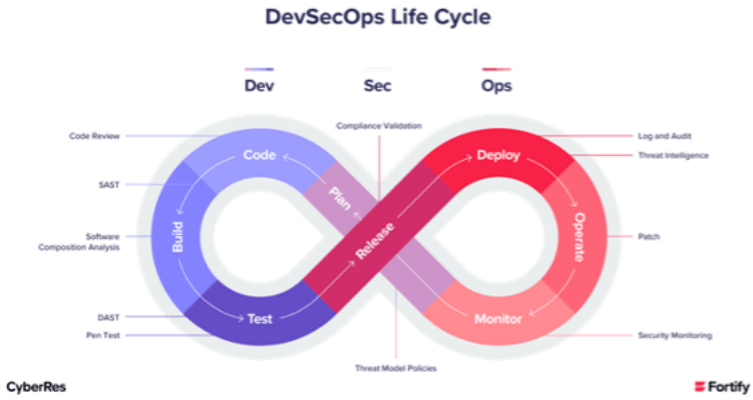
Get unlimited access Open in app

Nandita Sahu Follow
Jul 2 · 7 min read · Listen

Save

# DevSecOps — Implementing Secure CI/CD Pipelines




DevSecOps LifeCycle

### Why DevSecOps?

• DevSecOps brings security closer to IT and business objectives by minimizing vulnerabilities earlier in the application development life cycle.

• Keeping this in mind, our team automated security to secure the broader environment and data, as well as the CI/CD process.

• Integrating security measures with minimal disruption to operations, staying current with technologies like containers and microservices.

- In DevSecOps security is built for containers and microservices.

### What is DevSecOps?

• DevSecOps brings security closer to IT and business objectives by minimizing vulnerabilities earlier in the application development life cycle.

• In a DevSecOps environment, IT professionals/security team works with developers to automate security checks throughout the development cycle.

• The benefit of DevSecOps:

• Enhanced automation throughout the software delivery pipeline which eliminates mistakes and reduces attacks and downtime.

- For teams looking to integrate security into their DevOps framework, the process can be completed seamlessly using the right DevSecOps tools and processes.

### DevOps Vs DevSecOps ??

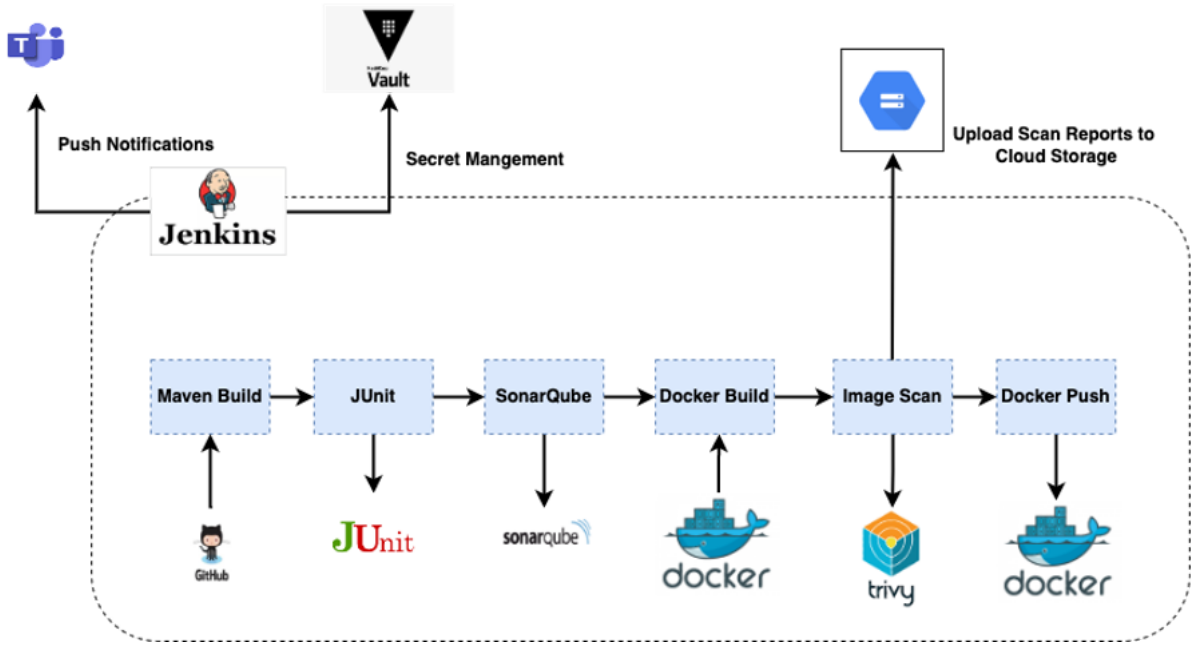
Get unlimited access Open in app

| | |
|---|---|
| DevOps integrates operations into the development/release cycle | DevSecOps integrates security aspect into the development/release cycle. |
| DevOps increases the speed at which software is developed and delivered | DevSecOps increases the security with which software is developed and delivered. |
| DevOps automates much of the software lifecycle. | DevSecOps requires merging and automating many of the traditional practices of security engineers, operations teams, and development teams. |
| The concept of security begins right after the development pipeline | Application security begins during the build process |
| Renews focus on the customers<br>Simplifies development focus<br>Supports end-to-end responsibility | Can spot bugs early on<br>Reduce risk and legal liability<br>Reduce costs on resource management |

### Objectives

- Remove Manual Build and Deploy Process
- Integrate security into our DevOps pipeline
- Integration of secret management tool to secure secrets
- Implement efficient, continuous, automated and secure development and deployment process
  - Integration of Teams for entire product development life cycle
  - Deploy the solution to the public cloud (GCP) upon highlighting all the security vulnerabilities and compliance requirements.

### Architecture Diagram for CI/CD Pipeline




CI/CD Pipeline

Here, we have a taken a simple Maven Project to show a demo.

### Tools Used

Get unlimited access Open in app


the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.


It is a secret management tool specifically designed to control access to sensitive credentials in a low-trust environment. It can be used to store sensitive values and at the same time dynamically generate access for specific services and applications


Building our Maven Application using pom.xml file


It is a unit testing framework for the Java programming language.


It is used for continuous analysis of source code quality by performing analysis on your code to detect duplications, bugs, security vulnerabilities and code smells on programming languages.


It is an open source containerization platform.

Get unlimited access Open in app


It is a simple and comprehensive vulnerability and secret scanner for containers and other artifacts. Trivy detects vulnerabilities of OS packages . It also scans Infrastructure as Code(IAC) files such as Terraform and Kubernetes, to detect potential configuration issues that expose your deployments to the risk of attack. It also scans hard like passwords, API keys and tokens.


It is a service for storing your objects in Google Cloud. An object is an immutable piece of data consisting of a file of any format. You store objects in containers called buckets.


It is a service provided by Docker for finding and sharing container images with your team.

It is a collaboration app built for hybrid work so you and your team stay informed, organized, and connected all in one place.

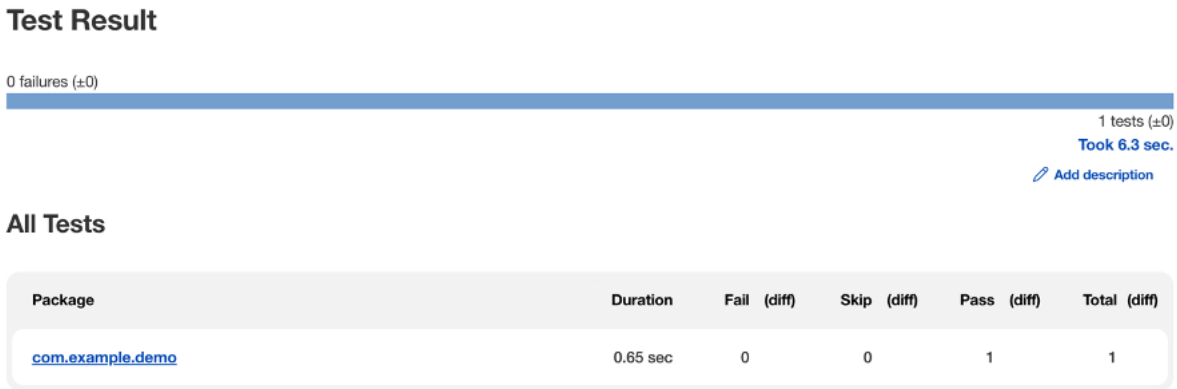
**Jenkinsfile for CI/CD Pipeline**

Get unlimited access Open in app

```
tools {
    maven 'maven-3.8.6'
}
stages {
    stage('Checkout git') {
        steps {
           git branch: 'sonar', url: 'https://github.com/darinpope/java-web-app'
        }
    }

    stage ('Build & JUnit Test') {
        steps {
            sh 'mvn install'
        }
        post {
           success {
                junit 'target/surefire-reports/**/*.xml'
            }

        }
```

Initially, we are cloning git repository from sonar branch into the Jenkins workspace and in the second stage we are building our maven application from pom.xml which is creating an Artifact in Jenkins workspace folder.

In the post success step we are using JUnit for unit testing . The test reports are generated into the Jenkins workspace target folder.

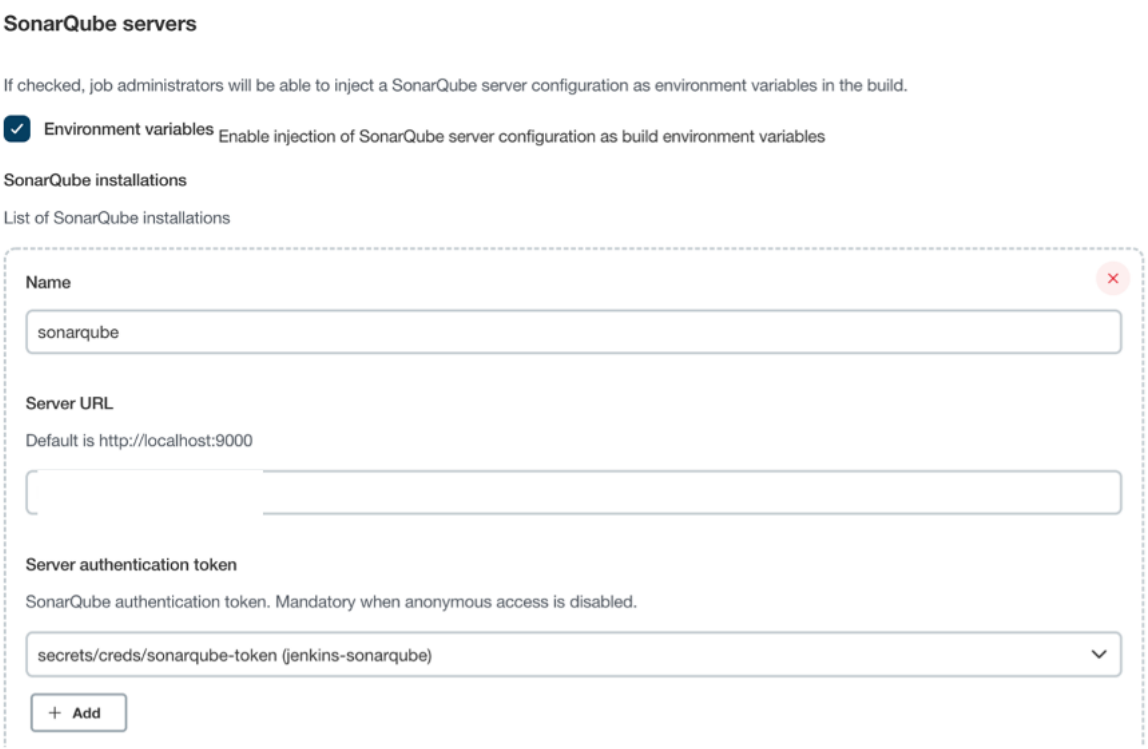



Get unlimited access Open in app

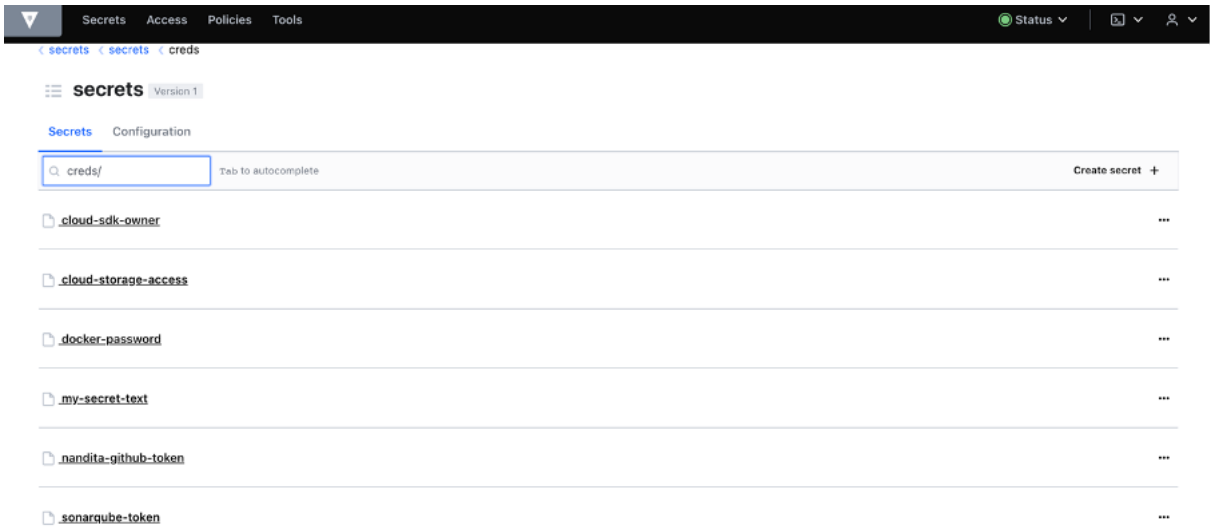
```
                withSonarQubeEnv(installationName: 'sonarqube') {
                    sh 'mvn clean org.sonarsource.scanner.maven:sonar-maven-plugin:3.9.0.2155:sonar'
                }
            }
        }
```

In the next stage we are doing Code Quality Assurance Test using SonarQube. It will analyze the code of maven build and will publish the reports into the SonarQube portal authenticating with the token credentials verified by the HashiCorp Vault. The '**installationName'** parameter is the name which we have used Manage Jenkins (Configure System  Sonarqube servers).

Go to Manage Jenkins  Configure System and '**Name'** and '**Server Url'** and '**Server Authentication Token'** , which token is stored in Vault server folder(secrets/creds/sonarqube-token).



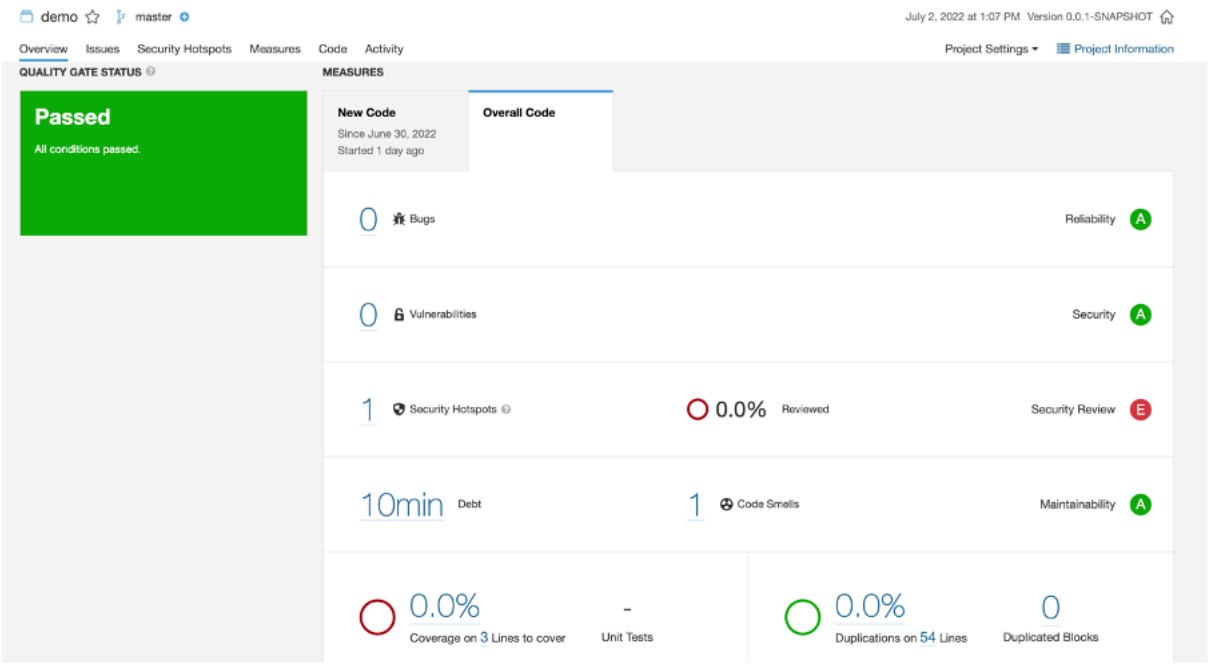

Here, we have used vault as secret management tool to store our secrets(credentials) which are used in CI/CD Pipeline.




To know how we have Integrated Vault with Jenkins, please refer my earlier post :

Get unlimited access Open in app

medium.com




Dashboard of SonarQube

Here, we can see that the code quality is passed, and it can also detect the bug and vulnerabilities present in the code.

```
stage('Building Docker Image'){
    steps{
        sh '''
        sudo docker build -t nanditasahu/devsecops-demo:$BUILD_NUMBER .
        sudo docker images
        '''
    }
}
```

In the next stage we are building Docker Images with image name as nanditasahu/devsecops-demo(Repository name) and tag name as $BUILD_NUMBER which returns the current build number of the job in Jenkins. It is also showing the top-level images, their repository and tags, and their size.

```
+ sudo docker images
REPOSITORY                  TAG                 IMAGE ID       CREATED          SIZE
<none>                      <none>              ae8515c625c0   5 minutes ago    261MB
<none>                      <none>              7ae369ba1abd   27 hours ago     394MB
nanditasahu/devsecops-demo  20                  700235f375c4   2 days ago       278MB
nanditasahu/devsecops-demo  3                   700235f375c4   2 days ago       278MB
adoptopenjdk/openjdk11      alpine-slim         c52a369ce47e   3 days ago       261MB
tomcat                      8.5-jdk17-corretto  e6fa6f079286   10 days ago      493MB
sonarqube                   latest              75c013514322   3 weeks ago      534MB
accurics/terrascan          latest              b2b7efbeaaf5   3 months ago     116MB
```

```
stage('Image Scanning Trivy'){
    steps{
       sh 'sudo trivy image nanditasahu/devsecops-demo:$BUILD_NUMBER > $WORKSPACE/trivy-image-scan/trivy-i

    }
```

Get unlimited access Open in app

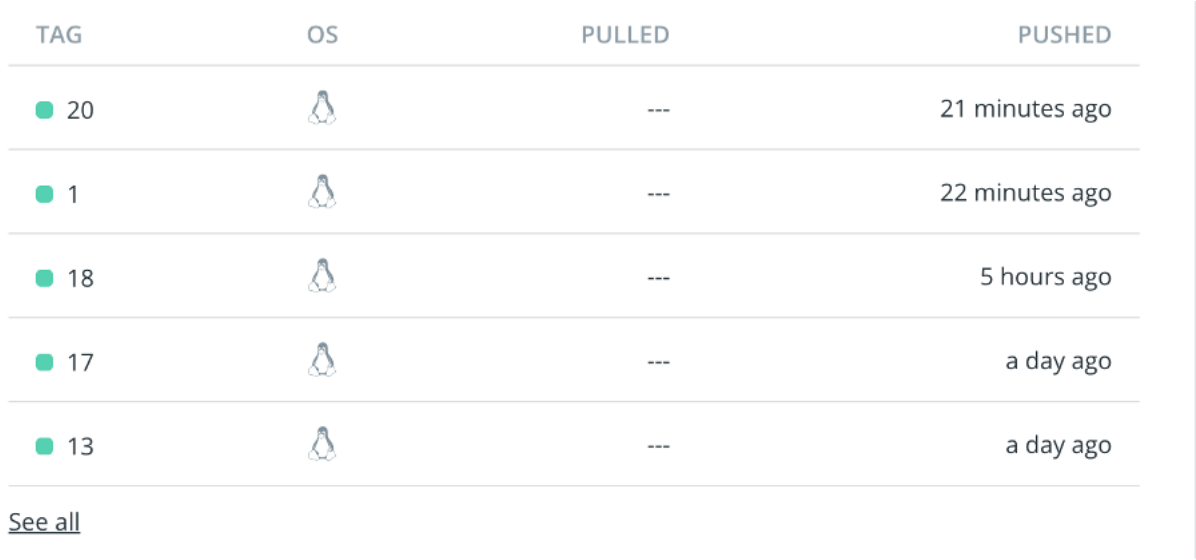
Docker Hub.

```
2022-07-02T07:37:44.815Z        [34mINFO [0m    Need to update DB
2022-07-02T07:37:44.815Z        [34mINFO [0m    Downloading DB...
2022-07-02T07:37:47.705Z        [34mINFO [0m    Detected OS: alpine
2022-07-02T07:37:47.705Z        [33mWARN [0m    This OS version is not on the EOL list: alpine 3.14
2022-07-02T07:37:47.705Z        [34mINFO [0m    Detecting Alpine vulnerabilities...
2022-07-02T07:37:47.711Z        [34mINFO [0m    Number of PL dependency files: 29
2022-07-02T07:37:47.711Z        [34mINFO [0m    Detecting jar vulnerabilities...
2022-07-02T07:37:47.723Z        [33mWARN [0m    This OS version is no longer supported by the distribution: alpine 3.14.6
2022-07-02T07:37:47.723Z        [33mWARN [0m    The vulnerability detection may be insufficient because security updates are not provided

nanditasahu/devsecops-demo:18 (alpine 3.14.6)
=============================================
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)


app/lib/jackson-databind-2.11.4.jar
===================================
Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 1, CRITICAL: 0)

+-----------------------------------------+------------------+----------+-------------------+---------------+---------------------------------------+
|                 LIBRARY                 | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION |                 TITLE                 |
+-----------------------------------------+------------------+----------+-------------------+---------------+---------------------------------------+
| com.fasterxml.jackson.core:jackson-databind | CVE-2020-36518   | HIGH     | 2.11.4            | 2.12.6.1, 2.13.2.1 | jackson-databind: denial of service   |
|                                         |                  |          |                   |               | via a large depth of nested objects   |
|                                         |                  |          |                   |               | -->avd.aquasec.com/nvd/cve-2020-36518 |
+-----------------------------------------+------------------+----------+-------------------+---------------+---------------------------------------+

app/lib/jakarta.el-3.0.3.jar
============================
Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 1, HIGH: 0, CRITICAL: 0)

+-------------------------+------------------+----------+-------------------+---------------+---------------------------------------+
|         LIBRARY         | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION |                 TITLE                 |
+-------------------------+------------------+----------+-------------------+---------------+---------------------------------------+
| org.glassfish:jakarta.el | CVE-2021-28170   | MEDIUM   | 3.0.3             |               | jakarta-el: ELParserTokenManager      |
|                         |                  |          |                   |               | enables invalid EL                    |
|                         |                  |          |                   |               | expressions to be evaluate            |
|                         |                  |          |                   |               | -->avd.aquasec.com/nvd/cve-2021-28170 |
+-------------------------+------------------+----------+-------------------+---------------+---------------------------------------+
```

```
stage('Pushing Docker Image into Docker Hub'){
    steps{
        withCredentials([vaultString(credentialsId: 'vault-dockerhub-password', variable: 'DOCKERHUB_PASSWORD')]) {
        sh '''
        sudo docker login -u nanditasahu -p $DOCKERHUB_PASSWORD
        sudo docker push nanditasahu/devsecops-demo:$BUILD_NUMBER
        '''
       }
    }
}
```

In the next stage, we are pushing the docker image into Docker Hub.

We are first login into the Docker hub using the username and password, which is been passed by vault and then we are pushing the image to Docker Hub.



Get unlimited access Open in app

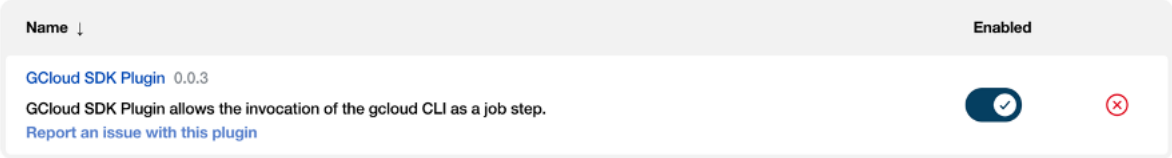
```
[Pipeline] sh
+ sudo docker login -u nanditasahu -p ****
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

```
stage ('Uploading Reports to Cloud Storage'){
    steps{
           withCredentials([vaultFile(credentialsId: 'cloud-storage-access', variable: 'CLOUD_CREDS')]) {
           sh '''
           gcloud version
           gcloud auth activate-service-account --key-file="$CLOUD_CREDS"
           gsutil cp -r $WORKSPACE/trivy-image-scan/trivy-image-scan-$BUILD_NUMBER.txt gs://devsecops-reports
           gsutil ls gs://devsecops-reports
           '''
       }

    }
}
```

In the next stage, we are uploading the reports to the GCP Cloud Storage buckets. To use gcloud cli in Jenkins we need to first install

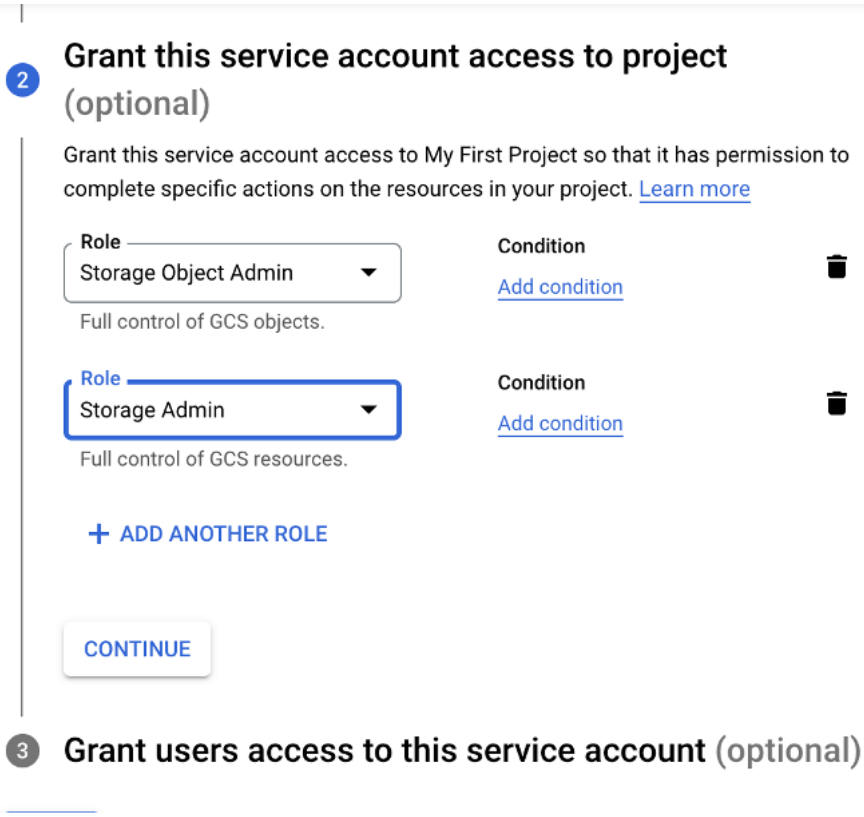
GCLOUD SDK Plugin in Manage Jenkins and install Gcloud CLI in the compute engine where Jenkins is running using the steps below:

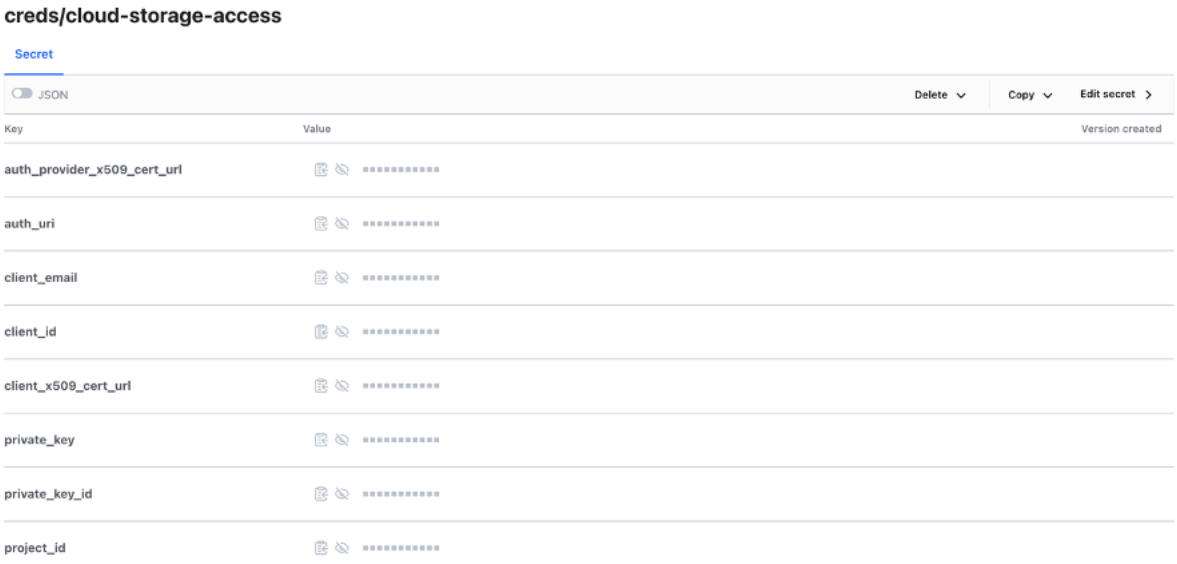
**Install the gcloud CLI | Google Cloud**
This page contains instructions for choosing and maintaining a Google Cloud CLI installation. The Google Cloud CLI...
cloud.google.com




And then create a Service Account which has the roles of Storage Admin and Storage Object Admin. It will allow Jenkins to push trivy scan reports in to the bucket. Add the json key which you get into the vault server and then integrate with Jenkins Credentials.

Get unlimited access Open in app




In the pipeline we are first checking the gcloud version and then we are activating the service account created with the key file.

Then we are copying the trivy scan file from Jenkins workspace to the GCP Cloud Storage and then we are listing the contents of the cloud storage bucket using gsutil command.

Get unlimited access Open in app

```
beta 2022.06.24
bq 2.0.75
bundled-python3-unix 3.9.12
core 2022.06.24
gsutil 5.10
+ gcloud auth activate-service-account --key-file=****
Activated service account credentials for:
+ gsutil cp -r /var/lib/jenkins/workspace/DevSecOps_Final/trivy-image-scan/trivy-image-scan-20.txt gs://devsecops-reports
Copying file:///var/lib/jenkins/workspace/DevSecOps_Final/trivy-image-scan/trivy-image-scan-20.txt [Content-Type=text/plain]...
/ [0 files][    0.0 B/ 11.7 KiB]
/ [1 files][ 11.7 KiB/ 11.7 KiB]
Operation completed over 1 objects/11.7 KiB.
+ gsutil ls gs://devsecops-reports
gs://devsecops-reports/trivy-image-scan-10.txt
gs://devsecops-reports/trivy-image-scan-11.txt
gs://devsecops-reports/trivy-image-scan-12.txt
gs://devsecops-reports/trivy-image-scan-13.txt
gs://devsecops-reports/trivy-image-scan-17.txt
gs://devsecops-reports/trivy-image-scan-18.txt
gs://devsecops-reports/trivy-image-scan-20.txt
gs://devsecops-reports/trivy-image-scan-3.txt
gs://devsecops-reports/trivy-image-scan-4.txt
```

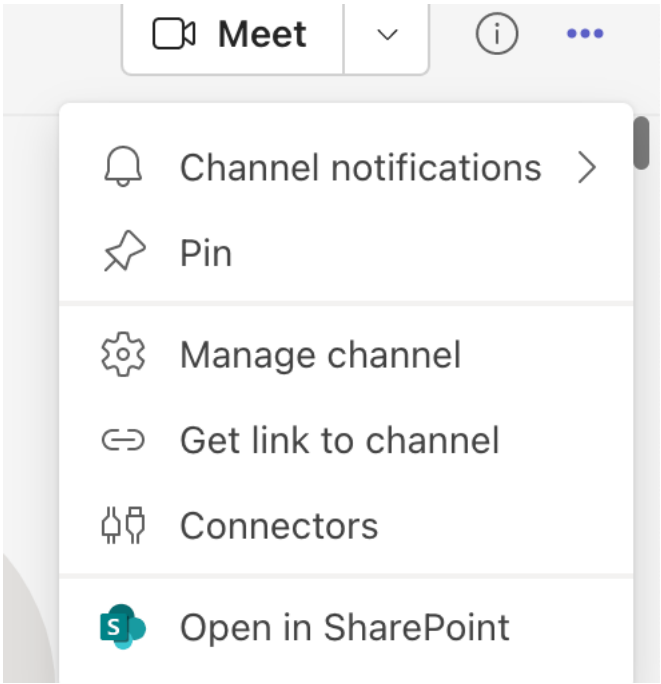
```
stage('Cleaning up DockerImage'){
    steps{
        sh 'sudo docker rmi nanditasahu/devsecops-demo:$BUILD_NUMBER'
    }
}
```

As a best practice, since we don't have the requirement to use the docker images we are cleaning the docker images.

We have also integrated Jenkins with Teams so that we get notifications for successful build of the job or build failure or abort and many more .

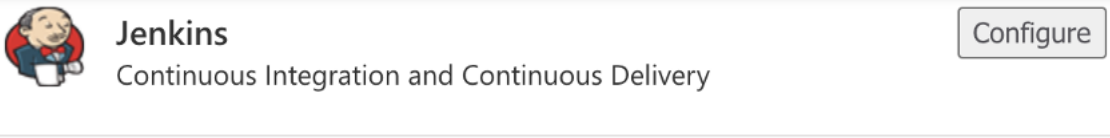
**Steps for Integration Teams with Jenkins**

Create a Teams Channel and once the channel is created, click connector, and add Jenkins

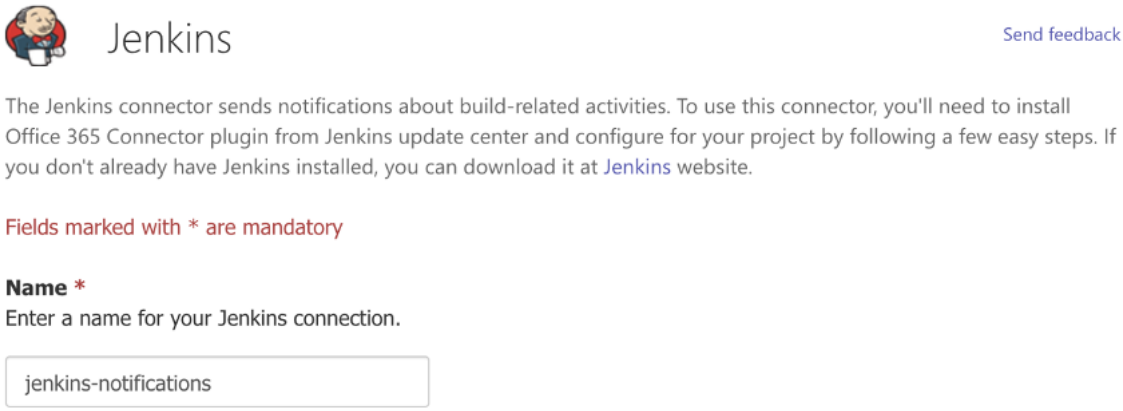



Select Jenkins and click **Configure**.

Get unlimited access Open in app

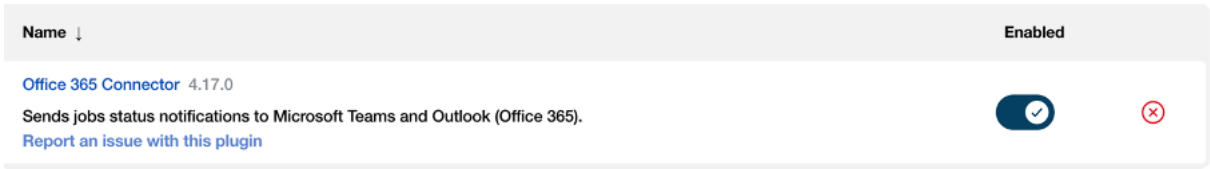



Enter a **name** for the Jenkins connection.



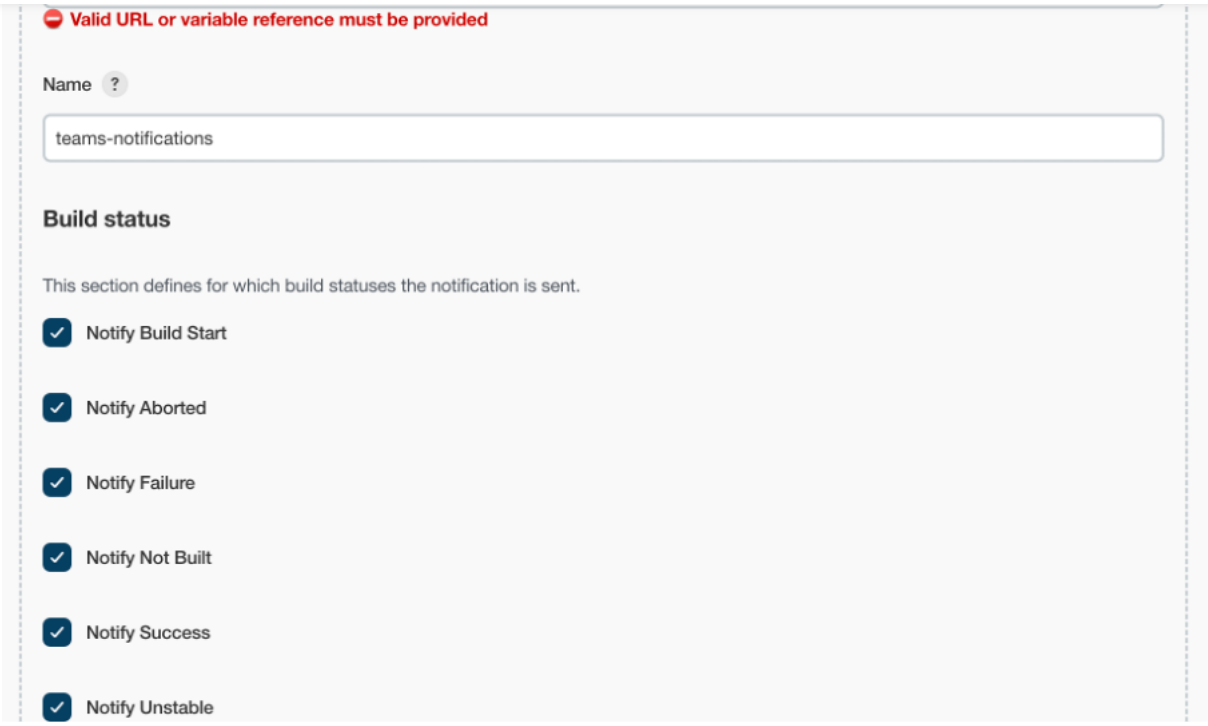

Copy the **webhook URL** and add the url in the Jenkins pipeline


Install the **Office 365 Connector** in Manage Plugins.

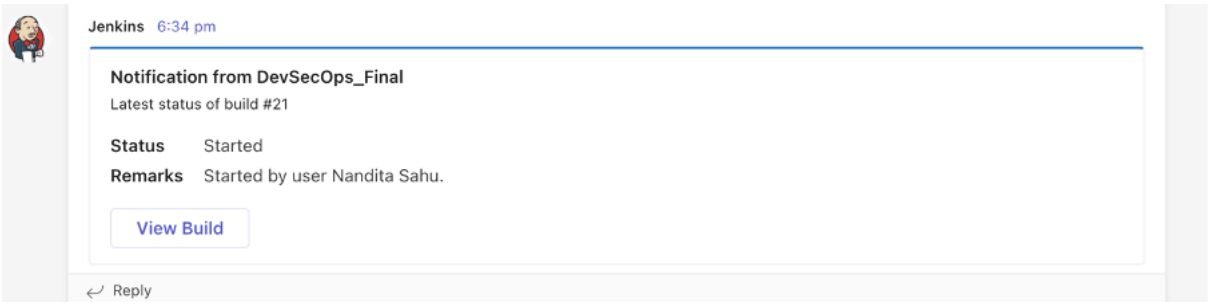



Open your Jenkins Pipeline and in the section **Office 365 Connector** tab paste the Webhook Url and check for all those boxes for which you want to receive events and then click the **Save** button.

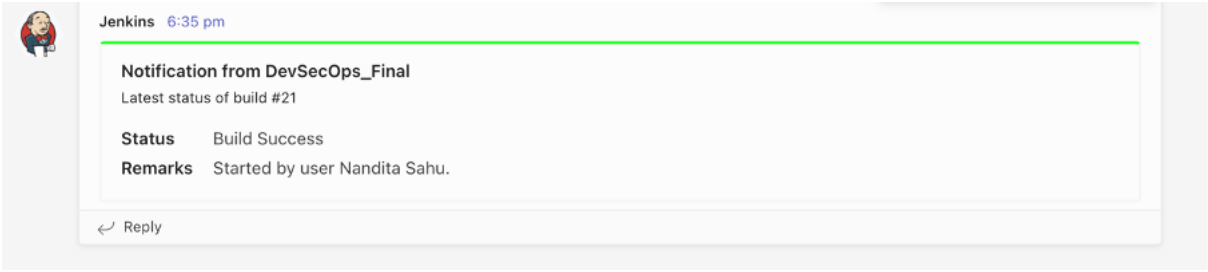
Get unlimited access Open in app

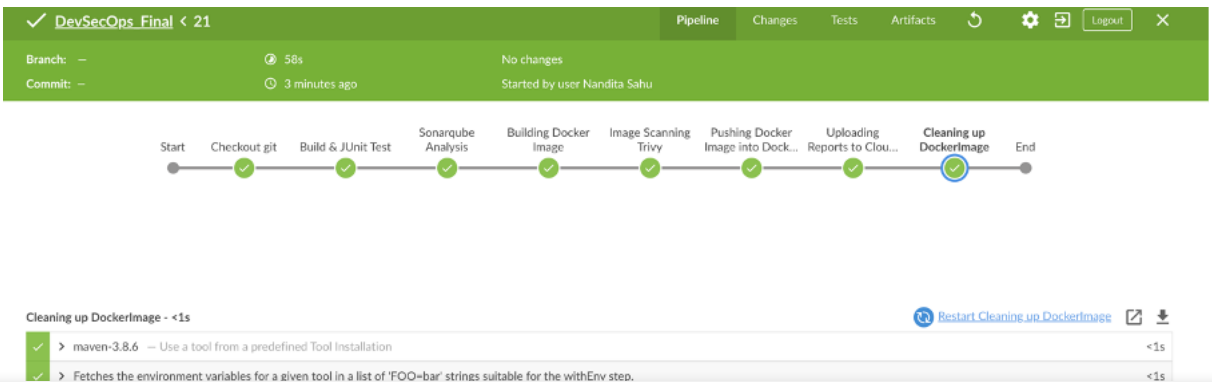



Once the build starts, you'll get notifications in the jenkins-notification channel.




After the build is completed, you will get notifications in the jenkins-notification channel.



Get unlimited access Open in app

To use the above Jenkinsfile and Maven Code use the below repository:

**GitHub - Savegirlchild/DevSecOps_Pipeline**

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or...

github.com