DevSecOps Pipeline - Integration of security tools in CI/CD

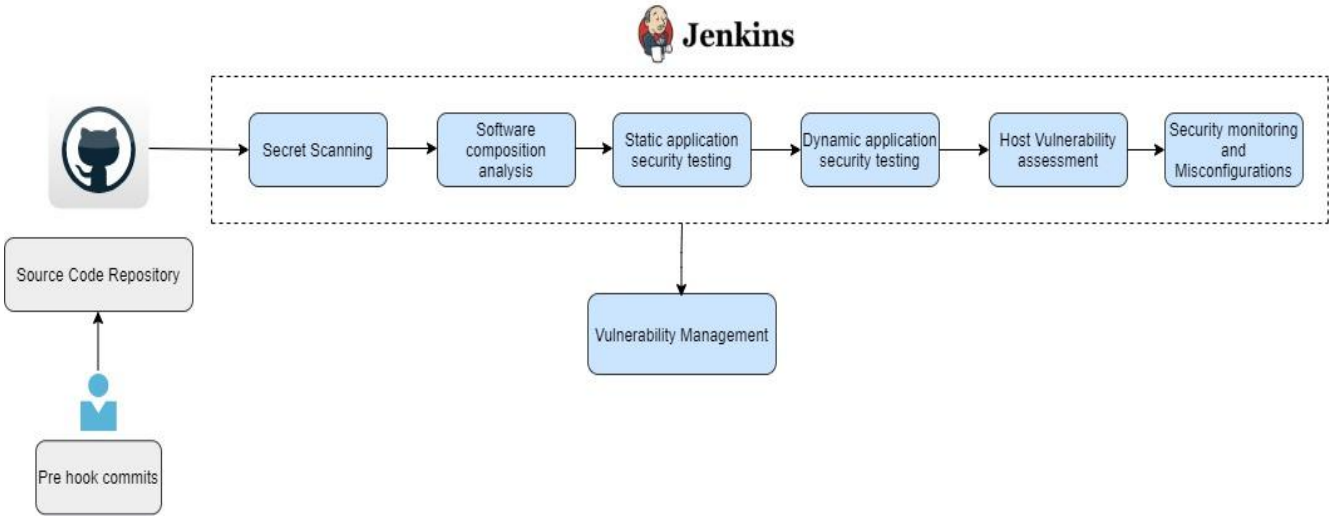
For implementing devsecops pipeline we are using a Java based vulnerable WebGoat application.For setting up the infrastructure on cloud (AWS), we will require 4 ec2 instances (t2.medium)

- Jenkins Server
- SAST, DAST tools
- Application Server
- Vulnerability Management server

and for cloud security monitoring :

- AWS Config service
- AWS Security Hub service
- AWS GurdDuty
- IAM access analyzer
- AWS Macie

Pipeline Architecture

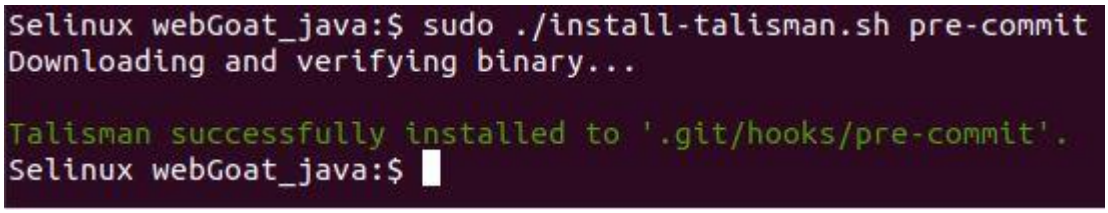


Lab Environment

1) Install Pre-commit or Pre-push hooks on developer workstation.

   Talisman is a tool that installs a hook to a repository to ensure that potential secrets or sensitive information do not leave the developer's workstation.

   Install talisman as a pre-commit hook



   It validates the outgoing changeset for things that look suspicious - such as potential SSH keys, authorization tokens, private keys

2) For setting up the jenkins server Launch an ec2-instance (t2.medium)

Pre-requisites

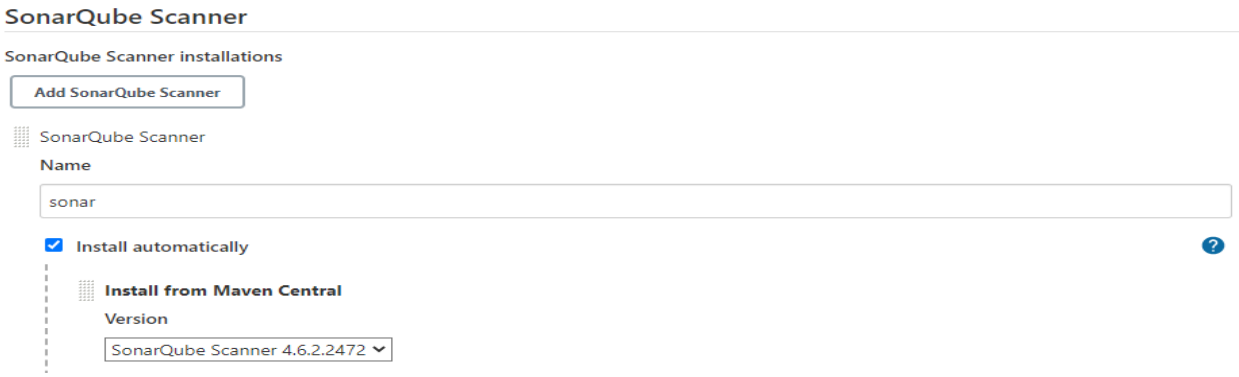
- JDK 8 or 11, Maven
- Install jenkins on that instance
- Once jenkins is installed , install suggested plugins

After installing suggested plugins install following plugins

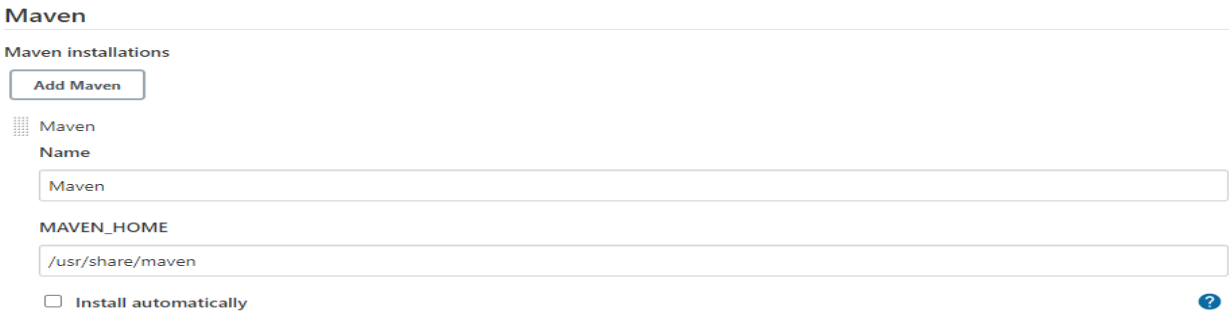
1. Blue Ocean
2. Maven integration
3. SSH agent
4. Sonarscanner
5. OWASP Dependency-Check

Configure maven, sonarqube and owasp dependency check tools in global tool configurations of jenkins(Jenkins -> Manage jenkins -> Global tool configuration)

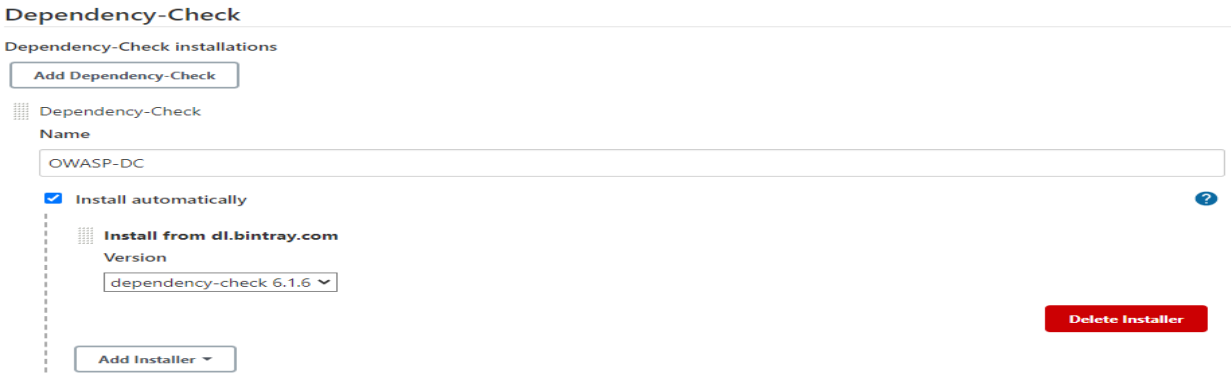
- For SonarQube



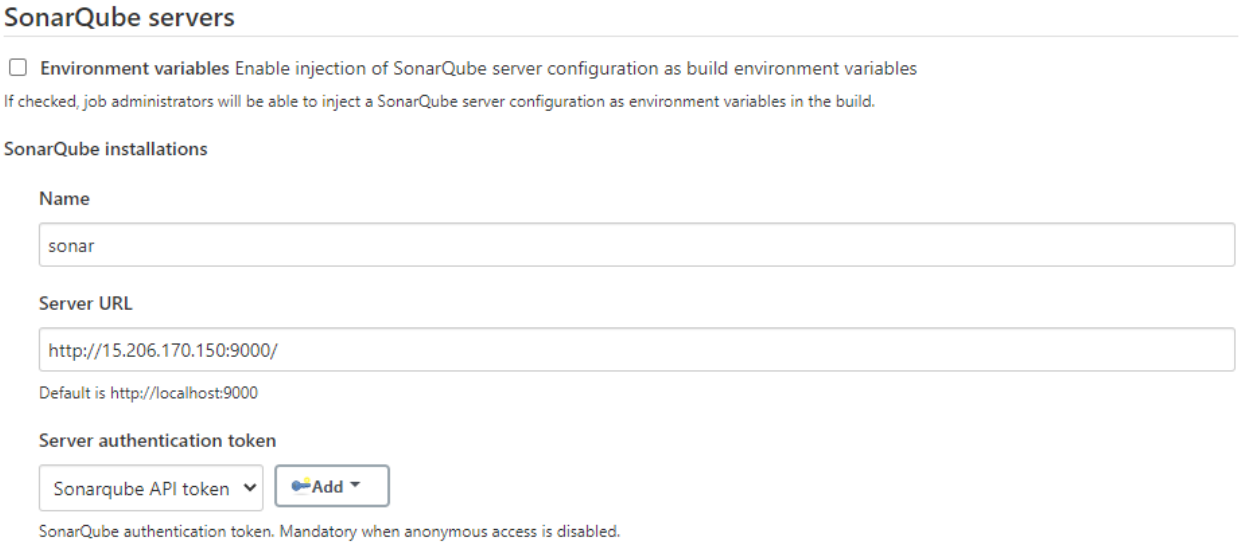
- For Maven



- For OWASP Dependency Check



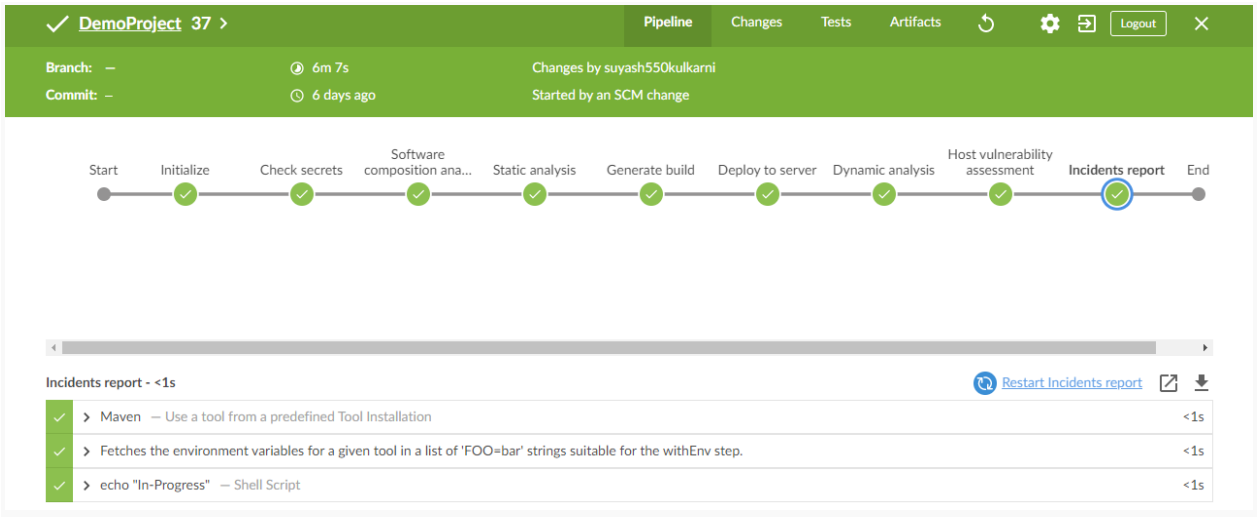
Launch one more ec2-instance(t2.medium) and install sonarqube on that server. After installing sonarqube server configure sonarqube server URL and API key in jenkins (Jenkins -> Manage jenkins ->Configure system)



Install trufflehog tool on jenkins for scanning secret

- `pip install truffleHog3`

Create a pipeline and configure a project URL, Jenkins file path and branch name. After creating a pipeline for every commit pipeline will trigger and will identify security vulnerabilities in each stage. The scan results will be uploaded to the vulnerability management tool.

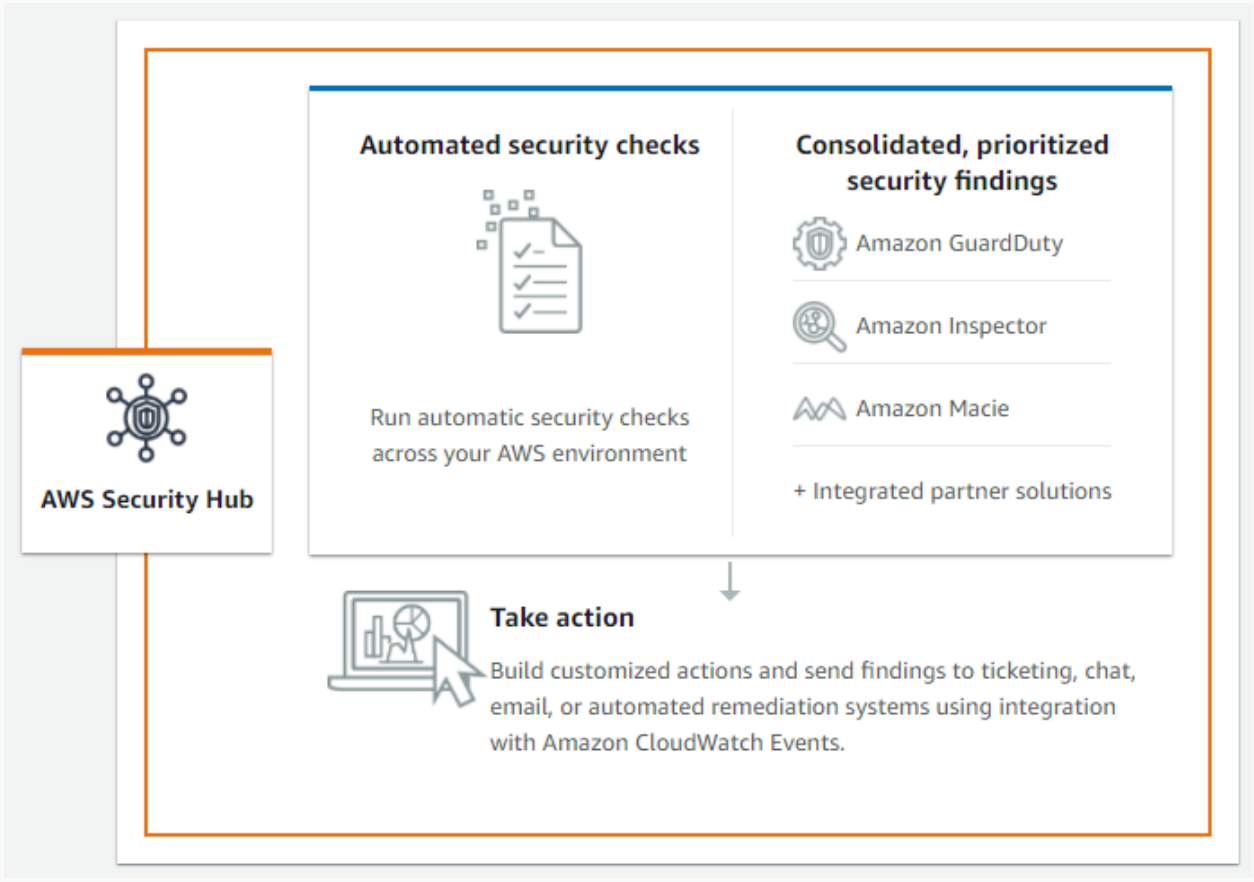

DemoProject 37 >
Pipeline
Changes
Tests
Artifacts
Logout
Branch: –
Commit: –
6m 7s
6 days ago
Changes by suyash550kulkarni
Started by an SCM change
Start
Initialize
Check secrets
Software composition ana...
Static analysis
Generate build
Deploy to server
Dynamic analysis
Host vulnerability assessment
Incidents report
End
Incidents report - <1s
Restart Incidents report
Maven — Use a tool from a predefined Tool Installation
<1s
Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.
<1s
echo "In-Progress" — Shell Script
<1s

3) Security Monitoring and Misconfigurations

Prerequisites

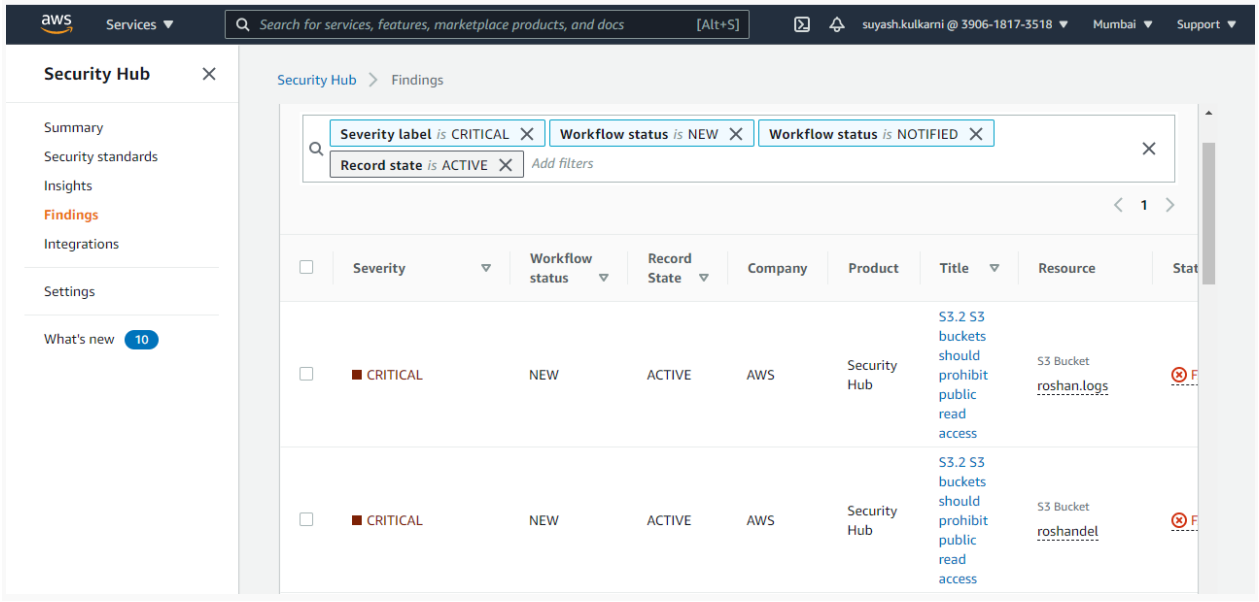
- Enable AWS config and global services
- Enable GuardDuty, Macie
- Create IAM access analyzer

AWS Security Hub gives you a comprehensive view of your security alerts and security posture across your AWS accounts. We can consolidate different accounts in AWS Security Hub



Findings for infrastructure security misconfigurations are based on the following standards:

- CIS AWS Foundations Benchmark v1.2.0
- AWS Foundational Security Best Practices v1.0.0
- PCI DSS v3.2.1

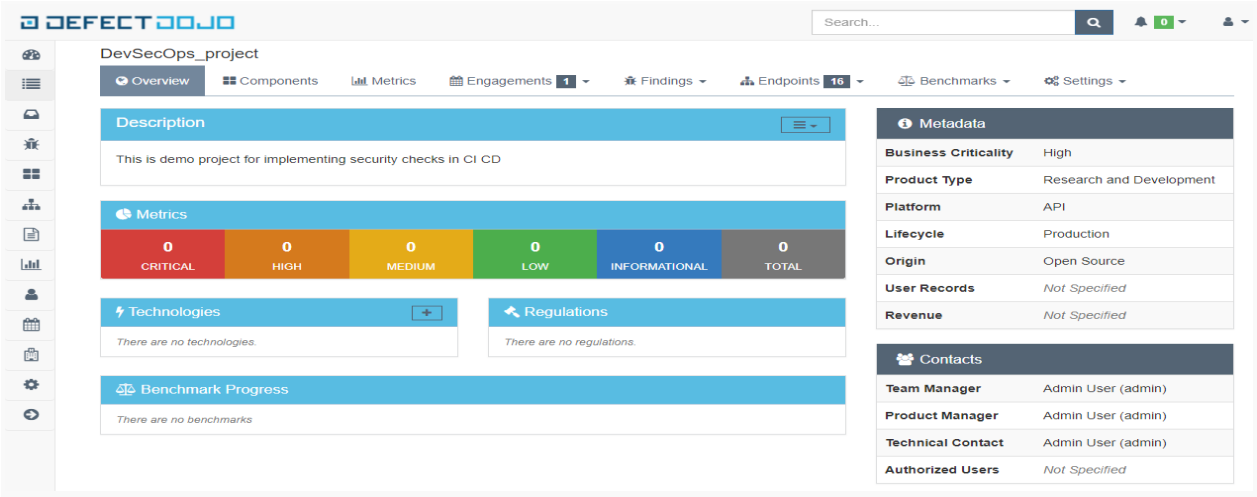

aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
suyash.kulkarni @ 3906-1817-3518
Mumbai
Support
Security Hub
Summary
Security standards
Insights
Findings
Integrations
Settings
What's new
10
Security Hub > Findings
Severity label is CRITICAL
Workflow status is NEW
Workflow status is NOTIFIED
Record state is ACTIVE
Add filters
1
Severity
Workflow status
Record State
Company
Product
Title
Resource
CRITICAL
NEW
ACTIVE
AWS
Security Hub
S3.2 S3 buckets should prohibit public read access
S3 Bucket
roshan.logs
CRITICAL
NEW
ACTIVE
AWS
Security Hub
S3.2 S3 buckets should prohibit public read access
S3 Bucket
roshandel

4) Vulnerability management server

Launch an ec2 instance and install defectdojo as a vulnerability management tool on the server.

```
git clone https://github.com/DefectDojo/django-DefectDojo
cd django-DefectDojo
# building
docker-compose build
# running
docker-compose up
```

Login to the application and create a product, engagements



To import the result of different security findings we are using defect dojo API's (https://defectdojo.readthedocs.io/en/latest/api-v2-docs.html)

For example, Import ZAP scan report to defectdojo

- ```
  curl -X POST "http://<IP_address>:8080/api/v2/import-scan/" -H "accept: application/json" -H  "Content-Type: multipart/form-data" -H  "X-CSRFToken: vUsaTeXI2m1I94DBRizQBR2dpS68XO4HD70CQx1q5bPxLiGGpylhcIWbiw8uVSfR" -F "scan_date=2021-06-17" -F "minimum_severity=Info" -F "active=true" -F "verified=true" -F "scan_type=ZAP Scan" -F
  ```
  ```

```
"file=@zap_report" -F "engagement=1" -F
"close_old_findings=false" -F "push_to_jira=false"
```

After Importing the results , you can check the findings under engagements