



# Administrator's Guide

## Purpose of this Guide

This guide serves as a focused documentation index for administrators looking for **step-by-step onboarding, deployment, configuration, and operational guidance**.

This documentation is suitable for users deploying AccuKnox in real-world cloud-native environments. This curated guide is tailored for technical administrators and DevSecOps engineers who are looking for **concrete, task-oriented onboarding assets, installation steps, and configuration references**.

[AccuKnox Administrator's Guide](#)

[AccuKnox Enterprise Architecture](#)

[Core Components](#)

[Control Plane Architecture](#)

[Cloud Architecture](#)

[Externalized Storage Architecture](#)

[On-Premises Deployment Architecture](#)

[Scaling Considerations](#)

[Log & Data Storage](#)

[Customer Data Flow](#)

[Rules Engine Architecture](#)

[Integrations Architecture](#)

[Compliance Frameworks](#)

[Additional Resources](#)

[Getting Started With Technical Support](#)

[Product Documentation](#)

[Email Support and Procedures](#)

[Support Workflow](#)

[Priority Levels](#)

[Case Information Required](#)

[Video Conferencing Options](#)

[Case Resolution](#)

[Case Closure](#)

[Resources](#)

[FAQs](#)

[AccuKnox OnPrem Deployment Guide](#)

[System Requirements](#)

[Installation Steps](#)

[Use the following commands](#)

[Use of Private/Local Container Registry \(or air-gapped mode\)](#)

[Update the override-values.yaml](#)

[Install AccuKnox base dependencies](#)

[Install AccuKnox pre-chart](#)

[Install AccuKnox microservices chart](#)

[Install nginx ingress \(if any other self-managed Kubernetes\)](#)

[Verification of installation](#)

[Runtime Security Prerequisites](#)

[AccuKnox Agents](#)

[Pre-requisites](#)

[SSO Login Guide](#)

[1. Inviting a New User](#)

[2. User Receives Invitation](#)

[3. User Login Options](#)

[Notes](#)

[Onboarding Assets – High-Level Overview](#)

[Customer Environments](#)

[Cloud Onboarding Options](#)

[Kubernetes – AWS EKS / On-Prem / Fargate](#)

[Virtual Machines – EC2 / On-Prem](#)

[Container Registry](#)

[AI/ML Workloads – SageMaker / Bedrock](#)

[Deployment References](#)

[CSPM Pre-requisite for AWS](#)

[AWS Account onboarding](#)

[AWS IAM User Creation](#)

[AWS Onboarding](#)

[Onboarding AWS Organization Accounts to AccuKnox](#)

[Prerequisites](#)

[Step-by-Step Onboarding Process](#)

## [Post-Onboarding](#)

### [CSPM Pre-requisite for Azure](#)

#### [Azure Account onboarding](#)

[Rapid Onboarding \(via Azure\)](#)

[From AccuKnox SaaS UI](#)

### [CSPM Pre-requisite for GCP](#)

#### [GCP Account onboarding](#)

[From AccuKnox SaaS UI](#)

### [How to Deboard a Cloud Account](#)

#### [Kubernetes Security Onboarding](#)

[Features Supported for Kubernetes](#)

[K8s Runtime Visibility and Security](#)

[K8s Misconfiguration Scanning](#)

[K8s Identity & Entitlements Management](#)

[K8s CIS Benchmarking](#)

[DISA STIGs Support](#)

[In-Cluster Container Image Scanning](#)

[Admission Controller Support](#)

[Cluster Access to Control Plane](#)

#### [Cluster Onboarding](#)

[AccuKnox Agents](#)

#### [Cluster Onboarding with Access Keys](#)

[Onboarding](#)

#### [Onboard Cluster for Misconfiguration Scanning](#)

#### [CIS Benchmarking Compliance Scan Onboarding](#)

[Step 1: Generate an Access Token](#)

[Step 2: Onboard Your Cluster](#)

[Step 3: Deploy the Scanner Using Helm](#)

[Step 4: View Compliance Findings](#)

#### [Cluster Offboarding](#)

[Agents Uninstallation](#)

[Cluster Deletion](#)

#### [Runtime Security Deployment for Openshift](#)

[Operator Installation](#)

[ElasticSearch Integration](#)

[KubeArmor Instance Installation](#)

[Kibana Dashboard Setup](#)

#### [Onboarding and Deboarding VMs with Docker](#)

[Docker](#)

[Onboarding](#)

[Troubleshooting](#)

[Deboarding](#)

[Onboarding and Deboarding VMs with Systemd](#)

[Systemd](#)

[Network Requirements](#)

[Onboarding](#)

[Onboarding Worker Nodes](#)

[Troubleshooting](#)

[Deboarding](#)

[SystemD Based Non-BTF Environments](#)

[Compiling system monitor](#)

[Onboard the node](#)

[VM Onboarding using Access Keys](#)

[Overview](#)

[Pre-requisites](#)

[Onboarding](#)

[Onboarding Worker Nodes](#)

[Troubleshooting](#)

[Deboarding](#)

[In-Cluster Image Scanning with Helm](#)

 [Installation Guide](#)

[Dockerhub Registry Onboarding](#)

[Prerequisites](#)

[Steps to Add a Registry](#)

[Viewing Registry Scan Details](#)

[JFrog Container Registry Onboarding](#)

[AccuKnox Support for JFrog Container Registry Scanning](#)

[CWPP Report Generation](#)

[Regex](#)

[Reports Configuration](#)

[How to Configure Custom Reports](#)

[On-demand custom Report generation](#)

[Scheduling Custom Report](#)

[RINC](#)

[Supported reports](#)

[Installation](#)

[Passing Database Credentials](#)

[Accessing RINC's web interface](#)

[Advanced](#)  
[CWPP Troubleshooting](#)  
[Requirements](#)  
[Script To automate this process](#)  
[CSPM Troubleshooting Guide](#)  
[Step 1: Validate Prerequisites](#)  
[Step 2: Verify Cloud Scan Status](#)

---

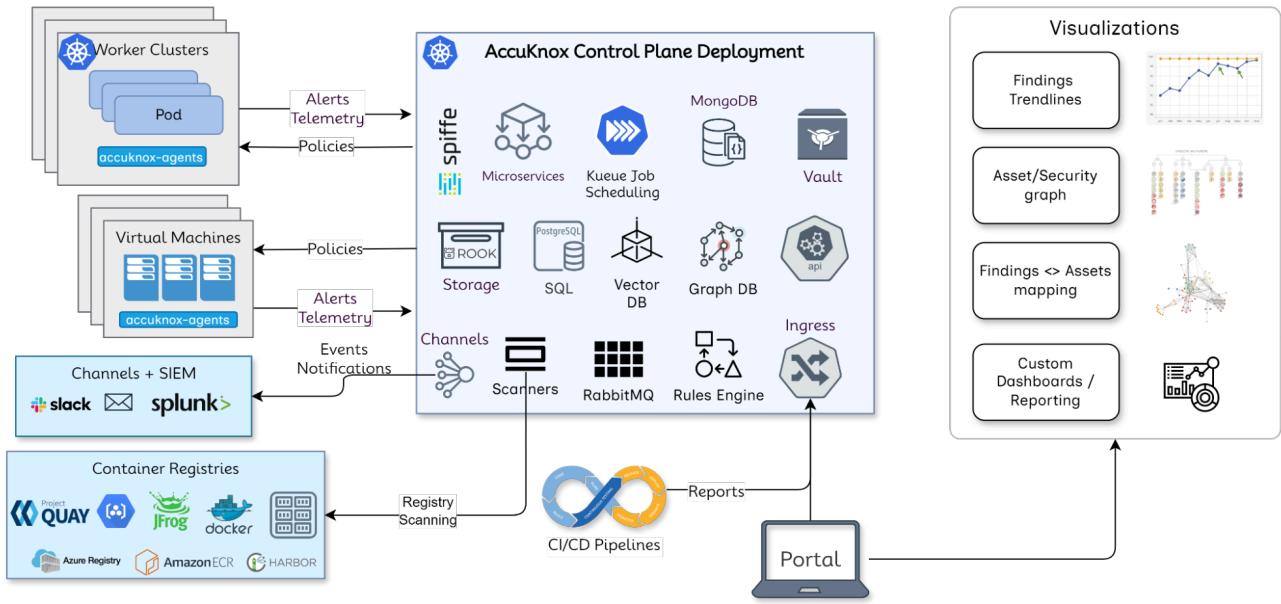
# AccuKnox Enterprise Architecture

AccuKnox's Cloud-Native Application Protection Platform (CNAPP) offers a unified **AppSec + CloudSec** solution, integrating modules like ASPM, CSPM, CWPP, KIEM, and GRC. This architecture ensures comprehensive security across the software development lifecycle.

## Core Components

### Control Plane Architecture

- **Microservices:**
  - *Divy*: Handles API requests.
  - *Celery*: Manages asynchronous tasks.
  - *Kueue*: Schedules Kubernetes-native jobs.
- **Parser Jobs:** Process asset and findings data, updating databases accordingly.
- **Alerts & Telemetry:** Ingested via RabbitMQ, processed for real-time insights.
- **Secure Onboarding:** Utilizes SPIFFE-based control plane for cluster onboarding.
- **Storage/Databases:**
  - *RDS*: Stores CSPM, KSPM, and ASPM data.
  - *MongoDB*: Handles streaming telemetry.
  - *Neo4j*: Manages metadata for KIEM.
- **Integrations:** Interfaces with SIEM tools (e.g., Splunk, Rsyslog) and ticketing systems (e.g., JIRA, Slack).

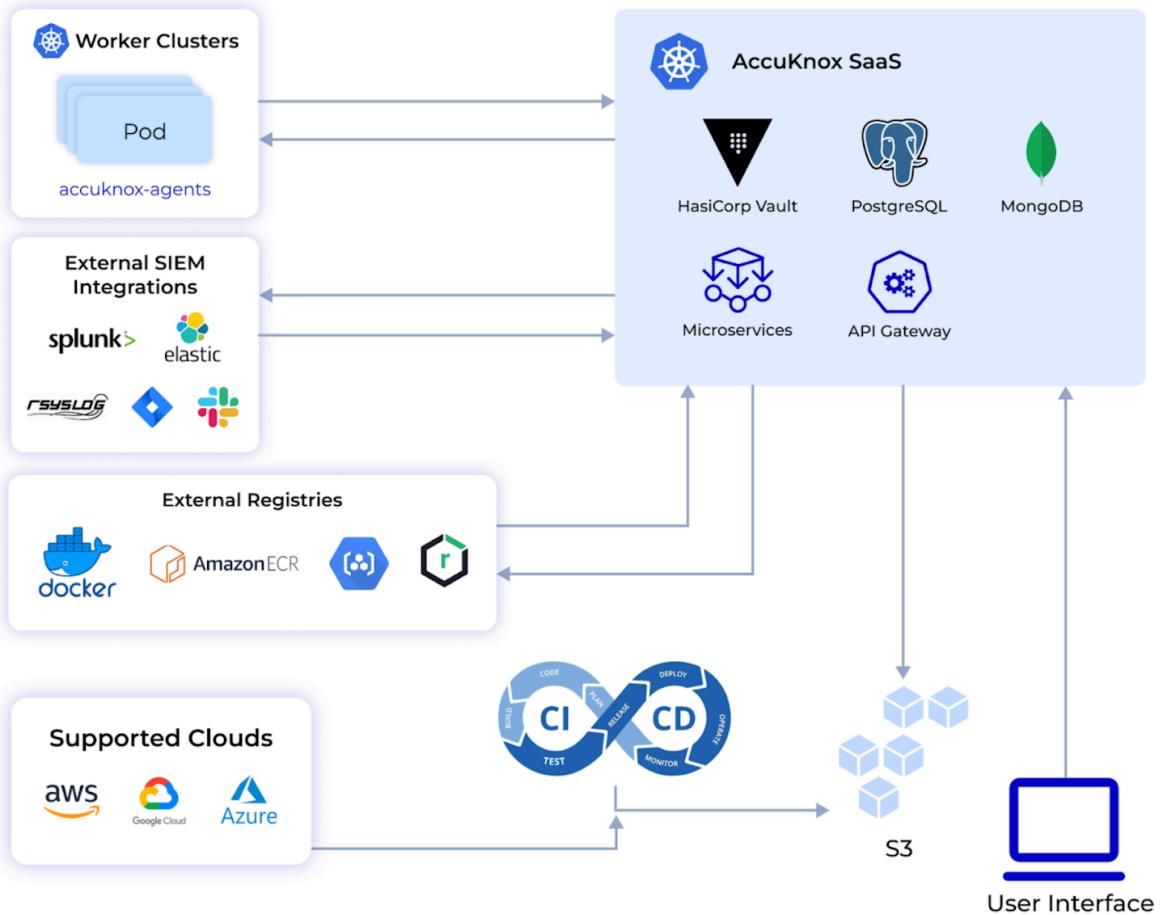


## Key Components

1. Playbook job scheduling: Microservices (Divy), Kueue scheduler, Celery tasks
2. Parser jobs for asset + findings database
3. Alerts and telemetry handling via RabbitMQ
4. SPIFFE-based secure cluster onboarding
5. Storage layer: RDS, MongoDB, Neo4j
6. External integrations & triggers handling

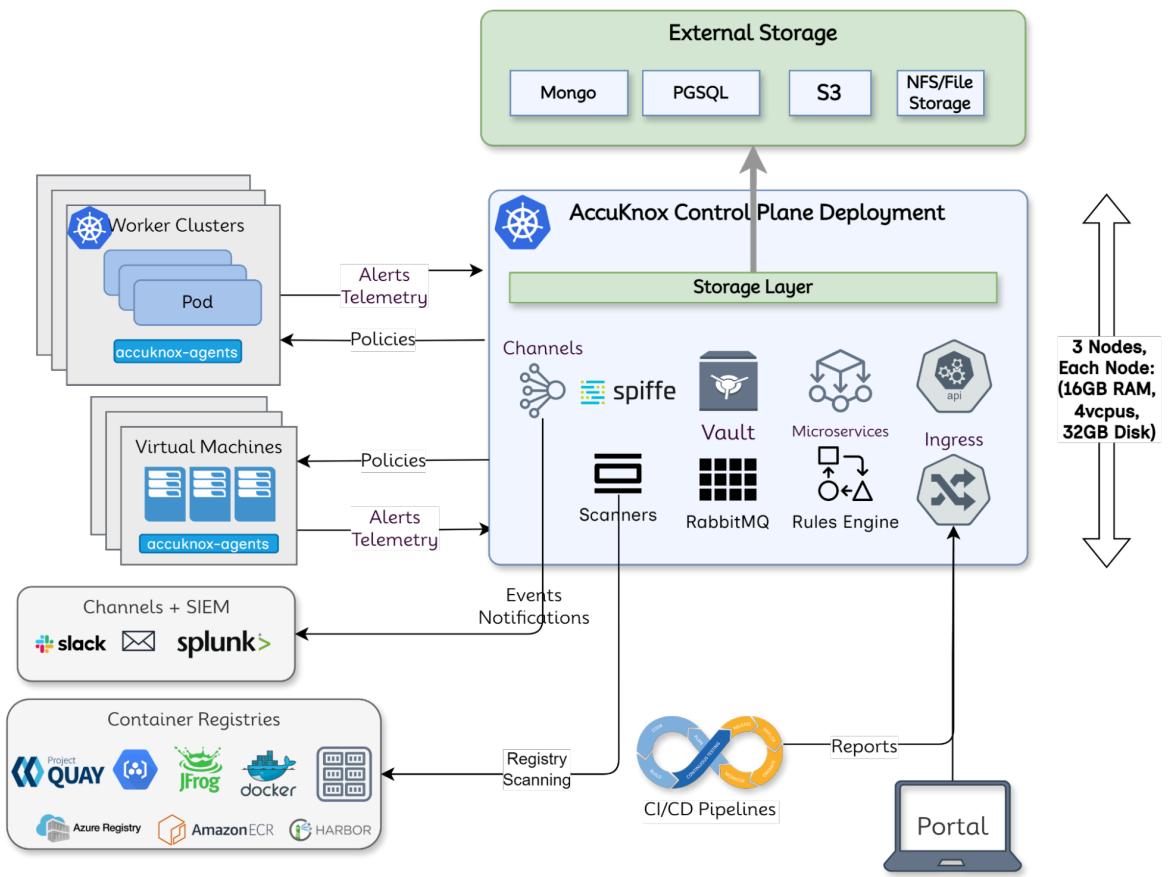
## Cloud Architecture

## ACCUKNOX Enterprise Architecture



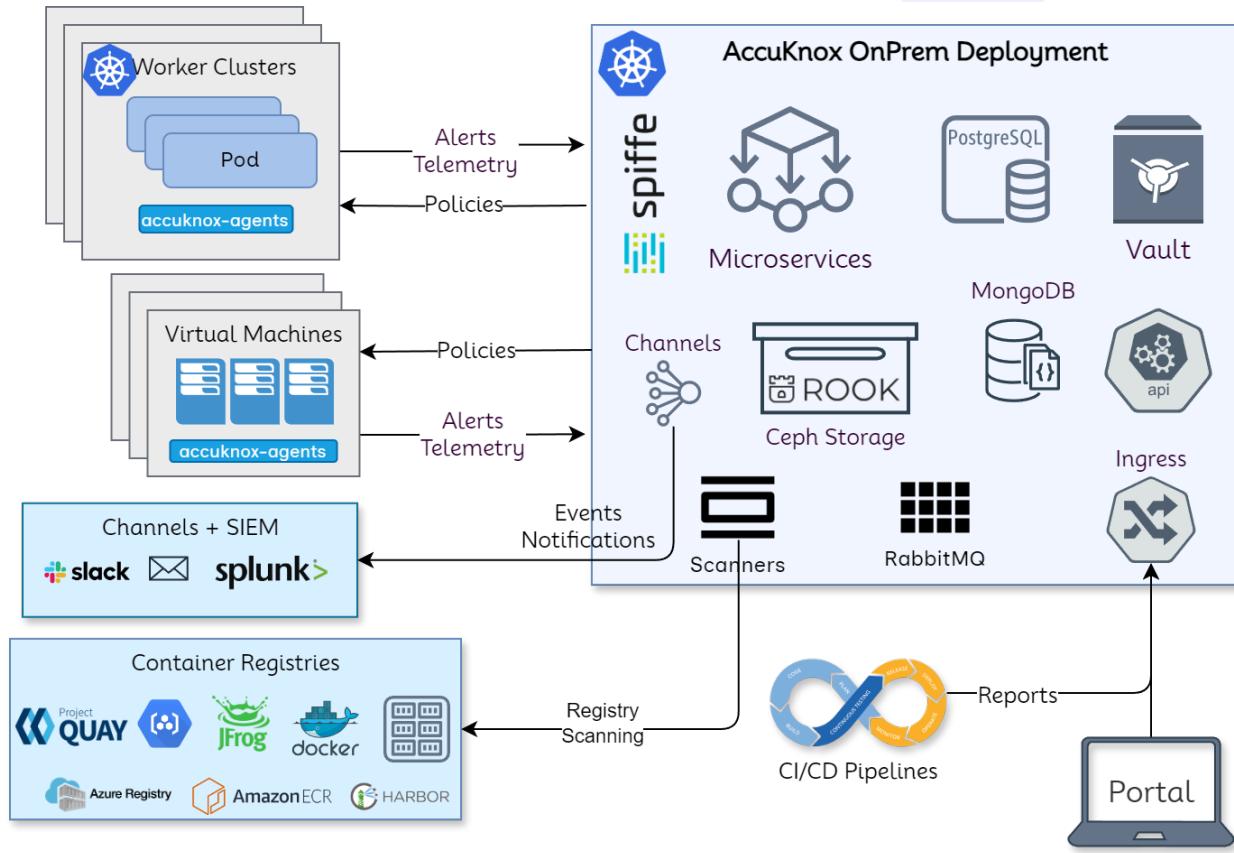
- SaaS and On-Prem support identical services (except AskADA AI Copilot – SaaS only)
- Tenant-level feature control
- Models:
  - SaaS: AWS-managed (Aurora, S3)
  - On-Prem: Full in-cluster setup (for air-gapped environments)
  - Externalized: Uses customer DB/storage

## Externalized Storage Architecture



- Supports deployments with customer-managed storage
- Enables hybrid cloud use cases
- Flexible DB integration (e.g., existing RDS, MongoDB, etc.)

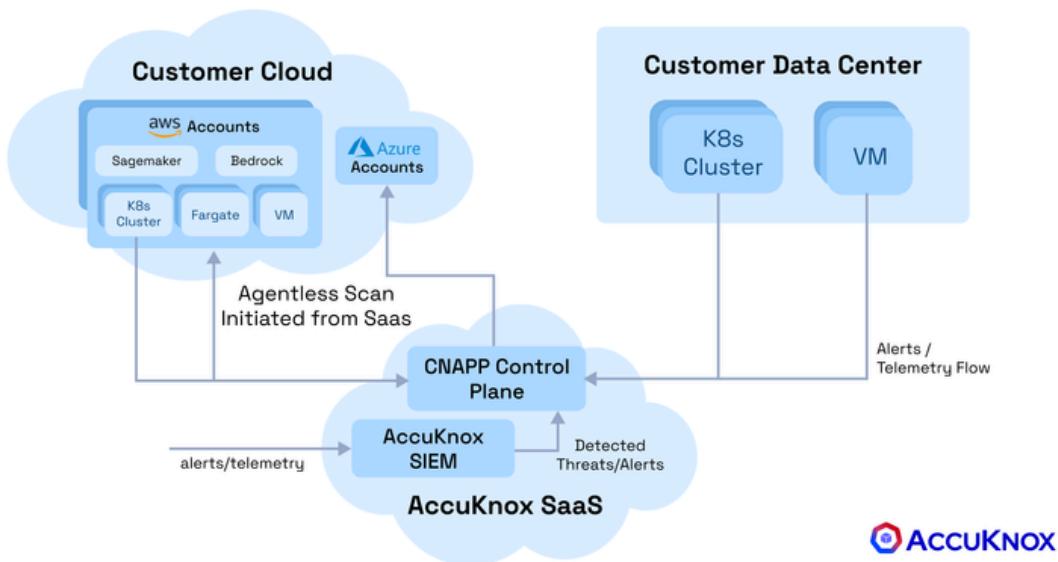
## On-Premises Deployment Architecture



- K8s-native deployment
- No reliance on AWS managed services
- Designed for high-security & compliance environments

[Deployment Details →](#)

## Scaling Considerations



## Key Choke Points

1. **Playbook Jobs:** One AWS account = 272 jobs across regions
  - Kueue ensures tenant-aware resource allocation
2. **Parser Jobs:** Celery tasks parse reports & update DB
3. **Telemetry Overload:** Managed via thresholds & redirection to SIEM

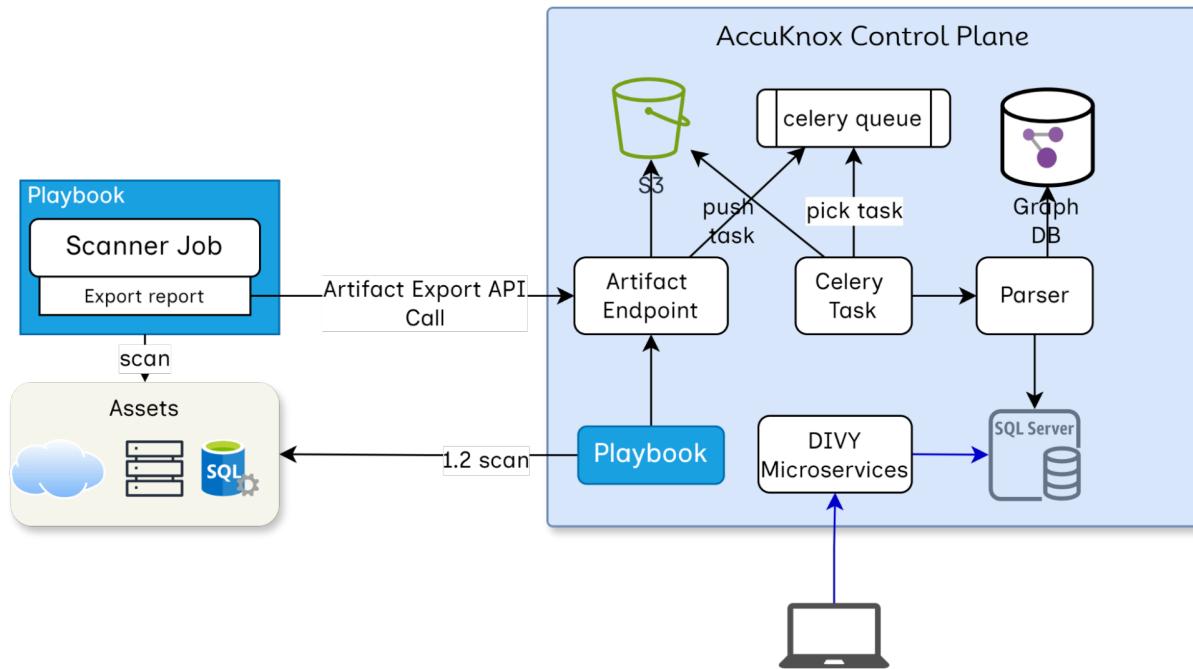
## Noisy Neighbor Mitigation

- Celery replicated per tenant (currently manual)
- Kueue isolates playbook jobs per tenant
- RMQ overload handled by telemetry offload

## Log & Data Storage

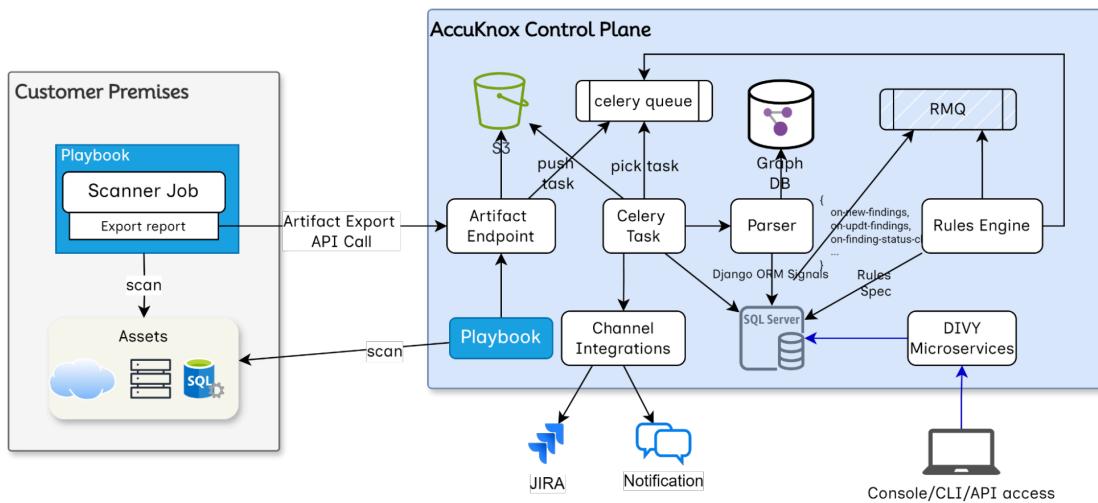
- **RDS:** CSPM, KSPM, ASPM (per-tenant tables)
- **MongoDB:** Telemetry logs (per-tenant collections)
- **Neo4j:** GraphDB for metadata (KIEM), expanding to assets/findings in v3.0

## Customer Data Flow



1. Playbook execution (on-prem or SaaS)
2. Report generated (assets/findings JSON)
3. Sent to control plane via Artifact API (token-based)
4. Saved in S3 + Celery task triggered
5. Celery pulls from S3 and parses
6. DB + Graph updated
7. UI fetches via Divy APIs

## Rules Engine Architecture



- Parser emits events → Rules Engine evaluates
- Tenant-specific rule specs evaluated
- Actions (e.g., notifications, tickets) sent as Celery tasks
- Fully asynchronous, scalable via queues

## Integrations Architecture

### Integration Matrix



**70+**  
Integrations



- **CLI-based:** TruffleHog, Sonarqube, Trivy, Zap, Kubebench
- **API-based:** Checkmarx, Nessus
- **SIEM:** One-way push (e.g., Splunk, Sentinel)

- **Ticketing:** Bidirectional (e.g., Jira, ServiceNow)

## Integration Timelines

- CLI-based: 1 sprint
- API-based: 2–3 weeks
- SIEM: 1 sprint
- Ticketing: 3–5 sprints

[Explore Integrations →](#)

## Compliance Frameworks

The screenshot shows the AccuKnox platform's compliance reporting feature. On the left is a dark sidebar with navigation links: Dashboard, Runtime Protection, Issues, Runtime Protection, Collectors, Monitors / Alerts, Identity, Reports, Notifications, and Settings. The main area is titled 'Compliance' and features a search bar at the top right. Below the search bar are filters for 'All Clouds', 'All Sources', and 'All Regions'. A blue button labeled 'ADD CUSTOM COMPLIANCE' is also present. The page displays three compliance reports in cards:

- CIS Kubernetes 3.0.0**: Compliance Score 95% (95% up from 92%), Controls Passed 80% (8/10), Rules Passed 20% (10/50). Top Controls by Rule Failures: Install-and-Maintain-Network-Security-Controls (100), Apply-Secure-Configurations-to-All-System-Components (80), Protect-Stored-Account-Data (75), Log\_and\_Monitor\_All\_Access (60), Identify\_Users\_and\_Authenticate\_Access\_to\_System (50).
- CIS EKS 1.4.0**: Compliance Score 45% (45% up from 32%), Controls Passed 40% (4/10), Rules Passed 30% (3/10). Top Controls by Rule Failures: Install-and-Maintain-Network-Security-Controls (5), Encryption (20), Private Access to Control Plane (35), VPC Flow Logs (50), Least Privilege Policies (75).
- CIS AWS 2.1.0**: Compliance Score 82% (85% up from 88%), Controls Passed 50% (5/10), Rules Passed 90% (90/100). Top Controls by Rule Failures: Multi-Factor Authentication (MFA) (85), Root Account Usage (95), IAM User Management (60), Access Key Management (90), Resilience and Backup Requirements (82).

Each card has 'OVERVIEW' and 'FINDINGS' buttons.

Supports over 30 regulatory standards, including:

- **General:** ISO 27001, PCI DSS, SOC2.
- **Industry-Specific:** HIPAA, GDPR.

## Additional Resources

- Deployment Models
- Integrations Playbook
- Telemetry Logs
- On-Prem Installation Guide

## Info

AccuKnox offers rapid protection for Kubernetes and other cloud workloads using Kernel Native Primitives like AppArmor, SELinux, and eBPF. For assistance in planning your cloud security strategy, feel free to reach out.

## Getting Started With Technical Support

AccuKnox has active support teams across global regions. The Technical Support team is highly skilled in AccuKnox products and understands customer needs.

As a customer with AccuKnox Support, you're entitled to a number of predetermined technical support contacts who can help debug critical issues and provide solutions. These contacts must be specifically named individuals.

You can:

- Create support cases
- Search the [AccuKnox Knowledge Base](#)
- Review product documentation

## Roles and Responsibilities

Role	Description
<b>Customer</b>	<ul style="list-style-type: none"><li>• Communicate business impacts of technical issues</li><li>• Provide logs, debug data, diagnostic files, etc.</li><li>• Respond timely to information or follow-up requests</li><li>• Engage internal teams as needed</li><li>• Have internet access for meetings</li></ul>

<b>AccuKnox Solutions Engineer</b>	<ul style="list-style-type: none"> <li>• Understand business impact</li> <li>• Provide technical product expertise</li> <li>• Troubleshoot and resolve issues</li> <li>• Share timely status updates</li> </ul>
<b>AccuKnox Technical Support Manager</b>	<ul style="list-style-type: none"> <li>• Ensure high-level technical expertise in Support</li> <li>• Monitor critical issues</li> </ul>
<b>AccuKnox Customer Success Manager</b>	<ul style="list-style-type: none"> <li>• Understand customer requirements</li> <li>• Recommend matching AccuKnox solutions</li> </ul>

## Product Documentation

-  [AccuKnox Help Center](#)
-  [Certification & Training](#): On-demand and instructor-led sessions to enable your team

## Email Support and Procedures

- Email: **support@accuknox.com**
- Or raise a support ticket via: AccuKnox Support Portal

### Note:

- o First-time users must sign up via Jira
- o Try Incognito Mode if you face access issues
- o Support responds within **<24 working hours**

## Support Workflow

- Once a ticket is created, users can track the status via ticket ID

## Priority Levels

Technical Priority	Description
<b>P1 - Critical</b>	Product is completely non-functional; critical business impact
<b>P2 - High</b>	Product is severely degraded; severe business impact

<b>P3 - Medium</b>	General errors; business still functional
<b>P4 - Informational</b>	Assistance or basic info; minimal/no business impact

 Related article: *Technical Support Case Priorities* — visit the Knowledge Base for examples.

## Case Information Required

Please have the following information ready when submitting a case:

1. Contact Name and Organization
2. Business Impact and project context
3. Affected Product
4. Priority Level
5. Relevant screenshots, logs, diagnostic files
6. Was it working before? When did it break? Any changes?
7. Error messages (if any)
8. Frequency and timing of the issue

Technical Support may ask for further info or coordinate with your technical team to isolate known issues.

## Video Conferencing Options

- AccuKnox may initiate Zoom or Google Meet sessions.
- Sessions are scheduled for 30 minutes with a predefined agenda.
- If you're >5 minutes late or absent, the session may be rescheduled.
- Live troubleshooting will follow the session.

## Case Resolution

A case is considered resolved when one of the following is provided:

- Official product behavior documentation
- A verified workaround
- A software update/patch
- A fix in documentation

## Case Closure

A case is closed when:

- Customer confirms the resolution, or
- There's no response for a reasonable period

In rare cases (e.g., customer unresponsiveness or unprofessional behavior), AccuKnox may close the case independently.

Closed cases may be **reopened within 3 days**.

---

## Resources

- [On-Prem Deployment Guide](#)
  - [Help Portal](#)
- 

## FAQs

### 1. Can we engage on a messaging stream for continuous support?

Yes, we can create a **temporary Slack channel** for real-time communication.

### 2. What are the system requirements for On-Prem deployment?

Node s	vCPU s	RAM (GB)	Disk (GB)
4	8	32	256
5	4	16	128

### 3. Is a completely air-gapped On-Prem environment supported?

 Yes, AccuKnox fully supports **air-gapped** environments.

#### 4. How do upgrades work and how frequently are updates released?

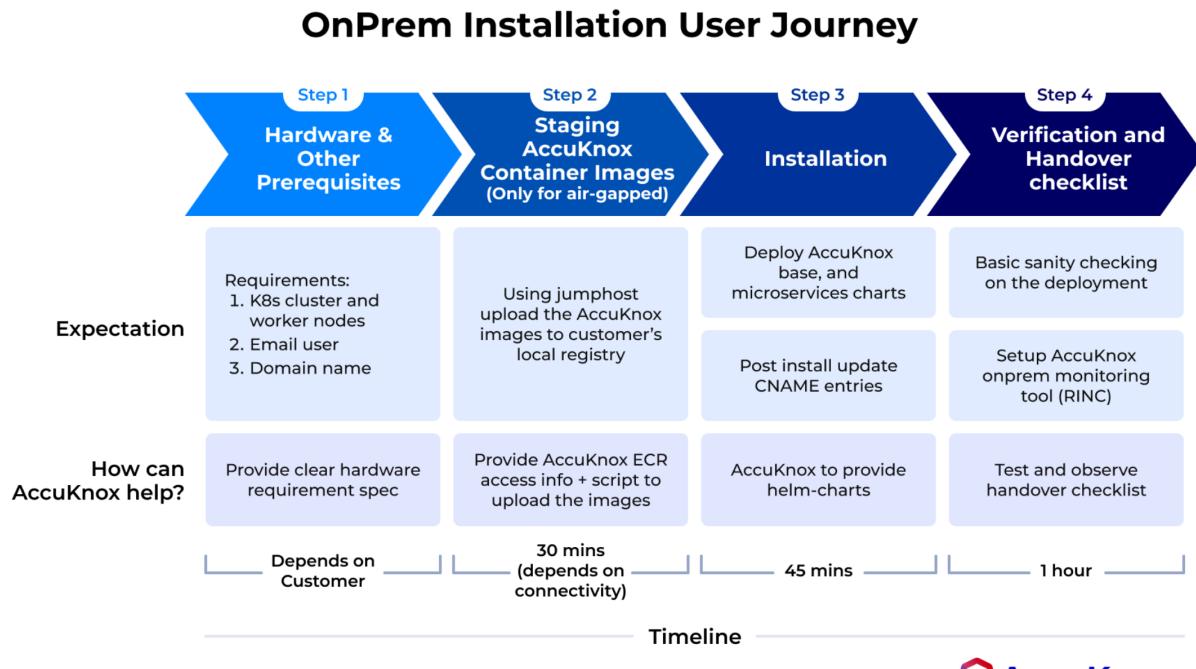
- Software updates are released **monthly**
- Latest package is shared with installation instructions
- AccuKnox Engineering/DevSecOps teams are available to assist if required

 [Release Notes and FAQs](#)

## AccuKnox OnPrem Deployment Guide

### Onboarding Steps for AccuKnox

The onboarding process for AccuKnox's on-prem security solution consists of four key steps that the user must complete. Let's go through each step in a thorough, step-by-step manner:



## Step 1: Hardware & Prerequisites

- Verify hardware, email user, and domain configurations.
- Ensure your environment meets all requirements.
- Time estimate: **Varies**, allocate sufficient time for review and adjustments.

## Step 2: Staging AccuKnox Container Images (*For airgapped environments only*)

- Stage AccuKnox container images in the airgapped setup.
- Reconfirm hardware, email user, and domain requirements.
- Time estimate: **~1 hour**.

## Step 3: Installation

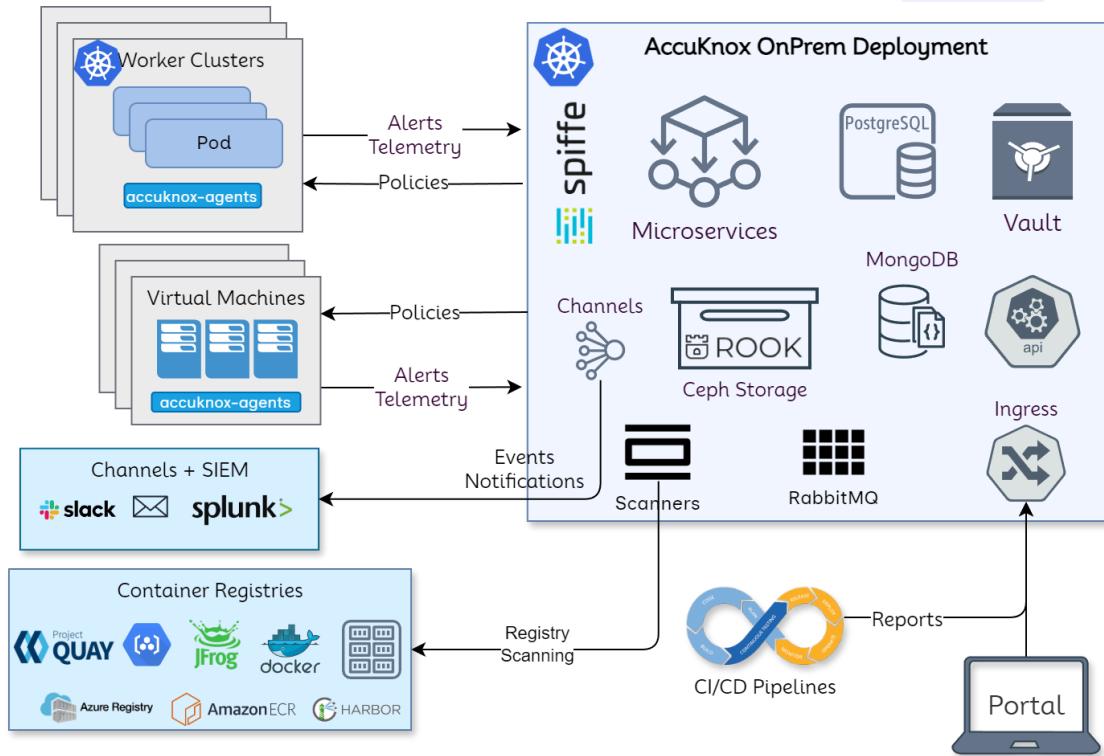
- Install the AccuKnox system within your environment.
- Ensure all prerequisites remain satisfied.
- Time estimate: **~45 minutes**.

## Step 4: Verification/Validation

- Confirm all previous steps were completed successfully.
- Validate hardware, email user, and domain configurations.
- Time estimate: **~1 hour**.

AccuKnox onprem deployment is based on Kubernetes native architecture.

### High-Level Architecture Overview



AccuKnox onprem deployment is based on Kubernetes native architecture.

## AccuKnox OnPrem k8s components

### Microservices

Microservices implement the API logic and provide the corresponding service endpoints. AccuKnox uses Golang-based microservices for handling streaming data (such as alerts and telemetry) and Python-based microservices for other control-plane services.

### Databases

PostgreSQL is used as a relational database and MongoDB is used for storing JSON events such as alerts and telemetry. Ceph storage is used to keep periodic scanned reports and the Ceph storage is deployed and managed using the Rook storage operator.

### Secrets Management

Within the on-prem setup, there are several cases where sensitive data and credentials have to be stored. Hashicorp's Vault is used to store internal (such as DB username/password) and user secrets (such as registry tokens). The authorization is

managed purely using the k8s native model of service accounts. Every microservice has its service account and uses its service account token automounted by k8s to authenticate and subsequently authorize access to the secrets.

## Scaling

K8s native horizontal and vertical pod autoscaling is enabled for most microservices with upper limits for resource requirements.

## AccuKnox-Agents

Agents need to be deployed in target k8s clusters and virtual machines that have to be secured at runtime and to get workload forensics. Agents use Linux native technologies such as eBPF for workload telemetry and LSMs (Linux Security Modules) for preventing attacks/unknown execution in the target workloads. The security policies are orchestrated from the AccuKnox onprem control plane. AccuKnox leverages SIFFE/SPIRE for workload/node attestation and certificate provisioning. This ensures that the credentials are not hardcoded and automatically rotated. This also ensures that if the cluster/virtual machine has to be deboarded then the control lies with the AccuKnox control plane.

## System Requirements

### Worker Node Requirements

Nodes	vCPUs	RAM (GB)	Disk (GB)
6	4	16	256

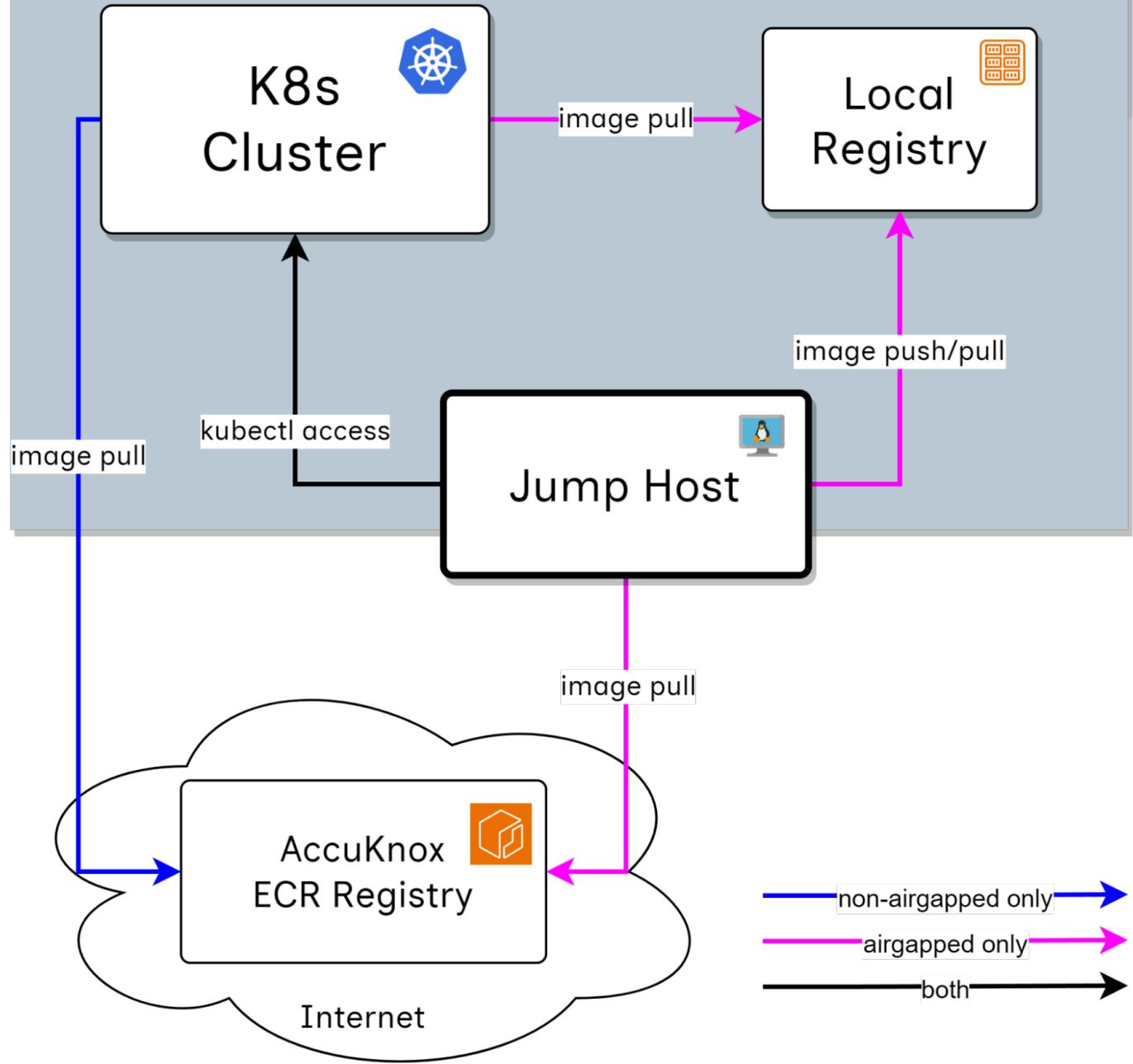
## Kubernetes Requirements

- **Ingress Controller (load balancers)**
  - For access to the application
- **Persistent Volumes (PV), provisioner/controller (block device/disks)**

- Used as data storage for SQL, MongoDB, scanned artifacts
- Other internal app usages
- **DNS CNAME provisioning**
  - Needed for application access & communication
  - Certs would use this CNAME so that address changes won't impact the cert validation
- **Email account configuration**
  - Need email username, and password
  - Used for user sign-in, password change, scan notification, sending reports

## Jump Host

## OnPrem Air-Gapped env



Jump Host Pre-requisites

<b>Tool</b>	<b>Version</b>	<b>Install Command</b>
jq	1.6	<code>apt install jq</code>
unzip	x.x	<code>apt install unzip</code>
yq	v4.40.x	<code>VERSION=v4.40.5 &amp;&amp; BINARY=yq_linux_amd64 &amp;&amp; wget https://github.com/mikefarah/yq/releases/download/\${VERSION}/\${BINARY}.tar.gz -O -   tar xz &amp;&amp; mv \${BINARY} /usr/bin/yq</code>
helm	v3.x.x	<code>curl -s https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3   bash</code>
kubectl	Supported by your k8s cluster	-
aws	v2	<code>curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" &amp;&amp; unzip awscliv2.zip &amp;&amp; sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update</code>
docker	v20.xx	<code>apt install docker.io</code>

---

Storage	80GB	-
---------	------	---

## Installation Steps

- Onprem Deployment Installation Document (this document)
- Helm charts archive
- Kubectl and Helm tools are pre-requisite tools for using these helm charts

## Use the following commands

```
tar xvf accuknox-helm-charts.tgz  
cd Helm-charts
```

## Use of Private/Local Container Registry (or air-gapped mode)

If you want to use your private/local registry as the exclusive source of images for the entire cluster, please install the accuknox-onprem-mgr component first.

---

<b>Value</b>	<b>Description</b>	<b>Provider</b>
registry.us ername	Registry User	Customer

---

---

registry.password	Registry Password	Customer
registry.address	The registry server address	Customer
ecr.user	Credential to pull images from AccuKnox registry	AccuKnox
ecr.password	Credential to pull images from AccuKnox registry	AccuKnox

---

```
cd airgapped-reg
```

```
# configure aws cli with AccuKnox provided secrets
aws configure

# connect to docker Accuknox docker registry
aws ecr get-login-password --region us-east-2 | docker login --username AWS
--password-stdin 956994857092.dkr.ecr.us-east-2.amazonaws.com

# connect to airgapped registry
docker login <registry_address>

# upload images to private registry
./upload_images.sh <registry_address>
./upload_onboarding_images.sh <registry.address>

# upload helm charts to private registry
./upload_helm.sh <registry.address>

# create a namespace
MGR_NS="accuknox-onprem-mgr"
CERT_MGR_NS="cert-manager"
kubectl create ns $MGR_NS
kubectl create ns $CERT_MGR_NS
```

```

kubectl create secret docker-registry airgapped-reg --docker-server=<registry.address>
--docker-username=<registry.username> --docker-password=<registry.password> -n
$MGR_NS

kubectl create secret docker-registry airgapped-reg --docker-server=<registry.address>
--docker-username=<registry.username> --docker-password=<registry.password> -n
$CERT_MGR_NS

# <registry_address> can include port as well

./install-certmanager.sh <registry_address>

./install-onprem-mgr.sh <registry_address>

kubectl apply -k .
kubectl apply -f onprem-mgr.yaml

```

## Update the override-values.yaml

[ONLY FOR air-gapped/private registry ENVIRONMENT]: Set global.onprem.airgapped to true in override-values.yaml file.

## Before you start

- set your domain name in the override values by changing by your domain
- set your ssl preferences in the override values by changing the ssl block
- If you wish to bring in your own MongoDB, PostgreSQL, NFS share or S3, disable global.postgres.airgapped and global.mongodb.enabled rookceph.enabled in override-values.yaml.

### If the environment is OpenShift then set:

```

global:
  platform: "openshift"

```

### If environment is airgapped or using private registry make ssl.certmanager.install:"false"

```

ssl:
  certmanager:
    install: false

```

## Auto-generated self-signed certificate

We auto generate the needed self signed certificates for the client. To enable this option, the ssl section the override values file should be set as follow:

```
ssl:  
  selfsigned: true  
  customcerts: false
```

## Certificate signed by a known authority

The client provides a certificate signed by a known signing authority To enable this option, the ssl section the override values file should be set as follow:

```
ssl:  
  selfsigned: false  
  customcerts: true
```

## Self-signed certificates (provided by the customer)

The client provides a self signed certificate. To enable this option, the ssl section the override values file should be set as follow:

```
ssl:  
  selfsigned: true  
  customcerts: true
```

AccuKnox installation package will contain override-values.yaml file that contains installation-specific options to be configured.

1. override to your domain
2. set your ssl preferences in the override values by changing the ssl block.

## Install AccuKnox base dependencies

```
kubectl create namespace accuknox-chart  
helm upgrade --install -n accuknox-chart accuknox-base accuknox-base-chart  
--create-namespace -f override-values.yaml
```

## IMPORTANT

Some resources deployed in the above step require some time to provision. If the user executes the next command without waiting for the proper provisioning of the previous command the installation may break and will need to start over.

Run the below script to make sure that the provisioning was done successfully.

```
while true
do
    status=$(kubectl get cephcluster -n accuknox-ceph rook-ceph
-o=jsonpath='{.status.phase}')
    [[$(echo $status | grep -v Ready | wc -l) -eq 0]] && echo "You can proceed" && break
    echo "wait for initialization"
    sleep 1
done
```

## Install AccuKnox pre-chart

### IMPORTANT

Contact your AccuKnox representative to acquire the credentials for `ecr.user` and `ecr.password` values.

Value	Description	Provider
<b>email.user</b>	Email user will send signup invites, reports, etc.	Customer
<b>email.password</b>	Email Password	Customer

---

<b>email.host</b>	The Email server address	Customer
<b>email.from</b>	The Email sender address ( <a href="mailto:noreply@domain.com">noreply@domain.com</a> )	Customer
<b>ecr.user</b>	Credential to pull images from AccuKnox registry	AccuKnox
<b>ecr.password</b>	Credential to pull images from AccuKnox registry	AccuKnox
<b>global.externalServices.p ostgres.user</b>	Postgres username, if using an external DB	Customer
<b>global.externalServices.p ostgres.password</b>	Postgres password, if using an external DB	Customer
<b>global.externalServices.p ostgres.host</b>	Postgres host, if using an external DB	Customer
<b>global.externalServices. mongo.user</b>	Mongodb username, if using an external DB	Customer

---

---

<b>global.externalServices.mongo.password</b>	Mongodb password, if using an external DB	Customer
<b>global.externalServices.mongo.host</b>	Mongodb host, if using an external DB	Customer
<b>global.externalServices.nfs.server</b>	NFS server address	Customer
<b>global.externalServices.s3.host</b>	S3 datastore host	Customer
<b>global.externalServices.s3.port</b>	S3 datastore port	Customer
<b>global.externalServices.s3.accessKey</b>	S3 access key	Customer
<b>global.externalServices.s3.secretKey</b>	S3 secret access key	Customer
<b>global.externalServices.s3.bucket</b>	S3 bucket name	Customer

---

```
helm upgrade --install -n accuknox-chart accuknox-pre pre-chart --create-namespace -f  
override-values.yaml --set global.email.from="" --set global.email.user="" --set  
global.email.password="" --set global.email.host="" --set ecr.user="" --set ecr.password=""
```

Or, if using an external PostgreSQL or Mongo DB,

```
helm upgrade accuknox-pre pre-chart \  
--install \  
-namespace accuknox-chart \  
--create-namespace \  
-values override-values.yaml \  
--set global.email.user="" \  
--set global.email.password="" \  
--set global.email.host="" \  
--set ecr.user="" \  
--set ecr.password="" \  
--set global.externalServices.postgres.user="" \  
--set global.externalServices.postgres.password="" \  
--set global.externalServices.postgres.host="" \  
--set global.externalServices.mongo.user="" \  
--set global.externalServices.mongo.password="" \  
--set global.externalServices.mongo.host=""
```

## Install AccuKnox microservices chart

Value	Description	Provider
<b>email.us er</b>	Email user will send signup invites, reports, etc.	Custom er
<b>email.pa ssword</b>	Email Password	Custom er

---

<b>email.host</b>	The Email server address	Customer
-------------------	--------------------------	----------

---

<b>email.from</b>	The Email sender address (e.g., <a href="mailto:noreply@domain.com">noreply@domain.com</a> )	Customer
-------------------	---	----------

```
helm upgrade --install -n accuknox-chart accuknox-microservice  
accuknox-microservice-chart --set global.email.user="" --set global.email.from="" --set  
global.email.password="" --set global.email.host="" --create-namespace -f  
override-values.yaml
```

## DNS Mapping

Run the following script to generate the records you should add to your DNS zone.

```
./generate_dns_entries.sh
```

## Installing certificates

### Certificates signed by known authority

```
./install_certs.sh <certificate_path> <certificate_key_path> <ca_path>
```

### Self-signed certificates (provided by customer)

Install nginx ingress (if any other self-managed Kubernetes)

1. Install the nginx ingress chart

```
cd airgapped-reg/addons
```

```
helm upgrade --install ingress-nginx ingress-nginx \
--repo https://kubernetes.github.io/ingress-nginx \
--namespace ingress-nginx --create-namespace \
--version 4.11.2 -f ingress-nginx.yaml
```

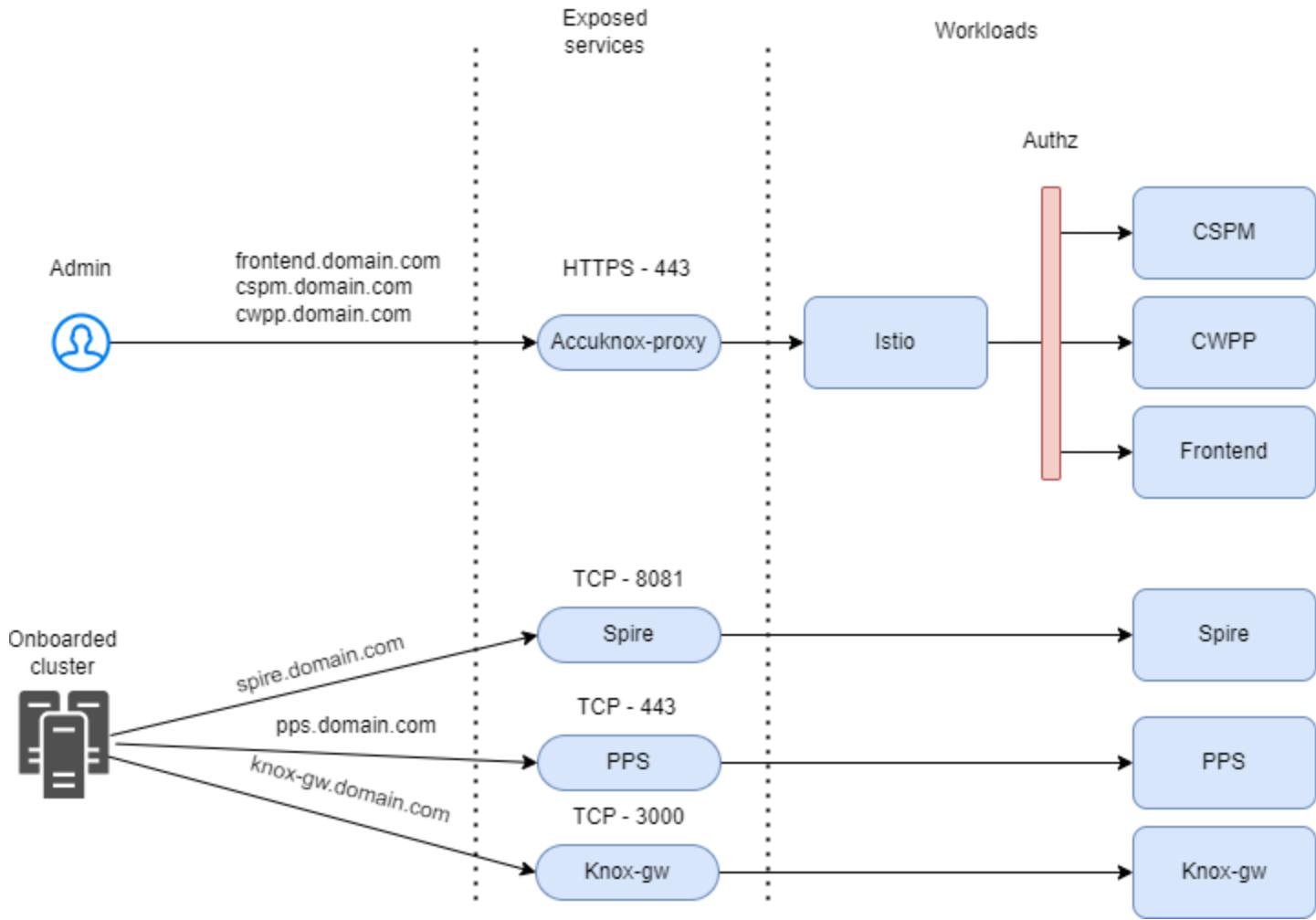
1. Update the domains in ingress.yaml and apply it

```
kubectl apply -f ingress.yaml
```

## Verification of installation

After successful installation, you should be able to access the following URLs:

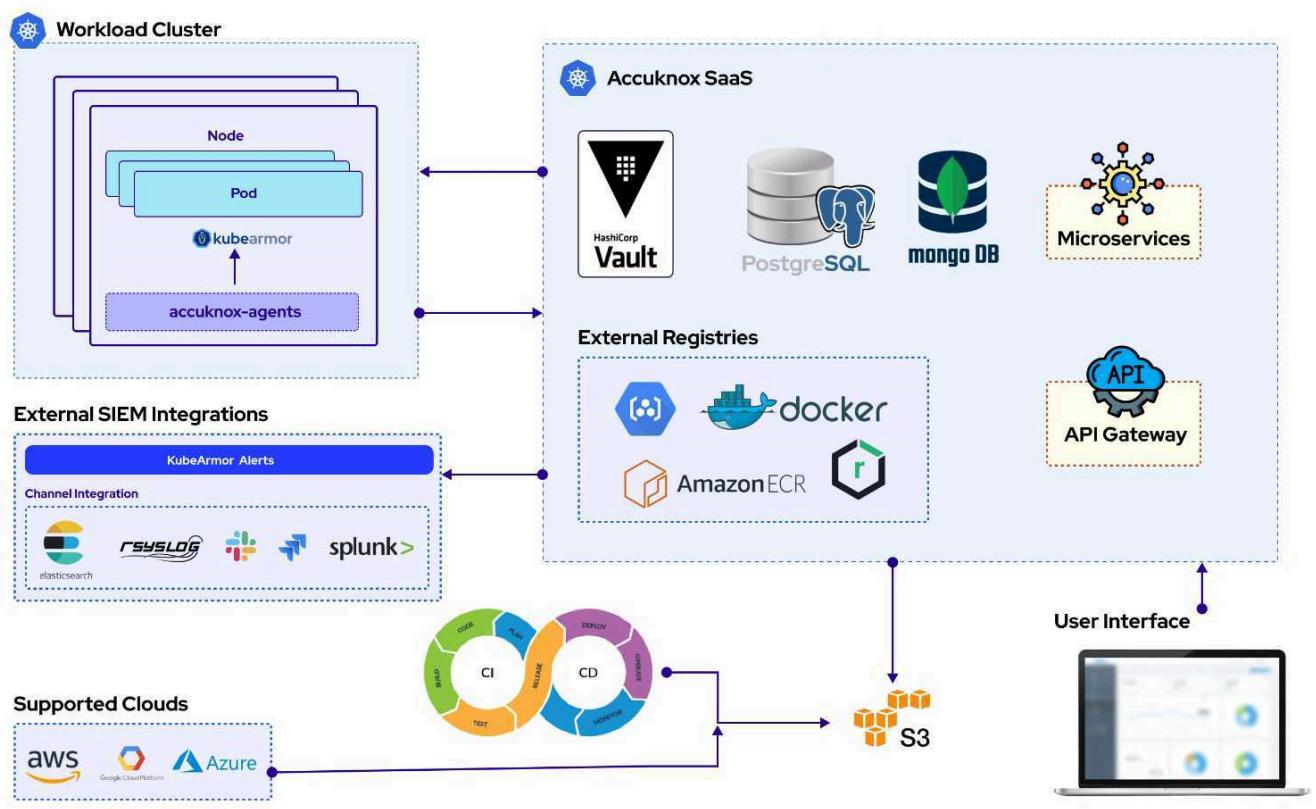
- <https://frontend.<your-domain.com>> — Access the **Sign-in page**.
- <https://cspm.<your-domain.com>/admin/> — Access the **CSPM Admin page**.
- <https://cwpp.<your-domain.com>/cm/> — Access the **CWPP Configuration Management page**.



## Runtime Security Prerequisites

In SaaS model of deployment the AccuKnox CNAPP will be hosted in our cloud environment and the agents deployed on the workloads will connect with the SaaS.

# ACCUKNOX Enterprise Architecture



## AccuKnox Agents

### Deployments

KubeArmor

### Deployment Type

DaemonSet

---

Shared  
Informer  
Agent

---

Deployment

Feeder  
Service

---

Deployment

Policy  
Enforcemen  
t

---

Deployment

Discovery  
Engine  
Agent

---

Deployment

- It is assumed that the user has some basic familiarity with Kubernetes, kubectl and helm. It also assumes that you are familiar with the AccuKnox opensource tool workflow. If you're new to AccuKnox itself, refer first to [opensource installation](#)
- It is recommended to have the following configured before onboarding:
  - a. [Kubectl](#)
  - b. [Helm](#)

## Pre-requisites

## Minimum Resource required

<b>Deployments</b>	<b>Resource Usage</b>	<b>Ports</b>	<b>Connection Type</b>	<b>AccuKnox Endpoint</b>
KubeArmor	CPU: 200 m, Memory: 200 Mi	-	-	-
Agents Operator	CPU: 50 m, Memory: 50 Mi	8081, 9090	Outbound	*.accuknox.com:8081 --> SPIRE Access  *.accuknox.com:9090 --> SPIRE Health Check
Discovery Engine	CPU: 200 m, Memory: 200 Mi	-	-	-
Shared Informer Agent	CPU: 20 m, Memory: 50 Mi	3000	Outbound	*.accuknox.com:3000 --> knox-gateway
Feeder Service	CPU: 50 m, Memory: 100 Mi	3000	Outbound	*.accuknox.com:3000 --> knox-gateway
Policy Enforcement	CPU: 10 m, Memory: 20 Mi	443	Outbound	*.accuknox.com:443 --> Policy Provider Service

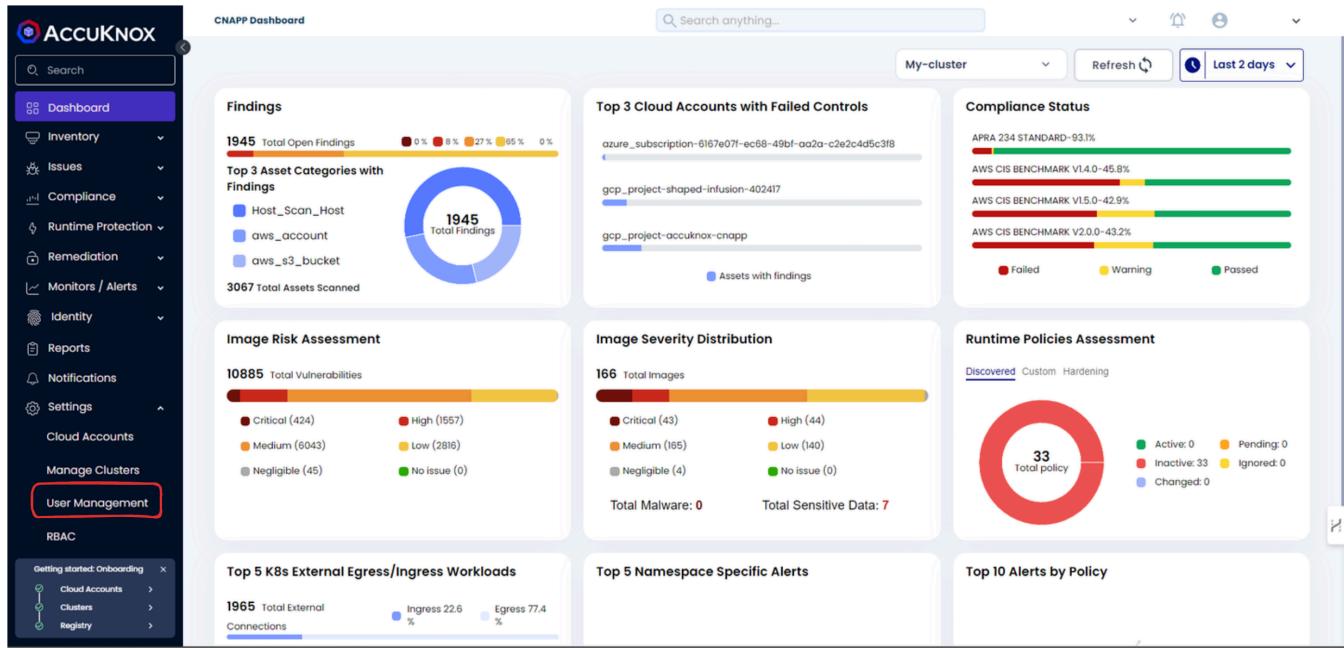
- These ports need to be allowed through firewall.

# SSO Login Guide

This guide covers the complete process from inviting a new user to logging in with SSO.

## 1. Inviting a New User

Log in to your AccuKnox dashboard.



The screenshot shows the AccuKnox CNAPP Dashboard. On the left, a sidebar menu includes 'Dashboard', 'Inventory', 'Issues', 'Compliance', 'Runtime Protection', 'Remediation', 'Monitors / Alerts', 'Identity', 'Reports', 'Notifications', 'Settings' (which is expanded to show 'Cloud Accounts', 'Manage Clusters', and 'User Management'), and 'RBAC'. A red box highlights the 'User Management' link. The main dashboard area has several cards: 'Findings' (1945 Total Open Findings), 'Image Risk Assessment' (10885 Total Vulnerabilities), 'Top 3 Asset Categories with Findings' (Host\_Scan\_Host, aws\_account, aws\_s3\_bucket), 'Top 3 Cloud Accounts with Failed Controls' (azure\_subscription-6167e07f-ec68-49bf-aa2a-c2e2c4d5c3f8, gcp\_project-shaped-infusion-402417, gcp\_project-accuknox-cnapp), 'Compliance Status' (APRA 234 STANDARD - 93.1%, AWS CIS BENCHMARK V1.4.0 - 45.8%, AWS CIS BENCHMARK V1.5.0 - 42.9%, AWS CIS BENCHMARK V2.0.0 - 43.2%), 'Image Severity Distribution' (166 Total Images), 'Runtime Policies Assessment' (33 Total policy), 'Top 5 K8s External Egress/Ingress Workloads' (1965 Total External Connections, Ingress 22.6%, Egress 77.4%), 'Top 5 Namespace Specific Alerts', and 'Top 10 Alerts by Policy'.

Navigate to "User Management" in the left sidebar menu. Click the "User +" button in the top right corner of the Users page.

The screenshot shows the ACCUKNOX interface with the 'User Management' tab selected in the sidebar. The main area displays a table of users in the workspace, with columns for First Name, Last Name, Email, Role, and Recent Login. A red box highlights the 'User +' button in the top right corner of the header.

In the "Invite User" form, fill out the following details and hit send.

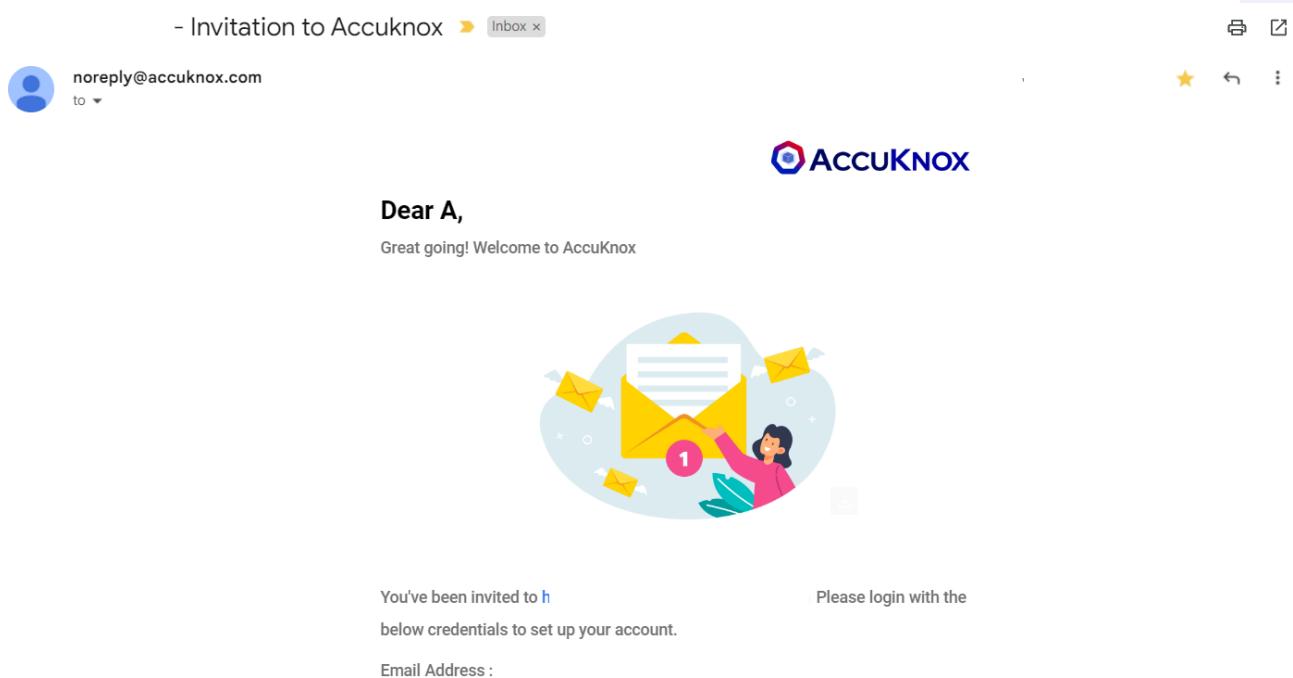
The screenshot shows the 'Invite User' form. It includes fields for Email (test@mailinator.com), First name (Name), Last name (Surname), Role (Editor), Labels (19JUNESS), Groups (aws-1), and an MFA checkbox. A red box highlights the 'Send' button in the bottom right corner.

## Note

You can view pending invitations in the "Pending Invites" tab on the Users page. You can resend or revoke invitations from this tab. Viewing all permissions of a user is possible via the main tab.

## 2. User Receives Invitation

The invited user will receive an email containing a link to accept the invitation and set up their account if they haven't already done so.



## 3. User Login Options

Users can log in to AccuKnox using two methods:

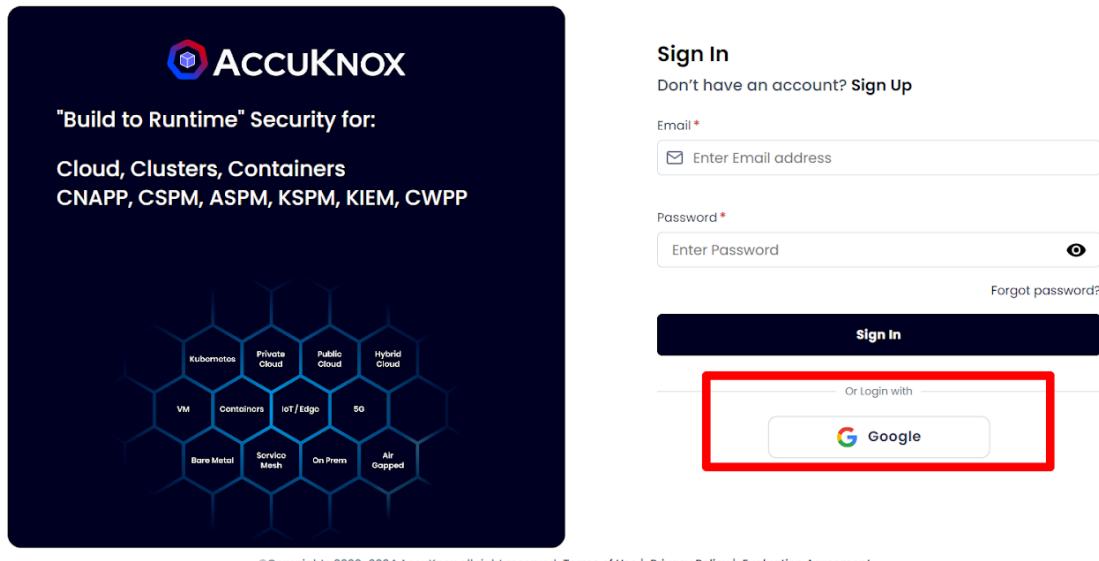
### Option A: Traditional Login

1. Go to the AccuKnox login page.
2. Enter the email address and password.
3. Click "Sign In".

#### Note

This requires you to use the MFA (multi-factor authentication) code if it was enabled during the invitation process. MFA is required for every sign-in attempt.

## Option B: Single Sign-On (SSO) with Google



1. Go to the AccuKnox login page.
2. Look for "Or login with" at the bottom of the form.
3. Click on the "Google" button.
4. If not already signed in to Google, enter Google account credentials.
5. Grant any necessary permissions for AccuKnox.

### Note

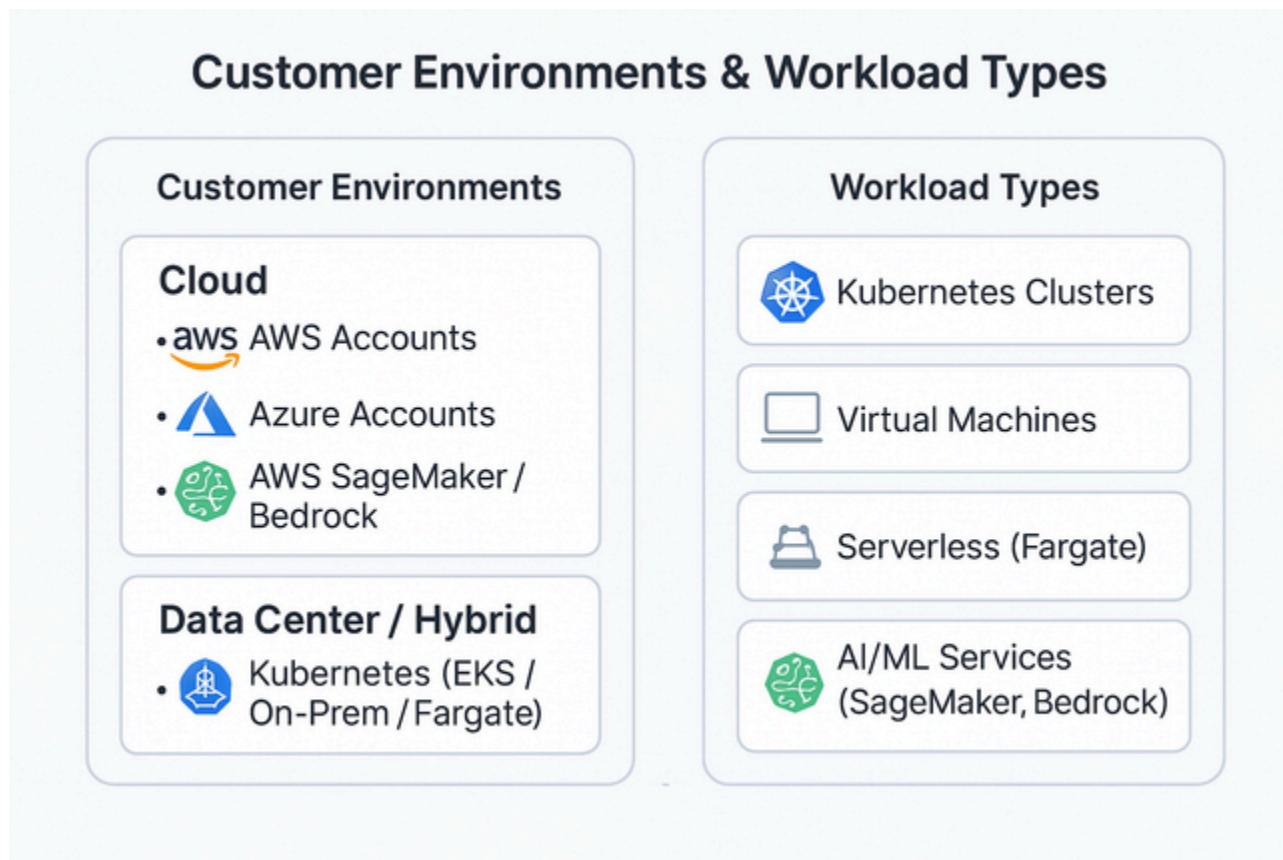
If you are already signed in to Google, you will be automatically logged in to AccuKnox. No need for MFA in this case.

## Notes

- SSO is currently only supported for Google accounts.
- Users must be invited with their Gmail address to use Google SSO.

- For the best experience, use the same email address for invitation and login.
- If you encounter any issues, contact your AccuKnox administrator or support team.
- Emails with + modifiers (e.g., [test+stable@gmail.com](mailto:test+stable@gmail.com) or [example+solutions@gmail.com](mailto:example+solutions@gmail.com)) are not supported for SSO. Please use a base email address.

## Onboarding Assets – High-Level Overview



## Customer Environments

**Cloud:**

- AWS Accounts
- Azure Accounts
- AWS SageMaker / Bedrock

**Data Center / Hybrid:**

- Kubernetes Clusters (EKS / On-Prem / Fargate)
- Virtual Machines (EC2 / On-Prem)

**Workload Types:**

- K8s Clusters
- Virtual Machines
- Serverless (Fargate)
- AI/ML Services (SageMaker, Bedrock)

**Security and Telemetry Flow:**

- Agentless scan initiated from SaaS
  - CNAPP control plane processes telemetry
  - Alerts and detections sent to SIEM
- 

# Cloud Onboarding Options

- Fully Agentless Mode
  - Account/Subscription Onboarding:
  - CloudFormation (recommended)
  - Terraform
  - Manual
  - AWS Organization Unit Onboarding:
  - Using cross-account tenant roles
-

# Kubernetes – AWS EKS / On-Prem / Fargate

## Risk Assessment

- CIS Benchmarks
- Misconfigurations
- KIEM Policies
- Agentless methods:
- Remote scanning via `kubeconfig`
- Kubernetes job-based scanning

## Runtime Security & Hardening

- Helm-based installation
- In-cluster image scanning:
- Operator and job-based deployment via Helm

## Fargate Runtime

- Supported via sidecar model
  - Deployable using Helm or Kubernetes manifests
- 

# Virtual Machines – EC2 / On-Prem

- Misconfiguration scanning via cloud account onboarding (agentless)
  - Risk assessment / STIGs scanning requires lightweight VM agent
-

# Container Registry

## SaaS-Based Scanning

- Registry onboarded via control plane
- Credentials: Username + API Token

## On-Prem Scanning

- Requires AccuKnox collector deployed on VM
  - Local scanning of registries enabled
- 

# AI/ML Workloads – SageMaker / Bedrock

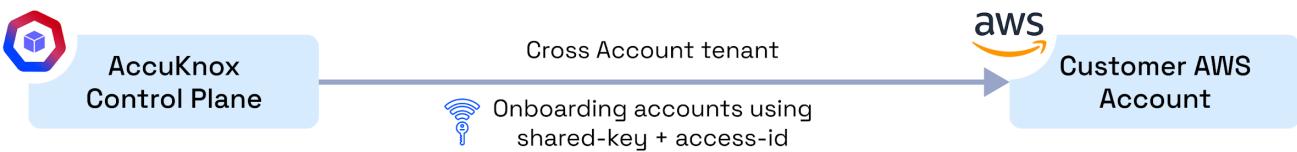
- Fully agentless
  - Selectable during cloud account onboarding:
  - General Cloud Assets
  - General Cloud + AI/ML Assets
- 

# Deployment References

- Separate detailed documentation provided for Helm charts, job configurations, and onboarding automation (CloudFormation, Terraform).

# CSPM Pre-requisite for AWS

When the AccuKnox control plane is hosted in a cloud environment, scanning is performed using Cloud account Readonly Access permissions.



AWS onboarding requires creation of an IAM user. Please follow the following steps to provide a user with appropriate read access:

**Step 1:** Navigate to IAM → Users and click on Add Users

A screenshot of the AWS IAM Users page. The left sidebar shows "Identity and Access Management (IAM)" with "Access management" expanded, showing "User groups" and "Users". The main area shows a table with one row, indicating "No resources to display". The table has columns for "User name", "Groups", "Last activity", "MFA", "Password age", and "Active key age".

**Step 2:** Give a username to identify the user

A screenshot of the "User details" step in the IAM User creation wizard. The "User name" field is filled with "sample-user". Below the field, a note says: "The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)". There is an optional checkbox "Provide user access to the AWS Management Console - optional" with a note: "If you're providing console access to a person, it's a best practice [link] to manage their access in IAM Identity Center". A note at the bottom says: "If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. Learn more [link]". At the bottom right are "Cancel" and "Next" buttons.

**Step 3:** In the "Set Permissions" screen:

- Select "Attach policies directly"
- Search "ReadOnly", Filter by Type: "AWS managed - job function" and select the policy

c. Search "SecurityAudit", Filter by Type: "AWS managed - job function" and select the policy

**Step 4:** Finish creating the user. Click on the newly created user and create the Access key and Secret Key from the Security Credentials tab to be used in the AccuKnox panel

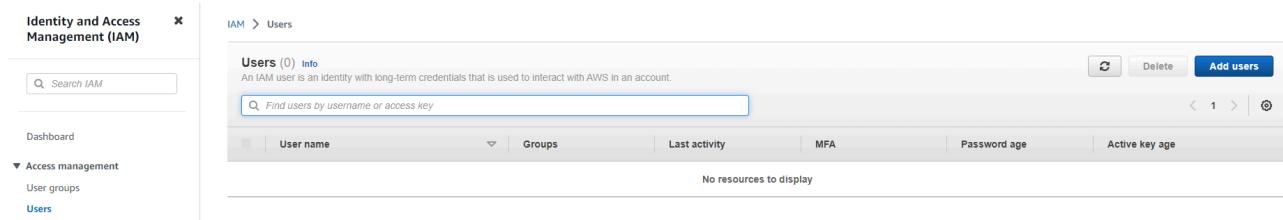
# AWS Account onboarding

In this section we can find the steps to onboard an AWS cloud account to the AccuKnox SaaS platform.

## AWS IAM User Creation

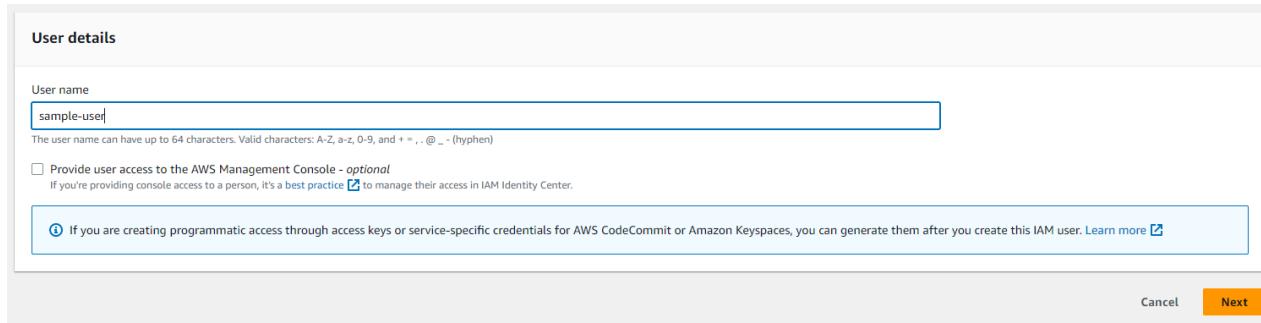
Please follow the following steps to provide a user with appropriate read access:

**Step 1:** Navigate to IAM → Users and click on Add Users



The screenshot shows the AWS IAM 'Users' page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and a 'Search IAM' bar. Under 'Access management', 'User groups' is expanded, and 'Users' is selected. The main area shows a table with columns: 'User name', 'Groups', 'Last activity', 'MFA', 'Password age', and 'Active key age'. A note at the top says 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' Below the table, it says 'No resources to display'. At the top right, there are 'Delete' and 'Add users' buttons, along with navigation icons.

**Step 2:** Give a username to identify the user



The screenshot shows the 'User details' step of the IAM user creation wizard. It has a 'User details' header. Under 'User name', the value 'sample-user' is entered into a field with a note below: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)'. There's an optional checkbox 'Provide user access to the AWS Management Console - optional' with a note: 'If you're providing console access to a person, it's a best practice [link] to manage their access in IAM Identity Center.' A note at the bottom says: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. Learn more [link]'. At the bottom right are 'Cancel' and 'Next' buttons.

**Step 3:** In the "Set Permissions" screen:

- Select "Attach policies directly"
- Search "ReadOnly", Filter by Type: "AWS managed - job function" and select the policy

Step 2  
Set permissions

Step 3  
Review and create

**Permissions options**

- Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1/1116)**  
Choose one or more policies to attach to your new user.

Filter by Type	
<input type="text" value="Q Readonly"/>	AWS managed - job function
1 match	< 1 > ⏪
<input checked="" type="checkbox"/> Policy name <a href="#">Readonly</a> ▲ Type Attached entities	
<input checked="" type="checkbox"/>  ReadOnlyAccess AWS managed - job function 0	

c. Search "SecurityAudit", Filter by Type: "AWS managed - job function" and select the policy

**Permissions policies (2/1116)**  
Choose one or more policies to attach to your new user.

Filter by Type	
<input type="text" value="Q security"/>	AWS managed - job function
1 match	< 1 > ⏪
<input checked="" type="checkbox"/> Policy name <a href="#">SecurityAudit</a> ▲ Type Attached entities	
<input checked="" type="checkbox"/>  SecurityAudit AWS managed - job function 0	

▶ Set permissions boundary - optional

Cancel Previous Next

**Step 4:** Finish creating the user. Click on the newly created user and create the Access key and Secret Key from the Security Credentials tab to be used in the AccuKnox panel

Permissions | Groups | Tags | **Security credentials** | Access Advisor

**Console sign-in**

Console sign-in link [Copy](#) Enable console access

**Multi-factor authentication (MFA) (0)**  
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Remove Resync Assign MFA device

Device type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			
<a href="#">Assign MFA device</a>			

**Access keys (0)**  
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

Create access key

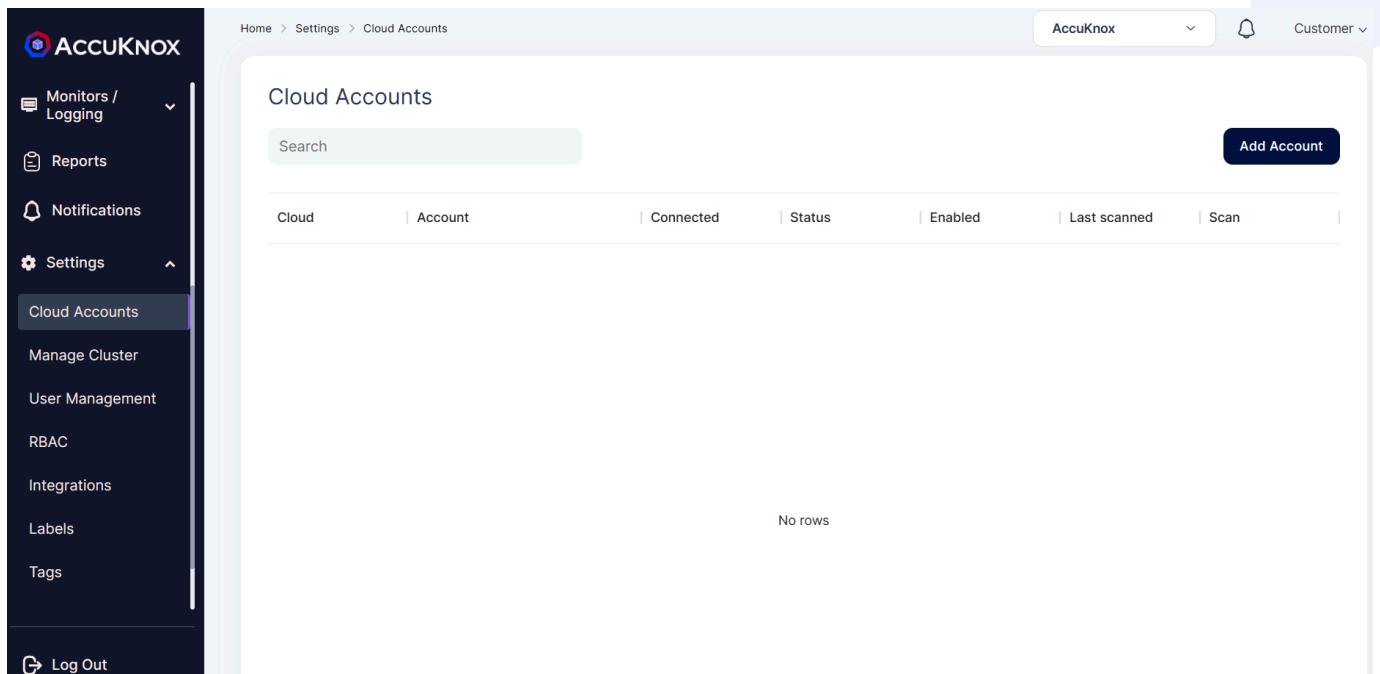
No access keys  
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

# AWS Onboarding

In this example we are onboarding AWS account using the Access Keys method.

**Step 1:** To onboard Cloud Account Navigate to *Settings*→*Cloud Accounts*



The screenshot shows the AccuKnox web interface. The left sidebar has a dark theme with white text and icons. The 'Cloud Accounts' option under the 'Settings' section is highlighted with a purple bar at the top. Other options in the sidebar include 'Monitors / Logging', 'Reports', 'Notifications', 'Manage Cluster', 'User Management', 'RBAC', 'Integrations', 'Labels', and 'Tags'. At the bottom of the sidebar is a 'Log Out' button. The main content area is titled 'Cloud Accounts' and contains a search bar and a large 'Add Account' button. Below the search bar is a table header with columns: Cloud, Account, Connected, Status, Enabled, Last scanned, and Scan. A message 'No rows' is displayed below the table. The top right corner shows the user's name 'AccuKnox', a bell icon, and a 'Customer' dropdown.

**Step 2:** In the Cloud Account Page select *Add Account* option

AccuKnox

Home > Settings > Cloud Accounts

Cloud Accounts

Search

Cloud	Account	Connected	Status	Enabled	Last scanned	Scan
No rows						

Add Account

**Step 3:** Select the AWS option

Cloud Account Details

Label & Tag

Set Up Connectivity

Select your Cloud Account

1 2 3

AWS

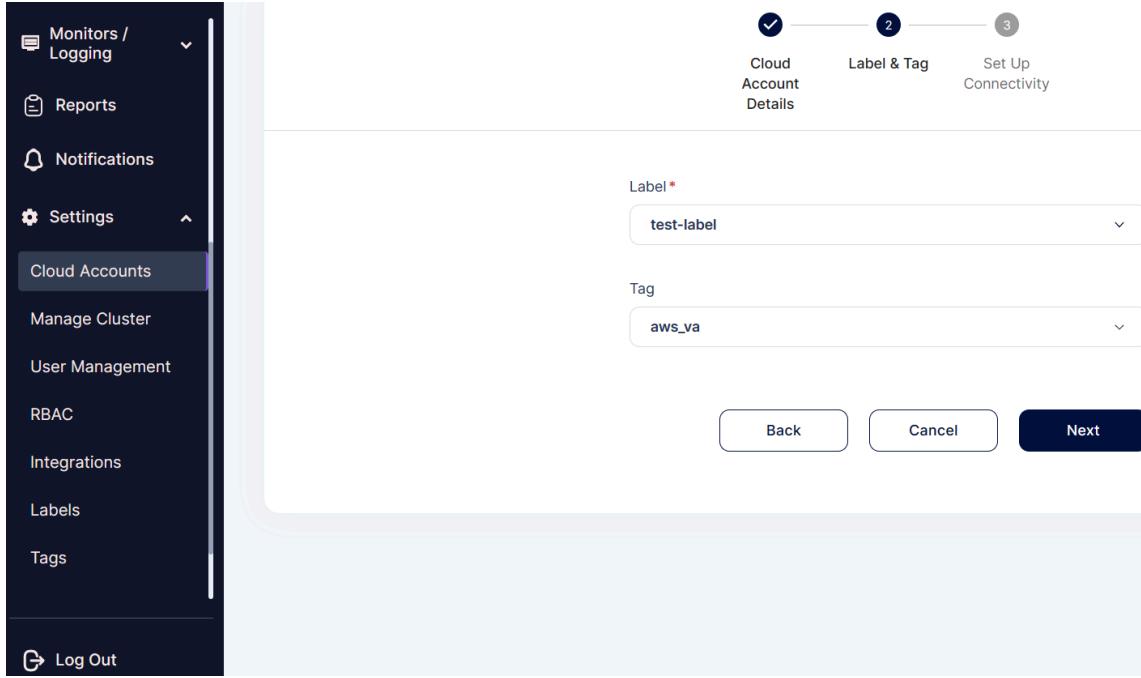
Amazon Web Service (AWS)

GCP

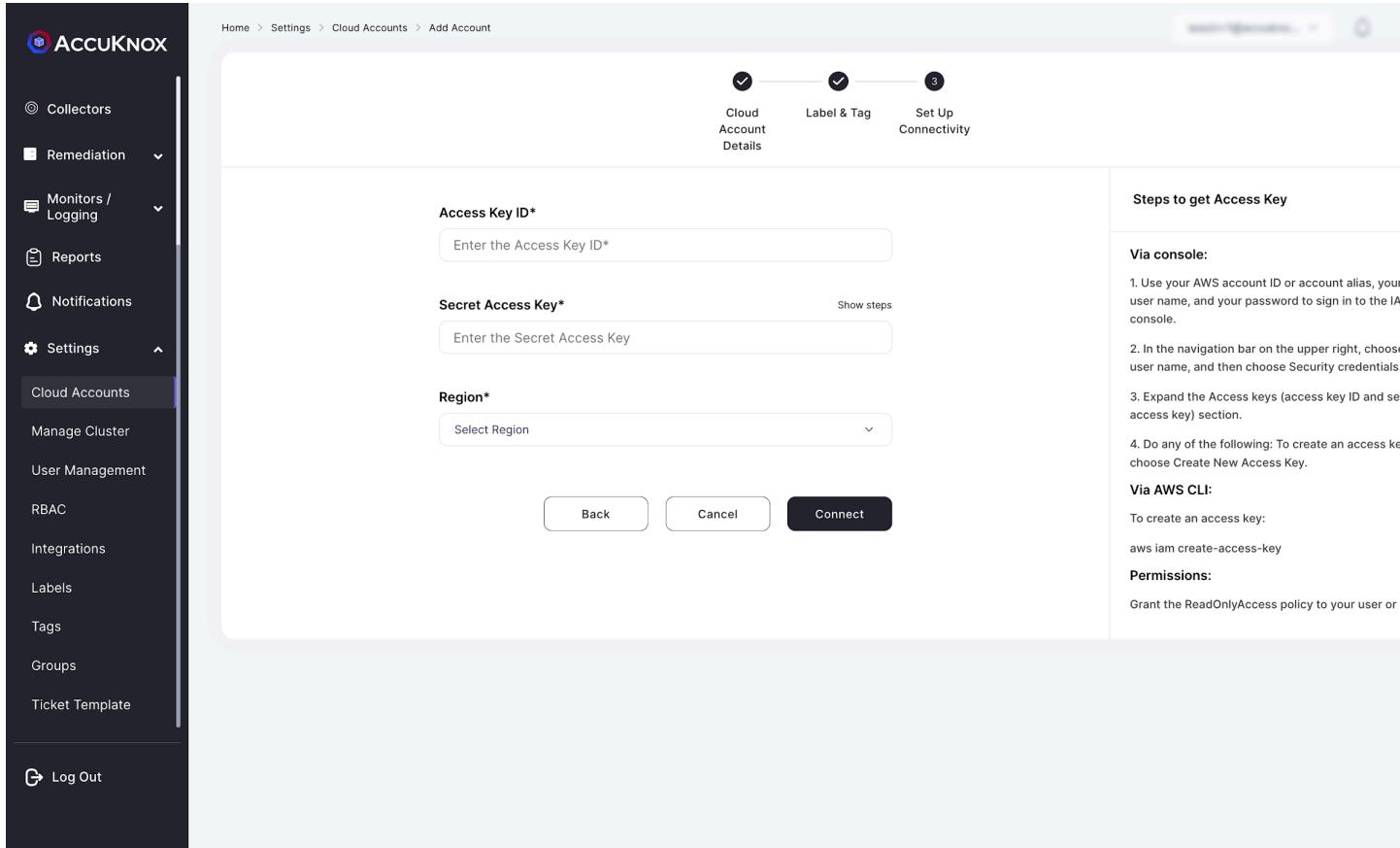
Google Cloud Platform (GCP)

Microsoft Azure

**Step 4:** In the next Screen select the labels and Tags field from the dropdown Menu.



**Step 5:** After giving labels and Tag in the Next Screen Provide the AWS account's Access Key and Secret Access Key ID and Select the Region of the AWS account.



**Step 6:** AWS account is added to the AccuKnox using Access Key Method. We can see the onboarded cloud account by navigating to Settings→cloud Accounts option.

The screenshot shows the AccuKnox web application. On the left, a dark sidebar menu includes options like Monitors / Logging, Reports, Notifications, Settings (with Cloud Accounts selected), Manage Cluster, User Management, RBAC, Integrations, Labels, Tags, and Log Out. The main content area has a header with a search bar and an 'Add Account' button. Below this is a table titled 'Cloud Accounts' with columns for Cloud, Account, Connected, Status, Enabled, Last scanned, and Scan. Two AWS accounts are listed:

Cloud	Account	Connected	Status	Enabled	Last scanned	Scan
aws	aws: 788471067825	2023-02-23	ON	19 days ago	-	<button>Scan</button>
aws	aws: 199488642388	2023-02-28	ON	14 days ago	-	<button>Scan</button>

---

## Onboarding AWS Organization Accounts to AccuKnox

Managing security across multiple AWS accounts is complex. **AWS Organizations** simplifies this by grouping accounts under one structure. **AccuKnox** enhances this by enabling organization-level onboarding—removing the need to add accounts individually. This ensures centralized visibility, consistent policy enforcement, and automatic coverage for new accounts.

This guide explains how to onboard your **AWS Organization root account** to AccuKnox.

## Prerequisites

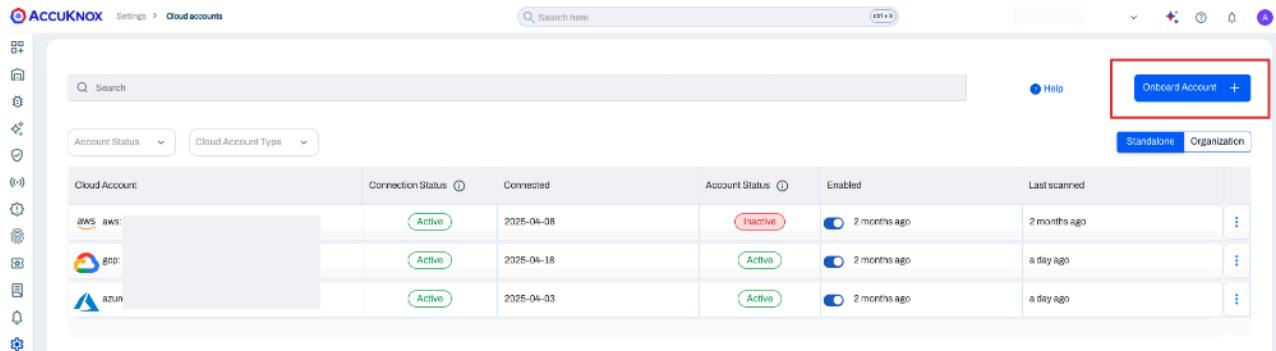
- You must have administrative access to your AWS Management Account and have permissions to deploy CloudFormation Stackset across the Organization.
- You need the AWS Organization ID of your root organization.

# Step-by-Step Onboarding Process

Follow these steps to connect your AWS Organization to AccuKnox:

## 1. Initiate Account Onboarding

In the AccuKnox platform, navigate to **Cloud Security** → **Cloud Accounts** from the left-hand navigation menu. Select the **Organization** button, and then select **Onboard Account**.



The screenshot shows the AccuKnox Cloud Accounts interface. At the top, there's a search bar and a help icon. Below it, there are two dropdown menus: 'Account Status' and 'Cloud Account Type'. A red box highlights the 'Onboard Account' button in the top right corner. In the center, there's a table listing three cloud accounts: AWS, GCP, and AZURE. Each account row includes columns for Cloud Account (with icons), Connection Status (Active), Connected date (e.g., 2025-04-08), Account Status (Inactive for AWS), Enabled status (switched on), and Last scanned date (2 months ago). At the bottom right of the table, there are three vertical dots for more options. To the left of the table, there's a sidebar with various icons.

## 2. Configure Organization Account Type and Labels

Select **Organization Account** as the account type.

Search here ctrl + k

1 Cloud Account Details      2 Label & Tag      3 Set Up Connectivity

Select your Cloud Accounts

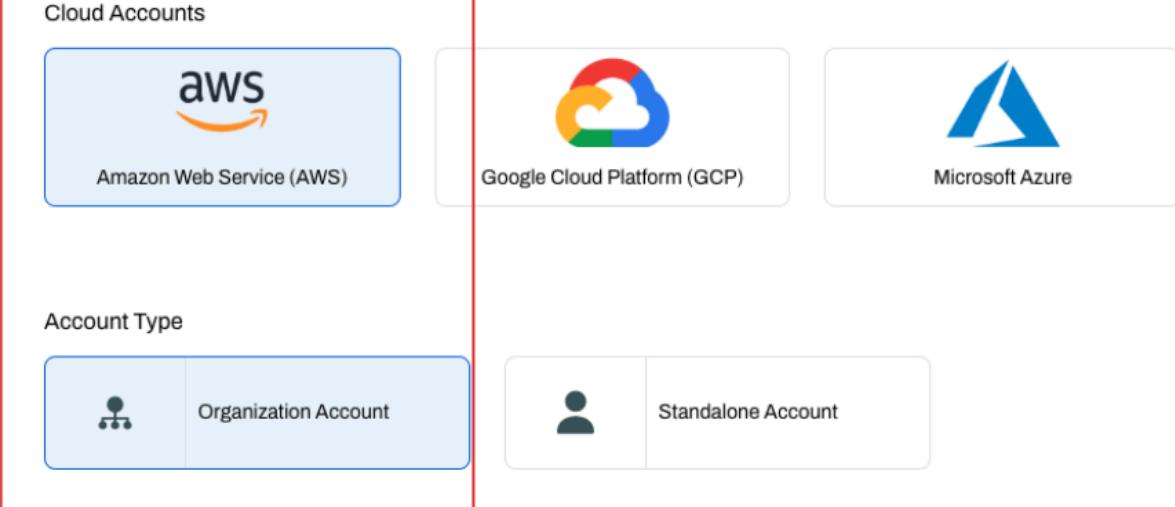
Cloud Accounts

-  Amazon Web Service (AWS)
-  Google Cloud Platform (GCP)
-  Microsoft Azure

Account Type

-  Organization Account
-  Standalone Account

**NEXT**      **BACK**



Next, select existing labels or create new ones to associate with all assets that will be discovered within this AWS Organization.

### 3. Enter AWS Organization Details

- Log in to the **AWS Console** → go to **AWS Organizations**.

- Copy your **Organization ID** (e.g., r-xxxxxxxxxx).

The root org ID has to be copied

**AWS Organizations**

**AWS accounts**

**AWS accounts**

Invitations  
Services  
Policies  
Settings  
Get started

Organization ID  
o-9zkfcs69bt

**AWS accounts**

Add an AWS account

Centralize root access for member accounts

request to create an AWS account has failed in the last 90 days.

Enable in IAM X

**Organization**

Organizational units (OUS) enable you to group several accounts together and administer them as a single unit instead of one at a time.

Search by name, email, account ID or OU ID.

Hierarchy List

Actions ▾

Account created/joined date

Root

crowww

sovrm

aksanable management account Joined 2025/03/11

- You must use the **root organization account**.
- In AccuKnox, paste the ID into the **AWS Organization ID** field.
- Select the AWS regions where your assets are located.

Cloud Account Details

Label & Tag

Set Up Connectivity

3

**Organization ID \***

Automatically connect to new Accounts

**Region \***

US +1

LAUNCH CLOUD FORMATION STACK SET

Create the CloudFormation stack in the management account. On completion, copy the ARN of the SecurityAuditor Role and enter below More help.

**Role ARN in management account \***

e.g. arn:aws:

Back Cancel Connect

## Note

At present, all assets discovered under this organization will inherit these selected labels. Granular labeling for individual assets will be an enhancement in future updates.

## 4. Enable Auto-Connect & Launch StackSet

- Toggle **Automatically connect to new accounts** (optional).
- Click **Launch CloudFormation StackSet** to open the AWS Console.

The screenshot shows the AWS CloudFormation console with the 'Stacks' section selected. On the left, there's a sidebar with various navigation links: CloudFormation (selected), Stacks, StackSets, Exports, Infrastructure Composer, IaC generator, Hooks overview, Hooks, Registry, Public extensions, Activated extensions, Publisher, Spotlight, and Feedback. The main content area is titled 'Quick create stack'. It includes a 'Template' section with a URL: <https://cloud-security-crapp.s3.us-east-2.amazonaws.com/aws-org-cf-dev-e4a0.yaml>. Below it is a 'Stack description' section: 'Deploys AccuknoxOrgSecurityAuditorRole in management account AND all member accounts / selected OUs'. There's a link to 'View template'. The next section is 'Provide a stack name', with a 'Stack name' input field containing 'ak-security-audit'. A note says: 'Stack name can include letters (A-Z and a-z), numbers (0-9) and hyphens.' The final section is 'Parameters', which contains a note: 'Parameters are defined in your template and allow you to input custom values when you create or update a stack.' Under 'AutoDeploy', the value 'true' is selected. The 'OrganizationalUnits' section has a note: 'List of Organizational Unit IDs.' and a dropdown menu. The 'Regions' section has a note: 'Comma-separated list of AWS regions where the StackSet should be deployed.' and a dropdown menu with 'us-east-1' selected.

## 5. Create the Stack in AWS

- Scroll down, check the box: "**I acknowledge that AWS CloudFormation might create IAM resources...**"
- Click **Create stack**.

## 6. Wait for StackSet Deployment

- Wait until the status shows **CREATE\_COMPLETE**.

Timestamp	Logical ID	Status	Detailed status	Status reason
2025-06-05 14:41:19 UTC+0530	StackSet	CREATE\_IN\_PROGRESS	-	Resource creation initiated
2025-06-05 14:41:17 UTC+0530	ManagementAccountRole	CREATE\_IN\_PROGRESS	-	Resource creation initiated
2025-06-05 14:41:16 UTC+0530	ManagementAccountRole	CREATE\_IN\_PROGRESS	-	-
2025-06-05 14:41:16 UTC+0530	StackSet	CREATE\_IN\_PROGRESS	-	-
2025-06-05 14:41:13 UTC+0530	ak-security-audit	CREATE\_IN\_PROGRESS	-	User initiated

 The 'Outputs' tab is also highlighted in red in the interface."/>

## 7. Copy Role ARN

- Go to the **Outputs** tab of the StackSet.
- Copy the value of `RoleArnInManagementAccount`.

The screenshot shows the AWS CloudFormation console with the 'Outputs' tab selected. A red box highlights the 'Outputs' tab in the top navigation bar. Below it, a table lists one output: 'ManagementAccountRoleArn' with the value 'arn:aws:iam::noxOrgSecuri'. An annotation with an arrow points from the text 'Copy this value once it is shown in the output tab' to the 'Value' column of the table.

CloudFormation > Stacks > ak-security-audit

Stacks (1)

ak-security-audit

Outputs (1)

Key	Value	Description
ManagementAccountRoleArn	arn:aws:iam::noxOrgSecuri	The ARN of the AccuKnoxOrgSecurityAuditor role in the management account

Copy this value once it is shown in the output tab

## 8. Connect in AccuKnox

- Paste the ARN in the **Role ARN** field.
- Click **Connect**.

Cloud Account Details      Label & Tag      Set Up Connectivity

Organization ID \*

Automatically connect to new Accounts

Region \*

US +1

Paste the copied value here

LAUNCH CLOUD FORMATION STACK SET

i Create the CloudFormation stack in the management account. On completion, copy the ARN of the SecurityAuditor Role → and enter below [More help](#).

Role ARN in management account \*

Back      Cancel      Connect

## 9. Confirm Onboarding

- You'll be redirected to the **Cloud Accounts** page.
- Refresh the page to see your AWS Organization listed.

The screenshot shows the AccuKnox interface for managing cloud accounts. On the left, there's a sidebar with various icons. The main area has a tree view under 'Account / Sub Account Name' showing a hierarchy from 'Root-H...' down to 'aws:1'. To the right of the tree is a table with columns: 'Connected Status', 'Last Scanned', 'Onboarded On', and 'Enabled'. The table lists several accounts, each with a status indicator (e.g., 'Inactive', 'Active'), a last scan time, an onboard date, and an enable status. A red box highlights the tree view.

Account / Sub Account Name	Connected Status	Last Scanned	Onboarded On	Enabled
Root-H...	Inactive	Jun 3, 2025 9:32 am	May 22, 2025 6:06 pm	May 22, 2025
AccuKnox-OU	Active	Jun 5, 2025 2:11 pm	May 22, 2025 6:06 pm	May 22, 2025
Test-OU	Active	Jun 5, 2025 9:54 am	May 22, 2025 6:06 pm	May 22, 2025
Sub Org Test	Active	Jun 5, 2025 2:12 pm	May 22, 2025 6:06 pm	May 22, 2025
test	Active	Jun 5, 2025 9:58 am	May 22, 2025 6:06 pm	May 22, 2025
aws:1	Active			
Root-H...	Active			

# Post-Onboarding

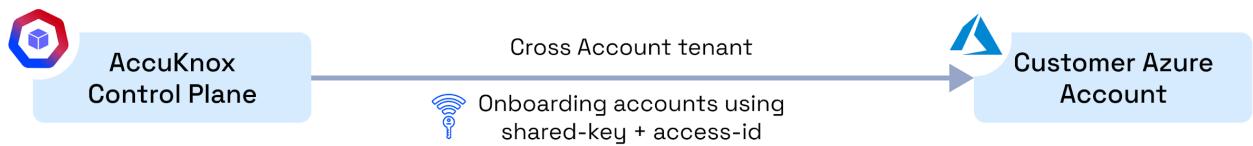
Once your AWS Organization is successfully onboarded:

- **Asset Discovery:** AccuKnox will start an inventory discovery process across all member accounts in the selected regions.
- **Security Scans:** Automated security scans will be scheduled to assess your cloud resources for misconfigurations, vulnerabilities, and compliance violations.
- **Dashboard Population:** Data will begin to populate your AccuKnox dashboards, providing insights into your organization's security posture. This may take some time depending on the size and complexity of your AWS environment.

You have now successfully onboarded your **AWS Organization** to **AccuKnox**, enabling comprehensive, centralized cloud security management.

## CSPM Pre-requisite for Azure

When the AccuKnox control plane is hosted in a cloud environment, scanning is performed using Cloud account Readonly Access permissions.



For Azure Onboarding it is required to register an App and giving Security read access to that App from the Azure portal.

**Step 1:** Go to your Azure Portal and search for *App registrations* and open it

The screenshot shows the Azure portal interface with the search bar at the top containing 'App registrations'. Below the search bar, there are several navigation tabs: 'All' (selected), 'Services (34)', 'Documentation (99+)', 'Resources (0)', 'Resource Groups (0)', 'Marketplace (0)', and 'Azure Active Directory (0)'. Under the 'Services' section, 'App registrations' is highlighted and selected. Other service options listed include 'App proxy', 'Function App', 'Application gateways', and 'Application groups'. In the 'Documentation' section, there are several links related to app registration, such as 'Registering applications', 'Implement app registration - Training', 'Remove limits on creating app registrations - Microsoft Entra', and 'Register your app with the Azure AD v2.0 endpoint - Microsoft Gr...'. At the bottom of the page, there is a link to 'Continue searching in Azure Active Directory' and a 'Give feedback' button.

**Step 2:** Here click on *New registration*

Home >

## App registrations

[New registration](#) [Endpoints](#) [Troubleshooting](#) [Refresh](#) [Download](#) [Preview features](#) | [Got feedback?](#)

**i** Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to support and maintain the existing features and Microsoft Graph. [Learn more](#)

All applications    Owned applications    Deleted applications

Start typing a display name or application (client) ID to filter these results

[Add filters](#)

7 applications found

Display name ↑↓

**Step 3:** Give your application a name, remember this name as it will be used again later, For the rest keep the default settings

Home > App registrations >

## Register an application

### \* Name

The user-facing display name for this application (this can be changed later).

Accuknox-may-2023



### Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

### Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform

e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

[By proceeding, you agree to the Microsoft Platform Policies](#)

[Register](#)

**Step 4:** Now your application is created, save *Application ID* and *Directory ID* as they will be needed to for onboarding on AccuKnox SaaS and then click on 'Add a certificate or secret'

3 ⌂ ...

[Delete](#) [Endpoints](#) [Preview features](#)

#### Essentials

Display name : Accuknox-may-2023

Client credentials : [Add a certificate or secret](#)

Application (client) ID :

Redirect URIs : [Add a Redirect URI](#)

Object ID :

Application ID URI : [Add an Application ID URI](#)

Directory (tenant) ID :

Managed application in I... : [Accuknox-may-2023](#)

Supported account types : [My organization only](#)

Info Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

Info Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

[Get Started](#) [Documentation](#)

**Step 5:** Click on new client secret and enter the name and expiration date to get *secret id* and *secret value*, save this secret value as this will also be needed for onboarding.

The screenshot shows the Azure portal interface for managing application registrations. The left sidebar shows navigation options like Overview, Quickstart, Integration assistant, Manage, Branding & properties, Authentication, Certificates & secrets (which is selected), Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, and Manifest. The main content area is titled 'Accuknox-may-2023 | Certificates & secrets'. It displays a message about credentials and a note about application registration certificates. Under the 'Client secrets' tab, there is one entry: 'may-2023' with a description 'Copied' and an expiration date '5/8/2025'.

**Step 6:** Next, go to *API permissions* tab and click on 'Add permission'

The screenshot shows the Azure portal interface for managing application registrations. The left sidebar shows navigation options like Overview, Quickstart, Integration assistant, Manage, Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions (which is selected), Expose an API, App roles, Owners, Roles and administrators, and Manifest. The main content area is titled 'Permission-screen | API permissions'. It displays a message about admin consent requirements. Under the 'Configured permissions' section, it shows a table with one row for 'User.Read' under 'Microsoft Graph (1)'. The table columns are API / Permissions name, Type, Description, Admin consent req..., and Status.

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	...

**Step 7:** On the screen that appears, click on 'Microsoft Graph'

Home > App registrations > Permission-screen

**Permission-screen | API permissions**

Search | Refresh | Got feedback?

Overview Quickstart Integration assistant

Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners Roles and administrators

The "Admin consent required" column shows the default value for all organizations where this app will be used. [Learn more](#)

**Configured permissions**

Applications are authorized to call APIs when they are granted permission to all the permissions the application needs. [Learn more about permissions](#)

+ Add a permission ✓ Grant admin consent for Default Directory

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and read

To view and manage consented permissions for individual apps, as well as consent for your organization, go to the [App registrations](#) page.

**Request API permissions**

Select an API

Microsoft APIs APIs my organization uses My APIs

Commonly used Microsoft APIs

**Microsoft Graph**  
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Microsoft Entra ID, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

**Azure DevOps**  
Integrate with Azure DevOps and Azure DevOps server

**Azure Service Management**  
Programmatic access to much of the functionality available through the Azure portal

**Azure Storage**  
Secure, massively scalable object and data lake storage for unstructured and semi-structured data

**Dynamics CRM**  
Access the capabilities of CRM business

**Intune**  
Programmatic access to Intune data

**Office 365 Management APIs**  
Retrieve information about user, admin,

**Step 8:** Next, select Application Permissions and then search for Directory.Read.All and click on Add permissions

## Request API permissions

All APIs Microsoft Graph https://graph.microsoft.com/ Docs

What type of permissions does your application require?

Delegated permissions Your application needs to access the API as the signed-in user.

Application permissions Your application runs as a background service or daemon without a signed-in user.

Select permissions

expand all

Directory.Read.All

Permission	Admin consent required
Directory (1)	
Directory.Read.All ⓘ Read directory data	Yes

Add permissions Discard

**Step 9:** Select 'Grant Admin Consent' for Default Directory and click on 'Yes'

**Grant admin consent confirmation.**

Do you want to grant consent for the requested permissions for all accounts in Default Directory? This will update any existing admin consent records this application already has to match what is listed below.

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				
Directory.Read.All	Application	Read directory data	Yes	⚠️ Not granted for Default ...
User.Read	Delegated	Sign in and read user profile	No	

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

**Step 10:** Now we need to give Security read permissions to this registered Application , to do that go to subscriptions

subscriptions

All Services (8) Marketplace (5) Documentation (99+) Resources (0) Resource Groups (0)

Azure Active Directory (0)

Services

- Subscriptions
- Billing subscriptions
- Event Grid Subscriptions
- Quotas
- Event Grid
- Management groups
- Service Bus
- Resource groups

Marketplace

- SharpCloud Subscriptions
- HARP Connect
- Medialine Managed Service in Subscriptions
- Barracuda WAF Add On Subscriptions
- UIB UnificationEngine® WhatsApp Business Platform Subscript...

Documentation

See all

**Step 11:** First save the subscription ID and click on the subscription name , here it is “Microsoft Azure Sponsorship”

The screenshot shows the Microsoft Azure Subscriptions page. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+ /)". Below the search bar, the title "Subscriptions" is displayed, along with a "Default Directory" link and three small icons. A navigation bar below the title includes links for "Add", "Manage Policies", "View Requests", and "View eligible subscriptions". There are also filter buttons for "Subscriptions == global filter", "My role == all", "Status == all", and a "Add filter" button. The main table lists one subscription: "Microsoft Azure Sponsorship" with "Subscription ID" and "Owner" roles. A search bar at the bottom left allows searching for any field.

**Step 12:** Navigate to Access control(IAM) and go to Roles , here select Add and Add role assignment

The screenshot shows the Microsoft Azure Access control (IAM) page for the "Microsoft Azure Sponsorship" subscription. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events, Billing, Invoices, Payment methods, Partner information, Settings, Programmatic deployment, and Resource groups. The "Access control (IAM)" link is currently selected. On the right, a modal dialog titled "Add role assignment" is open, with the "Roles" tab selected. It contains buttons for "Add co-administrator" and "Add custom role". A search bar at the bottom of the dialog shows the text "accuknox". The main area displays a table with columns for "Name" and "Description", showing "Showing 0 of 412 roles" and "No results.".

**Step 13:** Search for “Security Reader” Job function Role, select it and press next

## Add role assignment ...

Role • Members • Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Assignment type

Job function roles      Privileged administrator roles

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

P security reader		Type : All	Category : All
Name ↑↓		Description ↑↓	
Security Detonation Chamber Reader		Allowed to query submission info and files from Security Detonation Chamber	
Security Reader		Security Reader Role	

< Previous Page 1 of 1 Next >

**Step 14:** In the member section click on Select members it will open a dropdown menu on the right hand side

## Add role assignment ...

Role    **Members** •    Review + assign

**Selected role**    Security Reader

**Assign access to**     User, group, or service principal  
 Managed identity

**Members**    [+ Select members](#)

Name	Object ID
No members selected	

**Description**    Optional

**Step 15:** Here search for the Application that you registered in the beginning , select the application and click on *review and assign*.

## Select members

X

Select ⓘ

accuknox-may-2023



Accuknox-may-2023

Selected members:

No members selected. Search for and add one or more members you want to assign to the role for this resource.

[Learn more about RBAC](#)

**Step 16:** Similarly, we have to add another role. This time, search for *Log Analytics Reader*. Select it and click *next*

**Microsoft Azure**

Home > Microsoft Azure Sponsorship | Access control (IAM) >

### Add role assignment

**Role** Members **Review + assign**

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Assignment type

**Job function roles** Privileged administrator roles

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

Name	Description	Type	Category
Reader	View all resources, but does not allow you to make any changes.	BuiltInRole	General
App Compliance Automation Administrator	Create, read, download, modify and delete reports objects and related other resource objects.	BuiltInRole	None
Azure Arc Kubernetes Cluster Admin	Lets you manage all resources in the cluster.	BuiltInRole	Management + Govern
Azure Kubernetes Fleet Manager RBAC Cluster Admin	Lets you manage all resources in the fleet manager cluster.	BuiltInRole	None
Azure Kubernetes Service RBAC Admin	Lets you manage all resources under cluster/namespace, except update or delete resource quotas and namespaces.	BuiltInRole	Containers
Azure Kubernetes Service RBAC Cluster Admin	Lets you manage all resources in the cluster.	BuiltInRole	Containers
Graph Owner	Create and manage all aspects of the Enterprise Graph - Ontology, Schema mapping, Conflation and Conversational AI and Ingestions	BuiltInRole	None
Log Analytics Contributor	Log Analytics Contributor can read all monitoring data and edit monitoring settings. Editing monitoring settings includes adding the VM extension to ...	BuiltInRole	Analytics
<b>Log Analytics Reader</b>	Log Analytics Reader can view and search all monitoring data as well as and view monitoring settings, including viewing the configuration of Azure dia...	BuiltInRole	Analytics
Logic App Contributor	Lets you manage logic app, but not access to them.	BuiltInRole	Integration
Logic App Operator	Lets you read, enable and disable logic app.	BuiltInRole	Integration
Logic Apps Standard Contributor (Preview)	You can manage all aspects of a Standard logic app and workflows. You can't change access or ownership.	BuiltInRole	None
Logic Apps Standard Developer (Preview)	You can create and edit workflows, connections, and settings for a Standard logic app. You can't make changes outside the workflow scope.	BuiltInRole	None

**Step 17:** Now, click on *Select members*, select the application that was created similar to the previous role. Finally, click on *Review and Assign*.

**Microsoft Azure**

Home > Subscriptions > Microsoft Azure Sponsorship | Access control (IAM) >

### Add role assignment

**Role** **Members** **Review + assign**

**Selected role** Log Analytics Reader

**Assign access to**  User, group, or service principal  Managed identity

**Members** [+ Select members](#)

Name	Object ID	Type
No members selected		

**Description**

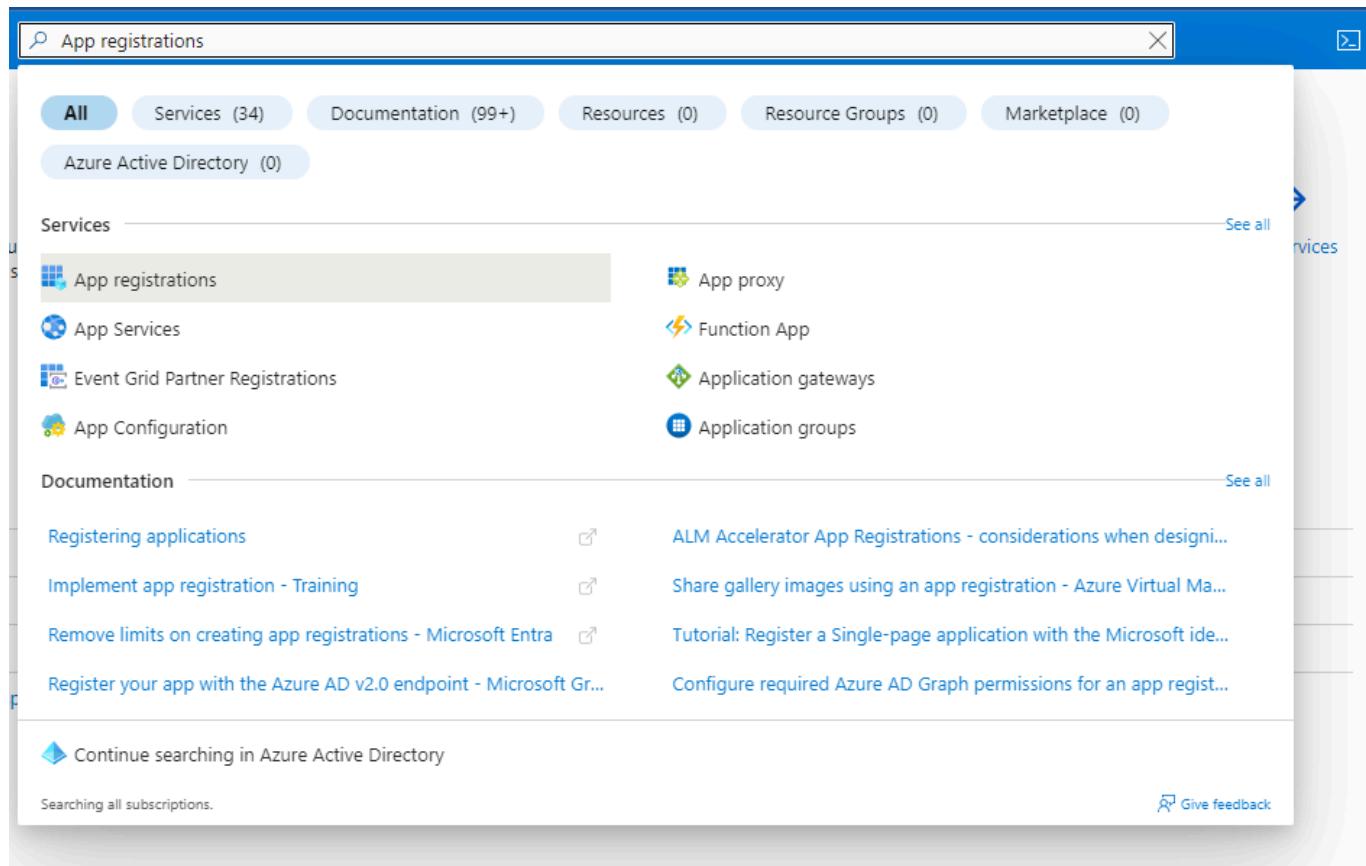
# Azure Account onboarding

In this section we can find the steps to onboard an Azure cloud account to the AccuKnox SaaS platform

# Rapid Onboarding (via Azure)

For Azure Onboarding it is required to register an App and giving Security read access to that App from the Azure portal.

**Step 1:** Go to your Azure Portal and search for *App registrations* and open it



The screenshot shows the Azure portal search results for "App registrations". The search bar at the top contains the text "App registrations". Below the search bar, there are several filter buttons: "All" (selected), "Services (34)", "Documentation (99+)", "Resources (0)", "Resource Groups (0)", and "Marketplace (0)". Under the "Azure Active Directory (0)" heading, there is a "See all" link. The main list under "Services" includes "App registrations" (selected), "App proxy", "App Services", "Function App", "Event Grid Partner Registrations", "Application gateways", "App Configuration", and "Application groups". Under the "Documentation" heading, there is a "See all" link. The list includes "Registering applications", "Implement app registration - Training", "Remove limits on creating app registrations - Microsoft Entra", "Register your app with the Azure AD v2.0 endpoint - Microsoft Gr...", "ALM Accelerator App Registrations - considerations when design...", "Share gallery images using an app registration - Azure Virtual Ma...", "Tutorial: Register a Single-page application with the Microsoft ide...", and "Configure required Azure AD Graph permissions for an app regist...". At the bottom of the page, there is a "Continue searching in Azure Active Directory" button and a "Give feedback" link.

**Step 2:** Here click on *New registration*

Home >

## App registrations

[New registration](#) [Endpoints](#) [Troubleshooting](#) [Refresh](#) [Download](#) [Preview features](#) | [Got feedback?](#)

**i** Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to support and maintain the existing features. [Learn more](#)

All applications    Owned applications    Deleted applications

Start typing a display name or application (client) ID to filter these r...

[Add filters](#)

7 applications found

Display name ↑↓

**Step 3:** Give your application a name, remember this name as it will be used again later, For the rest keep the default settings

Home > App registrations >

## Register an application

### \* Name

The user-facing display name for this application (this can be changed later).

Accuknox-may-2023

### Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

### Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Select a platform

e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the Microsoft Platform Policies [↗](#)

[Register](#)

**Step 4:** Now your application is created, save *Application ID* and *Directory ID* as they will be needed to for onboarding on AccuKnox SaaS and then click on 'Add a certificate or secret'

3 ⌂ ...

[Delete](#) [Endpoints](#) [Preview features](#)

### ^ Essentials

Display name : [Accuknox-may-2023](#)

Client credentials : [Add a certificate or secret](#)

Application (client) ID :

Redirect URIs : [Add a Redirect URI](#)

Object ID :

Application ID URI : [Add an Application ID URI](#)

Directory (tenant) ID :

Managed application in I... : [Accuknox-may-2023](#)

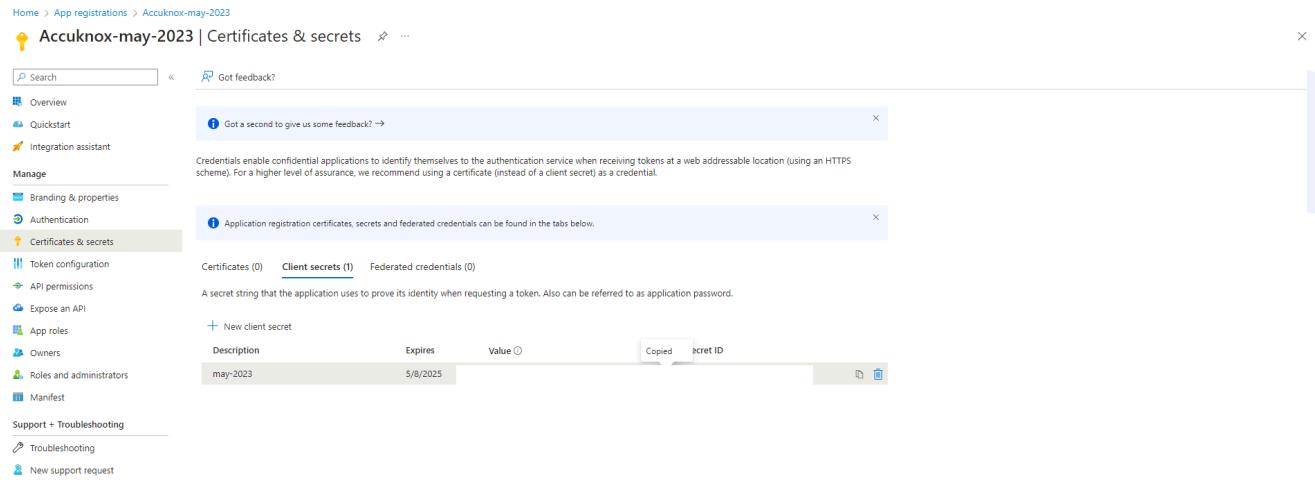
Supported account types : [My organization only](#)

[ⓘ Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations \(Legacy\)? \[Learn more\]\(#\)](#)

[ⓘ Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library \(ADAL\) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will be upgraded to Microsoft Authentication Library \(MSAL\) and Microsoft Graph. \[Learn more\]\(#\)](#)

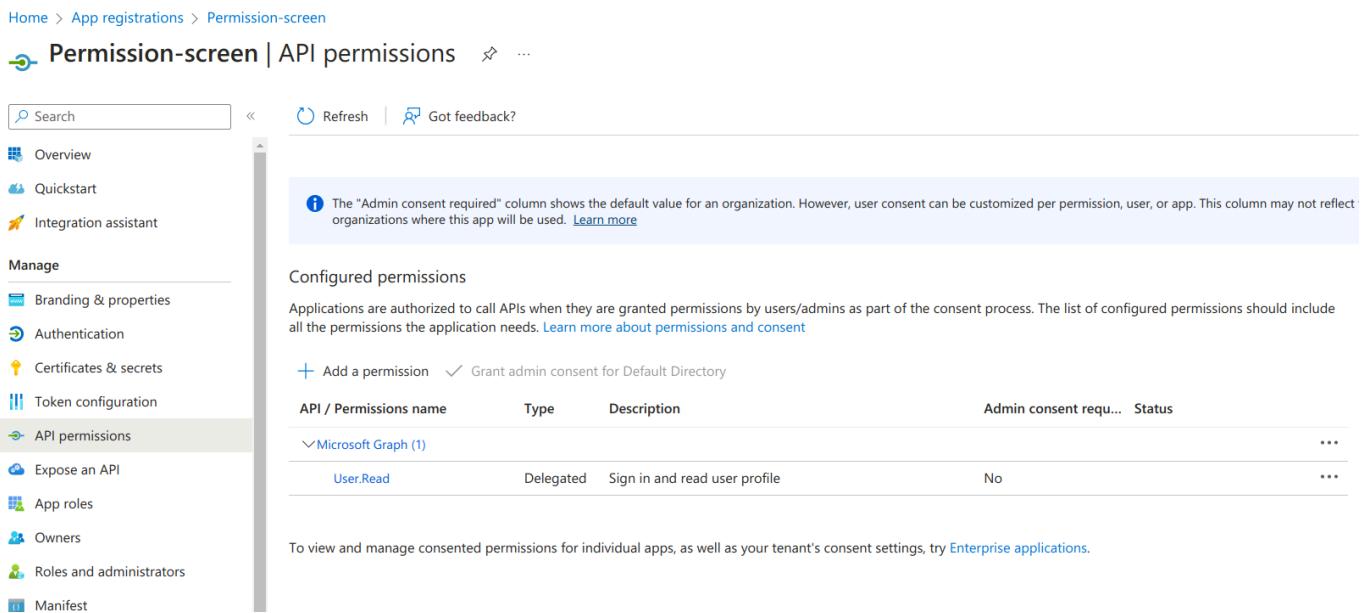
[Get Started](#) [Documentation](#)

**Step 5:** Click on new client secret and enter the name and expiration date to get *secret id* and *secret value*, save this secret value as this will also be needed for onboarding.



The screenshot shows the 'Certificates & secrets' tab selected in the left sidebar. A new client secret named 'may-2023' has been created. The 'Copied' button next to the secret value is highlighted, indicating it can be copied for use.

**Step 6:** Next, go to *API permissions* tab and click on 'Add permission'



The screenshot shows the 'API permissions' tab selected in the left sidebar. A single permission for 'Microsoft Graph (1)' is listed: 'User.Read' (Delegated) with 'Admin consent required' set to 'No'.

API / Permissions name	Type	Description	Admin consent req...	Status
User.Read	Delegated	Sign in and read user profile	No	...

**Step 7:** On the screen that appears, click on 'Microsoft Graph'

Home > App registrations > Permission-screen

## Permission-screen | API permissions

Search | Refresh | Got feedback?

Overview Quickstart Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions**
- Expose an API
- App roles
- Owners
- Roles and administrators

The "Admin consent required" column shows the default value for organizations where this app will be used. [Learn more](#)

**Configured permissions**

Applications are authorized to call APIs when they are granted permission to all the permissions the application needs. [Learn more about permissions](#)

+ Add a permission ✓ Grant admin consent for Default Direct

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and read

To view and manage consented permissions for individual apps, as well as consent for your organization, go to the [Consent screen](#).

## Request API permissions

Select an API

Microsoft APIs APIs my organization uses My APIs

Commonly used Microsoft APIs

**Microsoft Graph**  
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Microsoft Entra ID, Excel, Intune, Outlook/Exchange, OneDrive, SharePoint, Planner, and more through a single endpoint.

**Azure DevOps**  
Integrate with Azure DevOps and Azure DevOps server

**Azure Service Management**  
Programmatic access to much of the functionality available through the Azure portal

**Azure Storage**  
Secure, massively scalable object and data lake storage for unstructured and semi-structured data

**Dynamics CRM**  
Access the capabilities of CRM business

**Intune**  
Programmatic access to Intune data

**Office 365 Management APIs**  
Retrieve information about user, admin,

**Step 8:** Next, select Application Permissions and then search for Directory.Read.All and click on Add permissions

## Request API permissions

X

All APIs



Microsoft Graph

<https://graph.microsoft.com/> Docs

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

expand all

Directory.Read.All

X

Permission

Admin consent required

Directory (1)



Directory.Read.All ⓘ

Read directory data

Yes

Add permissions

Discard

**Step 9:** Select 'Grant Admin Consent' for Default Directory and click on 'Yes'

**Grant admin consent confirmation.**

You want to grant consent for the requested permissions for all accounts in Default Directory? This will update any existing admin consent records this application already has to match what is listed below.

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				
Directory.Read.All	Application	Read directory data	Yes	⚠️ Not granted for Default ...
User.Read	Delegated	Sign in and read user profile	No	

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

**Step 10:** Now we need to give Security read permissions to this registered Application , to do that go to subscriptions

subscriptions

All Services (8) Marketplace (5) Documentation (99+) Resources (0) Resource Groups (0)

Azure Active Directory (0)

Services

- Subscriptions
- Billing subscriptions
- Event Grid Subscriptions
- Quotas
- Event Grid
- Management groups
- Service Bus
- Resource groups

Marketplace

- SharpCloud Subscriptions
- HARP Connect
- Medaline Managed Service in Subscriptions
- Barracuda WAF Add On Subscriptions
- UIB UnificationEngine® WhatsApp Business Platform Subscript...

Documentation

See all

**Step 11:** First save the subscription ID and click on the subscription name , here it is "Microsoft Azure Sponsorship"

The screenshot shows the Microsoft Azure Subscriptions page. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+ /)". Below the search bar, the title "Subscriptions" is displayed, along with a "Default Directory" link and three small icons. A horizontal menu bar follows, featuring "Add", "Manage Policies", "View Requests", and "View eligible subscriptions". Below this is a search bar with the placeholder "Search for any field...". To its right are several filter buttons: "Subscriptions == global filter", "My role == all", "Status == all", and "Add filter". The main table has columns for "Subscription name ↑↓", "Subscription ID ↑↓", and "My role ↑↓". One row in the table is highlighted, showing "Microsoft Azure Sponsorship" in the first column and "Owner" in the third column.

**Step 12:** Navigate to Access control(IAM) and go to Roles , here select Add and Add role assignment

The screenshot shows the Microsoft Azure Access control (IAM) page for the "Microsoft Azure Sponsorship" subscription. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events, Billing, Invoices, Payment methods, Partner information, Settings, Programmatic deployment, and Resource groups. The main content area is titled "Microsoft Azure Sponsorship | Access control (IAM)". It features a search bar and buttons for "Add", "Download role assignments", "Edit columns", "Refresh", "Remove", and "Feedback". A modal window titled "Add role assignment" is open, showing tabs for "Assignments", "Roles" (which is selected), "Deny assignments", and "Classic administrators". It includes a search bar for "accuknox" and filters for "Type : All" and "Category : All". Below the modal, a message states: "Permissions. You can use the built-in roles or you can create your own custom roles. Learn more". The main table displays 0 of 412 roles, with columns for "Name ↑↓" and "Description ↑↓". A message at the bottom says "No results."

**Step 13:** Search for “Security Reader” Job function Role, select it and press next

## Add role assignment ...

[Role](#) • [Members](#) • [Review + assign](#)

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#) 

Assignment type

[Job function roles](#) [Privileged administrator roles](#)

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

P security reader		X	Type : All	Category : All
Name ↑↓	Description ↑↓			
Security Detonation Chamber Reader	Allowed to query submission info and files from Security Detonation Chamber			
Security Reader	Security Reader Role			

< Previous Page 1 of 1 Next >

**Step 14:** In the member section click on Select members it will open a dropdown menu on the right hand side

## Add role assignment ...

Role    **Members** •    Review + assign

**Selected role**    Security Reader

**Assign access to**     User, group, or service principal  
 Managed identity

**Members**    [+ Select members](#)

Name	Object ID
No members selected	

**Description**    Optional

**Step 15:** Here search for the Application that you registered in the beginning , select the application and click on *review and assign*.

## Select members

X

Select ⓘ

accuknox-may-2023



Accuknox-may-2023

Selected members:

No members selected. Search for and add one or more members you want to assign to the role for this resource.

[Learn more about RBAC](#)

**Step 16:** Similarly, we have to add another role. This time, search for *Log Analytics Reader*. Select it and click *next*

Microsoft Azure

Home > Microsoft Azure Sponsorship | Access control (IAM) >

### Add role assignment

**Role** Members **Review + assign**

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Assignment type

**Job function roles** Privileged administrator roles

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓
Reader	View all resources, but does not allow you to make any changes.	BuiltInRole	General
App Compliance Automation Administrator	Create, read, download, modify and delete reports objects and related other resource objects.	BuiltInRole	None
Azure Arc Kubernetes Cluster Admin	Lets you manage all resources in the cluster.	BuiltInRole	Management + Govern
Azure Kubernetes Fleet Manager RBAC Cluster Admin	Lets you manage all resources in the fleet manager cluster.	BuiltInRole	None
Azure Kubernetes Service RBAC Admin	Lets you manage all resources under cluster/namespace, except update or delete resource quotas and namespaces.	BuiltInRole	Containers
Azure Kubernetes Service RBAC Cluster Admin	Lets you manage all resources in the cluster.	BuiltInRole	Containers
Graph Owner	Create and manage all aspects of the Enterprise Graph - Ontology, Schema mapping, Conflation and Conversational AI and Ingestions	BuiltInRole	None
Log Analytics Contributor	Log Analytics Contributor can read all monitoring data and edit monitoring settings. Editing monitoring settings includes adding the VM extension to ...	BuiltInRole	Analytics
<b>Log Analytics Reader</b>	Log Analytics Reader can view and search all monitoring data as well as and view monitoring settings, including viewing the configuration of Azure dia...	BuiltInRole	Analytics
Logic App Contributor	Lets you manage logic app, but not access to them.	BuiltInRole	Integration
Logic App Operator	Lets you read, enable and disable logic app.	BuiltInRole	Integration
Logic Apps Standard Contributor (Preview)	You can manage all aspects of a Standard logic app and workflows. You can't change access or ownership.	BuiltInRole	None
Logic Apps Standard Developer (Preview)	You can create and edit workflows, connections, and settings for a Standard logic app. You can't make changes outside the workflow scope.	BuiltInRole	None

**Step 17:** Now, click on *Select members*, select the application that was created similar to the previous role. Finally, click on *Review and Assign*.

Microsoft Azure

Home > Subscriptions > Microsoft Azure Sponsorship | Access control (IAM) >

### Add role assignment

**Role** **Members** **Review + assign**

**Selected role** Log Analytics Reader

**Assign access to**  User, group, or service principal  Managed identity

**Members** [+ Select members](#)

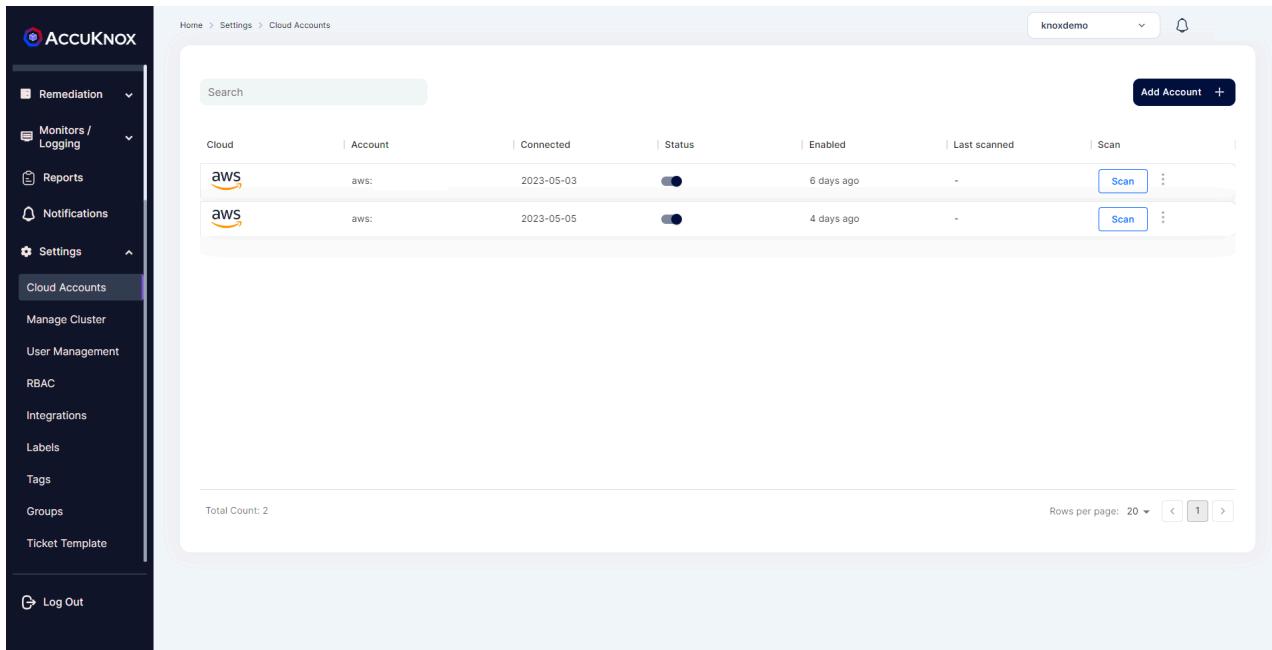
Name	Object ID	Type
No members selected		

**Description**

# From AccuKnox SaaS UI

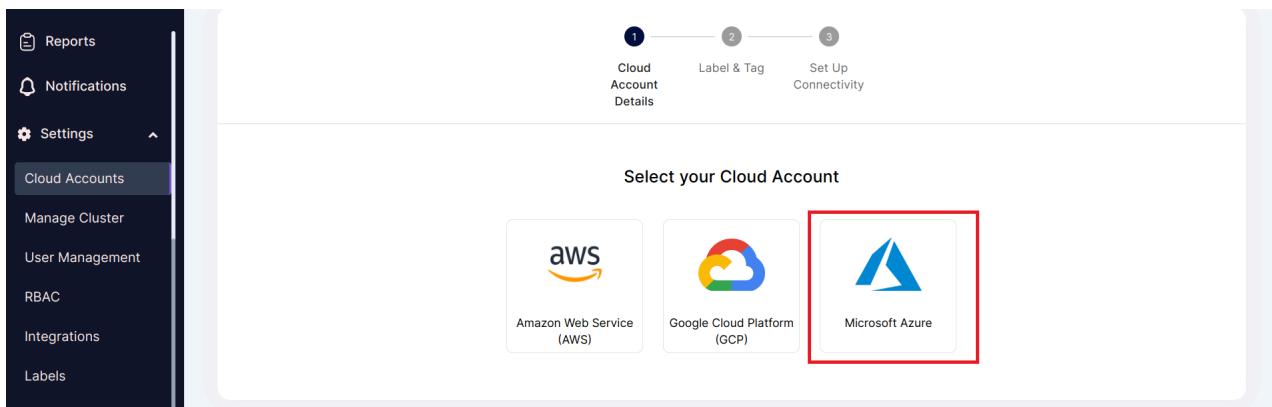
Configuring your Azure cloud account is complete, now we need to onboard the cloud account onto AccuKnox SaaS Platform.

**Step 1:** Go to settings → Cloud Account and click on Add Account



Cloud	Account	Connected	Status	Enabled	Last scanned	Scan	⋮
AWS	aws:	2023-05-03	ON	6 days ago	-	Scan	⋮
AWS	aws:	2023-05-05	ON	4 days ago	-	Scan	⋮

**Step 2:** Select Microsoft Azure as Cloud Account Type

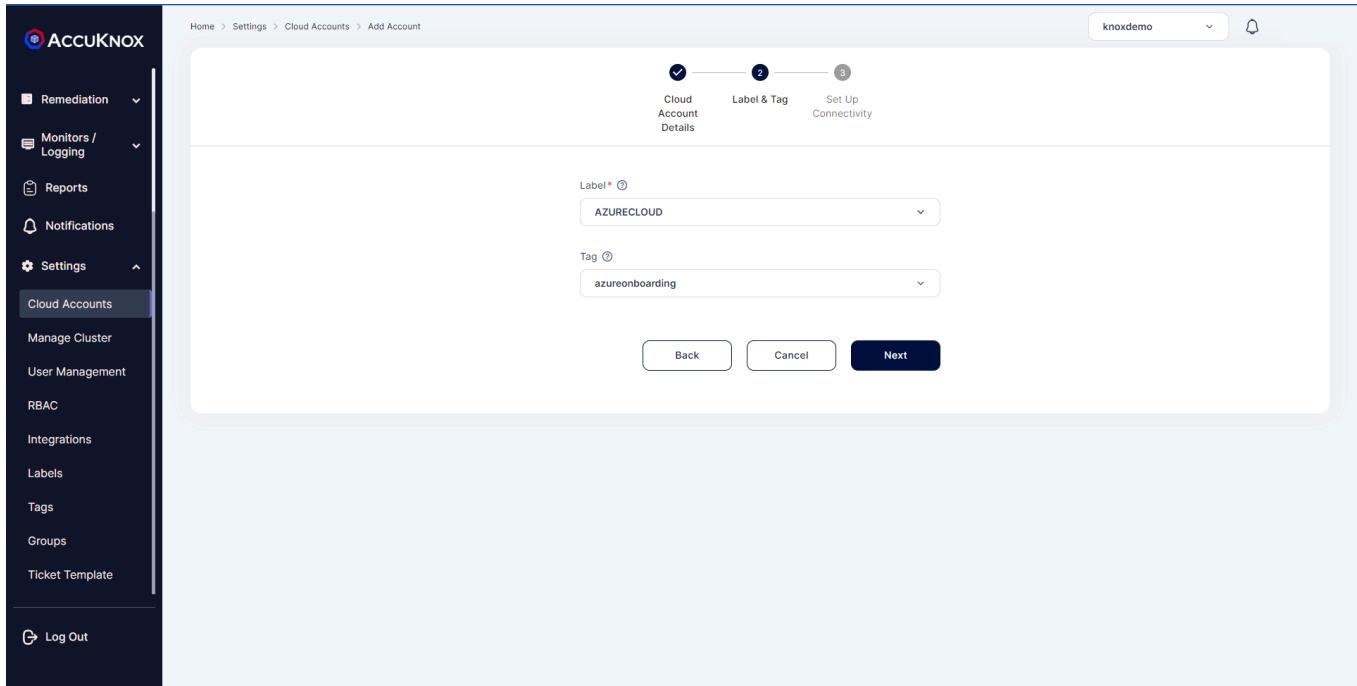


1 Cloud Account Details    2 Label & Tag    3 Set Up Connectivity

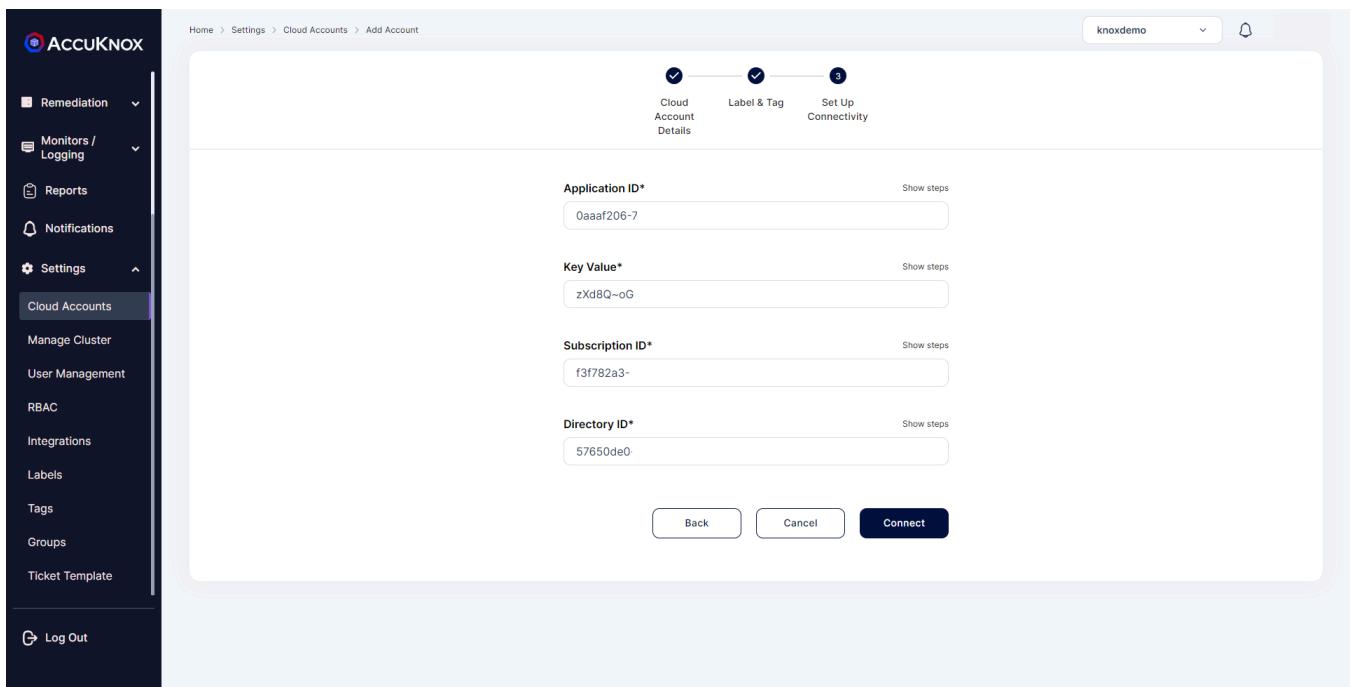
Select your Cloud Account

 Amazon Web Service (AWS)	 Google Cloud Platform (GCP)	 Microsoft Azure
--	---	---

**Step 3:** Select or create label and Tags that will be associated with this Cloud Account



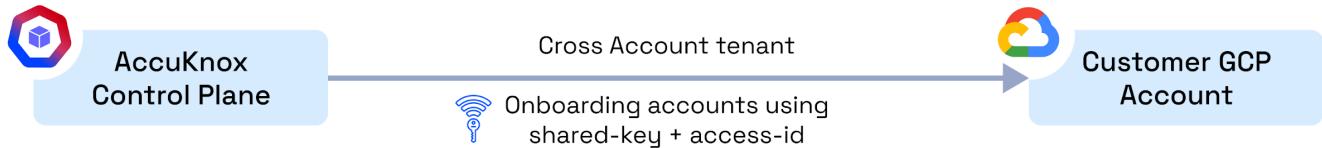
**Step 4:** Enter the details that we saved earlier during the steps for app registration and subscription id from subscriptions in azure portal and click on connect



**Step 5:** After successfully connecting your cloud account will show up in the list

## CSPM Pre-requisite for GCP

When the AccuKnox control plane is hosted in a cloud environment, scanning is performed using Cloud account Readonly Access permissions.



**Note:** Make sure the Below API Library is enabled in your GCP Account for onboarding into AccuKnox SaaS:

1. Compute Engine API
2. Identity and Access Management (IAM) API
3. Cloud Resource Manager API
4. Cloud Functions API
5. KMS API
6. Kubernetes API
7. Cloud SQL Admin API

For GCP there is a requirement for IAM Service Account Access.

**Step 1:** Log into your Google Cloud console and navigate to IAM & Admin choose “Roles” and Click “Create Role”

The screenshot shows the Google Cloud IAM & Admin interface. The left sidebar has a 'Roles' item selected, highlighted with a grey background. The main area displays a list of roles for the 'My First Project' project. The list includes:

Type	Title
<input type="checkbox"/>	<a href="#">Custom AK Role</a>
<input type="checkbox"/>	<a href="#">roles/artifactregistry.createOnPushRepoAdmin</a>
<input type="checkbox"/>	<a href="#">roles/artifactregistry.createOnPushWriter</a>
<input type="checkbox"/>	<a href="#">Access Approval Approver</a>
<input type="checkbox"/>	<a href="#">Access Approval Config Editor</a>
<input type="checkbox"/>	<a href="#">Access Approval Invalidator</a>
<input type="checkbox"/>	<a href="#">Access Approval Viewer</a>
<input type="checkbox"/>	<a href="#">Access Context Manager Admin</a>
<input type="checkbox"/>	<a href="#">Access Context Manager Editor</a>

**Step 2:** Name the “Role” and Click “Add Permission”

← → C 🔒 console.cloud.google.com/iam-admin/roles/create?project=centering-study-396808

Google Cloud My First Project Search (/) for resources, docs, products, and more

IAM & Admin

Workload Identity Federat...

Workforce Identity Federa...

Labels

Tags

Settings

Privacy & Security

Identity-Aware Proxy

**Roles**

Audit Logs

Essential Contacts

Asset Inventory

Create Role

Custom roles let you group permissions and assign them to principals in your project or organization. You can manually select permissions or import permissions from another role. [Learn more](#)

Title \* Custom Role 11 / 100 characters

Description Created on: 2023-09-25 22 / 256 characters

ID \* CustomRole778

Role launch stage Alpha ▾

+ ADD PERMISSIONS

This screenshot shows the 'Create Role' page in the Google Cloud IAM & Admin section. The left sidebar lists various IAM components like Workload Identity Federation, Workforce Identity Federation, Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, Audit Logs, Essential Contacts, and Asset Inventory. The 'Roles' option is selected and highlighted with a blue bar. The main form on the right has a title 'Custom Role', a description 'Created on: 2023-09-25', an ID 'CustomRole778', and a role launch stage set to 'Alpha'. A button at the bottom says '+ ADD PERMISSIONS'.

**Step 3:** Use the Service: storage filter then value as “storage.buckets.getIamPolicy”

## Add permissions

Filter permissions by role

Service : storage X

≡ Filter

storage.buckets.getIamPolicy

X

?

☰



Per Values



storage.buckets.getIamPolicy



storage.buckets.createTagBinding

Supported



storage.buckets.delete

Supported



storage.buckets.deleteTagBinding

Supported



storage.buckets.get

Supported



storage.buckets.getIamPolicy

Supported



storage.buckets.getObjectInsights

Supported



storage.buckets.list

Supported



storage.buckets.listEffectiveTags

Supported



storage.buckets.listTagBindings

Supported

1 – 10 of 28 < >

CANCEL

ADD

**Step 4:** Choose the permission and Click “Add” then Click Create in the same page.

## Add permissions

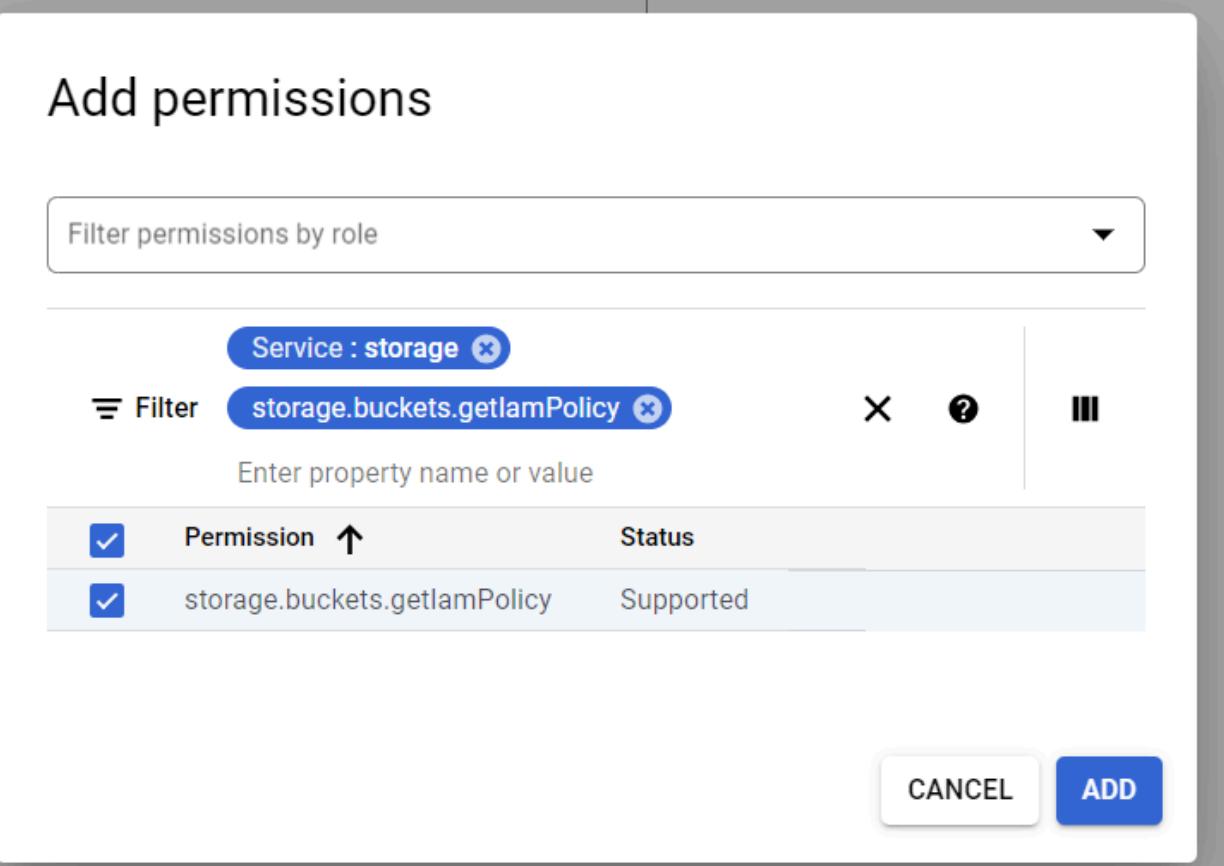
Filter permissions by role

Service : storage storage.buckets.getIamPolicy X ? ☰

Filter Enter property name or value

<input checked="" type="checkbox"/>	Permission ↑	Status
<input checked="" type="checkbox"/>	storage.buckets.getIamPolicy	Supported

CANCEL ADD



**Step 5:** In the Navigation Panel, navigate to IAM Admin > Service Accounts.



Google Cloud



IAM &amp; Admin



IAM



Identity &amp; Organization



Policy Troubleshooter



Policy Analyzer



Organization Policies



Service Accounts



Workload Identity Federat...



Workforce Identity Federa...

**Step 6:** Click on "Create Service Account"

The screenshot shows the Google Cloud IAM & Admin Service Accounts page. On the left sidebar, under the 'Service Accounts' section, 'Service Accounts' is selected. The main area displays a table of service accounts for the project 'My First Project'. The table columns are: Email, Status, Name, Description, and Key ID. The data in the table is as follows:

Email	Status	Name	Description	Key ID
<a href="#">accuknox-onboard@centering-study-396808.iam.gserviceaccount.com</a>	Enabled	accuknox-onboard		
<a href="#">accuknox-read@centering-study-396808.iam.gserviceaccount.com</a>	Enabled	accuknox-read	Readonly	
<a href="#">centering-study-396808@appspot.gserviceaccount.com</a>	Enabled	App Engine default service account		
<a href="#">250501744408-compute@developer.gserviceaccount.com</a>	Enabled	Compute Engine default service		

**Step 7:** Enter any name that you want on Service Account Name.

**Step 8:** Click on Continue.

## 1 Service account details

Service account name

AK-test

Display name for this service account

Service account ID \*

ak-test



Email address: ak-test@centering-study-396808.iam.gserviceaccount.com



Service account description

Describe what this service account will do

**CREATE AND CONTINUE**

**Step 9:** Select the role: Project > Viewer and click Add another Role.

The screenshot shows the Google Cloud IAM & Admin interface for creating a service account. On the left, a sidebar lists various IAM components: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federation, Workforce Identity Federation, Labels, Tags, Settings, Manage Resources, and Release Notes. The main panel is titled "Create service account" and displays two steps: "Service account details" (marked with a checkmark) and "Grant this service account access to project (optional)". A modal window titled "Select a role" is open, showing a list of roles and their descriptions. The "Project" role is selected, and the "Viewer" role is highlighted. A tooltip for "Viewer" indicates it allows "View most Google Cloud resources".

Select a role	IAM condition (optional)
Ops Config Monitoring	Roles
Organization Policy	Browser
Other	Editor
Project	Owner
Proximity Beacon	Viewer
Pub/Sub	
Pub/Sub Lite	

**DONE** **MANAGE ROLES**

**Step 10:** Click “Add Another Role” Choose “Custom” Select the created Custom Role.

[Create service account](#)

Filter Type to filter

Quick access	Roles
Currently used	Custom AK Role
Custom	Custom Role123
Basic	
By product or service	
Access Approval	
Access Context Manager	

**MANAGE ROLES**

+ ADD ANOTHER ROLE

**CONTINUE**

**Step 11:** Click on “Continue“ and ”Done“

## Service account details

### Grant this service account access to project (optional)

Grant this service account access to My First Project so that it has permission to complete specific actions on the resources in your project. [Learn more](#) 

Role	Viewer	IAM condition (optional) 	
------	--------	--	---

View most Google Cloud resources.  
See the list of included permissions.

Role	Custom Role123	IAM condition (optional) 	
------	----------------	--	---

Created on: 2023-09-25

[+ ADD ANOTHER ROLE](#)

[CONTINUE](#)

### Grant users access to this service account (optional)

**Step 12:** Go to the created Service Account, click on that Service Account navigate to the “Keys” section.

**IAM & Admin**

**AK-test**

DETAILS PERMISSIONS KEYS METRICS LOGS

**Keys**

⚠️ Service account keys could pose a security risk if compromised. We recommend you learn about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#). [Learn more about setting organization policies for service accounts](#)

**ADD KEY**

Type	Status	Key	Key creation date	Key expiration date
No rows to display				

**Step 13:** Click the “Add key” button and “Create new key”. Chosen Key type should be JSON format.

**AK-test**

DETAILS PERMISSIONS KEYS METRICS LOGS

**Keys**

⚠️ Service account keys could pose a security risk if compromised. We recommend you learn about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#). [Learn more about setting organization policies for service accounts](#)

**ADD KEY**

Create private key for "AK-test"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

**Key type**

**JSON**  
Recommended

**P12**  
For backward compatibility with code using the P12 format

CANCEL CREATE

**Step 14:** Click the “Create” button it will automatically download the JSON key.

# GCP Account onboarding

Here, we will see the steps to onboard a GCP cloud account to the AccuKnox SaaS platform

**Note:** Make sure the Below API Library is enabled in your GCP Account for onboarding into AccuKnox SaaS:

1. Compute Engine API
2. Identity and Access Management (IAM) API
3. Cloud Resource Manager API
4. Cloud Functions API
5. KMS API
6. Kubernetes API
7. Cloud SQL Admin API

For GCP there is a requirement for IAM Service Account Access.

**Step 1:** Log into your Google Cloud console and navigate to IAM & Admin choose “Roles” and Click “Create Role”

The screenshot shows the Google Cloud IAM & Admin interface. The left sidebar lists various management sections: Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, Roles (which is selected and highlighted in grey), Audit Logs, Essential Contacts, Asset Inventory, Quotas, Groups, and more. The main content area is titled "Roles for 'My First Project' project". It includes a brief description of what a role is and a "more" link. A "Filter" input field is present. Below is a table listing roles, each with a checkbox and a "Type" icon:

	Type	Title
<input type="checkbox"/>		<a href="#">Custom AK Role</a>
<input type="checkbox"/>		<a href="#">roles/artifactregistry.createOnPushRepoAdmin</a>
<input type="checkbox"/>		<a href="#">roles/artifactregistry.createOnPushWriter</a>
<input type="checkbox"/>		<a href="#">Access Approval Approver</a>
<input type="checkbox"/>		<a href="#">Access Approval Config Editor</a>
<input type="checkbox"/>		<a href="#">Access Approval Invalidator</a>
<input type="checkbox"/>		<a href="#">Access Approval Viewer</a>
<input type="checkbox"/>		<a href="#">Access Context Manager Admin</a>
<input type="checkbox"/>		<a href="#">Access Context Manager Editor</a>

**Step 2:** Name the “Role” and Click “Add Permission”

← → C 🔒 console.cloud.google.com/iam-admin/roles/create?project=centering-study-396808

☰ Google Cloud My First Project ▾ Search (/) for resources, docs, products, and more

IAM & Admin

- Workload Identity Federat...
- Workforce Identity Federa...
- Labels
- Tags
- Settings
- Privacy & Security
- Identity-Aware Proxy
- Roles**
- Audit Logs
- Essential Contacts
- Asset Inventory

Create Role

Custom roles let you group permissions and assign them to principals in your project or organization. You can manually select permissions or import permissions from another role. [Learn more](#)

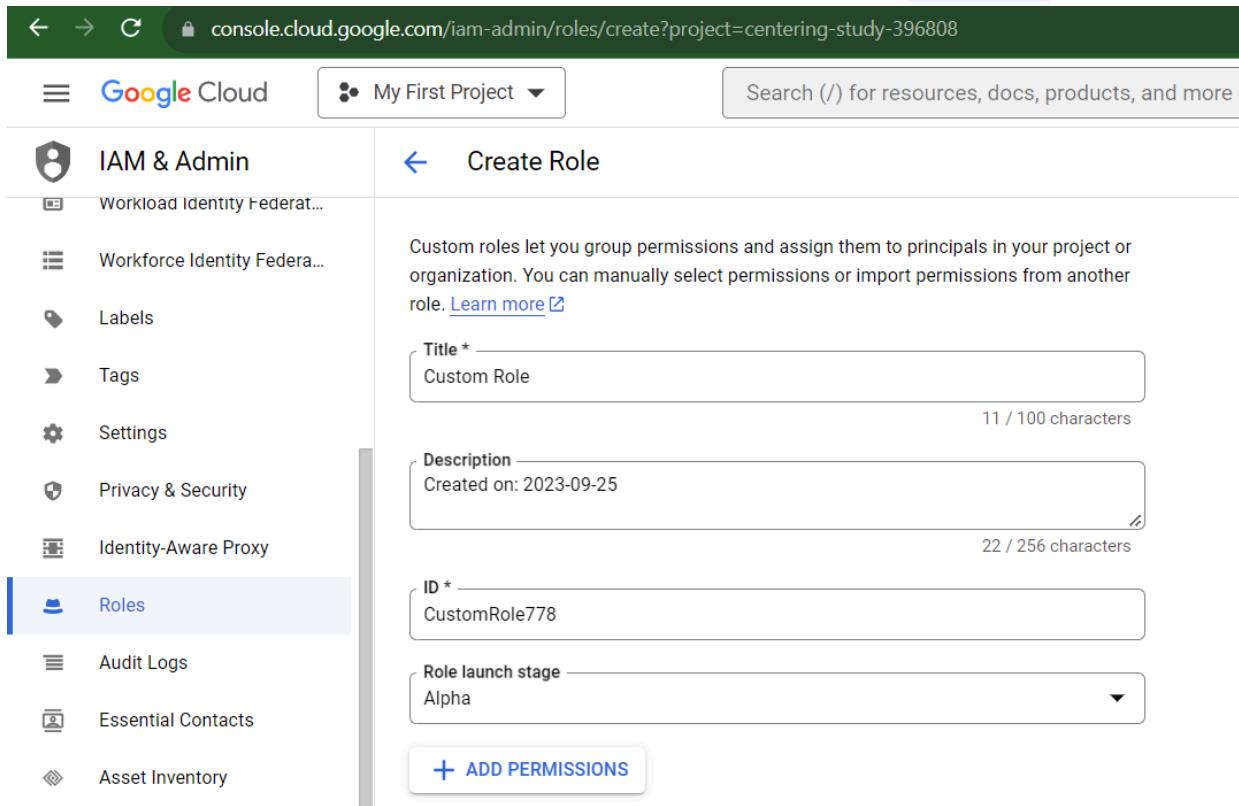
Title \* Custom Role 11 / 100 characters

Description Created on: 2023-09-25 22 / 256 characters

ID \* CustomRole778

Role launch stage Alpha ▾

**+ ADD PERMISSIONS**



**Step 3:** Use the Service: storage filter then value as “storage.buckets.getIamPolicy”

## Add permissions

Filter permissions by role ▾

Service : storage X

Filter

storage.buckets.getIamPolicy

X

?

☰

<input type="checkbox"/>	Per Values	
<input type="checkbox"/>	storage.buckets.getIamPolicy	Supported
<input type="checkbox"/>	storage.buckets.createTagBinding	Supported
<input type="checkbox"/>	storage.buckets.delete	Supported
<input type="checkbox"/>	storage.buckets.deleteTagBinding	Supported
<input type="checkbox"/>	storage.buckets.get	Supported
<input type="checkbox"/>	storage.buckets.getIamPolicy	Supported
<input type="checkbox"/>	storage.buckets.getObjectInsights	Supported
<input type="checkbox"/>	storage.buckets.list	Supported
<input type="checkbox"/>	storage.buckets.listEffectiveTags	Supported
<input type="checkbox"/>	storage.buckets.listTagBindings	Supported

1 – 10 of 28 < >

CANCEL

ADD

**Step 4:** Choose the permission and Click "Add" then Click Create in the same page.

## Add permissions

Filter permissions by role ▾

Service : storage X

Filter

storage.buckets.getIamPolicy X



Enter property name or value

<input checked="" type="checkbox"/> Permission ↑	Status
<input checked="" type="checkbox"/> storage.buckets.getIamPolicy	Supported

CANCEL

ADD

**Step 5:** In the Navigation Panel, navigate to IAM Admin > Service Accounts.



Google Cloud



IAM &amp; Admin



IAM



Identity &amp; Organization



Policy Troubleshooter



Policy Analyzer



Organization Policies



Service Accounts



Workload Identity Federat...



Workforce Identity Federa...

**Step 6:** Click on "Create Service Account"

The screenshot shows the Google Cloud IAM & Admin Service Accounts page. On the left, a sidebar lists various IAM-related options like IAM, Identity & Organization, Policy Troubleshooter, etc., with 'Service Accounts' selected. The main area displays a table of service accounts for the project 'My First Project'. The table columns are Email, Status, Name, Description, and Key ID. The listed accounts are:

Email	Status	Name	Description	Key ID
accuknox-onboard@centering-study-396808.iam.gserviceaccount.com	Enabled	accuknox-onboard		
accuknox-read@centering-study-396808.iam.gserviceaccount.com	Enabled	accuknox-read	Readonly	
centering-study-396808@appspot.gserviceaccount.com	Enabled	App Engine default service account		
250501744408-compute@developer.gserviceaccount.com	Enabled	Compute Engine default service		

**Step 7:** Enter any name that you want on Service Account Name.

**Step 8:** Click on Continue.

## 1 Service account details

Service account name

AK-test

Display name for this service account

Service account ID \*

ak-test

X C

Email address: ak-test@centering-study-396808.iam.gserviceaccount.com



Service account description

Describe what this service account will do

**CREATE AND CONTINUE**

**Step 9:** Select the role: Project > Viewer and click Add another Role.

The screenshot shows the Google Cloud IAM & Admin interface for creating a service account. On the left, a sidebar lists various IAM components: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federation, Workforce Identity Federation, Labels, Tags, Settings, Manage Resources, and Release Notes. The main panel is titled "Create service account" and displays two steps: "Service account details" (marked with a checkmark) and "Grant this service account access to project (optional)". A sub-step 2 is shown under "Grant this service account access to project". It includes a note: "Grant this service account access to My First Project so that it has permission to complete specific actions on the resources in your project." Below this is a "Select a role" dropdown menu with a "Filter" input field. The "Project" role is selected, and the "Viewer" role is highlighted with a tooltip: "Viewer View most Google Cloud res permissions." A "DONE" button is at the bottom left of the dropdown, and a "MANAGE ROLES" link is at the bottom right.

**Step 10:** Click “Add Another Role” Choose “Custom” Select the created Custom Role.

[Create service account](#)

Filter Type to filter

Category	Role
Quick access	Custom AK Role
Currently used	Custom Role123
Custom	
Basic	
By product or service	
Access Approval	
Access Context	
Manager	

[MANAGE ROLES](#)

[+ ADD ANOTHER ROLE](#)

[CONTINUE](#)

**Step 11:** Click on “Continue” and “Done”

## Service account details

### Grant this service account access to project (optional)

Grant this service account access to My First Project so that it has permission to complete specific actions on the resources in your project. [Learn more](#) 

Role	Viewer	IAM condition (optional) 
------	--------	--

[+ ADD IAM CONDITION](#) 

View most Google Cloud resources.  
See the list of included permissions.

Role	Custom Role123	IAM condition (optional) 
------	----------------	--

[+ ADD IAM CONDITION](#) 

Created on: 2023-09-25

[+ ADD ANOTHER ROLE](#)

[CONTINUE](#)

### Grant users access to this service account (optional)

**Step 12:** Go to the created Service Account, click on that Service Account navigate to the "Keys" section.

The screenshot shows the Google Cloud IAM & Admin interface. On the left, there's a sidebar with various options like IAM, Identity & Organization, Policy Troubleshooter, etc. The main area is titled 'AK-test' and has tabs for DETAILS, PERMISSIONS, KEYS (which is selected), METRICS, and LOGS. Under the KEYS tab, it says 'Keys' and includes a warning about service account keys being a security risk. It also says 'Add a new key pair or upload a public key certificate from an existing key pair.' Below that, there's a section for blocking key creation using organization policies. A large 'ADD KEY' button is present, followed by a table header for Type, Status, Key, Key creation date, and Key expiration date. The message 'No rows to display' is shown.

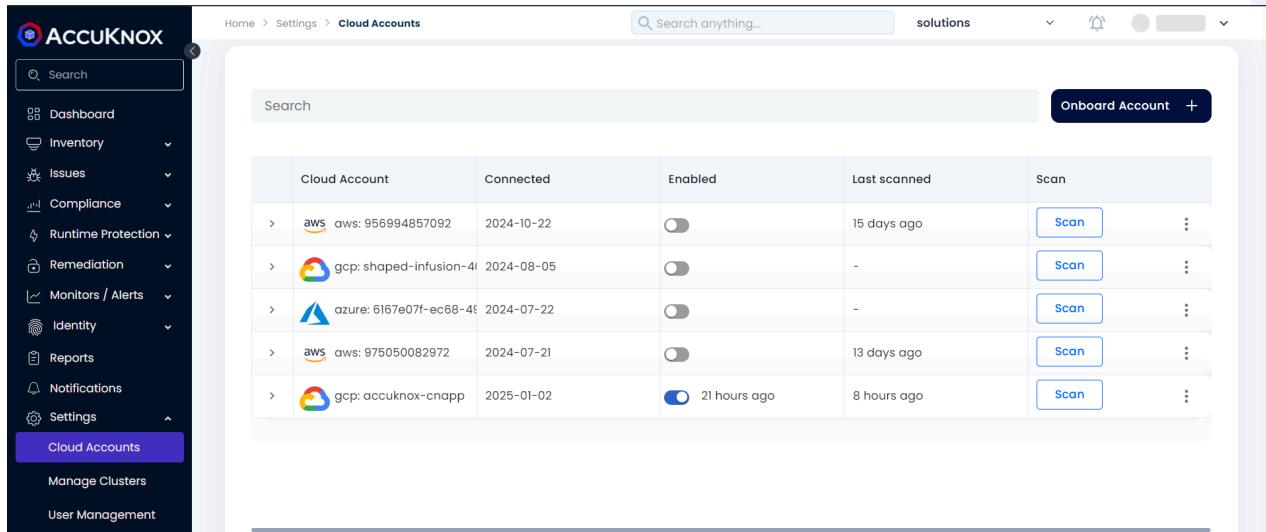
**Step 13:** Click the “Add key“ button and “Create new key “ . Chosen Key type should be JSON format.

This is a modal dialog box titled 'Create private key for "AK-test"'. It contains a warning about the security risk of service account keys. Below the warning, it says 'Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.' It has a 'Key type' section with two options: 'JSON' (selected) and 'P12'. The 'JSON' option is described as 'Recommended'. At the bottom, there are 'CANCEL' and 'CREATE' buttons.

**Step 14:** Click the “Create“ button it will automatically download the JSON key.

# From AccuKnox SaaS UI

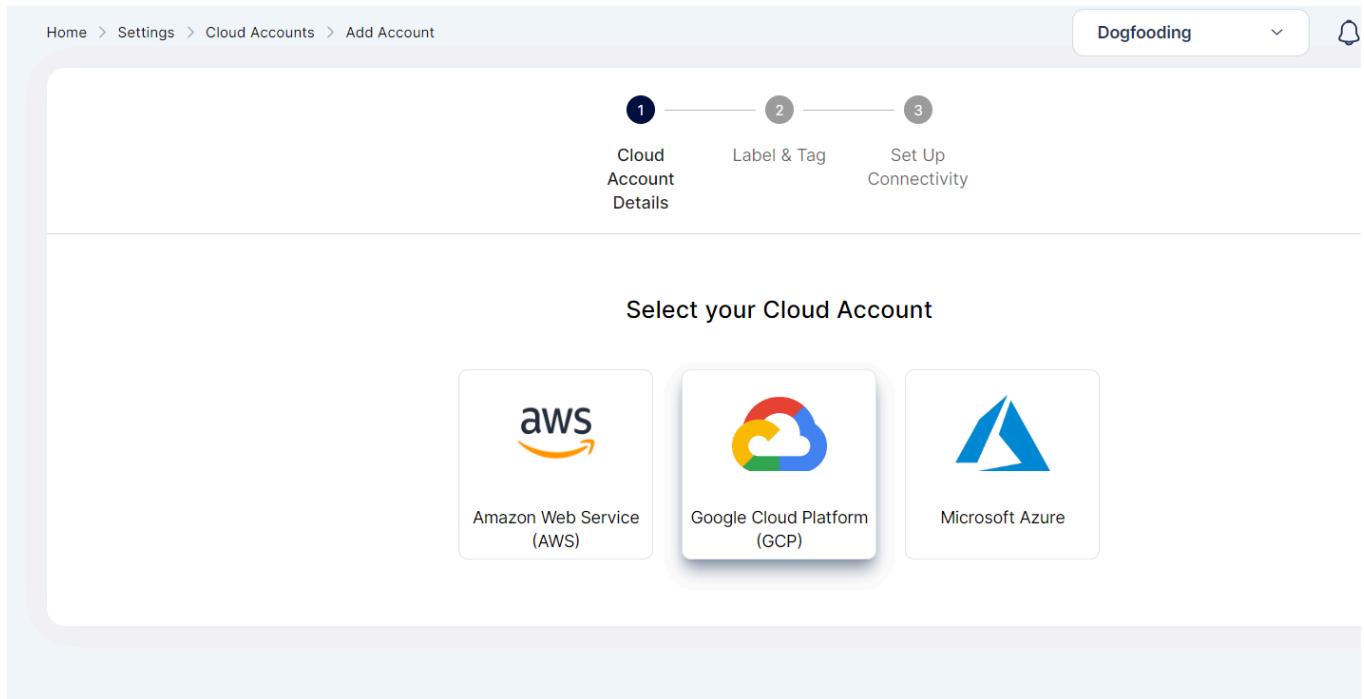
**Step 1:** Go to the AccuKnox SaaS. Navigate to the “Settings” → “Cloud Accounts” then “Add Account”.



The screenshot shows the AccuKnox SaaS interface. On the left is a dark sidebar with various navigation options like Dashboard, Inventory, Issues, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports, Notifications, and Settings. The 'Cloud Accounts' option is highlighted in purple. The main content area has a header 'Home > Settings > Cloud Accounts'. Below the header is a search bar and a button 'Onboard Account +'. A table lists five connected cloud accounts with columns for Cloud Account, Connected, Enabled, Last scanned, and Scan. Each account row includes a 'Scan' button and a more options menu (three dots). The accounts listed are aws, gcp, azure, aws, and gcp.

Cloud Account	Connected	Enabled	Last scanned	Scan
aws aws: 956994857092	2024-10-22	OFF	15 days ago	<button>Scan</button> ...
gcp: shaped-infusion-4f	2024-08-05	OFF	-	<button>Scan</button> ...
azure: 6167e07f-ec68-49	2024-07-22	OFF	-	<button>Scan</button> ...
aws aws: 975050082972	2024-07-21	OFF	13 days ago	<button>Scan</button> ...
gcp: accuknox-cnapp	2025-01-02	ON	21 hours ago	<button>Scan</button> ...

**Step 2:** Click the “GCP Platform”



The screenshot shows the 'Add Account' wizard. The top navigation bar includes 'Home > Settings > Cloud Accounts > Add Account' and a 'Dogfooding' dropdown. Below the navigation is a progress bar with three steps: 1. Cloud Account Details (highlighted in blue), 2. Label & Tag, and 3. Set Up Connectivity. The main content area is titled 'Select your Cloud Account' and features three buttons for AWS, GCP, and Microsoft Azure, each with their respective logos and names.

Cloud Account Details      Label & Tag      Set Up Connectivity

Select your Cloud Account

aws      Google Cloud Platform (GCP)      Microsoft Azure

**Step 3:** Create New Label and Add the Label for identifying the assets inside this account and add a Tag optionally.



2

3

Cloud  
Account  
Details

Label &amp; Tag

Set Up  
Connectivity

Label\* ⓘ

Select the label



Tag ⓘ

Select the tag



Back

Cancel

Next

**Step 4:** Enter the “Project ID”, “Client Email”(The Service Account mail ID) and “Private Key” from the downloaded File. Copy paste the entire downloaded file into the “Private Key” field . Then Click “Connect“

✓
✓
3

<b>Cloud Account Details</b>	<b>Label &amp; Tag</b>	<b>Set Up Connectivity</b>
--------------------------------------	------------------------	--------------------------------

[Show steps](#)

**Project ID**

centering-study-396808

**Client Email**

ak-test@centering-study-396808.iam.gserviceaccount.com

**Private Key**

```
study-396808.iam.gserviceaccount.com",
  "universe_domain": "googleapis.com"
}
```

[Back](#)
[Cancel](#)
Connect

The cloud account has been onboarded successfully

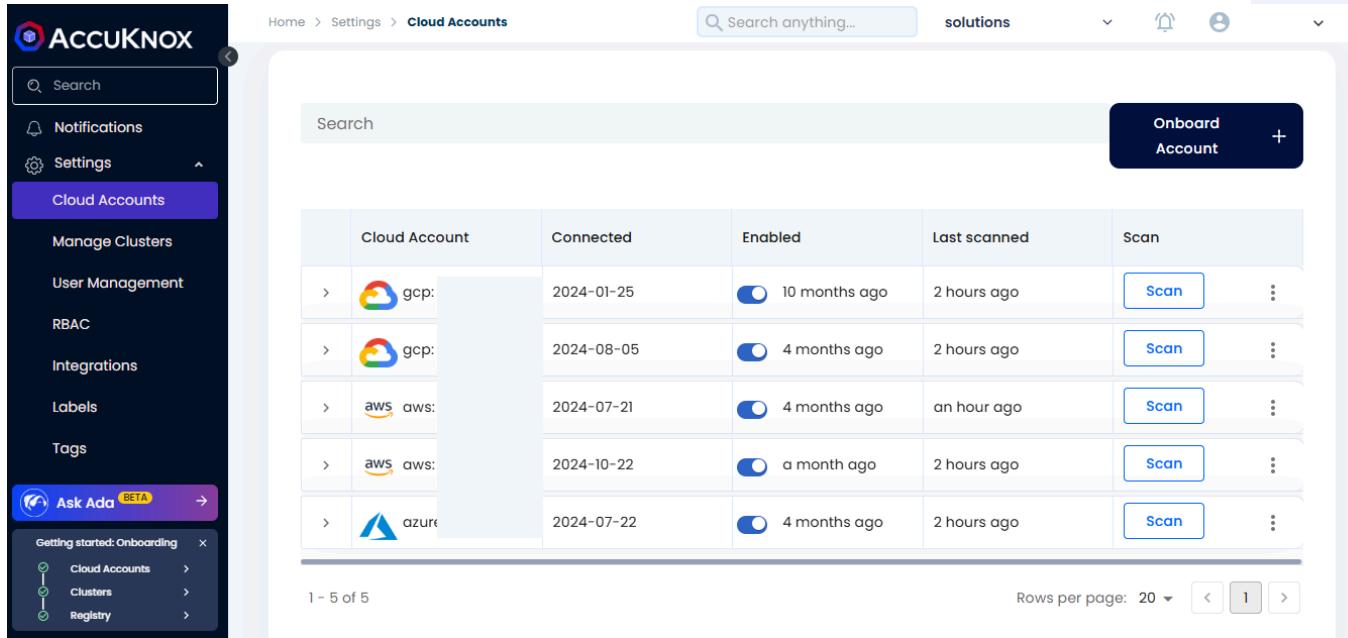
Account Connected Successfully ×

Cloud	Account	Connected	Status	Enabled	Last scanned	Scan
	aws: 956994857092	2023-09-21	<span style="border: 1px solid #002B36; border-radius: 50%; padding: 2px 5px;">●</span>	5 days ago	2023-09-25	<span style="border: 1px solid #002B36; border-radius: 5px; padding: 2px 10px; color: #002B36;">Scan</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;">...</span>
	gcp: centering-study-396808	2023-09-26	<span style="border: 1px solid #002B36; border-radius: 50%; padding: 2px 5px;">●</span>	a few seconds ago	-	<span style="border: 1px solid #002B36; border-radius: 5px; padding: 2px 10px; color: #002B36;">Scan</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;">...</span>

# How to Deboard a Cloud Account

This guide outlines the steps for offboarding a cloud account from AccuKnox SaaS.

**Step 1:** Login to AccuKnox SaaS and Go to Cloud Accounts under Settings.

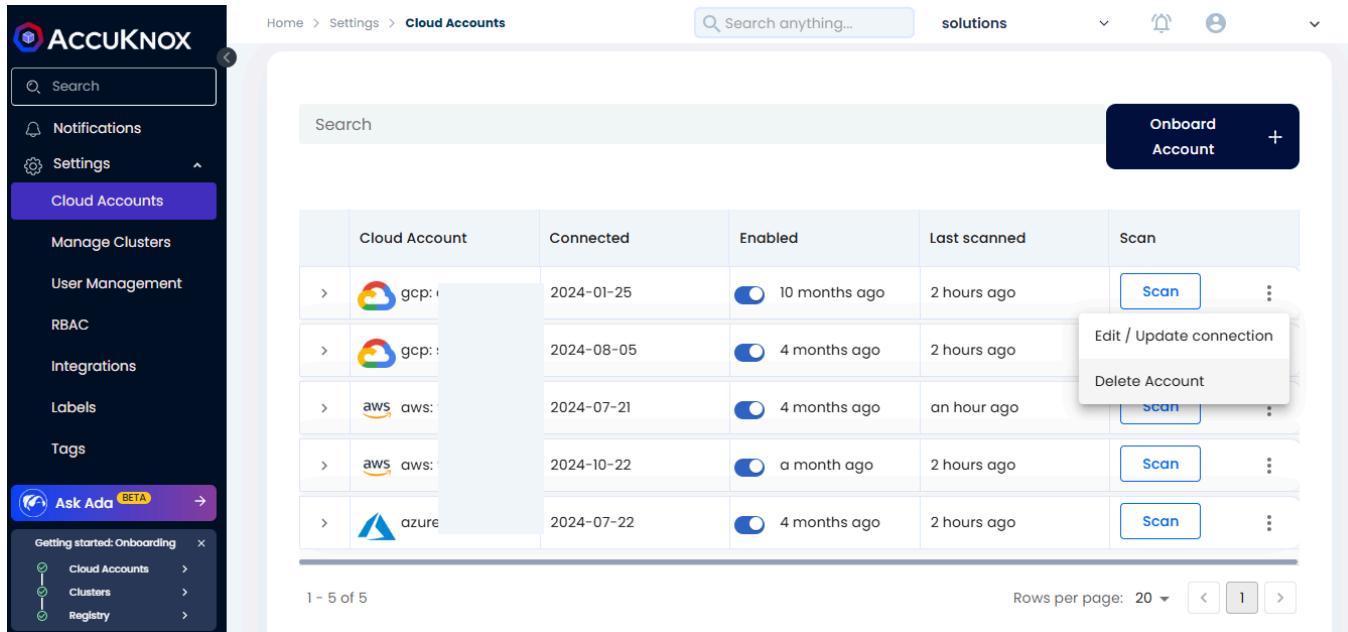


The screenshot shows the AccuKnox SaaS interface. On the left, there is a dark sidebar with the AccuKnox logo, a search bar, and several navigation items: Notifications, Settings (which is currently selected), Cloud Accounts (highlighted in purple), Manage Clusters, User Management, RBAC, Integrations, Labels, Tags, and Ask Ada (BETA). Below these are sections for Getting started: Onboarding, Cloud Accounts, Clusters, and Registry. The main content area is titled "Cloud Accounts" and shows a table with the following data:

Cloud Account	Connected	Enabled	Last scanned	Scan	⋮	
> gcp:	2024-01-25	<input checked="" type="checkbox"/>	10 months ago	2 hours ago	<button>Scan</button>	⋮
> gcp:	2024-08-05	<input checked="" type="checkbox"/>	4 months ago	2 hours ago	<button>Scan</button>	⋮
> aws	aws:	2024-07-21	<input checked="" type="checkbox"/>	4 months ago	<button>Scan</button>	⋮
> aws	aws:	2024-10-22	<input checked="" type="checkbox"/>	a month ago	<button>Scan</button>	⋮
> azure		2024-07-22	<input checked="" type="checkbox"/>	4 months ago	<button>Scan</button>	⋮

At the bottom of the table, it says "1 - 5 of 5". To the right, there are buttons for "Rows per page: 20" and navigation arrows. A "Search" bar is at the top right, and a "solutions" dropdown is also present.

**Step 2:** Select the cloud account and click “Delete” to delete the account from SaaS.



The screenshot shows the same AccuKnox SaaS interface as the previous step. The "Cloud Accounts" section is still selected. In the table, the second row for "gcp:" has been selected. A context menu is open over this row, displaying two options: "Edit / Update connection" and "Delete Account". The rest of the table and interface remain the same as in the first screenshot.

This will delete the cloud account from AccuKnox SaaS.

# Kubernetes Security Onboarding

## Features Supported for Kubernetes

- Supported on managed (EKS, AKS, OCI) and on-prem Kubernetes clusters
- Works on Kubernetes versions >= 1.18
- All features are modular and can be enabled independently
- Available via AccuKnox SaaS and On-Prem Control Plane with identical UX
- Runtime Security requires Linux kernel >= 4.15
- Only egress connectivity from K8s cluster to control plane is required

## K8s Runtime Visibility and Security

**Deployment Mode:** DaemonSet via Operator (default) or Kubernetes manifests

**Helm Command:**

```
helm upgrade --install agents oci://public.ecr.aws/k9v9d5v2/agents-chart \
--version "v0.10.0" \
--set joinToken="[TOKEN]" \
--set spireHost="spire.demo.accuknox.com" \
--set ppsHost="pps.demo.accuknox.com" \
--set knoxGateway="knox-gw.demo.accuknox.com:3000" \
--set admissionController.enabled=false \
--set kyverno.enabled=false \
-n agents --create-namespace
```

### **Features:**

- File, process, and network visibility
- MITRE-based policy enforcement (FIM, cryptojacking protection, etc.)
- Auto-discovery of ingress/egress and whitelisting policies

### **Control Plane Access:**

- PPS: Port 443
- SPIRE: Port 443
- Knox Gateway: Port 3000

# K8s Misconfiguration Scanning

**Deployment Mode:** Kubernetes cronjob

### **Helm Command:**

```
helm upgrade --install k8s-risk-assessment-job  
oci://public.ecr.aws/k9v9d5v2/k8s-risk-assessment-job \  
--set accuknox.tenantID="[TENANTID]" \  
--set accuknox.authToken="[AUTHTOKEN]" \  
--set accuknox.cronTab="30 9 * * *" \  
--set accuknox.clusterName="[CLUSTERNAME]" \  
--set accuknox.URL="cspm.demo.accuknox.com" \  
--set accuknox.label="[LABEL]" \  
--version=v1.1.3
```

### **Features:**

- Detection of misconfigurations and insecure configurations
- Includes checks for root containers, privilege escalation, and 100+ other rules

### **Control Plane Access:**

- HTTPS access to Artifact Endpoint

# K8s Identity & Entitlements Management

**Deployment Mode:** Kubernetes cronjob

**Helm Command:**

```
helm upgrade --install kiem-job oci://public.ecr.aws/k9v9d5v2/kiem-job \
--set accuknox.label="[LABEL]" \
--version v1.1.3 \
--set accuknox.URL="cspm.demo.accuknox.com" \
--set accuknox.authToken="[AUTHTOKEN]" \
--set accuknox.cronTab="30 9 * * *" \
--set accuknox.clusterName="[CLUSTERNAME]" \
--set accuknox.tenantID="[TENANTID]"
```

**Features:**

- Identifies overly permissive role bindings
- Graph-based identity view
- Detection of dangling service accounts and cross-namespace access

**Control Plane Access:**

- HTTPS access to Artifact Endpoint

# K8s CIS Benchmarking

**Deployment Mode:** Kubernetes cronjob

**Helm Command:**

```
helm upgrade --install cis-k8s-job oci://public.ecr.aws/k9v9d5v2/cis-k8s-job \
--set accuknox.url="cspm.demo.accuknox.com" \
--set accuknox.tenantId="[TENANTID]" \
--set accuknox.authToken="[AUTHTOKEN]" \
--set accuknox.cronTab="30 9 * * *" \
--set accuknox.clusterName="[CLUSTERNAME]"
```

```
--set accuknox.label="[LABEL]" \
--version v1.1.3
```

## Features:

- Benchmarks support for:
- Kubernetes (generic)
- EKS
- AKS
- GKE
- OKE not currently supported

## Control Plane Access:

- HTTPS access to Artifact Endpoint

# DISA STIGs Support

**Deployment Mode:** Kubernetes cronjob

## Helm Command:

```
helm upgrade --install k8s-stig-job oci://public.ecr.aws/k9v9d5v2/k8s-stig-job \
--set accuknox.url="cspm.demo.accuknox.com" \
--set accuknox.tenantId="[TENANTID]" \
--set accuknox.authToken="[AUTHTOKEN]" \
--set accuknox.cronTab="30 9 * * *" \
--set accuknox.clusterName="[CLUSTERNAME]" \
--set accuknox.label="[LABEL]" \
--version v1.1.3
```

## Features:

- DISA Special Technical Implementation Guidelines (STIGs) compliance

## Control Plane Access:

- HTTPS access to Artifact Endpoint

# In-Cluster Container Image Scanning

**Deployment Mode:** CronJob (per node job)

**Helm Command:**

```
helm install kubeshield kubeshield-chart \
--set scan.tenantId=<TENANTID> \
--set scan.artifactToken=<TOKEN> \
--set scan.artifactEndpoint="https://cspm.demo.accuknox.com/api/v1/artifact/" \
--set scan.label=<LABEL>"
```

**Features:**

- Direct in-cluster image scanning (no registry access required)
- Scans cached images on nodes
- Reports sent to AccuKnox console for triage

**Control Plane Access:**

- HTTPS access to Artifact Endpoint

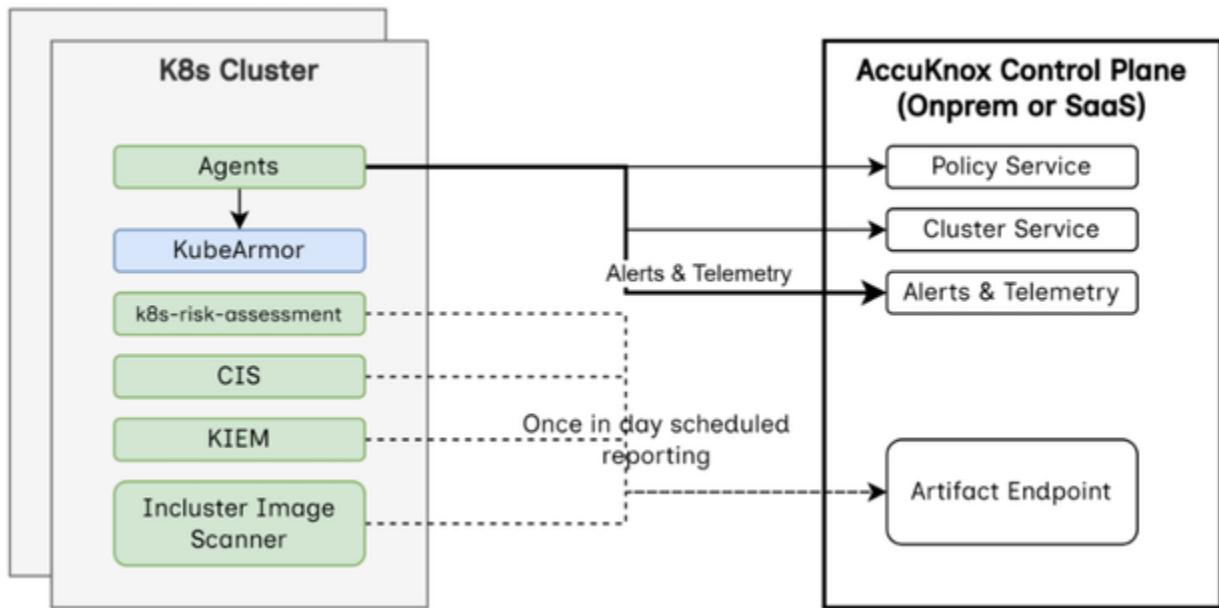
# Admission Controller Support

AccuKnox Admission Controller enforces:

1. Trusted registry enforcement for images
2. Deployment compliance with security best practices (no root, no host mounts, etc.)
3. Violations reported to AccuKnox Control Plane (visible under Monitors & Alerts)

# Cluster Access to Control Plane

Each feature requires outbound (egress) HTTPS access only. Refer to the access notes under each feature for exact service and port requirements.



## Cluster Onboarding

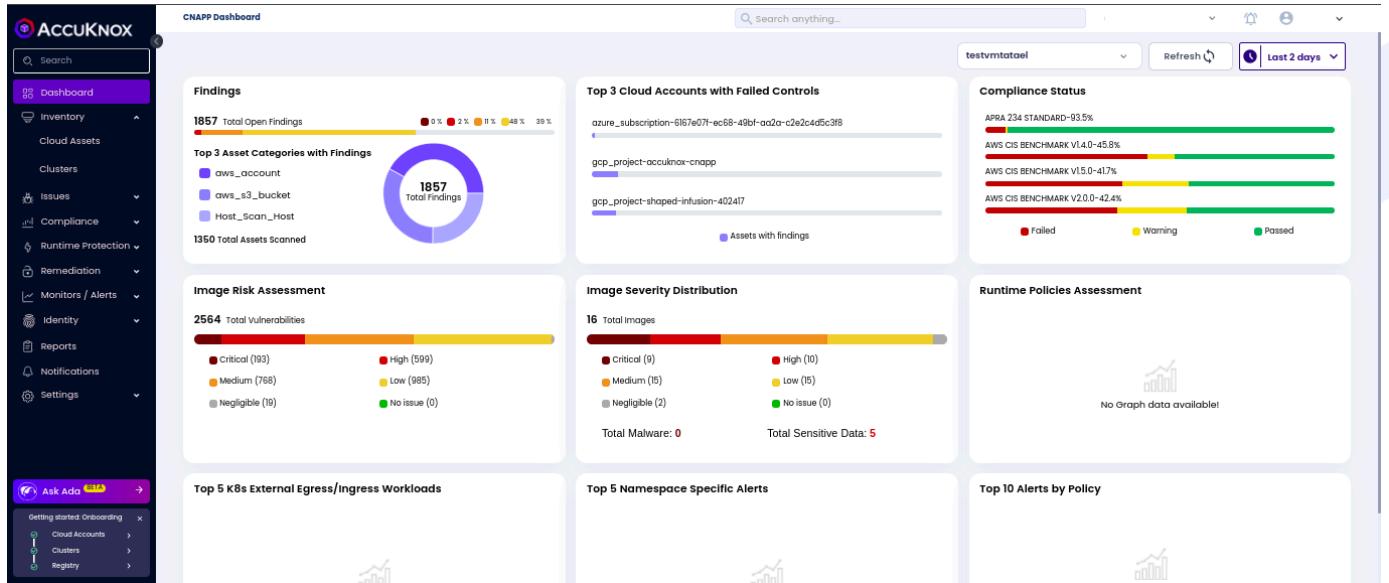
This is a detailed guide on how to onboard clusters to the AccuKnox SaaS platform. The guide covers the installation of KubeArmor and AccuKnox agents in the cluster to connect to the AccuKnox SaaS application.

---

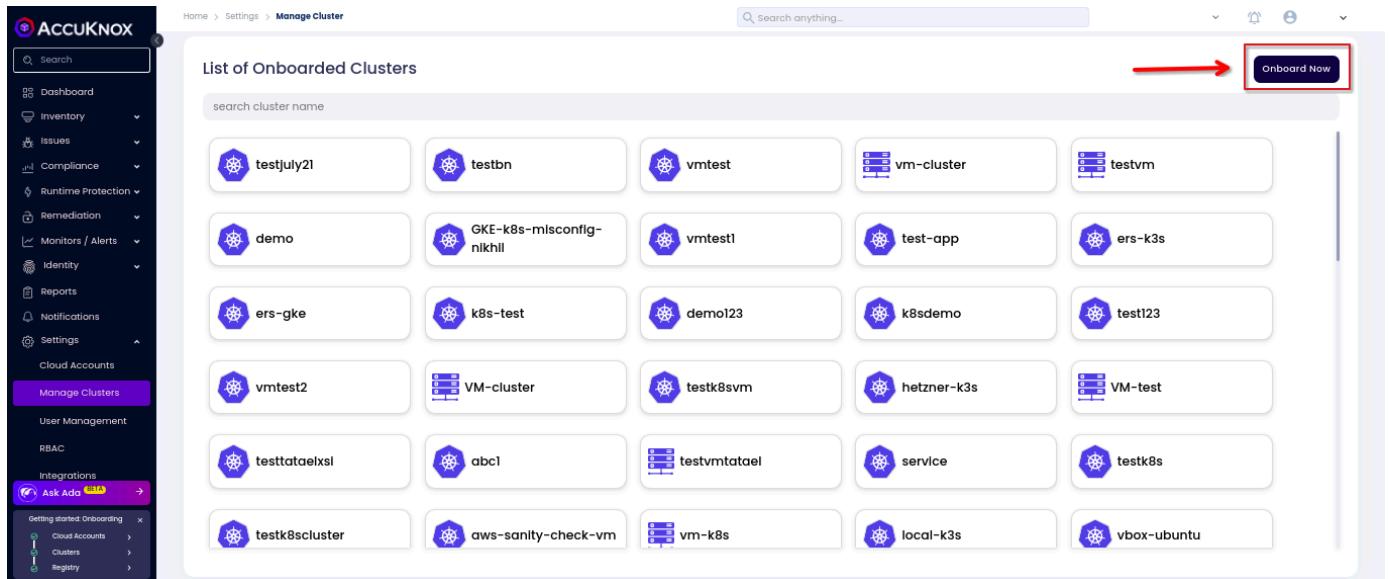
Below shown image is from an k3s cluster running in a local machine with Kali Linux Operating System. We can onboard this cluster by following the steps shown below

```
└$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx-demo 1/1     Running   0          22s
redis-demo 1/1     Running   0          14s
```

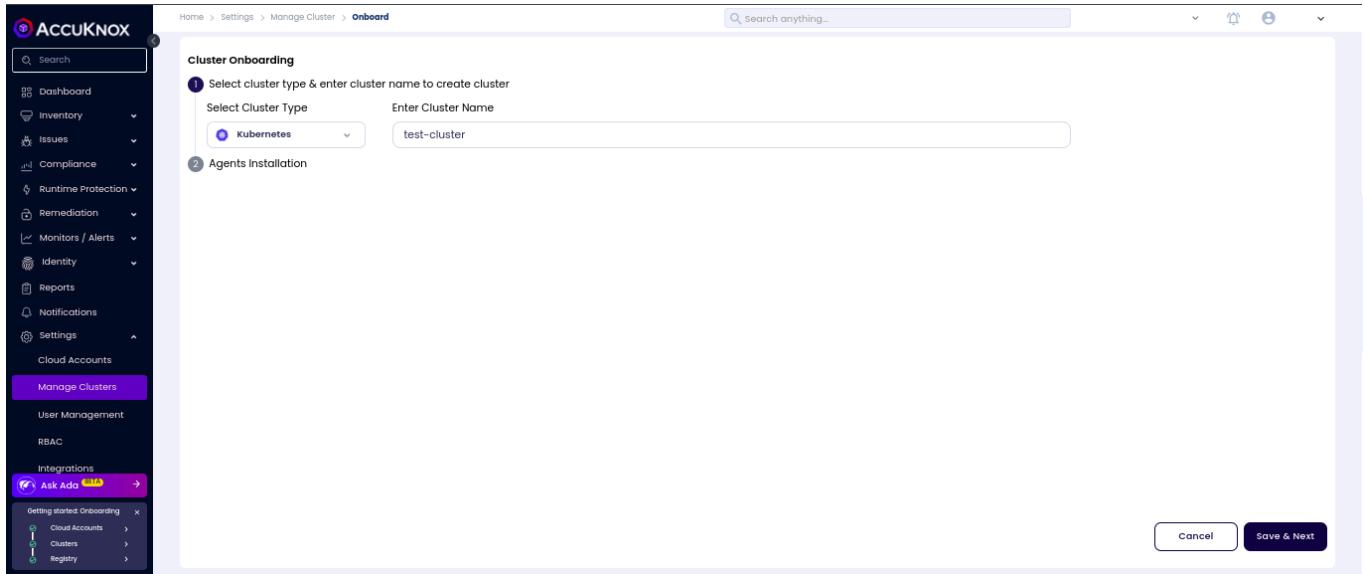
**Step 1:** As a first time user, the management console will show up the CNAPP dashboard without any data mentioned in widgets, since the cloud account and cluster onboarding is not done.



**Step 2:** Navigate to Manage Cluster from Settings Tab: From this page we can onboard the clusters running in various cloud platforms like GCP,AWS and Azure. We can onboard locally setup clusters using an cloud option. To onboard cluster select onboard now option



**Step 3:** In this screen, give any name to the cluster that you are going to onboard now.



#### Step 4: Installing KubeArmor and AccuKnox agents

We are going to install KubeArmor and AccuKnox-agents to connect to the AccuKnox SaaS application. For the agent installation selection click on the Runtime Visibility & Protection.

##### Step 4.1 KubeArmor Installation

#### KubeArmor

KubeArmor is a cloud-native runtime security enforcement system that restricts the behavior (such as process execution, file access, and networking operation) of containers and nodes at the system level.

With KubeArmor, a user can:

- Restrict file system access for certain processes
- Restrict what processes can be spawned within the pod
- Restrict the capabilities that can be used by the processes within the pod

KubeArmor differs from seccomp-based profiles, wherein KubeArmor allows to dynamically set the restrictions on the pod. With seccomp, the restrictions must be placed during the pod startup and cannot be changed later. KubeArmor leverages Linux Security Modules (LSMs) to enforce policies at runtime.

The screenshot shows the AccuKnox web interface with a sidebar containing various navigation options like Dashboard, Inventory, Issues, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports, Notifications, Settings, Cloud Accounts, Manage Clusters, User Management, RBAC, and Integrations. The main area is titled 'Cluster Onboarding' and shows a step-by-step process. Step 1: 'Select cluster type & enter cluster name to create cluster' with 'Kubernetes' selected and 'test-cluster' entered. Step 2: 'Agents Installation'. Under 'Runtime Visibility & Protection', there are sections for KIEM, Kubernetes CIS Benchmark, and Cluster Misconfiguration. Below these are two code snippets in boxes. The first box contains the command: 'curl -sfL http://get.kubearmor.io/ | sudo sh -s -- -b /usr/local/bin'. The second box contains the command: 'karmor install'. Both boxes have a red border around them.

KubeArmor is installed using the following commands:

```
curl -sfL http://get.kubearmor.io/ | sudo sh -s -- -b /usr/local/bin && karmor install
```

### Step 4.2: AccuKnox-Agents installation

After installing KubeArmor we are going to install AccuKnox Agents in the cluster.

# AccuKnox Agents

- KubeArmor:** KubeArmor is a cloud-native runtime security enforcement system that restricts the behavior (such as process execution, file access, and networking operation) of containers and nodes at the system level. KubeArmor dynamically set the restrictions on the pod. KubeArmor leverages Linux Security Modules (LSMs) to enforce policies at runtime.
- Feeder Service:** It collects the feeds from kubeArmor and relays to the app.
- Shared Informer Agent:** It collects information about the cluster like pods, nodes, namespaces etc.,
- Policy Discovery Engine:** It discovers the policies using the workload and cluster information that is relayed by a shared informer Agent.

The screenshot shows the AccuKnox Onboard interface. In the top navigation bar, the path is Home > Settings > Manage Cluster > Onboard. The left sidebar has sections like Dashboard, Inventory, Issues, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports, Notifications, Settings, Cloud Accounts, Manage Clusters, User Management, RBAC, Integrations, and Ask AccuKnox. The main area is titled 'Cluster Onboarding' with two steps: 'Select cluster type & enter cluster name to create cluster' (step 1) and 'Agents Installation' (step 2). Step 2 contains three tabs: 'Runtime Visibility & Protection' (selected), 'KIBM', and 'Kubernetes CIS Benchmark'. Under 'Runtime Visibility & Protection', there are sections for 'Download and Install KubeArmor CLI' (with a curl command) and 'Install KubeArmor'. The 'Install AccuKnox Agents' section contains a command-line interface (CLI) box with a red border around it, containing the following code:

```

1 helm upgrade --install agents oci://registry-1.docker.io/accuknox/accuknox-agents \
2   --version "v0.6.5"
3   --set joinToken="*****_*****_*****"
4   --set spireHost="spire.demo.accuknox.com"
5   --set ppsHost="pps.demo.accuknox.com"
6   --set knoxGateway="knox-gw.demo.accuknox.com:3000"
7   -n agents --create-namespace

```

A 'Finish' button is located at the bottom right of the interface.

AccuKnox Agents can be installed using the following command:

```

helm upgrade --install agents oci://registry-1.docker.io/accuknox/accuknox-agents
--version "v0.6.5"
--set joinToken="*****_*****_*****"
--set spireHost="spire.demo.accuknox.com"
--set ppsHost="pps.demo.accuknox.com"
--set knoxGateway="knox-gw.demo.accuknox.com:3000"
-n agents --create-namespace

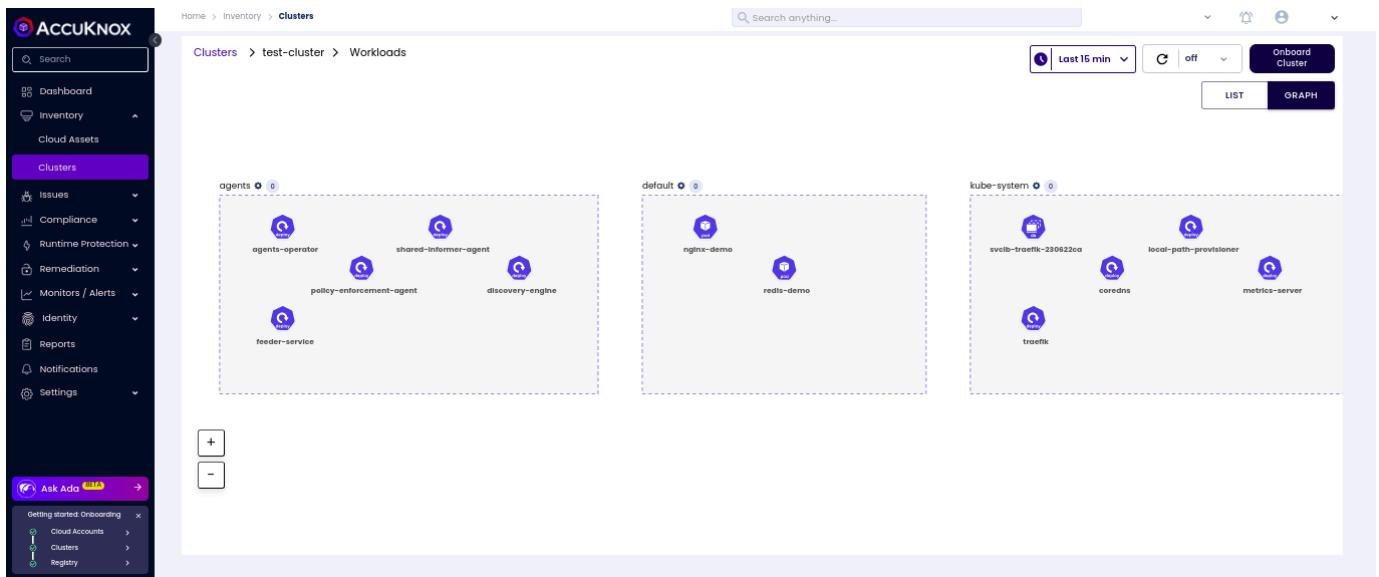
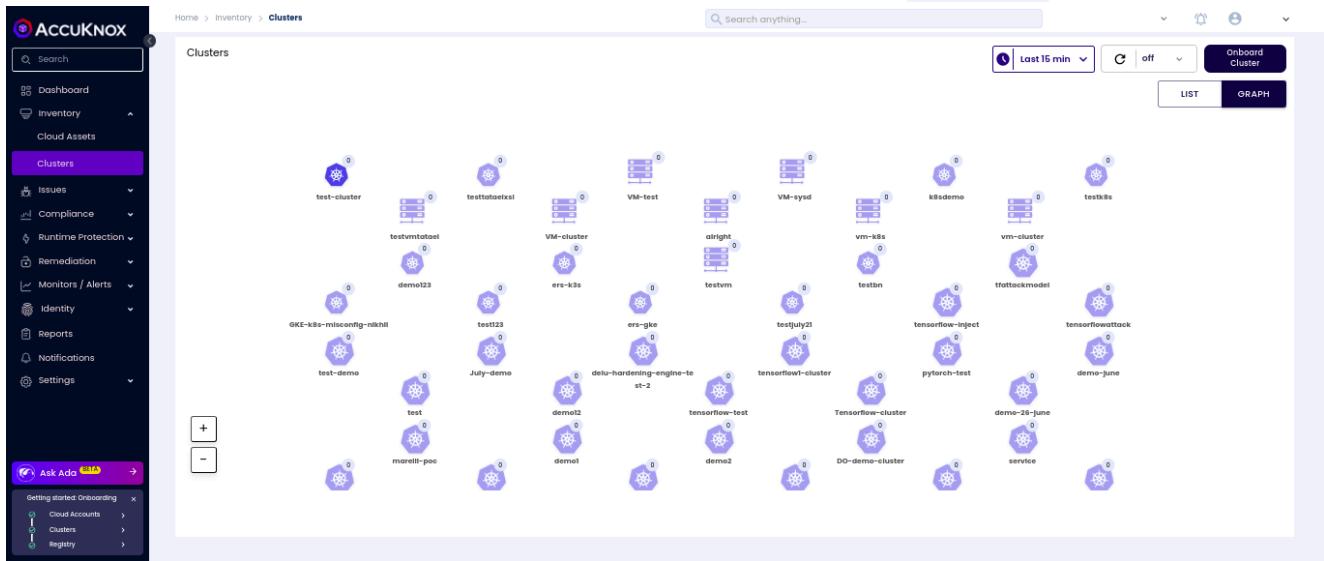
```

## Note

In the above command joinToken is specific to this example and it will vary based on the cluster

## Step 5: Onboarded Cluster

After installing all the AccuKnox agents the cluster is onboarded successfully into the SaaS application. We can see the workload details of the onboarded cluster by Navigating to Inventory→cloud Workloads option. There all the onboarded clusters will be listed out and all the inactive ones would be grayed out. By Double clicking on the active cluster user can get a more detailed view of the cluster.



# Cluster Onboarding with Access Keys

Streamlining cluster onboarding is made easy with access keys, allowing users to onboard multiple clusters using the same key. Additionally, users can set expiration times for these keys and specify the number of clusters each key can onboard. This process can be performed directly from the CLI if the access key is already created, offering enhanced flexibility and convenience.

## Pre-requisite:

1. Kubernetes (managed/un-manager) cluster

2. AccuKnox CNAPP login access
3. One or more clusters to onboard
4. Access Key (See how to [create](#))

# Onboarding

In the case of the Access key onboarding method, the User can directly onboard the VMs from the CLI, To Onboard a new cluster follow the below steps:

## Step1: Install KubeArmor

```
curl -sfL http://get.kubearmor.io/ | sudo sh -s -- -b /usr/local/bin  
karmor install
```

Output:

```
kubearmor/kubearmor-client info checking GitHub for latest tag  
kubearmor/kubearmor-client info found version: 1.3.0 for v1.3.0/linux/amd64  
kubearmor/kubearmor-client info installed /usr/local/bin/karmor  
kubearmor/kubearmor-client info karmor is installed in /usr/local/bin  
kubearmor/kubearmor-client info invoke /usr/local/bin/karmor or move karmor to your  
desired PATH
```

```
$ karmor install  
🛡️ Installed helm release : kubearmor-operator  
😊 KubeArmorConfig created  
⌚ This may take a couple of minutes  
🔥 KubeArmor Snitch Deployed!  
🔥 KubeArmor Daemonset Deployed!  
✅ Done Checking , ALL Services are running!  
⌚ Execution Time : 58.615464051s  
  
🔧 Verifying KubeArmor functionality (this may take upto a minute)...  
  
🛡️ Your Cluster is Armored Up!
```

## Step2: Install AccuKnox Agents

## AccuKnox-Agents:

The AccuKnox Agent is a K8s operator that installs the following agents:

- Feeder service: It collects KubeArmor feeds.
- Shared-informer-agent: This agent authenticates with your cluster and collects information regarding entities like nodes, pods, and namespaces.
- Policy-enforcement-agent: This agent authenticates with your cluster and enforces labels and policies.
- Discovery Engine: Discovery Engine discovers the security posture for your workloads and auto-discovers the policy set required to put the workload in least-permissive mode. The engine leverages the rich visibility provided by KubeArmor to auto-discover systems and network security postures.

The agent-operator also manages the agents' resource limits. The operator is in charge of spawning the agents based on the size of the cluster. If the cluster size changes, i.e., new nodes are added or existing nodes are deleted, then the operator scales up or down the resources accordingly.

AccuKnox Agents can be installed using the following command:

```
helm upgrade --install agents oci://registry-1.docker.io/accuknox/accuknox-agents \
--version "v0.5.11" \
--set spireHost="spire.demo.accuknox.com" \
--set ppsHost="pps.demo.accuknox.com" \
--set knoxGateway="knox-gw.demo.accuknox.com:3000" \
--set tokenURL="cwpp.demo.accuknox.com" \
--set clusterName="accuknoxcluster" \
--set accessKey=<token>" \
-n accuknox-agents --create-namespace
```

### Note

In the commands above, substitute **--set clusterName** with the desired cluster name, and replace the **<token>** with the **Access Keys** generated from UI. Adjust the URLs if required

### Note

Please check for the value of --version "v0.0.0" from the UI steps of cluster onboarding to make sure you are using the latest image tags

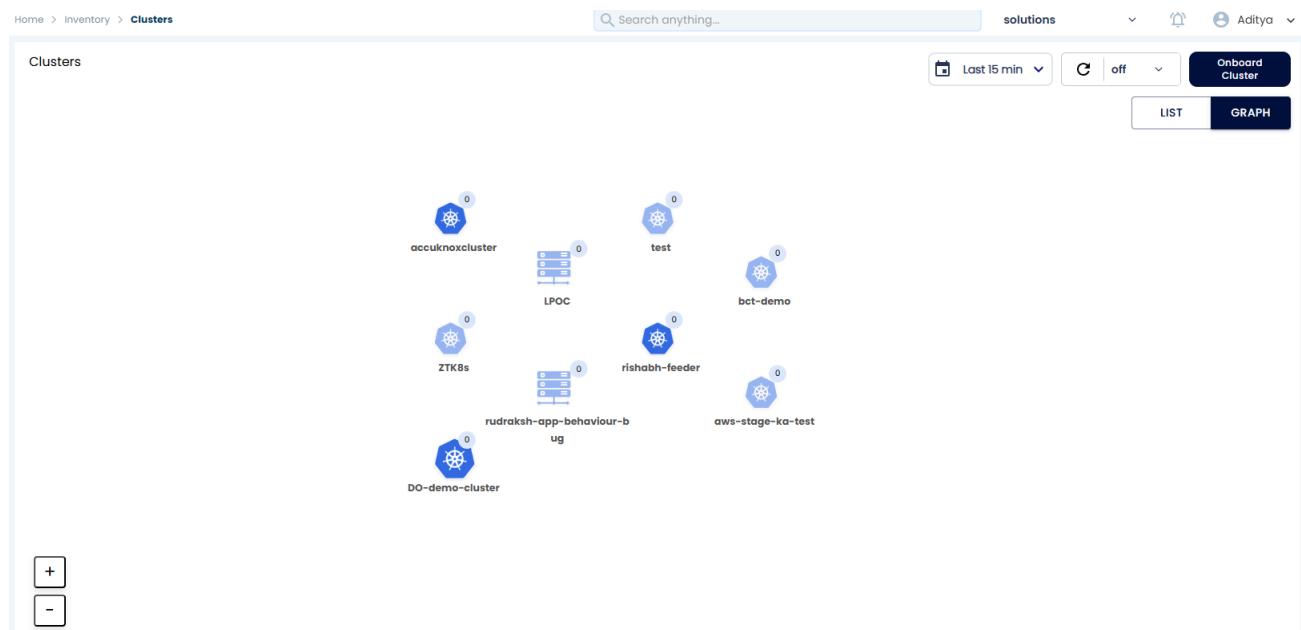
### Output

```
Release "agents" does not exist. Installing it now.  
Pulled: registry-1.docker.io/accuknox/accuknox-agents:v0.5.11  
Digest: sha256:6b7870020c0470741b7a89f47fd6f4e85882521721ce50407351d231508c6aaaf  
NAME: agents  
LAST DEPLOYED: Thu Jan 2 19:05:38 2025  
NAMESPACE: accuknox-agents  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None
```

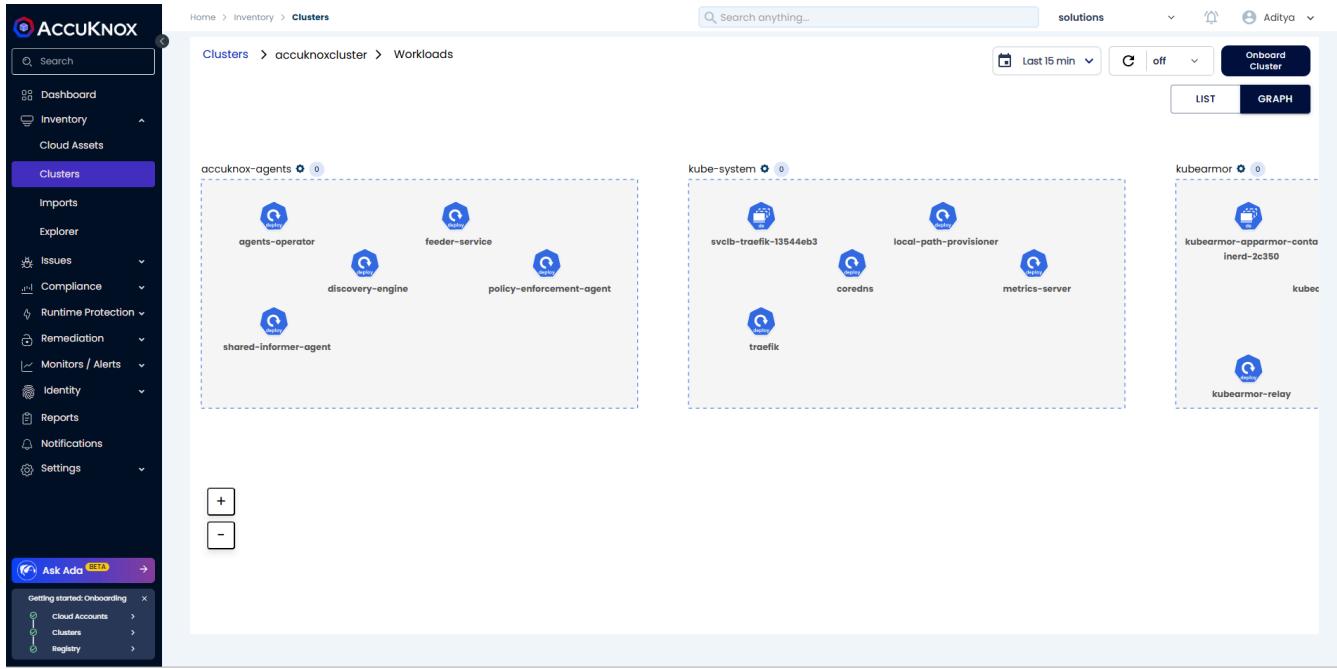
To verify please use

```
kubectl get po -n accuknox-agents
```

After installing all the AccuKnox agents, the cluster is onboarded successfully into the SaaS application. We can see the workload details of the onboarded cluster by Navigating to Inventory-> Clusters



## View the workloads



## Note

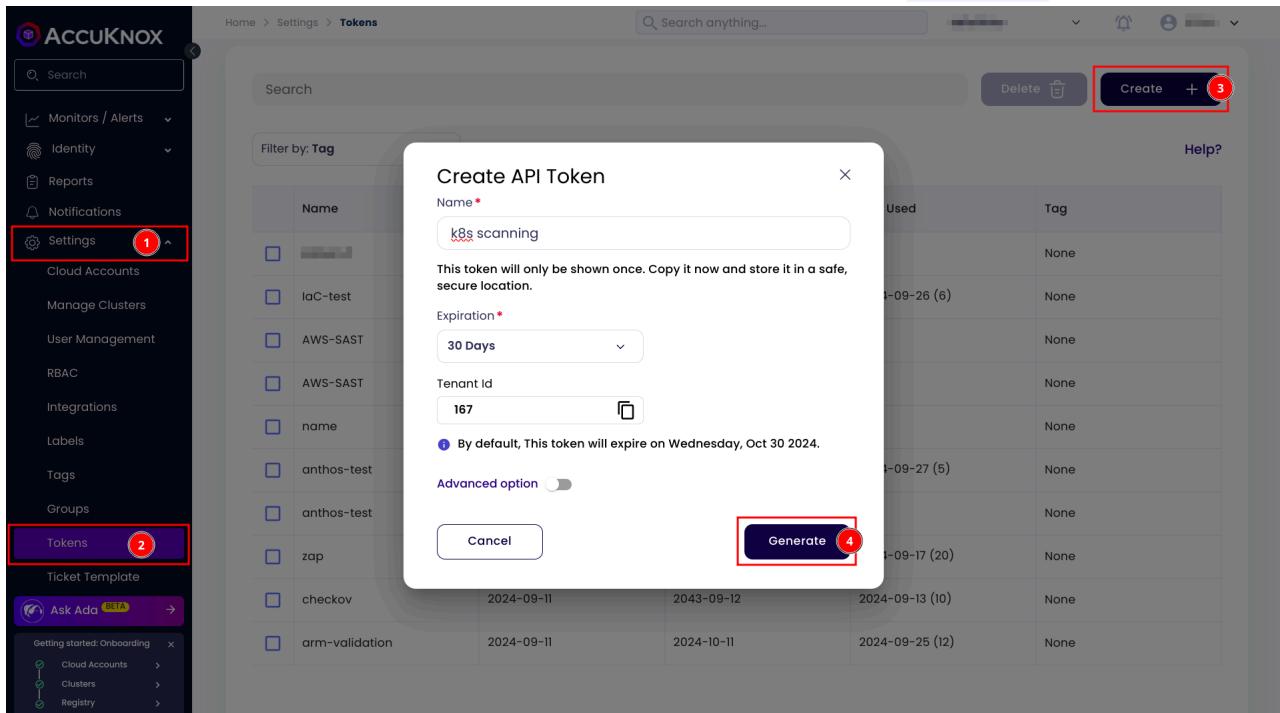
You can repeat the same command with different "**clusterName**" to onboard multiple cluster using access keys

# Onboard Cluster for Misconfiguration Scanning

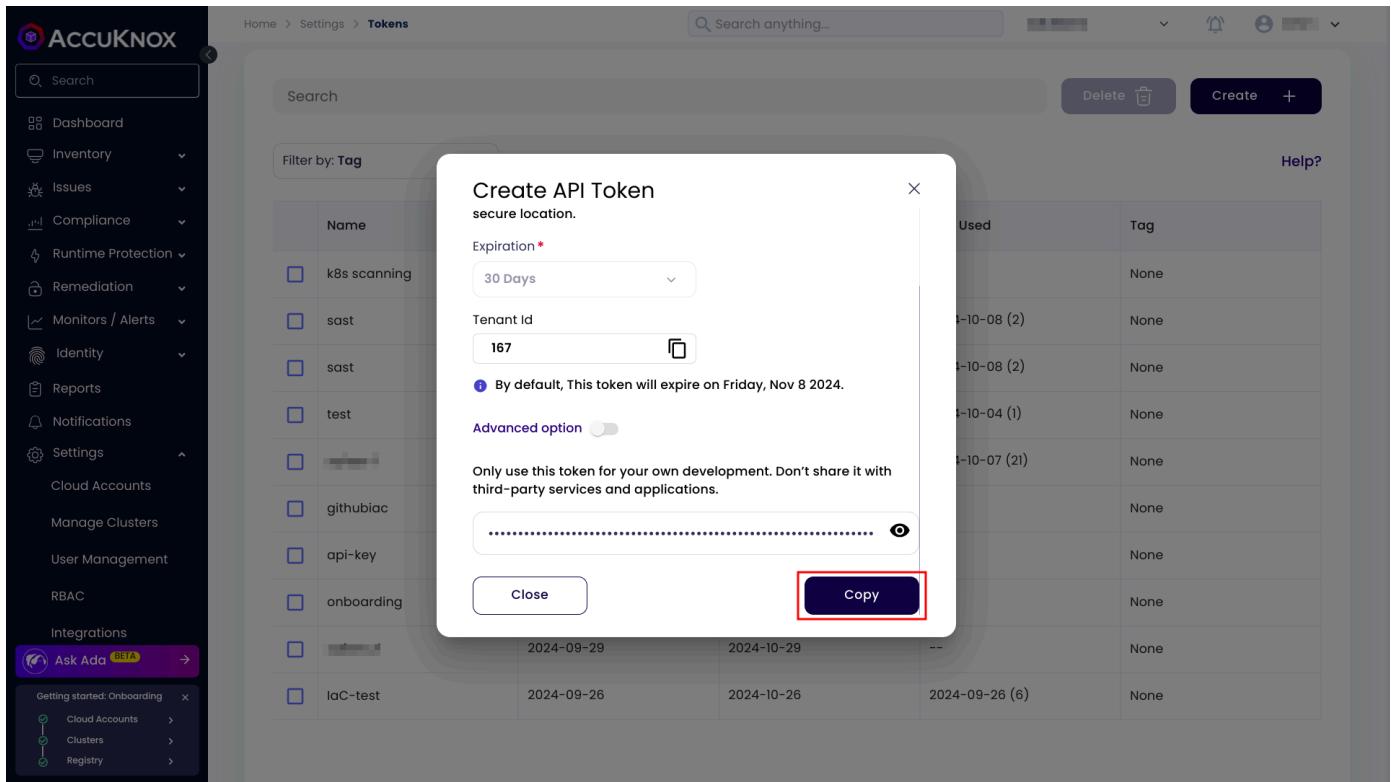
This guide outlines the steps for onboarding a cluster to AccuKnox SaaS for scanning cluster misconfigurations.

For onboarding a cluster and for scanning for misconfigurations you need to create a token first. For creating follow these steps:

Go to **Settings > Tokens** and click on the create button. Give your token a name and click on generate button.



Once the token is generated, copy it and take a note of it.



Now go to Settings > Manage Clusters, click on onboard now button or select an existing cluster.

The screenshot shows the AccuKnox interface. The left sidebar contains navigation items like Monitors / Alerts, Identity, Reports, Notifications, Settings (highlighted with a red box and number 1), Manage Clusters (highlighted with a red box and number 2), User Management, RBAC, Integrations, Labels, Tags, Groups, Tokens, and Ticket Template. A purple banner at the bottom left says 'Ask Ada' with a 'GET STARTED' button. The main content area is titled 'List of Onboarded Clusters' and shows a grid of 12 clusters with icons and names: insecure-scan, [redacted], [redacted], vm-1, vault5, [redacted], s-poc, [redacted], [redacted], ers1, [redacted], testvincent, gclus-0901, [redacted], tensorflowattack, DO-demo-cluster, and gke-demo-cluster. A search bar at the top says 'Search anything...'. The top right corner has a blue button labeled 'Onboard Now' with a red box and number 3.

Give your cluster a name. Under the Agents Installation section select Cluster Misconfiguration. Select a label and paste your token.

The screenshot shows the 'Cluster Onboarding' page. The left sidebar is identical to the previous one. The main content area is titled 'Cluster Onboarding' and has two steps: 1. 'Select cluster type & enter cluster name to create cluster' (with 'Kubernetes' selected) and 2. 'Agents Installation' (with 'Cluster Misconfiguration' highlighted with a red box and number 1). Below this is a 'Runtime Visibility & Protection' section with 'KIEM' and 'Kubernetes CIS Benchmark' options. The 'Schedule' section shows a cron-like schedule 'DEMO \* \* \* \*' (highlighted with a red box and number 2) and a note '(Server TimeZone: UTC) At 09:30 AM (User TimeZone: IST) At 03:00 PM next scan at: 2024-10-09 09:30:00 AM next scan at: 2024-10-09 03:00:00 PM'. The 'Prerequisites' section lists 'helm (v3.13.1 or later)' and 'token (Click to generate a new token)'. At the bottom right is a 'Finish' button (highlighted with a red box and number 3).

You can also change the schedule as per your requirement. Then next scan will happen based on the schedule. Scroll down and copy the helm command and run it inside a terminal. Then click on Finish button.

The screenshot shows the AccuKnox interface for cluster onboarding. The left sidebar has a purple 'Manage Clusters' button highlighted. The main area is titled 'Cluster Onboarding' and shows step 2 of 2: 'Agents Installation'. It includes fields for 'Select Cluster Type' (set to 'Kubernetes') and 'Enter Cluster Name' (set to 'Demo'). Below these are sections for 'Runtime Visibility & Protection', 'KIEM', 'Kubernetes CIS Benchmark', and 'Cluster Misconfiguration' (which is selected and highlighted in blue). A 'Prerequisites' section lists 'helm' (v3.13.1 or later) and 'token' (with a link to generate a new token). A 'Install k8s-risk-assessment-job' section contains a Helm command:

```
1 helm upgrade --install k8s-risk-assessment-job oci://public.ecr.aws/k9v9d5v2/k8s-risk-a
2   --set accuknox.tenantID=xxxx \
3   --set accuknox.authToken='xxxx'
4   --set accuknox.cronTab="30 9 * * *"
5   --set accuknox.clusterName="Demo"
6   --set accuknox.URL="cspm.demo.accuknox.com"
7   --set accuknox.label="DEMO"
8   --version=0.1.0
```

Annotation 1 points to the 'authToken' placeholder in the command. Annotation 2 points to the 'Finish' button at the bottom right of the page.

Once the scan is completed you can see the results on the findings page.

1. Go to the [Issues > Findings](#) page.
2. Select the Cluster Finding from the drop down.

The screenshot shows the ACCUKNOX interface. On the left, a dark sidebar lists various security categories like Dashboard, Inventory, Issues, Findings (which is highlighted with a red box and has a red number 1), Registry Scan, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports, Notifications, and Settings. Below this is an 'Ask Ada (BETA)' section. The main content area is titled 'Findings' and shows a table of findings. The table has columns for Identification numbers, Name, Assetname, Risk factor, and Pkg name. One row is highlighted with a red box and a red number 2, corresponding to the 'Cluster Findings' entry in the dropdown menu above the table.

Identification numbers	Name	Assetname	Risk factor	Pkg name
CVE-2023-4911, CWE-787, c...	glibc: buffer overflow in ...	jfrog.gcp.accuknox.com...	High	ld-linux
CVE-2023-6246, CWE-787, c...	glibc: heap-based buff...	jfrog.gcp.accuknox.com...	High	glibc
2024-10-09 12:50:01	CVE-2024-33602, CWE-46...	glibc: netgroup cache a...	Medium	ld-linux
2024-10-09 12:50:01	CVE-2023-4527, CWE-125, i...	glibc: Stack read overfl...	Medium	ld-linux
2024-10-09 12:50:01	CVE-2024-33600, CWE-476...	glibc: null pointer deref...	Medium	ld-linux
2024-10-09 12:50:01	CVE-2024-33601, CWE-617...	glibc: netgroup cache ...	Medium	ld-linux
2024-10-09 12:50:01	CVE-2023-6779, CWE-787, g...	glibc: off-by-one heap-...	High	glibc
2024-10-09 12:50:01	CVE-2023-6246, CWE-787, g...	glibc: heap-based buff...	High	ld-linux
2024-10-09 12:50:01	CVE-2023-5156, CWE-401...	glibc: DoS due to memo...	High	ld-linux

Click on any of the findings to see more details.

This screenshot shows the 'Findings' section with a different filter applied. The 'Group by' dropdown is set to 'Last seen'. The table now includes a column for 'Tool output'. The first row, which corresponds to the entry highlighted in the previous screenshot, is also highlighted with a red box.

Last seen	Name	Risk factor	Assetname	Tool output
2024-10-09 10:55:45	Applications credentials in configuration files	High	mysql	FAILED
2024-10-09 08:40:16	Applications credentials in configuration files	High	cis-k8s-cronjob	FAILED
2024-09-30 17:06:29	Anonymous access enabled	High	kubeadm:bootstrap-sig...	FAILED
2024-10-02 15:52:54	Applications credentials in configuration files	High	reporter-config	FAILED
2024-10-02 15:15:22	Applications credentials in configuration files	High	mysql	FAILED
2024-09-30 17:06:29	Applications credentials in configuration files	High	k8s-risk-assessment-jo...	FAILED
2024-10-09 08:40:16	Anonymous access enabled	High	system:public-info-vie...	FAILED
2024-09-30 17:06:29	Anonymous access enabled	High	system:public-info-vie...	FAILED
2024-07-27 11:10:13	Anonymous access enabled	High	system:public-info-vie...	FAILED

**Applications credentials in configuration files**

Age	Severity	SLA	Tickets Created
7 days	High	45 days	3

**Description** Result Solution References Source Code

Attackers who have access to configuration files can steal the stored secrets and use them. This control checks if ConfigMaps or pod specifications have sensitive information in their configuration.

- Finding for in resource k8s\_security\_Deployment | mysql
- Failing since about 6 day ago, on 03/10/2024
- Last detected on 09/10/2024

**Compliance Frameworks**  
No compliance found

**Asset Information**

```
{
  "id": "3bea491c-7049-4645-b292-2b6e8299f20c",
  "tickets_count": 0,
  "data_type": "cluster-misconfiguration",
  "hash": "89d735e8eae43e55c68578df3edb0d57",
  "history": [
    {
      "changed": {},
      "scan_id": "0ef547b5-e688-40ec-bbb4-32460505d01",
      "timestamp": "2024-10-03 03:53:49.195581+00:00"
    }
  ]
}
```

**Details** + Create Ticket

Asset mysql  
Asset Type k8s\_security\_Deployment  
Status Active  
Ignored No

**Notes** Add Comments and Press Ctrl + Enter to Submit

No data

# CIS Benchmarking Compliance Scan Onboarding

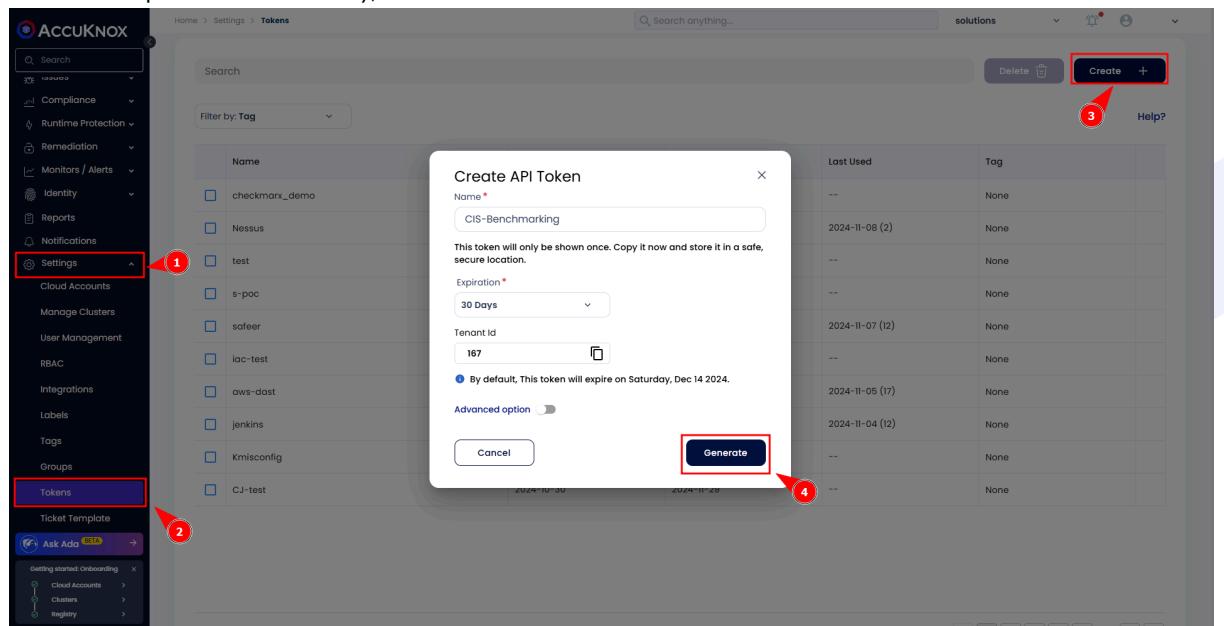
This guide details the steps to onboard a Kubernetes cluster to Accuknox SaaS for CIS Benchmarking compliance scanning, enabling you to monitor and improve cluster security in line with CIS standards.

## Step 1: Generate an Access Token

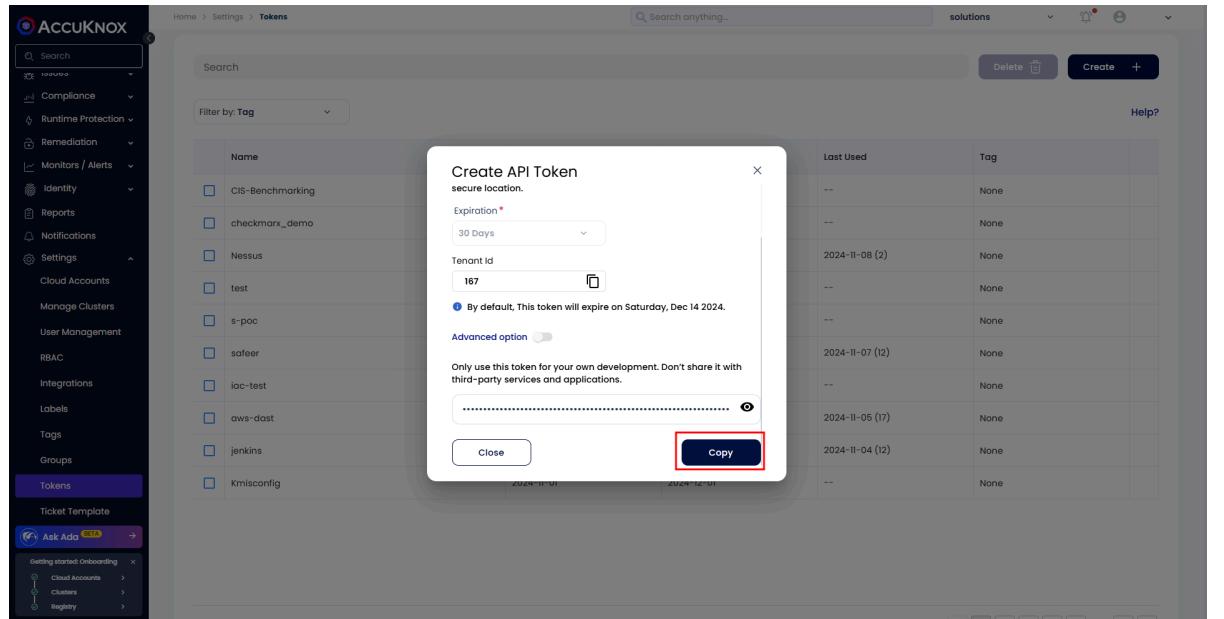
To begin, create a token that will authenticate your cluster for scanning. Follow these steps:

1. Navigate to **Settings > Tokens** in the Accuknox platform and Click on the **Create** button, give your token a descriptive name (e.g.,

"CIS-Compliance-Token"), and click **Generate**.



2. Once the token is generated, copy it and securely save it for later use.



## Step 2: Onboard Your Cluster

1. Go to **Settings > Manage Clusters** and Click **Onboard Now** or select an existing cluster if you're updating a previously onboarded cluster.

The screenshot shows the AccuKnox dashboard with the 'Manage Cluster' section selected. The main area displays a grid of 16 clusters, each with a small icon and a name. At the top right of this grid is a red box containing the 'Onboard Now' button. To the left of the grid, there are three numbered callouts: '1' points to the 'Manage Clusters' link in the sidebar; '2' points to the 'Ask Ada' button; and '3' points to the 'Onboard Now' button.

2. Enter a name for your cluster to identify it in Accuknox. From the scan type, choose **CIS Benchmarking**.
3. Select a label for easy identification and paste the token you generated in Step 1. Set a scan schedule based on your requirements. Accuknox will automatically run scans according to the selected schedule.

The screenshot shows the 'Cluster Onboarding' page. The 'Kubernetes CIS Benchmark' option is selected under 'Select cluster type & enter cluster name to create cluster'. In the 'Agents Installation' section, the 'Kubernetes CIS Benchmark' job is selected. The 'Prerequisites' section lists 'helm (v3.13 or later)' and 'token (Click to generate a new token)'. The 'Install CIS Benchmark Job' section contains a command line input field with the command: 'helm upgrade --install cis-k8s-job oci://public.ecr.aws/k9v9d5v2/cis-k8s-job \'. A 'Back' button is visible at the bottom right.

# Step 3: Deploy the Scanner Using Helm

1. Scroll down to the **Helm Command** section and copy the provided command.

The screenshot shows the AccuKnox interface for cluster onboarding. Step 1: Select cluster type & enter cluster name to create cluster. Step 2: Agents Installation. Under Runtime Visibility & Protection, 'Kubernetes CIS Benchmark' is selected. The 'Prerequisites' section lists 'helm (v3.13.1 or later)' and 'token (Click to generate a new token)'. The 'Install CIS Benchmark Job' section contains a red box around the following Helm command:

```
1 helm upgrade --install cis-k8s-job oci://public.ecr.aws/k9v9d5v2/cis-k8s-job \
2 --set accuknox.tenantId="1" \
3 --set accuknox.authToken="" \
4 --set accuknox.cronTab="0 9 * * *" \
5 --set accuknox.clusterName="CIS-Benchmarking" \
6 --version v1.1.3
7
```

The note below the command states: "The same Helm command has the capacity to facilitate the installation of multiple Jobs. The only variable that requires modification is the 'clusterName'".

2. Run this command in your terminal on a machine that has access to your Kubernetes cluster. The command will schedule the scan for CIS Benchmarking compliance.
3. Once the Helm installation is complete, return to the Accuknox platform and click **Finish**.

# Step 4: View Compliance Findings

After the initial scan is completed, you can view the compliance results:

1. Go to **Issues > Findings** in Accuknox.

2. Use the **Findings** dropdown to filter and select CIS k8s Benchmarking finding results.

The screenshot shows the ACCUKNOX interface with the 'Findings' section selected in the sidebar. The main table displays a list of findings related to the CIS K8s Benchmark. The columns include Assetname, Test number, Tool output, Cvss score, Description, Solution, Type, and Group text. Most findings show a 'FAILED' status with a Cvss score of 0.0. A specific row for 'DO-demo-cluster' is highlighted with a red border.

Assetname	Test number	Tool output	Cvss score	Description	Solution	Type	Group text
safeer	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...		
Test-k3s	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...		
DO-demo-cluster	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...		
insecure-scan	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...		
DO-demo-cluster	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...		
rk-k8s-stage	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...		
Test-k3s	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...		
safeer	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...		
insecure-scan	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...		
Test-k3s	I.1.3	FAILED	0.0	Ensure that the controll...	Run the below command...		
safeer	I.1.3	FAILED	0.0	Ensure that the controll...	Run the below command...		
insecure-scan	I.1.3	FAILED	0.0	Ensure that the controll...	Run the below command...		

3. Each result will provide details on specific CIS controls and any non-compliant configurations detected.

This screenshot shows the same ACCUKNOX interface, but the row for 'DO-demo-cluster' from the previous screenshot is now highlighted with a red border. The table includes additional columns: Expected result and Assetlabel. The 'DO-demo-cluster' row is now fully visible, showing its specific configuration details.

Assetname	Test number	Tool output	Cvss score	Description	Solution	Expected result	Assetlabel
safeer	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...	'permissions' is present	SAFEER
Test-k3s	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...	'permissions' is present	CISTEST
DO-demo-cluster	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...	'permissions' is present	KIEMDO
insecure-scan	I.1.1	FAILED	0.0	Ensure that the API serv...	Run the below command...	'permissions' is present	Test101
DO-demo-cluster	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...	'rootroot' is present	KIEMDO
rk-k8s-stage	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...	'rootroot' is present	nessus
Test-k3s	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...	'rootroot' is present	CISTEST
safeer	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...	'rootroot' is present	SAFEER
insecure-scan	I.1.2	FAILED	0.0	Ensure that the API serv...	Run the below command...	'rootroot' is present	Test101
Test-k3s	I.1.3	FAILED	0.0	Ensure that the controll...	Run the below command...	'permissions' is present	CISTEST
safeer	I.1.3	FAILED	0.0	Ensure that the controll...	Run the below command...	'permissions' is present	SAFEER
insecure-scan	I.1.3	FAILED	0.0	Ensure that the controll...	Run the below command...	'permissions' is present	Test101

This completes the onboarding process for CIS Benchmarking compliance scanning. You can review findings regularly to maintain and improve your cluster's CIS compliance.

## Cluster Offboarding

This guide outlines the steps for offboarding a cluster from AccuKnox SaaS. The process involves uninstalling the agents from the cluster and deleting the cluster from AccuKnox SaaS.

Below, you will find detailed instructions for agent uninstallation from your cluster CLI and deleting the cluster from AccuKnox SaaS. These steps apply to all clusters.

## Agents Uninstallation

Uninstall AccuKnox agents using the following commands:

```
helm uninstall agents -n agents && kubectl delete ns agents;
```

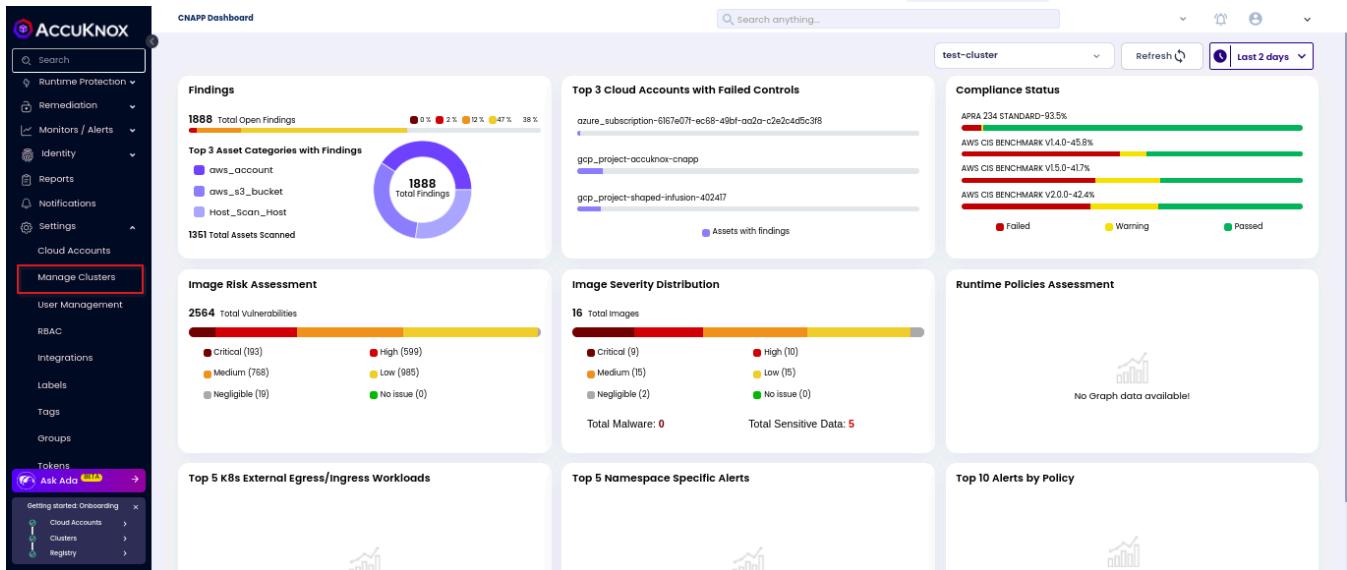
```
helm uninstall cis-k8s-job;  
helm uninstall kiem-job;  
helm uninstall k8s-risk-assessment-job
```

## Sample for Uninstalling Runtime Visibility & Protection agents

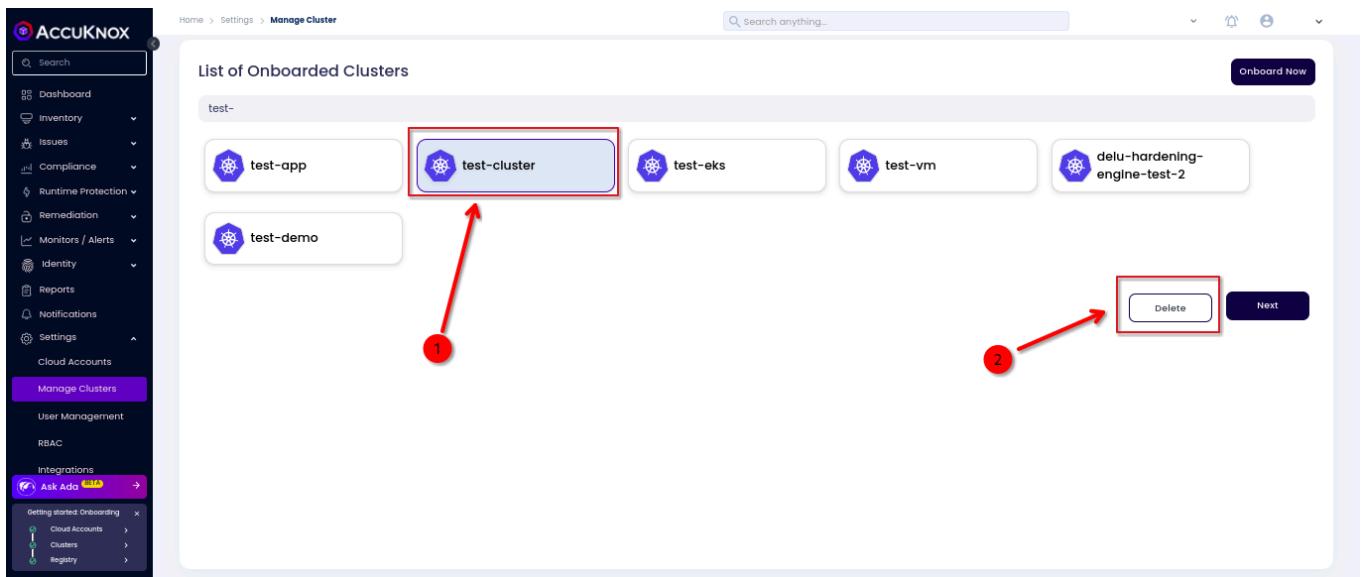
```
(AccuKnox㉿kali)-[~]  
└$ helm uninstall agents -n agents && kubectl delete ns agents  
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location:  
/etc/rancher/k3s/k3s.yaml  
WARNING: Kubernetes configuration file is world-readable. This is insecure. Location:  
/etc/rancher/k3s/k3s.yaml  
release "agents" uninstalled  
namespace "agents" deleted
```

## Cluster Deletion

**Step 1:** Login to AccuKnox SaaS and Go to Manage Cluster under Settings



**Step 2:** Select the cluster and click Delete to delete the cluster from SaaS.



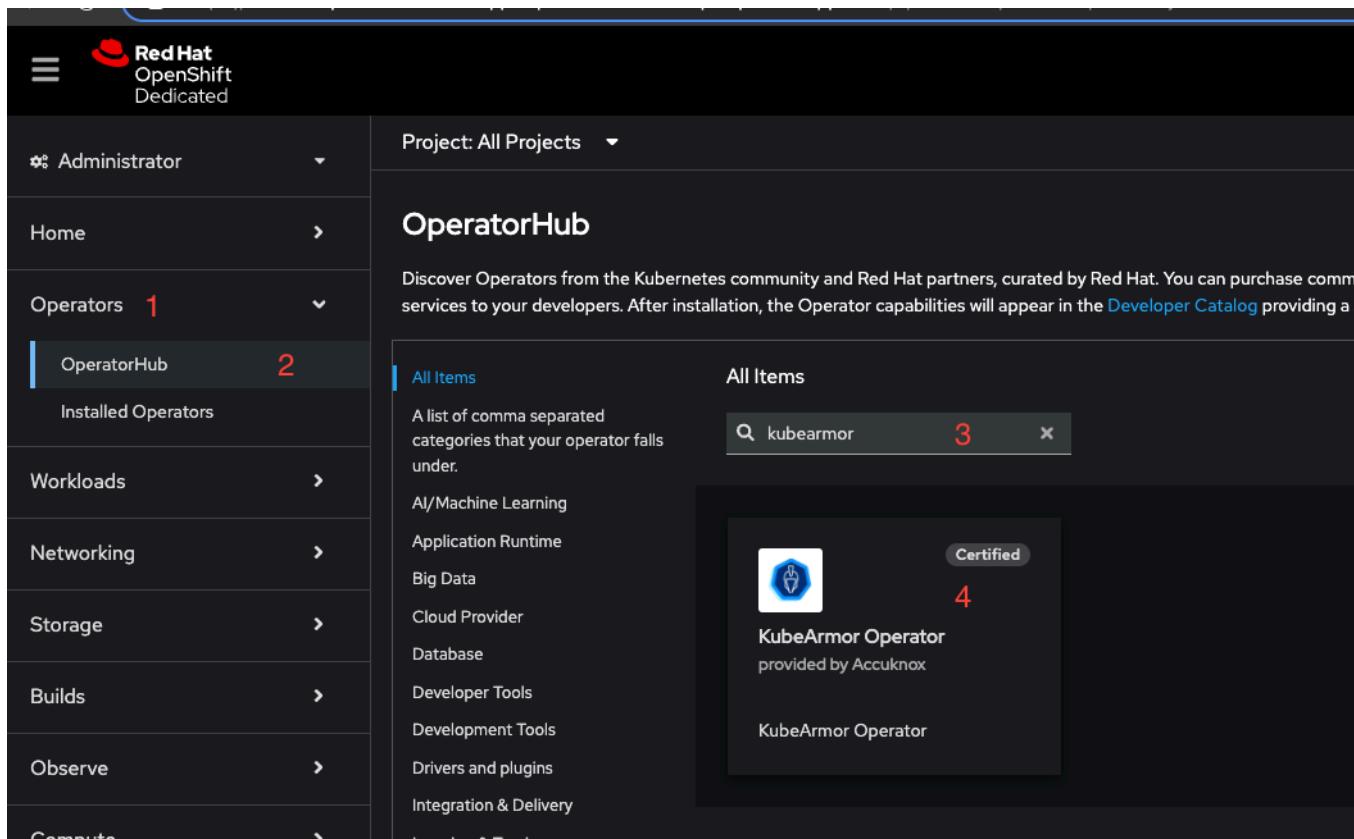
This will delete the cluster from AccuKnox SaaS.

# Runtime Security Deployment for Openshift

# Operator Installation

In the OpenShift console, install KubeArmor operator by following the instructions below:

- Under operators (1) select Operator Hub (2).
- Search for the word "kubearmor" (3) and select "KubeArmor Operator" (4).
- Install KubeArmor version "1.4.9" with default configurations (5, 6, 7).





## KubeArmor Operator

1.4.9 provided by Accuknox

Install

6

### Channel

stable

### Version

1.4.9

5

KubeArmor is a cloud-native runtime security solution for Kubernetes. It provides fine-grained control over process execution, file access, and network traffic at the container level. KubeArmor leverages the Container Security Model (CSM) to enforce the user-specified security policies across all containers in a cluster.

### Features and Benefits

OperatorHub > Operator Installation

## Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines ei

**Update channel \*** ?

stable

**Version \***

1.4.9

**Installation mode \***

All namespaces on the cluster (default)  
This mode is not supported by this Operator

A specific namespace on the cluster  
Operator will be available in a single Namespace only.

**Installed Namespace \***

Operator recommended Namespace: PR kubearmor

Select a Namespace

**Namespace creation**  
Namespace `kubearmor` does not exist and will be created.

**Update approval \*** ?

Automatic

Manual

---

7

**Install** **Cancel**

# ElasticSearch Integration

To integrate KubeArmor with Elasticsearch, the following inputs are required:

- **Username/Password:** If the Elasticsearch server requires authentication.
- **CA Certificate:** If Elasticsearch security is enabled.
- **URL of Elasticsearch:** Including protocol and port.

## Steps to Install

### Username/Password Installation

If the server does not require authentication, you can skip this step. To use username/password authentication with Elasticsearch, a Kubernetes secret called `elastic-secret` needs to be created in the `kubearmor` namespace.

Run the following command, replacing `<elastic-user>` and `<elastic-password>` with appropriate values:

```
kubectl create secret generic elastic-secret -n kubearmor --from-literal  
username=<elastic-user> --from-literal password=<elastic-password>
```

### CA Certificate Installation

To use HTTPS communication between the agents and Elasticsearch, a Kubernetes secret called `elastic-ca` needs to be created in the `kubearmor` namespace.

- Acquire the CA certificate used by Elasticsearch. If acquiring the certificate is not possible, set the `allowInsecureTLS` flag to true in the next steps.
- Save the certificate in a file and run the following command:

```
kubectl create secret generic elastic-ca -n kubearmor --from-file ca.crt=<cacert file name>
```

---

# KubeArmor Instance Installation

Once the steps in the previous chapter are completed, proceed with the agent installation from the OpenShift console.

## Steps to Install

1. Install the required SCC using the following command:

```
oc create -f  
https://raw.githubusercontent.com/kubearmor/KubeArmor/main/pkg/KubeArmorOperator/co  
nfig/rbac/kubearmor-scc.yaml
```

1. In the OpenShift console:
2. Under Operators (1), go to Installed Operators (2).
3. Select kubearmor (3) as the project.
4. Click on the KubeArmor Operator (4).
5. Create a KubeArmorConfig Instance (5).
6. In the form view:
7. Select **Adapters** (6) -> **Elasticsearch Adapter** (7).
8. Perform the following steps:
  - Enter the Elasticsearch URL in the field (8).
  - Enable Elasticsearch adapter by checking the checkbox (9).
  - Click on **Elasticsearch Authentication** (10) and:
    - Set the CA secret field (11) to elastic-ca.
    - To enable insecure TLS communication (if no certificate is available), check the allowInsecureTLS checkbox (11-b) and leave the field (11) empty.
9. Create the instance. The KubeArmorConfig Instance controls the installation of the agents in the entire cluster, and only one instance should be created per cluster.

Administrator 3

Home >

Operators 1 <span>1</span>

- OperatorHub
- Installed Operators 2 <span>2</span>
- Workloads >
- Networking >
- Storage >
- Builds >

Project: kubearmor <span>3</span>

## Installed Operators

Installed Operators are represented by ClusterServiceVersions within the project.

Name	Managed Namespaces
 Elasticsearch (ECK) Operator 2.16.0 provided by Elastic	NS kubearmor The operator is running multiple operators but is managed by one NS.
 KubeArmor Operator 4 1.4.9 provided by Accuknox	NS kubearmor

Installed Operators > Operator details

 KubeArmor Operator  
1.4.9 provided by Accuknox

Details YAML Subscription Events

### Provided APIs

**KAC KubeArmorConfig**

KubeArmorConfig is the Schema for the kubearmorconfigs API

 Create instance 5 <span>5</span>

**Adapters** 6

KubeArmor Relay Adapters

**Elasticsearch adapter** 7

Elasticsearch Adapter

Elasticsearch Endpoint URL

8

Elasticsearch endpoint url

Enable/Disable Elasticsearch adapter

enabled 9

Enable/Disable Elasticsearch Adapter

Elasticsearch index

kubearmor-alerts

Elasticsearch index mapping for kubearmor alerts

**Elasticsearch Authentication** 10

Elasticsearch Authentication Credentials

**Elasticsearch Authentication**

Elasticsearch Authentication Credentials

Elasticsearch Authentication Secret

elastic-secret

Elasticsearch Authentication Kubernetes Secret Name

Elasticsearch Password Key

password

Elasticsearch Authentication Kubernetes Secret Password Key

Allow Insecure TLS

allowInsecureTLS 11-b

Allow insecure tls communication for elasticsearch

CA Secret

11

CA Secret for secure tls communication to elasticsearch

# Kibana Dashboard Setup

Steps to Install

Along with this document, a file called `kubearmor-dashboard.ndjson` has been shared. Follow these steps to import the dashboard:

1. Under the **Management** tab, select **Stack Management**.
2. Navigate to **Saved Objects** under Kibana.
3. Click **Import** and select `kubearmor-dashboard.ndjson`.

The dashboard displays the following data:

- Total Events**:
  - Audit**: Count of records: 9
  - Block**: Count of records: 24
- Block events by namespace**:
  - default**: Block events: 24
- Namespace by Workload**:

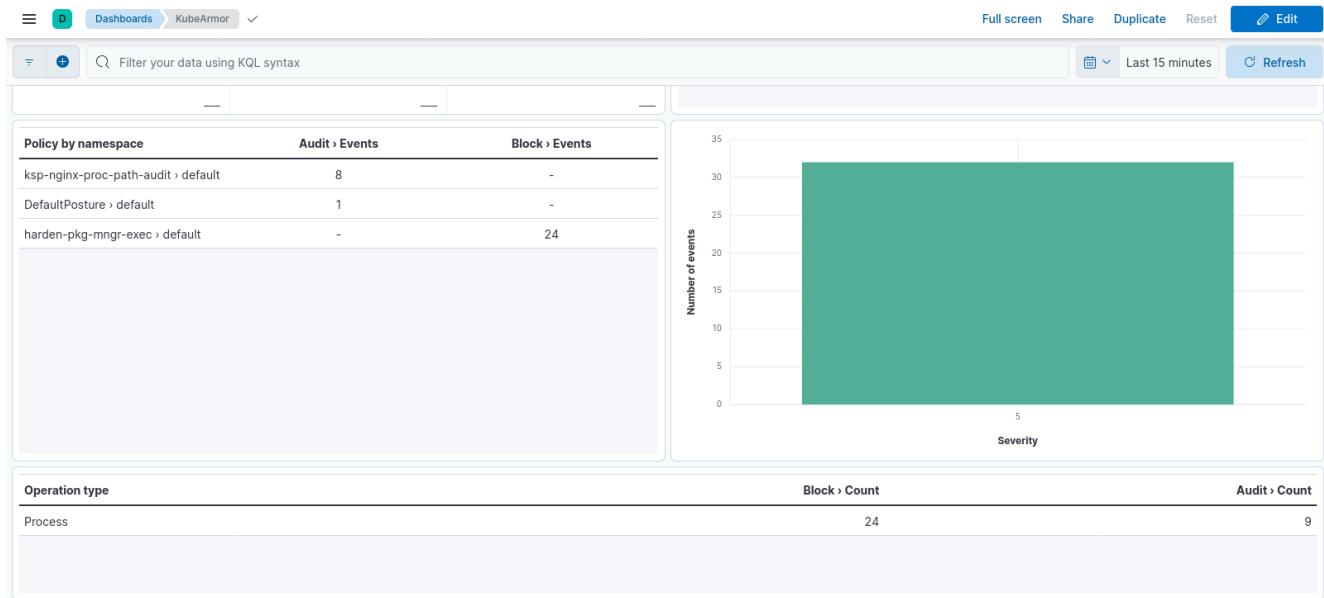
	Audit > Count	Block > Count
default > nginx	9	24

The dashboard displays the following data:

- Block events by namespace**:
  - default**: Block events: 24
- Namespace by Workload**:

	Audit > Count	Block > Count
default > nginx	9	24
- Policy by namespace**:

Policy	Audit > Events	Block > Events
ksp-nginx-proc-path-audit > default	8	-
DefaultPosture > default	1	-
harden-pkg-mngr-exec > default	-	24
- Number of events over time**: A bar chart showing the number of events from 15 to 35. The chart is filled with a solid teal color.



## Onboarding and Deboarding VMs with Docker

### Docker

Docker v19.0.3 and Docker Compose v1.27.0+ are required. Follow the latest [Install Docker Engine](#) for downloading. Ensure you also add your user to the docker user group: [Linux post-installation steps for Docker Engine](#).

Linux Kernel v5.8+ with BPF LSM support is needed. See how to [enable BPF LSM](#).

If the environment does not support Linux v5.8+ or BPF LSM and instead uses AppArmor, host enforcement will still work out of the box. However, to protect containers, new containers must be created with special options. Refer to the "[Support for Non-Orchestrated Containers](#)" documentation for more details.

### Resource Requirements

<b>Node Type</b>	<b>vCPU</b>	<b>Memory</b>	<b>Disk</b>
Control Plane Node	2	4 GB	24 GB
Worker Node	2	2 GB	12 GB

## Network Requirements

Connectivity between control plane node and worker nodes is a must. They should either be:

- Part of the same private network (**recommended & secure**)
- Control plane has a public IP (not recommended)

Ports required on the control plane VM:

<b>Component</b>	<b>Type</b>	<b>Ports</b>	<b>Endpoint</b>	<b>Purpose</b>
Knox-Gateway	Outbound to SaaS	3000	knox-gw.<env>.accuknox.com:3000	For Knox-Gat service

PPS	Outbound to SaaS	443	pps.<env>.accuknox.com	For PPS (Policy Provisioning Service)
Spire-Server	Outbound to SaaS	8081, 9090	spire.<env>.accuknox.com	For Spire-Server communication
KubeArmor Relay Server	Inbound in Control Plane	32768	-	For KubeArmor relay server on control plane
Shared Informer Agent	Inbound in Control Plane	32769	-	For Shared Informer agent on control plane
Policy Enforcement Agent (PEA)	Inbound in Control Plane	32770	-	For Policy Enforcement Agent on control plane
Hardening Module	Inbound in Control Plane	32771	-	For Discovery Hardening Module
VM Worker Nodes	Outbound from worker node to Control Plane	32768-32 771	-	For VM worker nodes to connect to control plane

By default, the network created by onboarding commands reserves the subnet 172.20.32.0/27. If you want to change it for your environment, you can use the `--network-cidr` flag.

You can check the connectivity between nodes using curl. Upon a successful connection, the message returned by curl will be:

```
$ curl <control-plane-addr>:32770  
curl: (1) Received HTTP/0.9 when not allowed
```

# Onboarding

Navigate to the onboarding page (Settings → Manage Cluster → Onboard Now) and choose the "VM" option on the instructions page. Then, provide a name for your cluster. You will be presented with instructions to download accuknox-cli and onboard your cluster.

The following agents are installed:

1. **Feeder-service** which collects KubeArmor feeds.
2. **Shared-informer-agent** authenticates with your VMs and collects information regarding entities like hosts, containers, and namespaces.
3. **Policy-enforcement-agent** authenticates with your VMs and enforces labels and policies.

## Install knoxctl/accuknox-cli

```
curl -sfL https://knoxctl.accuknox.com/install.sh | sudo sh -s -- -b /usr/bin
```

## Onboarding Control Plane

The command may look something like this:

```
$ knoxctl onboard vm Control-Plane-node \  
--version "v0.2.10" \  
--join-token="843ef458-cecc-4fb9-b5c7-9f1bf7c34567" \  
--spire-host="spire.dev.accuknox.com" \  
--pps-host="pps.dev.accuknox.com" \  
--knox-gateway="knox-gw.dev.accuknox.com:3000"
```

The above command will emit the command to onboard worker nodes. You may also use the `--Control Plane-node-addr` flag to specify the address that other nodes will use to connect with your cluster.

By default, the network created by onboarding commands reserves the subnet `172.20.32.0/27` for the accuknox-net Docker network. If you want to change it for your environment, you can use the `--network-cidr` flag.

## Onboarding Worker Nodes

The second command will be for onboarding worker nodes. It may look something like this:

```
knoxctl onboard vm node --Control Plane-node-addr=<control-plane-addr>
```

Example:

```
$ knoxctl onboard vm node --Control Plane-node-addr=192.168.56.106
Pulling kubearmor-init ... done
Pulling kubearmor ... done
Pulling kubearmor-vm-adapter ... done
Creating network "accuknox-config_accuknox-net" with the default driver
Creating kubearmor-init ... done
Creating kubearmor ... done
Creating kubearmor-vm-adapter ... done
onboard-vm-node.go:41: VM successfully joined with control-plane!
```

# Troubleshooting

If you encounter any issues while onboarding, use the commands below to debug:

```
docker logs spire-agent -f
docker logs shared-informer-agent -f
docker logs kubearmor-init -f
docker logs kubearmor -f
```

# Deboarding

Deboard the cluster from SaaS first.

To deboard the worker-vm/Node:

```
knoxctl deboard vm node
```

To deboard the Control-Plane VM:

```
knoxctl deboard vm Control-Plane-node
```

Sample Output:

```
$ knoxctl deboard vm Control-Plane-node
[+] Running 10/10
✓ Container shared-informer-agent    Removed          0.6s
✓ Container feeder-service           Removed          0.6s
✓ Container policy-enforcement-agent Removed        0.8s
✓ Container wait-for-it             Removed        0.0s
✓ Container kubearmor-vm-adapter   Removed        5.6s
✓ Container kubearmor-relay-server Removed      1.5s
✓ Container spire-agent            Removed        0.5s
✓ Container kubearmor              Removed      10.4s
✓ Container kubearmor-init         Removed        0.0s
✓ Network accuknox-config_accuknox-net Removed     0.3s
Please remove any remaining resources at /home/user/.accuknox-config
Control plane node deboarded successfully.
```

After that cleanup the ~/accuknox-config directory

```
sudo rm -rf ~/accuknox-config
```

## Onboarding and Deboarding VMs with Systemd

## Systemd

**Systemd** is a core component of modern Linux systems responsible for managing services and processes. It ensures that essential services start automatically during boot, remain running, and restart if they fail. In simple terms, systemd acts like a **controller** that organizes and oversees everything needed to keep the system stable and functional.

Currently, **root/sudo** permissions are needed for onboarding systemd. This is because KubeArmor requires privileges to protect the host and systemd services, packages are currently installed on the root directory.

Only in case of the control plane node, a working RabbitMQ server is required. This can be installed using Docker.

```
# Latest RabbitMQ 3.13
docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.13-management
```

Alternatively, you can install RabbitMQ using a package manager:

- **Linux, BSD, UNIX:** [Debian](#), [Ubuntu](#) | [RHEL](#), [CentOS Stream](#), [Fedora](#) | [Generic binary build](#) | [Solaris](#)
- **Windows:** [Chocolatey package](#) | [Windows Installer](#) | [Binary build](#)
- **MacOS:** [Homebrew](#) | [Generic binary build](#)
- [Erlang/OTP for RabbitMQ](#)

BTF support is needed. Any kernel version which has this should work. Check if BTF info is present with the script below:

```
if [ ! -e "/sys/kernel/btf/vmlinux" ]; then
    echo "BTF info not present"
else
    echo "BTF info present"
fi
```

If the script returns "BTF info not present," [BTF support is not available](#), and you should run the script below to build the required files on your system:

```
# Download KubeArmor
git clone https://github.com/kubearmor/KubeArmor/
cd KubeArmor/KubeArmor/packaging
./post-install.sh
```

## Note

For detailed instructions specific to SystemD Based Non-BTF Environments, please refer to this [guide](#).

### **Container Protection Requirements (Optional)**

If container protection is needed, a Linux Kernel with **BPF LSM** is desired. Generally, it is present in v5.8+. Here's a guide on enabling BPF LSM: [KubeArmor Getting Started FAQ](#).

If BPF LSM is not available, AppArmor should still work out of the box for host policy application. However, follow the guide [Support for non orchestrated containers](#) for each container.

## Resource Requirements

### **Control Plane Node (Minimum)**

<b>Resourc e</b>	<b>Require ment</b>
CPU	2 vCPU
Memory	4 GB
Disk	1 GB

### **Worker Node (Minimum)**

Resource	Requirement
CPU	2 vCPU
Memory	2 GB
Disk	500 MB

## Network Requirements

Connectivity between control plane node and worker nodes is a must. They should either be:

- Part of the same private network (**recommended & secure**)
- Control plane has a public IP (not recommended)

Ports required on the control plane VM:

Component	Type	Ports	Endpoint	Purpose
Knox-Gateway	Outbound to SaaS	3000	knox-gw.<env>.accuknox.com:3000	For Knox-Gat service

PPS	Outbound to SaaS	443	pps.<env>.accuknox.com	For PPS (Policy Provisioning Service)
Spire-Server	Outbound to SaaS	8081, 9090	spire.<env>.accuknox.com	For Spire-Server communication
KubeArmor Relay Server	Inbound in Control Plane	32768	-	For KubeArmor relay server on control plane
Shared Informer Agent	Inbound in Control Plane	32769	-	For Shared Informer agent on control plane
Policy Enforcement Agent (PEA)	Inbound in Control Plane	32770	-	For Policy Enforcement Agent on control plane
Hardening Module	Inbound in Control Plane	32771	-	For Discovery Hardening Module
VM Worker Nodes	Outbound from worker node to Control Plane	32768-32 771	-	For VM worker nodes to connect to control plane

Check the CWPP documentation for more details on the [network requirements](#).

You can check the connectivity between nodes using curl. Upon a successful connection, the message returned by curl will be:

```
$ curl <control-plane-addr>:32770  
curl: (1) Received HTTP/0.9 when not allowed
```

# Onboarding

Navigate to the onboarding page (Settings → Manage Cluster → Onboard Now) and choose the "VM" option on the instructions page. Then, provide a name for your cluster. You will be presented with instructions to download accuknox-cli and onboard your cluster.

The following agents will be installed:

1. **Feeder-service** which collects KubeArmor feeds.
2. **Shared-informer-agent** authenticates with your VMs and collects information regarding entities like hosts, containers, and namespaces.
3. **Policy-enforcement-agent** authenticates with your VMs and enforces labels and policies.

## Install knoxctl/accuknox-cli

```
curl -sfL https://knoxctl.accuknox.com/install.sh | sudo sh -s -- -b /usr/bin
```

## Onboarding Control Plane

The command may look something like this:

```
$ knoxctl onboard vm cp-node \  
--version "v0.2.10" \  
--join-token="843ef458-cecc-4fb9-b5c7-9f1bf7c34567" \  
--spire-host="spire.dev.accuknox.com" \  
--pps-host="pps.dev.accuknox.com" \  
--knox-gateway="knox-gw.dev.accuknox.com:3000"
```

## Note

By default, if Docker is not found, systemd mode of installation would be used. If you want to explicitly onboard using systemd services, add the `--vm-mode=systemd` flag to the above command.

The above command will emit the command to onboard worker nodes. You may also use the `--cp-node-addr` flag to specify the address that other nodes will use to connect with your cluster.

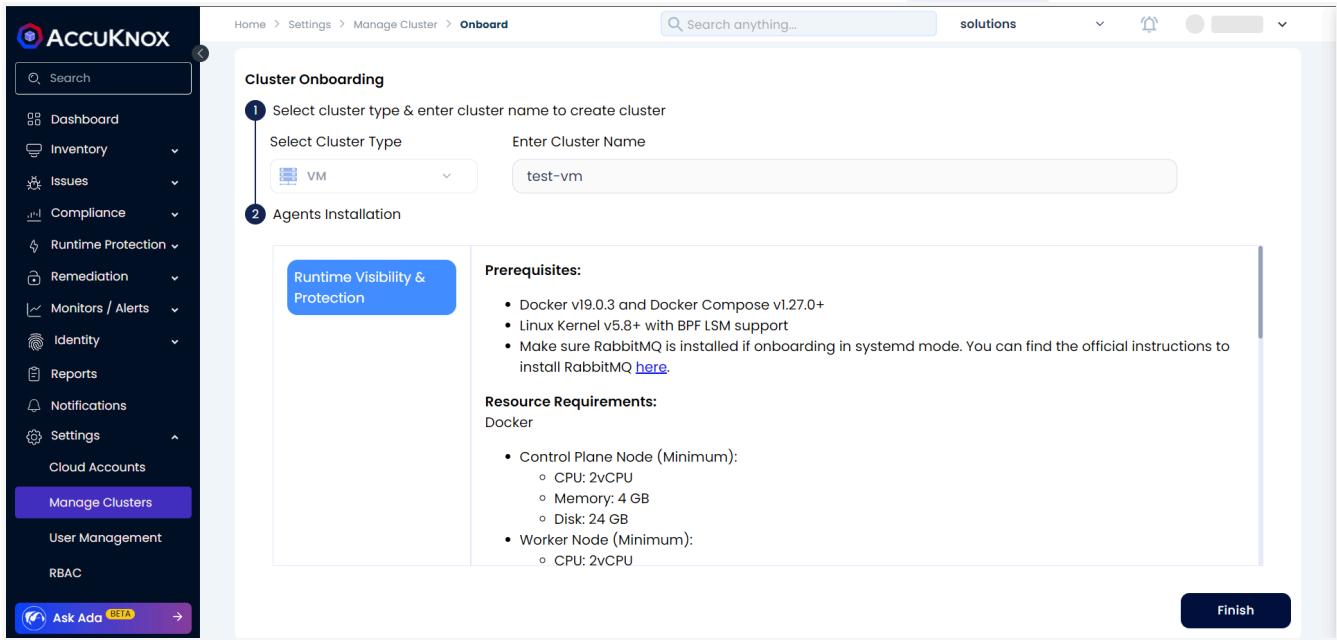
# Onboarding Worker Nodes

The second command will be for onboarding worker nodes. It may look something like this:

```
knoxctl onboard vm node --cp-node-addr=<control-plane-addr>
```

Example:

```
$ knoxctl onboard vm node --cp-node-addr=192.168.56.106
Pulling kubearmor-init ... done
Pulling kubearmor ... done
Pulling kubearmor-vm-adapter ... done
Creating network "accuknox-config_accuknox-net" with the default driver
Creating kubearmor-init ... done
Creating kubearmor ... done
Creating kubearmor-vm-adapter ... done
onboard-vm-node.go:41: VM successfully joined with control-plane!
```



# Troubleshooting

If you encounter any issues while onboarding, use the commands below to debug:

```
sudo journalctl -xeu <service-name>.service
```

Replace `<service-name>` with one of the following:

- `kubearmory`: Logs show policy enforcement and monitor Kubernetes workloads; useful for debugging misconfigurations or runtime issues.
- `kubearmory-relay-server`: Bridges KubeArmor clients with external log systems; logs debug communication or relay errors.
- `kubearmory-vm-adapter`: Tracks policy enforcement in VMs; logs diagnose policy application on non-Kubernetes workloads.
- `accuknox-policy-enforcement-agent`: Enforces security policies; logs troubleshoot policy errors or conflicts.
- `accuknox-shared-informer-agent`: Shares Kubernetes resource data; logs debug metadata collection issues.
- `accuknox-sumengine`: Processes telemetry data; logs resolve performance or data processing errors.
- `accuknox-discover-agent`: Discovers potential policies; logs analyze policy suggestions.
- `spire-agent`: Manages workload identities; logs debug identity issuance and attestation issues.

- accuknox-hardening-agent: Automates system hardening; logs troubleshoot configuration and hardening conflicts.

# Deboarding

Deboard the cluster from SaaS first.

To deboard the worker-vm/Node:

```
knoxctl deboard vm node
```

To deboard the Control-Plane VM:

```
knoxctl deboard vm cp-node
```

Sample Output:

```
$ knoxctl deboard vm cp-node
[+] Running 10/10
✓ Container shared-informer-agent    Removed          0.6s
✓ Container feeder-service           Removed          0.6s
✓ Container policy-enforcement-agent Removed          0.8s
✓ Container wait-for-it             Removed          0.0s
✓ Container kubearmor-vm-adapter   Removed          5.6s
✓ Container kubearmor-relay-server Removed          1.5s
✓ Container spire-agent            Removed          0.5s
✓ Container kubearmor              Removed          10.4s
✓ Container kubearmor-init         Removed          0.0s
✓ Network accuknox-config_accuknox-net Removed          0.3s
Please remove any remaining resources at /home/user/.accuknox-config
Control plane node deboarded successfully.
```

After that cleanup the ~/accuknox-config directory

```
sudo rm -rf ~/accuknox-config
```

# SystemD Based Non-BTF Environments

# Compiling system monitor

Some Kernels don't have BTF information available which is required by KubeArmor's system monitor to work out of the box. Thus, the monitor has to be built either on the target machine or on a machine which matches the kernel version of the target machine.

There are two ways to do it, you can chose either one:

## Compile system monitor using Docker (Recommended and reliable)

1. Dependencies:
  - Make sure you have docker installed
  - Make sure you have linux-headers installed for your package
2. Run the kubearmor-init container using the below command which will generate the file /tmp/system\_monitor.bpf.o.

```
sudo docker run --rm -d --name=kubearmor-init --privileged \
-v "/tmp:/opt/kubearmor/BPF:rw" \
-v "/lib/modules:/lib/modules:ro" \
-v "/sys/kernel/security:/sys/kernel/security:ro" \
-v "/sys/kernel/debug:/sys/kernel/debug:ro" \
-v "/media/root/etc/os-release:/media/root/etc/os-release:ro" \
-v "/usr/src:/usr/src" \
kubearmor/kubearmor-init:stable
```

## Compile system monitor directly (Might not work for some versions)

Get the KubeArmor version from [Release v1.4.3 - kubearmor/KubeArmor](#)

Fetch and install KubeArmor by running

```
VER="1.4.3" # set according to the latest version
```

```
curl -sfLO  
<https://github.com/kubearmor/KubeArmor/releases/download/v${VER}/kubearmor_${VER}_li  
nux-amd64.deb>  
sudo apt install ./kubearmor_${VER}_linux-amd64.deb
```

The above will generate the system monitor file at /opt/kubearmor/BPF/system\_monitor.bpf.o. Copy it to some other path.

## Onboard the node

Once you've compiled the monitor, you can specify it while onboarding the control plane/node.

Install knoxctl - the accuknox CLI by running the below command

```
curl -sfL <https://knoxctl.accuknox.com/install.sh> | sudo sh -s -- -b /usr/local/bin
```

Onboard your node/control plane by running the respective command with the below additional flags

```
sudo knoxctl onboard vm cp-node \  
... usual flags  
--skip-btf-check=true \  
  
--system-monitor-path=/tmp/system_monitor.bpf.o
```

## VM Onboarding using Access Keys

## Overview

The access key method simplifies the onboarding of multiple VMs as control plane VMs. The process mirrors that of SystemD mode and Docker Container mode. Using

the access key, users can onboard a VM directly from the CLI without needing to access the AccuKnox SaaS interface.

Users can select either SystemD or Docker Container mode for onboarding, as the same access key works for both. Moreover, the access key provides enhanced flexibility, enabling the onboarding of multiple control plane VMs with a single key

Here we will follow the SystemD mode of onboarding

## Pre-requisites

1. [Access Key](#)
2. [Resource requirements](#)
3. [Network requirements](#)
4. BTF support is enabled in the VM
5. [RabbitMQ](#) should be installed

## Onboarding

In the case of the Access key onboarding method User can directly onboard the VMs from the CLI

### NOTE

We don't need to follow AccuKnox UI for the access key method of the VM onboarding; we will be using a command to do the same from the CLI.

### Install knoxctl/accuknox-cli

```
curl -sfL https://knoxctl.accuknox.com/install.sh | sudo sh -s -- -b /usr/bin
```

### Onboarding Control Plane

The command may look something like this:

```
knoxctl onboard vm cp-node \
--version v0.8.1 \
--spire-host=spire.demo.accuknox.com \
--pps-host=pps.demo.accuknox.com \
--knox-gateway=knox-gw.demo.accuknox.com:3000 \
--vm-name="accuknox-vm" \
--access-key-url="cwpp.demo.accuknox.com" \
--access-key="access-token"
```

In the above command, You need to replace the --access-token value with the created [access key](#), and substitute --vm-name with the desired vm name. After replacing the value the command will look like this:

By default, if Docker is not found, systemd mode of installation would be used. If you want to explicitly onboard using systemd services, add the --vm-mode=systemd flag to the above command.

## Output

```

Downloading agents...
Downloading Agent - kubearmor | Image - docker.io/kubearmor/kubearmor-systemd:1.4.3_linux-amd64
kubearmor version 1.4.3_linux-amd64 downloaded successfully

Downloading Agent - vm-adapter | Image - docker.io/accuknox/vm-adapter-systemd:0.1.4_linux-amd64
vm-adapter version 0.1.4_linux-amd64 downloaded successfully

Downloading Agent - kubearmor-relay-server | Image - docker.io/accuknox/kubearmor-relay-server-systemd:0.0.4_linux-amd64
kubearmor-relay-server version 0.0.4_linux-amd64 downloaded successfully

Downloading Agent - spire-agent | Image - docker.io/accuknox/spire-agent-systemd:1.9.4_linux-amd64
spire-agent version 1.9.4_linux-amd64 downloaded successfully

Downloading Agent - accuknox-shared-informer-agent | Image - docker.io/accuknox/accuknox-shared-informer-agent-systemd:0.7.3_linux-amd64
accuknox-shared-informer-agent version 0.7.3_linux-amd64 downloaded successfully

Downloading Agent - accuknox-policy-enforcement-agent | Image - docker.io/accuknox/accuknox-policy-enforcement-agent-systemd:0.6.4_linux-amd64
accuknox-policy-enforcement-agent version 0.6.4_linux-amd64 downloaded successfully

Downloading Agent - accuknox-feeder-service | Image - docker.io/accuknox/accuknox-feeder-service-systemd:0.7.4_linux-amd64
accuknox-feeder-service version 0.7.4_linux-amd64 downloaded successfully

Downloading Agent - accuknox-sumengine | Image - docker.io/accuknox/accuknox-sumengine-systemd:0.2.4_linux-amd64
accuknox-sumengine version 0.2.4_linux-amd64 downloaded successfully

Downloading Agent - accuknox-discover | Image - docker.io/accuknox/accuknox-discover-systemd:0.2.4_linux-amd64
accuknox-discover version 0.2.4_linux-amd64 downloaded successfully

Downloading Agent - accuknox-hardening-agent | Image - docker.io/accuknox/accuknox-hardening-agent-systemd:0.2.4_linux-amd64
accuknox-hardening-agent version 0.2.4_linux-amd64 downloaded successfully

All agents downloaded successfully.

Configuring services...

Enabling services...
Started kubearmor.service
Started kubearmor-vm-adapter.service
Started kubearmor-relay-server.service
Started spire-agent.service
Started accuknox-shared-informer-agent.service
Started accuknox-policy-enforcement-agent.service
Started accuknox-feeder-service.service
Started accuknox-sumengine.service
Started accuknox-discover.service
Started accuknox-hardening-agent.service

Cleaning up downloaded assets...
VM successfully onboarded!

Now run the below command to onboard any worker nodes.
Please assign appropriate IP address to --cp-node-addr to make sure
that worker nodes can connect to this node
knoxctl onboard vm node --vm-mode="systemd" --version=v0.8.1 --cp-node-addr=<address-of-this-node>

```

The above command will emit the command to onboard worker nodes. You may also use the `--cp-node-addr` flag to specify the address that other nodes will use to connect with your cluster.

## NOTE

The user needs to repeat the CLI onboarding command to onboard multiple control plane VMs using the access key

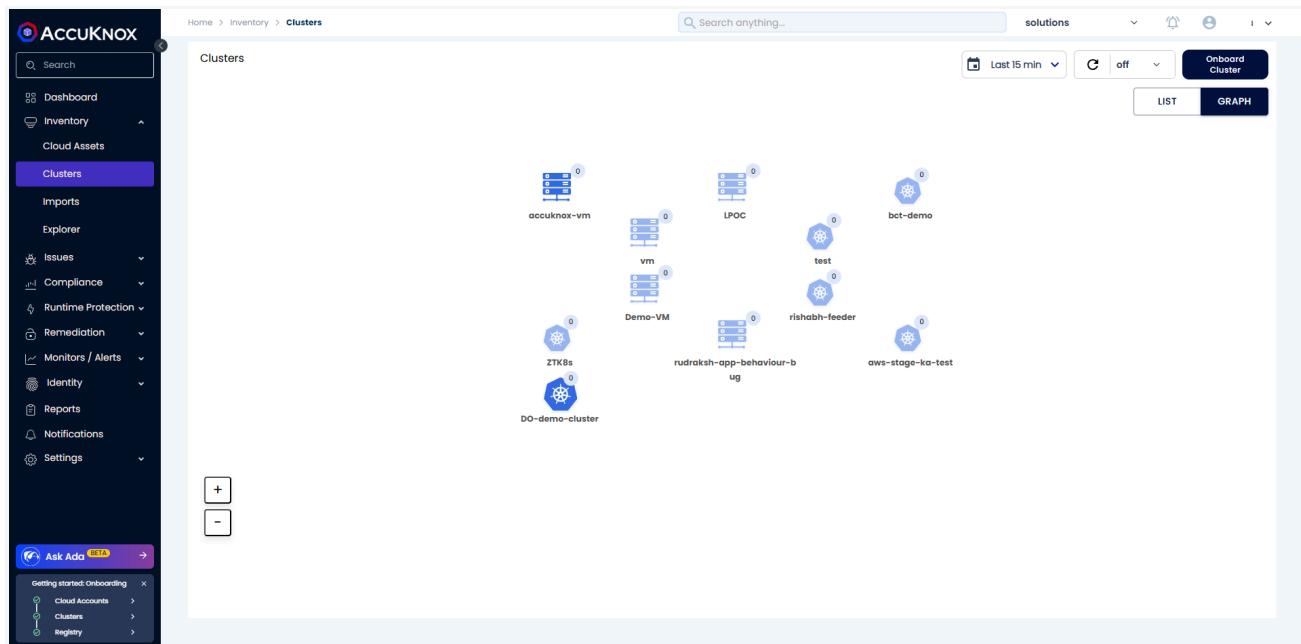
# Onboarding Worker Nodes

The second command will be for onboarding worker nodes. It may look something like this:

```
knoxctl onboard vm node --vm-mode="systemd" --version=v0.8.1  
--cp-node-addr=<control-plane-addr>
```

Example:

```
$ knoxctl onboard vm node --vm-mode="systemd" --version=v0.8.1  
--cp-node-addr=192.168.56.106  
Pulling kubearmor-init ... done  
Pulling kubearmor ... done  
Pulling kubearmor-vm-adapter ... done  
Creating network "accuknox-config_accuknox-net" with the default driver  
Creating kubearmor-init ... done  
Creating kubearmor ... done  
Creating kubearmor-vm-adapter ... done  
onboard-vm-node.go:41: VM successfully joined with control-plane!
```



# Troubleshooting

If you encounter any issues while onboarding, use the commands below to debug:

```
sudo journalctl -xeu <service-name>.service
```

Replace <service-name> with one of the following:

- kubearmor: Logs show policy enforcement and monitor Kubernetes workloads; useful for debugging misconfigurations or runtime issues.
- kubearmor-relay-server: Bridges KubeArmor clients with external log systems; logs debug communication or relay errors.
- kubearmor-vm-adapter: Tracks policy enforcement in VMs; logs diagnose policy application on non-Kubernetes workloads.
- accuknox-policy-enforcement-agent: Enforces security policies; logs troubleshoot policy errors or conflicts.
- accuknox-shared-informer-agent: Shares Kubernetes resource data; logs debug metadata collection issues.
- accuknox-sumengine: Processes telemetry data; logs resolve performance or data processing errors.
- accuknox-discover-agent: Discovers potential policies; logs analyze policy suggestions.
- spire-agent: Manages workload identities; logs debug identity issuance and attestation issues.
- accuknox-hardening-agent: Automates system hardening; logs troubleshoot configuration and hardening conflicts.

# Deboarding

Deboard the cluster from SaaS first.

To deboard the worker-vm/Node:

```
knoxctl deboard vm node
```

To deboard the Control-Plane VM:

```
knoxctl deboard vm cp-node
```

Sample Output:

```
$ knoxctl deboard vm cp-node
[+] Running 10/10
✓ Container shared-informer-agent    Removed          0.6s
✓ Container feeder-service           Removed          0.6s
✓ Container policy-enforcement-agent Removed          0.8s
✓ Container wait-for-it             Removed          0.0s
✓ Container kubearmor-vm-adapter   Removed          5.6s
✓ Container kubearmor-relay-server  Removed          1.5s
✓ Container spire-agent             Removed          0.5s
```

✓ Container kubearmor	Removed	10.4s
✓ Container kubearmor-init	Removed	0.0s
✓ Network accuknox-config_accuknox-net	Removed	0.3s
Please remove any remaining resources at /home/user/.accuknox-config		
Control plane node deboarded successfully.		

After that cleanup the ~/accuknox-config directory

```
sudo rm -rf ~/accuknox-config
```

## In-Cluster Image Scanning with Helm

AccuKnox offers an in-cluster container image scanning solution designed to periodically inspect container images deployed within your Kubernetes (K8s) environment. This automated scanning process detects known vulnerabilities, promoting compliance and enhancing your cluster's overall security. All scan results, including detailed vulnerability insights, are automatically sent to the AccuKnox Control Plane, where they can be viewed and managed through an intuitive user interface.

## Installation Guide

Follow these steps to deploy the in-cluster image scanner using Helm:

### 1. Create a Label

In the AccuKnox Control Plane, create a unique **Label**. This will be associated with the container image scan reports.

### 2. Generate a Token

From the AccuKnox Control Plane:

- Generate an **Artifact Token**

- Note down both the **Token** and your **Tenant ID**

### 3. Schedule and Deploy the Scanner via Helm

Use the following Helm command to install the scanner in your Kubernetes cluster:

```
helm install kubeshield oci://public.ecr.aws/k9v9d5v2/kubeshield-chart -n agents
--create-namespace \
--set scan.tenantId="" \
--set scan.authToken="" \
--set scan.url="" \
--set scan.label="" \
--set scan.cronTab="30 9 * * *" \
--version "v0.1.2"
```

Replace the parameters (,, , and ``) with the appropriate values.

#### Sample Output

```
Pulled: public.ecr.aws/k9v9d5v2/kubeshield-chart:v0.1.1
Digest: sha256:a4c1a8948db7a24d8990b71b53184f564960b2b39dbd6cba1cd6104c12add75
NAME: kubeshield
LAST DEPLOYED: Mon May  5 10:08:24 2025
NAMESPACE: agents
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

#### Parameters:

Variable	Sample Value	Description

---

tenantId	11	AccuKnox Tenant ID
authToken	eyJhbGc...	AccuKnox Token
url	cspm.accuknox.co m	AccuKnox CSPM API Endpoint
label	kubeshield	AccuKnox Label
cronTab	30 9 ***	Schedule in Cron

---

**Note:** Deploy the Scanner via Helm (One Time) If you don't want to schedule and just want to trigger scan for one time, remove this flag `--set scan.cronTab`

## Post-Installation

Once the scanner is deployed and completes a scan cycle, results will be visible in the **Findings** or **Registry Scan** sections within the AccuKnox Control Plane.

- Navigate to **Issues -> Findings**
  - Switch to **Findings** tab
  - Select **Container Image Findings** & do **Group by** based on **Label Name**
  - You should be able to see the data for the **Label** used in above command
-



## Scan Status from Cluster

- 💡 Check if `kubeshield-controller-manager` is running fine or not

```
kubectl get po -n kubeshield
NAME                                READY   STATUS    RESTARTS   AGE
kubeshield-controller-manager-5dd5cbc6d4-8xg8k  1/1    Running   0          22s
```

STATUS should be **Running**

# Dockerhub Registry Onboarding

**Docker Hub is a cloud-based repository for storing, sharing, and managing Docker container images. It's like a library for container images, where you can find and download pre-built images or upload your own.**

## Prerequisites

### Personal Account

- **Requires:**
  - Username
  - Password
- **Explanation:** A personal account is used by individual users who own or manage their own Docker Hub repositories. These credentials authenticate access to the user's personal space in Docker Hub.

Authentication Type:  Personal  Organisation  Docker Trusted Registry

Username \*

## Organization Account

- **Requires:**
  - Organization Name
  - Username
  - Password
- **Explanation:** An organization account is suitable for teams and enterprises managing shared Docker Hub repositories. It allows multiple users to collaborate under a unified organization while maintaining individual user roles and permissions.

**Note:** Users must have pull permissions to access images stored in the enterprise repositories.

Authentication Type:  Personal  Organisation  Docker Trusted Registry

Organisation Name \*

Username \*

Password \*

# Steps to Add a Registry

1. Navigate to the Registry Scan Section

- Go to Issues > Registry Scan.

Repositories	C	H	M	L	N	O	Registry Name
> rajvanshi/alpine	8	38	23	1	0	61	rajvanshi
> rajvanshi/dogfooding	0	2	28	4	0	60	rajvanshi
> rajvanshi/dwva	182	432	572	100	41	61	rajvanshi
> rajvanshi/juice-shop	15	42	65	17	1	61	rajvanshi
> rajvanshi/storm	54	134	198	141	0	60	rajvanshi
> jayjersan/accuknox-job	0	0	2	6	0	60	Jest
> jayjersan/agents-operator	4	18	47	34	0	60	Jest
> jayjersan/discovery-engine-discover	2	8	48	62	0	60	Jest
> jayjersan/discovery-engine-hardening	3	9	56	64	0	60	Jest
> jayjersan/discovery-engine-sumengine	0	18	56	62	0	60	Jest

## 2. Add a New Registry

- Click on Add Registry.

Repositories	C	H	M	L	N	O	Registry Name
> rajvanshi/alpine	8	38	23	1	0	61	rajvanshi
> rajvanshi/dogfooding	0	2	28	4	0	60	rajvanshi
> rajvanshi/dwva	182	432	572	100	41	61	rajvanshi
> rajvanshi/juice-shop	15	42	65	17	1	61	rajvanshi
> rajvanshi/storm	54	134	198	141	0	60	rajvanshi
> jayjersan/accuknox-job	0	0	2	6	0	60	Jest
> jayjersan/agents-operator	4	18	47	34	0	60	Jest
> jayjersan/discovery-engine-discover	2	8	48	62	0	60	Jest
> jayjersan/discovery-engine-hardening	3	9	56	64	0	60	Jest
> jayjersan/discovery-engine-sumengine	0	18	56	62	0	60	Jest

## 3. Provide Registry Details

- **Registry Name:** Enter a name for your registry.
- **Label:** Add a label to associate findings to a particular label.
- **Description:** Provide additional information about the registry.
- **Registry Type:** Select Docker Hub from the dropdown menu.

Registry Name \* TestRegistryDhub

Label \* dhubRegistry

Registry Type \* Docker Hub

## 4. Authentication Type

- **Choose an appropriate authentication type based on your Docker Hub configuration:**
- **Personal: Requires your Docker Hub Username and Password.**

Authentication Type:  Personal  Organisation  Docker Trusted Registry

Username \* Enter Username

Password \* Enter Password

- **Organization: Requires your Organization Name, Username, and Password.**

Authentication Type:  Personal  Organisation  Docker Trusted Registry

Organisation Name \* Enter Organisation or Namespace

Username \* Enter Username

Password \* Enter Password

## 5. Configure Advanced Settings

### Image Updated Within Last

Choose one of the following options:

- **X Days: Scans only images updated within the last X days.**
- **All: Scans all images, regardless of the update time.**

### Image Pulled Within Last

Choose one of the following options:

- **X Days: Scans only images pulled within the last X days.**
- **All: Scans all images, regardless of the pull time.**

#### Advance Settings

Image Updated within last:

Days  Days  All

Image Pulled within last:

Days  Days  All

#### Name/Tag Pattern:

**Specify patterns to include or exclude images for scanning. Use the - symbol to explicitly exclude patterns.**

**By default, images are excluded unless explicitly included through patterns.**

**To exclude specific images, use the - symbol. For example: - To exclude cwpp/ubuntu:v1, use the pattern -\*:v1. - To include cwpp/ubuntu:latest, specify a pattern like \*:latest.**

**Note: Only images matching the pattern will be scanned. For instance, using \*:latest ensures only images with the latest tags are scanned.**

The screenshot shows a search bar with the placeholder "Type a value and press [Enter]". Inside the bar, there are two entries: "-\*:v1" and "cwpp/ubuntu:latest", separated by a space. Both entries have small 'X' icons to their right.

#### Schedule and Certificate

**Set the scan schedule using a CRON expression. For example: - CRON Expression: 18 minute 07 hour \* day (month) \* month \* day (week).**

Schedule:

18	07	*	*	*
minute	hour	day (month)	month	day (week)

(Server TimeZone: UTC) [?](#)      (User TimeZone: IST) [?](#)

At 07:18 AM      At 12:48 PM

[next](#) scan at: 2024-12-17 07:18:00 AM      [next](#) scan at: 2024-12-17 12:48:00 PM

**Toggle Trigger Scan on Save to directly initiate the scan for the first time without waiting for the scheduled time.**

# Viewing Registry Scan Details

**Once the configuration is complete, your registry is ready for scanning. Scans will occur based on the defined schedule and criteria. Ensure all advanced settings align with your organizational requirements for optimal results.**

To view the scan results:

1. Navigate to Issues > Registry Scan.
2. Find your repository to view the findings.

Image Name	Critical	High	Medium	Low	Negligible	Registry Name
rajvanishi/juice-shop:latest	15	42	65	17	1	rajvanishi

3. Alternatively, select Scan Queue to check the scan status.

Pending	In Progress	Finished	Failed	Unsupported
0	0	26	0	0

# JFrog Container Registry Onboarding

JFrog Container Registry is a secure, universal repository manager specifically optimized for storing and managing container images. Widely adopted by DevOps and software teams, it supports Docker and Helm images, offering seamless integration with CI/CD pipelines to enhance workflows and ensure image security and traceability.

JFrog Artifactory offers two primary deployment options:

1. **Cloud-Based:** Managed by JFrog, offering scalability and minimal maintenance for teams preferring a ready-to-use solution.
2. **Self-Hosted:** On-premise for strict security needs, giving organizations control over configurations, with support for deployment in isolated networks.

## AccuKnox Support for JFrog Container Registry Scanning

AccuKnox provides robust security scanning for container images stored in the JFrog Container Registry, regardless of deployment type. Supporting both cloud-based and self-hosted JFrog instances.

- **Cloud-Based JFrog Scanning:** For the JFrog Container Registry deployed in the cloud, AccuKnox connects seamlessly to scan images and detect vulnerabilities in real time.
- **Self-Hosted JFrog Scanning:** AccuKnox also supports self-hosted JFrog Container Registry deployments, providing vulnerability scanning for images in private, on-premise environments.
  - **Isolated Network Support:** AccuKnox can connect to self-hosted JFrog instances in isolated or air-gapped networks. This enables secure scanning in environments with strict compliance or network restrictions, ensuring continuous monitoring without compromising security.

The following steps outline how to onboard your JFrog Container Registry into the AccuKnox platform for ongoing security scanning, giving you real-time insights into vulnerabilities and risks within your container images.

## Scanning an Isolated Registry

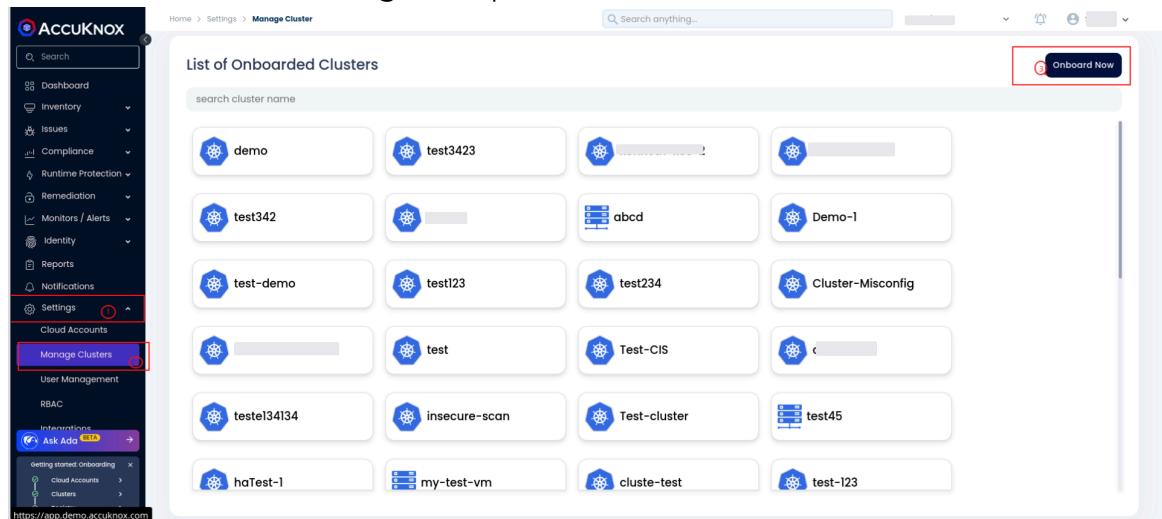
**Important:** If you're using a non-isolated JFrog Container Registry (cloud-based or non-isolated self-hosted), you can skip this section. This part applies **only** to **isolated JFrog instances**.

To get started with scanning a JFrog isolated container registry, ensure the following prerequisites are met:

1. Set up an **isolated JFrog container registry**.
2. Ensure you have access to a Kubernetes cluster where the **AccuKnox agents** can be onboarded.

Once your registry is set up, the next step is to onboard the AccuKnox agents to your Kubernetes cluster.

1. Navigate to **Settings > Manage Cluster** in the AccuKnox platform.
2. Click **Onboard Now** to begin the process.



The screenshot shows the AccuKnox platform interface. On the left, there's a sidebar with various navigation options: Dashboard, Inventory, Issues, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports, Notifications, Settings (which is currently selected and highlighted with a red circle), Cloud Accounts, and Manage Clusters (also highlighted with a red circle). The main content area is titled 'List of Onboarded Clusters' and shows a grid of 16 cluster entries. Each entry includes a small icon, the cluster name, and some status indicators. In the top right corner of the main area, there's a prominent red button labeled 'Onboard Now'.

3. Provide an appropriate name for your cluster in the form that appears. During the agent installation process, ensure that the **Scanner for**

## Isolated Registry Scan option is enabled.

The screenshot shows the AccuKnox web interface under the 'Agents' tab. On the left sidebar, there's a 'Manage Clusters' section. In the main area, there's a 'Cluster Onboarding' step 1: 'Select Cluster Type' (set to 'Kubernetes') and 'Enter Cluster Name' (set to 'spoc'). Step 2: 'Agents Installation' includes sections for 'Runtime Visibility & Protection' (KIEIM), 'Kubernetes CIS Benchmark', and 'Cluster Misconfiguration'. Below these is a 'Install AccuKnox Agents' section containing a command-line snippet:

```
1 helm upgrade --install agents oci://registry-1.docker.io/accuknox/accuknox-agents \
2   --version "v0.8.0" \
3   --set joinToken="" \
4   --set spireHost="spire.demo.accuknox.com" \
5   --set ppsHost="pps.demo.accuknox.com" \
6   --set knoxGateway="knox-gw.demo.accuknox.com:3000" \
7   --set install.localRegistryAgent=true \
8   -n agents --create-namespace
```

A red box highlights the checkbox labeled 'Enable Scanner for Isolated Registry Scan' at the bottom of the command snippet.

- Run the following Helm command to install the AccuKnox agents

```
helm upgrade --install agents oci://registry-1.docker.io/accuknox/accuknox-agents \
--version "v0.8.0" \
--set joinToken="" \
--set spireHost="spire.demo.accuknox.com" \
--set ppsHost="pps.demo.accuknox.com" \
--set knoxGateway="knox-gw.demo.accuknox.com:3000" \
--set install.localRegistryAgent=true \
-n agents --create-namespace
```

- Verify the installation of the agents by running the following command:

```
kubectl get pods -n agents
```

```
→ ~ kubectl get pods -n agents
NAME                               READY   STATUS    RESTARTS   AGE
agents-operator-7645bccd5c-t5tx2   1/1     Running   0          63s
feeder-service-5f7b45884c-sbppz   1/1     Running   0          43s
local-registry-agent-7cb7484f5b-nkd8w 1/1     Running   0          43s
policy-enforcement-agent-544d59cf8-mrpkf 1/1     Running   0          42s
rabbitmq-755c547b88-tfpkp        1/1     Running   0          63s
shared-informer-agent-8589b8f6cf-82nwc 1/1     Running   0          42s
→ ~
```

Once the agents are installed, navigate to the Cluster View in AccuKnox to ensure that your onboarded cluster is live and ready for scanning. This completes the onboarding process for scanning an isolated container registry in AccuKnox. The next step is to configure the registry scanning, as outlined in the previous sections.

## Configuring the **JFrog** Registry

For this example, we'll proceed with **JFrog Self-hosted**.

Next, configure the self-hosted registry to begin scanning. Choose between **JFrog Cloud or Self-hosted**.

1. Go to **Settings -> Integration -> Registry**.
2. Click on the **Add Registry** button
3. Fill out the required fields such as:
  - a. Name
  - b. Description
  - c. Registry Type
  - d. URL
  - e. Credentials
  - f. Cron Expression (for scheduled scans)
4. If your JFrog Container Registry is in an isolated mode, ensure that the **Isolated Registry** flag is enabled in the onboarding form
5. Test the connection. If the configuration is correct, you will receive a successful response.

The screenshot shows the 'Add Registry' page in the AccuKnox interface. The left sidebar has 'Integrations' selected. The main form is titled 'Add' and includes the following fields:

- Registry Name:** jfrog-spoc
- Label:** SPOC
- Registry Type:** JFrog Artifactory
- Description:** Registry Scanning with JFrog at AccuKnox
- Authentication Type:** Self Hosted (radio button selected)
- Isolated Registry:** (checkbox is unchecked)
- Registry URL:** (redacted)
- Self Signed Certificate:** (checkbox is unchecked)
- Username:** (redacted)
- Password:** (redacted)
- Advance Settings:** Name / Tag Pattern: \*latest
- Schedule:** (set to 19 hours, 06 minutes, daily (weekly))
- (Server TimeZone: UTC)**: At 06:19 AM
- (User TimeZone: IST)**: At 11:49 AM
- Trigger scan on save:** (checkbox checked)
- test connection** button
- Cancel** and **Save** buttons

6. Once the connection is verified, save the form and create the registry. After the registry is configured and connected, it will appear as **Active** in the registry list.

The screenshot shows the AccuKnox interface with the 'Registry' tab selected. A success message 'Registry Added Successfully' is displayed at the top right. The main content area is a table listing registered sources:

Name	Type	URL	Status	Last Scan status	Last Scan On	Label	Actions
jfrog-s poc	JFrog Artifactory	https://jfrog.gcp.accuknox.com	ACTIVE	NOT SCANNED	Date Not Available	S POC	⋮
UbuntuScan	Docker Hub		ACTIVE	COMPLETED	11/12/2024 15:12 PM	UbuntuScan	⋮
	Docker Hub		ACTIVE	COMPLETED	11/12/2024 14:06 PM	thomasin33	⋮
Docker-Personal-2	Docker Hub		ACTIVE	COMPLETED	11/12/2024 17:47 PM	label1	⋮
Docker-personal	Docker Hub		ACTIVE	COMPLETED	11/12/2024 15:29 PM	label1	⋮
	Docker Hub		ACTIVE	COMPLETED	11/12/2024 14:33 PM	label1	⋮
	Docker Hub		ACTIVE	COMPLETED	11/13/2024 00:00 AM	label1	⋮
test-dvwa	Docker Hub		ACTIVE	COMPLETED	11/12/2024 16:05 PM	label1	⋮
docker-test	Docker Hub		ACTIVE	COMPLETED	11/13/2024 09:22 AM	label1	⋮
27gor	Google Artifact Registry (GAR)	us-central1	ACTIVE	COMPLETED	11/12/2024 14:23 PM	label1	⋮

AccuKnox will begin scanning at the scheduled time specified during the configuration or If you've enabled the **Trigger scan on the save** option, the first scan will start immediately. Once the scan completes, navigate to the registry page to view the results.

## Viewing Scan Details

After the scan is completed, you can explore detailed information about the registry:

1. Go to **Issues** -> **Findings** -> **Registry Scan**.
2. Filter the results to view the onboarded registry.
3. Click on an image to see a detailed view of the metadata, vulnerabilities, and other scan details.

In the **JFrog Self-hosted Registry** that we onboarded to AccuKnox during this presentation, there is a specific package, **accuknox/nginx**. Below, you can see the associated vulnerabilities for this image, as highlighted in the following screenshots.

To get more detailed information about the vulnerabilities associated with the image, simply click on the container image in the AccuKnox dashboard. This will allow you to view the metadata, including any embedded secrets and a comprehensive list of the vulnerabilities identified in the image. You will also be able to explore the severity of these vulnerabilities, CVSS scores, and recommended remediation actions.

Integrating JFrog Container Registry with AccuKnox ensures continuous security scanning for container images, whether cloud-based or self-hosted. For isolated networks, AccuKnox provides secure, compliance-friendly scanning, helping you detect and address vulnerabilities efficiently.

# CWPP Report Generation

## Understand the Regex to Select the Cluster Name and Namespace

The CWPP report generation utilizes regular expressions (regex) to specify and filter cluster names and namespaces. The syntax for regex follows a particular pattern to ensure accurate selection.

# Regex

**Regex Syntax Format:** Cluster Name Selection / Namespace Selection

## Rules for Regular Expression

### Excluding

- To exclude a specific cluster or namespace, prefix it with a hyphen (-).

### NOTE

To exclude any cluster or namespace, it must be included in the selection first.

### Select all

- Use an asterisk (\*) to select all clusters or namespaces.

### Delimiter

- A forward slash (/) is used to delimit the cluster name selection from the namespace selection.

## Examples

- cluster1/ns1: Include only namespace ns1 from cluster cluster1.
- cluster1/\*: Include all namespaces from cluster cluster1.
- cluster1/ns\*: Include namespaces starting with ns from cluster cluster1.
- -cluster1/ns3: Exclude namespace ns3 from cluster cluster1.
- \*/ns1: Include namespace ns1 from all clusters.
- \*/\*: Include all namespaces from all clusters.

# Reports Configuration

Reports can be configured in two ways: On Demand and Scheduled.

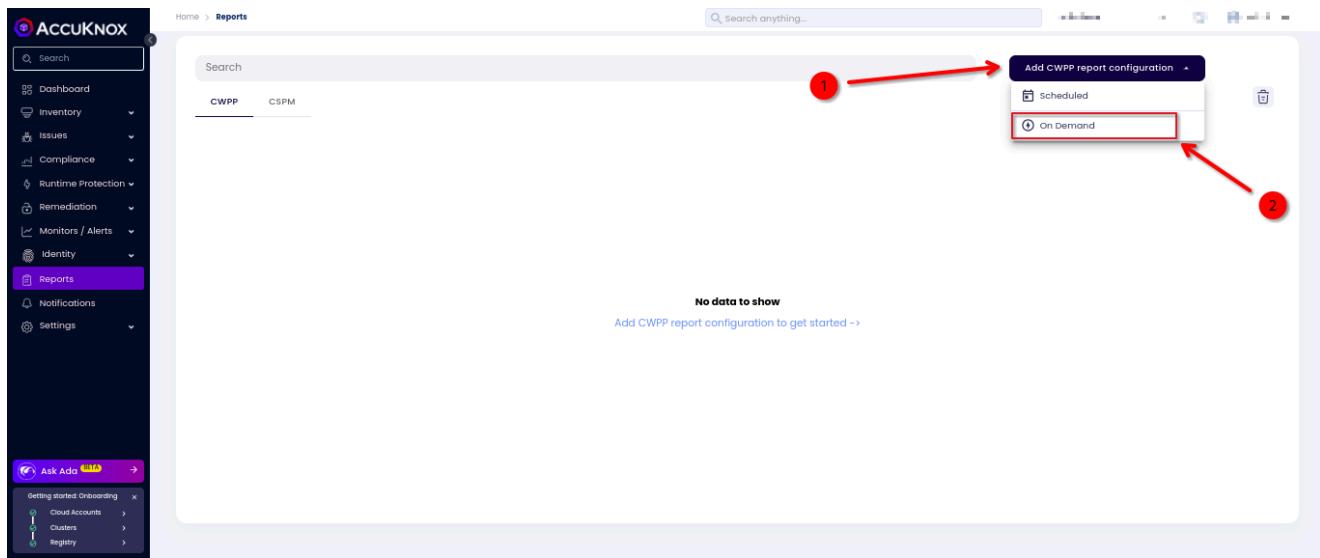
## 1. On Demand Report Configuration

In On Demand Report, you can generate the report for the clusters shortly after the configuration is completed.

To generate On Demand reports:

### **Step 1:** Add CWPP Report Configuration

- Go to the Reports section in AccuKnox SaaS.
- Choose "On Demand" from the drop-down menu.



**Step 2:** In the Configuration user needs to provide the details about Name, Description and Cluster and NameSpace.

## NOTE

The cluster field drop-down will show all the clusters that are active during the report generation.

The screenshot shows the AccuKnox On-Demand Configuration interface. The left sidebar has a dark theme with various navigation options like Dashboard, Inventory, Issues, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports (which is selected and highlighted in purple), Notifications, and Settings. A 'Getting started: Onboarding' section lists Cloud Accounts, Clusters, and Registry. The main content area is titled 'On Demand Configuration' and shows the 'Details' tab for CWPP. It includes fields for 'Name\*' (cwpp-on-demand), 'Description\*' (cluster report), a 'CWPP Dashboard' section with a dropdown for 'Cluster / Namespace\*' containing 'qa-/\*', and a 'Duration\*' dropdown set to '7 Days'. At the bottom right is a large blue 'Save & Generate Report' button.

By clicking Save and Generate Report it will generate the report in the PDF format as per the selected duration.

The image shows two screenshots of the AccuKnox interface. The top screenshot, labeled '1', shows the 'Reports' section with a dropdown menu for 'Add CWPP report configuration'. The bottom screenshot, labeled '2', shows the generated 'AccuKnox Runetime Protection(CWPP)Report' for a cluster named 'qa-030524'. The report details 25 total workloads, with 6 protected and 19 unProtected. It also lists the 'Top 5 Namespaces with unique Violations' and states 'There is no violations for the given time frame'.

## 2. Scheduled Report Configuration

To get the report of the clusters automatically as per the frequency that chosen .i.e by weekly or by monthly or daily this is the go to way.

**Step 1:** To Add CWPP report configuration as Scheduled and choose the Scheduled option from the drop down.

The screenshot shows the 'Reports' section of the AccuKnox UI. A red arrow labeled '1' points to the 'Scheduled' option in the dropdown menu for 'Add CWPP report configuration'. Another red arrow labeled '2' points to the 'On Demand' option in the same menu. The main panel displays a message: 'No data to show' and 'Add CWPP report configuration to get started ->'.

**Step 2:** In the Configuration user needs to provide the details about their Name, Email, Selecting the Cluster, Namespace in the regex format and Frequency of the report then click the Generate Report.

**Scheduled Configuration**

Name \*: Automatic Daily Report

Email \*:

Description \*: For Clusters

**CWPP Dashboard**  
Select Clusters/Namespace in Regex format

Cluster / Namespace \*:  
Hetzner/container\_namespace x k3saffan/nginx x  
psa/default x

Include Future Clusters/Namespace:  All future onboarded clusters/namespaces will be included/excluded with this matching regex

**Frequency & Duration**  
Select the frequency for report generation

Frequency \*: Daily (Reports will be generated at 09:00 AM UTC everyday)

**Summary of Report**

- List of workload with no protection  
This will provide list of workloads where there is no policy applied
- List of workload with policy violation  
This will provide list of workloads where there policy is applied

**Generate Report**

<input type="checkbox"/>	Name	Email	Frequency	Last reports	View
<input type="checkbox"/>	cwppautomaticdailiyreport	[REDACTED]	Daily	--	<a href="#">Generate Report</a>
<input type="checkbox"/>	cwpp-on-demand	[REDACTED]	On Demand	2024-05-06	<a href="#">Generate Report</a>
<input type="checkbox"/>	test-01	[REDACTED]	On Demand	2024-05-06	<a href="#">Generate Report</a>
<input type="checkbox"/>	RPG401-402-403	[REDACTED]	Daily	2024-05-06	<a href="#">View Report</a>
<input type="checkbox"/>	test352235	[REDACTED]	On Demand	2024-05-03	<a href="#">Generate Report</a>
<input type="checkbox"/>	test211	[REDACTED]	On Demand	2024-04-29	<a href="#">Generate Report</a>
<input type="checkbox"/>	after-deploy-default	[REDACTED]	On Demand	2024-04-29	<a href="#">Generate Report</a>
<input type="checkbox"/>	test1121	[REDACTED]	On Demand	2024-04-29	<a href="#">Generate Report</a>

**Add CWPP report configuration**

**Step 3:** After finishing the configuration the report would be scheduled to be sent to you in the email. Users can reconfigure the past configurations by clicking on them to edit the configuration.



## CWPP Reports ➔



AccuKnox <noreply@accuknox.com>



Hey,

Your AccuKnox Runtime Protection (CWPP) Report [REDACTED] is ready to access, Please click the below to access or download the attachment

Access Link : <https://app.stage.accuknox.com/reports>

Thank You,  
AccuKnox Team.

### NOTE

The report will be sent to the Email-ID daily at 09.00AM UTC.

## How to Configure Custom Reports

AccuKnox's latest feature update provides new custom reporting feature capabilities that can help users get the reports customized as per their requirements.

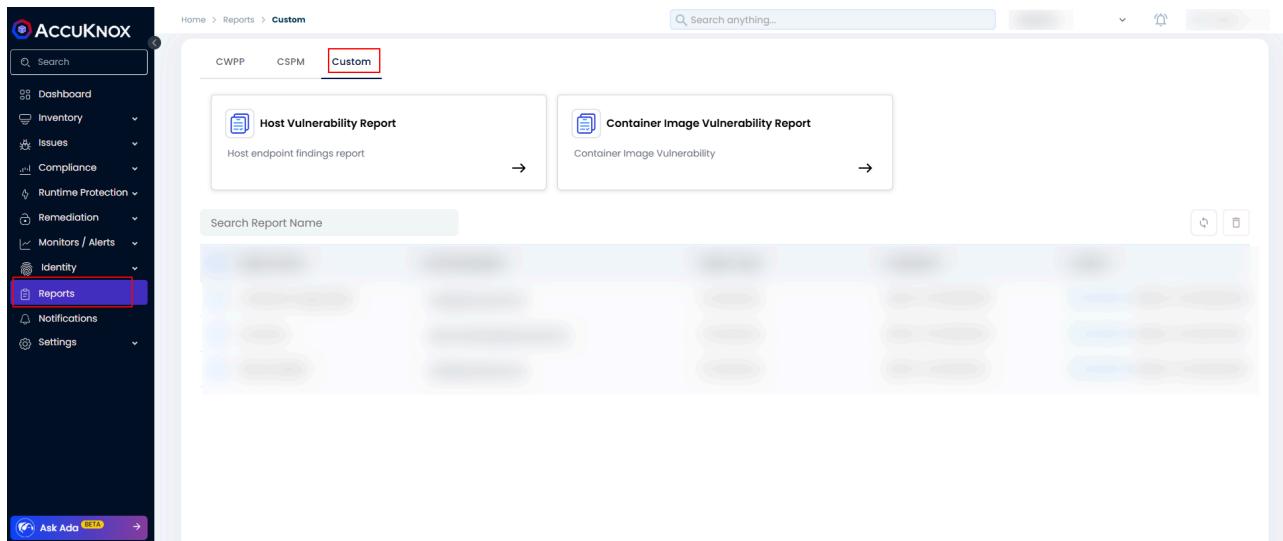
### NOTE

For this feature to be enabled the customers need to inform the Support team ([support@accuknox.com](mailto:support@accuknox.com)) regarding their requirements for custom reporting. Then the AccuKnox Support team can configure the report template from the backend. After which the users can generate an on-demand report or configure a scheduled report.

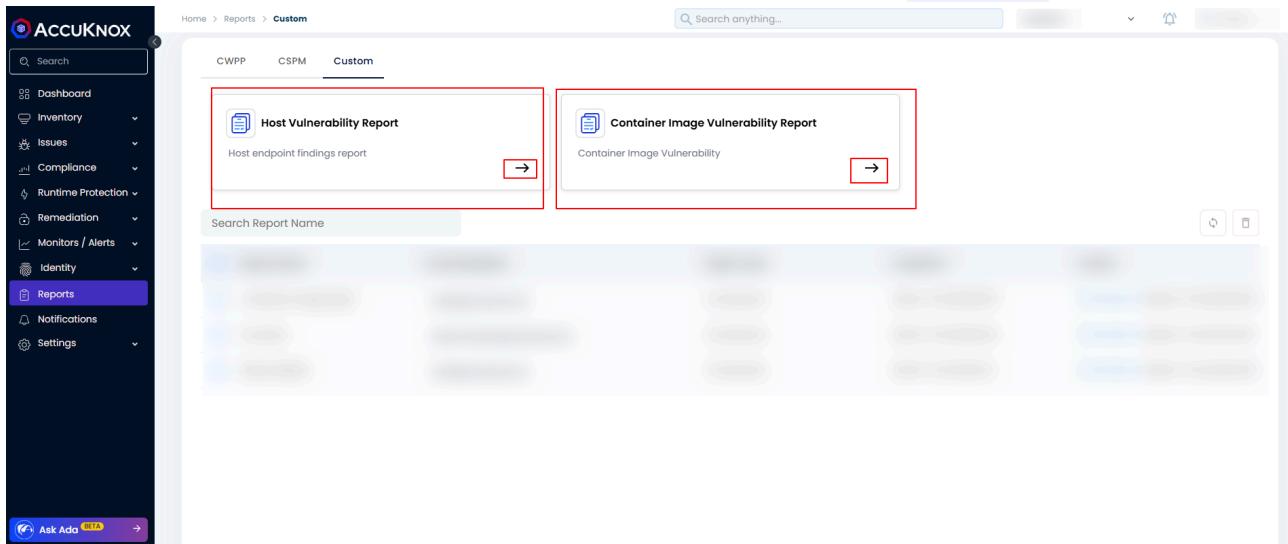
To generate an on-demand or scheduled report, users must follow the steps below.

# On-demand custom Report generation

**Step 1:** Users will need to navigate to the Reports->Custom Reports Section.



**Step 2:** Now the users will need to select any one report which they want to configure from the customized reports that are shown in the UI.



**Step 3:** Users can configure the report as a scheduled report or generate it as an on-demand one. Users can select any one option and fill out the necessary details. Like if it is an on-demand report the users will need to fill in the following fields

Like the report name, an email address where the report needs to be sent, and the duration for which the report needs to be generated from the drop-down list options shown in the UI. After filling out these options the save button will be enabled and users can save it.

**Host Vulnerability Report** ⓘ Change

Configure schedule &amp; email recipients for this report

**OnDemand** Schedule

Name \*

Test

Email \*

Enter email and press enter

ⓘ Type values and press [Enter]

Description

Description...

Date Range \*

Last 2 days

^

Last 2 days

Last 7 days

Last 15 days

Last 30 days

**Save****Cancel**

**Step 4:** Once the on-demand report is saved the users can see the report in the UI with the progress state mentioned

Home > Reports > Custom

Search anything...

CWPP CSPM Custom

Host Vulnerability Report  
Container Image Vulnerability Report

Report Name Email Recipients Report Type Created At Actions

Report Name	Email Recipients	Report Type	Created At	Actions
Test	[REDACTED]	On Demand	2024-11-18 18:34:23 IST	<span>PROGRESS</span>
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

**Step 5:** After the report generation is completed you can see the Generate option in the UI as well as the report will be mailed to the email address. If the user wants to see the report in the UI they can click on the Generate report.

Home > Reports > Custom

Search anything...

CWPP CSPM Custom

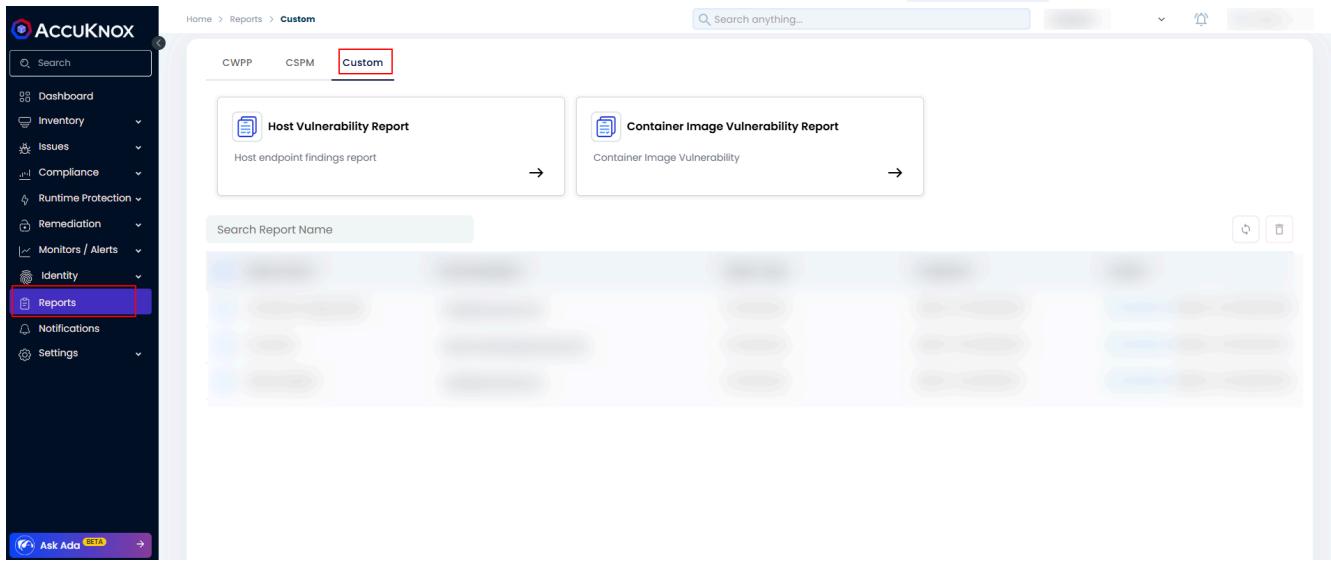
Host Vulnerability Report  
Container Image Vulnerability Report

Report Name Email Recipients Report Type Created At Actions

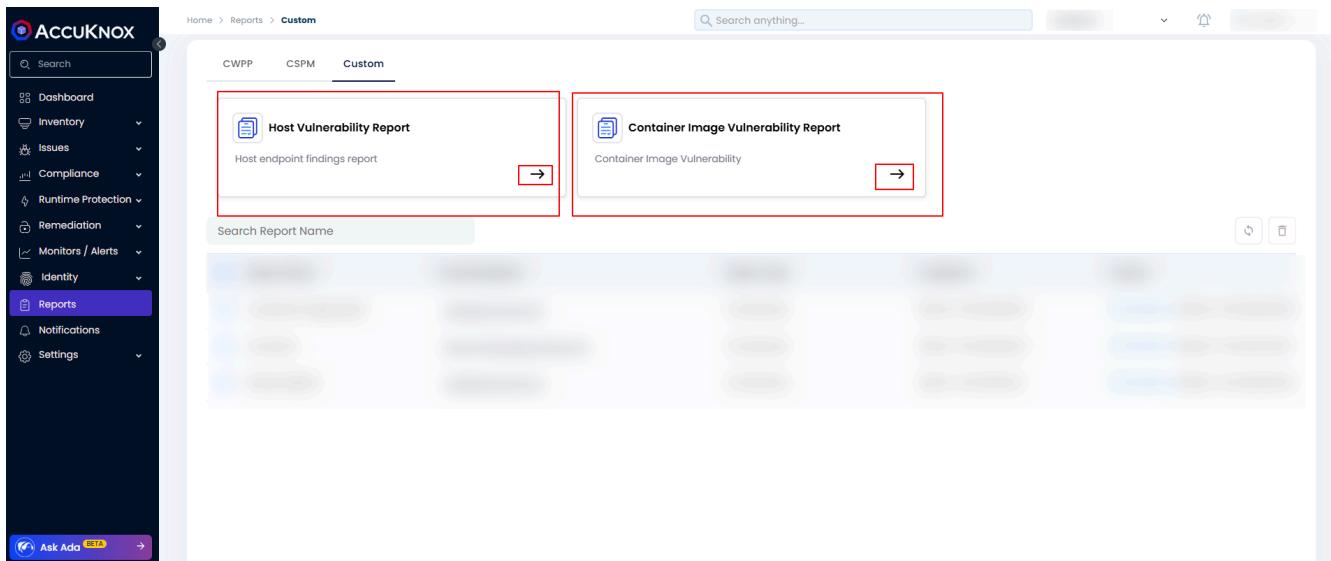
Report Name	Email Recipients	Report Type	Created At	Actions
scheduled_report	[REDACTED]	Schedule	2024-11-18 19:02:32 IST	<span>VIEW (2024-11-18 19:05:28 IST)</span>
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

# Scheduling Custom Report

**Step 1:** Users will need to navigate to the Reports->Custom Reports Section.



**Step 2:** Now the users will need to select any one report which they want to configure from the customized reports that are shown in the UI.



**Step 3:** Now the users will have the option to configure the report as a scheduled report or generate it as an on-demand one. Users can select any one option and fill out the necessary details. If the users want to schedule a custom report then they will have to fill out the following details like name, duration, and scheduling frequency. AccuKnox provides 3 scheduling frequency options.

1. Daily Report: users can select the frequency as daily to receive the report every day at the configured time.

Home > Reports > Custom > [Create](#) Search anything...

**Container Image Vulnerability Report** ⓘ [Change](#)

Configure schedule & email recipients for this report

[OnDemand](#) [Schedule](#)

Name \*  
scheduled\_report

Email \*  
[REDACTED]

Description  
test

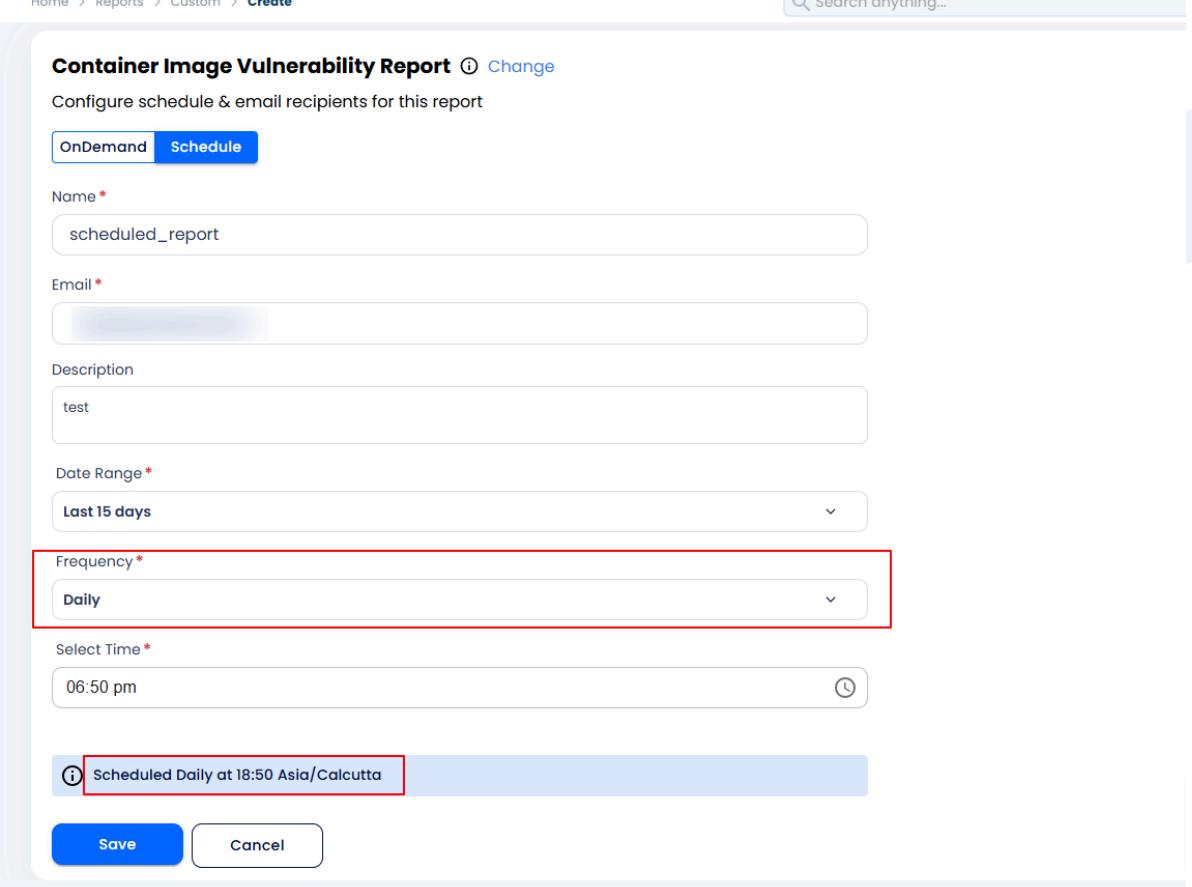
Date Range \*  
Last 15 days

Frequency \*  
Daily

Select Time \*  
06:50 pm ⌚

① Scheduled Daily at 18:50 Asia/Calcutta

[Save](#) [Cancel](#)



2. Weekly: Users can also schedule the report weekly and select the day on a week when the report needs to be generated.

**Container Image Vulnerability Report** ⓘ [Change](#)

Configure schedule & email recipients for this report

[OnDemand](#) [Schedule](#)

Name \*

Email \*

Description

Date Range \*

Select the day on which report needs to be generated

Frequency \*

 Weekly

Repeat by Day

M T W T F S S

Select Time \*

 06:50 pm

Scheduled Asia/Calcutta

[Save](#) [Cancel](#)



3. Monthly: Users can also configure the report duration as monthly where they will be getting the report on the 1<sup>st</sup> of every month. It will soon be

configurable as the user-defined date as well.

**Container Image Vulnerability Report** ⓘ Change

Configure schedule & email recipients for this report

[OnDemand](#) [Schedule](#)

Name \*

Email \*

Description

Date Range \*

Frequency \*

Monthly

Select Time \*

 ⏰

( ⓘ Scheduled Monthly on 1st at 18:50 Asia/Calcutta

[Save](#) [Cancel](#)

**Step 4:** Once the report generation is completed you can see the View option in the UI as well as the report will be mailed to the email address. If the user wants to see the report in the UI they can click on the View.

Report Name	Email Recipients	Report Type	Created At	Actions
scheduled_report	[REDACTED]	Schedule	2024-11-18 19:02:32 IST	<a href="#">VIEW (2024-11-18 19:05:28 IST)</a>

# RINC

RINC (short for "Reporting IN Cluster") is a simple and lightweight reporting tool that provides insights into the status of a Kubernetes cluster, as well as other services running within it.

It includes built-in alerting capabilities, allowing users to define alerts using an expression language. RINC comes with a set of practical and sensible pre-configured alerts, which are included in the provided Helm charts. If you need to customize or extend these alerts, you can easily do so using our expression language, which is powered by the [gval](#) Go library.

RINC also supports email integration, allowing you to receive alerts via email.

## Supported reports

- Kubernetes deployment and statefulset status reports
- Long-running job reports
- Registry scan job status reports
- Supports reporting jobs where the module container has succeeded but the artifact-api container has failed.
- Kubernetes deployment and statefulset image tag reports

- RabbitMQ metrics reports
- CEPH metrics reports
- Pod status reports
- PV Utilization report
- Pod & Node resource utilization report
- Token expiry report
- Nodes' time-in-sync report
- Connectivity & Status checks for,
- Vault
- MongoDB
- Redis/KeyDB
- Neo4j
- Postgresql
- Prometheus
- Metabase
- AWS RDS
- Weaviate
- Onboarded registries status report
- [Kueue](#) workload status report
- Supports reporting jobs where the module container has succeeded but the artifact-api container has failed.

# Installation

We recommend installing RINC through our provided helm charts.

*Note: RINC uses MongoDB as its data store and creates a new collection called "rinc" upon launch. It is recommended that you create a separate MongoDB user with R/W access to the "rinc" collection. See the section on [Minimum Required Database Permissions](#).*

VERSION=0.9.0

```
helm show values oci://public.ecr.aws/k9v9d5v2/accuknox-rinc --version "$VERSION" >
values.yaml
```

The file `values.yaml` is well-documented and includes all configurable options for RINC. Please go through it and adjust the values as needed to suit your preferences. See [passing database/vault credentials](#) to RINC.

By default, all reports are disabled and can be enabled by setting `enable` to `true` in the Helm chart values. For example, to enable the RabbitMQ report, set:

```
config:  
  rabbitmq:  
    enable: true
```

If you are using our Accuknox Helm charts, we provide an `accuknox-values.yaml` file with most of the values pre-configured.

```
helm pull oci://public.ecr.aws/k9v9d5v2/accuknox-rinc --version "$VERSION"  
tar xvzf "accuknox-rinc-$VERSION.tgz"  
less accuknox-rinc/accuknox-values.yaml
```

RINC supports reading secrets directly from Vault. If you are using Hashicorp's Vault, please refer to the section on [Vault](#).

After customizing the values to your preferences, run the Helm install command below to deploy RINC in your cluster:

```
NAMESPACE="accuknox-rinc"  
  
helm upgrade rinc oci://public.ecr.aws/k9v9d5v2/accuknox-rinc \  
  --install \  
  --namespace "$NAMESPACE" \  
  --create-namespace \  
  --version "$VERSION" \  
  --values values.yaml
```

To check if everything is healthy, run:

```
watch kubectl -n "$NAMESPACE" get pod,job,cronjob,secret,configmap
```

If everything appears healthy and running, congratulations! RINC has been successfully installed on your cluster.

# Passing Database Credentials

Database credentials are used for connectivity checks. There are 3 ways to pass your database credentials to RINC,

## 1. Using Helm:

Set `secretConfig.create` to true in the helm values and fill the secrets below to let Helm create a Kubernetes Secret that is mounted into RINC.

```
secretConfig:  
  create: true  
  config:  
    mongodb:  
      #####  
      ##### REDACTED #####  
      #####
```

## 2. Manually Creating a Secret:

Below is a template for the Secret manifest,

```
apiVersion: v1  
kind: Secret  
metadata:  
  name: credentials  
  namespace: accuknox-rinc  
type: Opaque  
stringData:  
  secret.yaml: |-  
    # Please fill in the configuration below if you have set `vault.use` to  
    # true above.  
  vault:  
    auth:  
      # vault auth type  
      #  
      # Possible values: "token", "kubernetes"  
      type: ""  
      # Token used to authenticate to vault. Required when auth type is set to  
      # "token".  
      token: ""
```

```
# Role name used to authenticate to vault. Required when auth type is set
# to "kubernetes".
role: ""
# Service-specify credentials.
#
# It is recommended to create a dedicated `rinc` user for each of the
# services.
mongodb:
  username: ""
  password: ""
email:
smtp:
  host: ""
  username: ""
  password: ""
  port: 587
rabbitmq:
  management:
    # basic auth username for the management api.
    username: ""
    # basic auth password for the management api.
    password: ""
ceph:
  # ceph reporter uses ceph's dashboard API to scrape ceph status and
  # metrics.
  dashboardAPI:
    # username to authenticate with ceph dashboard API.
    username: ""
    # password to authenticate with ceph dashboard API.
    password: ""
connectivity:
neo4j:
  # neo4j basic auth username
  username: ""
  # neo4j basic auth password
  password: ""
postgres:
  # postgresql auth username.
  username: ""
  # postgresql auth password.
  password: ""
rds:
  # aws access key id
  accessKeyId: ""
  # aws secret access key
  secretAccessKey: ""
tokenExpiry:
  # list of token whose expiry need to be checked.
  #
  # It is recommended to NOT specify the token value here as it will remain
  # static. If you are using Vault, you can specify the vault `path` as
  # documented in the `config` section. If you are NOT using Vault, you can
  # use ExternalSecrets that will periodically sync the token value.
```

```
tokens: []
# - name: ""
#   value: ""
cloudScan:
onboardedRegistries:
postgres:
# postgresql auth username.
username: ""
# postgresql auth password.
password: ""
```

```
kubectl apply -f credentials.yaml
```

This secret must then be referenced in the helm chart values,

```
# This section is for specifying an existing Kubernetes Secret that the Helm
# chart should reference
existingSecret:
# name of the existing Secret in the Kubernetes cluster
name: "credentials"
# key within the Secret, which corresponds to the specific value to be used.
key: "secret.yaml"
```

### 3. Reading credentials directly from Vault

RINC can read credentials directly from Vault. To configure RINC to connect to Vault, specify the connection details in the Helm values under secretConfig.config.vault and ensure that secretConfig.create is set to true. Helm will pass the Vault credentials to RINC via the created Kubernetes Secret, allowing RINC to use these credentials to connect to Vault and read the remaining credentials directly from it.

See the section on [Vault](#) for setting up the required Vault policies.

```
secretConfig:
create: true
config:
# Please fill in the configuration below if you have set `vault.use` to
# true above.
vault:
auth:
# vault auth type
#
# Possible values: "token", "kubernetes"
type: ""
# Token used to authenticate to vault. Required when auth type is set to
```

```
# "token".
token: ""
# Role name used to authenticate to vault. Required when auth type is set
# to "kubernetes".
role: ""
```

# Accessing RINC's web interface

By default, RINC is not exposed to the outside world. To access RINC's web interface, port-forward to the `rinc-web` service:

```
kubectl -n "$NAMESPACE" port-forward svc/rinc-web 8080:80
```

Now open `<http://localhost:8080>` in your browser.

## An overview of RINC's web interface

The screenshot shows the RINC web interface with the following sections:

- Top Navigation:** ACCUKNOW (with a blue hexagon icon), History, Console.
- Main Dashboard:** A grid of cards with status indicators and links:
  - RabbitMQ (⚠ 1 >)
  - CEPH (✖ 2 >)
  - Image Tags (▶)
  - Deployment & Statefulset Status (⚠ 3 >)
  - Kubernetes Jobs (⚠ 1 >)
  - PV Utilization (▶)
  - Resource Utilization (⚠ 1 >)
  - Connectivity (▶)
  - Pod Status (⚠ 1 >)
  - Token Expiry (▶)
  - Node's Time Sync Status (▶)
  - Cloud Scan (⚠ 1 ⓘ 1 >)
  - Queue Workloads (▶)
- Footer:** localhost:8080/20241224023001/cloudscan, Generated: 2024-12-24 02:30:01 UTC

If you open RINC's web interface immediately after installation, the reporting cronjob might not have scheduled yet, so you may see an empty welcome screen instead of the dashboard. However, don't worry - you can go to the Console by clicking on the top-right section of the page and start an "on-demand scan".

The screenshot shows the AccuKnox web interface with the 'Console' section selected. At the top, there is a 'Run Scan' button with a purple background and white text. Below it, there is a 'Logs' tab with a blue background and white text. The main area displays a message: 'Creates a new Kubernetes Job to execute an immediate scan.' with a small icon to its left.

This will immediately launch a Kubernetes Job to aggregate all the metrics and generate a report for you. The job will take some time depending on the size of your cluster and workloads. Once the job is completed, you will see a dashboard similar to the example above.

## An overview of the reports generated by RINC

The screenshot shows a detailed RabbitMQ report generated by RINC. At the top, it displays the title 'RabbitMQ (2024-12-24 10:30:06 UTC)'. Below this, there is a yellow banner with the text '⚠️ RabbitMQ ready messages count has reached 124222'. The report is divided into several sections:

- Alerts:** A yellow banner indicating a warning about the ready messages count.
- Summary:** A table showing various RabbitMQ statistics:
 

Total messages in Queues	124240
Unacknowledged messages in Queues	18
Ready messages in Queues	124222
RabbitMQ Version	3.12.13
Total channels	1858
Total connections	1848
Total consumers	38
Total exchanges	217
Total queues	222
- Nodes:** A table showing details for three RabbitMQ nodes:
 

Name	Running	CPU	Mem Used	Free Disk	Processes	Sockets	FD Used	Network Partitions	Uptime	Plugins
rabbit@rabbitmq-server-0.rabbitmq-nodes.rabbitmq	true	4	1026.34 MiB 1064.96 MiB high watermark	92.51 GiB 1.86 GiB low watermark	10545	692	766	No	2547h37m1s	rabbitmq_peer_discovery_k8s rabbitmq_prometheus rabbitmq_management
rabbit@rabbitmq-server-1.rabbitmq-nodes.rabbitmq	true	4	767.76 MiB 1064.96 MiB high watermark	92.53 GiB 1.86 GiB low watermark	7060	492	552	No	311h4m38s	rabbitmq_management rabbitmq_peer_discovery_k8s rabbitmq_prometheus rabbitmq_shovel rabbitmq_shovel_management
rabbit@rabbitmq-server-2.rabbitmq-nodes.rabbitmq	true	4	885.02 MiB 1064.96 MiB high watermark	92.51 GiB 1.86 GiB low watermark	9300	664	763	No	603h21m42s	rabbitmq_peer_discovery_k8s rabbitmq_prometheus rabbitmq_management
- Queues:** A table showing queue statistics:
 

Name	State	Messages	Unacknowledged Messages	Ready Messages	Durable
------	-------	----------	-------------------------	----------------	---------

Above is an example RabbitMQ report.

Every report begins with an **Alerts** section, displaying any fired alerts. The alerts are color-coded based on their severity:

1. **Red** - Indicates a critical alert.
2. **Yellow** - Indicates a warning.
3. **Info** - Provides useful information.

Critical alerts typically require immediate action. Warning alerts, if not addressed in time, may impact operations. Info alerts provide useful details, such as the number of onboarded registries and nodes.

As a cluster operator, ensure there are no critical alerts.

*Note: As described earlier, RINC supports email integration, allowing you to receive these alerts via email. Refer to the email section in the Helm chart to configure email integration.*

The rest of the report varies depending on the type of report and includes insights about the cluster/service.

## Fetching Old Reports

RINC retains old reports for the duration specified in config.maintenance.metricsRetention in the Helm values. To retrieve old reports, click on History at the top-right of the web interface to access the history page.



📅

Search

**January 2025 ▾**      ↑    ↓

S	M	T	W	T	F	S
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

[Clear](#)      [Today](#)

## History Page

On this page, select the desired date to fetch the reports and click **Search**.



01/07/2025



Search

02:30 UTC (2025-01-06)

10:30 UTC (3 days ago)

18:30 UTC (2 days ago)

History Search Results - All times are in UTC.

## Advanced

Minimum Required Database Permissions for RINC to  
Generate Reports

## **MongoDB:**

RW access to the `rinc` collection

## **Postgresql:**

SELECT access to the following within the `cwpp` schema (within the `accuknox` database) tables,

1. `registry_scan_details`
2. `registries`
3. `image_scan_details`
4. `registry_configuration`
5. `workspaces`
6. `clusters`
7. `node`

The query below creates a user named `rinc` with SELECT access to the listed tables under the `cwpp` schema.

```
CREATE USER rinc WITH PASSWORD 'tryguessingthis';
GRANT CONNECT ON DATABASE accuknox TO rinc;
GRANT USAGE ON SCHEMA cwpp TO rinc;
GRANT SELECT ON
    cwpp.registry_scan_details,
    cwpp.registries,
    cwpp.image_scan_details,
    cwpp.registry_configuration,
    cwpp.workspaces,
    cwpp.clusters,
    cwpp.nodes
TO rinc;
```

## **Neo4j:**

Neo4j requires authentication to ping the database. It is recommended you created a separate database called "rinc" and a user, also called "rinc". This database is not going to be used and is only present to allow RINC to authenticate with neo4j in order to test the connectivity.

## Vault Policy

If you are using Vault with Kubernetes auth, create a role and attach the necessary policy to allow reading your configured secrets.

Example, vault policy:

```
path "/accuknox/k8s/*" {  
    capabilities = ["read"]  
}  
  
path "/accuknox/aws/*" {  
    capabilities = ["read"]  
}  
  
path "/accuknox/artifacts/microservices/token" {  
    capabilities = ["read"]  
}
```

You also need to bind the role to the service accounts and namespace. RINC helm charts creates three service accounts. You can list them using,

```
kubectl -n "$NAMESPACE" get serviceaccounts
```

You should associate the role with all three service account names.

Once the role is created, refer to it in the Vault section of the Helm chart.

## CWPP Troubleshooting

If the user faces any issue related to clusters, then they should provide the logs information of their clusters for troubleshooting purposes.

## Requirements

### Getting Kubearmor Sysdump

Users can get the kubeArmor sysdump by using the following command:

```
karmor sysdump
```

## Getting logs from AccuKnox Agents

Along with KubeArmor Sysdump users will be required to send the logs of AccuKnox Agents running inside their cluster. To get the logs of each agent use the following commands:

```
kubectl logs -n accuknox-agents discovery-engine-xxxx-xxxx > discovery-engine-logs.txt  
kubectl logs -n accuknox-agents feeder-service-xxxx-xxx > feeder-service-logs.txt  
kubectl logs -n accuknox-agents policy-enforcement-agent-xxxx-xxx > PEA-logs.txt  
kubectl logs -n accuknox-agents shared-informer-agent-XXX-XXx > SIA-logs.txt
```

**Note:** In the above command replace the xxx-xxxx with your respective pod name that is running in accuknox-agents namespace.

The users will have to send this Karmor sysdump file and AccuKnox Agents logs to AccuKnox Solutions team for debugging the issue.

## Script To automate this process

- This script will save all the output Txt files in a single zip file
- karmor sysdump will run independently as it creates a separate zip file on its own

```
#!/bin/bash

# Function to get the pod name for a given deployment
get_pod_name() {
    local namespace=$1
    local deployment=$2
    kubectl get po -n "$namespace" -o=name | grep "$deployment" | awk -F/ '{print $2}'
}

# Function to fetch logs for a given pod and save them to a file
fetch_and_save_logs() {
    local namespace=$1
    local pod=$2
    local output_file=$3
    kubectl logs -n "$namespace" "$pod" > "$output_file"
}

# Main script starts here
```

```

# Set your desired namespace here
namespace="accuknox-agents"

# Get the pod names and store them in variables
discovery_engine_pod=$(get_pod_name "$namespace" "discovery-engine")
feeder_service_pod=$(get_pod_name "$namespace" "feeder-service")
pea_pod=$(get_pod_name "$namespace" "policy-enforcement-agent")
sia_pod=$(get_pod_name "$namespace" "shared-informer-agent")

# Create a temporary directory to store the log files
temp_dir=$(mktemp -d 2>/dev/null || mktemp -d -t 'mytmpdir')

# Fetch and save the logs to separate files in the temporary directory
fetch_and_save_logs "$namespace" "$discovery_engine_pod"
"$temp_dir/discovery-engine-logs.txt"
fetch_and_save_logs "$namespace" "$feeder_service_pod" "$temp_dir/feeder-service-logs.txt"
fetch_and_save_logs "$namespace" "$pea_pod" "$temp_dir/PEA-logs.txt"
fetch_and_save_logs "$namespace" "$sia_pod" "$temp_dir/SIA-logs.txt"

# Create a ZIP archive of all the log files
zip_file="agents_logs_archive.zip"
zip -j "$zip_file" "$temp_dir"/*.txt

# Clean up the temporary directory
rm -rf "$temp_dir"

echo "Logs have been fetched and saved to the ZIP archive: $zip_file"

# Execute 'karmor sysdump'
karmor sysdump

echo "karmor sysdump executed."

```

Users can now send the zip files generated for troubleshooting.

**Note:** Need to install zip as a pre-requisite in linux before running the above script.

```
sudo apt install zip
```

## Output

```
~/logs$ ./script.sh
adding: discovery-engine-logs.txt (deflated 95%)
adding: feeder-service-logs.txt (deflated 64%)
adding: PEA-logs.txt (deflated 43%)
adding: SIA-logs.txt (deflated 55%)
Logs have been fetched and saved to the ZIP archive: agents_logs_archive.zip
getting logs from kubearmor-p4rlf
tar: removing leading '/' from member names
Sysdump at karmor-sysdump-Thu Oct 19 07_45_31 UTC 2023.zip
karmor sysdump executed.
~/logs$ ls
agents_logs_archive.zip 'karmor-sysdump-Thu Oct 19 07_45_31 UTC 2023.zip' script.sh
~/logs$ |
```

# CSPM Troubleshooting Guide

This guide helps troubleshoot onboarding and scanning issues for the Accuknox CNAPP SaaS deployment across AWS, Azure, and GCP.

## Step 1: Validate Prerequisites

Ensure the required permissions are granted to the user or application for the respective cloud account.

### AWS Permissions

1. **Login to AWS Console.**
2. Navigate to **IAM > Users**.
3. Select the user created for AccuKnox onboarding.

The screenshot shows the AWS Identity and Access Management (IAM) service. In the left sidebar, under 'Access management', the 'Users' option is selected. The main area displays a table titled 'Users (439)'. A search bar at the top right shows the query 'cloud\_'. One result is listed: 'cloud\_read\_access' with a path of '/'. The user has 0 groups, last activity 3 hours ago, and no MFA or password age information. Action buttons for 'Delete' and 'Create user' are visible.

### 1. Go to the **Permissions** tab:

- Confirm the following policies are attached:
  - ReadOnlyAccess (AWS Managed - Job Function)
  - SecurityAudit (AWS Managed - Job Function)

This screenshot shows the 'Permissions' tab for the 'cloud\_read\_access' user. The left sidebar is identical to the previous screenshot. The main area shows 'Permissions policies (2)' attached directly to the user. Two policies are listed: 'ReadOnlyAccess' and 'SecurityAudit', both categorized as 'AWS managed - job function' and attached 'Directly'. Below this, there's a section for 'Permissions boundary (not set)', 'Generate policy based on CloudTrail events' (with a note about generating a new policy from access activity), and a message stating 'No requests to generate a policy in the past 7 days.'

## Azure Permissions

1. **Login to Azure Portal.**
2. **Navigate to App Registrations:**

- Select the application registered for onboarding.
- Go to the **API Permissions** tab and verify:
  - Directory.Read.All is listed under **Application Permissions**.

The screenshot shows the Microsoft Azure portal interface for managing API permissions. The left sidebar shows various tabs like Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage, and API permissions. The API permissions tab is selected. The main content area shows the 'Configured permissions' section for the app 'azure-onboarding-user'. It lists two permissions: 'Microsoft Graph (2)' and 'User.Read'. For 'Microsoft Graph (2)', the 'Admin consent required' status is 'Yes' and the 'Status' is 'Granted for Default Directory'. For 'User.Read', the 'Admin consent required' status is 'No' and the 'Status' is 'Granted for Default Directory'. There are also some informational messages at the top about tenant-wide consent and admin consent requirements.

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (2)				...
Directory.Read.All	Application	Read directory data	Yes	Granted for Default Dire... ***
User.Read	Delegated	Sign in and read user profile	No	Granted for Default Dire... ***

## 1. Navigate to **Subscriptions**:

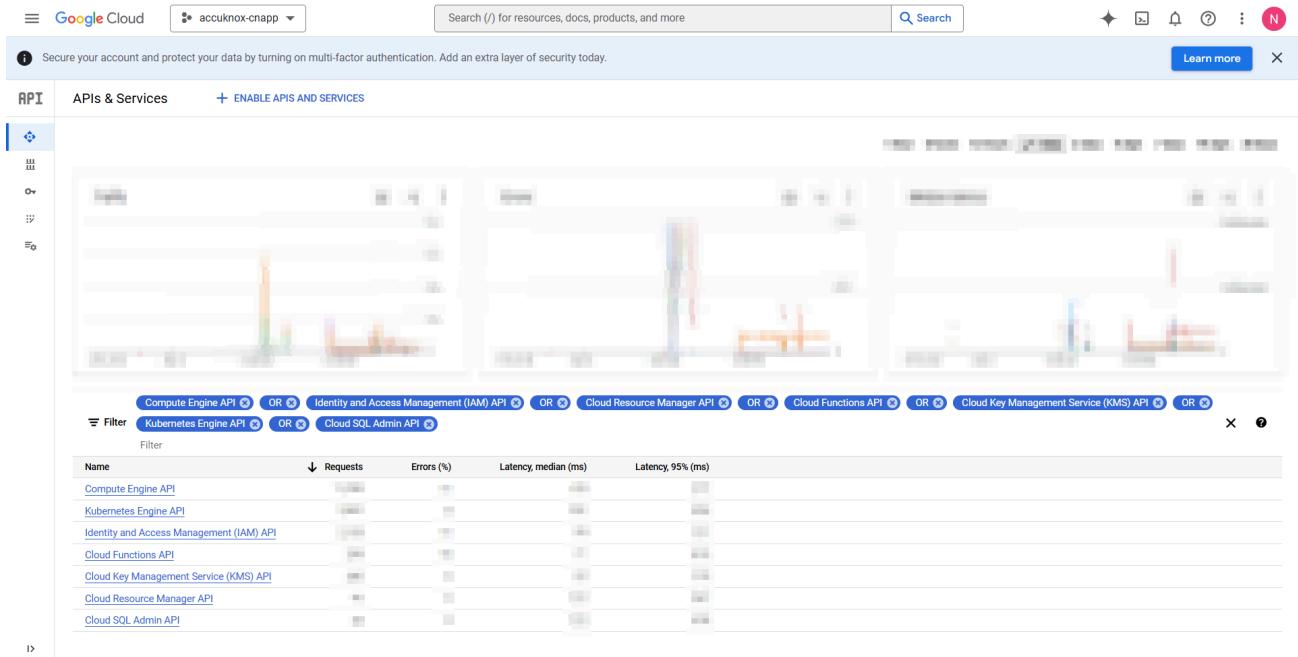
- Select the relevant subscription.
- Go to **Manage > Access control (IAM)**.
- Verify the registered application has the following roles assigned:
  - Security Reader (Job Function Role for subscriptions)
  - Log Analytics Reader (Job Function Role for subscriptions)

The screenshot shows the Microsoft Azure Subscriptions page for the 'Pay-As-You-Go' subscription. The left sidebar lists 'Subscriptions' and shows one item: 'Pay-As-You-Go'. The main content area is titled 'Pay-As-You-Go | Access control (IAM)'. It displays the number of role assignments (19) and provides links to 'View assignments' and 'Edit columns'. A search bar and filter options ('Add', 'Advanced options') are at the top. The 'Access control (IAM)' tab is selected, showing a list of roles assigned to service principals. The table includes columns for Name, Type, Role, Scope, and Condition. Two entries are listed under 'Log Analytics Reader (1)' and 'Security Reader (1)', both assigned to the service principal 'azure-onboarding-user'.

Name	Type	Role	Scope	Condition
azure-onboarding-user	App	Log Analytics Reader	This resource	None
azure-onboarding-user	App	Security Reader	This resource	None

## GCP Permissions

1. **Login to Google Cloud Console.**
2. Navigate to **IAM & Admin > IAM:**
  - Find the service account created for onboarding.
  - Verify the following roles are assigned:
    - a. roles/viewer (Viewer Role)
    - b. roles/iam.securityReviewer (Security Reviewer Role)
    - c. roles/logging.viewer (Log Viewer Role)
3. Navigate to **APIs & Services > Library:**
  - Ensure the following APIs are enabled:
    - a. Compute Engine API
    - b. Identity and Access Management (IAM) API
    - c. Cloud Resource Manager API
    - d. Cloud Functions API
    - e. KMS API
    - f. Kubernetes API
    - g. Cloud SQL Admin API



If permissions and APIs are configured correctly, proceed to the next step.

#### Refer to the prerequisites for more info:

- AWS Onboarding Prerequisites
- Azure Onboarding Prerequisites
- GCP Onboarding Prerequisites

## Step 2: Verify Cloud Scan Status

1. Log in to the **AccuKnox SaaS platform**.
2. Navigate to **Settings > Cloud Account**.
3. Select the **specific cloud account** in question.
4. Review the **status of the cloud scan**:

The screenshot shows the ACCUKNOX Cloud Accounts dashboard. The left sidebar contains navigation links such as Dashboard, Inventory, Issues, Compliance, Runtime Protection, Remediation, Monitors / Alerts, Identity, Reports, Notifications, Settings, and Cloud Accounts. A purple bar at the bottom indicates 'Getting started Onboarding'.

The main area displays a table of Cloud Account scans. The columns are: Cloud Account, Connected, Enabled, Last scanned, Scan, Start Timestamp, Status, Job Type, Job Name, and Description.

**Cloud Account: gcp: accuknox-cnapp**

Start Timestamp	Status	Job Type	Job Name	Description
2024-12-06 08:04:32 (UTC)	Success	Findings Scan	[REDACTED]	unable to retrieve container logs for containerId://c7...
2024-12-06 08:02:13 (UTC)	Success	Assets Scan	[REDACTED]	
2024-12-05 11:40:16 (UTC)	Success	Findings Scan	[REDACTED]	unable to retrieve container logs for containerId://35e...
2024-12-05 11:37:07 (UTC)	Success	Assets Scan	[REDACTED]	

**Cloud Account: gcp: shaped-infusion-402417**

Start Timestamp	Status	Job Type	Job Name	Description
2024-08-05	Success	Findings Scan	[REDACTED]	Starting the main function at l733452461!196165 Che...
2024-07-21	Success	Assets Scan	[REDACTED]	

**Cloud Account: aws: aws-[REDACTED]**

Start Timestamp	Status	Job Type	Job Name	Description
2024-10-22	Success	Findings Scan	[REDACTED]	unable to retrieve container logs for containerId://8a...
2024-12-06 08:04:19 (UTC)	Failed	Findings Scan	[REDACTED]	unable to retrieve container logs for containerId://8a...
2024-12-06 08:03:01 (UTC)	Success	Assets Scan	[REDACTED]	
2024-12-06 08:02:56 (UTC)	Failed	AWS Tools Scan	[REDACTED]	unable to retrieve container logs for containerId://fc3...
2024-12-06 08:02:28 (UTC)	Failed	AWS Tools Scan	[REDACTED]	unable to retrieve container logs for containerId://ba...
2024-12-05 11:37:49 (UTC)	Failed	Findings Scan	[REDACTED]	unable to retrieve container logs for containerId://ba...
2024-12-05 11:37:28 (UTC)	Success	Assets Scan	[REDACTED]	

Rows per page: 20 ▶ 1 ▶