Arcturus Application Guide

Purpose Statement 1
Arcturus 2
Configuration 3
Cryptography 3
AWS 3
Gitlab 4
Access Controls 6
Authentication with CAC or ECA 10
Account Management 14
Session Management 15
Code Compliances 19
DoD Ports and Protocols Database 19
Customer Deployment 20
GUI 20
IPv6 22
Attack & Vulnerability Prevention 23
Arcturus 23
AWS 23
Gitlab 24
Clean Up 24
Verification 26
Decommission (End of Life) 26
Alerts 27
Logging 29
Auditing 31
Redundancy 33
Documentation 33
Extra 35
Annual Assessments 35
Personnel 35

Purpose Statement

Cloud One (C1) has an Amazon Web Services Impact Level 5 Authority to Operate (AWS IL5 ATO). As a new Software as a Service (SaaS) provider to C1, Clarity must prove compliance to various policies for the Arcturus tool set. The Application Security and Development (ASD) Security Technical Implementation Guide (STIG) is a set of requirements designed to be applied to all enterprise applications connected via the network and is used as a tool to improve the

security of Department of Defense (DOD) information systems. This document details and proves technical machinations referred to in the checklist (.ckl) provided.

Arcturus

Arcturus is a set of tools for hosting application code, testing and scanning applications, and deploying applications to the cloud within CloudOne. Arcturus works with customers to onboard them into Gitlab with project group creation and initial Jarvis pipeline configurations. The Jarvis pipeline builds, tests, scans, and deploys the application to the CloudOne cloud automatically.

Arcturus includes the CI/CD pipeline, runners, the three separate stages of build, test, and scan, and cloud delivery. The CI/CD pipelines are Continuous Integration (CI), Continuous Delivery (CD), and Continuous Deployment (CD) pipelines which automate phases of software development, from the build, to testing and scanning, through to release and deployment. Runners are machines, or VMs, configured and registered with GitLab which are provisioned to run the CI/CD jobs.

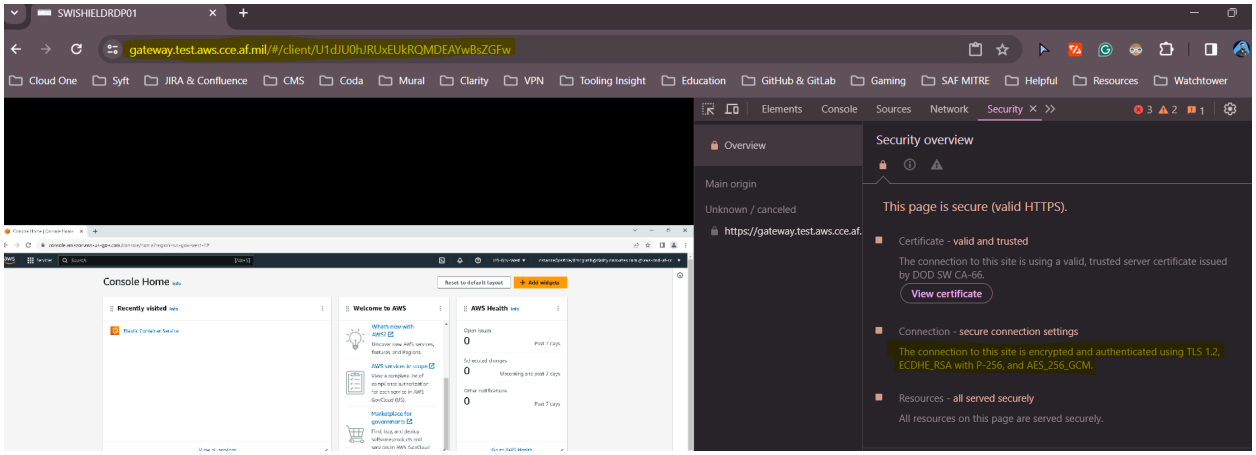
First, code is pushed to a GitLab repository, the build stage is triggered and code is compiled and ready for the next stage. During the test stage, all tests are run, including unit tests, UI tests, and functional tests. The scan stage includes scanning jobs that are used to help perform quality and security checks for an app. Once the test stage completes successfully, the delivery stage packages the compiled code, along with all its dependencies, into a binary artifact available for customer download. For iOS, an .ipa file is generated and for Android an .apk file is generated. The Arcturus pipeline offers functionality to automate delivery of binaries to target distribution platforms or app stores, such as Apple App Store, MAAS360, or Google Play Store. Cloud delivery is through C1 AWS.

Configuration

Cryptography

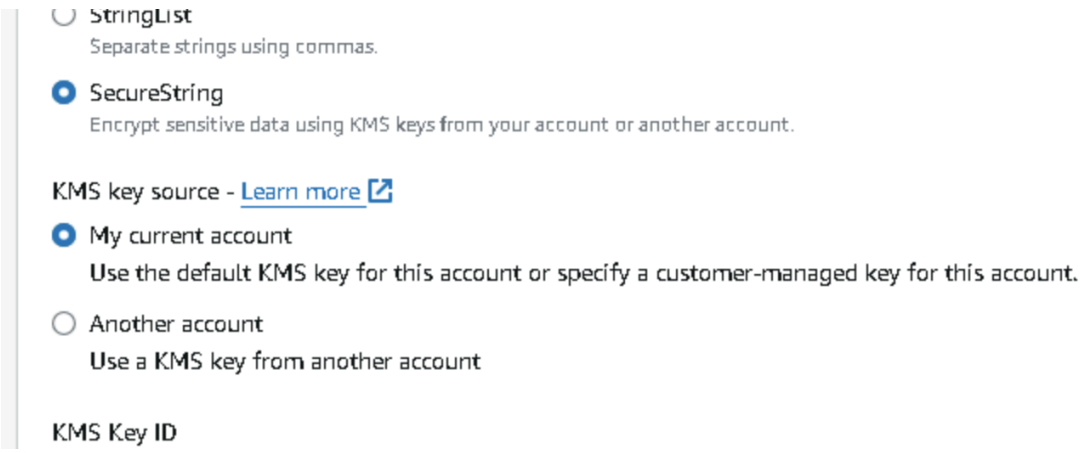
AWS

(V-222507, V-222589, V-222508, V-254803) Arcturus uses AWS Cloudwatch to cryptographically protect the integrity of audit information. Cloudwatch encrypts data at rest as well as end-to-end encryption of data in transit. (V-222597, V-222396, V-222397) AWS uses TLS to prevent unauthorized disclosure of information and hashing to detect changes of information during transmission.



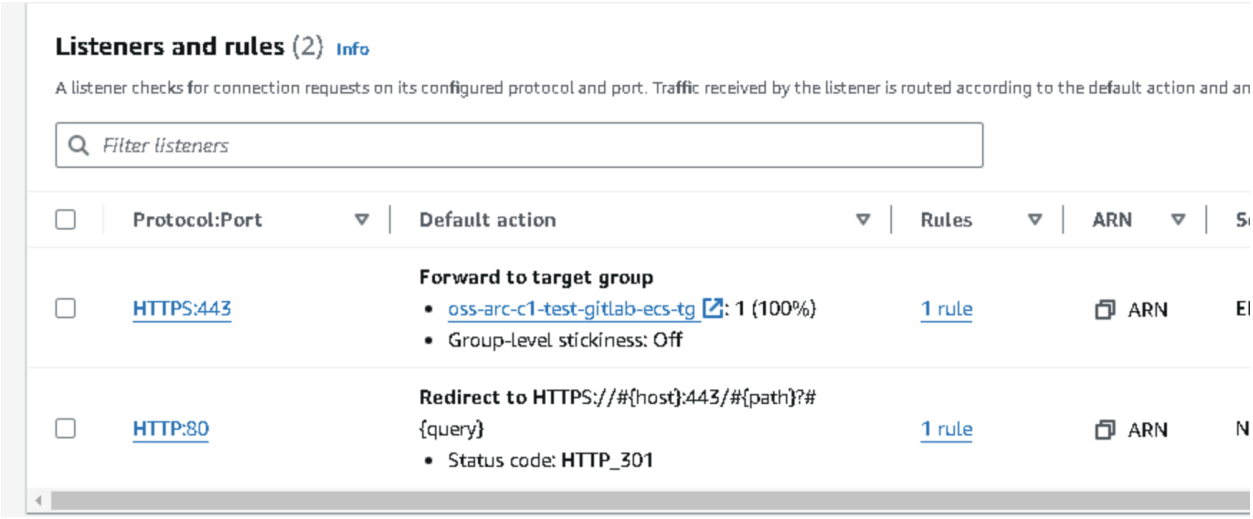
AWS TLS proof.

(V-222587, V-222588) Arcturus uses the AWS Secrets Manager and the AWS Systems Manager Parameter Store to prevent unauthorized modification of organization-defined information. AWS Secrets Manager uses envelope encryption (symmetric AES-256 encryption algorithm, asymmetric is RSA) to encrypt secrets in AWS Key Management Service (KMS). Parameter Store uses AWS KMS as well to encrypt and decrypt the parameter values of secure string parameters. This assures appropriate cryptography to protect stored DoD information. The screenshots below prove we are using AWS TLS and KMS.



Proof we are using KMS

We explicitly force HTTPS through the AWS load balancers.



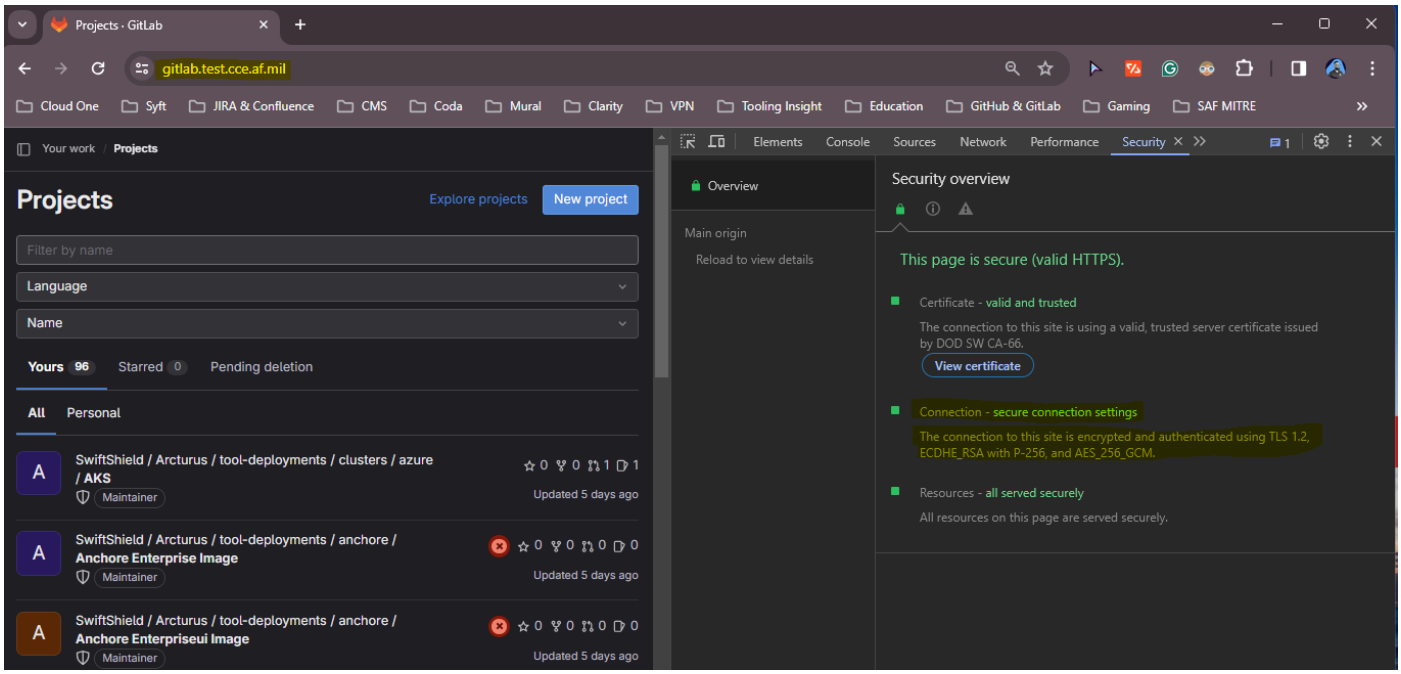
(V-222555, V-222570, V-222571, V-222572, V-222583) Arcturus uses AWS Govcloud, which operates using FIPS 140-2 validated cryptographic modules. The only cryptographic module Arcturus is using is AWS. This would be applied to any and all random number generators, encryption, key exchange, digital signature, and hash functionality through AWS. This ensures protection of unclassified information, and this pertains to cryptographic hashes and application component signatures. (V-222645) Application files are cryptographically hashed prior to deploying to DoD operational networks with the AWS S3 hashing, which checks for object integrity.

Gitlab

Arcturus uses Gitlab, which uses session cookies. (V-222575, V-222576) The Gitlab session cookies have the HTTPOnly flag and the secure flag set automatically. Please see the image below showing our Gitlab sessions have both the HttpOnly and Secure flags.

Name	Value	Domain	Path	Expires / Ma…	Size	HttpOnly	Secure	SameSite	Partition Key
arp_scroll_position	4389.5	.gitlab.test.cce.af.mil	/	2024-03-08…	25				
visitor_id	21c965ee-6bfb-48ac-8051-57c9bf84e4db	gitlab.test.cce.af.mil	/	2025-03-11…	46		✓	None	
event_filter	all	gitlab.test.cce.af.mil	/	Session	15		✓	None	
_gitlab_session	202bdf0dcf487322b5bff47d114a61fe	gitlab.test.cce.af.mil	/	Session	47	✓	✓	None	
preferred_language	en	gitlab.test.cce.af.mil	/	Session	20		✓	None	

(V-222534) Gitlab authenticates endpoint devices using mutual SSL/TLS. See screenshot below.



(V-222509, V-222596, V-222598, V-222599) Gitlab cryptographically hashes application files, using checksums, prior to deploying to DoD by hashing docker containers. Files are checked by Gitlab for any changes in checksum, or cryptographic hash value, validating and ensuring integrity. Audit tools are validated by checking the files in cryptographic hash value using tags which are appended at the end of images, calling checksums which correspond to the algorithm used for the integrity check.

Gitlab ensures confidentiality and integrity of information during reception, preparation for transmission, and after-transmitted protection through hashing and checking for changes in hashes. The Sonarqube and Anchore scanners also assist in checking data integrity. The tokens are sent during authentication, data transmission is handled, and integrity checked. See the image below for an example:

```
Search job log

1  Running with gitlab-runner 16.2.1 (674e0e29)
2    on ecs_runner_12 h6H6aspY, system ID: r_7c1AMYl0ybm7
3    feature flags: FF_ENABLE_BASH_EXIT_CODE_CHECK:true, FF_USE_NEW_BASH_EVAL_STRATEGY:true
4  Resolving secrets
5  Preparing the "docker" executor
6  Using Docker executor with image gitlab-registry.test.cce.af.mil/c1-swift-shield/arcturus/container-imag
7  Starting service gitlab-registry.test.cce.af.mil/c1-swift-shield/arcturus/container-images/build-docker-
8  Authenticating with credentials from job payload (GitLab Registry)
9  Pulling docker image gitlab-registry.test.cce.af.mil/c1-swift-shield/arcturus/container-images/build-doc
10 Using docker image sha256:519fae193386aa641d85aaa5f495ea67c49b8237112e55e6182c78300b785603 for gitlab-re
   eld/arcturus/container-images/build-docker-dind:v24.0.7-0 with digest gitlab-registry.test.cce.af.mil/c1
   ages/build-docker-dind@sha256:a01662725300d28c789773bf0552a31cc19573172615e2a5aac8ecec69cb820e ...
11 Waiting for services to be up and running (timeout 60 seconds)...
12 Authenticating with credentials from job payload (GitLab Registry)
13 Pulling docker image gitlab-registry.test.cce.af.mil/c1-swift-shield/arcturus/container-images/build-doc
14 Using docker image sha256:305f651285c0e2b7d13c39fca8620b0b9f48df1f5c8308488b1a3fa7e5e233c5 for gitlab-re
   eld/arcturus/container-images/build-docker:v24.0.7-0 with digest gitlab-registry.test.cce.af.mil/c1-swi
   build-docker@sha256:a0de6efcf8cc1c35c4c3dd2540f88915f0f2861a464a5d69b7509744ef55fdb6 ...
15 Preparing environment
```

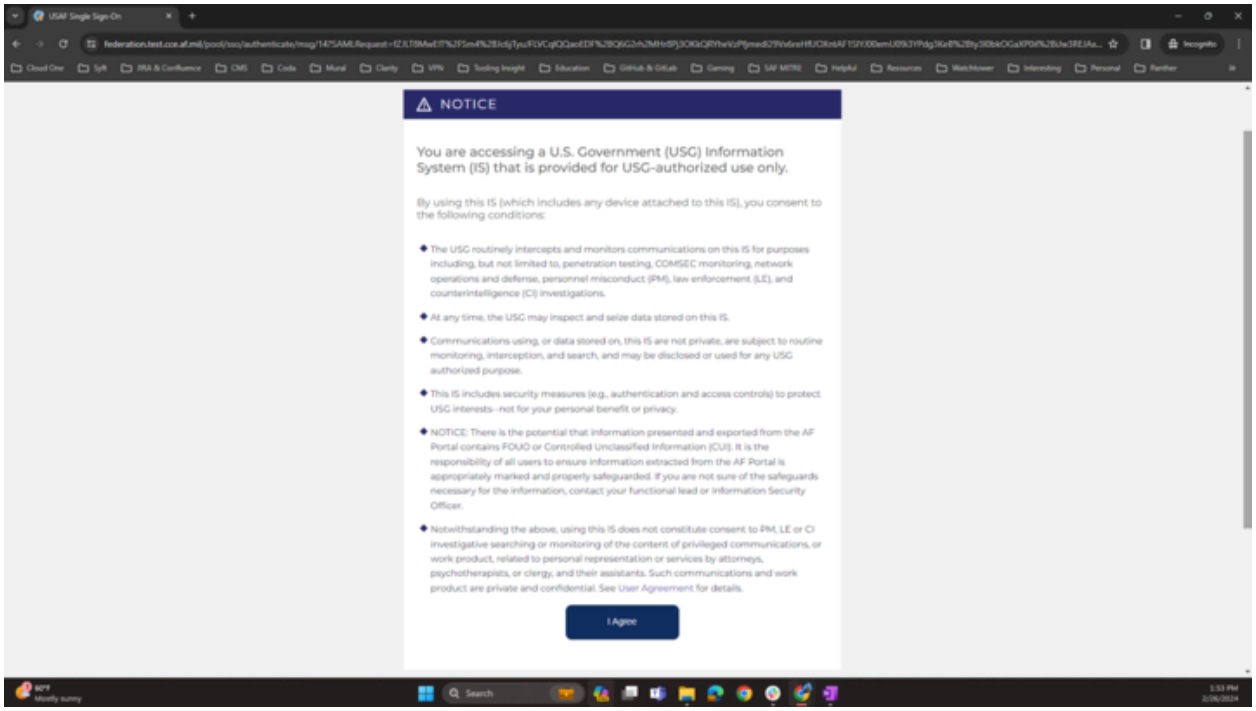
Access Controls

(V-222434, V-222435, V-222436) The DOD Notice and Consent Banner will appear as a splash screen when anyone first logs in to Gitlab using their CAC or ECA. This is retained until users acknowledge the usage conditions and take explicit actions to log on for further access. The banner must include the following:

> U. S. Government (USG) Information System (IS) - Use and Consent
>
> You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only.
>
> By using this IS (which includes any device attached to this IS), you consent to the following conditions:
>
> USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
> At any time, the USG may inspect and seize data stored on this IS.
> Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG-authorized purpose.
> This IS includes security measures (e.g., authentication and access controls) to protect USG interests--not for your personal benefit or privacy.

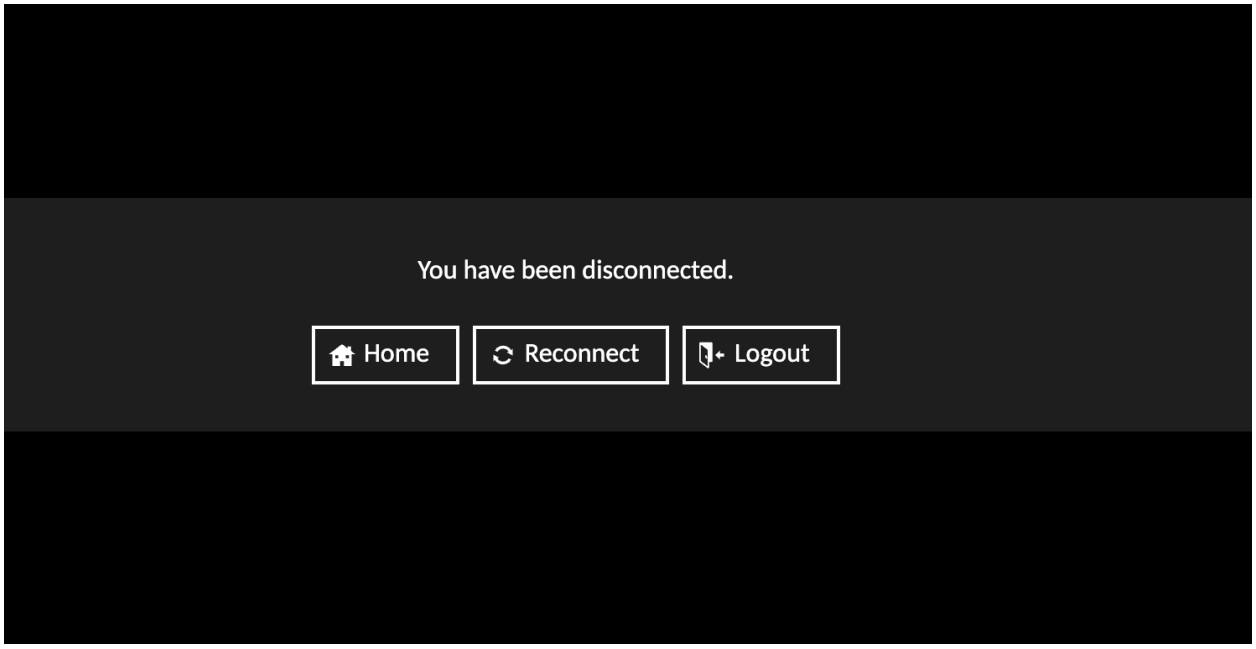
> Notwithstanding the above, using this IS does not constitute consent to PM, LE, or CI investigative searching or monitoring of the content of privileged communications, or work product, related to personal representation or services by attorneys, psychotherapists, or clergy, and their assistants. Such communication and work product are private and confidential. See User Agreement for details.

The screenshot below shows we are using this banner agreement:



Access privileges to the Configuration Management (CM) repository are reviewed annually, not every three months.

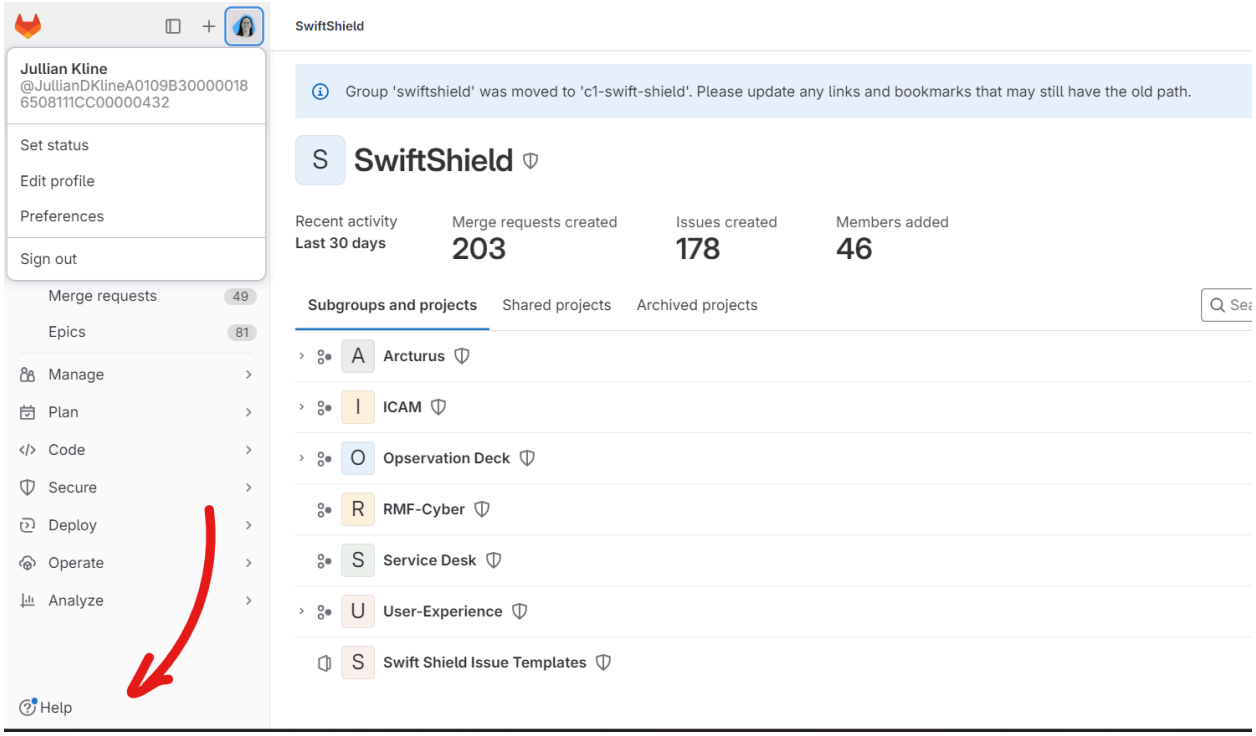
(V-222429, V-222514, V-222565) Privileges are limited in several ways to prevent changes to the software resident within software libraries. All deployments are deployed on AWS. Arcturus does use Gitlab pipeline templates in the Jarvis pipeline, which can be considered software libraries. The privileges to the Gitlab pipeline templates are handled in Gitlab and Jarvis is further isolated so customers cannot modify them. Gitlab handles many user permission configurations. Non-privileged users are prevented from executing privileged functions by requiring an ECA or CAC, which are a type of electronically verified and accepted Personal Identity Verification (PIV) credentials, and further would need approval from the Arcturus Engineering Team, when accessing the application for maintenance purposes. (V-222564) Arcturus verifies termination of remote maintenance sessions.



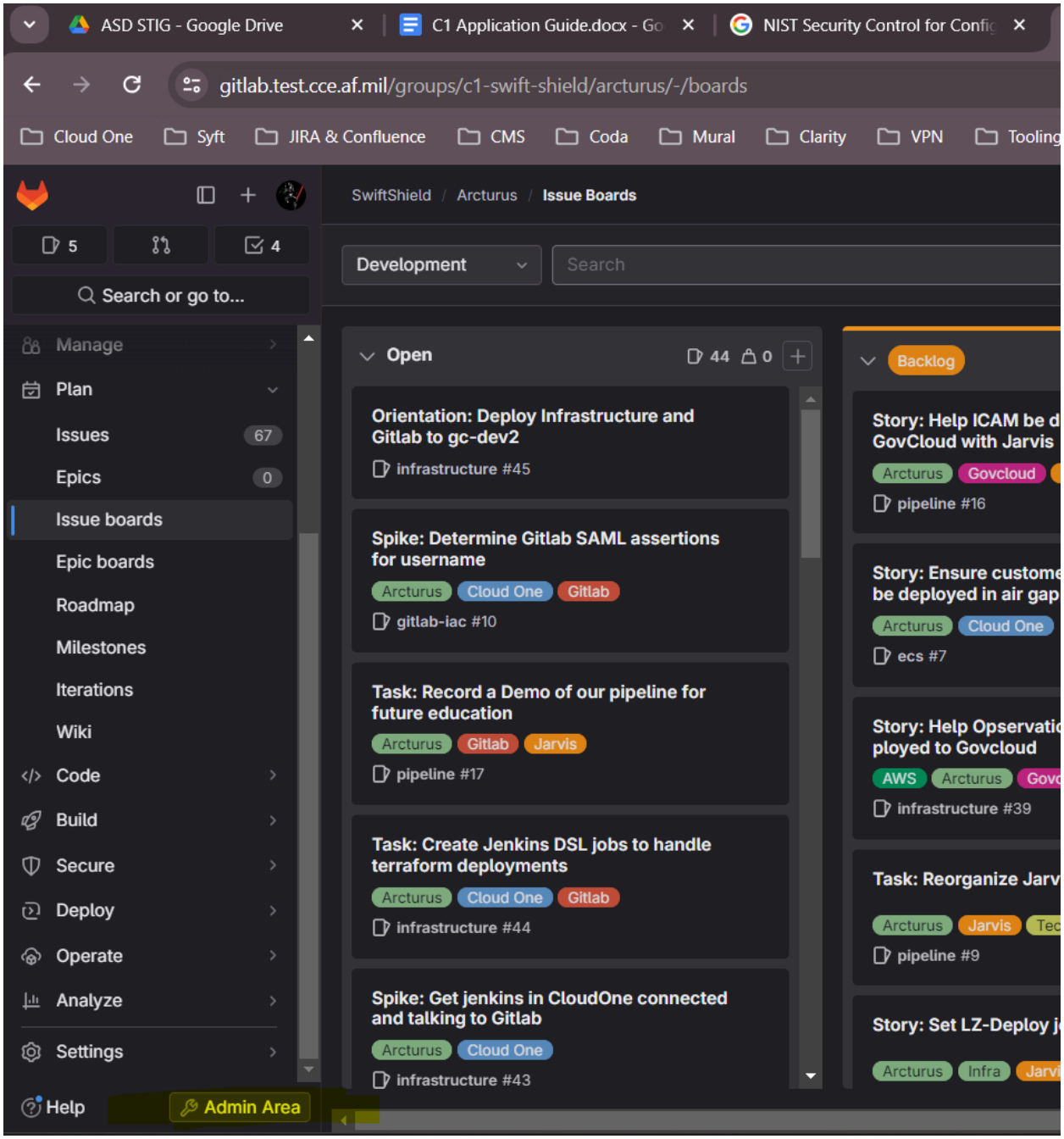
Maintenance session verifying termination.

(V-222626, V-222591) The application user data is separated by Gitlab into a different directory than the application code and user file permissions. All user data is in Gitlab and all the code is in the databases of Gitlab, separating them. Execution domains are separate as well, for each executing process.

(V-222511, V-222590, V-222600, V-222430, V-222407) Gitlab prevents unnecessary information from being disclosed to users by separating admin abilities through an admin button. Only admin accounts are able to see and click the admin button *after* having logged in. If a user is not an admin, they will not be able to see the admin button. This restricts access associated with changes to application configurations and configures all accounts with minimalist privileges. Non-privileged users cannot change application configurations. This isolates security functions from non-security functions. Below you will first see a screenshot of a non-privileged user, without the admin button on the bottom left, and then a screenshot of a privileged user, with the admin button visible. In the admin portal, account management functions are supported and Gitlab provides automation tools.



Non-privileged users do not have the admin button.

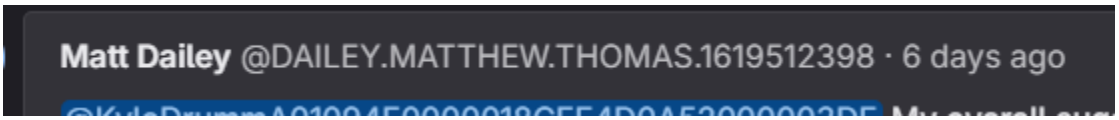


Privileged users have the admin area button.

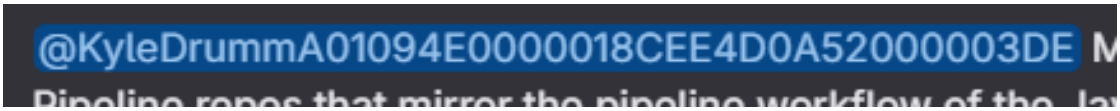
Authentication with CAC or ECA

(V-222522, V-222524, V-222525, V-222529, V-222533, V-222556, V-222557, V-222558, V-222560, V-222425) Arcturus uniquely identifies and authenticates organizational and non-organizational users by requiring CACs and ECAs, which conforms to FICAM-issued profiling requirements. This assures that users are authenticated with an individual authenticator

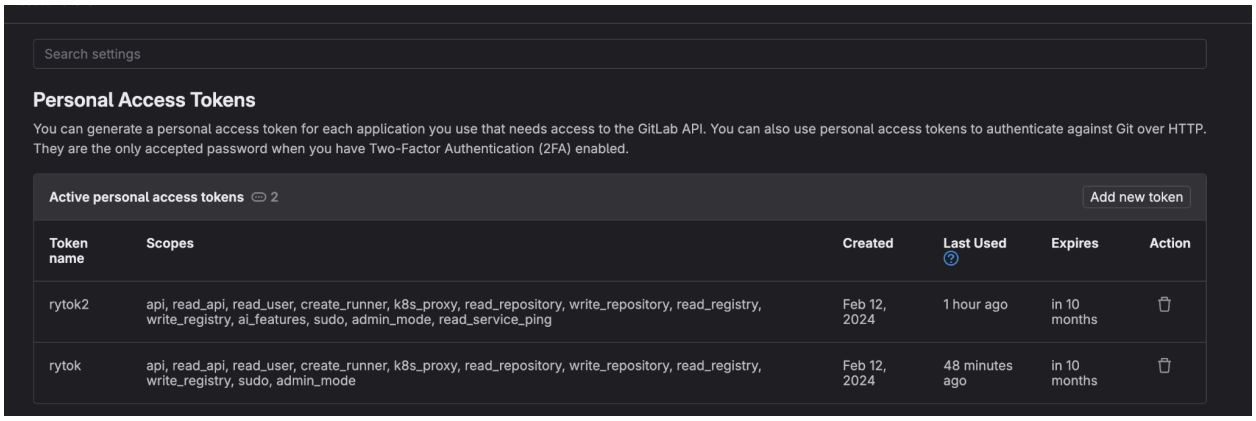
prior to using a group authenticator, because CACs and ECAs are designated to specific individuals, not groups.
(V-222523, V-222526, V-222527, V-222528) CACs and ECAs are multifactor because they incorporate something one has and something one knows. One is required for all network and local access to privileged and non-privileged accounts. All network connected endpoint devices are authenticated before establishing any connection through CACs, ECAs, or personal access tokens. (V-222584) DoD-approved certificate authorities are allowed for verification of the establishment of protected sessions through CAC usage.



A CAC user identifier.



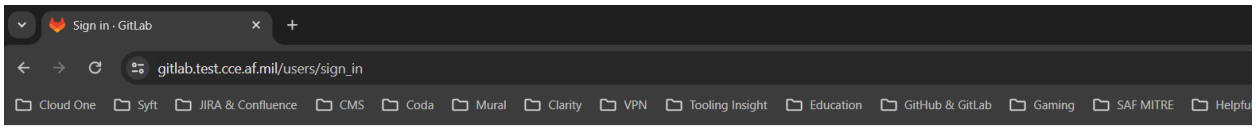
An ECA user identifier.

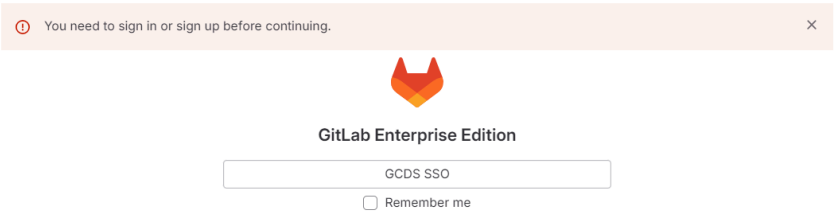


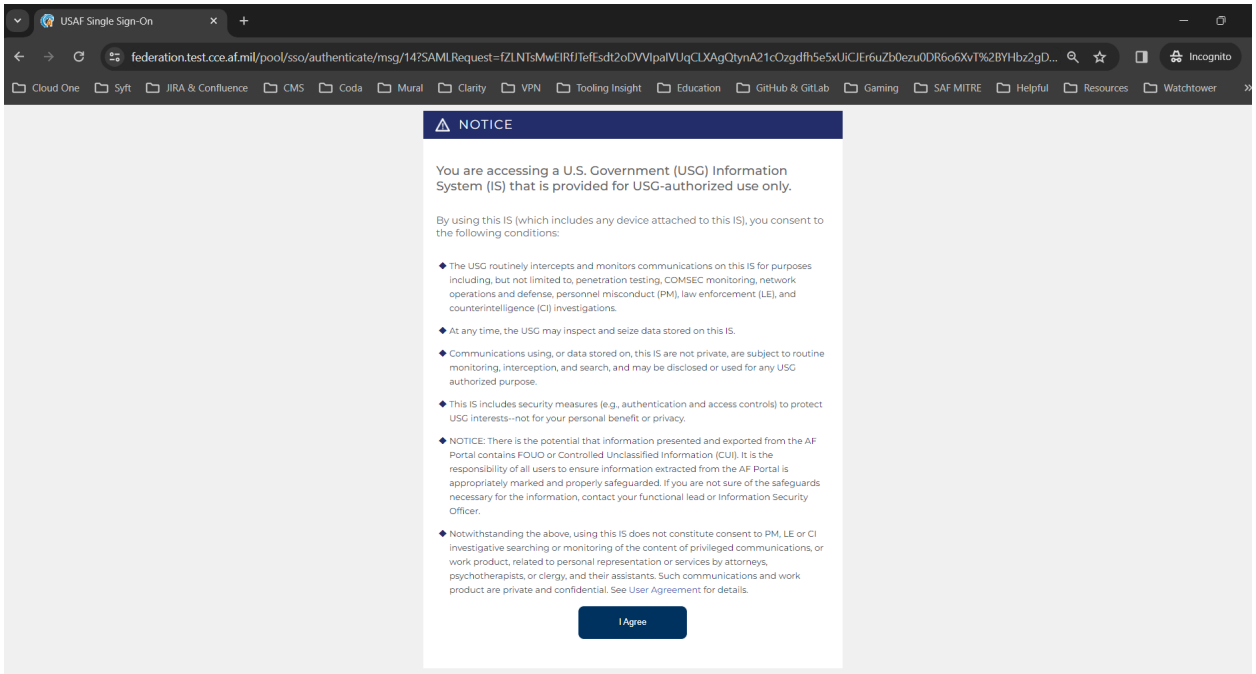
Personal Access Token

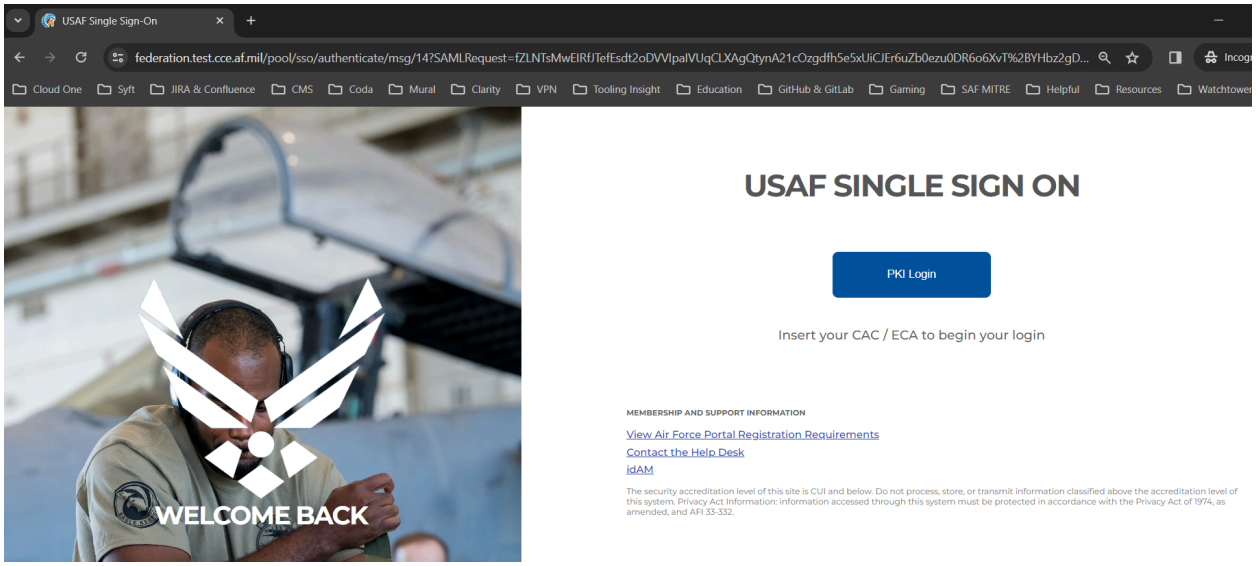
(V-222592) Personal access tokens prevent unauthorized and unintended information transfer via shared system resources. Gitlab's PATs are attached to a single username for authentication and are required when two-factor authentication or SAML is used.

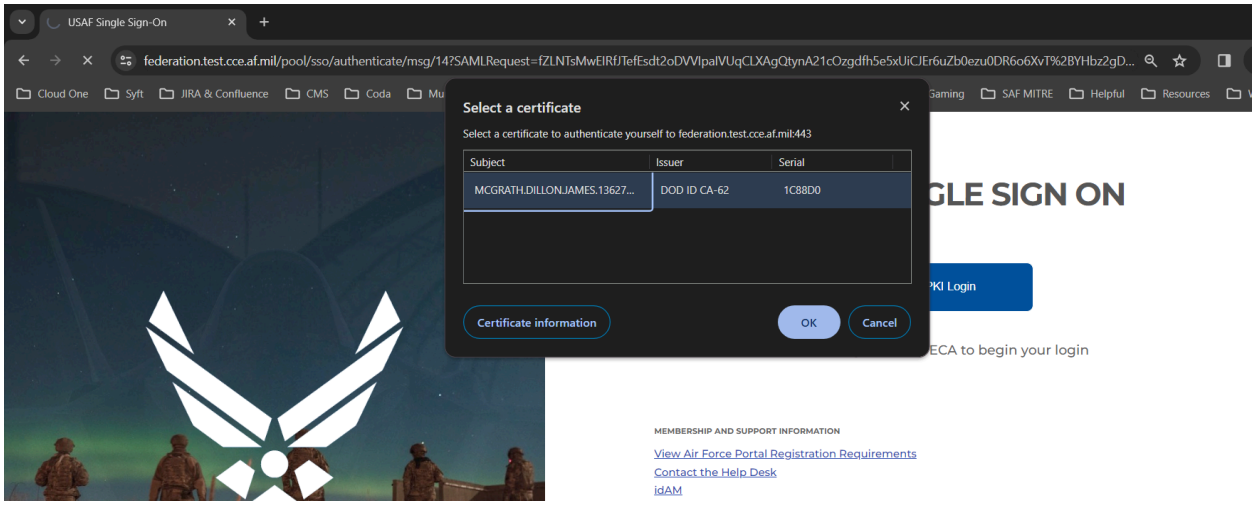
Logging in with CAC or ECA is provided through GCDS cloud one authentication. Below are screenshots of this process.


Sign in · GitLab
gitlab.test.cce.af.mil/users/sign_in
Cloud One
Syft
JIRA & Confluence
CMS
Coda
Mural
Clarity
VPN
Tooling Insight
Education
GitHub & GitLab
Gaming
SAF MITRE
Helpful


You need to sign in or sign up before continuing.
GitLab Enterprise Edition
GCDS SSO
Remember me


USAF Single Sign-On
federation.test.cce.af.mil/pool/sso/authenticate/msg/14?SAMLRequest=fZLNTsMwEIRfJTefEsdt2oDVVIpaIVUqCLXAgQtynA21cOzgdfh5e5xUiCJEr6uZb0ezu0DR6o6XvT%2BYHbz2gD...
Incognito
Cloud One
Syft
JIRA & Confluence
CMS
Coda
Mural
Clarity
VPN
Tooling Insight
Education
GitHub & GitLab
Gaming
SAF MITRE
Helpful
Resources
Watchtower
NOTICE
You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only.
By using this IS (which includes any device attached to this IS), you consent to the following conditions:
The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
At any time, the USG may inspect and seize data stored on this IS.
Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG authorized purpose.
This IS includes security measures (e.g., authentication and access controls) to protect USG interests--not for your personal benefit or privacy.
NOTICE: There is the potential that information presented and exported from the AF Portal contains FOUO or Controlled Unclassified Information (CUI). It is the responsibility of all users to ensure information extracted from the AF Portal is appropriately marked and properly safeguarded. If you are not sure of the safeguards necessary for the information, contact your functional lead or Information Security Officer.
Notwithstanding the above, using this IS does not constitute consent to PM, LE or CI investigative searching or monitoring of the content of privileged communications, or work product, related to personal representation or services by attorneys, psychotherapists, or clergy, and their assistants. Such communications and work product are private and confidential. See User Agreement for details.
I Agree





(V-222530, V-222531) When setting up an ECA through IdenTrust, there is a point when a user creates their Certificate Password. IdenTrust recommends it be between 8-30 characters with at least 1 lower case letter, 1 uppercase letter, and 1 special character. If a user picked a minimum requirement password, it would be easy for an attacker to brute force or sniff it during a replay attack, however ECAs are replay-resistant because the device must also have the proper certificates downloaded for the user to log in, to either a privileged or non-privileged account.

CACs are physical cards which are inserted into CAC readers on devices that must have the proper DoD certificates. When CAC users log in, they must supply their password, which has similar password requirements to the ECA. An attacker would not be able to brute force this, even if they stole the CAC card, because a user will be locked out of their CAC account after 3 incorrect password attempts. This makes CACs replay-resistant and keeps non-privileged accounts from gaining access.

(V-222550, V-222551, V-222552, V-222553) CACs and ECAs are PKI supplied by the DOD. Certificate Revocation is handled by GCDS. They validate certificates through a certification path, including status information, to the DOD, which is an accepted trust anchor. CACs and ECAs enforce authorized access to the corresponding private key, and map the authenticated identity to the individual user account. SAML tokens point to the CAC or ECA to identify users by DOD ID or ECA. It pulls the first and last name among other information. Users are redirected to our identity provider, who reads and verifies CAC/ECAs by constructing the SAML token and sending the token to Gitlab. Below is a screenshot of a SAML token and its configuration.

```
%{ if cloudone }
gitlab_rails['omniauth_auto_link_saml_user'] = true

gitlab_rails['omniauth_providers'] = [
  {
    name: 'saml',
    args: {
            assertion_consumer_service_url: 'https://repo.bespin.${saml_environment}cce.af.mil/users/auth/saml/callback',
            idp_cert_fingerprint: '${fingerprint}',
            idp_sso_target_url: 'https://federation.${federation_environment}.cce.af.mil/pool/sso/saml/authenticate',
            issuer: 'https://repo.bespin.${saml_environment}cce.af.mil/',
            name_identifier_format: 'urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress',
            attribute_statements: {
               email: ['otherMail'],
               nickname: ['common_name']
               }

           },
    label: 'GCDS SSO' # optional label for SAML login button, defaults to "Saml"
  }
]
%{ endif }
```

SAML Configuration.

```
Name
saml
authenticate?SAMLRequest=fZJLT8MwEIT%2FSm4%2BJc6Lt...VvXgf8%2FTZZk00Tt4naSCuxA6Xq3c0jE%2BGq%2F9ffXWH0B
federate?SAMLRequest=fZJLT8MwEIT%2FSm4%2BJc6LtlhNp...2F9ffXWH0B&u=https%3A%2F%2Fgitlab.test.cce.af.mil
14?SAMLRequest=fZJLT8MwEIT%2FSm4%2BJc6LtlhNpYgKqRI...2F9ffXWH0B&u=https%3A%2F%2Fgitlab.test.cce.af.mil
federate?SAMLRequest=fZJLT8MwEIT%2FSm4%2BJc6LtlhNp...2F9ffXWH0B&u=https%3A%2F%2Fgitlab.test.cce.af.mil
```

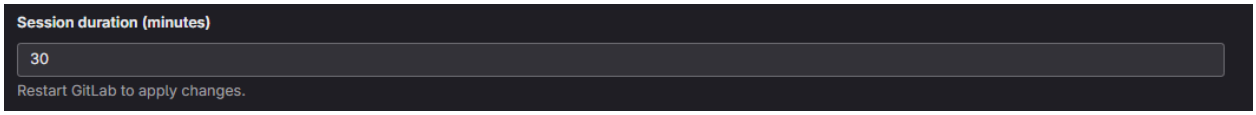
SAML token can be seen encrypted in the network console in Chrome when one logs into Gitlab.

Gitlab ensures confidentiality and integrity of information during reception, preparation for transmission, and after-transmitted protection through hashing and checking for changes in hashes. The Sonarqube and Anchore scanners also assist in checking data integrity. The tokens are sent during authentication, data transmission is handled, and integrity checked.

Account Management

(V-222520, V-222521) When user privilege is escalated and user roles are changed, Gitlab requires a refresh, but not a reauthentication. This is a finding left open, but counterbalanced by

the other CIA Arcturus controls, such as least privilege and access management. Accounts have a 30 minute timeout.



(V-222411) Accounts are disabled after 90 days of inactivity by default. This is a finding, as Gitlab does not allow for any duration of inactivity less than 90 days.

(V-222433) After 35 days of inactivity, user accounts are deactivated. A member of the Arcturus team will be able to reactivate an account using the following process:

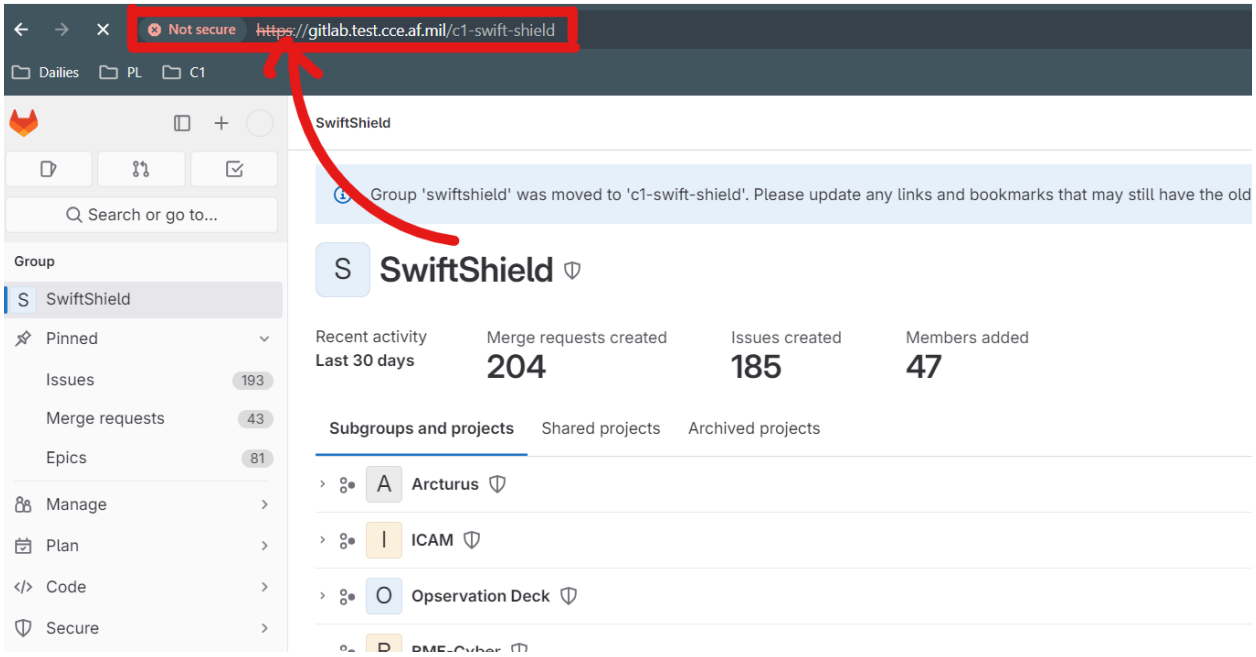
1. Login to Gitlab
2. Navigate to the Admin Area
3. Navigate to Users
4. Click on the Deactivated tab
5. Search for the user who needs to be reactivated
6. Click the three dots next to the user and select activate

(V-222412) Gitlab treats tokens as service accounts, what Arcturus interprets as "application accounts". The Arcturus Gitlab User & Project Management document instructs application teams to only create group tokens when absolutely necessary.

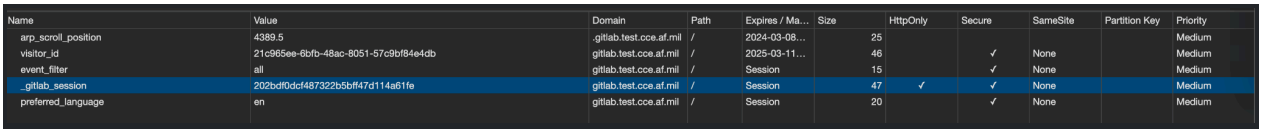
Please see the Documentation section for more information on the Account Management Process documentation.

Session Management

(V-222577, V-222579, V-222580, V-222581, V-222582) Gitlab does not expose session IDs and does not use URL embedded session IDs. Furthermore, session IDs in the web packets are hashed. Gitlab session data is in Redis, which requires server access to view. Gitlab uses and validates system-generated session identifiers that protect against session fixation, and does not reuse or recycle them.



Session ID not exposed in the URL.



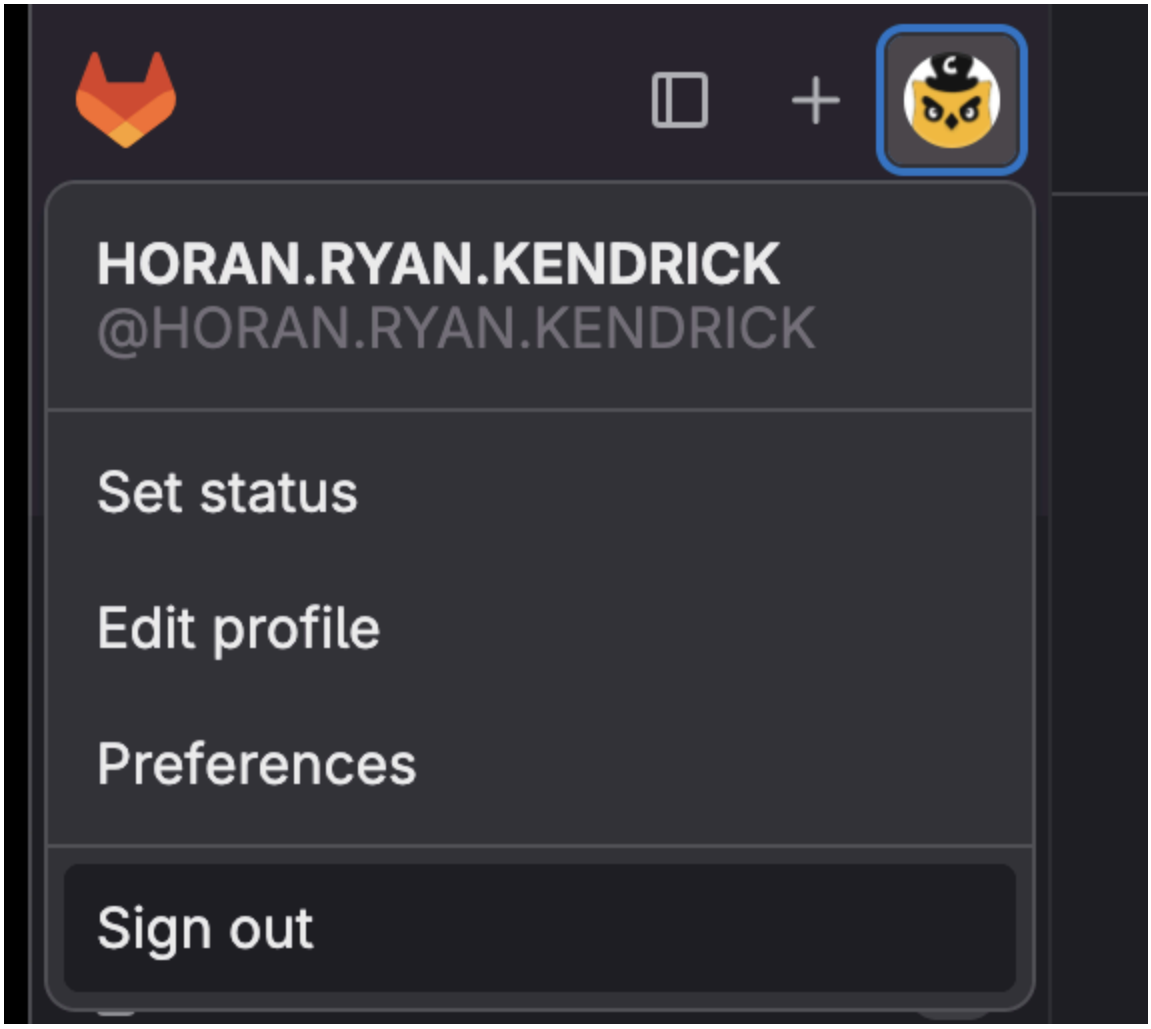
Name	Value	Domain	Path	Expires / Ma...	Size	HttpOnly	Secure	SameSite	Partition Key	Priority
arp_scroll_position	4389.5	.gitlab.test.cce.af.mil	/	2024-03-08...	25					Medium
visitor_id	21c965ee-6bfb-48ac-8051-57c9bf84e4db	gitlab.test.cce.af.mil	/	2025-03-11...	46		✓	None		Medium
event_filter	all	gitlab.test.cce.af.mil	/	Session	15		✓	None		Medium
_gitlab_session	202bdf0dcf487322b5bff47d114a61fe	gitlab.test.cce.af.mil	/	Session	47	✓	✓	None		Medium
preferred_language	en	gitlab.test.cce.af.mil	/	Session	20		✓	None		Medium

Session ID hashed in the web packets.

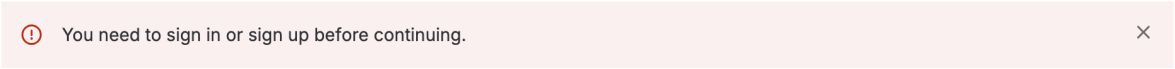
(V-222387) GitLab allows users to have up to 100 active sessions at once. If the number of active sessions exceeds 100, the oldest ones are deleted.

(V-222566, V-222390) Arcturus will expire idle user sessions after 30 minutes of inactivity. This is a finding as the STIG is specifying to expire idle user sessions after 10 minutes of inactivity for admin users and after 15 minutes of inactivity for regular users. 30 minutes was chosen for Arcturus instead because 10 and 15 minutes would cause a disturbance of workflow. The Arcturus team mediates this issue with proper user training around expected system use.

(V-222391) Gitlab provides a logoff capability in the top right corner of its UI, under a user's name.

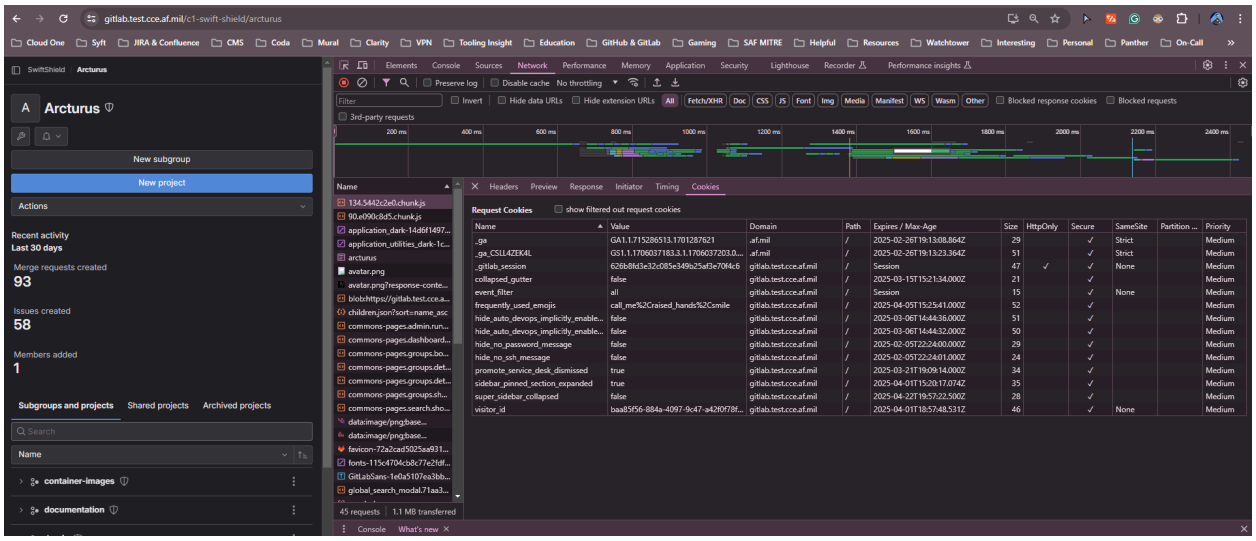


(V-222392) If ‘sign out’ is selected, the log in for CAC or ECA page shows, indicating the reliable termination of authenticated communications sessions as traffic is redirected to the log in page. If a user tries to reload a previously opened page an error message will appear.

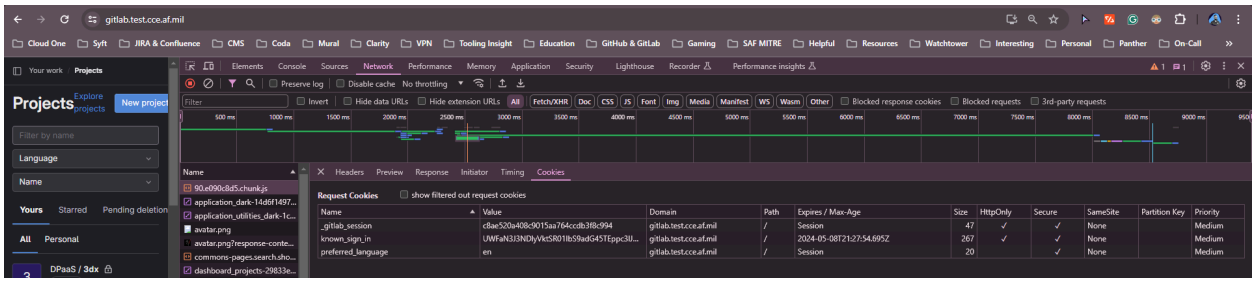


(V-222549) If an account is deleted, Gitlab will terminate existing user sessions.

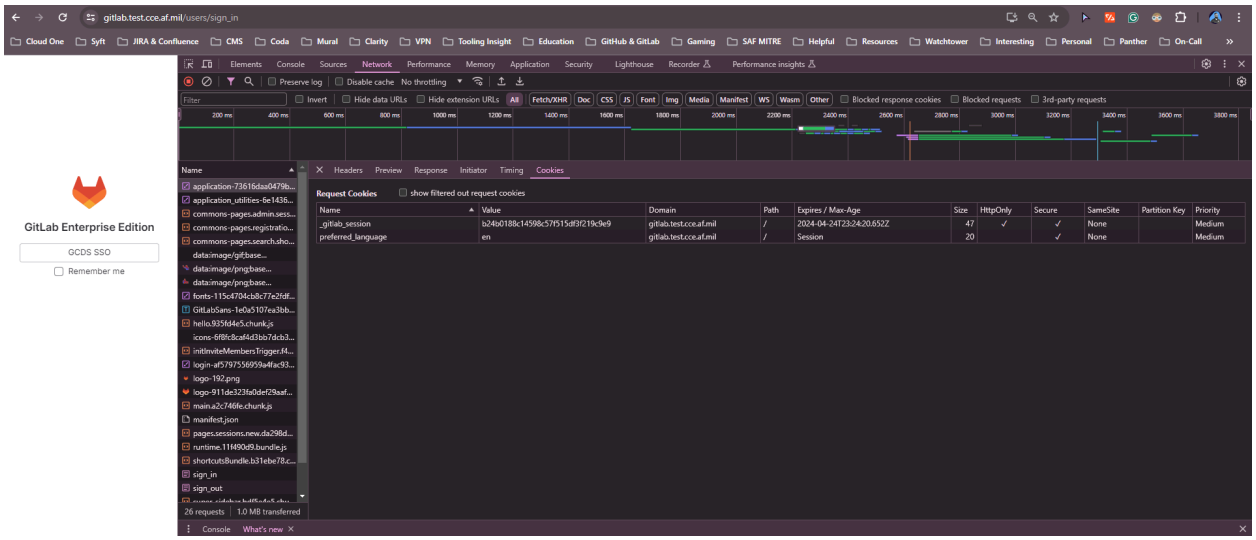
(V-222388, V-222578) On session end, the session ID value and cookie are destroyed. All temporary storage and cookies are cleared when the session is terminated. Below are screenshots of the session starting, just after authentication, and just after session termination. None of the session ID’s remain the same and the data is cleared. This is handled automatically by Gitlab.



Cookie creation.



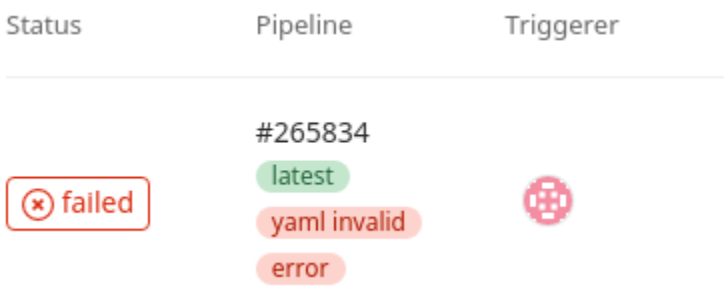
After authentication.



After session termination.

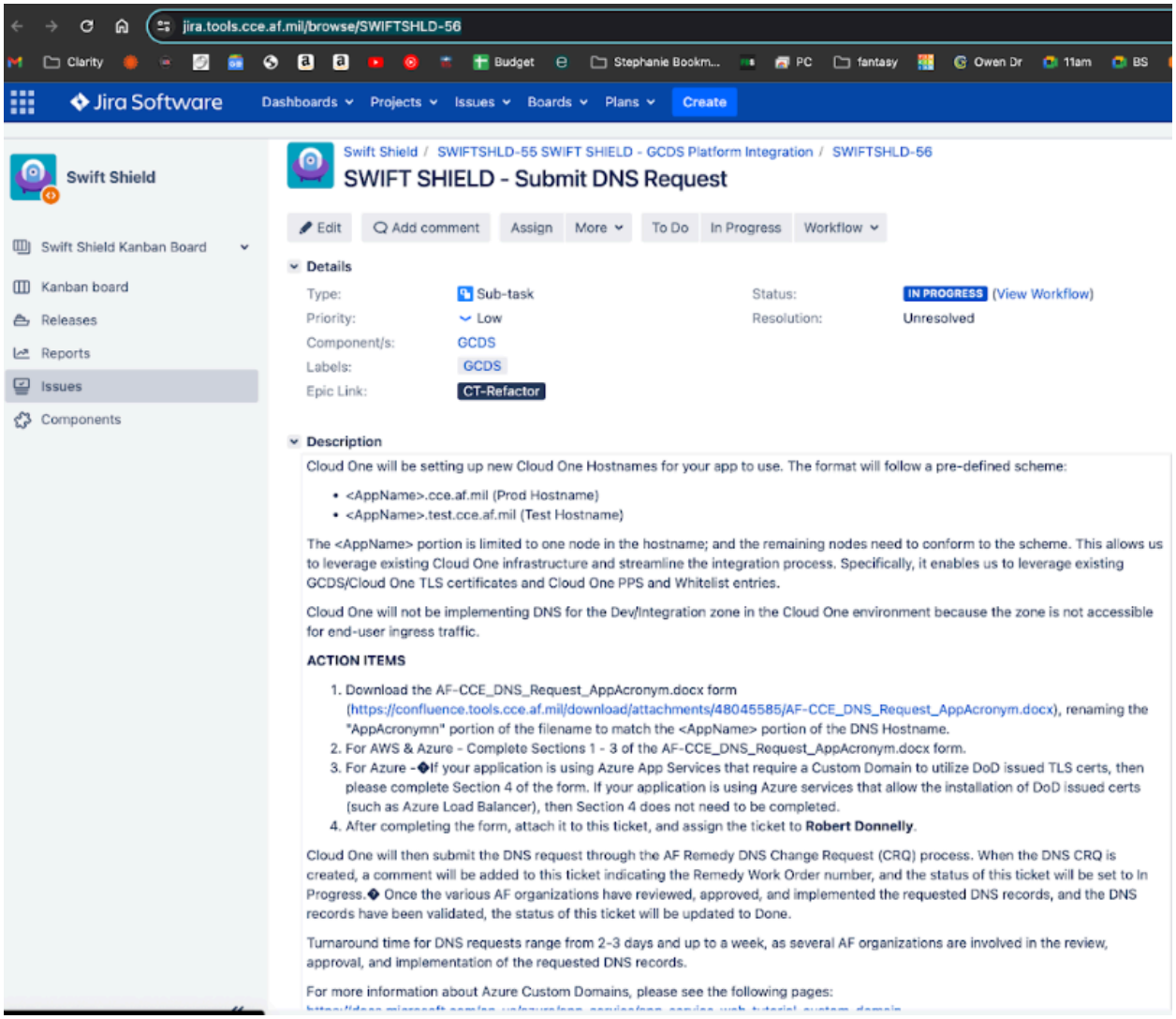
Code Compliances

(V-222653) The Arcturus team follows the Gitlab CI/CD Pipeline set of coding standards.
(V-222427) Information flow is controlled within the system through Gitlab's architecture.
(V-222650) Any flaws found during a code review are tracked by Gitlab. Gitlab's defect tracking system notifies Arcturus engineers.



DoD Ports and Protocols Database

(V-222629) Arcturus provisioned their DNS with Akamai and GCDS. Akamai registered the necessary ports and protocols with the Ports and Protocols Database. (V-222519, V-222628) GCDS and Akamai control which ports Arcturus can externally communicate with, and only use functions, ports, and protocols permitted to it in the PPSM CAL.



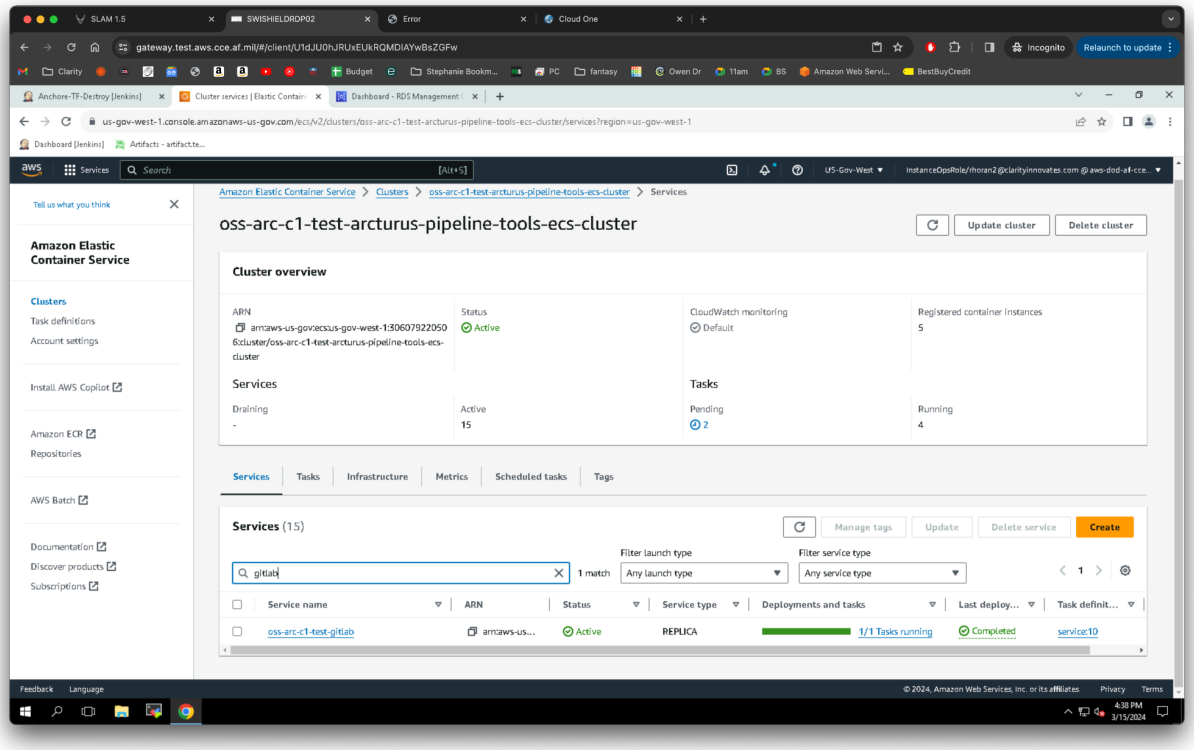
DNS record being requested in Jira.

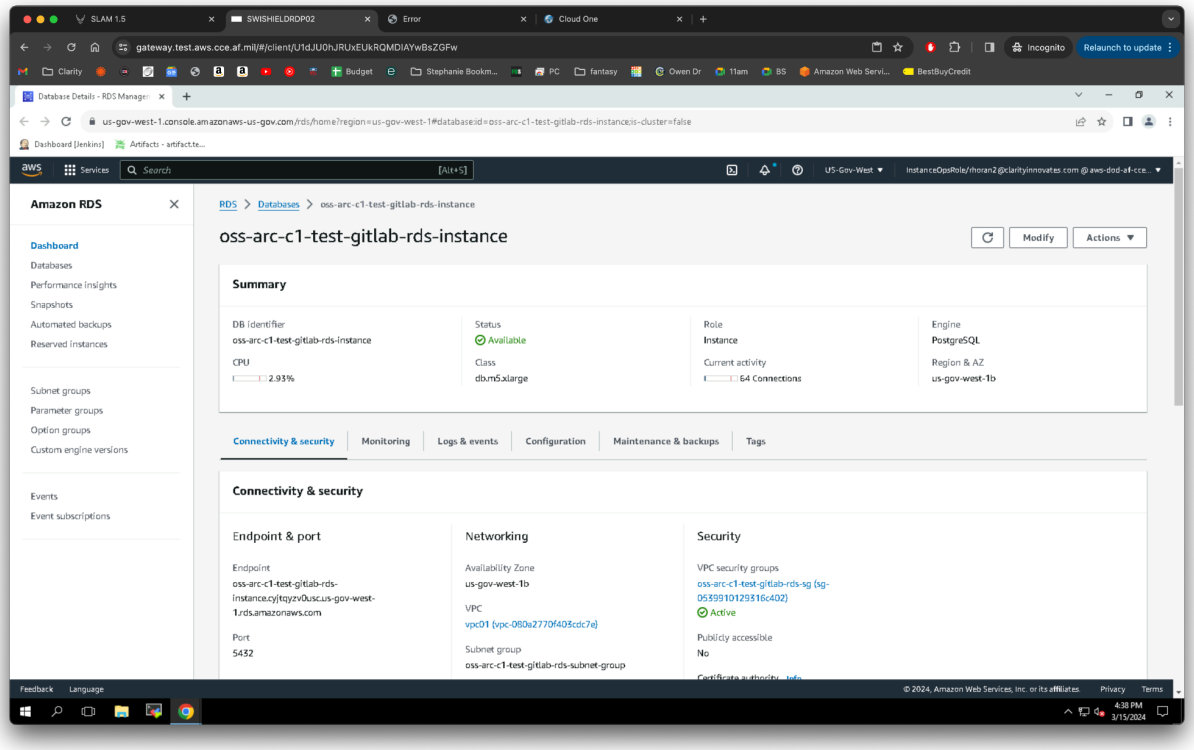
Customer Deployment

(V-222635) AWS has different accounts for each different application being handled by Arcturus, meaning that mission critical applications are deployed individually, and any container escape risk would be assumed by AWS. Arcturus simply needs to assure a job is pointed to the correct account when deploying.

GUI

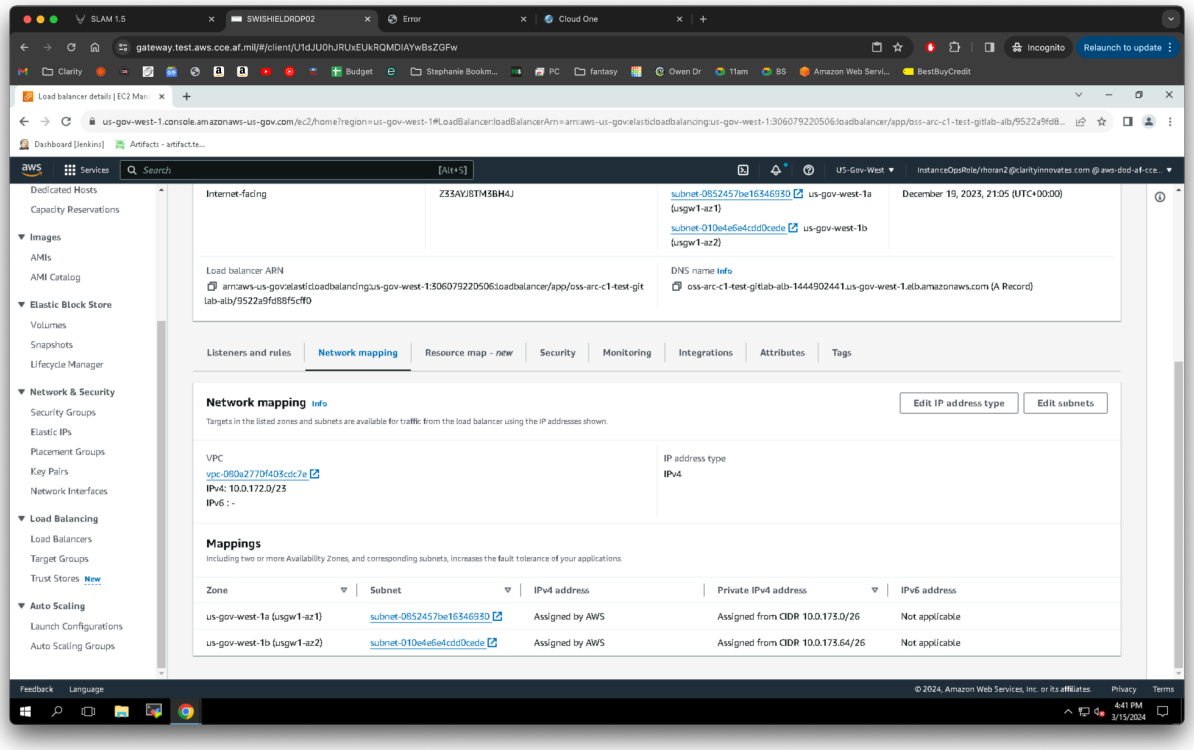
(V-222574) The Arcturus application user interface is separated from data storage and management interfaces. The data storage and management is done in the AWS RDS/S3 buckets and the user interface is done through Elastic Container Service (ECS).


SLAM 1.5
SWISHIELDRDP02
Error
Cloud One
gateway.test.aws.cce.af.mil/#/client/U1dJU0hJRUxEUkRQMDIAYwBsZGFw
Incognito
Relaunch to update
Clarity
Budget
Stephanie Bookm...
PC
fantasy
Owen Dr
11am
BS
Amazon Web Servi...
BestBuyCredit
Anchore-TF-Destroy [Jenkins]
Cluster services | Elastic Contain
Dashboard - RDS Management
us-gov-west-1.console.amazonaws-us-gov.com/ecs/v2/clusters/oss-arc-c1-test-arcturus-pipeline-tools-ecs-cluster/services?region=us-gov-west-1
Dashboard [Jenkins]
Artifacts - artifact.te...
aws
Services
Search
[Alt+S]
US-Gov-West
InstanceOpsRole/rhoran2@clarityinnovates.com @ aws-dod-af-cce...
Tell us what you think
Amazon Elastic Container Service
Clusters
Task definitions
Account settings
Install AWS Copilot
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Amazon Elastic Container Service > Clusters > oss-arc-c1-test-arcturus-pipeline-tools-ecs-cluster > Services
oss-arc-c1-test-arcturus-pipeline-tools-ecs-cluster
Update cluster
Delete cluster
Cluster overview
ARN
arn:aws-us-gov:ecs:us-gov-west-1:306079220506:cluster/oss-arc-c1-test-arcturus-pipeline-tools-ecs-cluster
Status
Active
CloudWatch monitoring
Default
Registered container instances
5
Services
Draining
-
Active
15
Tasks
Pending
2
Running
4
Services
Tasks
Infrastructure
Metrics
Scheduled tasks
Tags
Services (15)
Manage tags
Update
Delete service
Create
Filter launch type
Filter service type
gitlab
1 match
Any launch type
Any service type
Service name
ARN
Status
Service type
Deployments and tasks
Last deploy...
Task definit...
oss-arc-c1-test-gitlab
arn:aws-us...
Active
REPLICA
1/1 Tasks running
Completed
service:10
Feedback
Language
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
4:38 PM
3/15/2024



IPv6

(V-222634) [AWS services support for IPv6](#) includes support for dual stack configuration (IPv4 and IPv6) or IPv6 only configurations. Please see the screenshot below showing Arcturus does have IPv6 enabled.



Attack & Vulnerability Prevention

Arcturus

(V-222515, V-222648) Arcturus implements a number of scanners: OWASPDC, Sonarqube, Anchore, Trivy, and KICS. Each time code is committed, a scan job is performed on the code and outputs a scan report. The scan report flags active vulnerabilities in the code, assessing applications. (V-222605) Arcturus is protected against canonical representation vulnerabilities by running AWS inspector scans and Assured Compliance Assessment Solution (ACAS) scans. (V-222624) Furthermore, NOVA scanner performs active vulnerability and fuzz testing on Arcturus.

AWS

(V-222567, V-222593, V-222594, V-222602, V-222603, V-222604, V-222607, V-222608, V-222609, V-222612, V-222667) Arcturus is protected against DoS attacks, overflow attacks, and race conditions because we use AWS Elastic Load Balancer, which automatically scales to absorb the additional traffic or work when these types of attacks are detected. The ELB restricts the ability of attackers to launch DoS attacks, overflow attacks, race conditions, Cross-Site Scripting (XSS) attacks, SQL injection, XML-oriented attacks, input-handling attacks, and Cross-Site Request Forgery (CSRF) attacks. XSS, SQL injection, XML-oriented attacks, input-handling attacks, and CSRF are prevented by the ELB validating URLs and code

checking. Command injection is similarly prevented by the AWS ELB, and Arcturus validates pipelines before they are executed to ensure there are no malicious pipeline commands executing.

Gitlab

(V-222606) Gitlab validates all input prior to executing any commands. It also sanitizes and verifies, covering active vulnerability and fuzz testing of Arcturus.

(V-222656) Arcturus is not subject to error handling vulnerabilities because Gitlab is responsible for the container orchestration through pipeline jobs. Operations occur in separate instances (containers) and if an error occurs that takes down the container, it is “caught” by killing the instance and bringing up a new one.

Clean Up

(V-222658, V-222518) All unsupported software products in Arcturus do not need to be removed or decommissioned because they would not have been set up in the first place. For a feature to exist in Arcturus, it needs to be configured, and Arcturus engineers do not configure a software unless it is needed and supported. The Gitlab configuration file has to have a product or feature explicitly set as true. This means all non-essential capabilities are disabled.

SwiftShield / ••• / server / Gitlab lac / **Repository**

```
# gitlab_rails['gitlab_shell_git_timeout'] = 800

### Extra customization
# gitlab_rails['extra_google_analytics_id'] = '_your_tracking_id'
# gitlab_rails['extra_piwik_url'] = '_your_piwik_url'
# gitlab_rails['extra_piwik_site_id'] = '_your_piwik_site_id'

##! Docs: https://docs.gitlab.com/omnibus/settings/environment-variables.html
# gitlab_rails['env'] = {
#   'BUNDLE_GEMFILE' => "/opt/gitlab/embedded/service/gitlab-rails/Gemfile",
#   'PATH' => "/opt/gitlab/bin:/opt/gitlab/embedded/bin:/bin:/usr/bin"
# }

# gitlab_rails['rack_attack_git_basic_auth'] = {
#   'enabled' => false,
#   'ip_whitelist' => ["127.0.0.1"],
#   'maxretry' => 10,
#   'findtime' => 60,
#   'bantime' => 3600
# }

###! **We do not recommend changing these directories.**
# gitlab_rails['dir'] = "/var/opt/gitlab/gitlab-rails"
# gitlab_rails['log_directory'] = "/var/log/gitlab/gitlab-rails"

### GitLab application settings
# gitlab_rails['uploads_directory'] = "/var/opt/gitlab/gitlab-rails/uploads"

#### Change the initial default admin password and shared runner registration tokens.
####! **Only applicable on initial setup, changing these settings after database
####!   is created and seeded won't yield any change.**
# gitlab_rails['initial_root_password'] = "password"
# gitlab_rails['initial_shared_runners_registration_token'] = "token"
```

Gitlab configuration file disabling a feature, because it is not set to “true”.

(V-222535) Device identifiers are disabled after 35 days of inactivity.

Maximum allowable lifetime for access token (days)

35

When left blank, default value of 365 is applied. When set, value must be 365 or less. When changed, existing access tokens with an expiration date beyond the maximum allowable lifetime are revoked.

PATs expire after 35 days.

(V-222613) All organization-defined software components are removed after updated versions have been installed because Gitlab is running on a container and when Gitlab updates, it creates a new container, thus removing the previous one completely. The docker file being used has a variable inside it (see screenshot below) which indicates a brand new container is in use when it changes, acting as a new operating system entirely.

Dockerfile 791 B

```
# Subject to SBIR Data Rights - Clarity Innovations LLC

# Based off of gitlab/gitlab-ee dockerhub image
# https://hub.docker.com/r/gitlab/gitlab-ee/
ARG GITLAB_VERSION

FROM gitlab/gitlab-ee:$GITLAB_VERSION

USER root

RUN apt-get update \
    && apt-get install -y jq curl gettext-base \
    && apt-get install -y apt-utils \
    && apt-get install -y libgnutls30 \
    && pip3 install trufflehog

# Install DoD CA Certs
COPY certs/  /etc/gitlab/trusted-certs/
# RUN update-ca-trust enable \
#    && update-ca-trust extract

WORKDIR /opt/gitlab/embedded/service/gitlab-shell/hooks/post-receive.d
COPY scripts/post-receive.sh ./post-receive
RUN chmod +x ./post-receive \
    && chown git:git ./post-receive

WORKDIR /var/log
RUN touch trufflehog.log \
    && chown git:root trufflehog.log
```

Anytime this changes a new container gets deployed.
Previous container is removed.

Verification

(V-222615) Correct operations of security functions are verified several ways. For example, many security functions are inherited from AWS as well as what C1 has provisioned on AWS. Gitlab has access logs and AWS has cloudtrail, which is explained more in the Alerts, Logging, and Auditing sections of this document. Access controls and account management are explained more in the Configuration > Access Controls section of this document, etc.

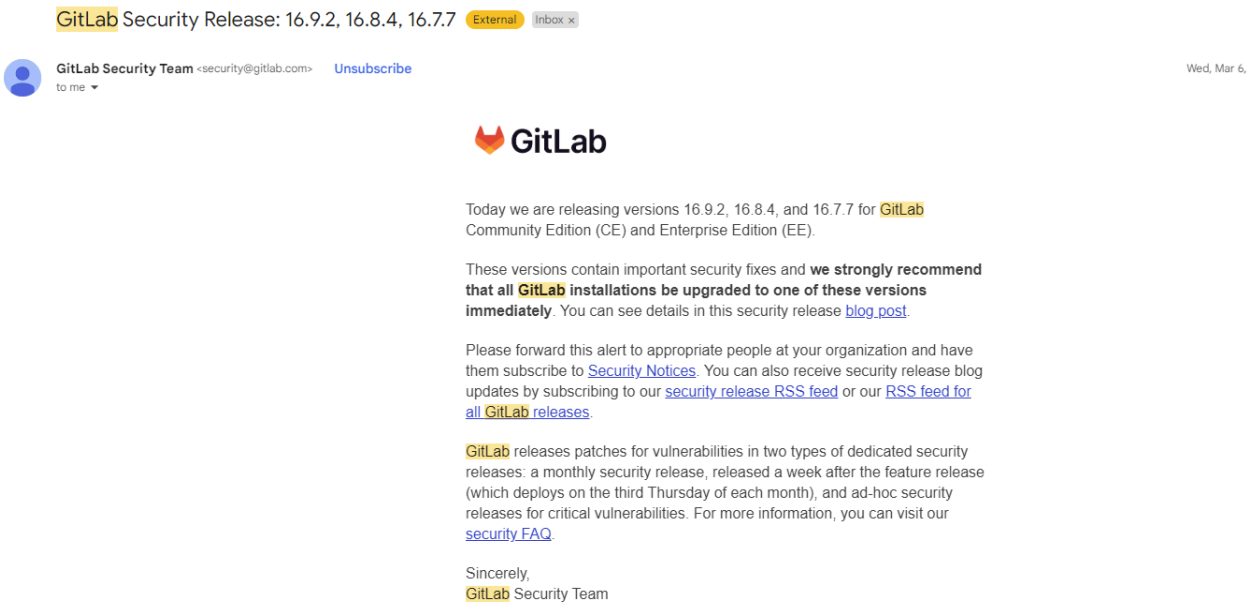
Decommission (End of Life)

(V-222659) In the event of the decommissioning of Arcturus, since all tools are deployed to AWS via terraform, the Arcturus team can simply run a Terraform Destroy command. This command deletes the database and any corresponding storage and services. User and service account credentials will be deactivated because they will no longer exist. This terminates all

regular and enterprise data, thus all data flows, and, when the Destroy command is run, will take about 5 minutes to complete.

Alerts

(V-222669, V-222670, V-222654) Administrators are registered to receive update notifications through Gitlab update emails.Product updates and security related patches are available on various software and tool release notes, such as for Gitlab, Sonarqube, Anchore, KICS, and Trivy.



GitLab Security Release: 16.9.2, 16.8.4, 16.7.7 External Inbox ×

GitLab Security Team <security@gitlab.com> Unsubscribe Wed, Mar 6,
to me

GitLab

Today we are releasing versions 16.9.2, 16.8.4, and 16.7.7 for GitLab Community Edition (CE) and Enterprise Edition (EE).

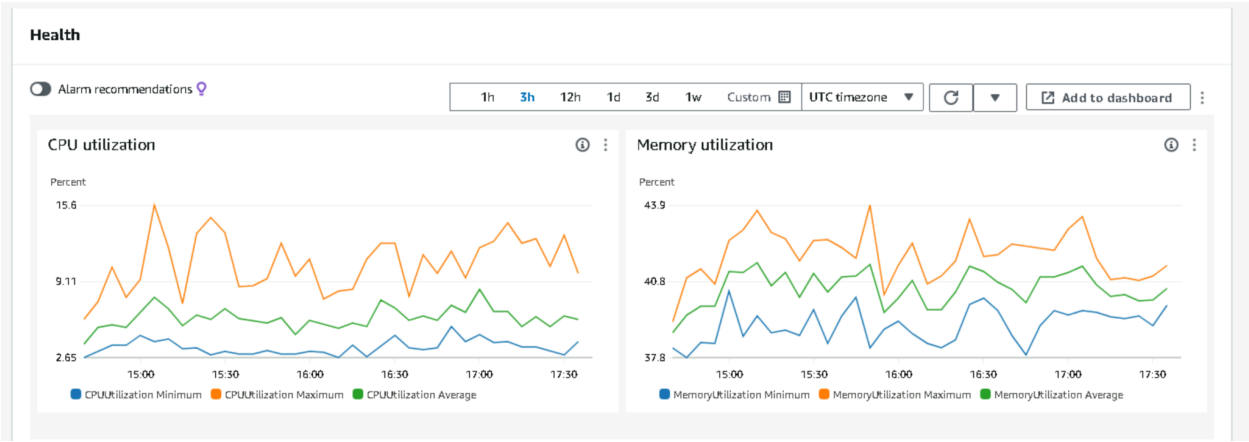
These versions contain important security fixes and **we strongly recommend that all GitLab installations be upgraded to one of these versions immediately**. You can see details in this security release blog post.

Please forward this alert to appropriate people at your organization and have them subscribe to Security Notices. You can also receive security release blog updates by subscribing to our security release RSS feed or our RSS feed for all GitLab releases.

GitLab releases patches for vulnerabilities in two types of dedicated security releases: a monthly security release, released a week after the feature release (which deploys on the third Thursday of each month), and ad-hoc security releases for critical vulnerabilities. For more information, you can visit our security FAQ.

Sincerely,
GitLab Security Team

(V-222668) Administrators are alerted when low resource conditions are encountered through cloud watch logs.



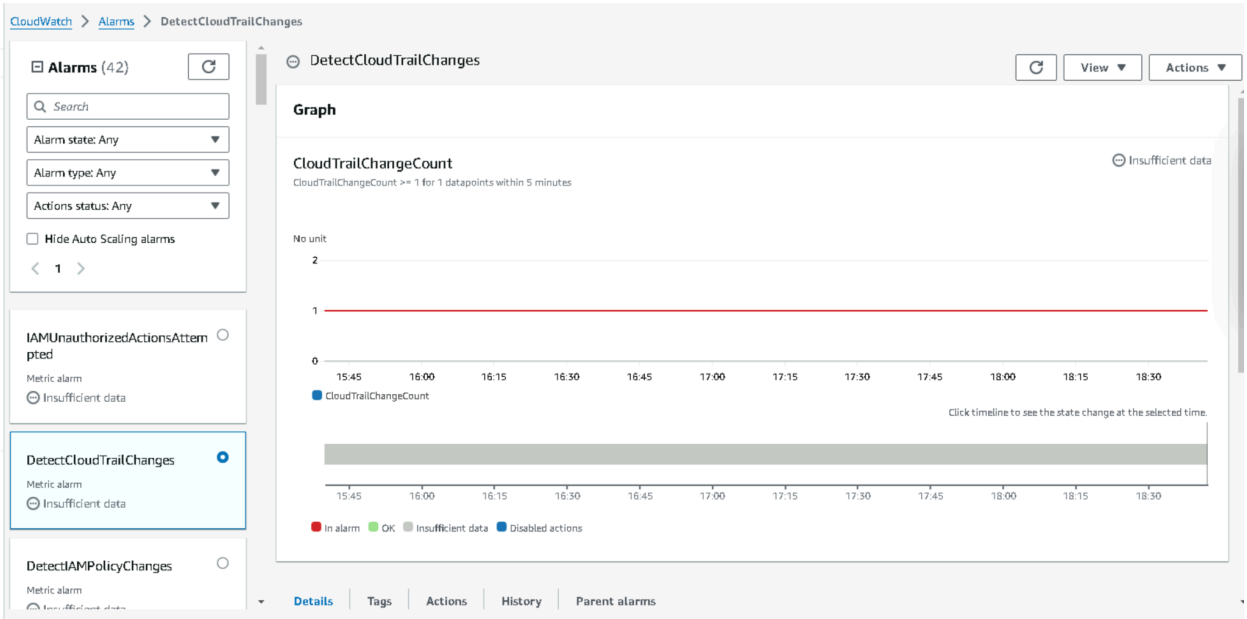
Cloudwatch showing resource levels.

Name	State	Last state update	Conditions
oss-arc-c1-test-gitlab-runner-ecs-cluster-mem-utilization-alarm	OK	2024-02-15 16:19:54	mem_used >= 13743895347.2 for 2 datapoints within 4 minutes
oss-arc-c1-test-gitlab-runner-ecs-cluster-cpu-utilization-alarm	OK	2024-02-15 16:19:00	CPUUtilization >= 80 for 2 datapoints within 4 minutes
oss-arc-c1-test-arcturus-pipeline-tools-ecs-cluster-mem-utilization-alarm	OK	2023-12-20 15:45:09	mem_used >= 13743895347.2 for 2 datapoints within 4 minutes
oss-arc-c1-test-arcturus-pipeline-tools-ecs-cluster-cpu-utilization-alarm	OK	2023-12-19 21:07:35	CPUUtilization >= 80 for 2 datapoints within 4 minutes

Cloudwatch alert for low resources.

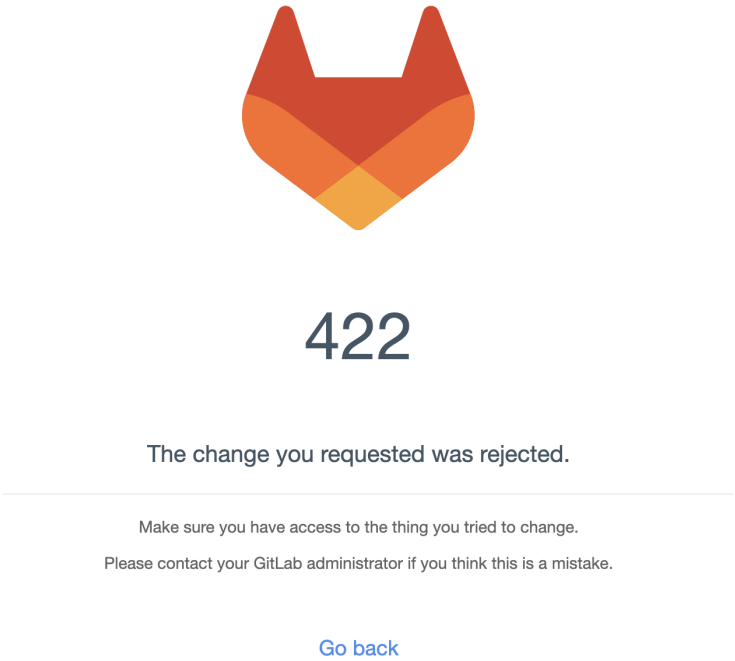
(V-222422, V-222417, V-222418, V-222419, V-222420) System Administrators will be notified via email defined by GitLab group and user management policy when an account is enabled, created, modified, disabled, or removed.

(V-222611, V-222485, V-222484, V-222617) Only administrators have access to receiving error messages and viewing logs on them in cloudwatch. A user cannot see either of these without admin access. The ISSO and SA (at a minimum) are alerted in the event of an audit processing failure. These alerts are immediate and in real-time for all audit failure events and failed security verification tests.



A Boundary Alert which goes to CloudWatch.

(V-222610) Users are provided with necessary corrective actions without revealing information that could be exploited by adversaries.



Example of a Non-privileged User Error Message.

(V-222660) Arcturus is currently in the process of defining decommission and end of life policy alerts for customers. This is a finding at this time, but will be revisited and completed during the ATO process.

Logging

(V-222482, V-222480, V-222487, V-222475) Arcturus centralizes logging with AWS CloudWatch. CloudWatch stores the logs and CloudWatch Logs Insights allows for interactive search and analysis of uniquely identified CloudWatch logs.

(V-222449, V-222477, V-222512, V-222473, V-222497, V-222498, V-222499, V-222478, V-222476, V-222438) AWS Cloudwatch logs the identity of the process, not user, but has relevant data enough for non-repudiation. (V-222532) Arcturus doesn't allow endpoint device connections to the pipeline that are not users, and all users are authenticated for non-repudiation. (V-222448, V-222470, V-222447, V-222439, V-222446) The logs include connecting IP addresses, destination IP addresses, header information such as GET, POST, User-Agent, referrer, X-Forwarded-For, expiration, date, UTC time codes, and are updated within seconds of an action being taken. Full commands are shown, indicating the outcome of events. This information is all carried over in the auditing of these logs.

Timestamp	Message
	There are older events to load. *Load more.*
2024-03-29T16:04:14.835+00:00	10.0.173.51 - - [29/Mar/2024:16:04:14 +0000] "POST /api/v4/jobs/request HTTP/1.1" 204 0 "" "gitlab-runner 16.2.1 (16-2-stable; go1.20.5; linux/amd64)…
2024-03-29T16:04:15.836+00:00	10.0.173.51 - - [29/Mar/2024:16:04:14 +0000] "POST /api/v4/jobs/request HTTP/1.1" 204 0 "" "gitlab-runner 16.2.1 (16-2-stable; go1.20.5; linux/amd64)…
2024-03-29T16:04:15.836+00:00	10.0.173.123 - - [29/Mar/2024:16:04:15 +0000] "GET /api/graphql?query=query%20getPipelineHeaderData(%24fullPath%3A%20ID!%2C%20%24iid%3A%20ID!)%20%7B%…
2024-03-29T16:04:15.836+00:00	10.0.173.51 - - [29/Mar/2024:16:04:15 +0000] "POST /api/v4/jobs/request HTTP/1.1" 204 0 "" "gitlab-runner 16.2.1 (16-2-stable; go1.20.5; linux/amd64)…
2024-03-29T16:04:16.086+00:00	10.0.173.123 - - [29/Mar/2024:16:04:15 +0000] "GET /c1-swift-shield/arcturus/infrastructure/-/issues/52/realtime_changes HTTP/1.1" 200 2108 "https://…
2024-03-29T16:04:16.336+00:00	10.0.173.123 - - [29/Mar/2024:16:04:15 +0000] "GET /c1-swift-shield/arcturus/infrastructure/-/merge_requests HTTP/1.1" 200 15335 "https://gitlab.test…
2024-03-29T16:04:16.336+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "POST /api/v4/jobs/request HTTP/1.1" 204 0 "" "gitlab-runner 16.2.1 (16-2-stable; go1.20.5; linux/amd64)…
2024-03-29T16:04:16.336+00:00	10.0.173.123 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/page_bundles/issuable_list-aa6ef454edcfcb005892a5b1e2146cfce6122fa4467e0fbea734e7733b5384d…
2024-03-29T16:04:16.336+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/page_bundles/labels-4fb6b3c79179f6479d66609ef5c200c4a61d0b3fc42569efdfe91c525e55c7c9.css HT…
2024-03-29T16:04:16.336+00:00	10.0.173.123 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/webpack/commons-initGFMInput-pages.admin.abuse_reports.show-pages.admin.topics.edit-pages.…
2024-03-29T16:04:16.336+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/webpack/commons-pages.dashboard.issues-pages.groups.epics.index-pages.groups.issues-pages.g…
2024-03-29T16:04:16.586+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/webpack/commons-pages.groups.issues-pages.groups.merge_requests-pages.projects.issues.index…
2024-03-29T16:04:16.586+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "GET /uploads/-/system/user/avatar/34/avatar.png?width=40 HTTP/1.1" 304 0 "" "Mozilla/5.0 (Macintosh; In…
2024-03-29T16:04:16.586+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/webpack/15.1e070b86.chunk.js HTTP/1.1" 304 0 "" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_…
2024-03-29T16:04:16.586+00:00	10.0.173.123 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/webpack/pages.projects.merge_requests.index.5aa828a6.chunk.js HTTP/1.1" 304 0 "" "Mozilla/…
2024-03-29T16:04:16.586+00:00	10.0.173.51 - - [29/Mar/2024:16:04:16 +0000] "GET /assets/webpack/commons-pages.groups.epics.new-pages.groups.epics.show-pages.groups.merge_requests-…

(V-222444) None of the logs used have Passwords, Session IDs, Application source code, encryption keys, nor sensitive data such as Personal Health Information (PHI), Personally Identifiable Information (PII), or government identifiers such as SSN, etc. (V-222479, V-222472) AWS and Gitlab both have transactional logs on the databases showing sequential records of all changes to the database.

```
{
    "severity": "INFO",
    "time": "2024-03-29T16:12:11.722Z",
    "correlation_id": "01HT5FQ9F02EXSFARA91ZC4AMB",
    "meta.caller_id": "Projects::MergeRequests::ContentController#widget",
    "meta.remote_ip": "70.253.93.74",
    "meta.feature_category": "code_review_workflow",
    "meta.user": "SMITH.AARON.JAMES.1604560841",
    "meta.user_id": 78,
    "meta.project": "c1-swift-shield/arcturus/tool-deployments/gitlab/server/gitlab-iac",
    "meta.root_namespace": "c1-swift-shield",
    "meta.client_id": "user/78",
    "mergeability_project_id": 85,
    "mergeability_merge_request_id": 728,
    "mergeability.check_approved_service.successful.values": [
        false
    ],
    "mergeability.check_approved_service.duration_s.values": [
        0.025045645888894796
    ],
    "mergeability.check_approved_service.db_count.values": [
        13
    ],
    "mergeability.check_approved_service.db_cached_count.values": [
        1
    ],
    "mergeability.check_approved_service.db_primary_count.values": [
        13
    ],
    "mergeability.check_approved_service.db_main_count.values": [
        13
    ],
    "mergeability.check_approved_service.db_primary_cached_count.values": [
        1
    ],
    "mergeability.check_approved_service.db_main_cached_count.values": [
        1
    ],
    "mergeability.check_approved_service.db_primary_duration_s.values": [
        0.009000000000000001
    ],
    "mergeability.check_approved_service.db_main_duration_s.values": [
        0.009000000000000001
    ]
}
```


A User Changing Data.

(V-222437, V-222471, V-222450, V-222469) Gitlab's log system has an advanced system where everything is logged, including the date and time of shutdown events, the last successful logon, user actions involving access to data, and audits successful and unsuccessful attempts to grant privileges.

Auditing

(V-222489, V-222490, V-222491, V-222495) AWS CloudWatch and CloudWatch Logs Insights are used for a variety of auditing capabilities. CloudWatch stores the logs and CloudWatch Logs

Insights allows for interactive search and analysis of the CloudWatch logs. (V-222488) Log groups can be searched through using a query language. Log Insights uses a number of commands including 'filter', which is used to return only the log events that match one or more conditions. (V-222492, V-222494, V-222496) Logs can be downloaded from the CloudWatch console. (V-222841, V-222506) Arcturus audit records are generated on Gitlab and then off-loaded as logs to AWS CloudWatch instantly, as CloudWatch reads them directly from Gitlab. (V-222493) Both Gitlab and Insights have the capability for report generation as CSV files. (V-222413, V-222414, V-222415, V-222416, V-222421)Gitlab automatically audits account creation, account modification, account disabling, account removal, and account enabling.

(V-222413, V-222431, V-222441, V-222442, V-222443, V-222445, V-222451, V-222453, V-222454, V-222455, V-222457, V-222458, V-222460, V-222461, V-222462, V-222463, V-222464, V-222465, V-222466, V-222468, V-222467, V-222672) The Gitlab log system creates audit trails and records for all direct access to the information system, but all unsuccessful login attempts fail at the GCDS level. GCDS maintains all audit records for unsuccessful login attempts. Audit records are generated when successful access to objects occur, for privileged activities or other system-level access, and execution of privileged functions. This includes logon, deletion, modification, and access to security objects, categories of information, privileges, and events, including event disabling and termination. This also includes when successful attempts, deletion attempts, and access to security levels occur. It should be noted that Gitlab does not allow modification of security levels, specifically. Gitlab automatically, by default, initiates session auditing on startup with session ID creation, destruction, renewal, and timeout, and account creation. The audit records show the timestamp of when a user started and finished accessing the system, and logs when concurrent logons from different workstations occur.

(V-222486) There will not be any lack of audit storage capacity that may lead to an audit failure, because of the elastic cloud based nature of AWS, where logs are kept. This covers back ups and restorations of logs as well. If there is a pipeline issue leading to log failure, Gitlab will stop and flag the issue, effectively 'shutting down' until action is taken to mitigate the failure.

(V-222501, V-222500, V-222502, V-222503, V-222504, V-222505) Arcturus uses AWS Cloudwatch to cryptographically protect the integrity of audit information. Cloudwatch encrypts data at rest as well as end-to-end encryption of data in transit. For more information on AWS cryptography, please see this document > Configuration > Cryptography > AWS. Gitlab handles many user permission configurations. Non-privileged users are prevented from executing privileged functions by requiring an ECA or CAC. For more information on Arcturus access controls, please refer to this document > Configuration > Access Controls.

(V-222474) Gitlab audit event reports will capture the component, feature, or function of the application that triggered an audit event. The audit report contains different headers including the entity type (project, group, instance), target details, and a description of the action. (V-222621) All audit logs are retained for one year.

```
resource "aws_cloudwatch_log_group" "pipeline_tools_ecs_log_group" {
  name              = "/${var.org}/${var.gid}/${var.environment}/${var.pipeline_tools_cluster_name}/"
  retention_in_days = 365
  tags              = {}
}
```

retention_in_days is set to 365.

Redundancy

(V-222640) Gitlab provides a document of procedures for how to backup and how to restore. The backup document explains how to back up different kinds of data, provides a command-line interface to back up an entire instance, and provides alternative backup strategies. The restore document defines restore prerequisites, how to restore for different installations, and how to troubleshoot. (V-222638) Data backups occur every day at 0200 UTC. Data backups are executed on AWS EC2 and stored on AWS S3 storage, and the local gitlab container on EC2. All logs, data, audits, and anything that goes through CloudWatch are retained for 365 days. This ensures that Arcturus is protected from the loss of data in the event of a failure.

(V-222585) In the event of system failure, Gitlab's container will automatically restart and connect back to the database. By doing so, the container can resume to when the system level failure occurred. (V-222586) When the system failure takes place, any information necessary to determine the cause of failure, and any information necessary to return to operations with least disruption to mission processes, is preserved in AWS CloudWatch. Arcturus logs are stored in AWS CloudWatch, which would not be affected by an Arcturus failure. In the event of a system failure, backups will be implemented.

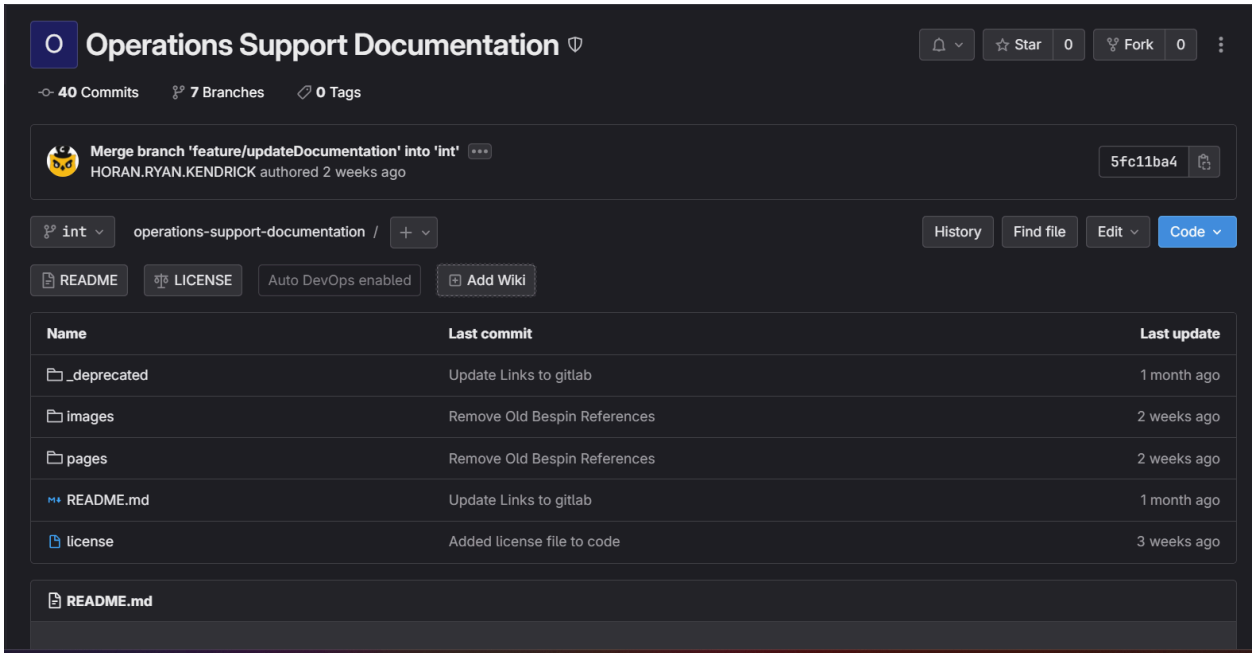
(V-222595) Arcturus uses AWS load balancers and a Web Application Firewall (WAF) as redundancy mechanisms. The AWS load balancers increase the fault tolerance of systems by automatically detecting server problems and redirecting client traffic to available servers. Specific to the DoD, Arcturus makes use of the WAF that is inherited by GCDS. This WAF works to protect against common web exploits and bots that can affect availability, compromise security, or consume excessive resources.

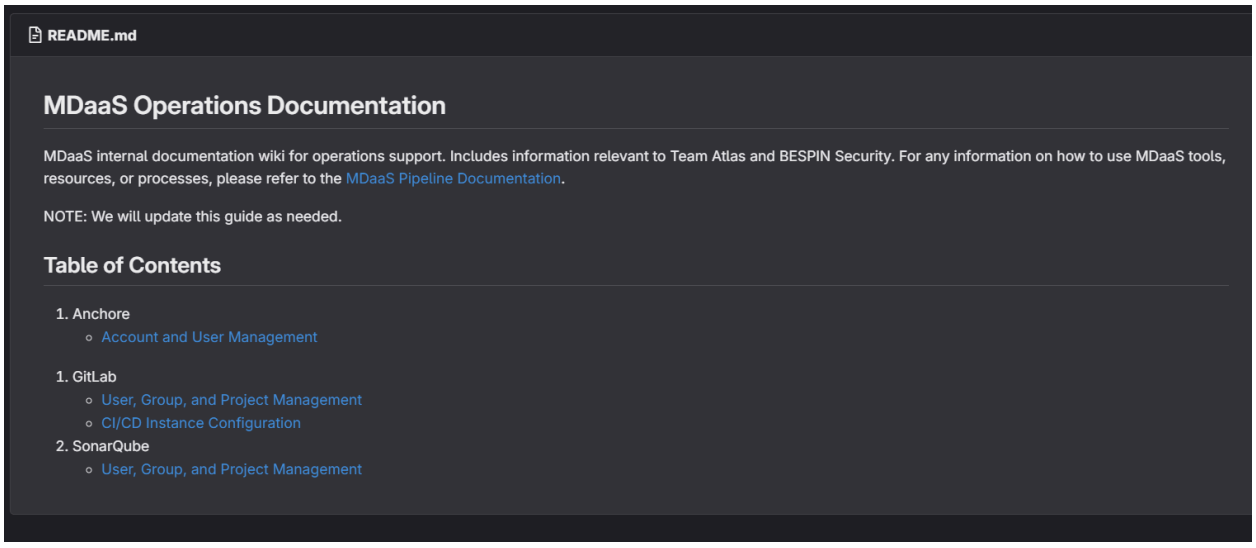
Documentation

(V-222622, V-222657, V-222614, V-222513, V-222623, V-222655, V-222652, V-222651, V-222637, V-222636, V-222625, V-222423, V-222516, V-222517, V-222426, V-222647) Arcturus has many documents for various policies and procedures including the Arcturus Reporting Policy, Arcturus Disaster Recovery/Continuity Plan, Arcturus Continuous Monitoring and Change Management Procedures, Arcturus Security Plan, Arcturus Threat Models, Arcturus Application Incident Response Plan, Arcturus Execution Flow Diagram and Design Document,

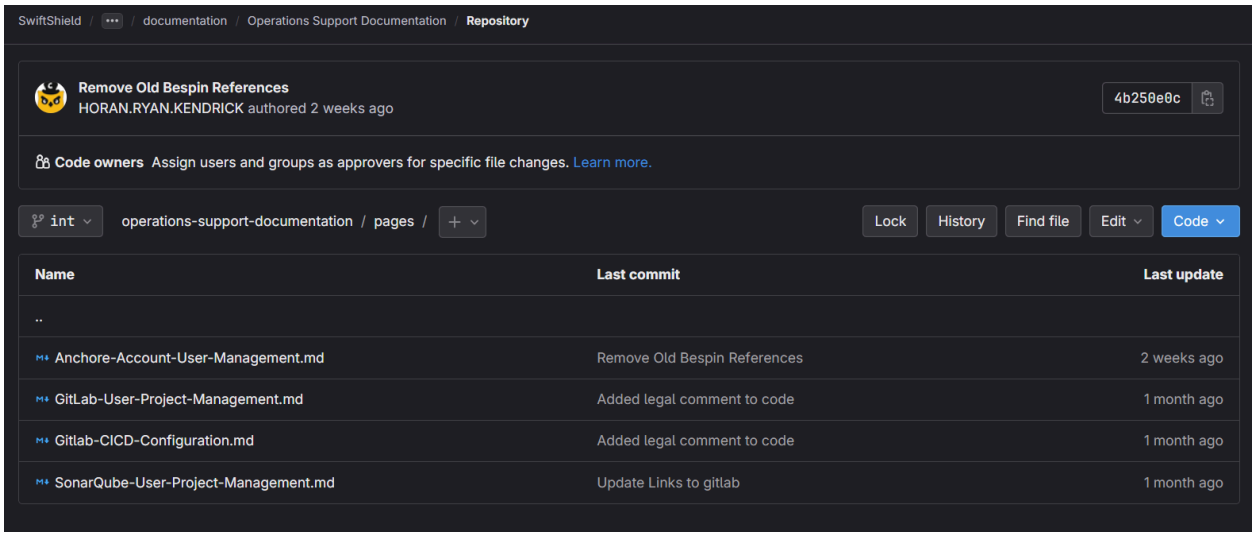
Arcturus Data Protection Policy, Arcturus Access Controls Policy, Arcturus Log Review Policy, Arcturus Test Procedure Plan, Arcturus User Agreement, and Arcturus User Guide.

(V-222632, V-222663) The Arcturus Application Configuration Guide is stored in Gitlab, which doubles as Arcturus' SCM plan. This guide contains information on Gitlab user, group, and project management and CI/CD instance configuration, as well as account and user management information for some of the implemented scanners. (V-222644) Arcturus uses Commercial-Off-The-Shelf (COTS) software and therefore, doesn't run any test plans because there is no software being coded. Gitlab, Sonarqube, Anchore, KICS, and Trivy all manage the test plans along with their release notes.





(V-222630) The Arcturus Configuration Management repository is the Terraform of the toolset that is stored throughout Gitlab. Arcturus developers ensure that the Terraform is properly patched and this ASD STIG is proving compliance. (V-222428, V-222619) Arcturus has several documents proving information and data flow controls, including Anchore Account User Management, Gitlab User Project Management, Gitlab CICD Configuration, Sonarqube User Project Management, and Account Management Process.



(V-222627) There is no DOD STIG or NSA guide for Gitlab. Gitlab is configured to security best practices.

Extra

Annual Assessments

(V-222673) Arcturus is a Clarity-owned product. All levels of Clarity program management, designers, developers, and testers are required to perform annual security training. Additionally, everyone working on the Arcturus pipeline must complete the C1 required cyber awareness training.

Personnel

(V-222646) Security testing on applications is conducted automatically via the Arcturus pipeline. As discussed in other vulnerability ID's, applications are run through a variety of scanners. Each time code is committed, a scan job is performed on the code and outputs a scan report. The scan report flags active vulnerabilities in the code, assessing applications.

(V-222633) The Arcturus team doesn't have a specific Configuration Control Board. Instead, the Configuration Management of Arcturus is done through sprint planning, milestone requirement

meetings, and lead syncs. In these meetings and planning sessions, developers, team leads, and project managers discuss changes to code, baselines, and any other work being done. These discussions are done in depth to ensure that the changes occurring are necessary.