
  ;;

jdk)
  find /usr/bin ! -type d \
  -a ! -name cd \
  -a ! -name ls \
  -a ! -name sh \
  -a ! -name bash \
  -a ! -name dash \
  -a ! -name dir \
  -a ! -name env \
  -a ! -name rm \
  -a ! -name find \
  -a ! -name mkdir \
  -a ! -name coreutils \
  -a ! -name test \
  -a ! -name java \
  -delete
  ;;

nginx)
  find /usr/bin ! -type d \
  -a ! -name cd \
  -a ! -name ls \
  -a ! -name sh \
  -a ! -name bash \
  -a ! -name dir \
  -a ! -name env \
  -a ! -name rm \
  -a ! -name find \
  -a ! -name grep \
  -a ! -name touch \
  -a ! -name sed \
  -a ! -name cat \
  -a ! -name coreutils \
  -a ! -name test \
  -delete
  ;;

nodejs)
  find /usr/bin ! -type d \
  -a ! -name cd \
  -a ! -name ls \
  -a ! -name sh \
  -a ! -name bash \
  -a ! -name dir \
  -a ! -name env \
  -a ! -name rm \
  -a ! -name find \
  -a ! -name cat \
  -a ! -name coreutils \
  -a ! -name test \
```

```
    -delete
    ;;

  python)
    find /usr/bin ! -type d \
    -a ! -name cd \
    -a ! -name ls \
    -a ! -name sh \
    -a ! -name bash \
    -a ! -name dash \
    -a ! -name dir \
    -a ! -name env \
    -a ! -name rm \
    -a ! -name find \
    -a ! -name coreutils \
    -a ! -name test \
    -a ! -name python* \
    -a ! -name pip* \
    -delete
    ;;

  tomcat)
    find /usr/bin ! -type d \
    -a ! -name cd \
    -a ! -name ls \
    -a ! -name sh \
    -a ! -name bash \
    -a ! -name dash \
    -a ! -name dir \
    -a ! -name env \
    -a ! -name rm \
    -a ! -name find \
    -a ! -name mkdir \
    -a ! -name coreutils \
    -a ! -name test \
    -a ! -name chown \
    -delete
    ;;

  *)
    find /usr/bin ! -type d \
    -a ! -name cd \
    -a ! -name ls \
    -a ! -name sh \
    -a ! -name bash \
    -a ! -name dir \
    -a ! -name env \
    -a ! -name rm \
    -a ! -name find \
    -a ! -name coreutils \
    -a ! -name test \
    -delete
    ;;
esac
```

```
# Remove broken symlinks (because we removed the targets above).
find $sysdirs -xdev -type l -exec test ! -e {} \; -delete

rm -rf /tmp/*
```

Use the following example as a reference for applying the hardening script in your Docker build process.

Example Dockerfile for hardening"

```
FROM registry.gamewarden.io/yourcompany/yourimage:1.0.0

USER 0

COPY hardening-scripts/hardening-script.sh /tmp/hardening-script.sh

RUN chmod +x /tmp/hardening-script.sh \
    && /tmp/hardening-script.sh 65532 nginx

USER 65532
```

Platform security validation

Game Warden's platform security is validated through:

- Continuous security reviews by the government.
- Regular third-party penetration testing.
- Managed enforcement of security controls across infrastructure, Kubernetes platforms, and applications.

FAQs

Why does Second Front request Dockerfiles when containers are pushed?

We request Dockerfiles, a text file containing a set of instructions that automate the process of building a Docker image, to support our custom container hardening process. Our hardening scripts remove unnecessary users, files, and commands from your base image. By reviewing your Dockerfile, we can better understand what elements your container needs to function correctly after hardening, allowing us to secure your container without breaking critical functionality.

Does an image go through the full pipeline on every run?

No. After the first full pipeline run, subsequent runs are significantly faster. Game Warden will compare vulnerabilities against previously approved justifications and only prompt for justification of any new findings.

What does the Game Warden custom hardening process include?

The hardening process varies depending on what your container includes (e.g., databases). Generally, it involves:

- Creating a non-root user with only necessary permissions.
- Removing unused user accounts and login shells.
- Locking down system directories and file permissions.
- Deleting unused files, binaries, kernel tunables, crontabs, and broken symlinks.

- Whitelisting only essential executables needed for your container to function.

We will work with your team to confirm that the hardened image preserves expected functionality.

What tool does Game Warden use to generate the Software Bill of Materials (SBOM)?

Game Warden uses Anchore Enterprise to generate SBOMs.

What security scanning tools does Game Warden use, and which ones should I use?

Game Warden uses the following tools:

- **Trivy** and **Anchore Enterprise** (which uses Grype and Syft) for CVE detection.
- **Anchore Compliance** for compliance checks.
- **ClamAV** for malware detection.

These scanners generate security findings that appear in Findings in the Game Warden web app.

We recommend that you pre-scan your images before pushing to Game Warden. Open-source tools such as Trivy and Grype can help identify most issues surfaced by our platform scanners. These tools are cost-effective and well-suited for early testing, even though they may not catch certain DoD-specific findings.

Observability in Game Warden

Game Warden's observability stack provides real-time visibility into application behavior, performance, and logs. It leverages the following tools:

Grafana

Grafana is a data visualization platform that allows you to **query, analyze, and visualize your application logs and metrics**. With Grafana, you can build interactive dashboards and panels for operational insights into your deployed services.

Grafana also supports **log-based alerting**, enabling you to set up alerts based on specific log patterns or errors detected in your application.

Loki

Loki is the logging backend that **stores and queries log data from your application**. It collects logs and runtime system data from your containers.

Loki organizes logs into streams and supports powerful querying through LogQL.

Mimir

Mimir is responsible for **metric aggregation**, **querying**, and **alerting**. It collects and stores time-series metrics from your applications and infrastructure, providing a robust backend for metric-driven insights in Grafana.

Alloy

Alloy handles the collection of **telemetry data** across your environment, integrating logs, metrics, and metadata for unified observability.

(Coming Soon) Tempo

Tempo will provide **tracing aggregation** and **querying** to help you visualize and troubleshoot distributed applications. Stay tuned as we roll out this functionality!

> **Note**
>
> This guide is intended for application engineers who are already familiar with technical concepts and need to explore customer-specific data within Grafana. It **does not** cover basic Grafana functionality.

Access Grafana dashboard

AFWERX deployments

The Grafana endpoints you'll use are managed by the Second Front team. Most production endpoints follow the same convention as your AFWERX endpoint:

`grafana-<company-id>.il2.afwerx.dso.mil`

Note that DEV, STG, and PRD environments each have their own unique URLs. If you need help identifying your specific environment endpoints, contact your Mission Success Manager.

You can access Grafana using your Platform One (P1) SSO credentials and Government Access Card.

1. Go to your Game Warden Grafana endpoint and click **Sign in with Game Warden SSO**.

2. On the **Log in or sign up** page, click **Continue With P1 SSO**.
3. Enter your P1 credentials and Multi-Factor Authentication (MFA) code.

DISA deployments

The Grafana endpoints you'll use are managed by the Second Front team. Most production endpoints follow the same convention as your DISA endpoint:

`grafana-<company-id>.il2.2f.mil`

Note that DEV, STG, and PRD environments each have their own unique URLs. If you need help identifying your specific environment endpoints, contact your Mission Success Manager.

Accessing the Grafana dashboard requires the same network access (NIPRNet-based IP) and authentication (CAC/ECA via Keycloak) protocols outlined in Access Control for DISA IL4 & IL5.

FedRAMP deployments

The Grafana endpoints you'll use are managed by the Second Front team. Most production endpoints follow the same convention as your FedRAMP endpoint:

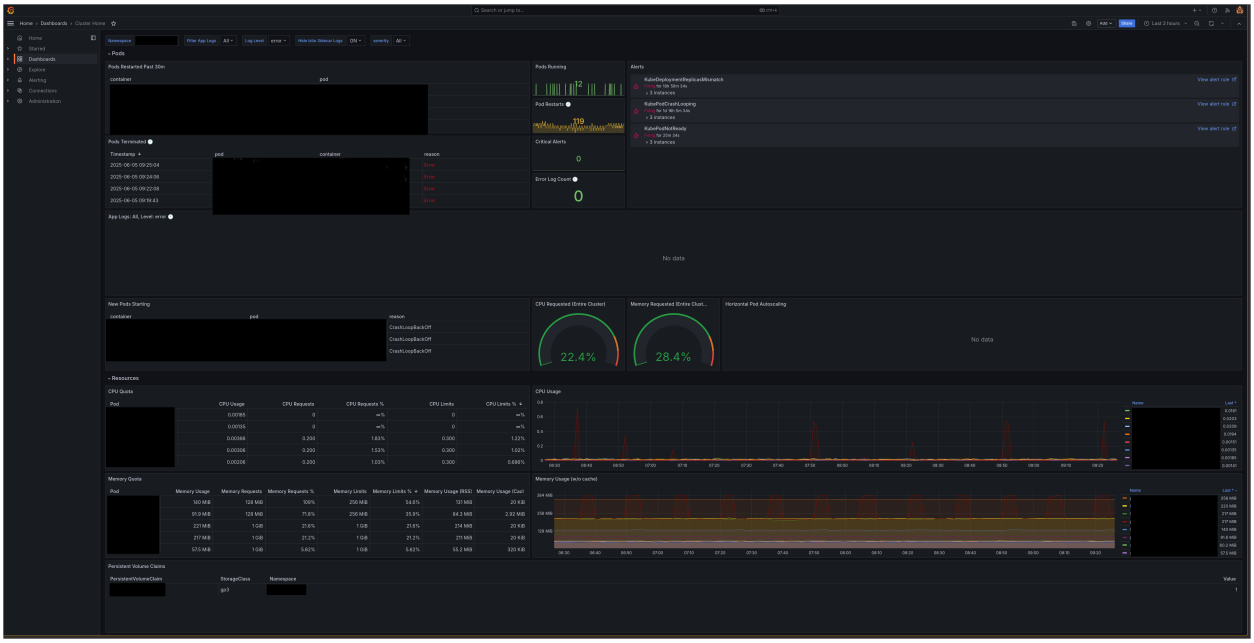
`grafana-company-id.fedramp.gamewarden.io`

Note that DEV, STG, and PRD environments each have their own unique URLs. If you need help identifying your specific environment endpoints, contact your Mission Success Manager.

You can access Grafana using your Game Warden credential.

1. Go to your Game Warden Grafana endpoint and click **Sign in with Game Warden SSO**.
2. On the **Log in or sign up** page, enter the username and password you created during registration.
3. Click **MFA Log In**.
4. Enter the current code from your authenticator app and click **MFA Log In**.

After logging in, you'll be directed to the Grafana dashboard.

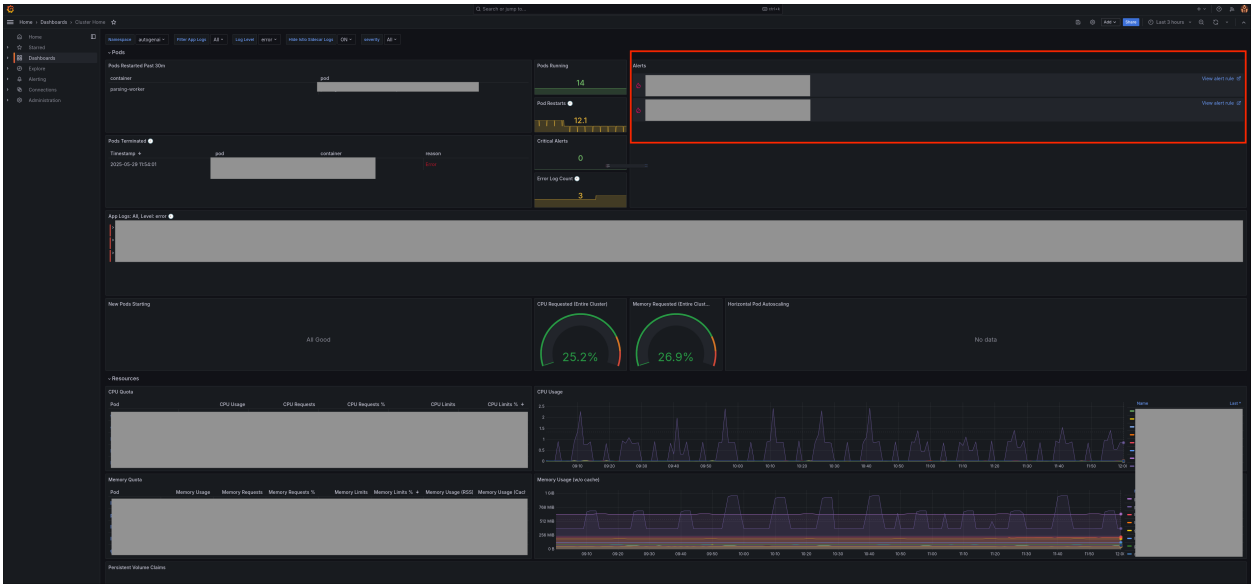


Common tasks in Grafana

Review the tabs below for common tasks you'll perform in Grafana.

Monitoring alerts

The Grafana home dashboard includes a dedicated **Alerts panel** to help you monitor which alerts are actively firing within your application namespace.



What's included?

The stack includes over **200 preloaded Prometheus alerts** and **200+ recording rules** for monitoring both application and infrastructure-level behavior. These alerts are also listed on the **Alerting** page in Grafana.

> **Tip**
>
> For detailed alert descriptions, see the Prometheus runbooks.

Alert behavior

- By default, Prometheus-based alerts are configured to send notifications to Slack. Certain alerts—such as those related to Kubernetes control plane components—are automatically silenced in cloud environments, as they are only relevant for bare-metal deployments.
- As a customer, you will receive alerts that are scoped specifically to your application namespaces. Meanwhile, infrastructure and platform-level alerts are monitored and managed by the Second Front team.

Manage custom alerts

- You **cannot modify** default Prometheus/Mimir alerts directly (e.g., silencing or adjusting intervals).
- You **can create custom Grafana alerts** using Grafana's native alert engine:
  - These alerts support querying both logs and metrics.
  - A one-time configuration by Second Front is required to route custom alerts to Slack.

Creating queries and dashboards

Using the Query Editor (Explore)

1. In the left navigation pane, go to **Explore > Queries**.
2. Select your datasource from the top-left dropdown (e.g., `ORG-prd-customer`).
3. Toggle the **Builder/Code** option to **Code**. This allows you to write a LogQL query using the following format:

   ```
   {log-stream-selector} | log-pipeline
   ```

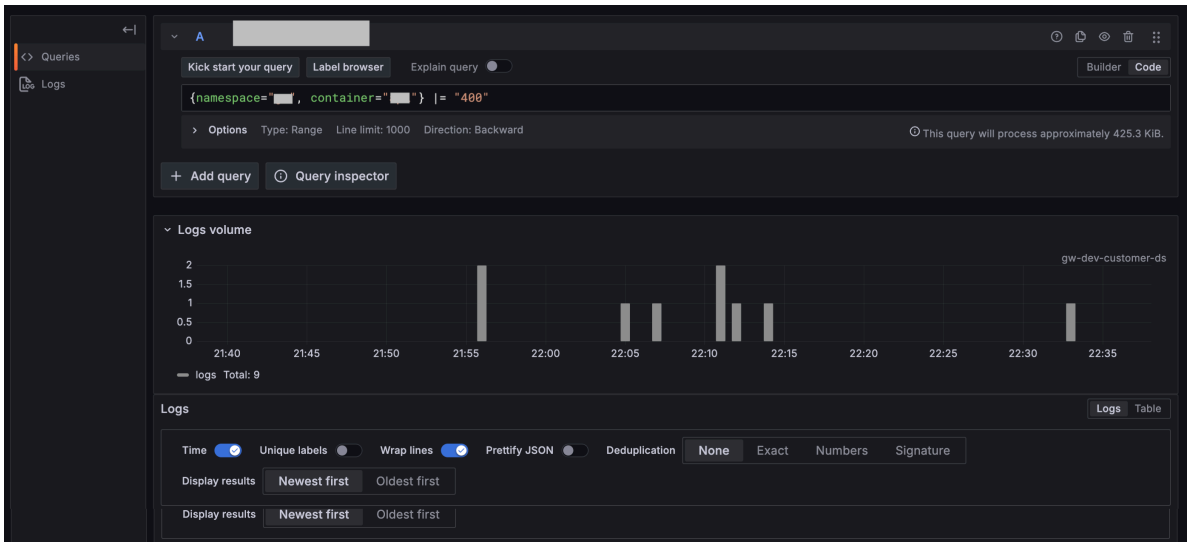
   For example, this query retrieves log entries containing the HTTP 400 error code:

   ```
   {namespace="YOUR_APPLICATION_NAMESPACE", container="CONTAINER_NAME"} |= "400"
   ```

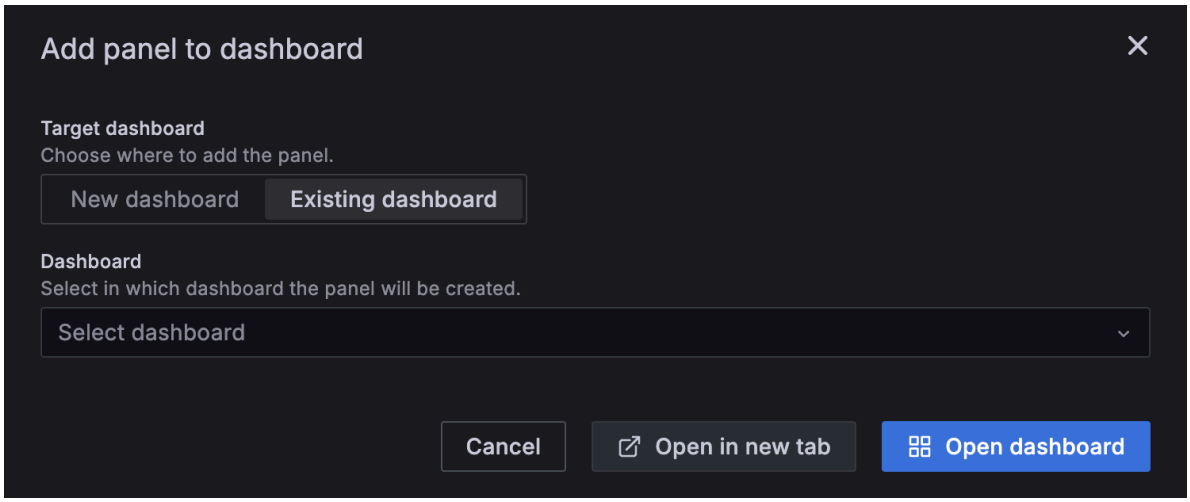
   And this entry is a common access log format, often generated by web servers like Nginx or Apache:

   ```
   127.0.0.6 - - [09/Jan/2023:22:40:17 +0000] "POST /api/request-path HTTP/1.1" 400 2 "-" "HTTP-AGENT/
   ```

4. You can toggle to the **Builder** UI and use fields such as **Metrics**, **Label Filters**, **Operations**, and **Legends** to help construct your query. Grafana provides auto-complete suggestions as you enter information.
5. In **Code** view, you'll also find a **Label browser** button (for log datasources) or a **Metrics browser** button (for metrics datasources).
6. Click **Run Query** (located to the left of the **Live** button) to view results.



7. Click **Add > Add to dashboard** and choose whether to open the query results in an existing or new dashboard, then click **Open dashboard**.



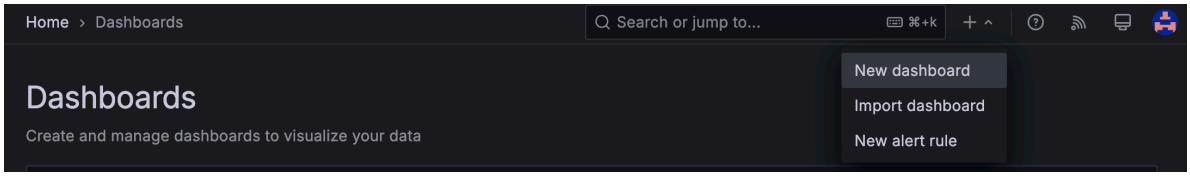
8. Click **Save dashboard**, provide a name and select a folder location. You can star the dashboard later for quick access.

   Tip

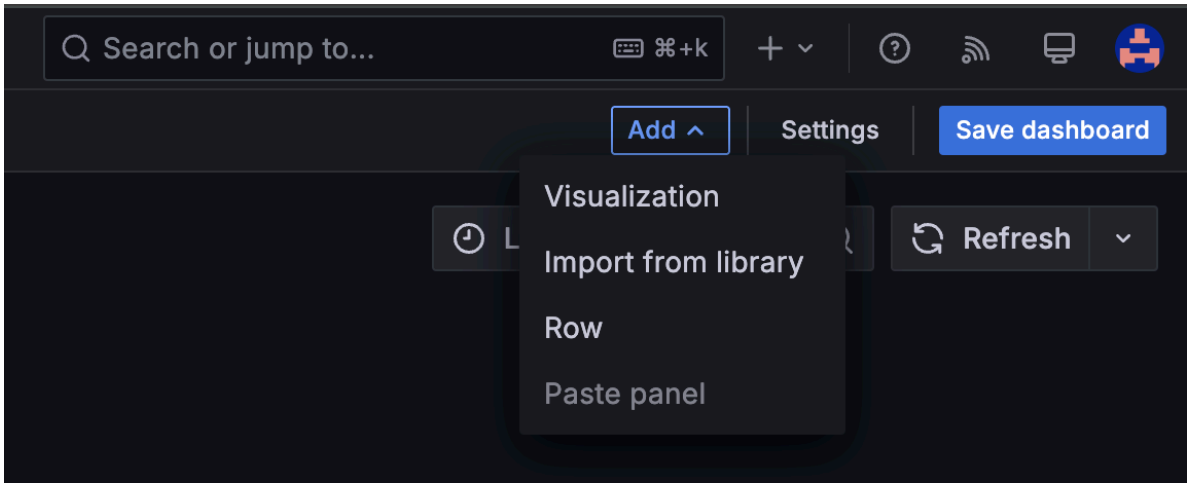
   Ensure log formats are consistent across your application to improve query reliability and log parsing.

Create dashboards

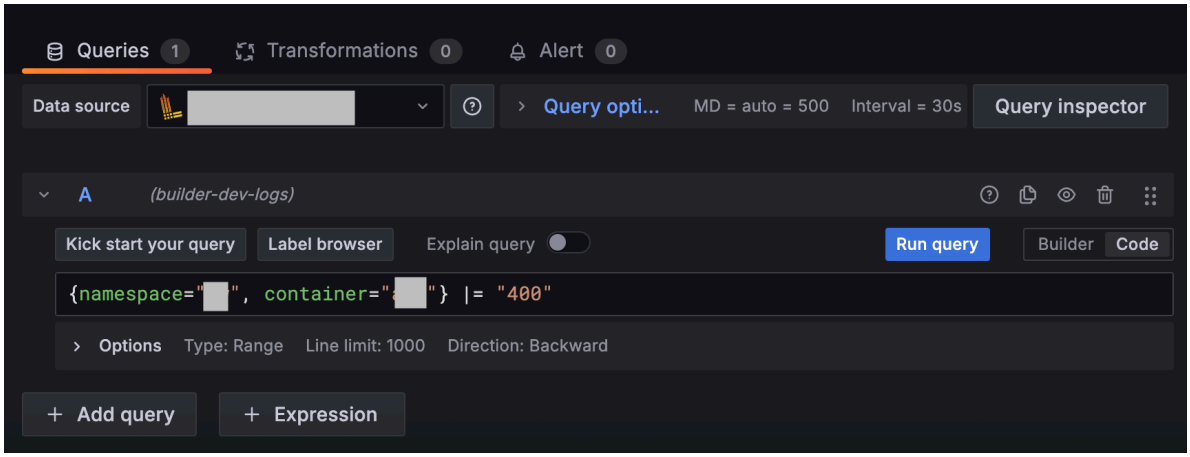
1. In the left navigation pane, go to **Dashboard > New dashboard**.



2. From the **Add** dropdown in the top-right corner, select **Visualization**.



3. Select your datasource from the top-left dropdown (e.g., `ORG-prd-customer`).
4. Create a query, then click **Run query**.



5. Click **Save dashboard**, then provide a name and select a folder location. You can star the dashboard later for quick access.

Troubleshooting: Example scenarios

Below are two common scenarios that highlight the importance of checking beyond container logs and incorporating Istio traffic and authentication insights in your troubleshooting process.

Scenario 1: Missing traffic or "No Healthy Upstream"

When traffic is completely absent from your service, it's often due to either an **authentication issue** or an **Istio-related networking issue**.

In Kubernetes clusters, while you can use container logs to look at service-specific events, much of the networking flow occurs through Istio sidecars. These sidecars act as independent containers within your pods and are not always reflected in the main container logs.

To investigate, go to the **Explorer** page in Grafana and query logs based on the **destination of the virtual service** in question.

For example, if the service name is `api`, query logs for Istio interactions and virtual service forwarding. Below is an example LogQL query:

```
{namespace="YOUR_NAMESPACE", container="istio-proxy"} |= "api"
```

- `{namespace="YOUR_NAMESPACE", container="istio-proxy"}` => This targets logs from the Istio sidecar (`istio-proxy`) within your namespace.
- `|= "api"` => Filters the logs to include entries that mention "api", which is typically how the virtual service or destination service would appear in the logs.

Look for log entries such as `no_healthy_upstream`, which suggests that traffic is not reaching a healthy endpoint or is being blocked by Istio.

```
2025-06-02 13:52:21.646 [2025-06-02T13:52:21.600Z] "GET / HTTP/1.1" 503 UH no_healthy_upstream - "-" 0 19 0 -
2025-06-02 13:52:21.646 [2025-06-02T13:52:21.565Z] "GET / HTTP/1.1" 503 UH no_healthy_upstream - "-" 0 19 0 -
2025-06-02 13:52:21.646 [2025-06-02T13:52:21.552Z] "GET / HTTP/1.1" 503 UH no_healthy_upstream - "-" 0 19 0 -
```

These logs can arise due to:

- Destination endpoint issues (e.g., the target service is unavailable or misconfigured).
- Istio routing rules that do not match the intended destination.

Scenario 2: Authentication failure

Game Warden deploys a cluster-wide authentication service called **authservice**, which manages traffic authorization based on rules defined for each application.

When traffic is blocked due to authentication issues, you'll see log entries indicating failed authentication attempts, often using codes such as `UAEX ext_authz_denied`.

To troubleshoot:

- Use **namespace-scoped** queries in Grafana to capture logs from all relevant services, not just individual containers.
- Look for authentication errors in logs and identify which service or token is failing the checks.
- Determine if the authorization policies need to be adjusted or if authservice rules need updates to accommodate new traffic patterns.

  Best practices for troubleshooting

  - While container-scoped logs are helpful for **isolating issues in a specific service**, using **namespace-scoped logs** provides better visibility into how services interact—especially for traffic routing and authentication flows.
  - In most cases, start with **namespace-scoped** queries to understand the full service communication context, and narrow down to container logs only when investigating a specific process or isolated error.

Best practices

Logging

Use structured logs:

- Format log messages as **JSON** or **key-value pairs** to make them easier to parse and analyze.
- Structured logging improves searchability, enables more precise queries, and ensures compatibility with observability tools like Loki and Grafana.
- Use structured logging libraries available in your programming language or implement consistent key-value formatting.

Include log levels:

- Incorporate standard log levels such as `DEBUG`, `INFO`, `WARNING`, and `ERROR`.
- Most logging libraries include log levels automatically. Make sure these levels are not stripped out or suppressed in production logs.
- Log levels allow Grafana to **filter logs by severity**, helping you quickly focus on critical issues.

Balance logs and metrics:

- If you're frequently logging an `INFO`-level message **just to track the frequency of an event**, consider creating a **metric** instead.
- Metrics (like counters and gauges) are more efficient for monitoring event counts and trends over time, while logs should focus on **diagnostics and warnings**.
- Reserve logs for events that need investigation (e.g., unexpected errors or critical warnings).

(Coming Soon) Metrics

We recommend integrating the **Prometheus client library** into your application to expose a `/metrics` endpoint. This enables:

- Collection of **custom application metrics** to complement your logs.
- Monitoring of event frequency, performance, and other key indicators directly in Grafana.

More guidance on integrating metrics and setting up the `/metrics` endpoint will be added soon.

Observability

Add contextual information:

- Enrich log entries with **high-cardinality identifiers** such as **event IDs**, **transaction IDs**, or **user IDs**.
- These identifiers support **traceability** and make it easier to correlate logs across services and requests.

Prepare for event tracing:

- Adding this contextual information now lays the groundwork for **event tracing**—a feature our SRE team will support as the observability stack matures.
- Specialized Grafana tooling for distributed tracing will be introduced in future updates to help monitor end-to-end application performance.

Support and resources

Ticketing and pairing sessions

To request technical assistance or customizations:

- Submit a pairing session request via the Support Ticketing system.
- After onboarding, customers receive up to four hours of pairing support for observability-related issues.

Resources

Loki

- Log Queries
- Query Examples

Grafana

- Getting Started with Grafana Dashboard Design
- Grafana Webinars and Videos
- Grafana Dashboards/Grafana Play Home
- Grafana Tutorials
- Awesome Grafana

Metrics (Prometheus and PromQL)

- Writing PromQL Queries
- Creating Prometheus Alerts
- Best Practices for Prometheus Alerting
- Setting up a /metrics Endpoint (Prometheus Docs)

Grafana Alerts

- Grafana Alerting Overview

Mimir

- Grafana Mimir Documentation

Alloy

- Grafana Alloy Documentation
- Alloy Concepts and Configuration

Tracing

- OpenTelemetry Documentation
- Tempo Documentation

Managed Services

Game Warden provides three types of managed services to simplify your deployment and reduce operational overhead:

Amazon Web Services (AWS)

Game Warden supports key AWS services including:

- Backup
- Elastic File System (EFS)
- Elastic Kubernetes Service (EKS)
- Relational Database Service (RDS)
- Simple Email Service (SES)
- Simple Storage Service (S3)
- DynamoDB

Big Bang Services

These include commonly used tools such as:

- HashiCorp Vault
- Istio

Game Warden–Managed Services

Game Warden may pull container images from Iron Bank – a DoD-approved container image repository – and run them within your Kubernetes environment.

As a customer, you are **not responsible** for supplying container images or remediating Common Vulnerabilities and Exposures (CVEs) for managed services. Game Warden handles this responsibility through internal policies, manual updates, and automated CI/CD pipelines.

Amazon Web Services (AWS)

Game Warden is hosted on AWS GovCloud (US-East). The tabs below outline the AWS services most commonly supported by Game Warden, along with their availability across DoD Impact Levels (IL), FedRAMP, and Commercial deployments.

DoD Deployment

Service Name	IL2	IL4	IL5	IL6
Aurora (Amazon Aurora MySQL)	Yes	Yes	Yes	Yes
DocumentDB (Amazon DocumentDB)	Yes	Yes	Yes	No
EBS (Elastic Block Store)	Yes	Yes	Yes	Yes
EC2 (Elastic Cloud Compute)	Yes	Yes	Yes	Yes
EFS (Elastic File Storage)	Yes	Yes	Yes	Yes

Service Name	IL2	IL4	IL5	IL6
EKS (Elastic Kubernetes Service)	Yes	Yes	Yes	Yes
ElastiCache	Yes	Yes	Yes	No
IAM (Identity and Access Management)	Yes	Yes	Yes	No
KMS (Key Management Service)	Yes	Yes	Yes	No
RDS (Relational Database Service)	Yes	Yes	Yes	Yes
SQS (Simple Queue Service)	Yes	Yes	Yes	No
S3 (Simple Storage Service)	Yes	Yes	Yes	Yes **Note**: Do not use MinIO for S3 bucket access. Utilize IAM Roles for Service Accounts (IRSA) to manage permissions securely.
VPC (Virtual Private Cloud)	Yes	Yes	Yes	No
SES (Simple Email Service)	Yes	Yes	Yes	No
Transit Gateway	Yes	Yes	Yes	No
Backup	Yes	Yes	Yes	No
DynamoDB	Yes	Yes	Yes	No

Warning

For Top Secret deployments, only EKS, RDS and S3 services are currently available.

FedRAMP & Commercial Deployments

Service Name	**FedRAMP**	**Commercial**
Aurora (Amazon Aurora MySQL)	Yes	Yes
DocumentDB (Amazon DocumentDB)	No	Yes
EBS (Elastic Block Store)	Yes	Yes
EC2 (Elastic Cloud Compute)	Yes	Yes
EFS (Elastic File Storage)	Yes	Yes
EKS (Elastic Kubernetes Service)	Yes	Yes
ElastiCache	Yes	Yes
IAM (Identity and Access Management)	Yes	Yes
KMS (Key Management Service)	Yes	Yes
RDS (Relational Database Service)	Yes	Yes
SQS (Simple Queue Service)	Yes	Yes
S3 (Simple Storage Service)	Yes	Yes
VPC (Virtual Private Cloud)	Yes	Yes

Service Name	FedRAMP	Commercial
SES (Simple Email Service)	Yes	Yes
Transit Gateway	Yes	Yes
Backup	Yes	Yes
DynamoDB	Yes	Yes

Applications should run inside your Kubernetes cluster as containerized workloads. Game Warden can support certain in-cluster services—such as service mesh or secrets management—while AWS-managed services such as RDS or S3 are hosted externally and accessed over the network.

> **Note**
>
> To ensure compatibility with our Kubernetes-based platform, we recommend containerizing the required functionality as part of your application deployment.
>
> If you're exploring serverless architecture, Knative offers a Kubernetes-native alternative that supports event-driven workloads and may serve as a suitable substitute. Our platform supports running Knative within your Kubernetes cluster as part of a containerized solution.

Google Cloud Platform (GCP)

The tabs below depict popular GCP services, their associated support status on Game Warden, and their availability at each Impact Level (IL):

Support

Game Warden currently supports the following services for customers:

Service Name	IL2	IL4	IL5
Cloud Identity	Yes	Yes	Yes
Google Kubernetes Engine (GKE)	Yes	Yes	Yes
Google Cloud Storage (GCS)	Yes	Yes	Yes
Virtual Private Cloud (VPC)	Yes	Yes	Yes

Can support

Game Warden can support the following services. Contact our Product team to confirm availability.

Service Name	IL2	IL4	IL5
Cloud HSM (Hardware Security Module)	Yes	Yes	Yes
Cloud Logging	Yes	Yes	Yes
Cloud Monitoring	Yes	Yes	No
Cloud Pub/Sub	Yes	Yes	No
Cloud SQL	Yes	Yes	No

Planned + Coming soon

Game Warden plans to support the following services soon. If interested, contact our Product team so we can prioritize accordingly.

- BigQuery
- Cloud Key Management Service

Not supported

Game Warden does not currently support the following services.

Service Name	IL2	IL4	IL5
Dataflow	No	No	No
Persistent Disk	No	No	No

> **Warning**
>
> GCP does not currently support IL6 or Top Secret deployments.

Big Bang

Big Bang is the underlying architecture that powers the Game Warden platform. Built on a Department of Defense (DoD)-approved framework, it provides a standardized set of services that run within the Kubernetes cluster provisioned via AWS—where your application is deployed.

Big Bang–managed services can be configured to run inside your Kubernetes cluster and may include tools such as HashiCorp Vault for secrets management and Istio for service mesh functionality. These services are deployed and maintained by the Game Warden team in accordance with security and operational requirements.

Iron Bank

Iron Bank is a Department of Defense (DoD)-approved container image repository that hosts hardened, continuously monitored images for use in secure environments. Game Warden can source container images from Iron Bank to support managed services within your Kubernetes cluster.

For example, if your application requires a caching service such as Redis (Remote Dictionary Server), Game Warden can deploy a Redis container image from Iron Bank—provided it meets our Acceptance Baseline Criteria. Only approved images are pulled and deployed to ensure compliance with DoD security standards.

Support and deployment

Game Warden provisions managed services upon request to ensure alignment with your application's needs and deployment context. These services are not automatically included and should be identified early in your engagement with the Game Warden team.

If you require managed services, we recommend communicating this need as early as possible—ideally in your Authorization Boundary Diagram, during onboarding, or in pre-sales discussions. The Game Warden team should be aware of your request prior to application deployment. If a managed service need arises later, you can still submit a request via Slack or by contacting us via Slack.

For services such as Backup, EFS, EKS, RDS, SES, and S3, the Game Warden team uses infrastructure as code (IaC) to provision and configure the necessary components, and connect them to your Kubernetes cluster. These services operate outside of the cluster but are tightly integrated. For example, we can create an S3 bucket and configure the necessary permissions for your cluster to access it. Deployments of services such as RDS, S3, and EFS are seamless to customers and can be supported at all DoD Impact Levels (ILs).

Requests for other managed services are subject to review. The team will evaluate the specific service, verify its alignment with security and compliance requirements, and determine whether it is authorized at the

requested IL. Additional government approvals may be necessary, particularly for IL4 and higher, and Game Warden leadership may be involved in the approval process.

Data Retention & Disposition Policy

Data retention describes what data must be stored and for how long. This data is commonly used for disaster recovery, network forensics, network analytics, and cybersecurity investigations. Proper data retention standards minimize the company's attack surface and prevent the accumulation of unnecessary data and the resulting costs.

This policy letter establishes the data retention policy for all production information stored on cloud infrastructure owned or operated by Second Front (2F). The scope of this policy includes all AWS regions and any other providers that host infrastructure managed by 2F for the Game Warden Platform as a Service (PaaS).

---

1. Data categorization & retention periods

This section defines types of data and how long each category must be retained. Retention periods are based on security, operational, and compliance requirements.

Access, authentication, and administrative auditable events

Retention period: 1 year

Audit scope includes:

- Successful and unsuccessful login attempts
- Privileged or system-level activity
- Session start and end times
- Concurrent logins from different workstations
- Access to protected objects or resources
- Program initiations and direct system access
- Account creation, modification, disabling, and termination
- Kernel module loads, unloads, and restarts

Backup requirement:

- Audit records must be backed up **at least weekly** to a **separate system** from the source.

Network activity logs

Retention period: 1 year

Includes:

- All **inbound and outbound** network traffic
- All **internal** traffic within the environment

Mission-critical backups

Retention period: 3 months

Includes:

- Backups and snapshots of systems, applications, and data that are **critical to organizational survival**

---

2. Superseding procedures

Retention periods explicitly stated in local network or system operating procedures may supersede this document **if they extend** the retention timelines defined above. This policy establishes the **minimum baseline** for how long data should be retained in cloud environments.

---

3. Superseded policies

This policy supersedes the **Second Front (2F) Data Retention Policy dated 20 August 2021**.

---

4. Reference documents

This section lists the authoritative documents and government directives that inform this policy.

Publication	**Title**	**Date**
Directive-type Memorandum (DTM) 22-001	DoD Standards for Records Management Capabilities in Programs Including Information Technology	02 June 2023
DoDD 5400.07	DoD Freedom of Information Act (FOIA) Program	05 April 2019
DoD Instruction 5015.02	DoD Records Management Program	24 February 2015
NARA Pubs	National Archives and Records Administration General Records Schedule	Latest

Significant Software Changes and Authorization Requirements in Game Warden

The Department of Defense (DoD) utilizes Game Warden, a Platform as a Service (PaaS) solution, to streamline cloud-hosted application development, deployment, and operations. Understanding how software changes within the Game Warden environment impact security and authorization is crucial. Game Warden defines significant software changes for applications and outlines when a new Deployment Passport is necessary based on deltas in a given cyber risk posture.

Defining significant changes

> **Security review requirement**
>
> **Any** significant change to your application will trigger a new security review and Deployment Passport.

A significant software change in a Game Warden-hosted application refers to any modification that could:

- **Alter the security posture**: Introduce new vulnerabilities, change data flows, add new user populations, or expose new attack surfaces within the Game Warden environment.
- **Materially increase cyber risk**: Expand the potential for unauthorized access, data breaches, or service disruptions specific to the Game Warden platform.

Game Warden considerations

- **Shared Responsibility Model**: Game Warden manages the security of the underlying Game Warden infrastructure and platform.
- **Built-in security features**: Game Warden incorporates various security features, such as continuous monitoring, automated security testing, and vulnerability scanning. Significant changes may require re-evaluation to ensure continued compliance with the terms of the authorization to operate.
- **Compliance requirements**: Game Warden facilitates compliance with DoD security standards. Significant changes may require re-evaluation to ensure continued compliance with the terms of the authorization to operate.

Examples of significant changes

- **Modifying core application logic**: Changes to the primary functionality or workflow of the application hosted on Game Warden.
- **Altering data handling**: Changing how sensitive data is stored, accessed, or processed within the Game Warden environment.
- **Integrating new services or APIs**: Adding new third-party integrations within Game Warden that could expose data or introduce new vulnerabilities.
- **Customizing security configurations**: Modifying firewall rules, access controls, or encryption settings within the Game Warden platform.
- **Leveraging new Game Warden features**: Utilizing newly released Game Warden features that significantly alter the application's behavior or capabilities.
- **Major system changes**: Adding application containers or services.

Routine updates vs. Significant changes in Game Warden

Feature	Routine Update	Significant Change
Scope of Change	Minor code adjustments, configuration tweaks, or content updates within the Game Warden environment.	Changes to core application logic, data handling, or integration with external services on Game Warden.
Security Impact	Minimal impact on the security posture or risk exposure of the Game Warden application.	Likely to introduce new vulnerabilities, alter data flows, or change the overall attack surface.
Authorization	Typically covered under the existing authorization and do not require a new review by the AO.	Requires a new authorization from the AO to ensure the modified software meets security standards.
Examples	Deploying new versions of the application with bug fixes, updating configuration files, or adding like content within Game Warden.	Implementing a new feature, integrating with a new API, or changing data storage mechanisms on Game Warden.

There are new change management requirements that necessitate a formal Security Relevant Change (SRC) be completed and signed off by the government. Depending on the impact of the change, approval authority will be our DIU Information Systems Security Manager (ISSM), the Government Security Controls Assessor (SCA), or the Authorizing Official (AO) himself for significant changes.

Changes to Game Warden, including our pipelines, IaC, SaC, and processes to yield deployment passports must be carefully evaluated before making the change. The Security Champions in the Product teams are the best way to ensure that proposed changes are correctly managed.

AWS Bedrock integration and updates

Before integrating Amazon Bedrock on Game Warden, use steps here to determine how your application will use the service.

Integrating AWS Bedrock into an application is considered a **significant change**:

- If the application is adding AWS Bedrock and the configuration matches Game Warden Baseline, a new Deployment Passport is required.
- If the application is adding AWS Bedrock and the configuration deviates from Game Warden Baseline, a new Deployment Passport and SRC is required.
- If the application has an approved Deployment Passport with AWS Bedrock and wants to change what AWS model is used, no Deployment Passport or SRC is required.

  Current Game Warden Baseline includes

  1. AWS Bedrock Guardrails
  2. Assessed IAM User role policy
  3. Assessed IAM Service role policy
  4. Customer application is a single tenant deployment
  5. All AWS FedRAMP High approved models within AWS GovCloud East
  6. AWS Bedrock RAG is limited to customer S3 buckets deployed as part of the CtF/Software Approval
  7. AWS Bedrock VPC only allows egress to the VPC Cidr (10.0.0.0/16)
  8. No AWS Bedrock Agents

For deployments in AWS GovCloud (US-East), visit Model support by AWS Region in Amazon Bedrock to check which models are currently supported.

Headlamp: Runtime Dashboard Guide

Headlamp is the new web-based dashboard for managing workloads and resources within your Game Warden Runtime environment. It replaces the legacy Argo UI and provides a secure, read-only view into your deployed applications, using the same identity and access model you already use today.

Headlamp offers a more complete view of your runtime cluster, including workloads, configuration, and status information, while remaining fully compliant with the zero-trust security controls of the Game Warden platform.

Key features

Category	Description
Unified Cluster View	Explore your application namespaces, workloads, configuration, and logs from a single interface.
Real-Time Status	Live updates for pods, deployments, and jobs without needing to refresh the page.
Secure Authentication	Uses the same Keycloak login process as the Argo UI—no new accounts or credentials required.
Role-Based Access Control	Headlamp enforces Kubernetes RBAC so users only see what their assigned role allows.
Namespace-Scoped Visibility	Customer users see only their organization's namespaces and deployed resources.
Performance and Accessibility	Built-in dark mode, keyboard shortcuts, and accessibility enhancements optimized for daily use.
Extensible by Design	Built on an extension framework that enables additional runtime-specific views and features in the future.

How to access Headlamp

Your Headlamp dashboard is available at a dedicated URL in the following format:

`https://headlamp-<customerName>.secondfront.com`

For example, if your organization name is ACME, your URL would be:

`https://headlamp-acme.secondfront.com`

> **Tip**
>
> Bookmark your Headlamp URL for quick access. It replaces your previous Argo UI link.

How to authenticate

Follow the steps below to access your Headlamp dashboard using your platform credentials (the same ones you use for Argo UI).

1. Go to your organization's Headlamp URL and click **Sign In with OIDC**.
2. Sign in through Keycloak using your platform credentials.
3. After successful authentication, you will be redirected to the Headlamp dashboard.

Access and user roles

Headlamp uses the same role-based access control (RBAC) model as the underlying runtime cluster. There are two primary personas:

Persona	Description	Access level
Administrator	Second Front operators and approved administrators who manage the runtime platform.	Full access across all namespaces and runtime resources.
Customer User	Customer personnel with access to their assigned application namespaces.	Read-only access within their organization's namespaces (for example: view Deployments, Pods, Logs, Events, and ConfigMaps).

You cannot modify or delete resources through Headlamp. This read-only access model helps ensure platform consistency and security.

Availability

- **Default Access**: Headlamp is automatically deployed for all customer runtimes beginning with version `v0.16` and later.
- **Older Versions**: If your runtime has not yet been upgraded, please reach out to your Second Front representative to request enablement.
- **Single Sign-On**: Headlamp uses your existing Keycloak identity. No separate credentials or manual provisioning are required.

Migration notes: Argo UI → Headlamp

Headlamp replaces Argo UI as the primary dashboard for monitoring deployed workloads. GitOps automation continues to run through Flux, no action required from you.

Use the table below to locate Argo UI functions in Headlamp:

Argo UI Function	Headlamp Location
Application sync and status	**Workloads** → **Deployments** or **GitOps** → **Flux Status** (extension)
Pod logs	**Workloads** → **Pods** → **Logs**
Events	**Workloads** → **Events**
Resource health	Status indicators on individual workload pages
Namespace visibility	Namespace selector in top navigation bar

---

FAQs

Do I need to request access to Headlamp?

No. All customers have access by default through their organization's Headlamp URL.

Can I make changes to my applications in Headlamp?

No. Headlamp is read-only for customers. All changes should continue to be made through your standard processes.

Will my login change from Argo UI?

No. You will sign in using the same Keycloak credentials you already use.

Can administrators view all customer namespaces?

Yes. Platform administrators retain full cluster visibility.

Support and feedback

If you encounter issues or would like to request additional visibility features, open a ticket through the Support Ticketing System.

App Central (Beta)

App Central is the central hub for managing your applications and deployments on the Game Warden platform. From here, you can:

1. Add new applications
2. View and complete the Body of Evidence (BoE) for associated deployments
3. Monitor service pipeline statuses and security findings in Findings
4. Deploy images to both Staging (STG) and Production (PRD) environments

App Central Legacy vs. New App Central Redesigned

We've redesigned App Central from the ground up to make your journey more intuitive and reduce friction at every step:

- **Streamlined deployments**: Multi-view Images table (Latest | Development Environment | All Deployments) with environment targeting eliminates guesswork and accelerates safe promotions.
- **Security visibility**: Instant CVE overview across environments plus clear image statuses (Approved, Pending, CVEs Present, Ready to Submit) so teams know exactly when to act.
- **Compliance tracking**: Authorization renewal deadlines and Body of Evidence (BoE) access are managed in one place, ensuring your compliance process moves smoothly.
- **Better navigation**: Per-app pages with organized tabs and application switcher dropdown for faster workflow.

App Central walkthrough

When in App Central, you can view and take action on all key aspects of your applications and deployments.

Feature	Description
Images tab	Displays all images for the selected application, organized by environment (e.g., Development (DEV), STG, and PRD). Shows vulnerability findings and their statuses. From here, you can deploy qualified images to STG and PRD environments.
Authorizations tab	Lists all deployments created for the selected application, along with their associated Impact Levels (ILs). From this page, you can complete the BoE for each deployment.
Application Details tab	Displays the application's basic and administrative details. You can update these details at any time.

Add new applications

To get started with a new application:

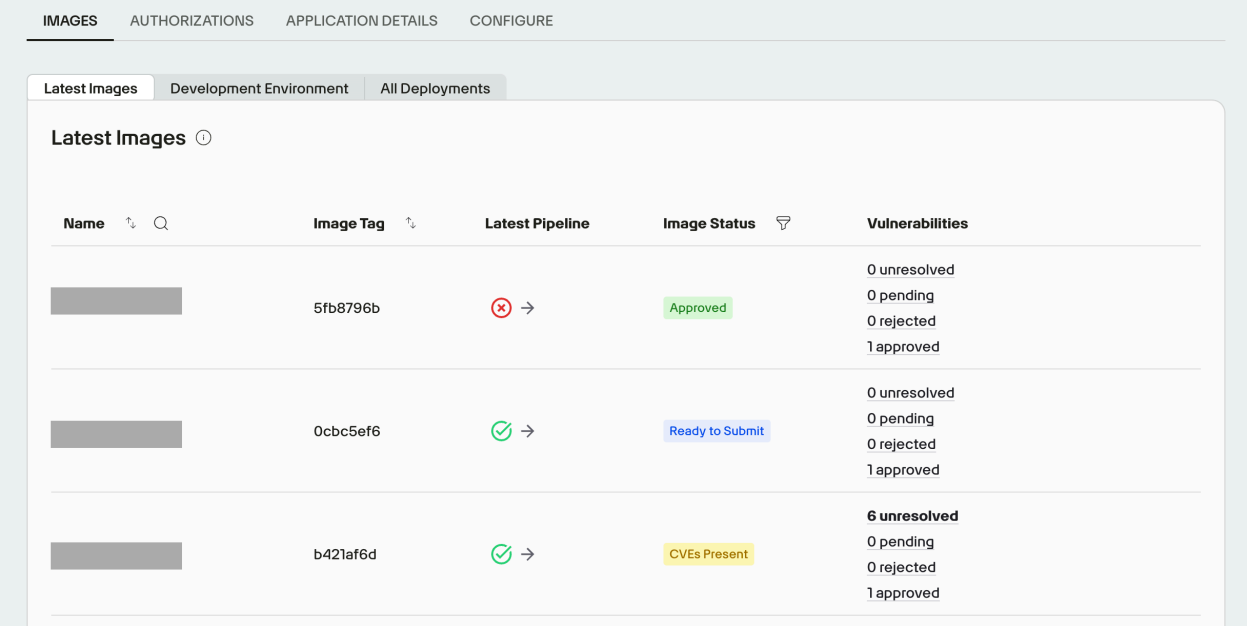
1. If this is your first application in Game Warden, click **Add new application**. Otherwise, click the dropdown list of all your applications and select **Add new application**.
2. Enter a name for your application, then click **Save** to create the application.
3. To provide application details, select the **Application Details** tab.
4. Enter the following information:
   - Basic information about the application (name, URL, description)
   - Company contact information

- Details of your Government Sponsor
- Proof of Authorization, such as a Certificate to Field (CtF)/Software Approval or an Authorization to Operate Letter

5. Click **Save changes**.

Now that your application has been created, the next step is to push its images to Harbor Registry. See Push Images to Harbor.

The images will then be built and hardened automatically. Once this process is complete, the images and their vulnerability findings will appear in the **Images** tab in App Central.



Complete the BoE

For each deployment, you are required to complete the BoE—a formal document that explains how your application meets Game Warden's Authority to Operate (ATO) security requirements. It is a critical component in obtaining your CtF/Software Approval.

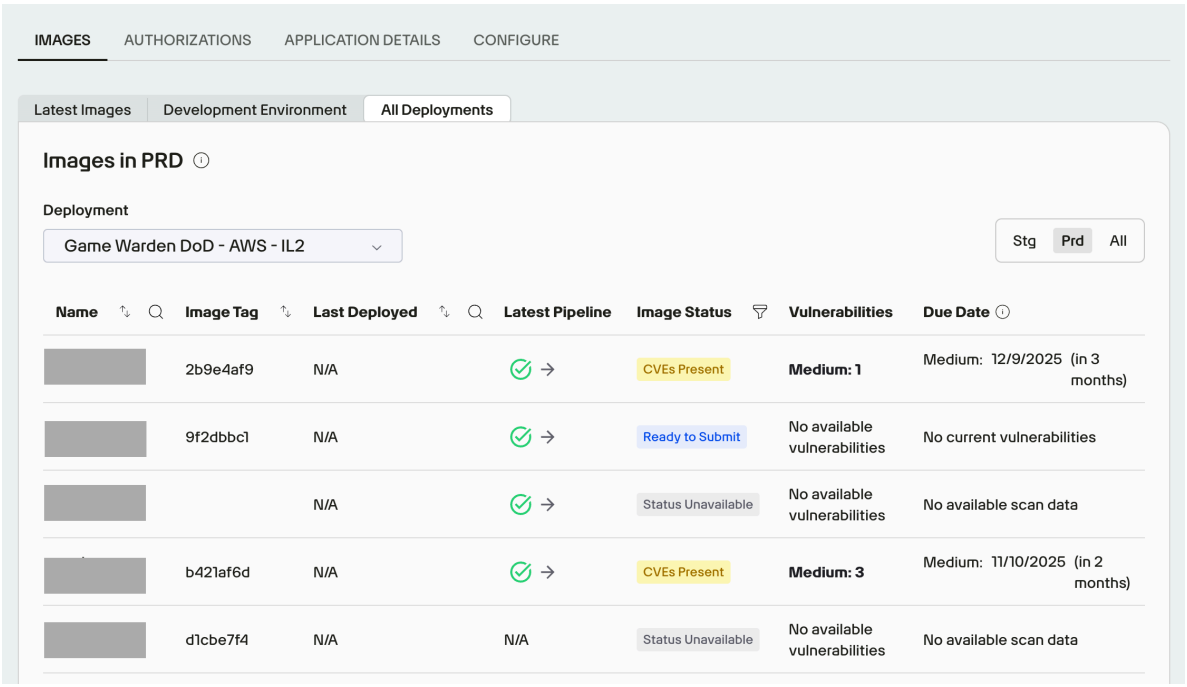
To complete the BoE:

1. Select the **Authorizations** tab, then scroll down to the **Deployments** panel.
2. Click the **Body of Evidence** button to open the BoE page. See Complete Body of Evidence for instructions on how to complete each section.

Deploy images to STG and PRD environments

Within the **Images** tab, you can:

- View recently pushed images –> select **Latest Images**.
- View images in the DEV environment –> select **Development Environment**. To see all discovered CVEs in DEV, go to the Findings page.
- View images currently deployed across STG and PRD –> select **All Deployments**. To see which image version is currently used in DEV, STG, and PRD, set the view toggle on this tab to **All**.



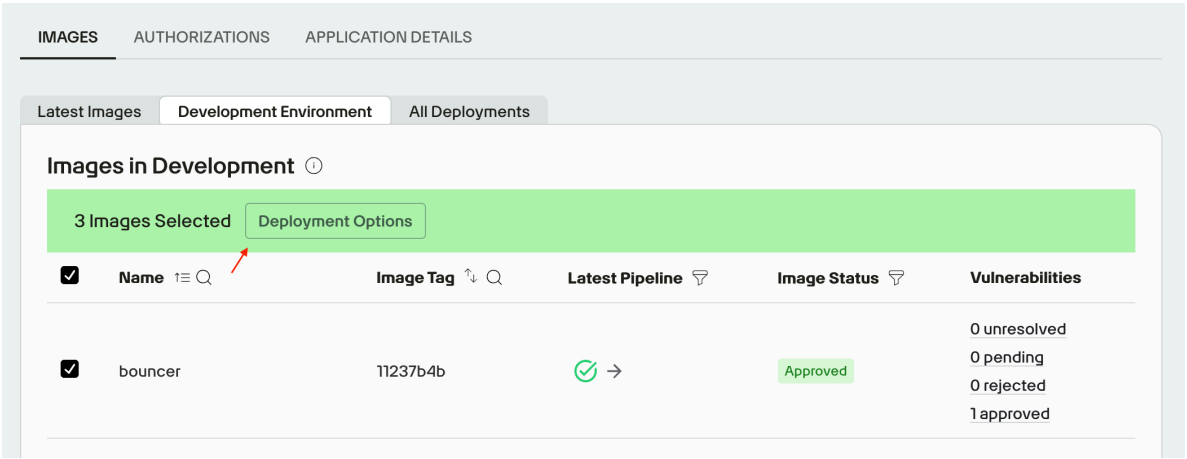
- Deploy **Approved** images from DEV to STG and PRD. If an image is associated with multiple deployments, you can deploy to a specific deployment or to all applicable deployments using the bulk deployment feature.

Bulk deployments

Bulk deployments let you deploy images to different deployments, or across environments within the same deployment (e.g., DEV → STG → PRD).

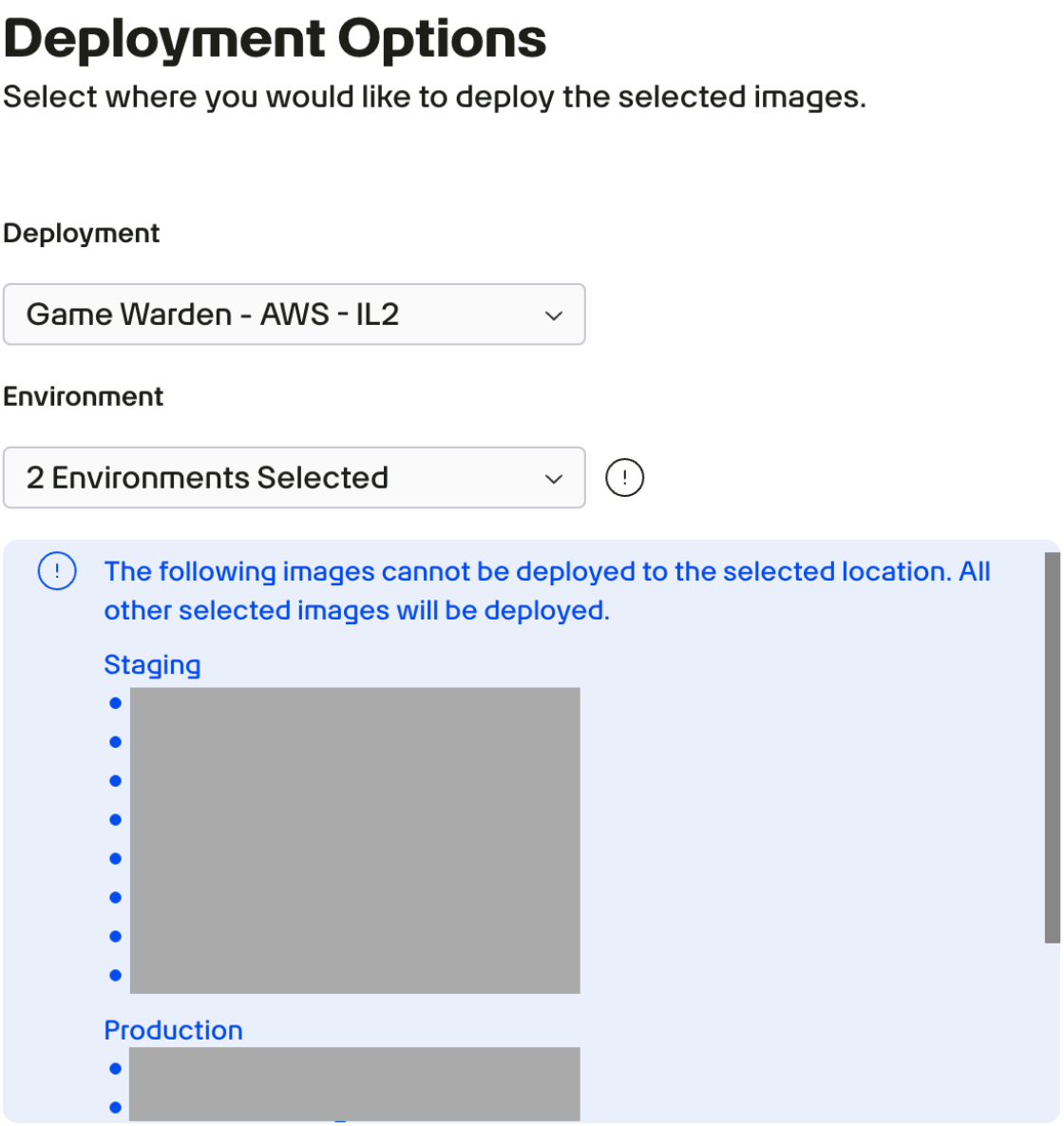
Steps to bulk deploy images:

1. Go to the **Images** tab and select **Development Environment**.
2. Select the images you want to deploy, then click **Deployment Options**.



3. In the **Deployment Options** panel, select the deployment and environment(s) to which you want to deploy the selected images.

   If any selected images don't meet the requirements for a given environment, they will be automatically excluded.



4. To perform multiple deployment actions in one workflow, click **Add More Deployments**.
5. Click **Confirm Selection** to complete.

Findings Overview

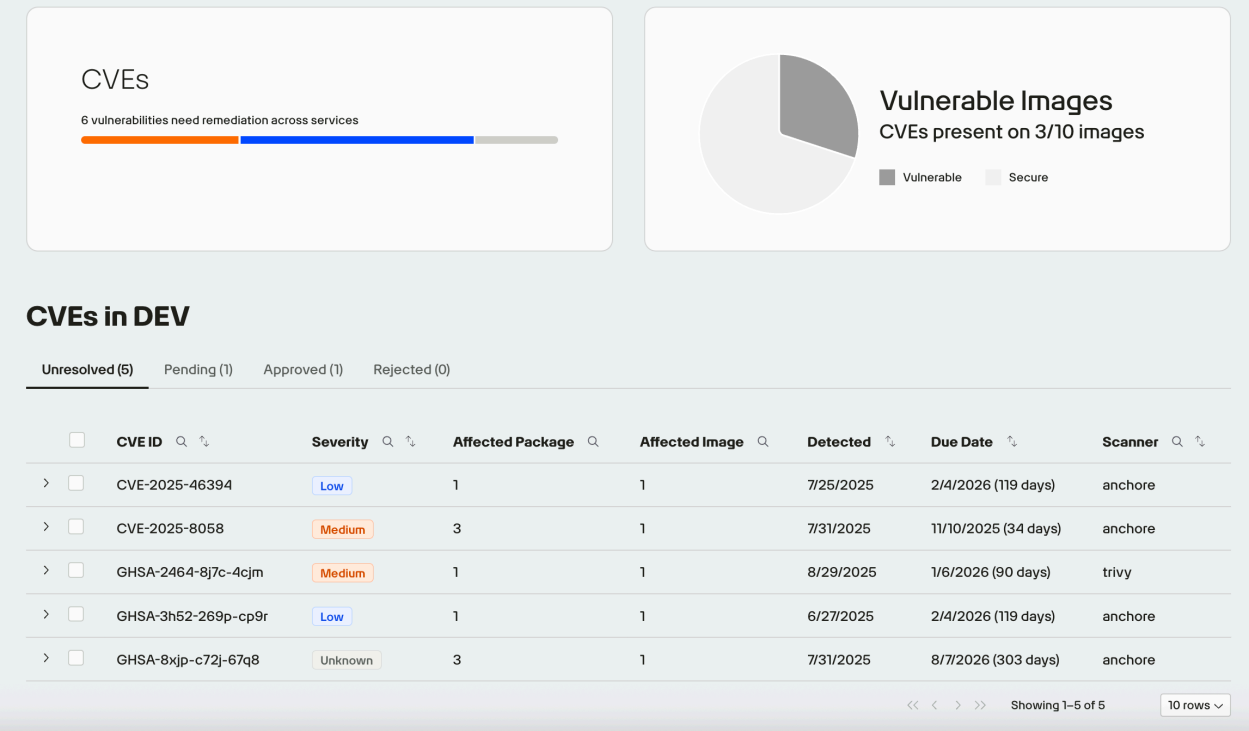
The Findings page provides an aggregated view of all Common Vulnerabilities and Exposures (CVEs) detected across your Development (DEV), Staging (STG), and Production (PRD) environments. Use it to view, resolve, or justify CVEs, address Anchore compliance results, and review security team responses to your proposed resolutions.

The sections below guide you through accessing findings in each environment.

Review findings in DEV environment

To view all CVEs detected in DEV, click **Findings** in the left navigation. From this page, you can:

- View discovered CVEs with affected package and image counts
- Submit justifications (risk acceptance, compensating controls, false positives)
- Review and remediate Anchore policy and compliance findings
- Track responses and dispositions from Second Front



Once you are granted access to your Harbor repository, you can begin pushing your containerized images. Our pipelines will automatically scan these images using integrated security tools and populate the Findings page with the results.

Scan time varies depending on the size and complexity of the image. If results do not appear in the Findings page after pushing your image, please notify our team by submitting a support ticket.

Review scan results

On the Findings page, information is organized to help you focus on CVEs and identify the top vulnerability priorities for your application. Each application package has its own Findings view with corresponding scan results.

The numbers display in the **Affected Package** and **Affected Image** show you how broadly a CVE is present in your application. Expand each row to view the exact affected packages and images.

Expand a row to see its affected packages and images. Click any affected image to view findings and download the raw scan data and Software Bill of Materials (SBOM).

Severity status hierarchy

Vulnerabilities are categorized by severity using industry-standard scoring:

Severity	Description
Critical	Immediate remediation required. These could be exploitable remotely and have high impact.
High	Should be addressed quickly but may not pose as severe a risk as Critical.
Medium	Important to fix, but not an urgent blocker.
Low	Minor issues or informational findings.
Policy	Issues that violate platform-defined Anchore compliance policies.

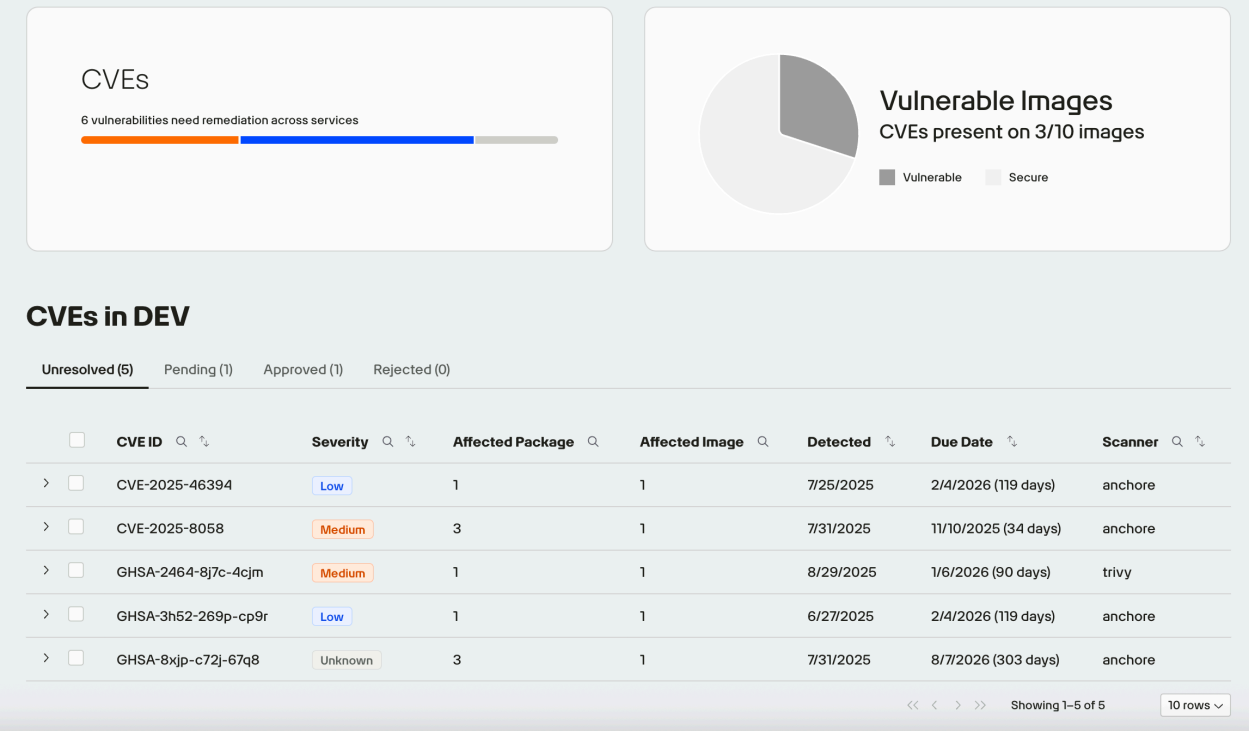
Resolution tracking

Each vulnerability is assigned one of the following statuses. Use the tabs to filter by status and track progress and decisions.

Status	Description
Unresolved	No action taken yet.
Pending	Justification submitted, awaiting review from Security.
Rejected	Justification denied; remediation is required.
Approved	Justification accepted; no further action needed.

Bulk actions for justifications

You can select multiple packages at once and apply a single justification to address the same vulnerability across all selected packages. This streamlines the process when multiple components share a common issue and resolution rationale.

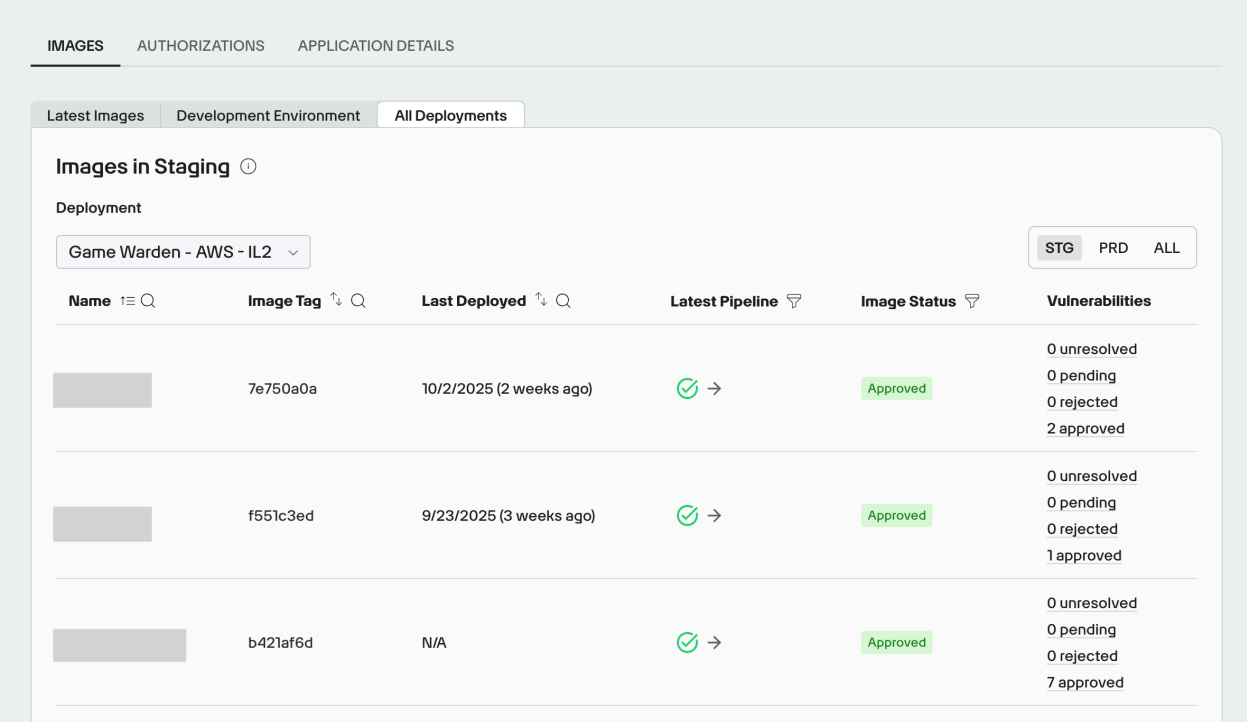


After you submit a justification, it moves to the **Pending** tab for review by the 2F Security team. If needed, you can edit the justification from this tab.

Review findings in STG and PRD environments

To view image findings for your app in STG and PRD:

1. In **App Central**, open the **Images** tab and click **All Deployments**.
2. Use the **Deployment** dropdown to select the deployment you want to review. *Apps can have multiple deployments across impact levels and may be deployed to STG and/or PRD within the same impact level.*
3. Use the **Environment** toggle (top right) to filter findings by **STG**, **PRD**, or **All**.
4. For each image, click the links in the **Vulnerabilities** column to jump to its CVE status view, with counts by justification status (Unresolved, Pending, Rejected, and Approved).



Warning

- If a CVE is detected in an image currently deployed to the PRD environment, you must upload a new image with the vulnerability remediated. Justifications for unresolved CVEs are only accepted for images in the DEV or STG environments.
- To submit your resolutions to the Game Warden team, you must address **all** security findings for your image.

Request security review

For DoD, FedRAMP and Commercial deployments, once all findings have been resolved or properly justified, click the **Ask for security review** button to send a review request. The Game Warden security team will review your submissions and either approve or deny your proposed resolutions.

If a new pipeline is run against the image under security review, the status will change to customer review and the **Ask for security review** button will need to be selected again to send a review request for the newly run pipeline push.

Tip

To help ensure your review is completed within a timely manner, focus on the **quality** of your submission:

- **Detailed Justifications**: For any accepted risk or exception, provide clear, specific, and well-supported justification and mitigation strategies *before* submission in Scan Lab.
- **Clear Documentation**: Ensure your documentation is well-structured and complete at the time of submission to reduce the need for back-and-forth clarification within Findings.

Download FedRAMP ConMon artifacts

After successfully resolving or properly justifying every vulnerability within Findings, the documentation **must be submitted** for official resolution and approval, involving the customer, their agency, and the 3PAO.

Follow these steps:

1. Go to **App Central** in Game Warden.
2. Open your application, click **Authorizations** and go to the **Document Repository** under the specific Deployments.

3. Select **Images**, then click the **pencil** icon.
4. From the dropdown, choose the relevant **image version**.
5. Click the **Export Vulnerabilities JSON** button.
6. Provide the generated **machine-readable JSON file** to the Security Advisory Services and Agency for review and approval.

Download artifacts within Findings

When viewing detailed findings of an image, click the **Download Artifacts** button in the upper-right corner to download your raw scan data and SBOM. This makes it easier to share information or work offline.

Production image scanning

Game Warden performs automated vulnerability scans on all customer-deployed **production container images** on the **first calendar day of each month**. This recurring scan schedule ensures that newly disclosed vulnerabilities are detected and surfaced in a timely manner.

When vulnerabilities are identified, your team will receive notifications through both **Slack** and the **Game Warden app**. These alerts enable prompt triage and resolution of security findings.

CVE due dates

- **CVE remediation deadlines** are enforced only after your application has been deployed to the PRD environment. Each CVE includes a **due date** for remediation or justification. This date aligns with the **Remediation/Justification timeline** specified in Table A of the Game Warden Acceptance Baseline Criteria.
- Your team is responsible for either resolving the CVE or submitting an acceptable justification by the stated deadline.

Body of Evidence Overview

A Body of Evidence (BoE) is a formal document that explains how your application meets Game Warden's Authority to Operate (ATO) security requirements. It is a critical component in obtaining your Certificate to Field (CtF)/Software Approval. The BoE includes required external approvals and proof of an active government contract for your organization.

Once submitted, the Game Warden security team will review the BoE as part of your Deployment Passport.

BoEs are specific to Production (PRD) environments and must align with the designated Impact Level (IL) of your deployment. **You are required to create a separate BoE for each IL environment your application will support.**

The BoE template includes the following components:

Component	Description
Deployment Information	Captures details about your application's architecture, technologies, programming languages, and government contract information.
Information Security	Documents how your application handles confidentiality, integrity, availability (CIA), PII, CUI, and security classification.
Images	Lists the version currently in use for each of the images in your application.
Authorization Boundary Diagram	A visual diagram that shows your application's architecture, data flows, and external connections to define its security boundary.
Role Identification	Identifies key stakeholders such as the government sponsor, system owner, product owner, and security manager.
Business Continuity	Provides emergency contact details for key personnel responsible for resolving outages or security incidents.
CAC Personnel	Lists individuals with Common Access Card (CAC) or similar credentials who require access to IL4+ environments.
SAST	Details your Static Application Security Testing tools and findings as part of the secure development lifecycle. Your organization is responsible for conducting SAST scans on the source code repositories from which containers are built and deployed to Game Warden.

Component	Description
DAST	Documents results from Dynamic Application Security Testing, used to detect runtime vulnerabilities. Second Front (2F) Systems performs and includes DAST artifacts in your application's Authorization Package.
SSDF Attestation	Affirms your development practices align with the Secure Software Development Framework (SSDF) as required by DoD guidance.
CtF Recommendation Memo	A memo summarizing the security posture of the application and recommending it for Certificate to Field (CtF)/Software Approval issuance.
AI Attestation	A required form disclosing whether AI is used in your application, and assessing associated risks and controls. → AI Attestation Form

---

What's next?

- **Create & Manage BoEs**: Step-by-step instructions for creating, updating, and managing your BoEs.
- **Complete BoEs**: Learn how to complete all sections of the BoE to ensure they are align with the designated IL of your deployment.
- **Authorization Boundary Diagram**: Diagram showing your application's architecture, data flow, and external connections.
- **Understand ATO and Deployment Passport**: Learn how Game Warden's ATO, Deployment Passport, and CtF/Software Approval work together to enable compliant deployments.

Create and Manage Body of Evidence

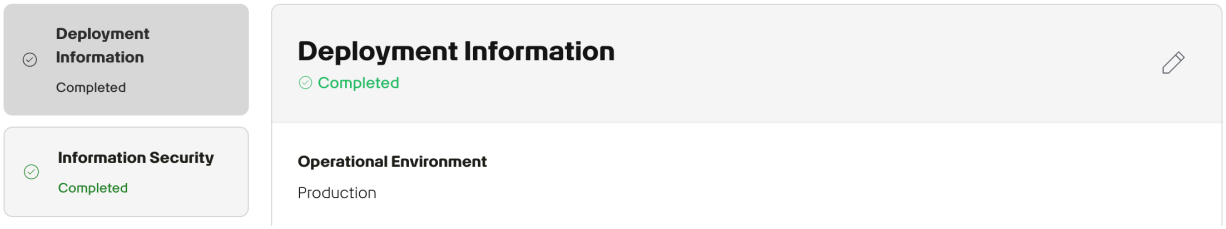
This guide walks you through how to create a Body of Evidence (BoE) each Impact Level (IL) environment where you plan to deploy your application.

Create a BoE

1. On the App Central page, select the **Authorizations** tab.
2. Scroll to the **Deployments** panel and click **Body of Evidence** to open the BoE page.
3. Complete the required sections of the BoE using the guidance in Complete Body of Evidence. For each section in the BoE, click **Save Draft** to save the progress you made on the form. Click **Save & Submit for Review** to mark the form as complete. If validation passes, the form status updates to **Complete**. If not, you will see field-level guidance to finish remaining items.

Manage existing BoEs

To resume or edit a BoE, click **Body of Evidence** to return to the page. Completed sections are marked in green for easy tracking, and each section can be edited using the pencil icon.



Best practice

Review and update your BoE monthly to reflect any changes in your application, especially as development progresses through the Software Development Lifecycle (SDLC).

Complete Body of Evidence

This guide walks you through how to complete all sections of a Body of Evidence (BoE) to **ensure alignment** with the designated Impact Level (IL) of your deployment.

Deployment Information

The Deployment Information form captures key technical and operational details about your application. Accurate and complete entries help the Game Warden security team evaluate your system's configuration and ensure compliance with DoD ATO requirements.

Below is a breakdown of each required field with guidance:

- **Operational Environment** - Describe the environment where your application will run, such as: Production; Test; Staging; Dev; Other. **Second Front is responsible for completing this field.**
- **Authorization Status** - Select from the available options to indicate the deployment's current status: Certificate to Field (CtF); Authority to Operate (ATO); Active; Inactive. **Second Front is responsible for completing this field.**
- **Assessment Date** - Provide the planned or completed date of the security assessment. **Second Front is responsible for completing this field.**
- **Security Reassess Interval** - This is the interval at which your system's security posture will be re-evaluated by Second Front. It should align with the security timeline outlined in the CtF authorization or Software Approval document. Consult your Technical Implementation Manager for more information.
- **List of Programming Languages** - Provide a comprehensive list of all programming languages used to build your application (e.g., Python, JavaScript, Go). Include any backend or scripting languages relevant to deployed services.
- **List of Dependencies** - List the services your application depends on — whether managed by Game Warden (e.g., AWS S3, SES) or managed by your team (e.g., external data connections or in-boundary containers/services deployed by the customer). **Note:** All dependencies should be reflected in the Authorization Boundary diagram.
- **List of Databases** - Indicate the type and technology used (e.g., PostgreSQL, MongoDB, DynamoDB). Mention whether it is managed by the customer or uses a Bedrock service.

  Tip

  - Be as specific and complete as possible.
  - Coordinate with your security and engineering teams to validate this information before submission.

Information Security

The Information Security form documents the sensitivity of the data your application handles and the security controls in place to ensure it aligns with DoD confidentiality, integrity, availability expectations, and data protection requirements.

Complete each field based on your current or planned deployment:

- **Confidentiality** – Confidentiality values are derived from security categorization of the application in tandem with your Mission Owner.
- **Integrity Level** – Integrity values are derived from security categorization of the application in tandem with your Mission Owner.
- **Availability Level** – Availability values are derived from security categorization of the application in tandem with your Mission Owner.
- **Classification Level** – Select the data classification your application will handle (e.g., Unclassified, CUI, Secret, TS/SCI, TS/SAP). This must align with the approved IL level of your deployment.

  - **Note:** If your organization does not have a Security Classification Guide (SCG), select **No** and leave this field blank. If you are unfamiliar with the SCG, select **Unsure** and leave this field blank.
- **Distribution Control Type** – Indicate any restrictions on data dissemination, such as NOFORN (Not Releasable to Foreign Nationals), ITAR(International Traffic in Arms Regulations), HIPAA (Health Insurance Portability and Accountability Act), or FEDCON (Federal Employees and Contractors Only).
- **Controlled Unclassified Information (CUI)** – Select applicable high-level CUI categories (e.g., NNPI, Intel, PRVCY, OPSEC, Other). You must **list the exact types of CUI** present in your application (e.g., biometric data, health records, personnel security clearance forms).
  - **Note:** If your application does not contain Personally Identifiable Information (PII), select **No** and leave this field blank. If you are unsure whether your application contains PII, select **Unsure** and leave this field blank.

Tip

- Coordinate with your mission owner or security officer to confirm data types and classification.
- Ensure your confidentiality, integrity, and availability levels align with the system's operational context and compliance obligations.
- Be specific—general statements may delay approval.

Images

Provide the version currently in use for each of the images in your application.

Important

- Images are updated to the current versions and match the versions reflected in your authorized environment.
- If an image will not be included in the application for which the CtF/Software Approval is being pursued, select **Excluded** from the dropdown.
- Review and attest to authorize the deployment of the selected services.

Authorization Boundary Diagram

An Authorization Boundary Diagram (ABD) is a visual representation of your system's software components, data flows, and security boundaries. As part of the BoE, you must provide a diagram that includes any external systems or services your application connects to outside of its deployment boundary. Include the system name, purpose, and any sensitive data exchanged.

See Authorization Boundary Diagram for more information.

Role Identification

The Role Identification form captures key stakeholders responsible for overseeing and supporting your application deployment in Game Warden.

You must provide the following details for each required role:

- Full Name
- Title
- Organization
- Email Address
- Phone Number

Required roles

- **Government System Owner** – The government official ultimately responsible for the application and its operation within the DoD environment. This person ensures the system complies with security requirements and approves changes or risk decisions.
- **Government Contract Sponsor** – The government representative responsible for funding and contractual oversight of the deployment. They are often the primary liaison between the government customer and your company.
- **Government Prime Contractor** – If your company is a subcontractor, this field should identify the prime contractor accountable for contract delivery. If you're the prime, indicate your own company's contracting POC.
- **Company Product Owner** – The individual at your company responsible for application functionality and delivery. They are expected to be the primary point of contact for questions about feature development and app roadmap.
- **Company Security Manager** – The security lead within your organization who ensures that security practices align with Game Warden requirements. This person will also coordinate with the Game Warden Security team during incidents or audits.

Business Continuity

You must provide **at least two to three emergency contacts** for each deployment.

Required fields for each contact

- **Full Name** – The full name of the emergency contact.
- **Email** – A monitored email address to reach the contact quickly.
- **Phone** – A direct phone number, preferably a mobile number, for urgent communication.
- **Title** – The individual's job title or role (e.g., DevOps Engineer, Security Lead).
- **Preferred Contact** – Indicate the preferred method of contact (e.g., email, phone, both).

  Important

  - This section is **required**. It ensures that Second Front can contact the appropriate individuals in case of emergency events—such as outages, zero-day vulnerabilities, or critical security incidents—affecting your application.
  - Ensure that the listed contacts are aware of their responsibilities and authorized to act on behalf of your organization during incidents.

CAC Personnel

For deployments in **Impact Level 4 (IL4)** or **Impact Level 5 (IL5)** environments, personnel must possess a valid Government Access Card—such as a CAC (Common Access Card), ECA (External Certificate Authority), or PIV (Personal Identity Verification)—in order to access the environment, including system logs.

You must provide information for **all company personnel and engineers** who hold applicable Government Access Cards and will be involved with this application.

Required fields for each CAC holder

- **Full Name** – The full name of the individual holding the access card.
- **Title** – Their job title or role within the company.
- **DoD Number** – The ID number found on the back of the CAC (e.g., DoDID#).
- **Expiration Date** – The expiration date printed on the access card.

  Important

Ensure all listed information is accurate and up to date to avoid access delays during IL4/IL5 deployment or support activities.

SAST

Upload the results from the Static Application Security Testing (SAST) you've performed on your application. SAST artifacts must be current — no older than 30 days before submission.

DAST

Documents results from the Dynamic Application Security Testing, used to detect runtime vulnerabilities. Second Front (2F) Systems performs and includes DAST artifacts in your application's Authorization Package.

SSDF Attestation

Based on their review of your application, Second Front security team will upload this document to confirm that your development practices align with the Secure Software Development Framework (SSDF), as required by DoD guidance.

AI Attestation

The AI Attestation section collects details about how your application uses Artificial Intelligence (AI) and/or Machine Learning (ML).

---

1. AI/ML usage

Start by confirming whether your application incorporates AI and/or ML technologies:

- **Artificial Intelligence**: Does your application use any AI-related functionality?
  - Select **Yes** if AI is used in any form (e.g., chatbots, recommendation systems, automation).
  - Select **No** if your application has no AI features.
- **Machine Learning**: Does your application use machine learning algorithms or models?
  - Select **Yes** if ML is used for tasks such as prediction, classification, or pattern detection.
  - Select **No** if ML is not part of your application.

---

2. Business case

Explain the reasoning behind your use of AI and the problems it solves:

- **Why does your application require AI?** - Provide a concise explanation (e.g., "to automate document classification").
- **What use case(s) are addressed by AI and what value is expected to be realized?** - Describe how AI improves your application and what benefits it delivers (e.g., better user experience, improved accuracy).

---

3. Type of AI

Select the types of AI used in your application:

- Predictive Analytics
- Machine Learning
- Deep Learning
- Natural Language Processing
- Computer Vision
- Reinforcement Learning
- Ensemble Model
- Generative AI

---

4. Machine Learning details

If your application uses ML:

- **Specify the Model Type**
  - Supervised Machine Learning
  - Unsupervised Machine Learning
  - Reinforcement Machine Learning
- **Are you using a foundational pre-trained model?** - Select **Yes** or **No**.
- **Have you done additional training on the model?** - Select **Yes** or **No**.

---

5. Machine Learning security

Provide responses to the following security considerations:

- Is customer data fed back into the model?
- What type of threat modeling was performed?
- Was vulnerability scanning performed? What tools were used?
- Was red teaming conducted? Who performed the testing and what were the results?
- How does the application prevent spilling data in the form of output?
- I verify that this application has considered and addressed the OWASP Top 10 for LLMs and Generative AI Apps.

---

6. DoD AI requirements

Explain how your application adheres to the DoD's Ethical Principles for AI:

- **Responsible** – Describe how appropriate judgment and oversight are ensured throughout the AI lifecycle.
- **Equitable** – Identify how your team works to minimize bias in data and models.
- **Traceable** – Explain how development processes, decisions, and data sources are auditable and transparent.
- **Reliable** – Describe your testing and assurance methods to validate AI safety and performance.
- **Governable** – Outline how the system can detect, prevent, and shut down unintended behaviors.

CtF Recommendation Memo

Second Front will upload to this section the risk recommendation memo — signed by a security control assessor or delegated representative — confirming authorization recommendation to the intended Impact Level and risk level determination of the application.

If you're unsure how to complete the BoE, contact your Second Front implementation engineer.

> **BoE for Commercial Deployment environment**
>
> If you're deploying into the Commercial Deployment environment, your BoE **does not require** the following sections:
>
> - Information Security
> - Role Identification
> - CAC Personnel
> - SSDF Attestation
> - CtF Recommendation Memo

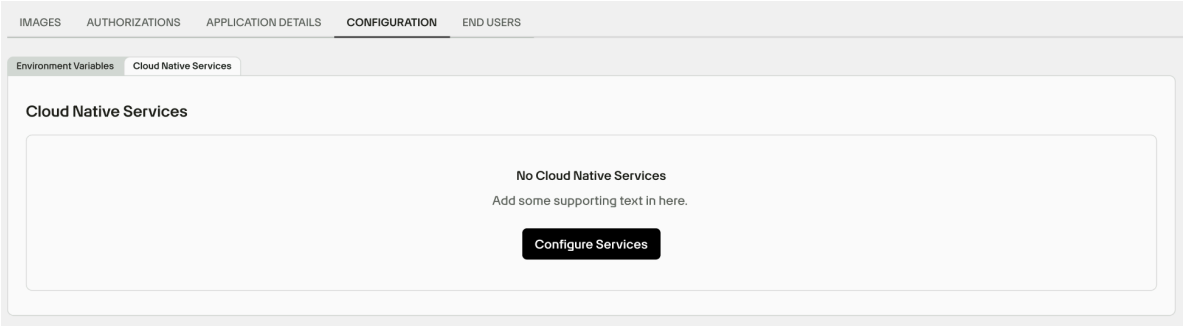
Configure Cloud Native Services

The Cloud Native Service (CNS) feature allows you to provision cloud resources and securely link them to your application using cloud-agnostic terminology. This guide walk you through how to configure cloud resources from the Game Warden app.

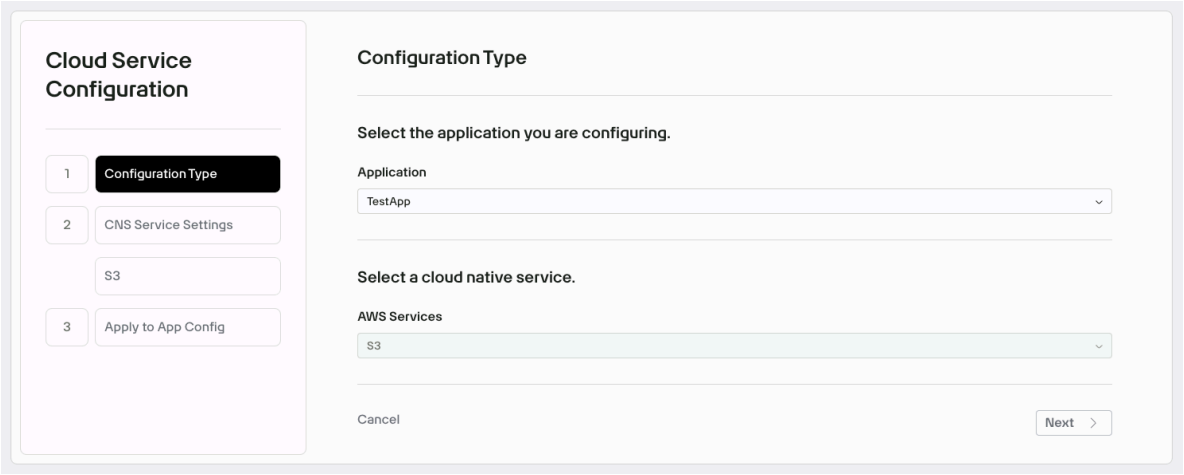
> **Feature availability**
>
> This feature is exclusive to customers on the Runtime deployment architecture. To verify your current architecture or to request a migration, please contact your Mission Success Manager.

Steps to configure a CNS

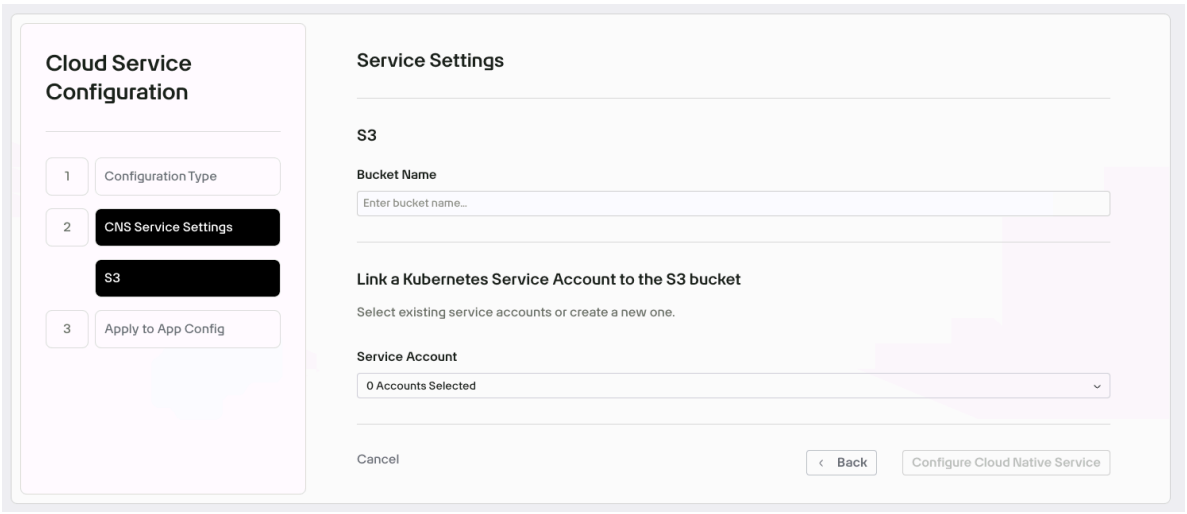
1. Log in to your **Game Warden** account with administrator privileges.
2. From the **App Central** page, select the **Configuration** tab.
3. Click the **Cloud Native Services** sub-tab to view the list of services currently configured for your application.



4. Click **Configure Cloud Native Service** to add a new resource.
5. In the **Configuration Type** section, select the application and the specific service (e.g., S3, ElastiCache, or SQS) from the dropdown menus, then click **Next**.



6. In the **CNS Service Settings** section, enter the bucket name and select or create a Kubernetes service account. Note that selecting a service account creates the **x-pod identity** required to link your Kubernetes (K8s) service account to the cloud resource.



7. Click **Configure Cloud Native Service** to finalize and deploy the configuration.

What happens when you link your S3 bucket to a K8s service account?

When you link a service account to an S3 bucket, Game Warden performs a sophisticated handshake in the background:

- **Export Identity Creation**: The system creates a hidden Export identity role (a specialized Pod Identity).
- **Cryptographic Binding**: This Export identity is strictly bound to your selected K8s service account.
- **Reusability**: Once an Export identity is created for a service account, it can be reused. You can bind the same service account to multiple cloud services (e.g., S3 and an RDS database), and they will all share that secure identity.
- **Final Configuration**: After configuring your services, Game Warden provides a single general editor to manage secrets and values, ensuring your application has the connection strings it needs to communicate with the newly provisioned resources.

By using **service account** as the primary term, Game Warden provides a cloud-agnostic experience. You don't need to be an AWS IAM expert to secure your app—you simply manage the Kubernetes identities you're already familiar with, while Game Warden handles the complex identity mapping (Export roles) required for a secure, ATO-ready environment.

Document Repository for FedRAMP Deployments

The Document Repository is the central location for managing FedRAMP-related artifacts required for Continuous Monitoring (ConMon). You can use it to:

- Upload and retrieve documents needed for ongoing FedRAMP compliance
- Submit any image version for security review
- Export JSON files containing vulnerability findings for any image in your application

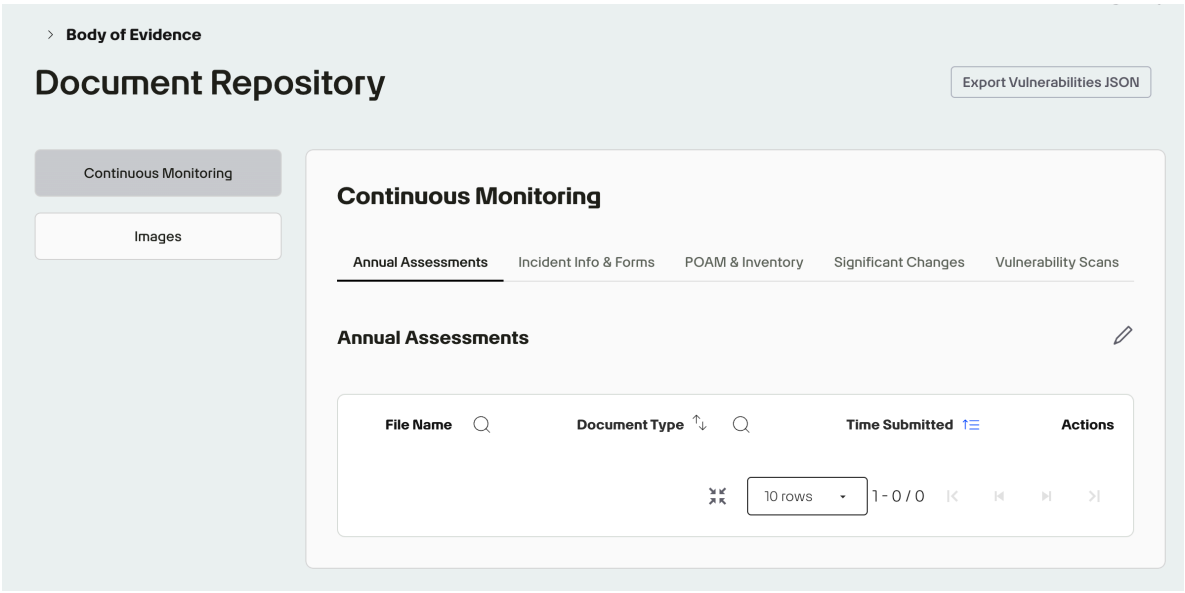
---

Accessing the Document Repository

1. In **App Central**, open your application.
2. Select **Authorizations**, then navigate to the **Document Repository** for the specific deployment.

3. Click **Continuous Monitoring** to access documents organized under the following tabs:
   - **Annual Assessments**: Upload annual security assessments, SAR updates, and required yearly artifacts.
   - **Incident Info & Forms**: Submit incident reports, follow-up documentation, and required forms.
   - **POA&M & Inventory**: Manage Plan of Action & Milestones (POA&M) files and system inventory updates.
   - **Significant Changes**: Document system changes that may impact FedRAMP authorization.
   - **Vulnerability Scans**: Upload scanning artifacts such as…

   To upload a file, click the **pencil** icon, drag and drop your document, and select the appropriate document type from the dropdown.



4. Click **Images** to select the image versions you are submitting for review. Selecting images in the document repository signals which versions will be part of the authorization decision. Game Warden uses this selection to control what can be deployed to Staging (STG) and Production (PRD).

Upload Documents to Game Warden

The Documents page serves as a central repository where both Second Front (2F) and customers can upload, store, and access important files throughout the operation lifecycle.

What you'll find here:

- **Reference Documents** – Files shared with you by Second Front during onboarding and beyond.
- **Application Documents** – Files essential for engineering and operations, such as Helm charts, Dockerfiles, and related resources. Upload these directly to ensure our teams have immediate access to the most up-to-date application and container specifications.
- **Approved CVE Justifications** – Full sets of vulnerability justifications reviewed and approved by the Second Front Security Team.
- **Authorization Artifacts** – Documents such as your Certificate to Field (CtF)/Software Approval or Authorization to Operate (ATO) Letter.
- **Meeting Recordings** – Recordings from relevant meetings, always with your permission.
- **Mission Fit Memorandum for Record** – The signed MFR document.
- **Scan Results** – Security scan outputs and reports for your application and containers.
- **Invoices** – Billing documents for your account.
- **Onboarding Documents** – Files relevant to your application onboarding process.
- **Authorization Boundary Diagrams** – Diagrams defining system boundaries for compliance and accreditation.
- **Static Application Security Testing Reports** – SAST reports generated during the security review process.

Upload a document to Game Warden

Important

The following documents must be uploaded to their designated locations:

- Proof of Authorization – A document such as a CtF/Software Approval or an ATO Letter.
- Mission Fit MFR

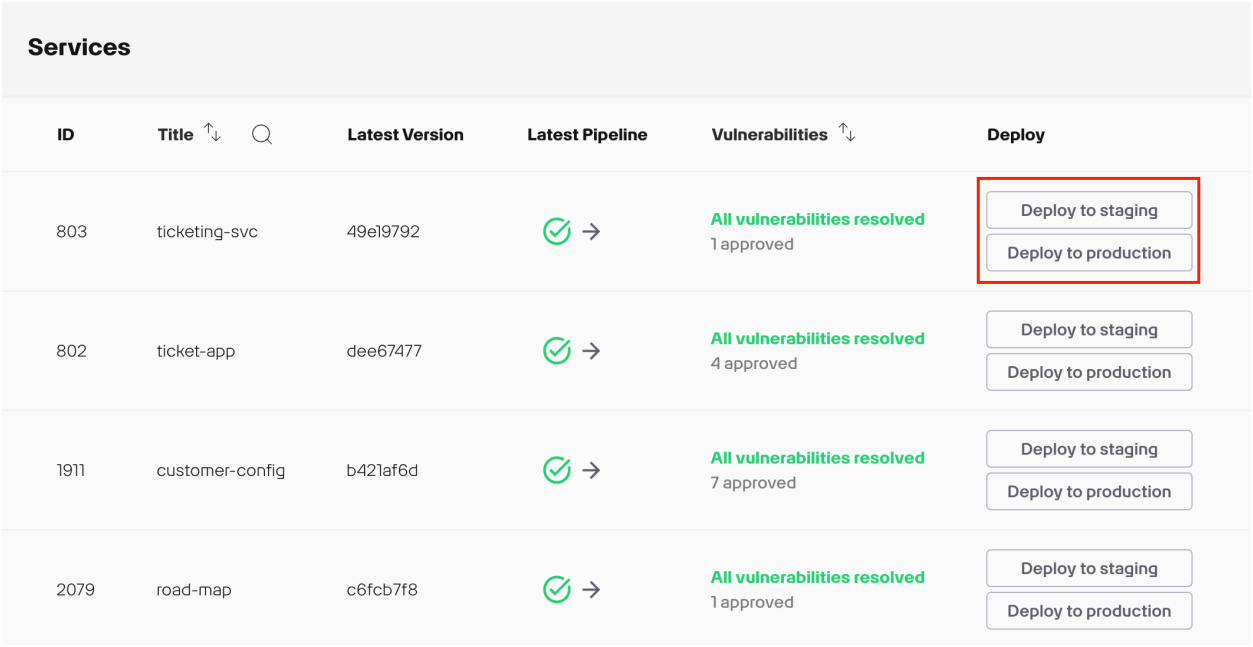
1. Log in to your Game Warden account as an **Admin**.
2. In the left navigation bar, click **Documents** to access the page, then click **New Document**.
3. Choose the file you want to upload from your device.
4. Enter a document name. Optionally, select the application and component it is associated with.
5. Select the appropriate document category.
6. Click **Upload** to finish.

Uploading a Government Contract

When uploading a Government Contract, you may leave the **Application**, **Component**, and **Upload Category** fields blank.

Self-Service Container Deployment

You can deploy approved container images to Staging (STG) and Production (PRD) directly from Game Warden. An image is considered approved when it has no unresolved Common Vulnerabilities and Exposures (CVEs) or when all findings have been remediated or formally risk-accepted with approved justifications.



Warning

This procedure applies to deployments for simple container updates. For deployment requests beyond simple container updates (i.e. new variables, running scripts, etc), submit a support ticket in lieu of this procedure.

Prerequisites

- Only users with the **Customer Admin role** can deploy images to STG and PRD environments.
- For the **Deploy to staging** button to be enabled, your container needs to be run through the Game Warden pipeline to include hardening and scanning. Any security findings surfaced by the Game Warden pipeline scanning tools must be remediated and approved by our security team. Once your container has cleared these gates, this button will turn blue and you can deploy your container to Game Warden's staging environment.
- For the **Deploy to production** button to be enabled, your container must meet the above requirements and be deployed to the staging environment.
- If either deployment button is disabled (gray out), you can hover over the button for clarifying information.

  Important

  If neither deployment button is enabled and you see **N/A**, this means your container has not yet run through the Game Warden pipeline and has no scan results, and therefore cannot be approved for deployment yet. If this indication is unexpected, file a support ticket from the Game Warden app.

Deploying to STG or PRD

1. Start the deployment

Click **Deploy to staging** or **Deploy to production** for the target component.

2. Deployment initializing

A notification confirms the job has started. While it's running, the button is disabled (gray out). Deployments typically take a few minutes. You can hover the gray button to see current status.

3. Check the result

- On **success**, the hover message will indicate completion.
- On **failure**, file a support ticket and include the component name, environment (STG/PRD), and timestamp.

4. Verify your app

Visit your application to validate its health and functionality.

FAQs

How do I deploy my app to additional Impact Levels (ILs)?

After your app is in PRD at a given IL, deploying to another IL requires:

- A new Body of Evidence (BoE) for the target IL
- A new Deployment Passport for the target IL

Once your Deployment Passport is returned with the Authorizing Official's signature, push to STG at the target IL for validation. After successful validation, you may push to PRD.

Do I need to move through lower ILs to reach my target IL?

No. If Game Warden holds the Authority to Operate (ATO) for your target IL, you can deploy directly to that IL's Production—provided you meet all Game Warden requirements.

What's the process to deploy to IL5 if I'm already at IL4?

Game Warden provisions the IL5 environments. You will complete an IL5 BoE and validate app functionality in IL5 STG.

Game Warden will then generate an IL5 Deployment Passport, brief the ISSM, and obtain signature. After approval, we deploy your app to IL5 PRD.

> **Note**
>
> You do not need to deploy to IL4 before IL5; many customers serve end users at different ILs.

If I run at IL4 and also want IL2, do I need separate STG/PRD environments?

Yes. IL2 and IL4 require separate STG and PRD environments to run side-by-side.

Submit Support Tickets

Welcome to our new support experience. We have transitioned our ticketing to Jira to provide better visibility and faster response times for your requests.

Accessing the Support Portal

The most efficient way to access the support portal is through the Game Warden app:

1. Log in to the Game Warden app.
2. Select **Submit a Ticket** from the left navigation bar.

First-time authentication

On your first visit, you'll need to authenticate your account. For the fastest setup and to sync your support history, log in using your environment credentials:

- **AFWERX Keycloak**
- **Commercial Keycloak**
- **FedRAMP Keycloak**

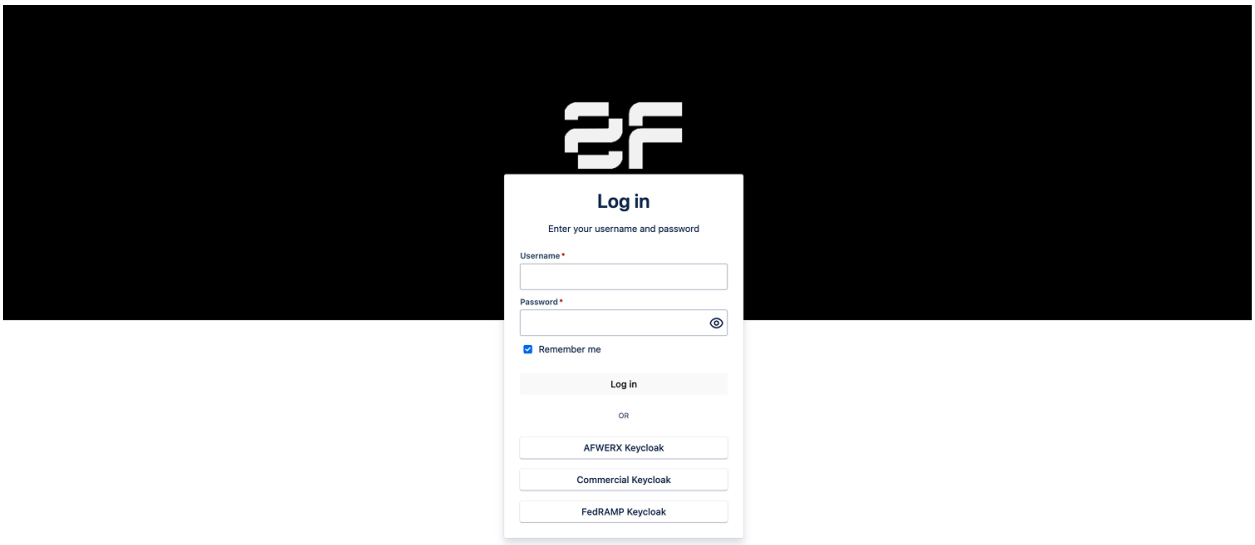


Figure 1: Customer Support Jira

Choosing a Support Category

Once on the support page, select the category that best fits your needs:

- **Troubleshooting Request**: Report errors or unexpected behavior within your deployed application.
- **Resource Provisioning**: Request new environment setups or dedicated resources (e.g., data storage).
- **Pairing Session**: Schedule time with a 2F engineer for technical reviews, setup, or configuration.
- **Bug Report**: Report system-level glitches or features that are not functioning as intended.
- **Security Incident**: Immediately flag security risks or suspected unauthorized leaks of CUI.

- **CtF Renewal**: Initiate the formal renewal process for your Certificate to Field (CtF)/Software Approval authorization.
- **General Inquiry**: Submit questions or requests that do not fit the specific categories above.

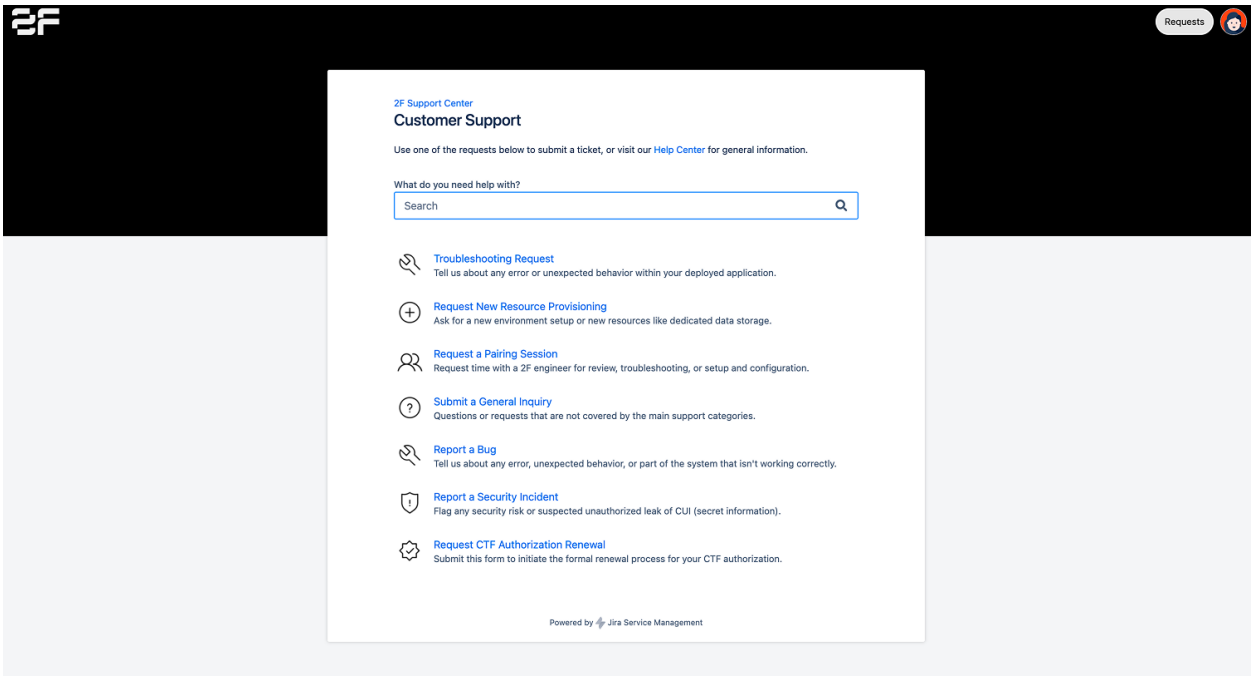


Figure 2: Customer Support Jira

Locating existing tickets

To view your migrated GitLab tickets or check current request status, click **Requests** in the top-right corner. Filter your view by:

- **My requests**: Tickets you created
- **All requests**: All tickets from your organization

Warning

If your issue is time-sensitive, contact your Technical Implementation Manager. They can escalate high-priority tickets and provide direct assistance.

In-App Notifications

Game Warden keeps you informed through real-time in-app notifications. These notifications alert you when key events occur, helping you stay up to date without needing to check multiple places.

You'll receive notifications when:

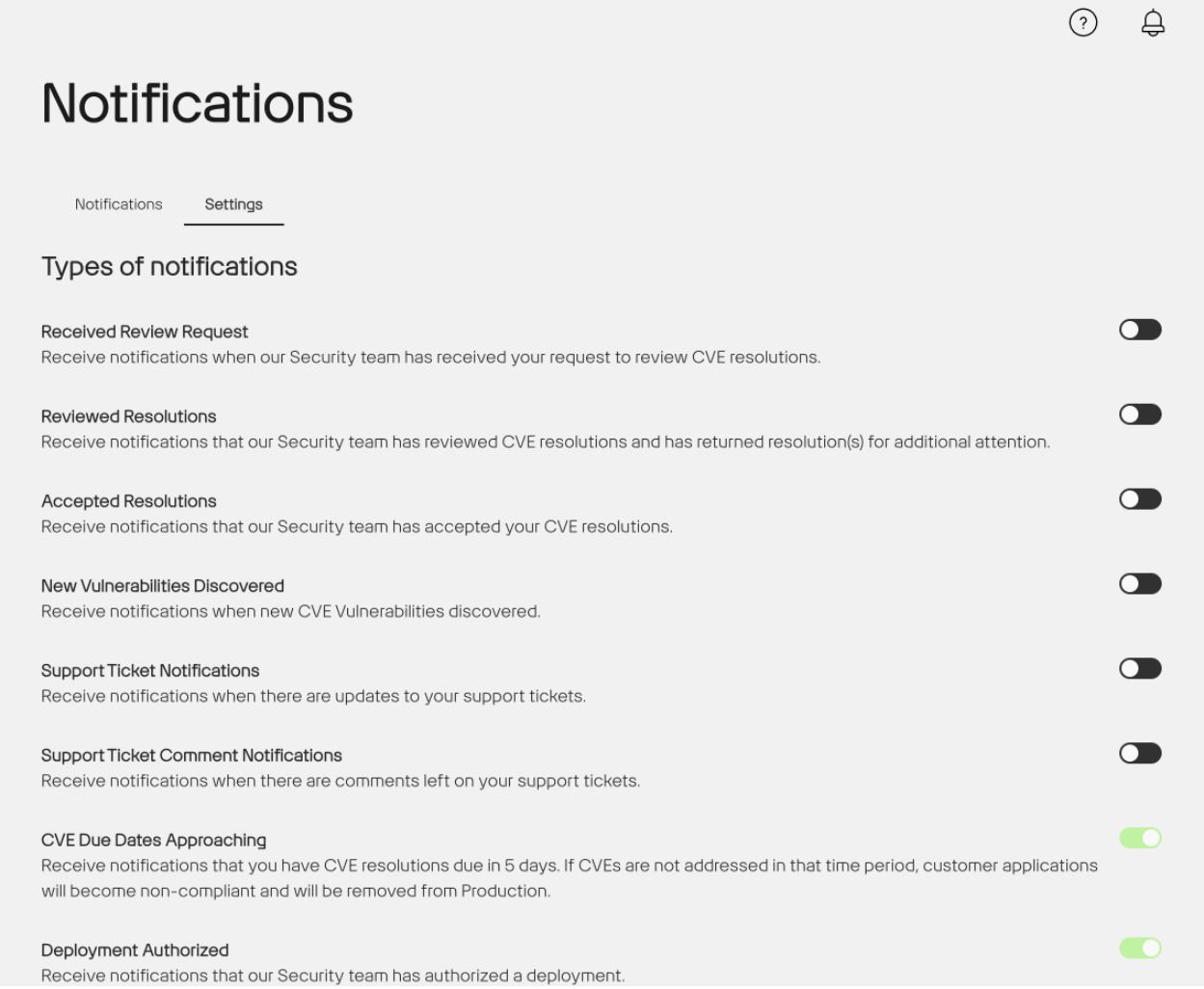
- A pipeline starts after you push a container image
- Scan results become available for your image
- You submit resolutions for security findings
- The Game Warden security team accepts or rejects your submitted resolutions
- A CVE's due date is approaching
- A deployment is authorized

Manage notification settings

Game Warden users with the **Customer Admin** role can manage which notifications are displayed in the app.

To customize your preferences:

1. From the App Central page, click the **bell icon** in the top-right corner to access the **Notifications** page.
2. Select the **Settings** tab.
3. Use the toggles to enable or disable each type of notification according to your needs.

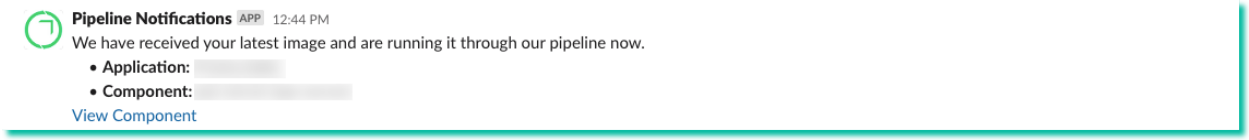

Notifications
Notifications
Settings
Types of notifications
Received Review Request
Receive notifications when our Security team has received your request to review CVE resolutions.
Reviewed Resolutions
Receive notifications that our Security team has reviewed CVE resolutions and has returned resolution(s) for additional attention.
Accepted Resolutions
Receive notifications that our Security team has accepted your CVE resolutions.
New Vulnerabilities Discovered
Receive notifications when new CVE Vulnerabilities discovered.
Support Ticket Notifications
Receive notifications when there are updates to your support tickets.
Support Ticket Comment Notifications
Receive notifications when there are comments left on your support tickets.
CVE Due Dates Approaching
Receive notifications that you have CVE resolutions due in 5 days. If CVEs are not addressed in that time period, customer applications will become non-compliant and will be removed from Production.
Deployment Authorized
Receive notifications that our Security team has authorized a deployment.

Slack Notifications

As part of your onboarding process, Second Front will establish a Slack Connect channel for your team once you've reached the DEV functionality stage. This channel delivers automated notifications to keep your team informed as you move through the onboarding workflow.

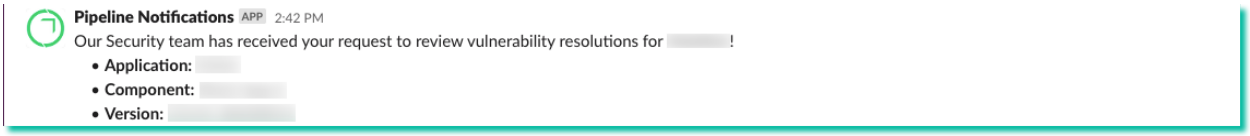
Below are the types of notifications you can expect:

- **Image Push Acknowledgement** - You'll receive this notification each time a new image is successfully pushed to your Harbor project.



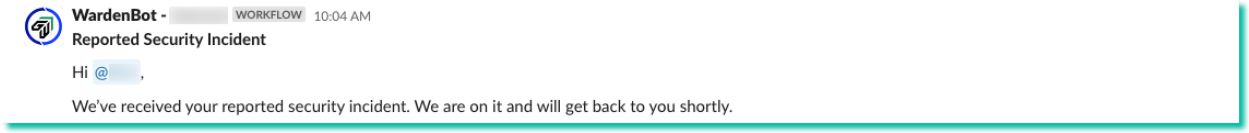
- **Scan Results Available** - Images pushed to Harbor are automatically scanned. You'll be notified when results are ready in Findings.

- **Vulnerability Review Confirmation** - Confirms that your vulnerability justifications have been received and are under review by the Security team.

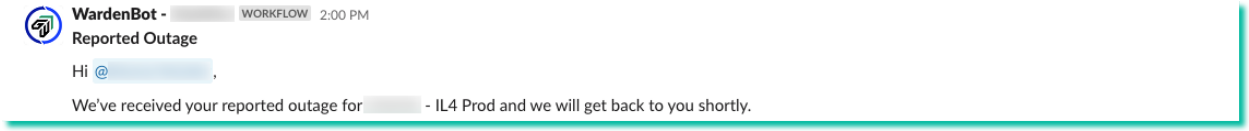


- **Vulnerability Resolutions Accepted** - You'll be notified when your submitted vulnerability resolutions are approved by the Security team.

- **Security Incident** - If you report a potential security issue, this alert confirms our team is investigating and taking action.



- **Application Outage** - Confirms that your reported outage is being reviewed and addressed by our team.



Note

If you're not receiving expected notifications, please reach out in Slack or submit a support ticket so we can assist you.

Manage End Users in Game Warden

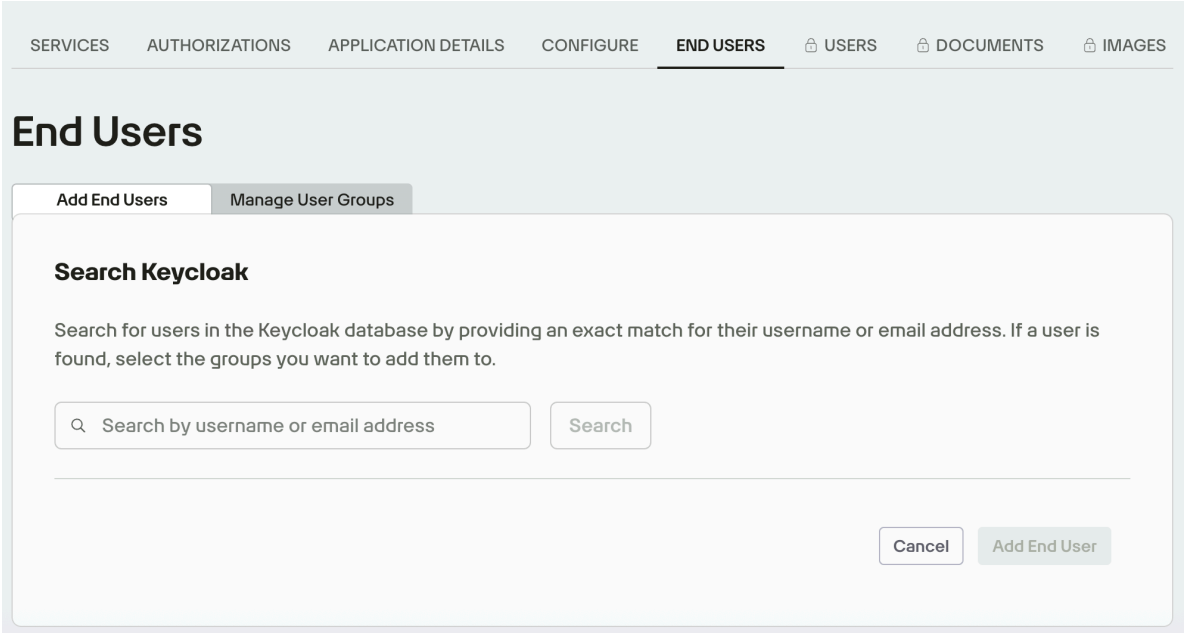
This guide walks you through how to provision and manage access for a new team member in the Game Warden app as an Admin.

Provision users to user groups

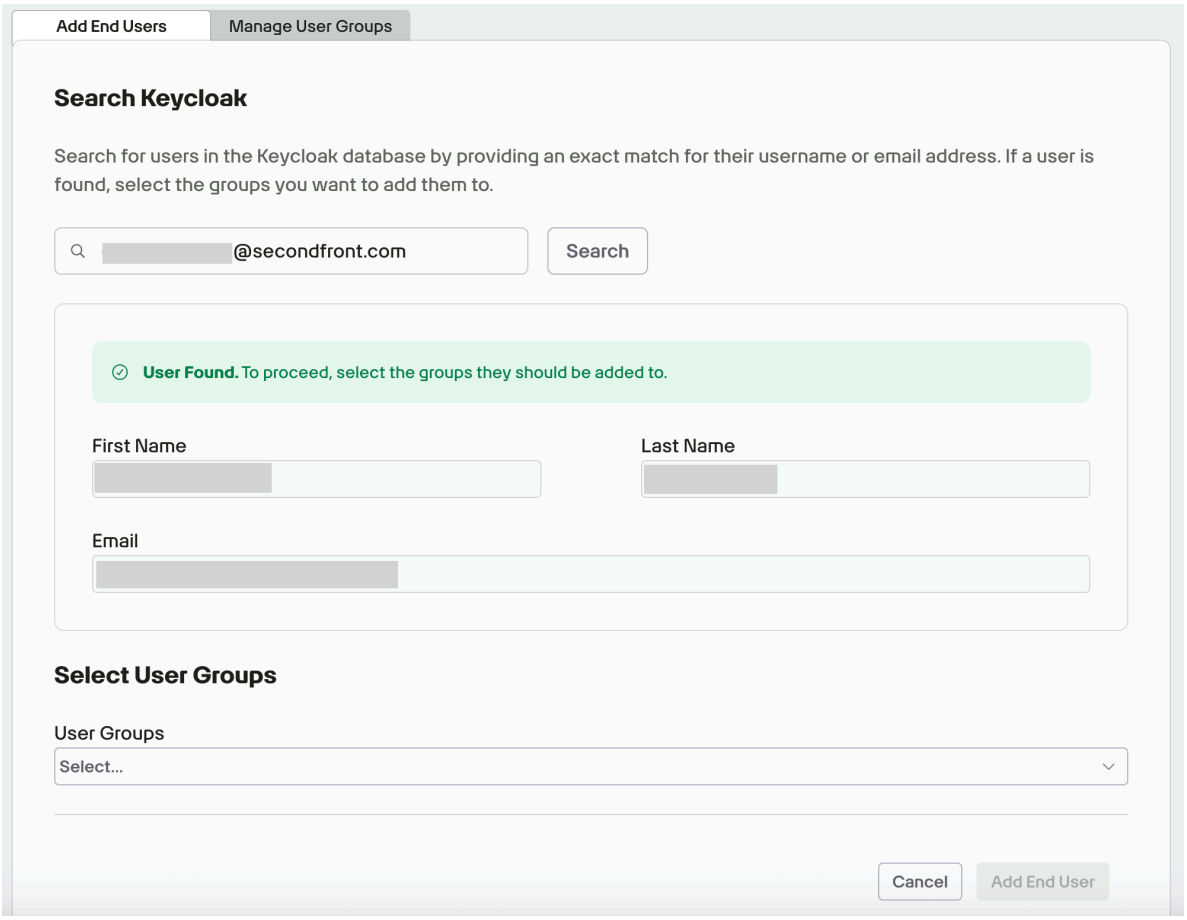
Prerequisite

- Users must first create a Game Warden account and log in at least once before they can be added to any user groups.
- Detailed instructions for setting up a Game Warden account are provided below:
  - For DoD, see Game Warden Account Setup Guide for DoD Deployments.
  - For FedRAMP, see Game Warden Account Setup Guide for FedRAMP Deployments.
  - For Commercial, see Game Warden Account Setup Guide for Commercial Deployments.

1. As an Admin, log in to the Game Warden app and navigate to the **End Users** tab.



2. Under the **Add End Users** tab, search for the new user by email address or username.



3. From the **User Groups** dropdown menu, select the groups the new user should belong to. Each group grants different permissions and access levels based on your company's configuration.
4. Click **Add End User** to complete the process.

Manage user groups

When creating custom user groups in the Game Warden app, Admins can organize users logically (e.g., `data_analysts`, `qa_team`, `read_only_users`), but these groups do not directly control access to services such as Harbor or Grafana within the Game Warden UI. Instead, these custom group assignments are included as attributes in the headers of authenticated network requests.

Downstream services—such as a developed API instance protected by an internal access proxy—can then inspect these headers to determine access. For example, the proxy might read the `groups` header and check whether it contains `developers` or `logging_access`, and grant dashboard editing privileges only to users in the `developers` group, while giving read-only access to those in `logging_access`. This approach gives customers the flexibility to define access control policies directly within their application or middleware, using Game Warden's group assignments as the source of truth.

The following are common tasks related to managing user groups:

Create user group

To create a new user group in Game Warden:

1. Under the **Manage User Groups** tab, click **Create Group**.

2. Enter a name for the new group and click **create**.

Create sub-group

To create a new sub-group:

1. Navigate to the user group, then click the **kebab menu**.
2. Select **Create Subgroup**, and enter a name for the new sub-group.
3. Click **create**.

Remove users from group/sub-group

To remove a user from a group or sub-group:

1. Navigate to the desired group or sub-group, then click the **kebab menu**.
2. Select **View Users** to display the list of users in that group or sub-group.
3. To remove a user, click the **X** icon next to their email address.

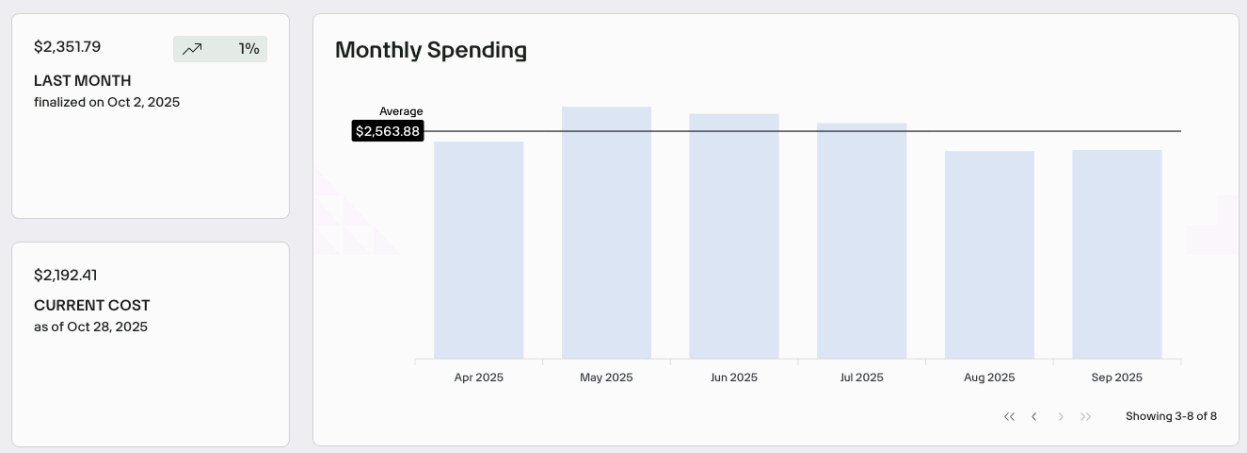
Cloud Cost

The Cloud Cost page provides a centralized view of your cloud usage and cost information within the Game Warden app. To access it, go to **Company Profile → Cloud Cost**.

> **Important**
>
> This feature is available for the Commercial and DoD environments only. FedRAMP support is in progress.

What you can do on the Cloud Cost page:

- Access and download detailed cloud cost statements directly from the app
- Compare usage and cost trends across months to understand drivers and spot changes in resource consumption
- See ongoing spend for the current month, refreshed daily, to catch anomalies early and avoid surprises



Meet Klowd: Your Game Warden AI Assistant

Klowd is an AI assistant built directly into Game Warden to help you work faster and smarter. Ask questions in plain language and get instant answers about security compliance, platform features, troubleshooting steps, and technical procedures.

Klowd searches the Game Warden Help Center, Second Front's public website, and core documentation to ground every response in accurate, current information. You'll receive clear guidance with relevant context and links to source materials, so you can quickly find what you need and get back to work.

Why use Klowd

Navigating complex dev-sec-ops environments can be challenging. Klowd acts as your 24/7 technical partner, providing:

- **Instant guidance**: Get immediate answers to "how-to" questions without searching through documentation.
- **Efficiency**: Reduce the time spent troubleshooting by letting Klowd summarize requirements and processes.

  Coming soon

  - **Contextual support**: Klowd understands the specific environment of Game Warden, offering relevant advice for your deployment.
  - **Vulnerability research**: Use Klowd to research and plan mitigations for security vulnerabilities that have been identified in you application.

How to access Klowd

To access Klowd, click the **question mark** button located on the upper right of the Game Warden web app.

Example prompts

Below are example prompts you can try using Klowd:

Getting Started & Onboarding

- "What is the complete onboarding process from start to production?"
- "How long does it typically take to deploy to Game Warden?"
- "What is the fastest path from development to production?"
- "Should I deploy to Commercial first or go straight to DoD?"
- "What resources does Second Front provide during onboarding?"
- "How do I know if my application is ready for Game Warden?"

Authorization & Compliance

- "What is the difference between ATO, CtF, and Deployment Passport?"
- "Do I need my own ATO or can I inherit Game Warden's?"
- "How long is a Certificate to Field (CtF) valid and how do I renew it?"
- "What triggers the need for a new Deployment Passport?"
- "What's an ISSM and why do I need their signature?"
- "Do I need separate authorization for each Impact Level?"
- "What security documentation do I need to provide for my app?"

Impact Levels & Environments

- "What is the difference between IL2, IL4, IL5, and IL6?"
- "Which Impact Level do I need for my specific data type?"
- "Does Game Warden support IL6 or classified deployments?"
- "What's the difference between NIPRNet and SIPRNet access?"
- "Can I deploy the same application to multiple Impact Levels?"
- "What environments are available (DEV, Staging, Production)?"

Access Control & Authentication

- "How do I set up my CAC card for authentication on macOS or Windows?"
- "What is Platform One SSO and how do I log in?"
- "Do end users need CAC cards to access my application?"
- "How do I configure OIDC or use my organization's existing SSO?"
- "How do I install and troubleshoot AppGate SDP?"
- "What is CNAP and how do I get my IP whitelisted?"
- "How do I implement custom authorization logic in my app?"

Architecture & Technical Requirements

- "What are the entrance requirements and containerization needs?"
- "Can I use a monolithic architecture or must it be microservices?"
- "How do I connect to third-party APIs or external databases?"
- "Do I need to use Iron Bank base images? What are they?"
- "How do I handle secrets and environment variables safely?"
- "What are the logging and health check requirements?"

Data & Database Management

- "What managed database solutions does Game Warden support?"
- "How do I populate my database or perform data migrations?"
- "How is data encrypted at rest and in transit?"
- "Can I access my database directly for troubleshooting?"
- "How do I perform and manage database backups?"

CI/CD, GitLab & Harbor

- "What does the CI/CD pipeline process look like and how long does it take?"
- "What is container hardening and why does Game Warden need my Dockerfiles?"
- "How do I push images to the Harbor registry and what do the tags mean?"
- "Can I run the pipeline or scans locally before pushing?"
- "How do I access my pipeline artifacts?"

Security & CVE Management

- "How do I view and remediate CVEs for my application?"
- "What is the difference between CVE remediation and justification?"
- "How long do I have to fix critical vulnerabilities before losing access?"
- "What is the Acceptance Baseline Criteria (ABC)?"
- "What is an SBOM?"
- "How do I generate SAST artifacts?"

Monitoring & Operations

- "How do I access application logs using Loki?"

- "What is Grafana and how do I create custom dashboards?"
- "How do I set up alerts and monitor performance?"
- "What are my 'Day 2' responsibilities after going live?"
- "What constitutes a 'significant software change' requiring re-authorization?"
- "How do I handle security incidents or test disaster recovery?"

Support & Troubleshooting

- "How do I submit a support ticket or request a pairing session?"
- "What is the expected response time for helpdesk tickets?"
- "My application is down in production—what is the escalation procedure?"

Klowd is evolving—share your feedback

Got a thought? If Klowd gives you a great answer (or a confusing one), please use the thumbs up/down icons in the chat. We are building Klowd for you, so let us know how we can make it better!

Implementation Kickoff Guide for AFWERX Deployment

This guide outlines the implementation process and shared responsibilities between your team and Second Front's Game Warden platform. It includes what's expected at each phase of onboarding, what artifacts you need to provide, and how we'll work together to securely deploy your application to Game Warden.

Implementation phases & milestones

Kickoff

- **What Second Front Does:**
  - Assign Mission Success Manager (MSM) and Technical Implementation Manager (TIM)
  - Deliver onboarding resources and guidance
- **What You Do:**
  - Assign technical and security leads
  - Review onboarding requirements
  - Obtain a signed MFR from their Government Mission Owner
  - Begin collecting technical artifacts

Configuration

- **What Second Front Does:**
  - Set up Development (DEV) environment
  - Configure Helm charts and pipelines
- **What You Do:**
  - Provide images and initial documentation
  - Complete access control setup with Platform One (P1) SSO, Keycloak, and Government Access Card (required for deployments to IL4+ environments)

Security review

- **What Second Front Does:**
  - Perform container scans (DAST)
  - Support Body of Evidence (BoE) documentation
- **What You Do:**
  - Perform container scans (SAST) and submit results to Second Front
  - Justify or remediate CVEs
  - Complete the BoE and Authorization Boundary Diagram

Approve

- **What Second Front Does:**
  - Submit Deployment Passport for Authorizing Official (AO) signature
  - Prepare Staging (STG) environments
- **What You Do:**
  - No action required

Validation

- **What Second Front Does:**
  - Deploy to STG environment
  - Assist with test cases and validation checks
- **What You Do:**
  - Perform final functional tests in STG

- Confirm production readiness

Deployed

- **What Second Front Does:**
  - Push to Production (PRD) environment
  - Monitor initial usage and performance
- **What You Do:**
  - Confirm PRD readiness
  - Begin end-user onboarding

Deployed to Classified Networks

- **What Second Front Does:**
  - Collaborate with customers to obtain the necessary documents for intake by the Department of Air Force CloudWorks (DAFCW). This process can start after submitting the Deployment Passport.
- **What You Do:**
  - Provide necessary documents required by DAFCW

Day 2 Operations

- **What Second Front Does:**
  - Provide ongoing support
  - Deliver quarterly business reviews and growth planning
- **What You Do:**
  - Join regular sync meetings
  - Review logs and scan results
  - Plan for roadmap milestones

Shared Responsibility Model

Second Front uses the Shared Responsibility Model to clarify which tasks are owned by the customer and which are managed by Second Front, helping streamline implementation and compliance. Refer to Game Warden's Shared Responsibility Model for more information.

Required technical artifacts

To support a smooth and secure implementation, please collect and submit the required items listed in the Technical Artifacts guide as early as possible in the process.

Security requirements

Security is core to the Game Warden platform. You must be prepared to support the following:

Artifact	Guidance
Authorization Boundary Diagram	Required for BoE and initial implementation planning. Must include all outbound integrations or API dependencies to initiate Approval to Connect (AtC) workflows early.

Artifact	Guidance
CVEs & Remediation	You'll use Findings to manage container scans. CVEs must be addressed per the Acceptance Baseline Criteria.
BoE	Gather required details early to avoid bottlenecks.
SAST Scan & AI Attestation	Prepare static analysis outputs and AI-related disclosures, if applicable.

Technical considerations

To ensure a smooth deployment process, your team should review:

- **Access Control** - Integrate with Keycloak and implement JWT authentication

  → Access Control Guide

- **CNAP Whitelisting** - Required for IL4+ environments

  → CNAP Whitelist

- **Logging & Monitoring** - Game Warden integrates with Loki and Grafana

  → Observability Stack

- **Pipelines & Image Push** - Push hardened images to the Harbor Registry

  → Push Images to Harbor

Action items & best practices

- Assign a dedicated technical and security lead
- Schedule weekly/bi-weekly syncs with your Game Warden team
- Set up P1 accounts for all relevant team members
- Review observability tools and security scanning expectations
- Begin preparing your BoE, authorization boundary, and technical artifacts
- Push initial image to Game Warden registry as early as possible

  Helpful links

  - Platform One Account Setup
  - App Central
  - Findings
  - Body of Evidence Guidance
  - Harbor Registry Instructions

Implementation Kickoff Guide for Commercial Deployment

This guide outlines the implementation process and shared responsibilities between your team and Second Front's Game Warden platform. It includes what's expected at each phase of onboarding, what artifacts you need to provide, and how we'll work together to securely deploy your application to Game Warden.

Implementation phases & milestones

Kickoff

- **What Second Front Does:**
  - Assign Mission Success Manager (MSM) and Technical Implementation Manager (TIM)
  - Deliver onboarding resources and guidance
- **What You Do:**
  - Assign technical and security leads
  - Review onboarding requirements
  - Begin collecting technical artifacts

Configuration

- **What Second Front Does:**
  - Set up Development (DEV) environment
  - Configure Helm charts and pipelines
- **What You Do:**
  - Provide images and initial documentation
  - Complete access control integration (e.g., Keycloak)

Security review

- **What Second Front Does:**
  - Perform container scans (DAST)
  - Support Body of Evidence (BoE) documentation
- **What You Do:**
  - Perform container scans (SAST) and submit results to Second Front
  - Justify or remediate CVEs
  - Complete the BoE and Authorization Boundary Diagram

Approve

- **What Second Front Does:**
  - Prepare Staging (STG) environments
- **What You Do:**
  - No action required

Validation

- **What Second Front Does:**
  - Deploy to STG environment
  - Assist with test cases and validation checks
- **What You Do:**
  - Perform final functional tests in STG
  - Confirm production readiness

Deployed

- **What Second Front Does:**
  - Push to Production (PRD) environment
  - Monitor initial usage and performance
- **What You Do:**
  - Confirm PRD readiness
  - Begin end-user onboarding

Day 2 Operations

- **What Second Front Does:**
  - Provide ongoing support
  - Deliver quarterly business reviews and growth planning
- **What You Do:**
  - Join regular sync meetings
  - Review logs and scan results
  - Plan for roadmap milestones

Shared Responsibility model

Second Front uses the Shared Responsibility Model to clarify which tasks are owned by the customer and which are managed by Second Front, helping streamline implementation and compliance. Refer to Game Warden's Shared Responsibility Model for more information.

Required technical artifacts

To support a smooth and secure implementation, please collect and submit the required items listed in the Technical Artifacts guide as early as possible in the process.

Security requirements

Security is core to the Game Warden platform. You must be prepared to support the following:

Artifact	Description
Authorization Boundary Diagram	Required for BoE and initial implementation planning. Must include all outbound integrations or API dependencies.
CVEs & Remediation	You'll use Findings to manage container scans. CVEs must be addressed per the Acceptance Baseline Criteria.
BoE	Gather required details early to avoid bottlenecks.
SAST Scan & AI Attestation	Prepare static analysis outputs and AI-related disclosures, if applicable.

Technical considerations

To ensure a smooth deployment process, your team should review:

- **Access Control** - Integrate with Keycloak and implement JWT authentication

  → Access Control Guide
- **Logging & Monitoring** - Game Warden integrates with Loki and Grafana

  → Observability Stack
- **Pipelines & Image Push** - Push hardened images to the Harbor Registry

  → Push Images to Harbor

Action items & best practices

- Assign a dedicated technical and security lead
- Schedule weekly/bi-weekly syncs with your Game Warden team
- Review observability tools and security scanning expectations
- Begin preparing your BoE, authorization boundary, and technical artifacts
- Push initial image to Game Warden registry as early as possible

  Helpful links

  - App Central
  - Findings
  - Body of Evidence Guidance
  - Harbor Registry Instructions

Implementation Kickoff Guide for FedRAMP Deployment

This guide outlines the FedRAMP implementation process and shared responsibilities between your team and Second Front's Game Warden platform. It includes what's expected at each phase of onboarding, what artifacts you need to provide, and how we'll work together to securely deploy your application and obtain FedRAMP Authorization to Operate (ATO).

Implementation phases & milestones

Pre-engagement alignment

- **What Second Front Does:**
  - Validate FedRAMP alignment and technical feasibility
- **What You Do:**
  - Identify a U.S. Federal Agency Sponsor
  - Confirm FedRAMP authorization intent
  - Complete the FedRAMP Technical Intake Form

Kickoff

- **What Second Front Does:**
  - Assign Mission Success Manager (MSM) and Technical Implementation Manager (TIM)
  - Deliver FedRAMP onboarding guidance
- **What You Do:**
  - Assign technical, compliance, and security leads
  - Review FedRAMP onboarding requirements
  - Begin collecting technical artifacts

Configuration

- **What Second Front Does:**
  - Configure development environments
  - Set up pipelines, Helm charts, and platform integrations
- **What You Do:**
  - Provide container images
  - Configure identity and access management
  - Begin platform integration activities

Gap assessment & documentation development

- **What Second Front Does:**
  - Coordinate Security Advisory Services and Agency engagement
  - Support control gap identification
- **What You Do:**
  - Participate in control gap analysis
  - Support remediation planning
  - Develop required FedRAMP documentation, including:
    - Body of Evidence (BoE)
    - Control implementation statements
    - Policies and procedures
    - Asset inventory
    - Plans of Action & Milestones (POA&Ms)

Security review

- **What Second Front Does:**
  - Perform platform-level security assessment
  - Review:
    - Dynamic Application Security Testing (DAST)
    - Container scan results
- **What You Do:**
  - Resolve vulnerabilities found in your application
  - Provide justifications for findings as needed
  - Close **Critical and High vulnerabilities**
  - Submit application for security review

Approve

- **What Second Front Does:**
  - Approve readiness for production deployment
- **What You Do:**
  - No action required

Deploy to Production

- **What Second Front Does:**
  - Enable production deployment
- **What You Do:**
  - Deploy application to Production environment through the Game Warden platform

Third Party Assessment (3PAO)

- **What Second Front Does:**
  - Coordinate 3PAO engagement
- **What You Do:**
  - Participate in 3PAO assessment activities
  - Support interviews and evidence review
  - Contribute to development of the Security Assessment Report (SAR)

  Important

  Customers are responsible for conducting Static Application Security Testing (SAST) and submitting the results to the 3PAO for review.

Authorization

- **What Second Front Does:**
  - Submit the authorization package to the sponsoring agency
  - Coordinate FedRAMP PMO review
- **What You Do:**
  - Support agency review and clarification requests. If approved, your application will be listed as "Authorized" on the FedRAMP Marketplace

Day 2 Operations & Continuous Monitoring (ConMon)

- **What Second Front Does:**
  - Monthly platform scans:
    - Operating system scans
    - Database scans
    - Web application scans

        * Container scans
        * Configuration scans
    - Monthly platform inventory submission to FedRAMP
    - Submission of customer scan artifacts to FedRAMP
    - POA&M submission
    - ConMon Executive Summary submission
- **What You Do:**
    - Monthly container vulnerability scans
    - Application container inventory
    - POA&M development and updates
    - Support continuous monitoring reporting

Shared Responsibility Model

Second Front uses the Shared Responsibility Model to clarify which tasks are owned by the customer and which are managed by Second Front, helping streamline implementation and compliance. Refer to Game Warden's Shared Responsibility Model for more information.

Required technical artifacts

To support a smooth and secure implementation, please collect and submit the required items listed in the Technical Artifacts guide as early as possible in the process.

Security requirements

Security is core to the Game Warden platform. You must be prepared to support the following:

Artifact	Description
Authorization Boundary Diagram	Visual representation of your system's FedRAMP boundary, showing what's included in the authorization scope and how external systems integrate.
CVEs & Remediation	You'll use Findings to manage container scans. CVEs must be addressed per the Acceptance Baseline Criteria.
BoE	Complete FedRAMP authorization package containing all security documentation, control implementations, test results, and supporting evidence required for ATO.
Control Implementations	Detailed documentation demonstrating how your system implements each required NIST SP 800-53 security control with evidence of compliance.
SAST Scan & AI Attestation	Prepare static analysis outputs and AI-related disclosures, if applicable.

Technical considerations

- **Access Control**: Identity federation, least privilege, and strong authentication
- **Logging & Monitoring**: Centralized logging, audit trails, and monitoring
- **Pipelines & Image Push**: Secure CI/CD pipelines, signed images, and controlled promotion
- **Configuration Management**: Change control and configuration baselines

---

Action items & best practices

- Secure a Federal Agency Sponsor early
- Assign clear compliance ownership
- Start system security control planning and BoE development immediately
- Resolve vulnerabilities early in the process
- Align application architecture with FedRAMP authorization boundaries
- Treat the 3PAO assessment as a formal audit
- Establish ConMon workflows before authorization

Game Warden's Shared Responsibility Model

The Game Warden Shared Responsibility Model outlines the division of responsibilities between Second Front (2F) and our customers across key areas of the platform, including infrastructure, security, development, and accreditation.

This model benefits customers by:

- Clarifying ownership of operational and compliance tasks, reducing confusion during implementation and ongoing operations.
- Streamlining collaboration, ensuring that both Second Front and customers can focus on their core responsibilities.
- Supporting faster authorization, by identifying who is accountable for delivering required artifacts and security controls.

The sections below outline specific areas of responsibility across key deployment types:

DoD Deployment

- **Infrastructure**
  - **Responsibility Area**: Compute, storage, database, networking
  - **Responsibility**: 2F
- **Platform**
  - **Responsibility Area**: CI/CD, container & runtime scans, deployment environments, identity & access management
  - **Responsibility**: 2F
- **Security**
  - **Responsibility Area**: Continuous monitoring, 24/7 incident response, penetration testing, DoD compliance
  - **Responsibility**: 2F
  - **Responsibility Area**: Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST)
  - **Responsibility**: Shared
- **Development**
  - **Responsibility Area**: App development, container builds, client-side data, encryption & data integrity, data seeding / migration, vulnerability remediation
  - **Responsibility**: Customer
- **Accreditation**
  - **Responsibility Area**: CtF/Software Approval package preparation and submission
  - **Responsibility**: 2F
  - **Responsibility Area**: Deployment manifests & container artifacts
  - **Responsibility**: Shared
  - **Responsibility Area**: CtF/Software Approval supporting documentation
  - **Responsibility**: Customer

FedRAMP Deployment

The journey to FedRAMP authorization and deployment with 2F progresses through these phases:

Development → Infrastructure → Platform → Security → 3PAO → Government Accreditation

The sections below outline the roles and responsibilities in the collaboration model:

Development (Customer)

- App development
- Container builds

- Client-side data, encryption & data integrity
- Data seeding and/or migration
- Vulnerability remediation

Infrastructure (2F)

- Compute
- Storage
- Database
- Networking

Platform (Customer & 2F)

- CI/CD
- Container & runtime scans
- Deployment environments
- Identity & access management (IAM) (Customer)

Security (2F)

- Continuous monitoring
- 24/7 incident response
- SAST & DAST
- Penetration testing (customer's application)
- Compliance readiness:
  - Gap assessment and documentation support, including preparation of the Body of Evidence (BoE), Security Assessment Plan (SAP), and Risk Assessment Report (RAR).
  - Evidence gathering (screenshots, audit logs)
  - Plan of Action & Milestones (POA&M)
  - Ongoing continuous-monitoring guidance & updates

3PAO (Customer)

> **Important**
>
> Customers engage directly with your 3PAO (Third-Party Assessment Organization) and are billed by the 3PAO. You may leverage 2F's negotiated discounted rates when selecting a participating 3PAO.

Work with your selected 3PAO to produce the Security Assessment Report (SAR):

- Risk assessments (OS, web, database, container, and other scans)
- Security controls testing
- FedRAMP Initial Authorization Checklist
- Penetration testing (customer's web application)
- Security remediation & mitigation plan

Government Accreditation (Government)

The Agency Sponsor or FedRAMP Program Management Office (PMO) conducts SAR review and issues the ATO:

- Sponsoring Authorizing Official reviews the complete authorization package (BoE, SAP, SAR)
- If the assessment passes, the Agency Sponsor issues the **ATO**

Commercial Deployment

- **Infrastructure**
  - **Responsibility Area**: Compute, storage, database, networking
  - **Responsibility**: 2F
- **Platform**
  - **Responsibility Area**: CI/CD, container & runtime scans, deployment environments, identity & access management
  - **Responsibility**: 2F
- **Security**
  - **Responsibility Area**: Continuous monitoring, 24/7 incident response, penetration testing, DoD compliance
  - **Responsibility**: 2F
  - **Responsibility Area**: Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST)
  - **Responsibility**: Shared
- **Development**
  - **Responsibility Area**: App development, container builds, client-side data, encryption & data integrity, data seeding / migration, vulnerability remediation
  - **Responsibility**: Customer

Technical Artifacts

To ensure a smooth onboarding and authorization process, Game Warden requires a set of technical artifacts that describe your application's architecture, configuration, and operational requirements. These materials help the Second Front (2F) team understand how your system functions and how it should be deployed securely.

Note

- Submit diagrams and visuals in `.jpeg`, `.png`, or `.pdf` formats.
- Submit Dockerfiles in `.txt` format.

Required artifacts

Please prepare and submit the following:

1. Authorization boundary diagram
2. Product networking requirements
3. Third-party software and services
4. Application database requirements
5. List of running containers
6. Dockerfiles (`.txt` format)
7. Environment variable keys / secrets overview
8. Helm charts or Docker Compose files (Helm preferred)

Info

Learn more about Supported Design Patterns and Architectures.

---

1. Authorization boundary diagram

Provide a visual diagram of your product's architecture, clearly outlining all components that fall within and outside the Game Warden boundary. This includes internal services, external integrations, and system dependencies.

Your diagram **must represent**:

- Networking flows (ingress, egress, bidirectional traffic)
- Third-party software and services
- Databases and storage locations

For more guidance, see Authorization Boundary Diagrams.

---

2. Product networking requirements

List all network communication your product requires. Include:

- Ingress, egress, and bidirectional data flows
- IP addresses, ports, and protocols
- Annotations to clarify which flows apply to which services (if you have multiple apps)

Example:

Inbound	Outbound
`8.8.8.8:443` – datasource for XYZ.	`103.92.44.2:3306` – MySQL DB for XYZ

These data flows must also be visually represented in your Authorization Boundary Diagram.

---

3. Third-party software and services

List any external tools, APIs, services, or platforms your application integrates with—especially those not embedded directly in your codebase.

Example:

- Application X consumes RabbitMQ traffic from a DISA IL4 environment
- Application X connects to GCCS-J at Scott AFB
- Application Y processes S3 uploads via Lambda trigger

These external services must be shown in your Authorization Boundary Diagram to accurately depict the system boundary.

---

4. Application database requirements

Describe your database usage and deployment needs:

- Type of database (e.g., MySQL, PostgreSQL, SQL-compatible)
- Where it's hosted (internal container, managed cloud service, etc.)
- Data encryption, retention, or replication concerns

Example:

App X uses a containerized MySQL database that stores user profiles. Data is encrypted at rest and compatible with any SQL backend.

Your database must also be represented in your Authorization Boundary Diagram.

---

5. Running containers

List all containers your application uses in production. This helps Game Warden validate your deployment structure and track dependencies during onboarding and scanning.

What to include:

- Each container name and associated image
- Deployment method (e.g., Kubernetes or Docker Swarm)
- Exposed ports

Example:

Name	Mode	Replicas	Image	Ports
`my_app`	Replicated	1/1	`my_app:latest`	*:443 → 443
`my_app_postgres`	Replicated	1/1	`my_app_postgres:latest`	*:8100 → 8100
`rabbitmq`	Replicated	1/1	`docker.io/rabbitmq:latest`	*:5671 → 5671

These containers should also be reflected in your Authorization Boundary Diagram to ensure a complete view of deployed components.

---

6. Dockerfiles

Submit all Dockerfiles in `.txt` format. These help our team proactively review application dependencies, identify embedded software, and ensure adherence to best practices.

---

7. Environment variable keys and secrets

List all **environment variable keys** used by your application, along with their **purpose or function**. Do **not** include actual secrets or passwords.

What to include:

- Variable name (the "key")
- Description of what it's used for

Example:

`mysql_db_password`: Authenticates DB access from the application

> **Important**
>
> Submit only keys and descriptions. Do not submit the actual values. Game Warden is not requesting secrets, only the structure of what your application expects.

---

8. Helm charts or Docker Compose files

Game Warden prefers **Helm charts** and uses an internal tool (Helminator) to generate required deployment artifacts. While Docker Compose files are accepted initially, customers are expected to transition to Helm during onboarding.

What to include:

- Complete Helm chart or Compose file
- Definitions of services, volumes, dependencies, ports, and protocols

Helm inputs should align with the structure and flow shown in your Authorization Boundary Diagram.

> **Note**
>
> If you're not already using Helm, familiarize yourself with Kubernetes-based deployments prior to onboarding.

Harbor Overview

Game Warden uses Harbor as its secure container image registry. Harbor is an open-source, feature-rich solution that manages all images pushed into the platform. As part of the Game Warden CI/CD process, Harbor integrates with tools such as:

- **Anchore Enterprise** and **Checkmarx supported Zed Attack Proxy** for vulnerability scanning and compliance
- **ClamAV** for malware detection

As your images move through the scanning and hardening process, Harbor automatically appends status tags to the end of each image name.

This guide focuses on manually pushing images to Harbor via Docker commands in a terminal. If you're using automated pipelines, reach out to your Technical Implementation Manager for robot credentials that prevent session expiration and streamline the process.

> **Note**
>
> If you're deploying unmodified Iron Bank images, the Game Warden team will pull them directly into the platform for you.

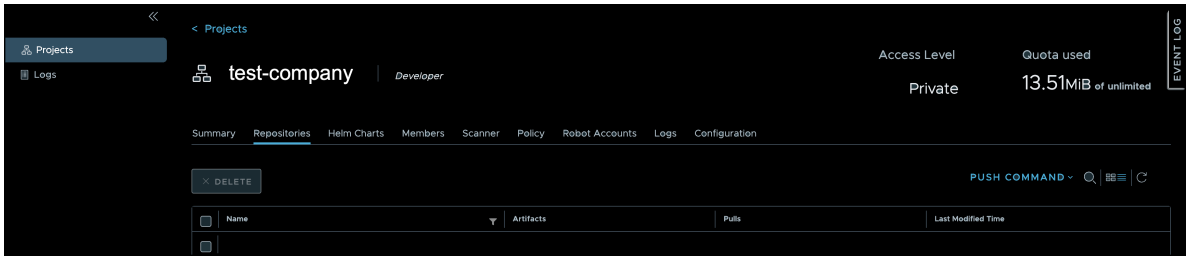
Prerequisites

- You have a Platform One (P1) Single Sign-On (SSO) account.
- You've been granted permission by Game Warden to push images on behalf of your company.
- Your company has submitted a list of authorized users (username + email) to Game Warden.

Access Harbor

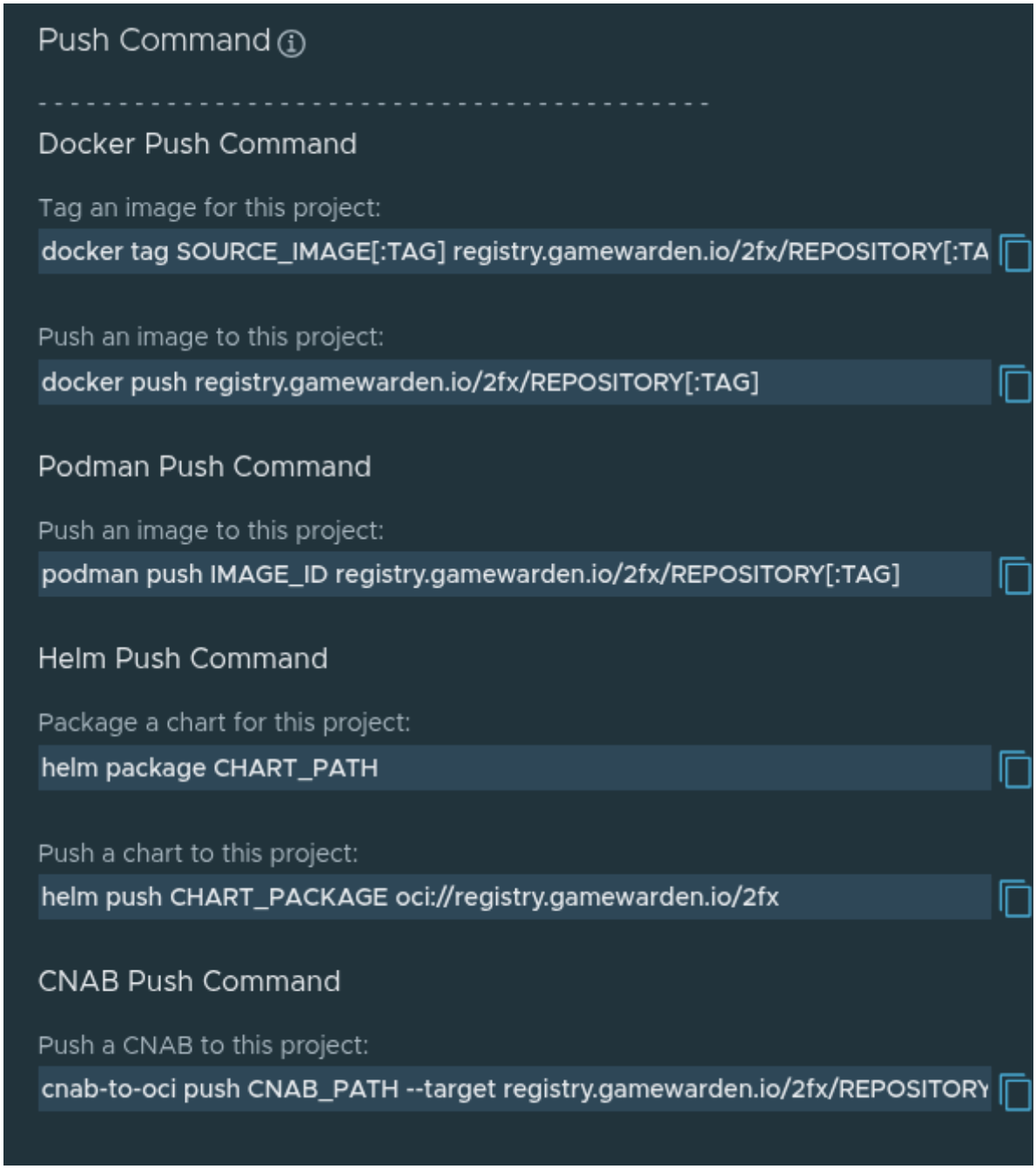
1. Go to Harbor Registry.
2. Select **Login with Game Warden SSO** and enter your P1 SSO credentials.
3. Once logged in, you will be directed to the **Projects** page.

4. Locate and select your assigned project to display all associated artifacts.



5. Click the **Push Command** button to open a modal that provides Docker push command syntax. This syntax is meant to guide you when manually tagging and pushing images to Harbor. You can copy the

sample lines, such as **Tag an image for this project** or **Push an image to this project** directly into your terminal.

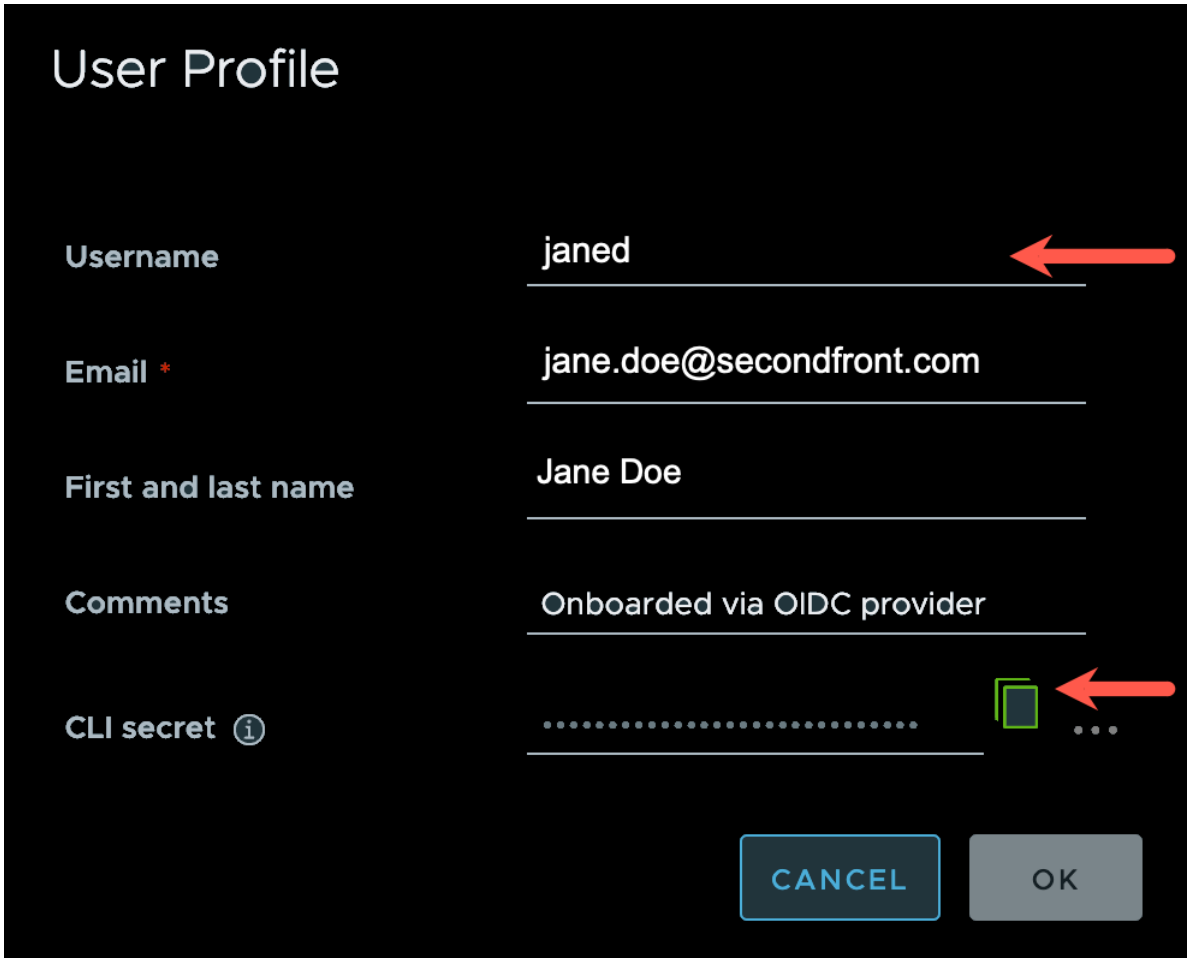


For detailed instructions and example commands, review the Harbor Image Push guide. Be sure to update the syntax to match your image name and tag.

Retrieve Harbor CLI credentials

1. Click your P1 account name in the top-right corner.
2. From the dropdown, choose **User Profile**.

3. In the **User Profile** modal:
    - Note the **Username** — you'll need it for Docker login.
    - Click the **Copy** icon next to the **CLI secret** field. A "Copy success" banner will appear, and your CLI secret will be temporarily stored in your clipboard. Be sure to paste it promptly when logging in with Docker.



4. Click **Cancel** to close the modal.

Push Images to Harbor

This guide explains how Harbor handles image storage and artifact retention, and provides step-by-step guidance for pushing images using Docker commands.

Image capacity and artifact retention

Image capacity

The Harbor registry currently retains only the **five most recent versions of each image**. When a **sixth version** is pushed, the oldest version is automatically deleted. The Game Warden team is working to expand this capacity in the future.

Artifact retention

Game Warden's CI/CD pipelines generate artifacts each time an image is pushed. These artifacts accumulate over time and consume storage. To manage storage and maintain performance:

- Artifacts **older than 15 days** are automatically purged.
- You must initiate the pipeline to push your image to DEV, STG, or PRD **within 15 days** of pushing it to Harbor.
- Once artifacts are purged, you must update and re-push your image to continue deployment.
- Purged artifacts **cannot be retrieved**.

Steps to push images

The example commands below show how to push an image using the following test data:

- Username: `janed`
- Project name: `test-company`
- Image name: `test-image`
- Image tag: `0.0.1`

Replace `<target registry>` with your organization's Harbor registry URL.

- **DoD Deployment**: registry.gamewarden.io
- **FedRAMP Deployment**: registry.fedramp.gamewarden.io
- **Commercial Deployment**: registry.secondfront.com

1. Log in to Harbor via Docker

Open a terminal and log in to Harbor using your **username** and **CLI secret**.

```
docker login -u <username> -p <cli-secret> <target registry>
```

Example command:

```
docker login -u janed -p password123 <target registry>
```

If successful, you'll see: `Login Succeeded`.

If login fails, ensure your Harbor session is still active in a browser. You may need to re-authenticate.

2. Tag the image

Avoid using the following disallowed tags:

- `-hardened`

- `latest`
- `.sig`
- `sha256`
- `development`
- `staging`
- `production`

Use a **semantic version tag** such as `0.2.1`. Your Deployment Passport requires a valid version number.

`docker tag <source-image>:<tag> <target registry>/<project>/<repository>:<tag>`

Example command:

`docker tag your-registry/your-project/previous-image-name:previous-tag \<target registry>test-company/t`

3. List Docker images (optional)

You can verify the image exists locally:

`docker image ls`

4. Push the image to Harbor

`docker push <target registry>/<project>/<repository>:<tag>`

Example command:

`docker push /test-company/test-image:0.0.1`

You should see confirmation messages in your terminal as the image uploads.

5. Confirm the image in Harbor

1. Log into to Harbor with your Platform One (P1) SSO credentials.
2. Select your assigned project from the **Projects** page.
3. On the **Repositories** tab, find your pushed image listed. If the image is not visible:
   - Check your terminal output for errors or warnings.
   - Confirm your user permissions for pushing images.
   - Reach out to Second Front via Slack for assistance.

6. Associate new images with an application

New versions of an existing image appear in **App Central** automatically after they are pushed to Harbor. For a brand-new image (i.e., a new name), you must first associate it with an application.

To associate new images with their respective applications:

1. Log in to Game Warden using your P1 SSO credentials.
2. From the left navigation, select **Images** to open the images dashboard.
3. On the **Inactive Images** tab, locate your recently pushed image.
4. Select the image(s) you want to add to an application.
5. Click **Select Application** in the green action bar, then choose the application that should own the image. The image now appears under **Images** in **App Central** for that application.

   What happens after image push

   Once the image is pushed:

   - A webhook triggers a pipeline that pulls your Dockerfile.
   - The image is built and hardened automatically.

- The hardened version becomes available in Harbor.
- If you push the same image again with a new version tag, a new pipeline is triggered, and both versions will appear in Harbor.

Harbor will always display the most recent push timestamp for each tagged version.

Automated Harbor Credentials

Automated Harbor Credentials allow Game Warden users to securely and programmatically push Docker images to the Harbor Image Registry. This automation improves the efficiency, consistency, and security of your CI/CD pipelines.

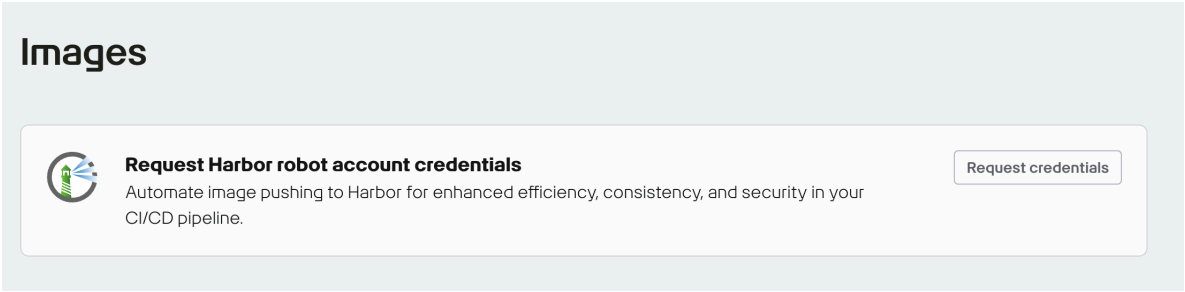
Below is step-by-step guidance on how to request and use automated credentials.

---

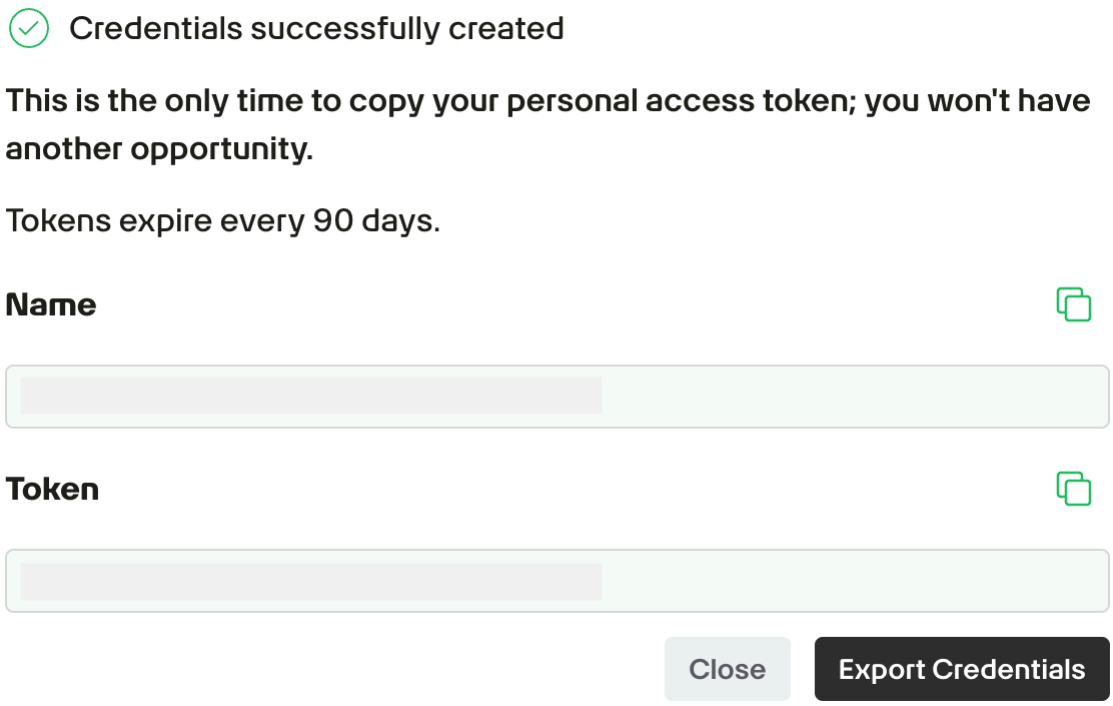
1. Request Harbor credentials

To generate Harbor credentials:

1. Log in to your Game Warden account, then go to the **Images** page.
2. Click the **Request credentials** button in the top-right corner.



You'll see a pop-up window displaying your token and username.



3. Click **Export Credentials** to download them to your machine, then click **Close** to exit the modal.

Important

Tokens are only viewable at the time of creation. Save them immediately. If lost, you'll need to repeat the steps to generate a new token.

---

2. Contact Second Front

Contact your Technical Implementation Manager (TIM) to notify them that you've generated automated Harbor credentials. The Game Warden team will configure your pipeline to use these credentials for automatically pushing images to Harbor.

---

3. Configure your pipeline to automate image pushing

Once the Game Warden team configures your pipeline, you can automate image pushing within your CI/CD workflows.

Important

- Store credentials securely as CI/CD secrets.
- If you're pushing a new image or changing image names, make sure your TIM has configured the corresponding pipeline. Otherwise, the image will not be scanned in Findings.

Authorization Boundary Diagram

An Authorization Boundary Diagram (ABD) is a visual representation of your system's software components, data flows, and security boundaries. It shows how data moves between internal and external systems, which ports and protocols are used, and where system components reside in the Game Warden environment. This diagram helps ensure your system connects securely with the platform and meets all deployment and security requirements.

To streamline onboarding and reduce friction during the Authority to Operate (ATO) process, you must provide an ABD that includes system components, data connections, and the boundaries they traverse.

Understanding data flow and external connections

As part of preparing your ABD, it's important to illustrate how data moves in and out of your system. This helps ensure your application integrates smoothly with Game Warden and meets the security expectations required for deployment.

We look for three types of data flow in your diagram:

- **Ingress** – When data enters your system (e.g., user input or file uploads).
- **Egress** – When data exits your system (e.g., sending logs or API responses).
- **Bidirectional** – When data flows both ways (e.g., interactions with third-party APIs).

Clearly marking these flows makes it easier for our team to assess how your system interacts with other services. See External Data Connections Overview for security considerations and best practices.

Authorization Boundary Diagram requirements

When creating your ABD, clearly show how data moves both within and beyond your application's boundary. Specifically:

- Indicate the direction of data flow: ingress, egress, or bidirectional.
- Specify the ports and protocols used for each connection.
- Include all containers, external systems, and managed services.

This level of detail helps us evaluate security, compliance, and connectivity effectively.

What to include in your diagram

Include the following details:

- **Ports and Protocols in Use** - Specify which ports and protocols are used for each connection (e.g., HTTPS on TCP/443). This helps us understand your system's exposure and ensure it aligns with security best practices.
- **Ingress and Egress Flows** - Indicate which data flows are inbound, outbound, or bidirectional. This helps clarify how external systems interact with your application.
- **External Services or Systems** - Identify any services outside of Game Warden (e.g., customer APIs, logging services) and show how your application connects to them.
- **Data Movement Constraints** - To avoid unintentional data movement across environments—often called *data spillage*—be sure to:
  - Prevent IL4+ data from being transferred to lower IL environments such as IL2.
  - Show government-approved services (such as the Cloud Native Access Point, Boundary Cloud Access Point (BCAP) or proxy gateways) when required by policy.

Boundary examples

The following diagrams illustrate how external users access the Game Warden environment. On the right side of each diagram, you'll see two types of users:

1. One accessing through the Department of Defense (DoD)'s Non-classified Internet Protocol Router Network (NIPRNet)
2. Another accessing via the public internet

At higher Impact Levels (IL4 and IL5), access is controlled through a CNAP/BCAP, which does not exist at IL2. This is the primary difference between IL2 and IL4/IL5 environments. CNAP/BCAP uses an IP whitelist to screen NIPRNet users, allowing access only from approved DoD IP addresses.

Once screened, both NIPRNet and public internet users route through an Internet Gateway to reach the Game Warden Authorization Boundary. From there, they can access AWS services and application containers.

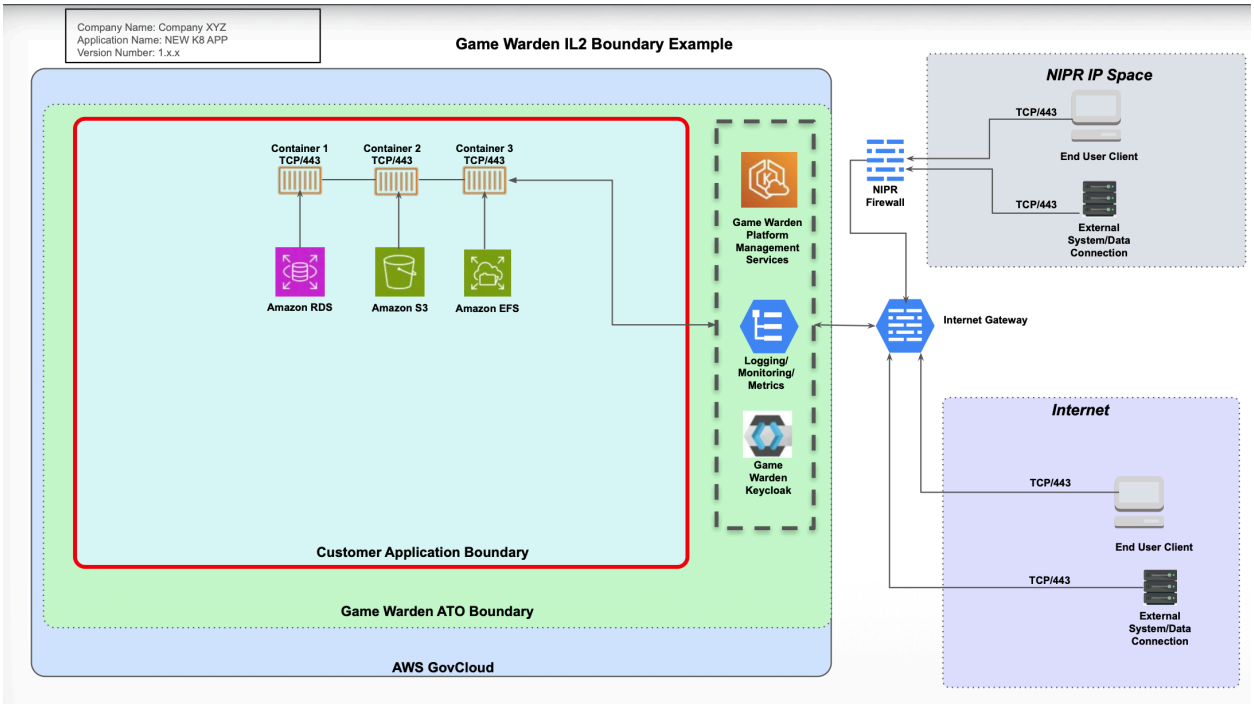
All customer environments are protected by Keycloak for authentication. For applications hosted on Platform One (P1), users must log in with P1 credentials. CNAP/BCAP also acts as a gatekeeper for P1 access at IL4 and IL5.

As shown in the diagrams, external access to the system may involve the following paths:

- NIPRNet
- Public Internet
- Platform One (P1)
- AWS GovCloud

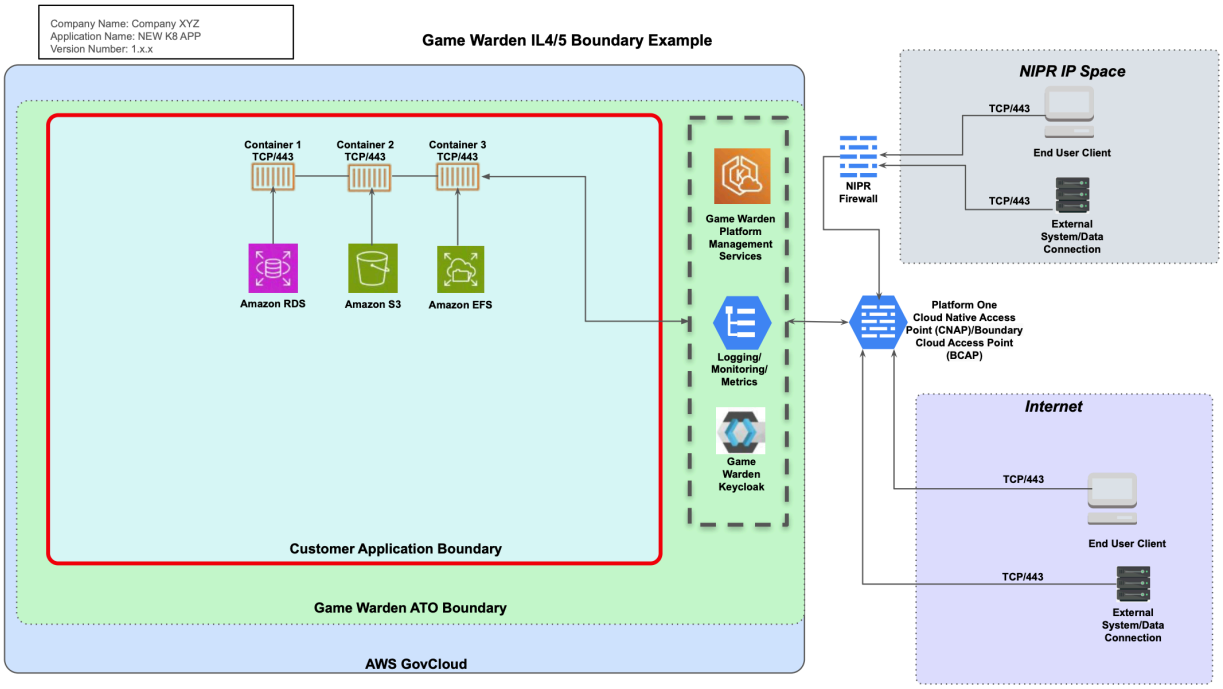
IL2 Boundary Example

The diagram below displays the external data connections, customer containers, and the required platform services.



IL4/IL5 Boundary Example

The diagram below displays the external data connections, customer containers, and the required platform services.



Note

- If your Virtual Desktop Infrastructure (VDI) is on NIPRNet, you can access IL4 and IL5 environments without using Appgate SDP, the DoD-approved authentication service.
- If you are not on NIPRNet or using a NIPRNet-based VPN (such as Air Force Desktop Anywhere), you must use Appgate SDP to access IL4/IL5 applications on `afwerx.dso.mil`. This is a DoD P1 team requirement.
- **For AFWERX customers**: Show CNAP in your diagram (base environment: afwerx.dso.mil)
- **For DISA customers**: Show BCAP in your diagram (base environment: 2F.mil)

Need help with creating your diagram or have questions about your architecture? Reach out to your Technical Implementation Manager (TIM) or Mission Success Manager (MSM).

Populating Databases

There are multiple ways to populate a database, depending on your environment configuration, where your data resides, and application-specific requirements. Please notify the Game Warden team in advance if you require support for data seeding or migration.

Generally, you should either:

- Include services in your application to handle seeding and migration, or
- Provide scripts for the Game Warden team to execute on your behalf.

Data seeding vs. Data migration

Both processes involve managing data but serve different purposes:

- **Seeding**: Populates the database with initial data so the application can run.
- **Migration**: Transfers data from one environment or system to another, typically using a database dump.

  Warning

  In IL4+ environments, direct database access is not permitted.

Data seeding

Seeding provides an application with its first set of data. Examples include:

- Creating an initial Admin user
- Populating locations such as cities, states, and countries
- Seeding demo or test data to verify app functionality

Seed data may be real or synthetic, depending on your needs. We recommend that you handle your own seeding. This gives you more flexibility to choose formats, tools, and timing.

Game Warden engineers can provide support if needed. For instance, you may send seed data to Game Warden, and our team can mount it as a Docker volume to ensure data persists across runs.

> **Data seeding support**
>
> If Game Warden handles your data seeding, this must occur **before** the Functional Assessment, as a populated database is required for testing.
>
> You must provide:
>
> - Seed data
> - Seeding scripts or services
> - Clear execution instructions

Seeding process steps

The process for seeding data into Development (DEV), Staging (STG), or Production (PRD) environments depends on your existing standards, tooling, and whether your application exposes endpoints for data seeding.

- If your application has endpoints enabled for seeding, you are encouraged to manage this process independently.
- If endpoints are not available, or if data sensitivity requires tighter controls, the Game Warden team can assist. In that case, you must provide your seeding scripts and data files.

Game Warden will then issue AWS credentials via the Department of Defense Secure Access File Exchange (DoD SAFE). You will use these credentials to upload your seeding assets to an S3 bucket managed by Game Warden.

Data migration

Data migration involves moving data between systems or environments. This is common when transferring Controlled Unclassified Information (CUI) into Game Warden.

If your application and data sensitivity allow it, you can manage your own migrations. Otherwise, Game Warden will assist using your scripts and our credentials.

There are two primary methods for migrating data into Game Warden:

From an external accredited platform

Submit a request—typically through the external platform's ticketing system—to initiate the data transfer. Once approved, the Game Warden team will provide AWS credentials via the DoD SAFE . The external platform's engineering team can then upload your data to a designated S3 bucket. Using the migration scripts or instructions you provide, the Game Warden team will handle the import into your Game Warden database.

From a database you control

You can request a migration through your customer Slack channel. In most cases, the Game Warden team will already be aware of this need and help coordinate a transfer time. We will provide AWS credentials via DoD SAFE so you can upload your data to a secure S3 bucket. Our team will then use your provided scripts or instructions to migrate the data into your Game Warden database.

If your application supports migration via API endpoints, you may also manage the migration yourself. However, ensure that the data is handled in accordance with all relevant sensitivity and compliance requirements.

Environment-specific considerations

Seeding and migrating data are environment-dependent processes. Although DEV typically requires seeding to get started, STG and PRD often involve data migration closer to go-live.

Your approach may vary depending on the Impact Level (IL), data sensitivity, and your goals. For example:

- You might seed the same data across DEV, STG, and PRD.
- In lower ILs such as IL2, you must use mock or synthetic data—CUI is not allowed.
- In IL4+, seed or migrate data into STG first for validation before moving to PRD.

The Game Warden team commonly recommends dropping a data snapshot into STG for testing. If PRD testing succeeds, it's complete; otherwise, we'll help you revert and retry. Whenever possible, schedule migrations during low-traffic periods to minimize disruption.

Environment-specific considerations

- **DEV**: Use mock/test data only — no CUI. ***STG**: Validate with production-like data.
- **PRD**: Final deployment using mission data.

For IL4+ environments, always validate in STG before migrating to PRD.

Downtime

Migrations, especially from external platforms, typically involve some downtime. We strongly discourage migrating live data from an active application, as this risks data integrity.

We will work with your team to plan migrations during low-usage periods to reduce impact.

Best practices

- Know exactly what data needs to be seeded or migrated.
- Schedule activities to minimize disruption.
- Use tools such as Flyway for automated data migrations or seedings.
- Ensure scripts are tested and included when requesting Game Warden support.
- Always verify data after migration is complete.

Memorandum for Record Requirements

To initiate the onboarding process for AFWERX customers working toward a Certificate to Field (CtF) for Impact Level (IL) 2, 4, or 5 deployments, a **Memorandum for Record (MFR)** must be submitted **before** onboarding can begin. We will request signed MFRs at the customer kickoff.

> **Important**
>
> The MFR is specific to AFWERX deployments only. Other deployments (DISA IL5 or FedRAMP) serving non-AFWERX customers are not subject to this requirement.

What is the MFR?

The MFR is a formal document required by Game Warden's Authority to Operate (ATO). It verifies the mission need, sponsorship, and contract type for your application deployment.

Complete the MFR

1. Download and open the PDF template.
2. Fill out the required field
3. Save the document and send to your Mission Owner for signature.
4. Submit the signed MFR to your Mission Success Manager.

How many MFRs do you need?

You will need **at least one MFR for each instance** of your application deployed for a different mission owner or use case. An instance refers to a specific copy or deployment of your application—even if it shares the same base code.

> **Example**
>
> Suppose your application, *Galactic Donkey*, needs to be deployed for multiple mission owners. Sometimes, multiple mission owners may share the same instance, but in other cases, each mission owner may have their own instance of the application. Even though it's the same core application, each copy is treated as a separate instance when deployed for different use cases or mission owners.
>
> **Application: Galactic Donkey Instances**
>
> - **Mission A**: A copy of *Galactic Donkey* for Mission Owner A
> - **Mission B**: A copy of *Galactic Donkey* for Mission Owner B
>
> In this scenario, you would need **two MFRs**—one for each instance. If Mission Owners A and B **share the same instance**, only **one MFR** is needed.

MFR expirations

Each MFR must be reverified **every six months** from the finalized, AFWERX ISO signature date.

- MFRs are included with your BoE submission for CtF
- MFR expiration date & CtF expiration date are mutually exclusive

If an updated, fully signed MFR cannot be provided, your application will be offboarded from the platform until a valid MFR is submitted.

Deployment Passport Submission Process

This guide outlines the steps to get your Deployment Passport signed by Game Warden's Authorizing Official (AO) after all required documentation has been collected and verified.

Prerequisites

Before the Game Warden security team can prepare your Deployment Passport for review, the following requirements must be met:

1. Resolve all security findings

- **Responsibility**: Customer & Second Front
- **Requirements**: All security findings must be addressed and accepted by the Game Warden security team.

2. Verify Body of Evidence (BoE)

- **Responsibility**: Customer
- **Requirements**: Ensure the Body of Evidence (BoE) is accurate and up to date, including:
  - Components updated to the current version.
  - Mark components not included in the Deployment Passport as *Excluded*.
  - Identify CUI type (if applicable).
  - Personnel details are complete.
  - Government contract information is accurate.
  - All external databases, systems, and dependencies are listed (rare cases).

3. Update the Authorization Boundary Diagram

- **Responsibility**: Customer
- **Requirements**: Diagrams must:
  - Be clear and organized.
  - Match container names in the BoE exactly.
  - Show all external connections and data flows (ingress/egress).
  - Match dependencies, systems, and database names exactly to the BoE.
  - For *External Data Connections*, provide name, ports, protocols, direction, and a complete narrative of data flow.

4. Update the Information Security section

- **Responsibility**: Customer
- **Requirements**: With your Government System Owner and Contract Sponsor, determine:
  - Confidentiality, Integrity, and Availability levels.
  - Classification Level and Security Classification Guide (SCG).
  - Distribution Control Type.
  - CUI status (IL2 contains no CUI).
  - PII details with your Privacy Official.

5. Update the Deployment Information section

- **Responsibility**: Customer
- **Requirements**: Include:
  - All programming languages used.
  - Dependencies (excluding databases).

- Databases used (names must match diagram).

6. Harden all pipelines in Harbor

- **Responsibility**: Second Front
- **Requirements**: Harden all pipelines in Harbor.

7. Upload SAST scan results and attestation

- **Responsibility**: Customer
- **Requirements**: Must be uploaded to the BoE in Game Warden **within 30 days of Deployment Passport submission**.

8. DAST scan approval

- **Responsibility**: Second Front
- **Requirements**:
  - Functioning application in DEV.
  - No architecture or code changes.
  - All findings resolved before submission.

Submission process

Once all prerequisites are complete, your Technical Implementation Manager (TIM) will submit an internal support ticket to initiate the formal review process.

> **Code Freeze**
>
> While your Deployment Passport is under review, do **not** make any changes to the application. Any change may generate new CVEs, requiring resolution and restarting the process.

Review stages

1. Security Team Review

The Game Warden security team checks your Deployment Passport for:

- Complete documentation.
- All CVE findings, SAST, and DAST findings resolved.
- Consistency between the deployed application, BoE, and Authorization Boundary Diagram.

If issues are found, your TIM or Mission Success Managers will work with you to resolve them before resubmission.

2. Third-Party Review (required for AFWERX customers)

Once approved internally, the Deployment Passport is sent to independent assessors for an external accuracy and completeness review.

3. Authorizing Official Review

The final review is done by the Authorizing Official. Upon approval, your application receives a Certificate to Field (CtF)/Software Approval and inherits Game Warden's Authority to Operate (ATO) in DoD environments.

Final steps before deployment

For first-time deployments:

1. Your TIM will schedule a **Pre-Deployment Brief** with Second Front, your team, and your Government Sponsor.
2. This meeting serves as the last review before deployment.
3. Upon completion, your application will be cleared for deployment to the **staging environment**.

FAQ

Can you submit an IL4 renewal Deployment Passport at the same time as an IL5/IL6 Deployment Passport that is being worked on separately, with an earlier renewal date?

Yes — if it's the same application, you can submit all Deployment Passports together. This approach will likely reduce the Authorizing Official's workload.

The DAST scan is run against the DEV environment. As long as you are using a single DEV environment **and** there are **no differences** in the application, the same DAST scan can be used for both submissions.

How Game Warden Protects Your Data

At Second Front Systems, protecting your data is a top priority. We use industry-standard security practices, strict access controls, and continuous monitoring to secure the Game Warden platform and your application data.

> **Security Incident**
>
> If you ever suspect your application's security may have been compromised, report the incident immediately by following the procedure outlined in the Customer Incident Reporting Procedure.

Security best practices

Game Warden incorporates the following industry-standard security measures:

- Annual security awareness training (including phishing and remote work topics)
- Regular penetration tests and security audits
- Customer data and clusters deleted upon request or as required
- CI/CD pipelines secured with industry-standard tools
- End-to-end data encryption (in transit and at rest)
- Periodic vendor security reviews
- Background checks for all employees prior to access
- RBAC and least privilege access enforced across teams
- Data collection limited to what's necessary for service delivery
- Continuous security log reviews
- Regular data snapshots, with support for forensic retention if needed

---

Data encryption

All websites and microservices in Game Warden use SSL/TLS encryption. Sensitive data (e.g., connection credentials) is encrypted both in transit and at rest using industry-standard algorithms. We routinely audit our certificates and encryption protocols to maintain data protection.

> **TLS encryption**
>
> Your application does not need to implement TLS. Game Warden's service mesh uses TLS 1.3 by default.

Security audits & penetration testing

Having to achieve FedRAMP® High Authorization, the Game Warden is now authorized to handle the Federal Government's most sensitive unclassified data. To maintain this authorization and ensure ongoing security posture, Game Warden conducts comprehensive external penetration testing at minimum annually. These assessments employ current industry-standard tools and methodologies to rigorously evaluate platform security controls and identify potential vulnerabilities before they can be exploited.

Game Warden's tenancy model

Game Warden is a multi-tenant SaaS platform that uses strict isolation between tenants:

- Each customer has a dedicated namespace with isolated databases and storage.
- Boundaries are enforced by Istio service mesh using a deny-by-default policy.

- Each customer has a private Harbor image repository.

Access control

Access to Game Warden environments varies depending on the deployment type and security Impact Level (IL). The table below outlines the required access controls:

Deployment Type	Access Requirements
DoD IL2	- Platform One Single Sign-On (P1 SSO) + Keycloak
DoD IL4	- All IL2 requirements + Government-issued access card + Appgate SDP
DoD IL5	- All IL4 requirements + IL5-specific hardening and compliance
FedRAMP / Commercial	- Game Warden account + Keycloak

Secret management

We use SOPS (Secret OPerationS) along with AWS Key Management System (KMS) to encrypt secrets in YAML and JSON files, ensuring no plaintext secrets exist in source code.

Physical security

Game Warden's infrastructure is hosted in secure data centers compliant with:

- ISO 27001
- SOC 1 / SOC 2 / SSAE 16 / ISAE 3402
- PCI DSS Level 1
- FISMA Moderate
- Sarbanes-Oxley (SOX)

For more details, see cloud provider security documentation:

- AWS Security
- Google Cloud Security
- Azure Security

Shared Responsibility Model

Security is a shared responsibility between Game Warden and our customers. For more information, refer to the Game Warden Shared Responsibility Model.

Protecting Customer Source Code & Intellectual Property with Game Warden

At the core of Game Warden's design philosophy is the principle that customer intellectual property (IP) is sovereign and must remain protected. Our platform is built to ensure that customers maintain **control and ownership of their source code** and application assets, while still benefiting from a secure, compliant, and continuously monitored hosting environment.

Unlike traditional hosting or managed service models where vendors may request or require access to customer codebases, Game Warden is structured around **zero access to application source code**. We do not require or retain customer repositories or raw application development files. Instead, software updates are delivered to Game Warden in the form of **Docker container images**. These images contain only the **compiled binaries and runtime dependencies** necessary to deploy and operate the application in production. This containerized delivery model ensures that the underlying source code remains within the customer's control and development pipeline.

This architectural choice provides several important protections:

- **Separation of Responsibilities**: Customers manage and control their own IP, while Game Warden manages and secures the hosting environment. This clear separation reduces risk and ensures that sensitive IP never leaves the customer's boundary.
- **Confidentiality by Design**: Since no source code is ever transmitted, stored, or accessed by Game Warden, there is no vector by which proprietary algorithms, trade secrets, or sensitive business logic could be exposed to our systems or personnel.
- **Integrity of Customer Ownership**: The IP embodied in an application remains solely with the customer. Game Warden's platform and underlying infrastructure remain distinct and independent, ensuring that customers maintain full ownership rights without encumbrance.
- **Security Alignment**: By relying on containerized, binary-only delivery, customers can apply their own secure software development lifecycle (SSDLC) practices, code scanning, and IP protection controls upstream. Game Warden focuses on securing the runtime environment in alignment with the Department of Defense (DoD), FedRAMP, and other compliance frameworks.

In practice, this means that customers can innovate, develop, and protect their applications with confidence, knowing that Game Warden serves as a **secure and compliant execution environment**, not as a custodian of their code. Our responsibility is to deliver the highest levels of security, observability, and compliance at the platform layer, while respecting and protecting the integrity of customer IP at the application layer.

This approach embodies Game Warden's partnership principle: you retain ownership of your IP, and we retain ownership of our platform. This model lets you innovate and protect your IP while Game Warden provides the secure, compliant, continuously monitored execution environment. Clear boundaries ensure your code remains yours and our platform remains ours, giving you full ownership of your most valuable assets while operating in DoD and other high-compliance sectors.

The list of the technical assets Game Warden does require to onboard customer applications can be found here. You can read more about how Game Warden protects your data by visiting our Help Center.

Understanding the Risk Management Framework

The Risk Management Framework (RMF) is a structured process that integrates essential security, privacy, and risk management activities into the system development life cycle.

Although the government and private sectors may apply RMF differently, both approaches highlight the importance of managing risks effectively throughout an application's life cycle. At Second Front (2F), our implementation of RMF is designed to streamline this process for mission and application owners alike.

> **Benefit of using 2F**
>
> 2F simplifies RMF adoption by minimizing the documentation and coordination needed to begin onboarding applications to the Game Warden platform. We provide early-stage support through our Mission Success team, ensuring customers understand their shared responsibilities throughout the process. Our goal is to give you confidence and clarity from day one.

Prepare

In this first phase, the Mission Owner (MO) determines the appropriate Department of Defense (DoD) Impact Level (IL) for the application—IL2, IL4, or IL5—based on the sensitivity of the data involved. This ensures the application is hosted in an appropriately secure environment.

2F collaborates with mission owners to understand the nature of the data and how the application will be used. For example, this may include:

- An application processing government data for analysis
- A system optimizing flight plans using Controlled Unclassified Information (CUI)

This phase also includes identifying key risk management roles on both the customer and 2F sides, outlining a shared responsibility model, and defining continuous monitoring processes.

Additionally, 2F will meet with the application owner to assess the system's handling of CUI.

Categorize

In the Categorize phase, the application and mission owner jointly assess the application's potential impact on confidentiality, integrity, and availability. This determines the system's risk level—low, moderate, or high—and confirms the applicable ILs (e.g., IL2–IL5).

In the government's RMF process, this step also includes:

- Documenting the categorization decision
- Having the categorization reviewed and approved by an Authorizing Official (AO)

At 2F, this step mirrors the government's process but is streamlined for faster execution.

The government application of the categorization step of RMF informs organizational risk management processes and tasks by determining the adverse impact in terms of the loss of confidentiality, integrity, and availability of systems and the information processed, stored, and transmitted by those systems. It involves documenting the security categorization of the system and information, completing the categorization decision, and having it reviewed and approved by the authorizing official.

Select

In the Select phase, the appropriate NIST 800-53 Rev. 4 security controls are chosen based on the application's categorization and environment. These controls are:

- Tailored to each application and mission owner
- Classified as system-specific, hybrid, or common
- Allocated to system components accordingly

2F also develops a continuous monitoring strategy and supports the development of the Body of Evidence (BoE) to document selected controls and responsibilities.

Game Warden accreditation

- Accredited to DoD IL5
- Deployable to IL6 via DAF CLOUDworks/ODIN
- In-process for FedRAMP High Baseline

Implement

In this phase, the selected controls must be implemented and reflected in the system's security and privacy plans. The application and MOs are expected to implement all assigned controls and update documentation accordingly.

2F Difference

The 2F security team enforces NIST 800-53 security controls and provides both a shared responsibility model and a customer risk management matrix. These resources help application and mission owners clearly understand which security controls are already covered by the Game Warden platform—and which ones they are responsible for. By clarifying this division of responsibility, customers can focus on the controls that directly impact their application and move efficiently through the steps required for production deployment.

2F Advantage

The key difference between how 2F and the government implement the RMF lies in responsibility for security controls. In a traditional government process, the application and MOs are responsible for implementing all required controls—often over 600—which can be time-consuming and resource-intensive. With the Game Warden platform, many of these controls are already handled by the platform itself. As a result, customers have fewer controls to address, allowing them to move their application into production and become operational much faster.

Assess

During the Assessment phase of the RMF, government stakeholders evaluate whether security controls are properly implemented, functioning as intended, and meeting system and organizational security and privacy requirements. This phase includes:

- Selecting qualified assessors
- Creating assessment plans
- Addressing any vulnerabilities or deficiencies identified (e.g., from scans)
- Developing a Plan of Action and Milestones (POA&M) to track and resolve outstanding control issues

2F Difference

2F's security assessors collaborate closely with the application owner to integrate the application into the development environment. They use a suite of security tools to scan for and identify various levels of Common Vulnerabilities and Exposures (CVEs). Once the scans are complete, the assessors work with the application owner to remediate any security findings. After all issues are addressed, the assessors generate a Deployment

Passport, which includes a Certificate to Field (CtF)/Software Approval, marking the application ready for deployment.

2F Advantage

Onboarding an application through 2F enables customers to work directly with our security assessor team to integrate your application into the Game Warden development environment, perform vulnerability scans, and address any findings. This collaborative process results in the issuance of a Deployment Passport and a CtF/Software Approval, authorizing the application for production use.

In contrast, the traditional government approach typically assigns an assessment team only for review—leaving application and mission owners fully responsible for implementing controls and resolving vulnerabilities. As a result, the standard Authority to Operate (ATO) process often takes significantly longer to reach production readiness.

Authorize

During the Authorization phase, the government AO reviews a comprehensive authorization package. This package typically includes the executive summary, BoE, assessment reports, and a plan of action with milestones (POA&M). Based on the evidence provided, the AO decides whether to grant an ATO or deny the request.

2F Difference

During this phase, the 2F security team submits the Deployment Passport to the AO for review and approval. Once the AO signs off on the Deployment Passport, the application owner is authorized to transition their application from the development environment into staging and production.

2F Advantage

The key advantage of using 2F over traditional government processes is the significantly shorter timeline to reach production. By partnering closely with the 2F security team, application owners can typically move into production in less than 90 days from the initial contract award. This accelerated pace is achieved through hands-on support throughout the RMF process, culminating in a Deployment Passport that is reviewed and approved by the AO.

Monitor

During the Monitoring phase, the government places full responsibility for ongoing security monitoring of the application and hosting environment on the application owner. As a result, application owners often need to hire a large internal team or contract external resources to manage continuous monitoring and compliance.

2F Difference

Our security team provides continuous monitoring for applications hosted on the Game Warden platform. Container scans are run 24/7 using the platform's integrated security suite. Scan results are shared with both the application owner and the Second Front security team to ensure transparency. When vulnerabilities are detected, detailed guidance is provided to help determine how each issue should be remediated or mitigated based on its risk and context.

2F Advantage

The biggest advantage for application owners using 2F and the Game Warden platform is the ability to inherit the platform's robust security posture. Our security assessor team continuously reviews the results of vulnerability scans run on application containers. As a result, application owners don't need to build or manage their own security tooling—the Game Warden platform provides integrated monitoring and assessment out of the box.

Security Compliance Policy

On October 14, 2022, the Department of Defense (DoD) granted Second Front Systems, Inc. (2F)—and, by extension, the Game Warden platform—its Authority to Operate (ATO). This authorization allows our Platform-as-a-Service (PaaS) to operate within the DoD network at Impact Levels (IL) 2, 4, and 5 for applications deployed to both the Staging (STG) and Production (PRD) environments. 2F develops, maintains, and operates the Game Warden platform.

Deployment Passport

The Security Compliance Policy applies to all 2F customers with applications hosted on our platform and deployed to STG or PRD at ILs 2, 4, and 5.

If you deploy applications to PRD at these ILs, you **must** have a signed Deployment Passport from the government's Information Systems Security Manager (ISSM).

The **Deployment Passport**—a Game Warden–specific term—is a body of evidence containing the artifacts required to meet ATO requirements. An ISSM-signed Deployment Passport allows you to inherit our ATO, granting permission to deploy applications into both STG and PRD at the applicable ILs.

> **Warning**
>
> The ATO is non-transferable and valid only for applications hosted on Game Warden.

The Deployment Passport includes:

- Authorization Boundary Diagram
- Body of Evidence (BoE)
- Common Vulnerabilities and Exposures (CVE) summary

For more information, see Deployment Passport.

Policy definition

The Security Compliance Policy defines the actions 2F will take if your application does not meet CVE remediation timelines. All customers must review, acknowledge, and address CVEs in accordance with our Acceptance Baseline Criteria.

Policy requirements

- **Ongoing maintenance** – Application developers must regularly update and maintain PRD applications.
- **Notification of findings** – 2F will notify developers when security findings are nearing the acceptable remediation timeline. Timely action is required to ensure applications can continue running in PRD.
- **Failure to remediate** – If required actions are not completed within a reasonable timeframe—agreed upon by the ISSM, 2F, and the application developer—2F will follow its ATO guidelines. This may result in service interruption, including removal of the application from STG and PRD.
- **Service restoration** – Service will be restored only after all outstanding vulnerabilities are resolved and approved by the ISSM. If the previously issued Deployment Passport has expired, a new Deployment Passport must be obtained before service is restored.
- **RBAC on Game Warden Platform** – We enforce least-privilege RBAC in our platform clusters.
  - **Allowed**: `Role` + `RoleBinding`; `ClusterRole` + `RoleBinding` (grants access only within the bound namespace).
  - **Not Allowed**: `ClusterRoleBinding` (currently not permitted because it grants cluster-wide privileges and introduces unacceptable risk to core platform services).

ITAR Compliance Notification

Game Warden is hosted in International Traffic in Arms Regulations (ITAR)-compliant environments, including AWS GovCloud and Google Cloud Platform (GCP) Assured Controls, providing a secure foundation for handling regulated data within U.S. data regions.

However, **ITAR compliance for your applications and data is not automatically inherited** from the Game Warden platform. Customers are ultimately responsible for ensuring their own compliance.

Key points

- **Customer Responsibility** - You are responsible for meeting all applicable ITAR requirements for your applications, data, and operations.
- **Notification Requirement** - If your operations or data are subject to ITAR, notify Second Front (2F) so your application is hosted in a Game Warden ITAR-compliant environment.
- **Data Ownership** - All data hosted on Game Warden remains the property of your government Mission Owner (MO). You are responsible for managing compliance practices, including access controls, data markings, and security protocols.
- **Licenses and Authorizations** - You must obtain any required export licenses or authorizations from the U.S. Department of State (DoS).

  Warning

  2F does **not** provide ITAR compliance services and does **not** facilitate the acquisition of export licenses.

Additional guidance

Although Game Warden addresses many ITAR compliance requirements at the platform level, you should carefully review your own ITAR obligations to ensure all regulatory requirements are met.

Refer to your hosting provider's ITAR compliance documentation for details on applicable controls and requirements:

- **AWS GovCloud** – ITAR Compliance for IL5
- **Google Cloud Platform** – ITAR Compliance

AI Tool & Plugin Guidance

Adopting AI tools can accelerate productivity—but only if evaluated and implemented securely. This guide helps Second Front customers:

1. Use the seven-dimension checklist to evaluate AI tools.
2. Shortlist tools that meet Second Front's security and business requirements.
3. Use the ROAD framework to implement and maintain models securely.

Evaluating AI tools & plugins

Use the seven dimensions below to vet any AI tool or plugin:

1. Purpose & Business Value

Evaluation Question	Why It Matters	Example
What problem or workflow does this tool solve?	Ensures the tool aligns with real use cases (e.g., automation, summarization, discovery).	Summarizing incident reports or automating code generation.
Does the ROI justify the investment (cost, time, training)?	Helps prioritize tools with high impact and efficient adoption.	$30/user/month tool that saves 2 hours/week of manual tagging.

2. Data Security & Privacy

Evaluation Question	Why It Matters	Example
Where does the input data go (stored, sent, trained on)?	Prevents accidental data leakage or exposure to external models.	Data may be used to retrain vendor models without explicit opt-out.
Does the tool comply with relevant regulations (e.g., NIST, FedRAMP, GDPR, CCPA)?	Verifies alignment with legal and organizational policy.	Required for systems operating in DoD or public sector.
Where is the data physically stored (data residency)?	Ensures geographic compliance with data sovereignty laws.	EU data must stay within EU-owned infrastructure.
Who has access to the data and how is it controlled?	Limits risk of internal misuse or unauthorized vendor access.	Role-based access control with audit logging.

3. IP & Legal

Evaluation Question	Why It Matters	Example
Who owns the AI-generated outputs?	Clarifies rights over deliverables and reduces IP disputes.	Is your organization the sole owner of AI-generated reports?
Could generated outputs carry copyright or license risks?	Mitigates reuse of copyrighted or GPL-licensed content.	Generated code may resemble open-source under restrictive licenses.
Are the vendor's ToS and DPAs acceptable to your legal team?	Protects your org from liability and clarifies responsibilities.	Review of terms may reveal data reuse clauses.

4. Model & Tool Performance

Evaluation Question	Why It Matters	Example
Are the outputs accurate and reliable?	Reduces risk of hallucinations or faulty recommendations.	Factual errors in policy summaries can lead to bad decisions.
Is there an audit trail for actions or content generation?	Supports traceability for compliance or incident review.	Logging inputs/outputs for each prompt.
Can human review be inserted before external use?	Allows verification of AI outputs in high-risk workflows.	Manual approval step before publishing generated content.

5. Integration & Operability

Evaluation Question	Why It Matters	Example
Does the tool offer APIs or SDKs for integration?	Ensures seamless fit into current systems and pipelines.	REST API that integrates with Slack or internal dashboards.
Can the tool scale with current and projected usage?	Prevents performance bottlenecks and cost overruns.	Handles 1000+ batch prompts for nightly data labeling.

6. Vendor Evaluation

Evaluation Question	Why It Matters	Example
Is the vendor trustworthy and transparent about security?	Reduces risk of poor security practices or unreported breaches.	Published audit reports or SOC 2 certification.
Does the vendor offer detailed technical docs or whitepapers?	Indicates maturity and openness.	Security whitepaper detailing model isolation.
Are the support and SLAs adequate for your needs?	Ensures timely response for high-impact issues.	Dedicated support within 4 hours for P1 issues.

7. Cost & Licensing

Evaluation Question	Why It Matters	Example
Is the pricing model predictable as usage grows?	Prevents unexpected costs as adoption scales.	Usage-based pricing can balloon with high volume.
Can you manage seats, roles, or licenses centrally?	Supports secure, auditable user access management.	Admin portal with SSO and RBAC support.

> **Tip**
>
> Create a simple scorecard for each tool to document your evaluation process.

Operationalizing AI/ML with the ROAD framework

Use the **ROAD framework** to move from prototype to production:

R – Requirements

Key Activity	Description	Example
Define the business problem	Ensure clarity on what the AI/ML system is solving.	Detect insider threats in real time.
Set measurable objectives	Define success criteria (e.g., accuracy, latency, savings).	90% threat detection rate with <2% false positives.
Gather constraints	Document compliance, timeline, privacy, and resource limits.	FedRAMP compliance within 3 months.
Align stakeholders	Confirm buy-in from legal, security, product, and engineering.	Weekly syncs with legal, data, and platform teams.

O – Operationalize Data

Key Activity	Description	Example
Data acquisition	Identify, collect, and define internal/external data sources.	Logs, cloud audit trails, user access records.
Data quality	Clean, validate, label, and normalize data.	Standardize timestamp formats across logs.
Data governance	Apply privacy, security, and retention controls.	Enforce encryption, RBAC, and retention windows.
Automate data pipelines	Build reproducible ETL/ELT flows with versioned data.	Use Airflow to run daily ingestion jobs.
Monitor data drift	Detect changes in incoming data distributions.	Alert if login behavior shifts >20% week-over-week.

A – Analytics

Key Activity	Description	Example
Model development	Build, train, and evaluate model candidates.	Train anomaly detector using historical alerts.
Experimentation	A/B test models, tweak features, and compare outputs.	Evaluate recall vs. false positives.
Responsible AI	Apply fairness, interpretability, and bias checks.	Use SHAP values to explain scoring.
Documentation	Track rationale, metrics, and decisions for auditability.	Model card with architecture, accuracy, and limitations.

D – Deployment

Key Activity	Description	Example
Operationalize model	Package and deploy models (batch, real-time, or edge).	Serve predictions via API using FastAPI or SageMaker.
Monitor performance	Track degradation, data drift, latency, and uptime.	Grafana alerts for latency >500ms.
Implement feedback loops	Collect real-world input to refine the model over time.	Flag model decisions users correct.
Ensure reliability & scalability	Handle production workloads and failover scenarios.	Auto-scaling Kubernetes pods on inference load.

Key Activity	Description	Example
Lifecycle management	Version, deprecate, or retrain models as needed.	Tag v1.2 as stable, archive v0.9.

Need help?

Submit a support ticket for guidance on:

- Reviewing AI tool evaluations
- Aligning with security and compliance requirements
- Deploying AI/ML in FedRAMP environments

Dynamic Application Security Testing

Dynamic Application Security Testing (DAST) is the process of analyzing a web application through the front-end to find vulnerabilities through simulated attacks. Game Warden conducts DAST as part of the routine security screening of your application.

DAST artifacts requirement

As part of our security screening, Second Front (2F) **performs and includes** DAST artifacts in **your application's Authorization Package**. These artifacts are critical to provide government accrediting officials with evidence needed to assess and approve your application.

DAST artifacts are required in the following cases:

- **Initial CtF/Software Approval**: Your application has not yet received a Certificate to Field (CtF)/Software Approval.
- **Renewal CtF/Software Approval**: Your application is undergoing CtF/Software Approval renewal.
- **Significant change**: Major updates since the last CtF/Software Approval will require reauthorization and a new CtF/Software Approval approved by the government accrediting official.
- **Ad-hoc requests**: To meet continuous monitoring requirements.

DAST Acceptable Baseline Criteria

2F uses a *"Meets / Does Not Meet"* framework for DAST evaluations:

- **Meets**:
  - Zero Critical/High security-related findings mapped to CWE/OWASP in the final report (except confirmed false positives).
  - All Medium/Low security-related findings mapped to CWE/OWASP include justifications, mitigations, and remediation timelines.
- **Does Not Meet**:
  - Any Critical/High security-related findings mapped to CWE/OWASP not remediated or documented as a false positive.
  - Any Medium/Low security-related findings mapped to CWE/OWASP missing a justification, mitigation, or remediation timeline.

DAST Acceptance Baseline Criteria severity table

Severity	Requirement	Remediation Timeline
Critical & High	Zero findings: all Critical and High findings must be remediated or documented as confirmed false positives **before** generating the final DAST artifact for submission.	Immediately, before DAST artifact submission.
Medium	Include justification, mitigation, and a remediation timeline plan.	Within 90 days of the scan date.
Low	Include justification, mitigation, and a remediation timeline plan.	Within 180 days of the scan date.

Authenticated scan prerequisites

For DoD deployments, if your application doesn't automatically grant Keycloak users access to your DEV endpoint, provision a standard test account for Second Front:

- **Email format**: {company}-test@secondfront.com (e.g., acme-test@secondfront.com)
- **Permissions**: basic/user-level access only (no admin or elevated roles)

This account must be able to sign in and exercise core application functionality needed for DAST without administrative privileges.

Scan prioritization and timeline

An authenticated scan will be conducted against the application in a production-like environment, typically the staging environment.

Static Application Security Testing

Static Application Security Testing (SAST) analyzes application source code, bytecode, or binaries without executing the application to detect security vulnerabilities. This white-box testing approach allows early detection and remediation of issues before deployment.

SAST helps identify:

- Code-level vulnerabilities (SQL Injection, XSS, insecure cryptography)
- Sensitive data exposure (hardcoded passwords, API keys, secrets)
- Issues aligned with Common Weakness Enumeration (CWE) or Open Worldwide Application Security Project (OWASP) Top 10 and other mappings such as CWE Top 25

Regular SAST scanning improves security posture, reduces compliance risk, and ensures production-ready code.

What is an Artifact?

An artifact is the output of your SAST scan and should include:

- Severity of each finding (Critical/High/Medium/Low)
- Justification and mitigation for Medium/Low findings
- CWE or OWASP Top 10 references for security findings
- Scanned commit hash or release version
- Scan tool name and version
- Scan date

The artifact is the official deliverable submitted to Game Warden for review and audit purposes.

SAST artifacts requirement

As part of our security screening of your application, Second Front Systems requires artifacts stemming from SAST. Artifacts from these testing regimes are crucial components of the **Authorization Package**, the body of evidence provided to government accrediting officials that facilitates a rapid risk determination for your application.

Customers must:

- Perform SAST scans on all codebases included in the containers/images deployed onto Game Warden.
  - Attest in the Body of Evidence (BoE) that the scanned code matches the deployed code.
- Generate SAST artifacts **no older than 30 days at the time of Authorization Package submission**.
- Remediate all Critical and High findings before submission.
  - Reports may only include Critical/High findings if documented as confirmed false positives with supporting evidence.
- Provide justifications, mitigations, and remediation timelines for Medium and Low findings.
- Upload artifacts in an accepted format (PDF, XML, HTML, JSON).

In the Game Warden application, you can upload these artifacts and check the **Attestation** checkbox in the BoE/BoE. Accepted artifact formats are **PDF**, **XML**, **HTML/HTM**, or **JSON**.

> **SAST artifacts requirements**
>
> SAST artifacts are required for any of the following:
>
> - **Initial Authorization**: Application has not received an authorization from a government accrediting official.

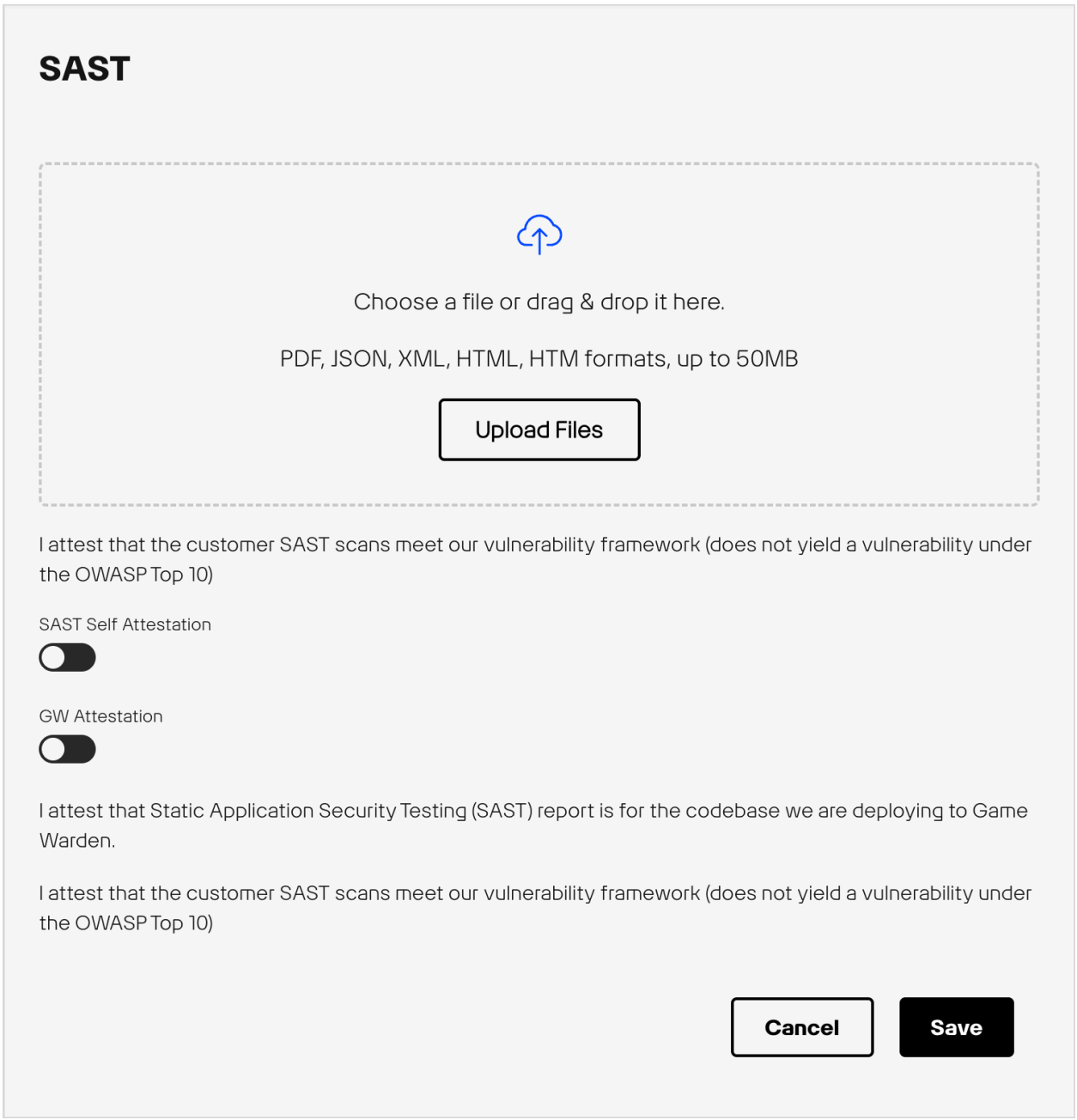


Figure 1: Uploas SAST Artifacts

- **Renewal Authorization**: Application currently undergoing authorization renewal with government accrediting official.
- **Significant Change Authorization**: The application requires a new authorization approved by government accrediting official due to a major version release or significant change from the previous authorization.
- **Ad hoc requests** to support government accrediting official continuous monitoring requirements.

Clarifying SAST Scan scope

To ensure complete coverage, customers should include:

- Application source code (all languages and modules used in deployment)
- Dependencies or libraries if scanned by SAST

Warning

Scanning only a development branch not intended for production or partial code may result in incomplete findings and non-compliant artifact submissions.

CWE guidance for SAST

The CWE is a standardized list of software and hardware weaknesses maintained by MITRE. Each CWE has a unique identifier (CWE-ID) describing the underlying code or design weakness.

CWE is widely referenced by SAST tooling to classify detected issues. *CWE is not a scanner itself and is only utilized as a classification system.* SAST tools map findings to CWEs, helping developers and security teams to understand the type and severity of the risk.

How CWE relates to SAST and Game Warden

- SAST tools map detected weaknesses to CWE-IDs.
- Each CWE indicates the type of weakness and potential security impact.
- Game Warden uses CWEs to define which issues are critical to remediate for production deployments.

Example

A SAST tool flags a SQL query constructed from user input. It maps to CWE-89 (SQL Injection), which falls under the broader Injection category within the OWASP Top 10 web application security risks. The SAST tool has flagged this as a critical finding and would need to be remediated prior to artifact submission.

SAST Acceptable Baseline Criteria

Second Front Systems employs a **"Meets/Does Not Meet"** framework to determine SAST artifact acceptance.

Benefits	Details
Government-equivalent security	Applications are hosted in a secure AWS US East environment that mimics IL2/IL4 practices, including: CVE scanning; SAST/DAST attestation from the security team (*); Body of Evidence (BoE); all without requiring a formal Certificate to Field (CtF)/Software Approval.
Faster time to market	No FedRAMP or IL4/IL5 authorization required to onboard, deploy, or launch. Start engaging customers and iterating on your product sooner.
Flexible onboarding and growth path	Start in Commercial and migrate seamlessly to IL2, IL4, IL5, or FedRAMP environments when you're ready-no need to start over.
Self-service access with managed support	Self-register and begin setup with help from Second Front implementation engineers and support teams.

Important

Only security-related findings mapped to CWE and OWASP are enforced under Game Warden SAST policy. Categories such as maintainability, code reliability, or other non-security findings do not require remediation unless they introduce a security risk.

SAST Severity and Remediation table

Severity	Requirement	Remediation Timeline
Critical & High	Zero findings: all Critical and High findings must be remediated or documented as confirmed false positives **before** generating the final SAST artifact for submission.	Immediately, before DAST artifact submission.
Medium	Include justification, mitigation, and a remediation timeline plan.	Within 90 days of the scan date.
Low	Include justification, mitigation, and a remediation timeline plan.	Within 180 days of the scan date.

Important

The final SAST artifact must not contain any unresolved Critical/High security-related findings.

- Remediate, rerun SAST scans and generate a clean SAST artifact
- Document false positives with supporting evidence
- Reports with open Critical and High findings will be rejected and delay authorization

---

Customer example SAST checklist and best practices

Before the SAST Scan:

- Identify all codebases deploying to Game Warden
- Configure SAST tool to focus on OWASP Top 10 and CWE security rules
- Only security-related CWE/OWASP findings need remediation; maintainability or non-security related findings can be de-prioritized unless they introduce security risk
- Align the scan with the exact branch or tag that will be deployed on Game Warden

SAST Scanning process:

- Run the scan in a CI/CD pipeline or automated job for reproducibility

After the SAST Scan:

- Capture the commit hash or release version of the scanned code
- Remediate all Critical and High security-related findings and rerun scan before submission
- Document remaining Critical and High findings as false positives with evidence if applicable

  Note: For Medium and Low security-related findings, provide justification, mitigations and remediation timelines per severity table.
- Confirm scanned code matched deployed code onto Game Warden
- Generate the report in an approved format (PDF, XML, HTML, JSON)
- Attach artifact to Body of Evidence (BoE) and fill out attestation

Final submission check:

Submit only when your SAST artifact contains no unresolved Critical/High security-related findings and all medium and low findings have justifications, mitigations and remediation timelines documented.

Ongoing best practices:

- Run SAST scans at least every 30 days or before major releases
- Integrate scans into CI/CD pipelines to catch issues early
- Track and trend security-related findings over time to monitor posture
- Update SAST rulesets and tooling regularly for new vulnerabilities

---

Common pitfalls

Do I need to fix duplicated strings, maintainability, code smells or other non-security related warnings reported by my SAST tool?

No. These issues are not required for Game Warden SAST submission unless they introduce a security risk. Focus primarily on security-related findings mapped to CWE and OWASP Top 10.

How should I handle false positives?

All false positives must be documented with supporting evidence in the SAST artifact. This ensures auditors and reviewers understand why a reported issue is not a real vulnerability.

Can I only scan a development branch or part of my code base not being deployed onto Game Warden?

No. Only scanning a subset of code or development branch not intended for production may result in incomplete findings and non-compliant artifacts. Always scan the full codebase that will be deployed onto Game Warden.

What if the artifact does not match the deployed code?

Ensure the SAST artifact corresponds to the exact commit or release version being deployed. Mismatched artifacts can lead to audit issues and rejected submissions.

How do I interpret tool categories such as maintainability or reliability?

For Game Warden, focus on security-related findings mapped to CWE and OWASP Top 10. Other categories (maintainability, reliability, code style) are optional unless they introduce a security risk and can be addressed according to your internal priorities.

SAST scanning tools

When selecting SAST tools suitable for government use, popular options include:

- Checkmarx
- Veracode
- SonarQube
- Fortify on Demand
- Coverity
- Aikido

These tools provide robust security features, strong compliance capabilities, and support for government-mandated standards, making them well-suited for sensitive applications.

Additional resources

- CWE Top 25
- MITRE CWE List
- OWASP Top 10

Acceptance Baseline Criteria

To ensure secure and reliable container deployment to the Production (PRD) environment, Game Warden enforces an **Acceptance Baseline Criteria (ABC)**. These requirements define the minimum security, compliance, and operational standards for container approval. Meeting these criteria is essential to maintain platform integrity and reduce risk.

Why this matters

Traditional DoD container hardening and certification processes are often time-consuming and inconsistent. These delays can prevent applications from being published to DoD repositories and available as hardened images for consumers.

Game Warden addresses this challenge by providing a **clear, consistent, and policy-aligned framework**, reducing ambiguity for contributors and accelerating time to deployment.

Scope and alignment

Game Warden's ABC aligns with the Defense Information Systems Agency (DISA) DevSecOps Enterprise Container Hardening Guide v1.2 (sections 2.2–2.4) to provide clear, consistent guidance for all users.

This living policy will be updated as needed. Major changes affecting partners or customers will be clearly communicated.

Methodology

Game Warden follows to the following process:

- Document and enumerate all container acceptance requirements
- Align with current DISA and DoD guidance
- Define clear, centralized minimum standards for container compliance
- Support exception processes with defined justification workflows

---

Minimum acceptance requirements

To be eligible for deployment in Game Warden, all containers must meet the following minimum acceptance criteria:

1. Fundamental requirements

- Remove unnecessary software or services not required in production
- Disable unused features and secure default configurations to reduce the attack surface

2. Scanning requirements

- Do not encrypt container contents to bypass scanners
- All built containers and downloaded files must be scanned for viruses using **ClamAV**
- Containers must comply with all Game Warden scanning policies

3. Compliance requirements

Base images

- Critical vulnerabilities must be mitigated through configuration, hardening, or alternative base images using the timelines in **Table A** and **Table B**.
- If flaws remain, the affected component must be removed or formally accepted as a risk by Second Front and the government Authorizing Official (AO).

Scanning schedule

- **Pre-deployment:** Applications that are not yet in staging (STG) or production (PRD) environments will be scanned during the pipeline build process. All identified vulnerabilities must be resolved and accepted **before** any container images—whether initial or updated—can be promoted to STG or PRD.
- **Post-deployment:** Applications already deployed to STG or PRD environments are scanned **daily**. Any newly discovered vulnerabilities must be addressed—through either remediation or mitigation—within the timeframes specified in **Table A** (vulnerability SLAs) and **Table B** (compliance SLAs).

Vulnerability resolution

- All vulnerabilities found across all containers must be either remediated or mitigated, in accordance with **Table C**. Exceptions may apply in the following cases:
  - The finding is a **false positive**, or has been officially disputed in the CVE database.
  - The CVE has been **officially withdrawn**, and the contributor does not contest the withdrawal.
  - The container's **configuration or installation** proves it is not vulnerable, with supporting documentation provided by the customer.
  - **No fix is currently available**, and the software is not end-of-life (EoL) and continues to be actively maintained. A justification is still required, and best-effort environmental mitigations must be applied.
  - The issue originates from an upstream package (not the OS distribution such as RedHat, Debian, Ubuntu) and the upstream maintainer has stated they will not provide a fix.
  - The vulnerability is marked as **WONT_FIX** by the OS distribution vendor. This exception only applies to OS-distributed packages.

Permissions

- Do not grant overly permissive file permissions within container images. This includes, but is not limited to:
  - Setting the **SUID or GUID bit** on any file
  - Allowing other write access to system directories or restricted files
  - Modifying read-only files (e.g., 0400, 0440) to be writable
  - Using "**777 permissions**", which allow unrestricted read/write/execute access
  - Granting other execute permissions to files outside standard executable paths such as `/bin` or `/usr/bin`

Support requirements

- All components in submitted containers must have active upstream support (e.g., commercial, or open source).

Remediation and mitigation guidelines

Remediation

A finding is considered **remediated** when:

- The vulnerable component is updated or removed
- The issue is no longer detected in regular scans

Refer to remediation timelines in **Table A**.

Mitigation

A finding is considered **mitigated** when:

- The component cannot be accessed or exploited under normal conditions
- The vulnerable function is disabled, blocked, or isolated
- A timeline for remediation is provided alongside detailed mitigation documentation

Supporting artifacts are strongly encouraged.

Compliance guidelines

Game Warden uses **Anchore** to enforce policy rules and perform automated compliance scanning, aligned with **NIST 800-53**.

Common findings include:

- Insecure file/folder permissions
- Secrets, embedded credentials, or hardcoded passwords
- Missing `RUN` and `HEALTHCHECK` commands in Dockerfiles (if applicable)
- Containers must execute as a **non-root** user unless justified

  Warning

  Failure to provide timely remediation, mitigation, or justification may result in:

  - Blocking container deployment to staging or production
  - Out-of-compliance designation under Game Warden standards

---

Reference tables

The following tables are referenced in previous sections:

Table A – Vulnerability Lifecycle Timelines and Service Level Agreements

Severity	CVSS Score	Remediation Timeline
Critical	9.0 – 10.0	**30 calendar days** from the date of detection if a fix is available. If there is no fix within timelines, justifications apply until a fix is available.
Important/High	7.0 – 8.9	**30 calendar days** from the date of detection if a fix is available. If there is no fix within timelines, justifications apply until a fix is available.

Severity	CVSS Score	Remediation Timeline
Moderate/Medium	4.0 – 6.9	**90 calendar days** from the date of detection if a fix is available. If there is no fix within timelines, justifications apply until a fix is available.
Low	0.0 – 3.9	**180 calendar days** from the date of detection if a fix is available. If there is no fix within timelines, justifications apply until a fix is available.
Negligible/N/A	N/A	**180 calendar days** from the date of detection if a fix is available. If there is no fix within timelines, justifications apply until a fix is available.

Table B – Compliance Results Lifecycle Timelines

Severity	Description	Remediation/Justification
Stop	Critical error that should stop the deployment by failing the policy evaluation, similar to a high vulnerability.	Remediate **30 calendar days** from the date of detection.
Warn	Issue a warning, similar to a Medium vulnerability.	Remediate **90 calendar days** from the date of detection.
Go	OK to proceed, similar to a Low vulnerability.	Remediate **180 calendar days** from the date of detection.

Table C – Justifications

Resolution: Upstream Contributor / Package Manager	**Finding Justification Guidelines**	**Additional Information**
False Positive	No mitigation or remediation required.	False positives include items that a scanner incorrectly identifies such as a wrong package or version. This does NOT include findings that are mitigated or "not exploitable". Evidence as to why the CVE is marked "False Positive" must be provided.

Resolution: Upstream Contributor / Package Manager	Finding Justification Guidelines	Additional Information
Disputed	No mitigation or remediation required	Issues marked as DISPUTED within the National Vulnerability Database (NVD) or CVE Program (CVE.org). This does NOT include issues a contributor is disputing. It must be marked as such within the NVD or CVE.org.
Won't Fix	Must be mitigated	Issues that will not be fixed by the vendor or upstream. They state they will not fix the security flaw. A mitigating action or statement must be provided.
Pending Resolution	Must be mitigated. Must be remediated as soon as a fix becomes available, following the remediation timeline (Table A).	Packages or libraries provided by the Operating System distribution or Upstream project are aware of vulnerability and are tracking the issue to fix. A mitigating action or statement must be provided.
Policy N/A	No mitigation or remediation required.	Product functionality requires security policy exceptions. (Only applies to policy findings, not CVEs.)
Other	—	Only used when all other resolutions do not apply. Comments are required. (e.g., CVE REJECTED by NVD)

Tip

For additional information, see Common Vulnerabilities and Exposures.

Addressing Common Vulnerabilities

Identifying and addressing vulnerabilities is a critical part of securing applications and infrastructure on the **Game Warden** platform. This guide explains the types of vulnerabilities you may encounter, the steps to remediate or justify them, and best practices for keeping your applications secure. It also outlines the processes, tools, and policies we follow to ensure compliance and maintain a strong security posture.

Common Terms

Below are common terms used when discussing cybersecurity vulnerabilities:

- **Vulnerability** - A weakness in a system, security process, internal control, or implementation that can be exploited or triggered by a threat source.
- **Remediate / Fix / Resolve** - Different terms for the same outcome: eliminating a vulnerability through a configuration change or patch.
- **Mitigation** - Applying compensating controls to reduce risk (e.g., adding a Web Application Firewall).
- **Justification** - Documenting why a vulnerability remains (e.g., no vendor patch exists, or exploitation is unlikely). For example: a vulnerability requiring physical access only.

Best practices

The NIST Application Security Guide outlines security concerns related to container technologies and offers practical recommendations for planning, implementing, and maintaining containers securely.

- **Justification and remediation are version-specific** — You must address vulnerabilities for the specific version you intend to include in your Body of Evidence (BoE).
- **Version accuracy is critical** to maintaining security and compliance.

For more details on mitigation and remediation, see Acceptance Baseline Criteria.

Tip

- In Findings, group Common Vulnerabilities and Exposures (CVEs) by "Policy" to view related vulnerabilities together and bulk-edit.
- You can select multiple CVEs at once to bulk-submit resolutions.

All CVEs must be remediated before pushing an image to Staging (STG) or Production (PRD) environments. Past-due remediation gives Game Warden the right to remove the affected environment at any time.

CVE status in Findings

Status	Definition
Unresolved	The CVE has not yet had a remediation or justification applied.
Pending	The CVE has been submitted for the Game Warden security team's review.
Accepted	The Game Warden security team has accepted your resolution.
Rejected	The Game Warden security team needs additional details for your resolution.

Vulnerability management guidelines for container images

Effective vulnerability management for container images requires a two-pronged approach: **maintaining the image itself** and **correcting defects in its configuration**.

Base image management

Keeping base images updated helps reduce the number of CVEs flagged by scanning tools.

- **Recommendation**: Use base images from trusted sources that are frequently updated.
- **Proactive Security**: By starting with a secure, current base, you minimize the inherited risk for all subsequent layers of the container.

Image configuration defects

Even fully patched images can be vulnerable if they are poorly configured. These risks are often not assigned a CVE but represent significant security risk.

Defect Type	Risk / Impact
Running as Root	Allows an attacker to gain maximum privilege if the container is compromised.
Unnecessary Services	Extra packages/services expand the attack surface.

Best practices for resolving container vulnerabilities

Whenever possible, the goal is to resolve vulnerabilities completely. Reviewing scan results and planning remediation efforts should follow a risk-based approach within the Game Warden application.

1. Prioritization & remediation strategy

- **Remediate Critical/High, Medium, and Low CVEs** to the fullest extent possible.
- **Prioritize with context** — do not rely on CVSS score alone. Focus on vulnerabilities that:
  - Have a high **Exploit Prediction Scoring System (EPSS)** score (indicating likely exploitation).
  - Appear in the **CISA Known Exploited Vulnerabilities (KEV) Catalog**: https://www.cisa.gov/known-exploited-vulnerabilities-catalog.
  - Have publicly available **exploitable Proof-of-Concept (PoC)** code.
- **Remove unnecessary packages**: Eliminate all non-essential packages, libraries, and tooling to reduce the total CVE surface area and shrink the attack footprint (also known as **image minimization**).

2. Resolution techniques

- **Rebuild and update regularly**: The most effective remediation approach is to rebuild container images using:
  - A **patched, up-to-date base image**, and
  - The latest secure versions of your application dependencies.
- **Apply fixes in the Dockerfile (not at runtime)**: Avoid patching containers after they are running. Instead:
  - Update the **Dockerfile** to include the required package upgrades or version bumps.
  - This ensures the remediation is **permanent, repeatable, and version-controlled**.

Reference

Game Warden Acceptance Baseline Criteria Policy

Documenting CVE statements and justifications

A CVE statement is a critical piece that summarizes your mitigation and justification actions to formally document the risk of a vulnerability. Accuracy and clarity are essential for audit and compliance.

Key principles of a CVE statement and risk exception

Principle	Clarity and Best Practice Guidance
Justifications Are Temporary	A justification is a **time-bound risk exception**. When a definitive fix becomes available (e.g., patch, package update, or new base image release), your team must prioritize applying the fix and verify resolution through a new scan.
Justification Scope Is Narrow	An approved justification for one image version (e.g., `app:v1.0.0`) does **not** automatically apply to a later version (e.g., `app:v1.0.1`). Container images are immutable—any change in code or dependencies creates a **new, distinct** image. Non-exploitable rationales must be **revalidated per image digest**.
Mitigation Must Be Contextual and Verifiable	Any mitigation (e.g., compensating controls) must be **directly verifiable**, **operational in the runtime environment**, and **monitored for continued effectiveness**. A justification is only valid while the compensating control remains active.

Writing an effective vulnerability justification

Use clear, specific, and technical language when documenting your rationale. The focus should be on why the vulnerability is not exploitable *in your specific context*.

Required Component	Best Practice for Clarity
Identify Affected Package & Location	State the package name, version, and the image layer it originates from. Clearly indicate whether it comes from the base image or from an application dependency.
Describe Vulnerability & Impact	Summarize the CVE and its potential impact. Explain the *exploitation vector*—how an attacker could realistically exploit the vulnerability.
Provide Mitigation Measures	Document the specific security controls or compensating controls in place that prevent exploitation. See Justification and exception handling for more details.
State Resolution / Fix Timeline	If a fix is planned, provide a concrete date or version number. This converts the justification into a **time-bound remediation plan**.
Include Risk Rationale	Provide a clear technical reason for risk acceptance, such as: "The vulnerable function is not invoked by the application," or "The required service is not running within the container."

Example of poorly written justifications

- **Example 1 - Pending Resolution**: https://bugzilla.redhat.com/show_bug.cgi?id=2133616

- **Example 2 - Not Vulnerable**: Sample Statement: Our application is not vulnerable to this CVE.

Example of well written justification

- **Example 1 - Pending Resolution**: The security fix has not yet been released in a final stable version. While a patch is included in `2.14.0-rc1`, this is a release candidate and not the final `2.14.0` release. Once the stable `2.14.0` version is published, we will update the library. In the interim, this vulnerability is not exploitable in our environment because it requires local attacker access, and this container does not allow local user access.
- **Example 2 - False Positive**: The scan tools are using "netty" to flag reactor-netty-http to have vulnerabilities. The actual vulnerabilities are with older versions of netty libraries from https://github.com/netty/netty. Reactor projects are from https://github.com/reactor/reactor-netty. reactor-netty-http and other Reactor libraries do include the libraries from netty project, but they are including version 4.1.82. Final which does not have any vulnerabilities mentioned in the CVE. reactor-netty-http with version 1.0.23 is not the same as netty-codec-http 4.1.44, which has the vulnerability.
- **Example 3 - Pending Resolution**: There is a fixed version but it has not yet been pushed to the main release. Red Hat states this issue affects RHEL 9. Our application does not use the affected package for any part of the application but is a required dependency of package. Affected package will be updated when a fix is released.

Justification and exception handling

When a vulnerability cannot be immediately fixed, proper documentation is a best practice. Be sure to select a justification option in Findings's dropdown.

- **Formal Risk Acceptance**: If a CVE must remain, a formal **risk acceptance** should be documented, requiring approval from security within the Game Warden application.
- **Compensating and Mitigation Controls**: Justify the risk acceptance by detailing multiple **compensating and mitigation controls**. Examples include:
  - The container is running in a **secure network segment** with strict egress rules and running as a non-root user with limited capabilities mitigating the impact of the vulnerability.
  - The vulnerability is in a code path or binary that is **never executed** at runtime and the package is only installed as a required dependency.

FAQs

What is the expectation for Medium and Low vulnerabilities? Can they wait until a future release?

If you are pursuing a Deployment Passport, all CVEs—regardless of severity—must be resolved. For CVEs surfaced during routine scanning, the Findings page will display the expected resolution timeline according to Game Warden's Acceptance Baseline Criteria.

What is the timeline for addressing or justifying vulnerabilities found during continuous scanning for an already deployed container image?

The Findings page provides remediation and mitigation timelines based on Game Warden's Acceptance Baseline Criteria (Table A).

What should we do if there is a Critical or High vulnerability, and no fix is available?

All vulnerabilities—Critical, High, Medium, or Low—without fixes must be investigated to identify possible remediation options. Guidance on mitigation, remediation, and example statements can be found in Game

Warden's Acceptance Baseline Criteria.

What if we can't upgrade a 3rd-party package that has security findings?

You must justify the finding by explaining why the package is required and noting if/when it will be updated. Reference Justification and Exception Handling section for specific details.

Common Vulnerabilities and Exposures 101

Vulnerabilities are weaknesses in software that attackers can exploit to gain unauthorized access to systems or data. As software and threat landscapes evolve, new vulnerabilities are discovered frequently by security researchers.

This guide explains the role of Common Vulnerabilities and Exposures (CVEs) in secure development, particularly in the context of Game Warden and Department of Defense (DoD) environments. It also describes how Game Warden streamlines CVE management to support compliance with Authority to Operate (ATO) and software authorization requirements to enable faster, more secure deployment to production.

CVEs in DoD environments

CVEs are critical to manage within DoD systems for several reasons:

- **Awareness**: CVEs provide a standardized way to identify and track known software vulnerabilities.
- **Risk prioritization**: CVEs allow DoD teams to focus on the most serious vulnerabilities first.
- **System compatibility**: Visibility into CVEs helps ensure interconnected systems remain secure.
- **Regulatory compliance**: CVEs support alignment with required cybersecurity frameworks.
- **Risk mitigation**: Structured CVE tracking improves risk management across mission-critical systems.
- **Collaboration**: CVEs are globally recognized, supporting coordination between DoD, industry, and allies.

Reducing CVEs is a security best practice—and a strict requirement when deploying to DoD environments.

> **Info**
>
> CVEs are scored using the Common Vulnerability Scoring System (CVSS) and listed in a global directory maintained by MITRE. CVEs are analyzed by NIST and published on several public registries.

CVEs and Game Warden

During onboarding, Game Warden scans your application containers for CVEs. These results are accessible through Findings, a CVE management tool built into the Game Warden web app. Use it to:

- Review CVE details and severity levels
- Apply remediation or submit justifications
- Request a security review within the platform

After deployment, Game Warden performs scans to ensure your containers remain secure and compliant with the Acceptance Baseline Criteria. Any new vulnerabilities must be resolved or justified within the required timelines.

> **Warning**
>
> Failure to remediate or justify vulnerabilities within the required SLAs may result in deployment delays or noncompliance status.

Findings overview

The Findings page provides an aggregated view of all CVEs detected across your Development (DEV), Staging (STG), and Production (PRD) environments. Use it to view, resolve, or justify CVEs, address Anchore compliance results, and review security team responses to your proposed resolutions.

Free scanning tools: Trivy and Anchore

Game Warden's scanning pipelines use enterprise versions of Anchore and Trivy to analyze containers. You can run free versions of these tools locally to preview your CVE posture before submission.

- Anchore defines policy rules aligned with NIST 800-53 and DoD standards.
- Trivy offers fast, lightweight container scanning.

  Anchore compliance justifications

  Anchore also detects DoD compliance issues based on NIST 800-53 policies. These findings are treated similarly to CVEs and must be resolved or justified.

  Severity levels:

  - **Go** – OK to proceed (Low severity)
  - **Warn** – Warning (Medium severity)
  - **Stop** – Critical finding; blocks deployment (High severity)

  Warning

  Second Front requires all vulnerabilities to be addressed unless a valid CVE statement and justification are provided.

Managing CVEs post-deployment

Applications deployed to staging or production are scanned monthly. All new vulnerabilities must be addressed within the timelines specified in Tables A and B of the Acceptance Baseline Criteria.

Note

- If remediation isn't possible for a critical/high vulnerability, a CVE justification and mitigation is required.
- As the threat landscape evolves, new vulnerabilities may surface. It is essential that customers continuously update their applications and container dependencies.

Docker Health Checks

The Defense Information Systems Agency (DISA) requires a Docker health check to monitor the health of applications running inside containers. This check validates that services are running and responsive, ensuring continued availability.

Game Warden recommends implementing a `healthcheck` command directly in your container image to meet this requirement.

As an alternative, you may use a Kubernetes Liveness Probe in place of a Docker `healthcheck`—but this must be justified in Findings to remain compliant.

DISA specification

From Container Image Creation and Deployment Guide (Section 2.6):

> Ensuring a container service is still executing and handling workloads is necessary to validate service availability. Adding the health check instruction, HEALTHCHECK, within Docker, to the container image, provides a mechanism for the container platform to periodically check the running container instance against that instruction to ensure the instance is still working. Based on the reported health status, the container platform can then exit a non-working container and instantiate a new one bringing the service back to an available state. Short lived containers that do not require a health check can be submitted for a waiver.

IA Control: SC-5 **CCI:** CCI-002385

What is a Docker health check command?

The Docker `healthcheck` command is embedded in your container image and runs at set intervals to determine if the container is healthy. If the check fails, the container is marked **unhealthy**, and depending on configuration, it may be restarted.

Important

- The Department of Defense (DoD) requires a health check even though newer versions of Docker do not enforce it.
- In Game Warden, missing health checks may appear as `Healthcheck not found` in scan results due to Anchore compliance scanning.
- Iron Bank base images **do not** include a `healthcheck` by default.

Health check options in Game Warden

You can meet DISA's requirement in two ways:

1. **Docker `healthcheck` (recommended)** - Add the `heathcheck` instruction directly to your Dockerfile.
2. **Kubernetes liveness probe (acceptable alternative)** - If you use a liveness probe instead, document this in Findings:
   - In **Vulnerability** dropdown, select **Policy N/A**.
   - In **Justification**, enter: `Using Kubernetes liveness probe`.

Common implementation methods

Dockerfile with cURL

Below is a Dockerfile that includes a `healthcheck` command using the `curl` command.

```
FROM python:3.8-slim
WORKDIR /app
COPY . /app
RUN pip install --no-cache-dir -r requirements.txt

HEALTHCHECK --interval=10s --timeout=5s --start-period=15s \
  CMD curl --fail localhost:8080/health || exit 1

EXPOSE 8080
CMD ["python", "app.py"]
```

Run this `curl` command every 10 seconds to verify `localhost:8000/health`. Mark the container unhealthy if the command fails.

Dockerfile with wget

You can use `wget` to add the health check to the Dockerfile:

```
FROM alpine:3.12
RUN apk update && apk add --no-cache wget

HEALTHCHECK --interval=10s --timeout=5s --start-period=15s \
   CMD wget --spider http://localhost || exit 1
```

This method allows you to check if a URL is accessible without downloading content (`--spider`).

Kubernetes liveness probe with cURL

A Kubernetes liveness probe checks the health of a container and determines whether or not it should be restarted.

To use a liveness probe in Kubernetes, you must define a pod that uses the image then configure a liveness probe that runs the `healthcheck` command.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-app-pod
spec:
  containers:
  - name: my-app-container
    image: my-app-image
    livenessProbe:
      exec:
        command:
        - curl
        - localhost:8080/health
      initialDelaySeconds: 15
      periodSeconds: 10
```

Kubernetes liveness probe with wget

Below is an example of a liveness probe using `wget`:

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-app
spec:
  containers:
  - name: my-container
    image: my-image
    ports:
    -  containerPort: 80
    livenessProbe:
      httpGet:
        path: /
        port: 80
      initialDelaySeconds: 5
      periodSeconds: 5
    readinessProbe:
      httpGet:
        path: /
        port: 80
      initialDelaySeconds: 10
      periodSeconds: 5
      failureThreshold: 3
```

Java Distroless containers

Distroless containers are minimal and lack shells or common utilities such as `curl` or `wget`.

Doing health checks in these containers require different approaches:

1. **Mini Web Server in a JAR** – Serve a `/health` endpoint and run a Java-based probe.
2. **wget from BusyBox** – Use a multi-stage build to copy wget into the container before adding a health check.

Mini Web Server in a JAR This method performs a health check by running a small, lightweight web server inside your container.

Step 1 - Create a mini web server: Similar to a standard web server, it serves content over HTTP. This server will respond to requests that indicate the container's health.

Step 2 - Bundle the mini web server into a Java ARchive (JAR): A JAR is similar to a `.zip` file—it compresses and packages multiple Java classes and resources into a single file. This approach allows you to reuse common features across multiple applications without rewriting code.

For example, you can add health check logic to a Java class, include it in a JAR, and run it in any containerized application.

Step 3 - Build the container: Run the following commands to compile the classes, create the JAR files, and build your Docker image:

```
javac TheComSunNetHttpServer.java
jar -c -e TheComSunNetHttpServer -v -f TheComSunNetHttpServer.jar TheComSunNetHttpServer.class
javac HealthCheck.java
jar -c -e HealthCheck -v -f HealthCheck.jar HealthCheck.class
docker build -f Dockerfile-java . -t java-healthcheck
```

Step 4 - Start the container: Launch the container in the background, listening on port 8080:

```
docker run -d -p 8080:8080 java-healthcheck sleep 2 curl http://localhost:8080
```

Step 5 - Verify the health check: Confirm the container is running and healthy:

```
docker ps
```

Example output:

```
docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 009a72a63438 java-healthcheck "/usr/bi
```

wget from BusyBox In this example, you'll use a **multistage Docker build** to copy `wget` from BusyBox before running it as part of your health check.

- **Why multistage builds?** They allow you to separate the Docker build process into multiple steps, keeping the final image smaller and more secure.
- **Why BusyBox?** Known as the "Swiss Army knife" for Unix/Linux, BusyBox packages many common Linux utilities—such as `cp`, `rm`, and `mkdir`—into a single lightweight executable. These tools are not available by default in distroless containers, making BusyBox a useful source for adding them.

Step 1 - Build the container: Run the following command:

```
javac TheComSunNetHttpServer.java jar -c -e TheComSunNetHttpServer -v -f TheComSunNetHttpServer.jar The
```

Step 2 - Start the container: Launch the container in the background, listening on port 8080:

```
docker run -d -p 8080 wget-healthcheck sleep 2 curl http://localhost:8080
```

Step 3 - Verify the health check: Confirm the container is running and healthy:

```
docker ps Example output:  docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 009a72a63438
```

Java-based health check in a Distroless container

The Java-based health check for Docker method demonstrates how to perform a health check without relying on additional utilities such as `curl` or `wget`.

This approach uses:

- **Java 11 single-file program** — allows you to run a .java file directly without compiling with `javac`.
- **Java HttpClient** — sends HTTP requests and validates responses.

```java
import java.io.IOException;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse.BodyHandlers;

public class HealthCheck {
  public static void main(String[] args) throws InterruptedException, IOException {
    var client = HttpClient.newHttpClient();
    var request = HttpRequest.newBuilder()
        .uri(URI.create("http://localhost:8080/actuator/health"))
        .header("accept", "application/json")
        .build();
    var response = client.send(request, BodyHandlers.ofString());

    if (response.statusCode() != 200 || !response.body().contains("UP")) {
      throw new RuntimeException("Healthcheck failed");
    }
  }
}
```

How it works:

1. Sends a `GET` request to `/actuator/health`.
2. Checks if:
   - The HTTP status code is **200 (OK)**.
   - The response body contains "UP", indicating a healthy service.
3. Throws a `RuntimeException` if either check fails; otherwise, it exits silently.

Docker image best practices

- Expose a health endpoint that returns 200 only when the app is actually ready (connections open, migrations done, deps reachable). This enables safe autoscaling and zero(ish)-downtime deploys.

  Example: App exposes `/healthz` and returns 200 only when ready

  ```
  HEALTHCHECK --interval=30s --timeout=3s --start-period=10s --retries=3 \
  CMD curl -fsS http://localhost:8080/healthz || exit 1
  ```

- Never run processes as root inside containers.

  Example: Create unprivileged user

  ```
  RUN addgroup -S app && adduser -S app -G app
  USER app
  ```

- Each `RUN`, `COPY`, `ADD` makes a layer. Combine related commands and clean caches to shrink images.
- Exclude anything not needed at runtime/build (node_modules, tests, .git, IDE files, secrets).
- Use immutable, descriptive tags (e.g., 1.4.3, 2025.10.01-sha.abcd123). This makes rollbacks/audits reliable.
- Never bake API keys, tokens, or certs into layers or files. Inject via env vars/secret stores at deploy. CI scanners (e.g., Trivy) should fail the build if secrets are detected.
- Build artifacts in one stage; ship only what's needed in the final image.
- Choose distroless/Alpine/Chainguard images when possible; avoid large bases unless you truly need them. If you only need to serve static files or a single binary, start from scratch (0-byte base). Smaller images lead to faster pulls and a smaller attack surface.
- If you only need to serve static files or a single binary, start from scratch (0-byte base).

Security Incident Report Guide

This guide explains how customers should report security incidents for applications deployed via Game Warden. Prompt, accurate reporting helps Second Front (2F) assess, contain, and remediate risks quickly while meeting DoD security requirements.

When to report

Incidents such as suspected unauthorized access, attempted breach, exposure of sensitive data, malware, or other malicious activity.

Common incident types:

Incident Type	Description
Data Spillage	Confidential data is released into an untrusted environment.
Data Breach	Unauthorized access, disclosure, or theft of sensitive data.
Malware Infection	Installation and execution of malicious software.
Denial of Service (DoS)	Intentional disruption of services.
Unauthorized Access	Accessing systems or data without authorization.
Insider Threat	Threats originating from employees or trusted individuals.
Phishing	Attempts to gain sensitive information via deceptive communication that results in actual compromise.
Ransomware	Malware that encrypts data for extortion.

Important

- If your event falls under one of these categories, you must **immediately** report the incident via email to the Second Front security team.
- If you're not sure which category applies, report the event anyway. Our team will triage and route appropriately.

How to report

Step 1 - Notify 2F

Send an email to security@secondfront.com.

Step 2 - Provide the following information

In your email, include the following information:

- **Company name**
- **Brief description of the incident** (Do not include CUI, PII, or classified information)
- **Type of incident** (Refer to the list above)
- **Affected application or service**
- **Current impact** (I.e., downtime, data exposure, degraded service)
- **Was sensitive data exposed?** (Yes/No)
- **Contact information** so the Security Team can follow up with you quickly

Step 3 - What happens next

Once your email is received, the 2F Security will:

1. Review the information provided.
2. Assess the incident and initiate incident response (IR) protocols.
3. Execute required containment, recovery, and/or mitigation steps.
4. Provide status updates and next steps via Slack or email.

Responsibility model

- **Platform-level response (Game Warden infrastructure)**: Managed by 2F.
- **Application-level response (your deployed apps)**: Managed by your team.
- Some IR controls are partially inheritable, but *primary responsibility for application-level security and recovery rests with the customer.*

Disaster Recovery Guide

This guide explains how customers should report service disruptions for applications deployed via Game Warden. Prompt, accurate reporting helps Second Front (2F) respond quickly and effectively.

When to report

Operational disruptions such as outages, data loss, or critical service degradation that may require recovery procedures.

Common event types:

Event Type	Description
Service Unavailability	The application is completely inaccessible to end-users (e.g., 503 errors, timeouts) due to underlying platform or infrastructure issues.
Data Loss or Corruption	Persistent data (databases, object storage) is accidentally deleted, corrupted, or rolled back to an incorrect state.
Cloud Region/Zone Outage	A major failure of the underlying hyperscaler IaaS provider (e.g., AWS/Azure) affecting the hosting region or availability zone.
Platform Component Failure	Essential PaaS services required for the application to run (e.g., Ingress controller, DNS, authentication services) are non-functional.
Malicious Data Destruction	A security compromise (such as ransomware or insider threat) that results in data encryption or deletion, requiring restoration from backups.
Resource Exhaustion	The application or platform runs out of critical resources (storage, compute quotas) causing a crash that requires intervention to restore service.
Failed Deployment/Upgrade	A platform or application update results in a catastrophic failure where standard rollback procedures fail, requiring a full recovery.

> **Important**
>
> If you're not sure which category applies, report the event anyway. Our team will triage and route appropriately.

How to report

Notify 2F

The below listed options are in priority order to ensure the most rapid response. If the report is not acknowledged or the recommended resource is not available, utilize another option to ensure rapid response.

1. Report via your customer Slack channel.
2. Create an Outage Ticket in the Game Warden application. See Submit Support Tickets for more details.
3. Send an email to devops@secondfront.com.

When reporting a disruption, please include the following details in your communication to 2F:

- Company name
- Brief description of the disruption
- Type of event (Refer to the list above)
- Current impact (i.e., downtime, degraded service)
- Affected application or service
- Contact information so the Security Team can follow up with you quickly

What happens next

Once your report is received, the 2F team will:

1. Review the information provided.
2. Assess the event and initiate disaster recovery (DR) protocols.
3. Execute required recovery and/or mitigation steps.
4. Provide status updates and next steps via Slack or email.

Responsibility model

- **Platform-level response (Game Warden infrastructure)**: Managed by 2F.
- **Application-level response (your deployed apps)**: Managed by your team.
- Some DR controls are partially inheritable, but *primary responsibility for application-level security and recovery rests with the customer.*

Integrate Amazon Bedrock into Game Warden-Deployed Applications

Amazon Bedrock is a fully managed AWS service that provides access to leading foundational models through a unified API. It enables you to incorporate generative AI capabilities into your applications without the need to develop or maintain underlying machine learning infrastructure.

This guide describes when to use Amazon Bedrock on Game Warden, common use cases, service requirements and restrictions, integration steps, and compliance and monitoring requirements. Following these recommended practices will ensure a secure integration of Amazon Bedrock into applications deployed on the Game Warden platform.

> **New CtF/Software Approval required**
>
> The addition of Bedrock is considered a significant change, and a new Certificate to Field (CtF)/Software Approval is required. See Significant Software Changes and Authorization Requirements in Game Warden for more information.

When to use Amazon Bedrock on Game Warden

You can use Amazon Bedrock on Game Warden to send prompts via the Amazon Bedrock API to supported models and receive generated text, code, or summaries in real-time. Only synchronous, on-demand model invocations are supported at this time; batch processing, model customization, and advanced orchestration are currently not supported. Enabling these advanced features is on the Product Roadmap.

Use Amazon Bedrock on Game Warden when:

- The application use-case can be supported by a dedicated instance of a cloud-provider-managed Large Language Model (LLM)
- Access to foundational models such as Claude 3 Haiku or Titan Text Embeddings V2 are required
- A consistent API is needed for interfacing with multiple model providers
- Seamless integration with other AWS services (e.g., S3) is needed
- Serverless inference capabilities are required without the overhead of managing machine learning infrastructure

Common use cases include:

- Building enterprise-grade chatbots or AI copilots
- Implementing document and data summarization, content extraction, and/ or question answering systems
- Enabling natural language and semantic search, and automated content generation
- Language translation

> **Amazon Bedrock Agents**
>
> - At this time, Bedrock agents are not authorized for AFWERX environments.
> - Bedrock Knowledge Bases are authorized and can be used in AFWERX environments.

AWS region support and Impact Level restrictions

Applications using Amazon Bedrock on Game Warden at Impact Levels 4 (IL4) and 5 (IL5) are limited, at this time, to the foundational models available in AWS GovCloud (US- and US-West).

The table below outlines Bedrock support across different Impact Levels:

Impact Level	Bedrock Support	Details
IL2	Supported	Invoke on-demand models available in commercial AWS regions.
IL4	Conditionally Supported	Invoke on-demand models available in GovCloud (US-East and US-West). Refer to Foundational models by AWS Region for more information.
IL5	Conditionally Supported	Invoke on-demand models available in GovCloud (US-East and US-West). Refer to Foundational models by AWS Region for more information.
IL6	Not Supported	Amazon Bedrock is not available for classified (IL6) environments.

Tip

Not all Amazon Bedrock foundational models or features are available in AWS GovCloud. Verify model support before finalizing integration plans.

Integration steps

Step 1 - Define your use case

Before integrating Amazon Bedrock on Game Warden, determine how your application will use the service:

- **Which foundation models will you call?** For deployments in AWS GovCloud (US-East and US-West), visit Model support by AWS Region in Amazon Bedrock to check which models are currently supported.
- **What is the justification for Amazon Bedrock integration?**
- **What data will be sent to and returned from Amazon Bedrock?**
- **Will any Controlled Unclassified Information (CUI) or sensitive data be processed?**

Clear definition of these parameters will inform and guide Second Front's evaluation of your use of artificial intelligence. Each deployment of Amazon Bedrock on Game Warden ensures LLM interactions, data, and usage metrics are logically and often physically separated at the AWS account and service level, aligning with enhanced security and regulatory requirements of serving national security missions.

Step 2 - Create a ticket in the Game Warden app

In the ticket, include the following information:

- Specify the AI model(s) intended for use.
- Confirm that the AI Attestation section in the Body of Evidence (BoE) is complete.
- Provide the business justification for integrating with Amazon Bedrock.
- Describe the data that will be sent to and returned from Amazon Bedrock.
- Indicate whether any Controlled Unclassified Information (CUI) or other sensitive data will be processed.

Once approved, 2F Engineering will configure Bedrock.

Note: If Bedrock is being configured for the first time, a new CtF/Software Approval is required.

Step 3 - Configure IAM roles and network access

Ensure your application can securely access Bedrock:

- Configure **egress routing** in your Kubernetes workload for outbound access to Bedrock endpoints.
- Base on your selected region, set the **AWS region** to `us-gov-east-1` or `us-gov-west-1` for IL4 and IL5 environments.

Step 4 - Connect to the Bedrock API

Review AWS documentation for getting started with the Bedrock API. Use the AWS SDK for Bedrock to integrate the service into your Game Warden hosted application. SDKs are available for:

- Python (Boto3)
- JavaScript
- Java
- Other languages via the AWS CLI or REST API

Commonly used open source libraries for generative AI applications, such as LangChain and LangGraph, also have integrations and abstractions for Amazon Bedrock.

Implement appropriate **retry logic**, **rate limiting**, and **output validation**-especially for workloads processing unstructured or dynamic input.

Upgrade AI model

Follow the steps below when you need to upgrade or switch AI models:

1. Update your AI Attestation for the new model.
2. Submit a Support Ticket in the Game Warden app for a configuration change request (see Integration steps, Step 2).

Once approved, 2F Engineering will update your Bedrock configuration to reference the new model.

Helpful resources

- Amazon Bedrock Documentation
- Bedrock IAM Permissions Guide
- AWS SDK Examples for Bedrock

Questions?

If you're unsure about your Bedrock integration or deployment impact level, contact your Second Front implementation engineer.

Integrate Cohere Models into Game Warden-Deployed Applications

Cohere is a leading artificial intelligence company specializing in the development and deployment of large language models (LLMs) and foundational models designed for enterprise applications. Through a strategic partnership with Cohere, Second Front (2F) is able to deploy Cohere models on the Game Warden platform, empowering customers with cutting-edge AI capabilities.

Cohere's product suite delivers advanced natural language processing (NLP) and image processing solutions that enable enterprises to harness models capable of comprehending, generating, searching, and interacting with human language-and, increasingly, with images. These models support a wide range of use cases, including content creation, semantic search, conversational AI, and knowledge management, all with robust enterprise-grade security and scalability.

The following Cohere models can be deployed on Game Warden:

- **Command** - Cohere's state-of-the-art family of generative models optimized for security and mission needs. With strong multilingual support, multi-modal capability, high performance, and advanced reasoning capabilities, the Command suit excels at retrieval-augmented generation (RAG), agentic workflows, and can process large context windows for complex tasks.
- **Embed** - Cohere's Embed models, including multi-lingual support, turn text and images into embeddings to enable semantic retrieval in search systems, RAG architectures, and agentic applications - powering answers, insights, and action across the enterprise.
- **Rerank** - Rerank passes only the most relevant documents into your RAG pipeline and agentic workflows - reducing token use, minimizing latency, and boosting accuracy.

This guide describes when to use Cohere models on Game Warden, requirements and restrictions, integration steps, and compliance and monitoring requirements. Following these recommended practices will ensure a secure integration of Cohere into applications deployed on the Game Warden platform.

> **New CtF/Software Approval required**
>
> The addition of Cohere is considered a significant change, and a new Certificate to Field (CtF)/Software Approval is required. See Significant Software Changes and Authorization Requirements in Game Warden for more information.

When to use Cohere on Game Warden

Deploying Cohere's models within Game Warden on a cloud service–provided Kubernetes cluster, or on-premise, to include classified environments, enables Game Warden users to securely integrate generative AI capabilities into their applications while maintaining full control over their data and systems. This approach addresses privacy, compliance, and operational requirements that are often critical for mission use cases.

Key reasons to deploy within a cloud service–provided Kubernetes cluster, or on-premise:

- **Data Privacy and Security**: Sensitive or regulated data never leaves the Game Warden secure environment, reducing risk and exposure to third-party cloud providers.
- **Compliance**: Meet strict industry and government compliance standards (such as HIPAA, GDPR, FINRA, etc.), which may prohibit certain data from leaving controlled infrastructure.
- **Customization and Integration**: Fine-tune models, integrate with proprietary datasets, and tightly couple AI with existing local workflows and databases.
- **Performance and Latency**: Reduce latency and ensure reliable, high-speed response times, particularly important for mission-critical applications.
- **Cost Predictability**: Provide more predictable operational costs, especially for high-volume usage, avoiding variable cloud charges.

- **Isolation**: Full isolation from the public internet and shared cloud resources, meeting the needs of organizations with highly sensitive or classified workloads.
- **Control**: Retain complete control over access, updates, and AI governance policies within their infrastructure.

Deployment requirements

To ensure optimal performance and reliable operation of Cohere models on Game Warden, deployments must meet the following support and infrastructure requirements. In addition to below information, please see Cohere documentation on Deploying Cohere Models in Private Environments. Your 2F Technical Implementation Manager will work closely with you to ensure requirements are met for:

Hardware Requirements

- **GPU Acceleration**: Cohere's models require state-of-the-art GPU hardware for production-grade inference and training. NVIDIA A100 GPUs (or equivalent, such as H100) are recommended to support performance and scalability standards.

Self-managed Kubernetes or On-premise

- **On-Premise & Private Cloud/Kubernetes**: For on-premise or self-managed Kubernetes clusters (running on any cloud or private infrastructure), direct access to supported NVIDIA A100+ GPU resources is required. Ensure hardware compatibility and sufficient resource allocation prior to deployment.
- **Support Scope**: Technical support is available for deployments that meet these hardware and infrastructure standards. Deployments on unsupported hardware, or in regions without guaranteed GPU availability, may not be eligible for full support or performance guarantees.

Cloud Platform Requirements

> **Note**
>
> Limited GPU availability may adversely impact deployments. Deployments in unsupported or under-resourced regions may experience delays or lack of support.

Amazon Web Services (AWS)

- **Required Instance Types**: Deployments must use P4d or P4de (P4DN) EC2 instance types, which provide the necessary NVIDIA A100 GPUs.
- **Regional Availability**: P4d/P4de instances are primarily available in select regions, with the highest availability in US West (Oregon). Availability in other regions is limited; please work with your Game Warden Technical Implementation Manager to verify instance stock before planning your deployment.

Cloud Platform (GCP)

- **Required Machine Types**: Use A2 (A2-highgpu or A2-megagpu) instances equipped with NVIDIA A100 GPUs.
- **Regional Availability**: Supported A2 machine types are available in limited GCP regions. us-central1 (Iowa) and europe-west4 (Netherlands) typically have the best availability, but verify current capacity with GCP before deployment.

Microsoft Azure (Azure)

- **Required VM Types**: Use ND A100 v4-series virtual machines, which feature NVIDIA A100 GPUs.

- **Regional Availability**: ND A100 v4-series VMs currently have the highest availability in select regions such as East US, South Central US, and select European data centers. Please consult Azure's product documentation or portal to confirm current regional availability.

Integration steps

Step 1 - Define your use case

Before integrating Cohere models on Game Warden, determine how your application will use the Cohere:

- **Which foundation models will you call?**
- **What is the justification for Cohere integration?**
- **What data will be sent to and returned from Cohere?**
- **Will any Controlled Unclassified Information (CUI) or sensitive data be processed?**

Clear definition of these parameters will inform and guide 2F's evaluation of your use of artificial intelligence. Each deployment of Cohere on Game Warden ensures LLM interactions, data, and usage metrics are logically and often physically separated at the cloud service provider account level, or at the hardware level for on-premise deployments, aligning with enhanced security and regulatory requirements of serving national security missions.

Step 2 - Create a ticket in the Game Warden app

In the ticket, include the following information:

- Specify the AI model(s) intended for use.
- Confirm that the AI Attestation section in the Body of Evidence (BoE) is complete.
- Provide the business justification for integrating with Cohere.
- Describe the data that will be sent to and returned from Cohere.
- Indicate whether any Controlled Unclassified Information (CUI) or other sensitive data will be processed.

Once approved, 2F Engineering will configure Cohere for your application.

Note: If Cohere is being configured for the first time, a new CtF/Software Approval is required.

Step 3 - Integrate Cohere Model(s) API to your application

Review Cohere documentation for getting started with the Cohere models' API and review Cohere Cookbooks for sample code for a range of use-cases. Integrate needed API interactions into your application.

The "single-serving-cohere-all" container is deployed with an accompanying service that listens on TCP port 8080 and forwards incoming traffic to port 8080 of the deployed "single-serving-cohere-all" pod.

Commonly used open source libraries for generative AI applications, such as LangChain and LangGraph, also have integrations and abstractions for Cohere models.

Implement appropriate **retry logic**, **rate limiting**, and **output validation**, especially for workloads processing unstructured or dynamic input.

Upgrade AI model

Follow the steps below when you need to upgrade or switch AI models:

1. Update your AI Attestation for the new model.
2. Submit a Support Ticket in the Game Warden app for a configuration change request (see Integration steps, Step 2).

Once approved, 2F Engineering will update your Cohere configuration to reference the new model.

Helpful resources

- Cohere
- Cohere Command Models
- Cohere Embed Models
- Cohere Rerank Model
- Deploying Cohere Models in Private Environments
- Cohere Cookbooks

Questions?

If you're unsure about your Cohere integration or deployment impact level, contact your Second Front implementation engineer.