Project: Enterprise Java CI/CD with Jenkins, GitOps, and Kubernetes Observability

GitHub Repo: https://github.com/Mithra1995/Jenkins-argocd

Description:

This project demonstrates a complete CI/CD and monitoring solution for a cloud-native Java microservice. The pipeline automates the stages from **code commit to production deployment**, integrates code quality analysis, enables **declarative GitOps-based deployment** using **Argo CD**, and provides **end-to-end observability** with Prometheus and Grafana. This setup is ideal for organizations adopting DevOps, GitOps, and Kubernetes-native practices.

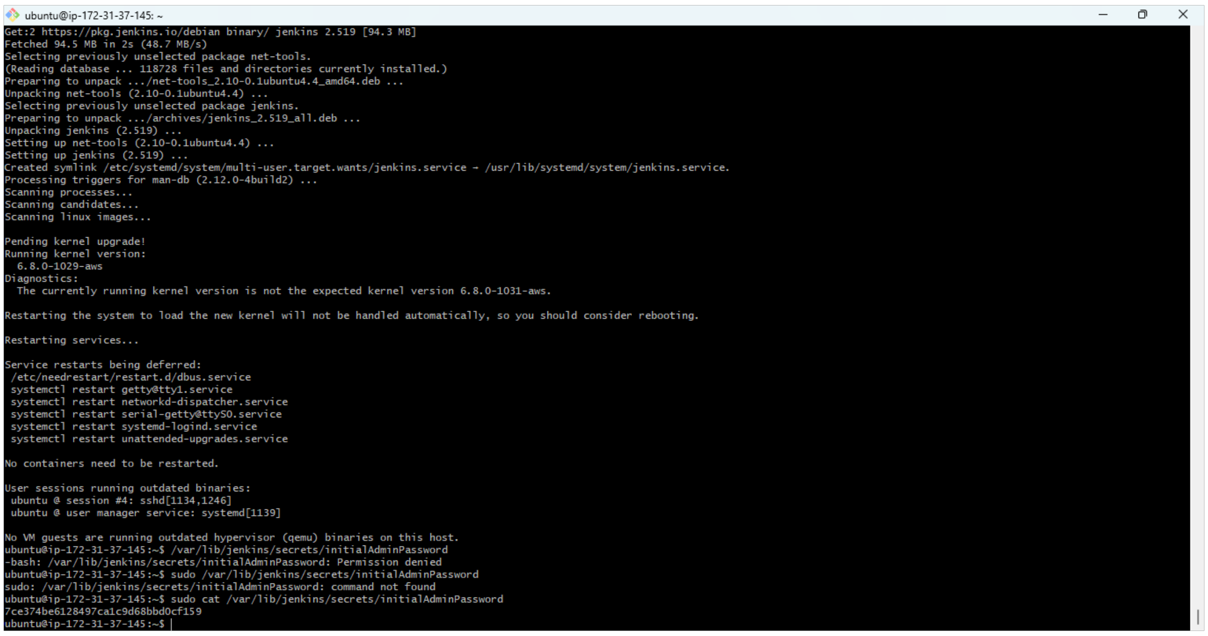
Technology Stack Overview:

Component	Purpose
Java + Maven	Application development and dependency management
Jenkins	Continuous Integration and Continuous Delivery (CI/CD) orchestration
SonarQube	Static code analysis and enforcement of quality gates
Docker	Containerization of the application
Kubernetes	Deployment and orchestration of containerized applications
Argo CD	GitOps-based Continuous Deployment
Prometheus	Metrics collection and monitoring
Grafana	Metrics visualization and alerting

Implementation: Step by step

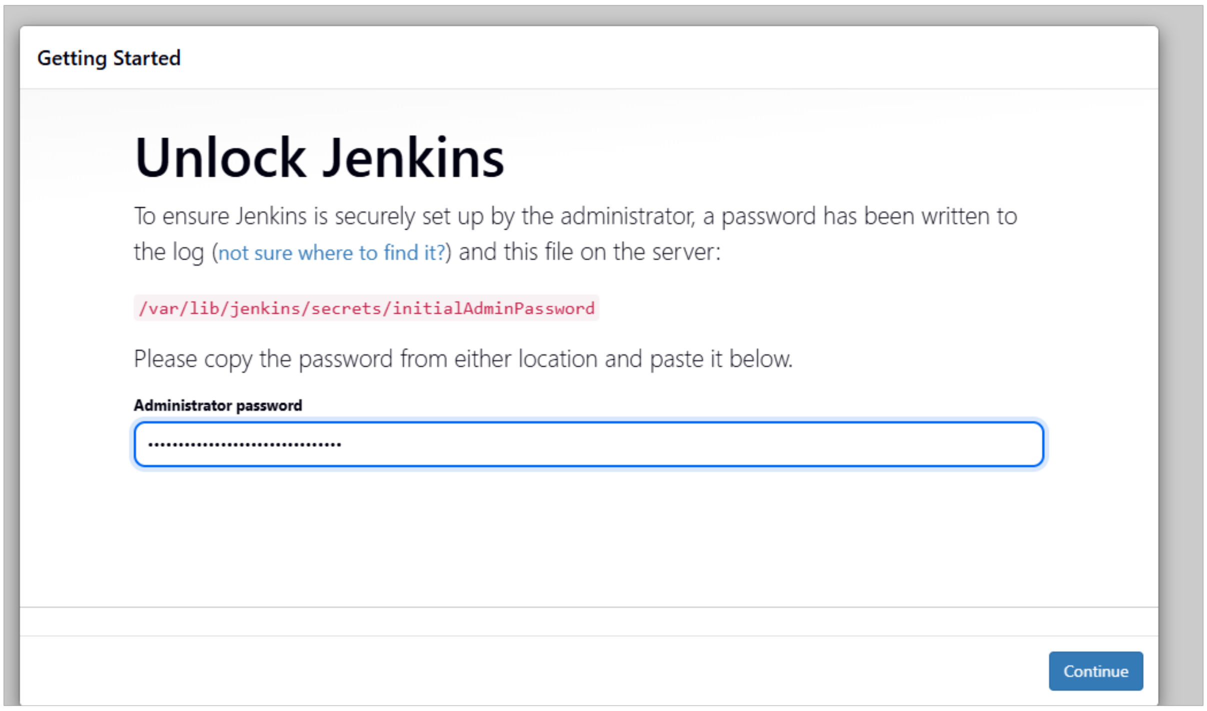
Step 1: Create Ec2 instance with instance type t2. large and volume size:50GB , then install below tools

- Java/Maven
- Jenkins
- Sonarqube
- Trivy
- Git
- Docker



```
ubuntu@ip-172-31-37-145: ~
Get:2 https://pkg.jenkins.io/debian binary/ jenkins 2.519 [94.3 MB]
Fetched 94.5 MB in 2s (48.7 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 118728 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4.4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4.4) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../archives/jenkins_2.519_all.deb ...
Unpacking jenkins (2.519) ...
Setting up net-tools (2.10-0.1ubuntu4.4) ...
Setting up jenkins (2.519) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-1029-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1031-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...

Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #4: sshd[1134,1246]
 ubuntu @ user manager service: systemd[1139]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-37-145:~$ /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-37-145:~$ sudo /var/lib/jenkins/secrets/initialAdminPassword
sudo: /var/lib/jenkins/secrets/initialAdminPassword: command not found
ubuntu@ip-172-31-37-145:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
7ce374be6128497ca1c9d68bbd0cf159
ubuntu@ip-172-31-37-145:~$ |
```

Step 2: Then provide creating the Jenkins webpage



Getting Started

Unlock Jenkins

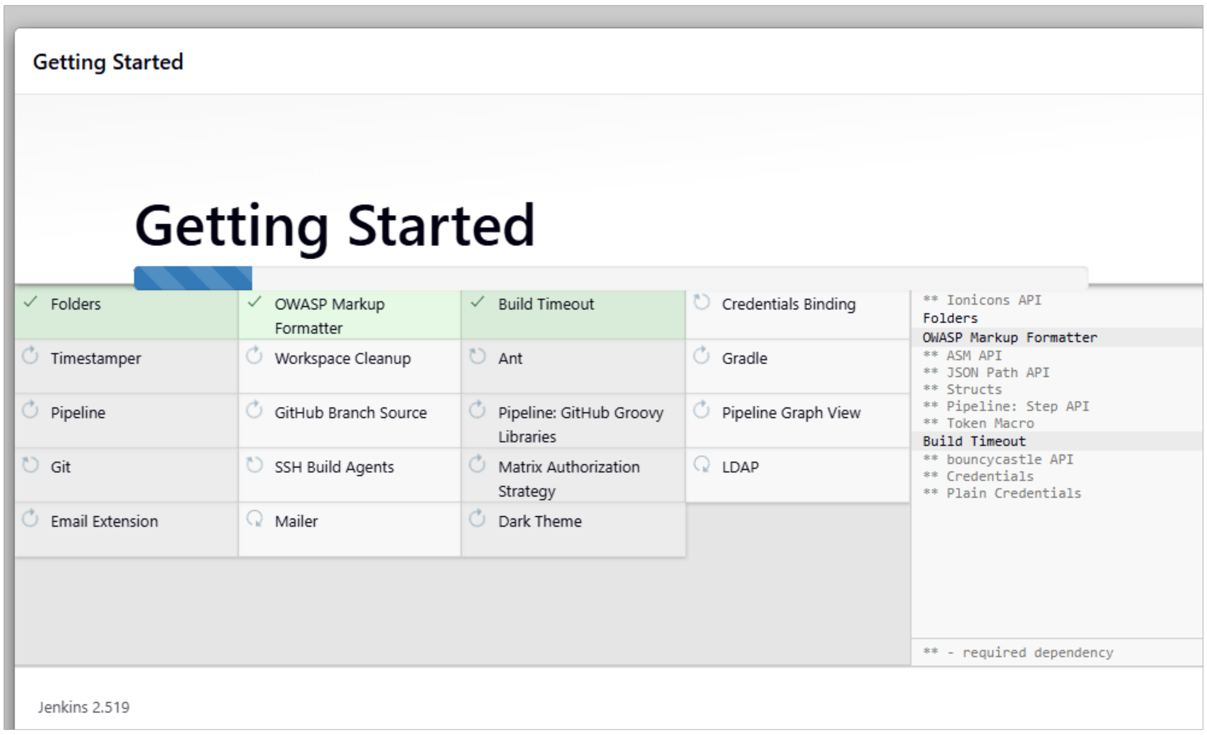
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

••••••••••••••••••••••••••••••••


Getting Started

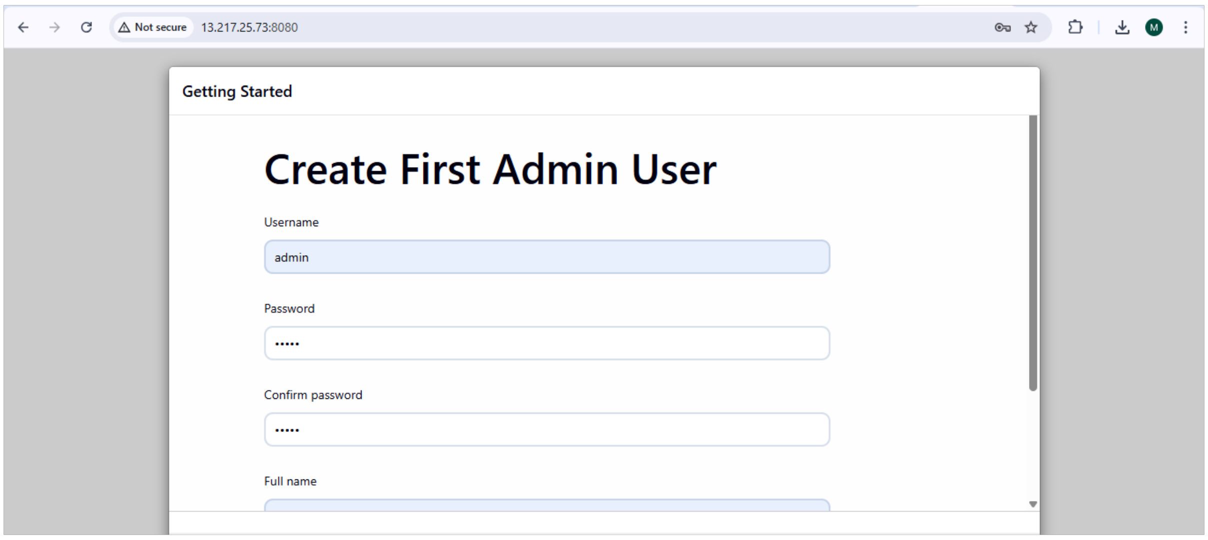
Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
Git	SSH Build Agents	Matrix Authorization Strategy	LDAP
Email Extension	Mailer	Dark Theme	

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** bouncycastle API
** Credentials
** Plain Credentials
```

** - required dependency

Jenkins 2.519


Not secure 13.217.25.73:8080

Getting Started

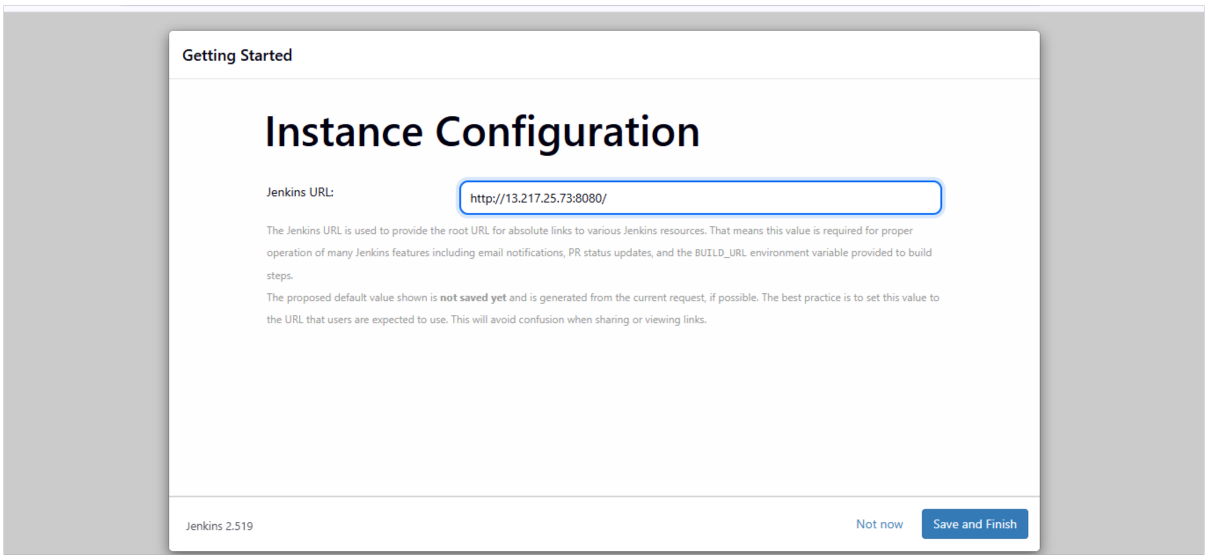
Create First Admin User

Username: admin

Password: •••••

Confirm password: •••••

Full name


Getting Started

Instance Configuration

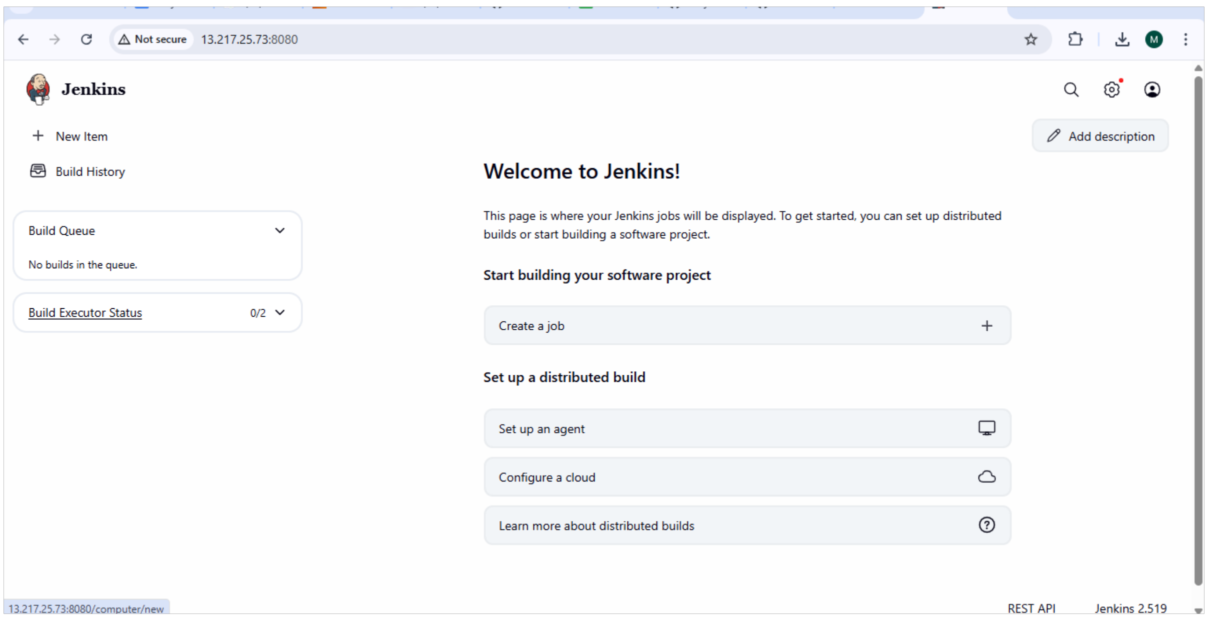
Jenkins URL: http://13.217.25.73:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.519

Step 3: Now create the Jenkins pipeline script for pushing the image to ECR and updating it in deployment.yaml file



Build Queue

No builds in the queue.

Build Executor Status 0/2

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

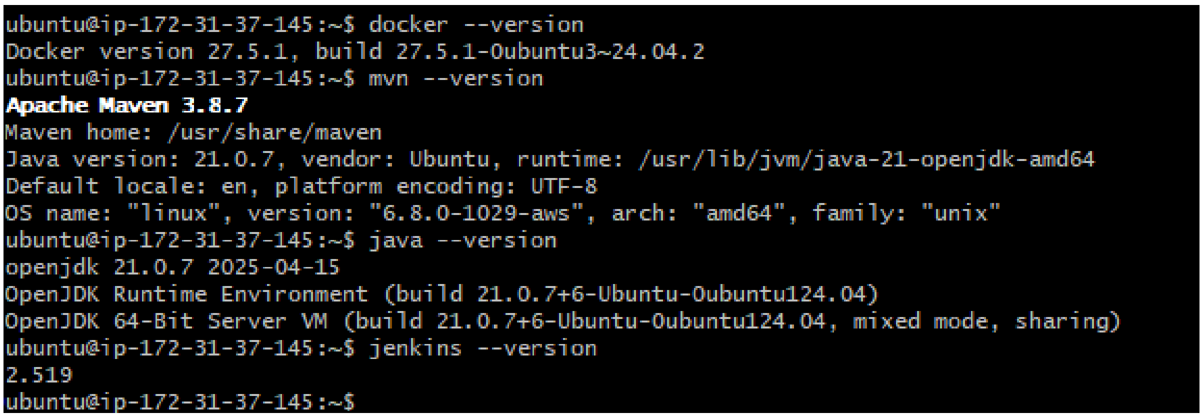
Create a job

Set up a distributed build

Set up an agent

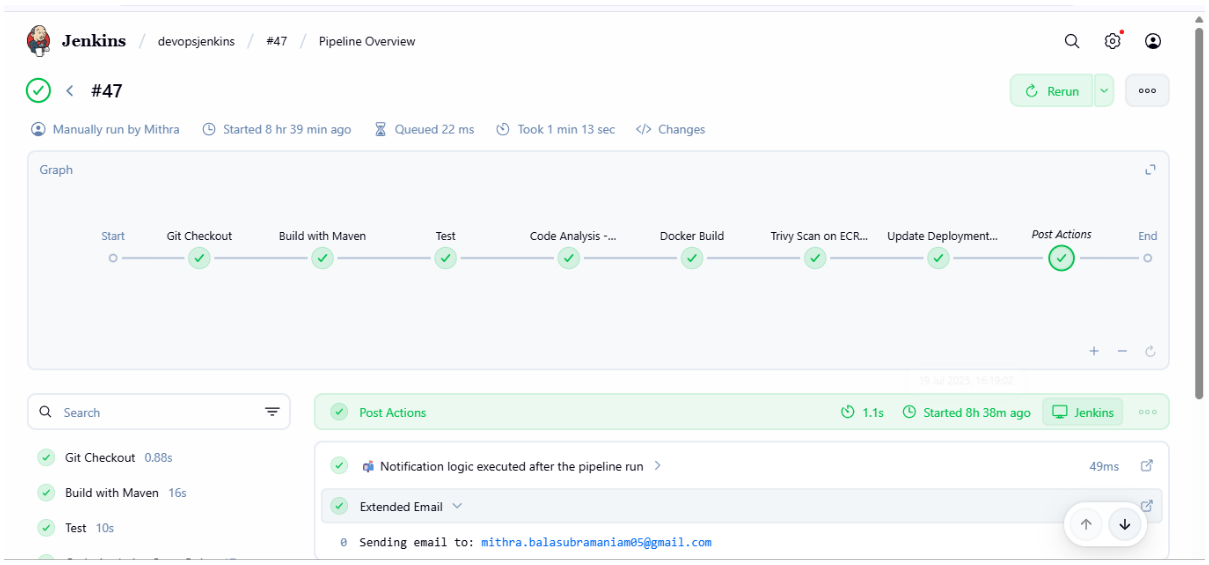
Configure a cloud

Learn more about distributed builds



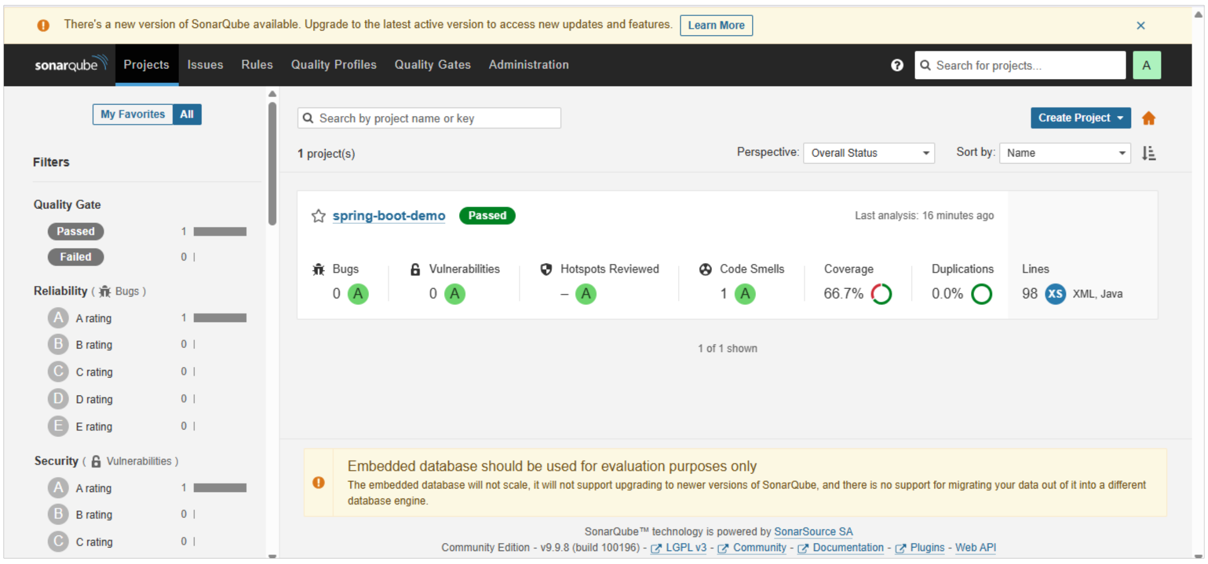
```
ubuntu@ip-172-31-37-145:~$ docker --version
Docker version 27.5.1, build 27.5.1-0ubuntu3~24.04.2
ubuntu@ip-172-31-37-145:~$ mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 21.0.7, vendor: Ubuntu, runtime: /usr/lib/jvm/java-21-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1029-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-37-145:~$ java --version
openjdk 21.0.7 2025-04-15
OpenJDK Runtime Environment (build 21.0.7+6-Ubuntu-0ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.7+6-Ubuntu-0ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-37-145:~$ jenkins --version
2.519
ubuntu@ip-172-31-37-145:~$
```

Step 4: Now the CI pipeline is working fine



```
0  Sending email to: mithra.balasubramaniam05@gmail.com
```

Step 5: We can see the SonarQube analysis for Code Analysis


There's a new version of SonarQube available. Upgrade to the latest active version to access new updates and features. Learn More

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects...

My Favorites All

Search by project name or key

Create Project

Filters

1 project(s)

Perspective: Overall Status

Sort by: Name

Quality Gate

Passed	1
Failed	0

Reliability (Bugs)

A rating	1
B rating	0
C rating	0
D rating	0
E rating	0

Security (Vulnerabilities)

A rating	1
B rating	0
C rating	0

spring-boot-demo Passed

Last analysis: 16 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	0 A	– A	1 A	66.7%	0.0%	98 XS XML, Java

1 of 1 shown

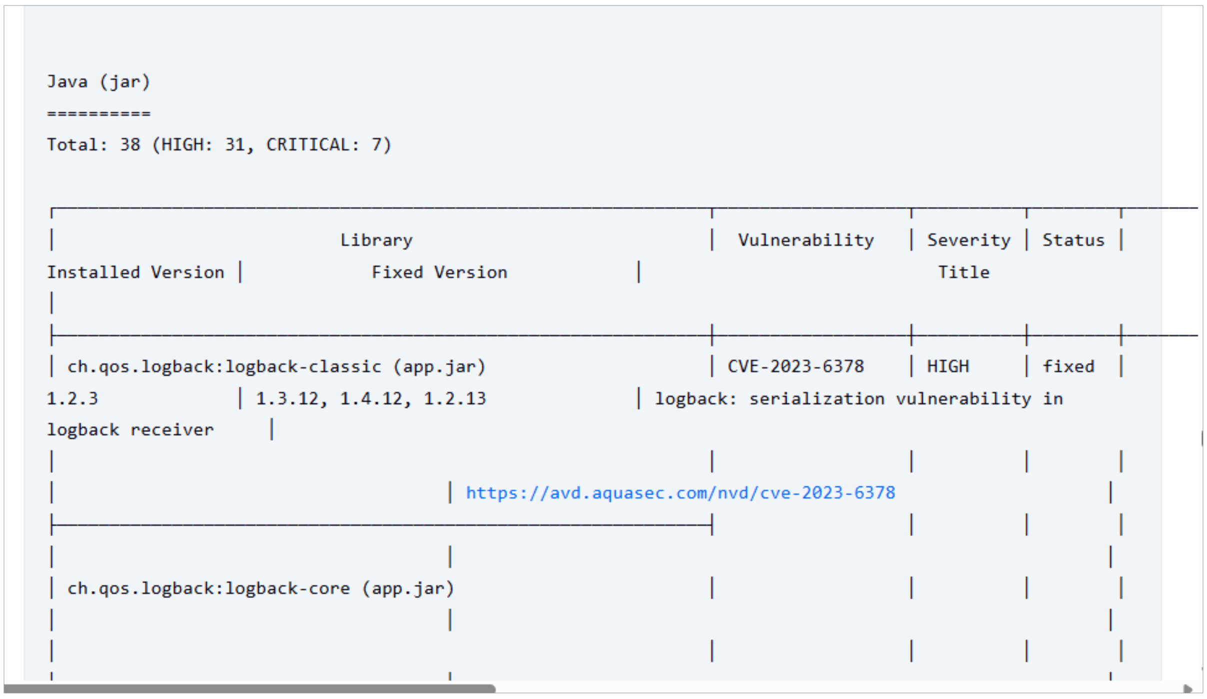
Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

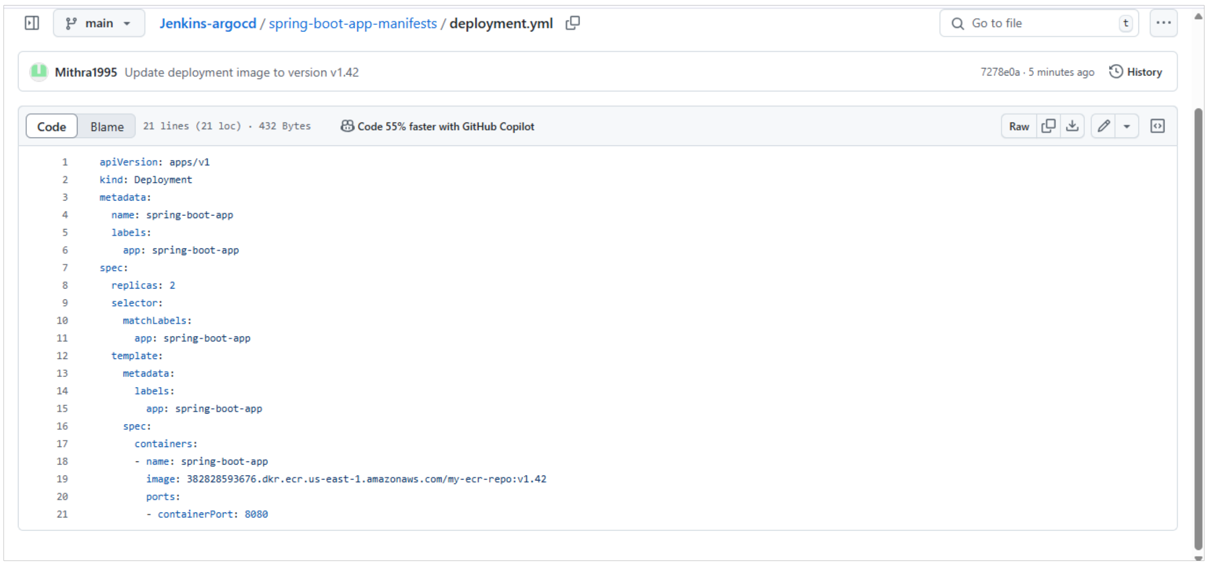
SonarQube™ technology is powered by SonarSource SA

Community Edition - v9.9.8 (build 100196) - LGPL v3 - Community - Documentation - Plugins - Web API

Step 6: Trivy scan for docker image vulnerability


```
Java (jar)
==========
Total: 38 (HIGH: 31, CRITICAL: 7)

┌──────────────────────────────────────────────────────────────┬──────────────────┬──────────┬────────┬────────
│                           Library                            │  Vulnerability   │ Severity │ Status │
Installed Version │           Fixed Version          │                    Title
│
├──────────────────────────────────────────────────────────────┼──────────────────┼──────────┼────────┼────────
│ ch.qos.logback:logback-classic (app.jar)                     │ CVE-2023-6378    │ HIGH     │ fixed  │
1.2.3             │ 1.3.12, 1.4.12, 1.2.13           │ logback: serialization vulnerability in
logback receiver     │
│                                                              │                  │          │        │
│                                      │ https://avd.aquasec.com/nvd/cve-2023-6378       │
├──────────────────────────────────────────────────────────────┤                  │          │        │
│                                      │                                                 │
│ ch.qos.logback:logback-core (app.jar)                        │                  │          │        │
│                                      │                                                 │
│                                                              │                  │          │        │
```

Step 7 : Image is pushed to deployment.yaml file



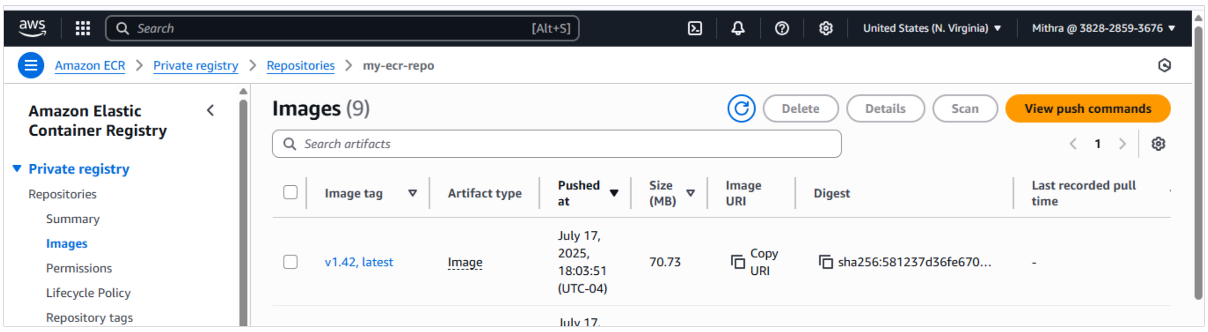
main · Jenkins-argocd / spring-boot-app-manifests / deployment.yml

Mithra1995 Update deployment image to version v1.42 · 7278e0a · 5 minutes ago · History

Code · Blame · 21 lines (21 loc) · 432 Bytes · Code 55% faster with GitHub Copilot

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spring-boot-app
  labels:
    app: spring-boot-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: spring-boot-app
  template:
    metadata:
      labels:
        app: spring-boot-app
    spec:
      containers:
      - name: spring-boot-app
        image: 382828593676.dkr.ecr.us-east-1.amazonaws.com/my-ecr-repo:v1.42
        ports:
        - containerPort: 8080
```

Step 8: Image is pushed to ECR



United States (N. Virginia) · Mithra @ 3828-2859-3676

Amazon ECR > Private registry > Repositories > my-ecr-repo

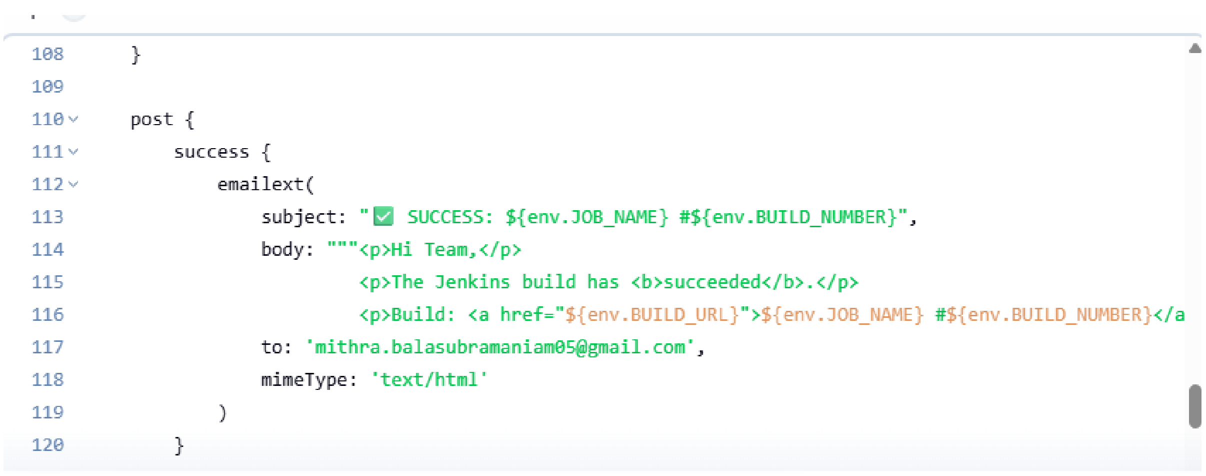
Amazon Elastic Container Registry

- Private registry
  - Repositories
    - Summary
    - Images
    - Permissions
    - Lifecycle Policy
    - Repository tags

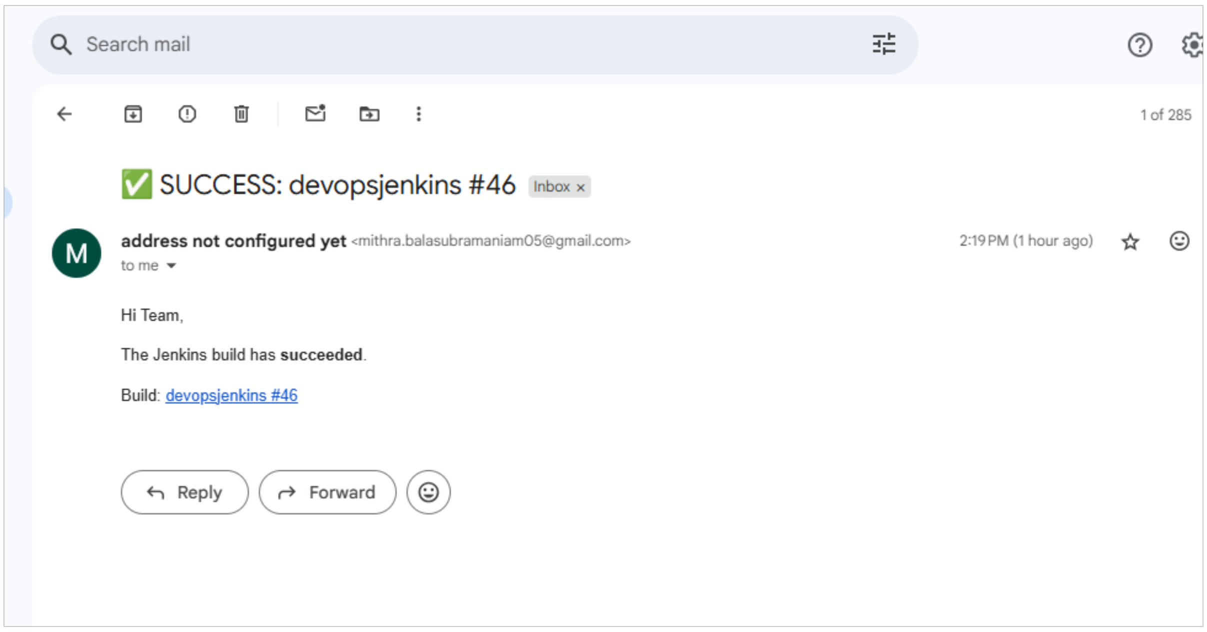
Images (9) · Delete · Details · Scan · View push commands

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
v1.42, latest	Image	July 17, 2025, 18:03:51 (UTC-04)	70.73	Copy URI	sha256:581237d36fe670...	-

Step 9: Add an email notification for post build success



```
    }

    post {
        success {
            emailext(
                subject: "✅ SUCCESS: ${env.JOB_NAME} #${env.BUILD_NUMBER}",
                body: """<p>Hi Team,</p>
                        <p>The Jenkins build has <b>succeeded</b>.</p>
                        <p>Build: <a href="${env.BUILD_URL}">${env.JOB_NAME} #${env.BUILD_NUMBER}</a
                to: 'mithra.balasubramaniam05@gmail.com',
                mimeType: 'text/html'
            )
        }
```



Search mail

1 of 285

✅ SUCCESS: devopsjenkins #46 Inbox ×

address not configured yet <mithra.balasubramaniam05@gmail.com>
to me

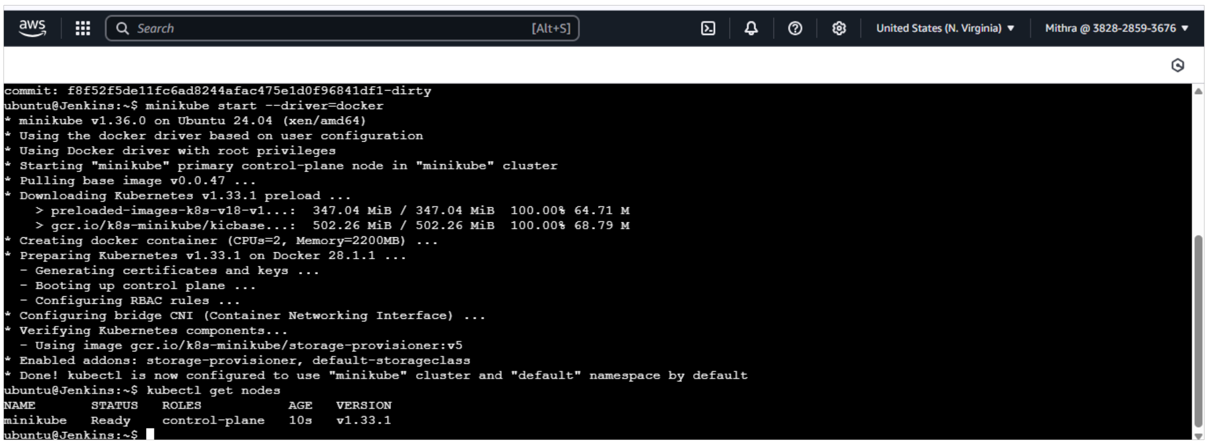
2:19 PM (1 hour ago)

Hi Team,

The Jenkins build has **succeeded**.

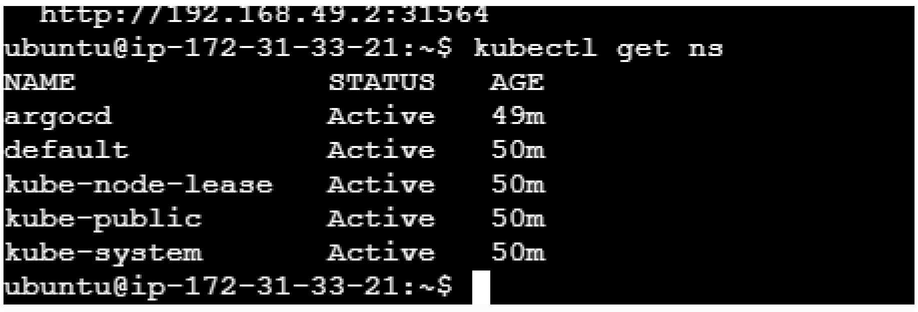
Build: devopsjenkins #46

Step 10 : Install Kubernetes(minikube)



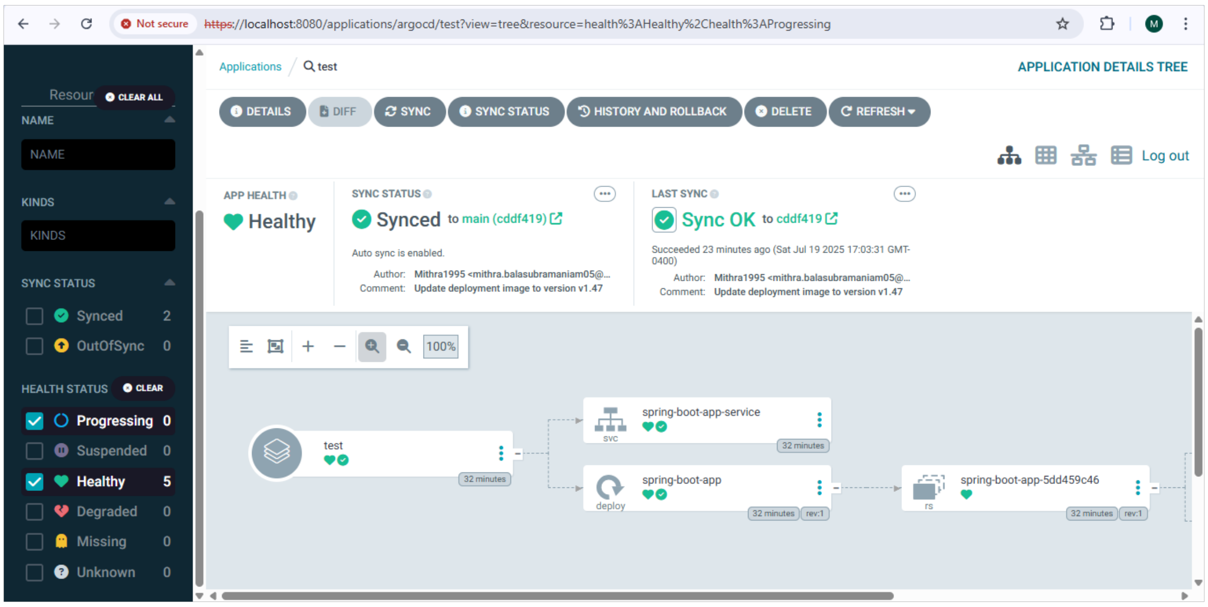
```
commit: f8f52f5de11fc6ad8244afac475e1d0f96841df1-dirty
ubuntu@Jenkins:~$ minikube start --driver=docker
* minikube v1.36.0 on Ubuntu 24.04 (xen/amd64)
* Using the docker driver based on user configuration
* Using Docker driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.47 ...
* Downloading Kubernetes v1.33.1 preload ...
    > preloaded-images-k8s-v18-v1...:  347.04 MiB / 347.04 MiB  100.00% 64.71 M
    > gcr.io/k8s-minikube/kicbase...:  502.26 MiB / 502.26 MiB  100.00% 68.79 M
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
ubuntu@Jenkins:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   10s   v1.33.1
ubuntu@Jenkins:~$
```

Step 11: Install argocd for Gitops



```
  http://192.168.49.2:31564
ubuntu@ip-172-31-33-21:~$ kubectl get ns
NAME              STATUS   AGE
argocd            Active   49m
default           Active   50m
kube-node-lease   Active   50m
kube-public       Active   50m
kube-system       Active   50m
ubuntu@ip-172-31-33-21:~$
```

Step 12: Open the argocd app and create the application with github page , once the health status is synced

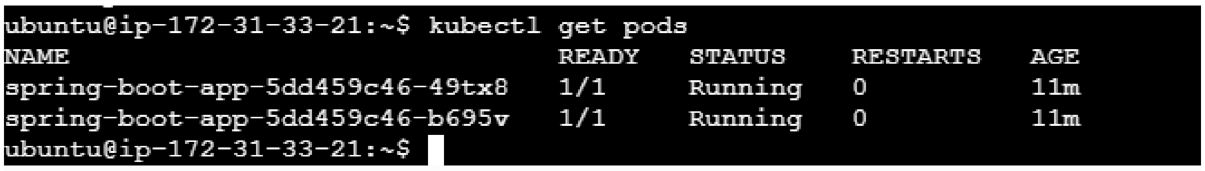


APP HEALTH
Healthy

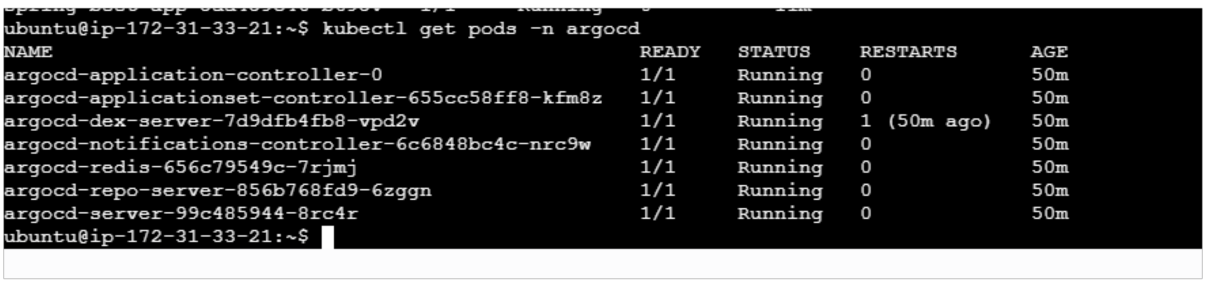
SYNC STATUS
Synced to main (cddf419)
Auto sync is enabled.
Author: Mithra1995 <mithra.balasubramaniam05@...
Comment: Update deployment image to version v1.47

LAST SYNC
Sync OK to cddf419
Succeeded 23 minutes ago (Sat Jul 19 2025 17:03:31 GMT-0400)
Author: Mithra1995 <mithra.balasubramaniam05@...
Comment: Update deployment image to version v1.47

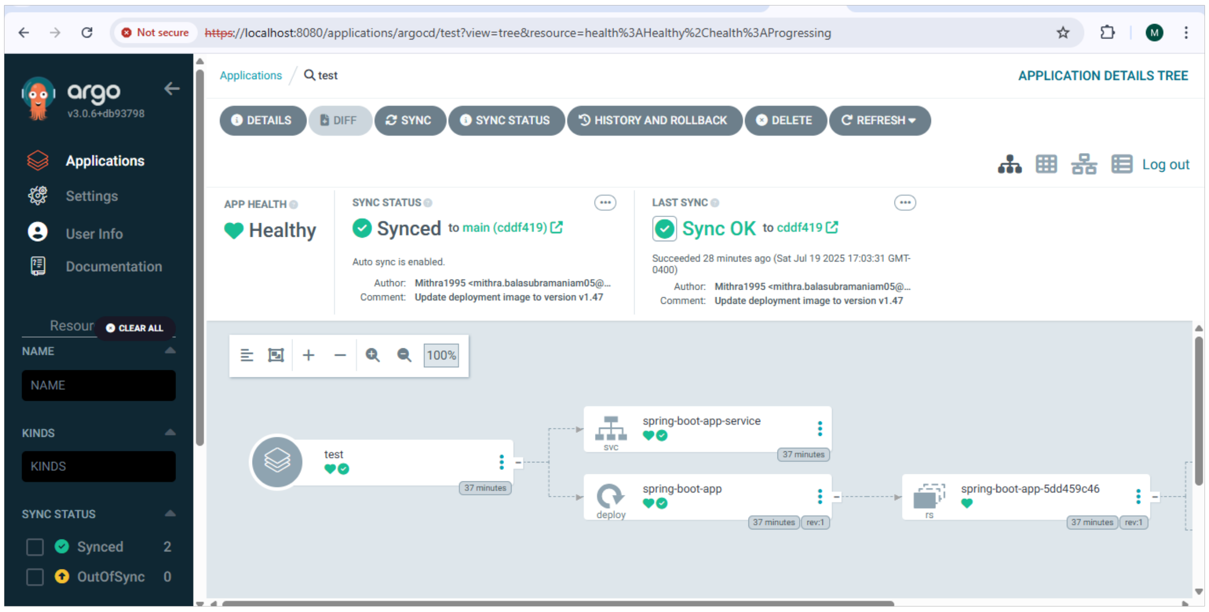
Step 13: Pods are created



```
ubuntu@ip-172-31-33-21:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
spring-boot-app-5dd459c46-49tx8    1/1     Running   0          11m
spring-boot-app-5dd459c46-b695v    1/1     Running   0          11m
ubuntu@ip-172-31-33-21:~$
```



```
ubuntu@ip-172-31-33-21:~$ kubectl get pods -n argocd
NAME                                                READY   STATUS    RESTARTS      AGE
argocd-application-controller-0                     1/1     Running   0             50m
argocd-applicationset-controller-655cc58ff8-kfm8z   1/1     Running   0             50m
argocd-dex-server-7d9dfb4fb8-vpd2v                  1/1     Running   1 (50m ago)   50m
argocd-notifications-controller-6c6848bc4c-nrc9w    1/1     Running   0             50m
argocd-redis-656c79549c-7rjmj                       1/1     Running   0             50m
argocd-repo-server-856b768fd9-6zggn                 1/1     Running   0             50m
argocd-server-99c485944-8rc4r                       1/1     Running   0             50m
ubuntu@ip-172-31-33-21:~$
```

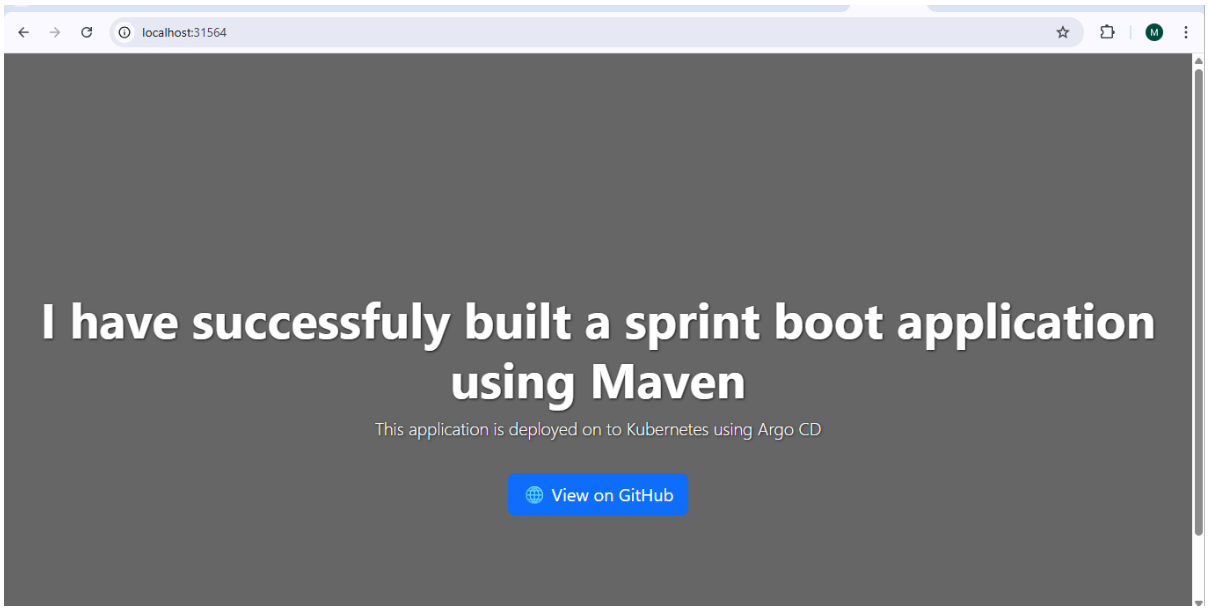


APP HEALTH
Healthy

SYNC STATUS
Synced to main (cddf419)
Auto sync is enabled.
Author: Mithra1995 <mithra.balasubramaniam05@...
Comment: Update deployment image to version v1.47

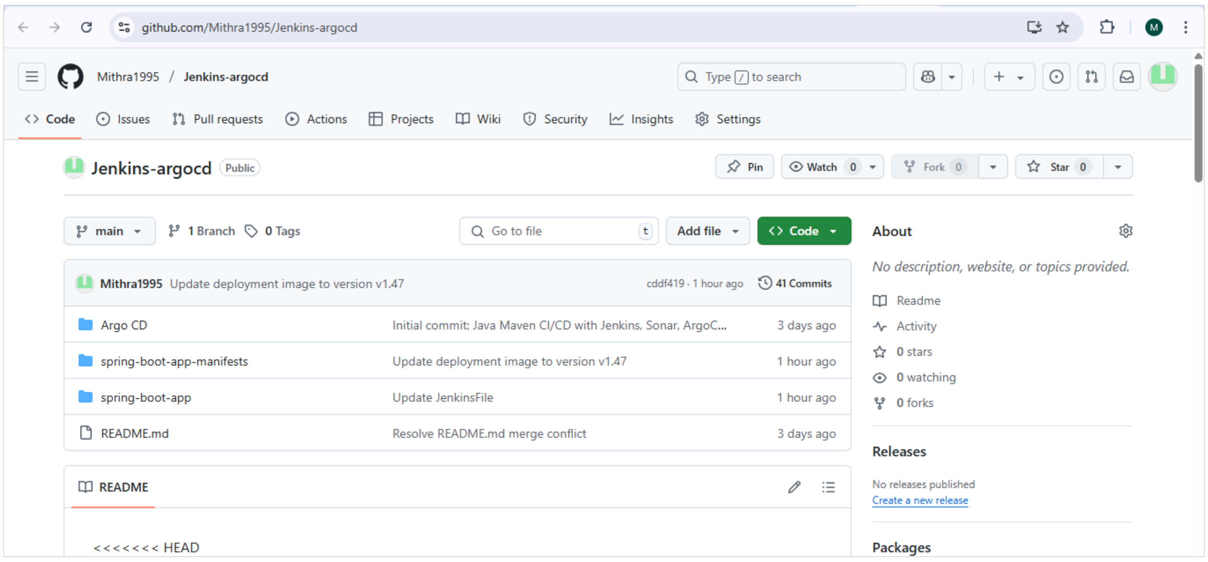
LAST SYNC
Sync OK to cddf419
Succeeded 28 minutes ago (Sat Jul 19 2025 17:03:31 GMT-0400)
Author: Mithra1995 <mithra.balasubramaniam05@...
Comment: Update deployment image to version v1.47

Step 14: check the nodeport ip with 31564 port number



I have successfuly built a sprint boot application using Maven

This application is deployed on to Kubernetes using Argo CD



github.com/Mithra1995/Jenkins-argocd

Mithra1995 / Jenkins-argocd

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Jenkins-argocd Public

main 1 Branch 0 Tags

Mithra1995 Update deployment image to version v1.47	cddf419 · 1 hour ago	41 Commits
Argo CD	Initial commit: Java Maven CI/CD with Jenkins, Sonar, ArgoC...	3 days ago
spring-boot-app-manifests	Update deployment image to version v1.47	1 hour ago
spring-boot-app	Update JenkinsFile	1 hour ago
README.md	Resolve README.md merge conflict	3 days ago

README

<<<<<<< HEAD

About

No description, website, or topics provided.

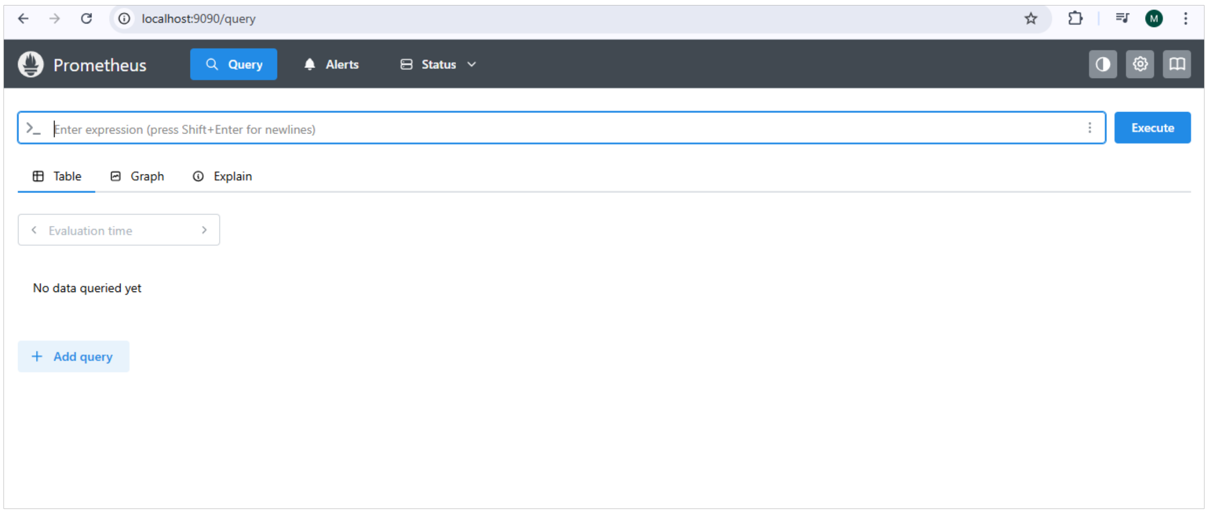
Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

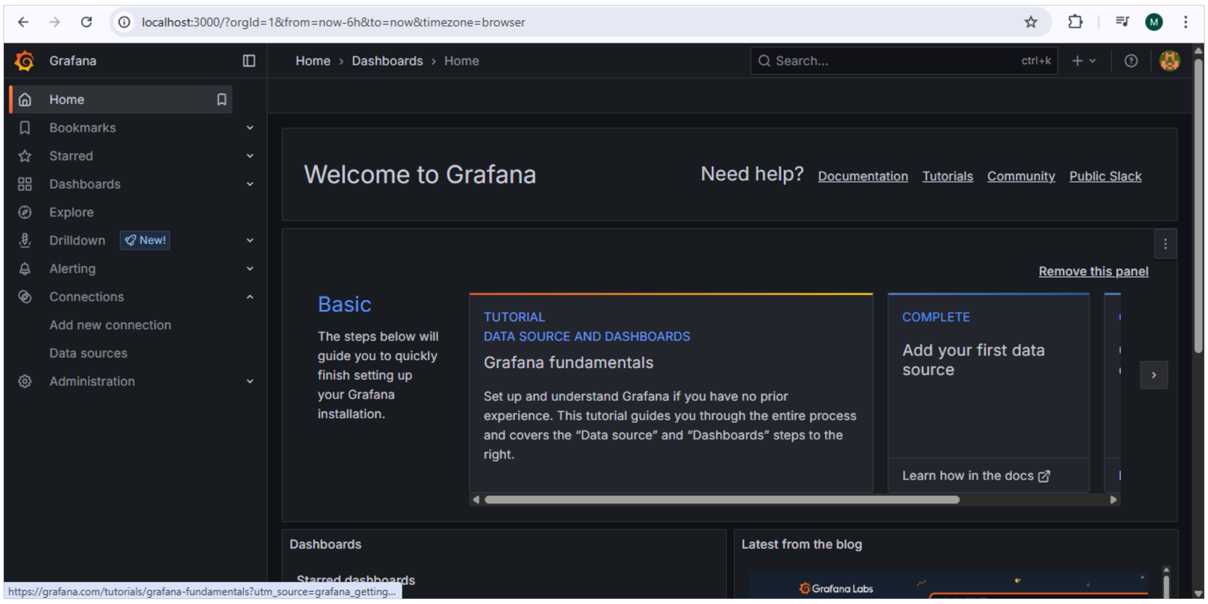
Step 15: Install Prometheus for monitoring



localhost:9090/query

No data queried yet

Step 16: Install Grafana for dashboard for metrics collections


localhost:3000/?orgId=1&from=now-6h&to=now&timezone=browser

Home › Dashboards › Home

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE
Add your first data source

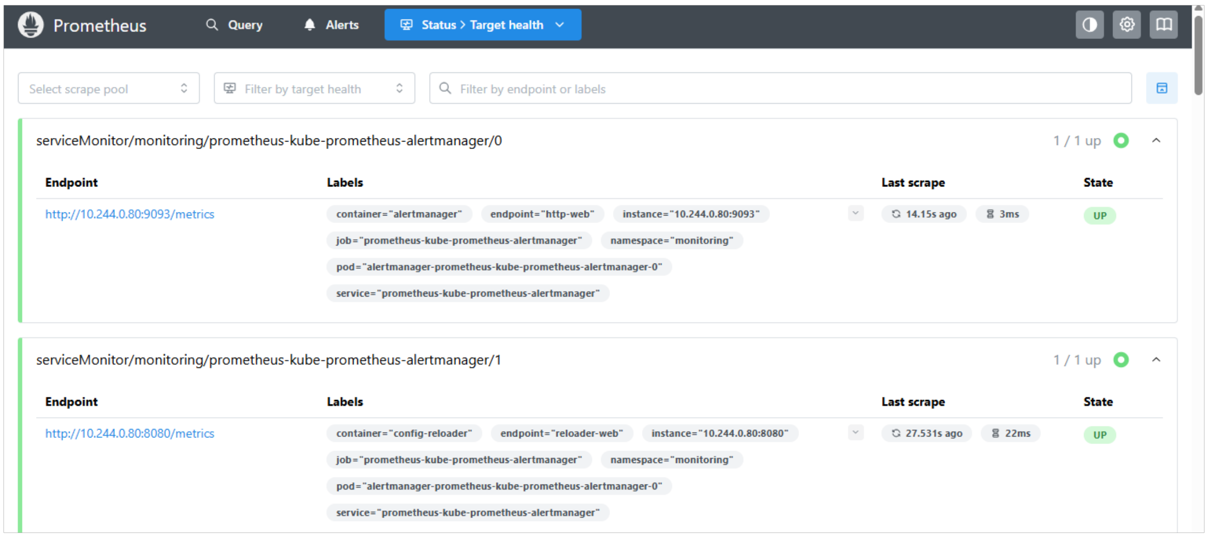
Learn how in the docs

Dashboards

Starred dashboards

Latest from the blog

https://grafana.com/tutorials/grafana-fundamentals?utm_source=grafana_getting...

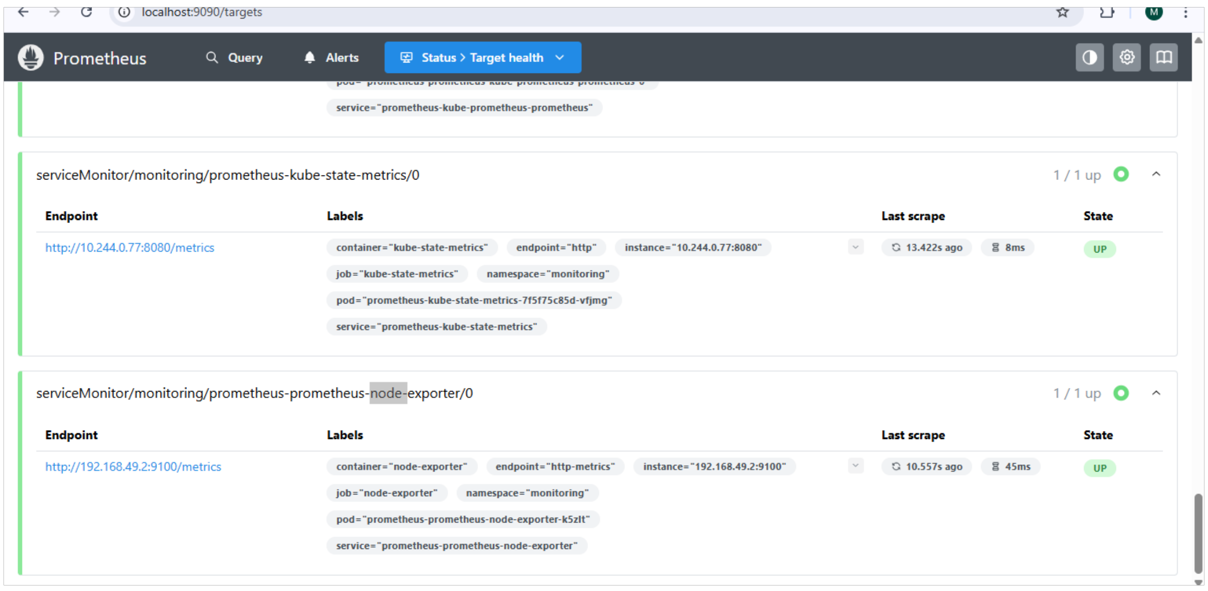


serviceMonitor/monitoring/prometheus-kube-prometheus-alertmanager/0 — 1 / 1 up

Endpoint	Labels	Last scrape	State
http://10.244.0.80:9093/metrics	container="alertmanager" endpoint="http-web" instance="10.244.0.80:9093" job="prometheus-kube-prometheus-alertmanager" namespace="monitoring" pod="alertmanager-prometheus-kube-prometheus-alertmanager-0" service="prometheus-kube-prometheus-alertmanager"	14.15s ago 3ms	UP

serviceMonitor/monitoring/prometheus-kube-prometheus-alertmanager/1 — 1 / 1 up

Endpoint	Labels	Last scrape	State
http://10.244.0.80:8080/metrics	container="config-reloader" endpoint="reloader-web" instance="10.244.0.80:8080" job="prometheus-kube-prometheus-alertmanager" namespace="monitoring" pod="alertmanager-prometheus-kube-prometheus-alertmanager-0" service="prometheus-kube-prometheus-alertmanager"	27.531s ago 22ms	UP



localhost:9090/targets

Prometheus Query Alerts Status > Target health

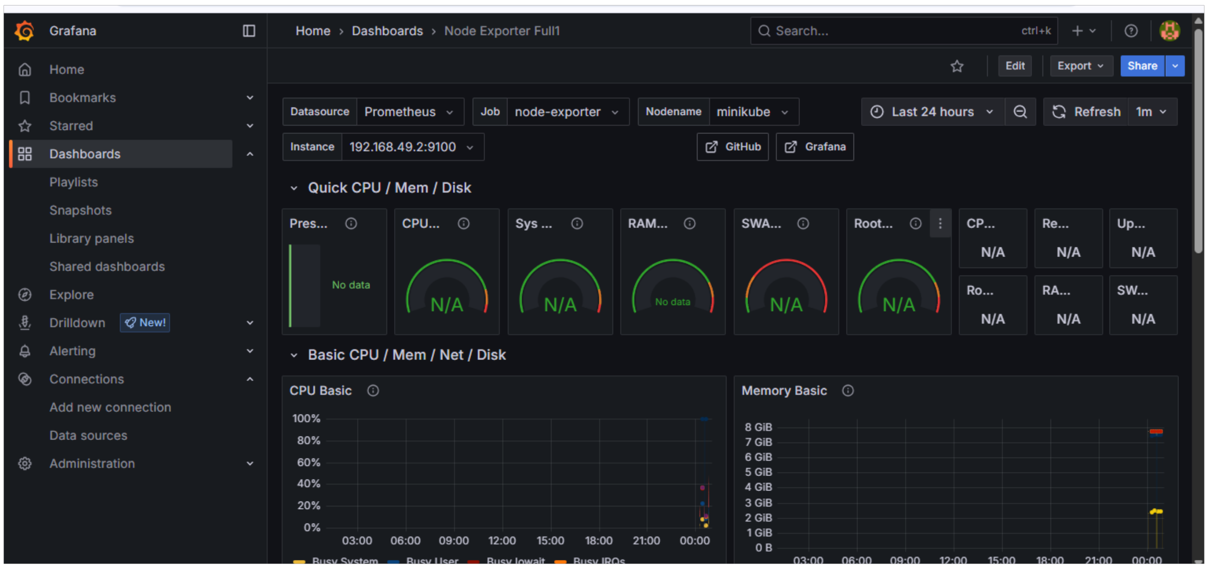
service="prometheus-kube-prometheus-prometheus"

serviceMonitor/monitoring/prometheus-kube-state-metrics/0 — 1 / 1 up

Endpoint	Labels	Last scrape	State
http://10.244.0.77:8080/metrics	container="kube-state-metrics" endpoint="http" instance="10.244.0.77:8080" job="kube-state-metrics" namespace="monitoring" pod="prometheus-kube-state-metrics-7f5f75c85d-vfjmg" service="prometheus-kube-state-metrics"	13.422s ago 8ms	UP

serviceMonitor/monitoring/prometheus-prometheus-node-exporter/0 — 1 / 1 up

Endpoint	Labels	Last scrape	State
http://192.168.49.2:9100/metrics	container="node-exporter" endpoint="http-metrics" instance="192.168.49.2:9100" job="node-exporter" namespace="monitoring" pod="prometheus-prometheus-node-exporter-k5zlt" service="prometheus-prometheus-node-exporter"	10.557s ago 45ms	UP



Home › Dashboards › Node Exporter Full1

Datasource: Prometheus | Job: node-exporter | Nodename: minikube | Instance: 192.168.49.2:9100 | Last 24 hours | Refresh 1m