

## Managing Security Across Multiple Environments with DevSecOps

### PHASE 3- SOLUTION DEVELOPMENT AND TESTING

College Name: Shetty Institute of Technology Kalaburagi.

#### Group Members:

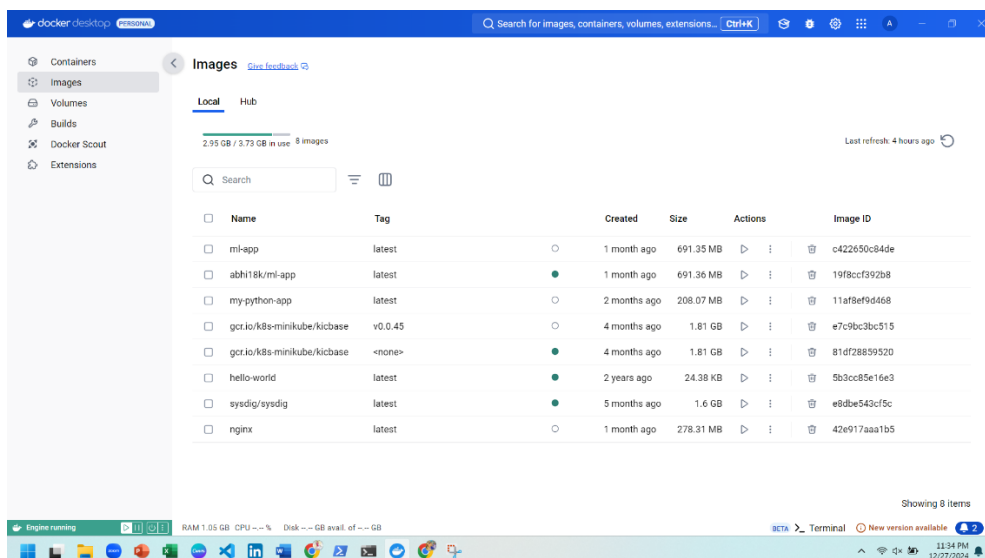
- Name: ABHISHEK S M**  
**CAN ID Number: CAN\_33981242**  
**WORK: SOLUTION DEVELOPMENT**
- Name: BHAGYASHREE M HANDI**  
**CAN ID Number: CAN\_33254590**
- Name: SHARANU**  
**CAN ID Number: CAN\_33489513**
- Name: SHASHIKIRAN**  
**CAN ID Number: CAN\_33287479**

## SOLUTION DEVELOPMENT:

### Implementing Containerization and Running Locally

#### Step 1: Set Up the Development Environment

1. Install Docker for containerization.



DEVOPS ENGINEER

## PHASE 3

### 2. Install Kubernetes (Minikube) for local container orchestration.

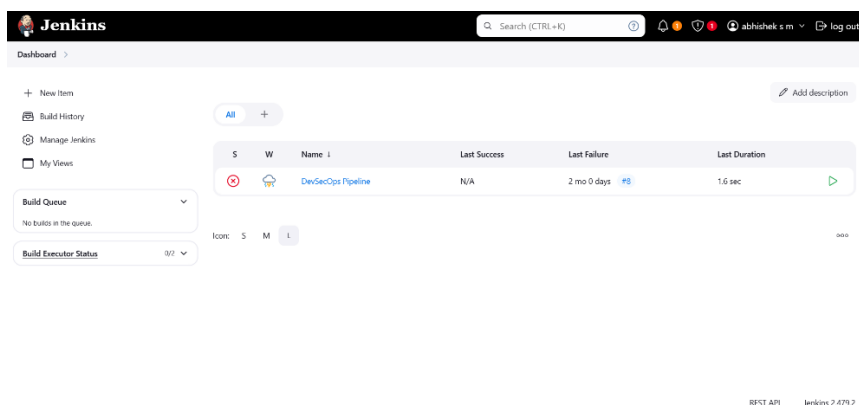
```
C:\Users\abhi\Desktop\Devops\ml-app>minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.26100.2314 Build 26100.2314
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Updating the running docker "minikube" container ...
! Failing to connect to https://registry.k8s.io/ from both inside the minikube container and host machine
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
  - Using image docker.io/kubernetes/metrics-scrapers:v1.0.8
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
  - Using image docker.io/kubernetes/dashboard:v2.7.0
* Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

* Enabled addons: default-storageclass, storage-provisioner, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

```
C:\Users\abhi\Desktop\Devops\ml-app>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

### 3. Install Jenkins for CI/CD pipeline automation.



### 4. Install Git for version control and repository management.

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abhi>git --version
git version 2.47.1.windows.1

C:\Users\abhi>|
```

### 5. Install SonarQube, Trivy, OWASP ZAP for security scanning and vulnerability

## PHASE 3

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\labbi> cd C:\Tools\Trivy
PS C:\Tools\Trivy> trivy.exe --version
Version: 0.58.1
PS C:\Tools\Trivy> trivy image myimage:latest
2024-12-27T21:10:12+05:30 INFO [vuln] Need to update DB
2024-12-27T21:10:12+05:30 INFO [vuln] Downloading vulnerability DB...
2024-12-27T21:10:12+05:30 INFO [vuln] Downloading artifact... repo="mirror.gcr.io/aquasec/trivy-db:2"
37.36 MiB / 37.36 MiB [
2024-12-27T21:12:09+05:30 INFO [vuln] Artifact successfully downloaded repo="mirror.gcr.io/aquasec/trivy-db:2"
2024-12-27T21:12:09+05:30 INFO [vuln] Vulnerability scanning is enabled
2024-12-27T21:12:09+05:30 INFO [secret] Secret scanning is enabled
2024-12-27T21:12:09+05:30 INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-12-27T21:12:09+05:30 INFO [secret] Please see also https://aquasecurity.github.io/trivy/v0.58/docs/scanner/secret#recommendation for faster se
cret detection
2024-12-27T21:12:14+05:30 FATAL Fatal error: image scan error: scan error: unable to initialize a scanner: unable to initialize an image scanner:
unable to find the specified image "myimage:latest" in ["docker" "containerd" "podman" "remote"]: 4 errors occurred:
* docker error: unable to inspect the image (myimage:latest): Error response from daemon: No such image: myimage:latest
* containerd error: containerd socket not found: /run/containerd/containerd.sock
* podman error: unable to initialize Podman client: no podman socket found: CreateFile podman\podman.sock: The system cannot find the path specified
* remote error: GET https://index.docker.io/v2/library/myimage/manifests/latest: UNAUTHORIZED: authentication required; [map(Action:pull Class: Name
library/myimage Type:repository)]

PS C:\Tools\Trivy> docker pull nginx:latest
latest: Pulling from library/nginx
1e189d2a407: Download complete
c4b47389a1: Download complete
2b99b9c5d9e5: Download complete
3d6e42e6c1c: Download complete
f0674088f88: Download complete
da8c133ff82: Download complete
4d98b7087f5: Download complete
Digest: sha256:42e917aa1b3bb0dd0f6f7f4f857099ac7747d7ef73b391c774a41a8b994f15
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
Command Prompt - vault ser x + v
-12-16714:00:53Z
2024-12-27T21:38:37.759+0530 [INFO] core: post-unseal setup complete
2024-12-27T21:38:37.759+0530 [INFO] core: root token generated
2024-12-27T21:38:37.759+0530 [INFO] core: pre-seal teardown starting
2024-12-27T21:38:37.759+0530 [INFO] rollback: stopping rollback manager
2024-12-27T21:38:37.759+0530 [INFO] core: pre-seal teardown complete
2024-12-27T21:38:37.760+0530 [INFO] core: cluster-listener: tcp: starting listener: listener_address=127.0.0.1:8201
2024-12-27T21:38:37.760+0530 [INFO] core: cluster-listener: serving cluster requests, cluster_listen_address=127.0.0.1:8201
2024-12-27T21:38:37.760+0530 [INFO] core: post-unseal setup starting
2024-12-27T21:38:37.760+0530 [INFO] core: loaded wrapping token key
2024-12-27T21:38:37.760+0530 [INFO] core: successfully setup plugin runtime catalog
2024-12-27T21:38:37.761+0530 [INFO] core: successfully setup plugin catalog: plugin-directory=""
2024-12-27T21:38:37.761+0530 [INFO] core: successfully mounted: typesystem version="v1.18.3" builtin.vault path=sys/ namespace="ID: root. Path: "
2024-12-27T21:38:37.762+0530 [INFO] core: successfully mounted: typeidentity version="v1.18.3" builtin.vault path=identity/ namespace="ID: root. Path: "
2024-12-27T21:38:37.762+0530 [INFO] core: successfully mounted: typecubbyhole version="v1.18.3" builtin.vault path=cubbyhole/ namespace="ID: root. Path: "
2024-12-27T21:38:37.762+0530 [INFO] core: successfully mounted: typetoken version="v1.18.3" builtin.vault path=token/ namespace="ID: root. Path: "
2024-12-27T21:38:37.762+0530 [INFO] rollback: starting the rollback manager with 256 workers
2024-12-27T21:38:37.762+0530 [INFO] rollback: starting rollback manager
2024-12-27T21:38:37.762+0530 [INFO] core: restoring leases
2024-12-27T21:38:37.763+0530 [INFO] expiration: lease restore complete
2024-12-27T21:38:37.763+0530 [INFO] identity: entities restored
2024-12-27T21:38:37.763+0530 [INFO] identity: groups restored
2024-12-27T21:38:37.763+0530 [INFO] core: post-unseal setup complete
2024-12-27T21:38:37.763+0530 [INFO] core: vault is unsealed
2024-12-27T21:38:37.769+0530 [INFO] core: successful mount: namespace="" path=secret/ type=kv version="v0.20.0" builtin"
WARNING: dev mode is enabled! In this mode, Vault runs entirely in-memory and starts unsealed with a single unseal key. The root token is already authenticated to the CLI, so you can immediately begin using Vault.

You may need to set the following environment variables:

PowerShell:
$env:VAULT_ADDR="http://127.0.0.1:8208"
cmd.exe:
set VAULT_ADDR=http://127.0.0.1:8208

The unseal key and root token are displayed below in case you want to seal/unseal the Vault or re-authenticate.

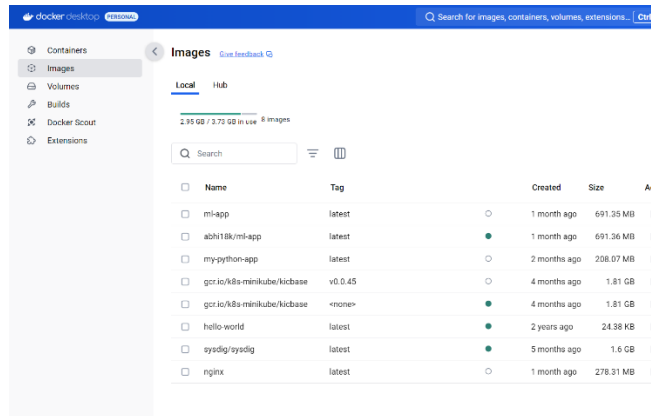
Unseal Key: ayuEAYhiy2XxuvxeMIAgrWkdyzIpNJti12e9JncH73I=
Root Token: hvs.ng5SLpxmMABp4Hx3MsV9cJf
```

## Step 2: Containerizing the Application

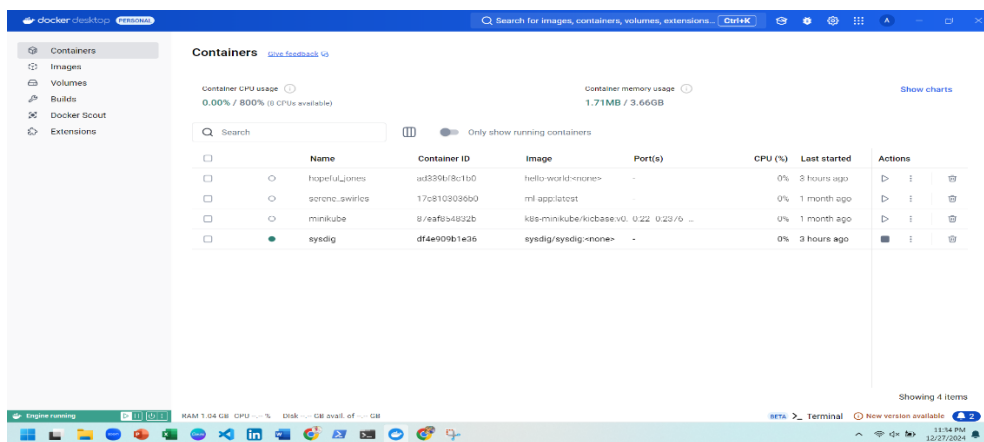
### 1. Create a Docker file for the application:

- Define the base image (e.g., node:16-alpine).
- Set up the working directory and dependencies.
- Expose required ports.
- Define the startup command (CMD).

## PHASE 3

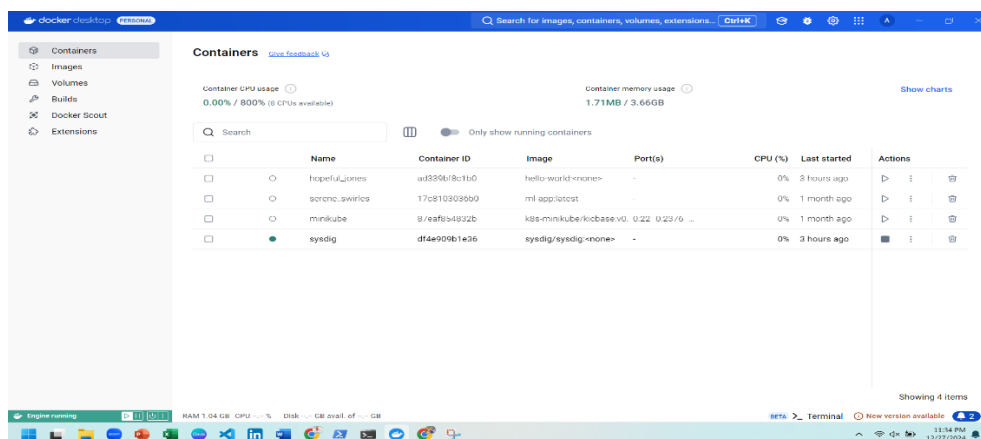


### 2. Build and run the container locally:



### 3. docker build -t ml-app .

docker run -d -p 8080:8080 ml-app



## Step 3: Set Up Container Security

### 1. Scan container images using Trivy or Snyk.

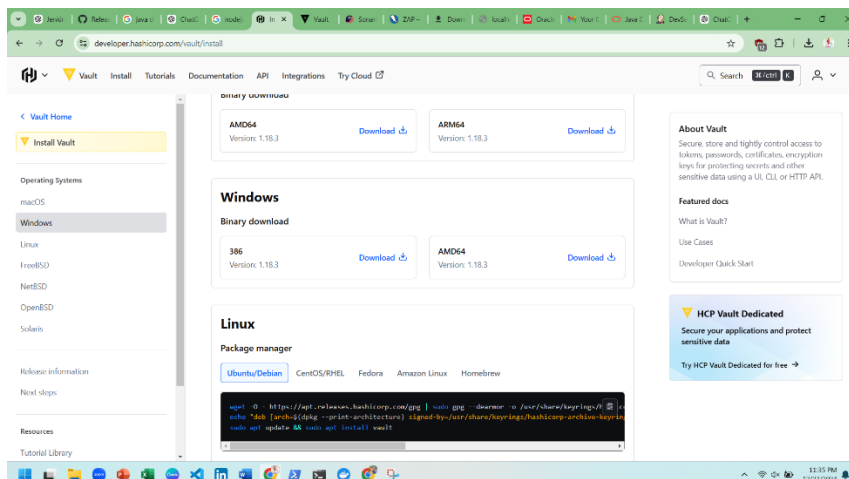
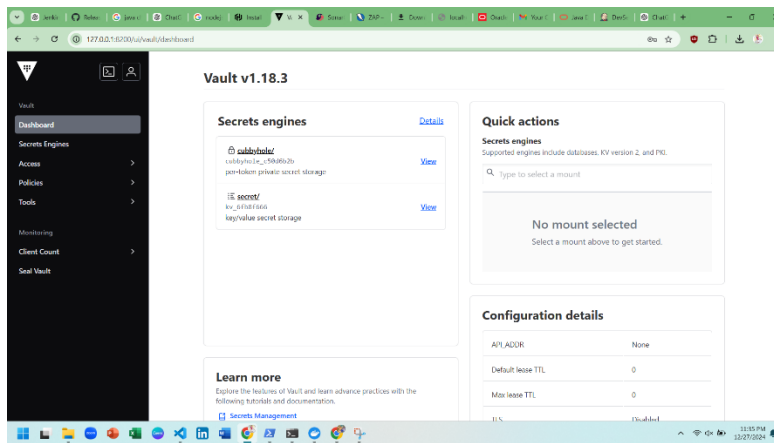
## PHASE 3

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\abhi> cd C:\Tools\Trivy
PS C:\Tools\Trivy> .\trivy.exe --version
Version: 0.58.1
PS C:\Tools\Trivy> trivy image myimage:latest
2024-12-27T12:10:12+05:30 [vulndb] Need to update DB...
2024-12-27T12:10:12+05:30 [vulndb] Downloading vulnerability DB...
2024-12-27T12:10:12+05:30 [vulndb] Downloading artifact... repo="mirror.gcr.io/aquasec/trivy-db:2"
57.96 MiB / 57.96 MiB [-----] 100.00% 516.71 KiB p/s 1m55s
2024-12-27T12:11:09+05:30 [vulndb] Artifact successfully downloaded
2024-12-27T12:12:09+05:30 [vulndb] Vulnerability scanning is enabled
2024-12-27T12:12:09+05:30 [vulndb] Secret scanning is enabled
2024-12-27T12:12:09+05:30 [secret] If your scanning is slow, please try "--scanners vuln" to disable secret scanning
2024-12-27T12:12:09+05:30 [secret] Please see also https://aquasecurity.github.io/trivy/v0.58/docs/scanner/secret#recommendation for faster se
cret detection
2024-12-27T12:12:14+05:30 [FATAL] Fatal error: image scan error: scan error: unable to initialize a scanner: unable to initialize an image scanner:
unable to find the specified image "myimage:latest" in ["docker" "containerd" "podman" "remote"]: 4 errors occurred:
+ docker error: unable to inspect the image (myimage:latest): Error response from daemon: No such image: myimage:latest
+ containerd error: containerd socket not found: /run/containerd/containerd.sock
+ podman error: unable to initialize Podman client: no podman socket found: CreateFile podman/podman.sock: The system cannot find the path specified
+ remote error: GET https://index.docker.io/v2/library/myimage/manifests/latest: UNAUTHORIZED: authentication required; [map[Action:pull Class: Name
:library/myimage Type:repository]]
PS C:\Tools\Trivy> docker pull nginx:latest
latest: Pulling from library/nginx
1e2094d2a807: Download complete
c18f27389ea1: Download complete
29d90ac5d8e5: Download complete
566e42bce1c1: Download complete
46b746d5f4f4: Download complete
da8cc133ff82: Download complete
bf86b74871f5: Download complete
Digest: sha256:42c137aaab35a8a8d8f6f74f4837498ac7747e7473b391c774a41a8b99415
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

2. Implement Kubernetes RBAC policies and network security rules.
3. Secure secrets using HashiCorp Vault.



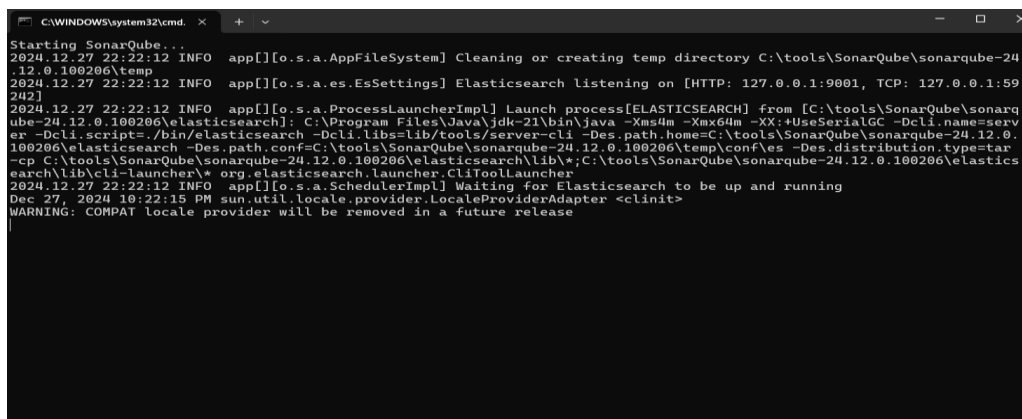
#### Step 4: Implement CI/CD Security

1. Set up Jenkins for automating CI/CD pipeline.
2. Integrate SonarQube for Static Application Security Testing (SAST).
3. Automate security scanning in the CI/CD pipeline using OWASP ZAP.
4. Deploy to a Kubernetes cluster using Minikube.

## SECTION 2: TESTING THE SOLUTION

#### Step 1: Static Code Analysis (SAST)

- Run **SonarQube** to detect security vulnerabilities in the source code.
- Fix issues before progressing to the build stage.

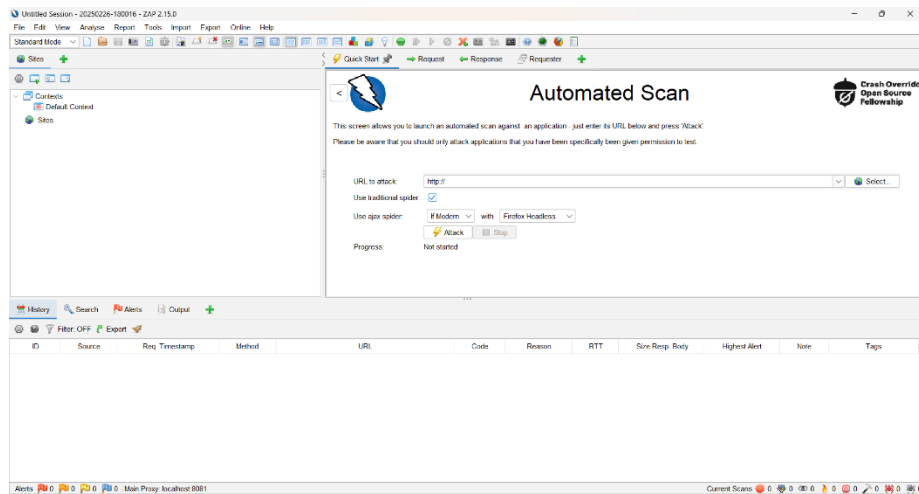


```
C:\WINDOWS\system32\cmd. x + ~
Starting SonarQube...
2024.12.27 22:22:12 INFO app[][o.s.a.AppFileSystem] Cleaning or creating temp directory C:\tools\SonarQube\sonarqube-24
.12.0.100206\temp
2024.12.27 22:22:12 INFO app[][o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:59
242]
2024.12.27 22:22:12 INFO app[][o.s.a.ProcessLauncherImpl] Launch process[ELASTICSEARCH] from [C:\tools\SonarQube\sonarq
ube-24.12.0.100206\elasticsearch]: C:\Program Files\Java\jdk-21\bin\java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=serv
er -Dcli.script=/bin/elasticsearch -Dcli.libs=lib/tools/server-cli -Des.path.home=C:\tools\SonarQube\sonarqube-24.12.0.
100206\elasticsearch -Des.path.conf=C:\tools\SonarQube\sonarqube-24.12.0.100206\temp\conf\es -Des.distribution.type=tar
-cp C:\tools\SonarQube\sonarqube-24.12.0.100206\elasticsearch\lib*;C:\tools\SonarQube\sonarqube-24.12.0.100206\elastics
earch\lib\cli-launcher*.org.elasticsearch.launcher.CliToolLauncher
2024.12.27 22:22:12 INFO app[][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
Dec 27, 2024 10:22:15 PM sun.util.locale.provider.LocaleProviderAdapter <clinit>
WARNING: COMPAT locale provider will be removed in a future release
```

#### Step 2: Dynamic Application Security Testing (DAST)

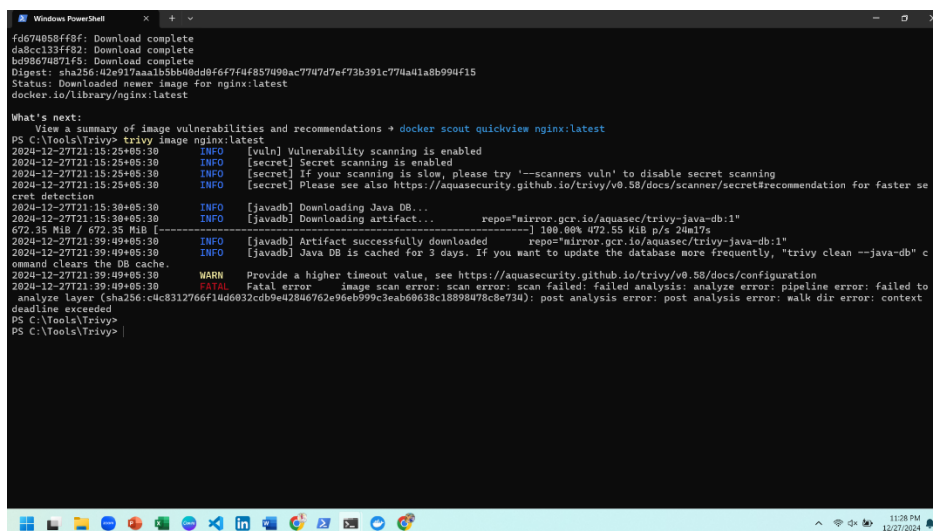
- Use **OWASP ZAP** to test for SQL Injection, XSS, and CSRF vulnerabilities.
- Run penetration tests against staging environments.

## PHASE 3



### Step 3: Container Security Testing

- Scan container images using **Trivy** or **Snyk**.
- Enforce least privilege access policies.



### Step 4: Infrastructure and Compliance Testing

- Scan Kubernetes configurations with **Checkov** to detect misconfigurations.
- Verify compliance with **GDPR**, **PCI-DSS**, **SOC2** security policies.

### Step 5: CI/CD Pipeline Security Validation

- Ensure security policies are enforced in the **Jenkins pipeline**.
- Monitor deployments for unauthorized changes using **ELK Stack**.

## **FUTURE IMPROVEMENTS**

1. **Enhance Security Automation:** Implement real-time security monitoring and automated patch management.
2. **Adopt Zero Trust Architecture:** Introduce strict authentication and continuous security verification.
3. **Implement Advanced Threat Detection:** Use AI-powered threat detection tools for anomaly monitoring.
4. **Optimize Multi-Cloud Security:** Extend security controls across **AWS, Azure, and Google Cloud.**
5. **Automate Security Audits:** Set up automated penetration testing and compliance validation.