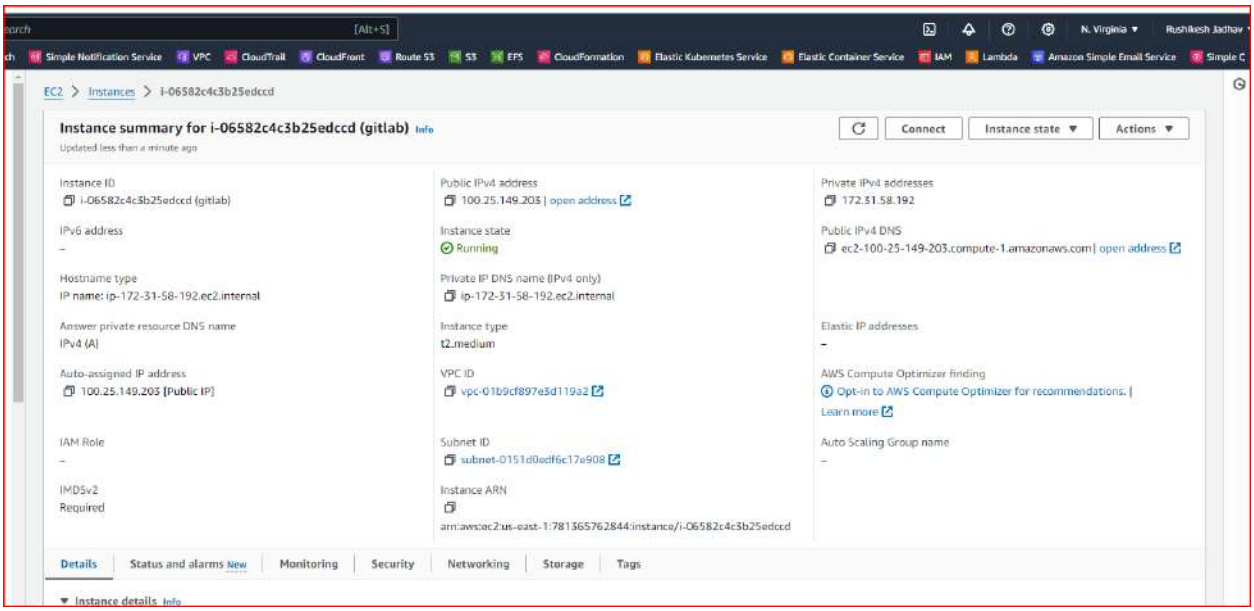
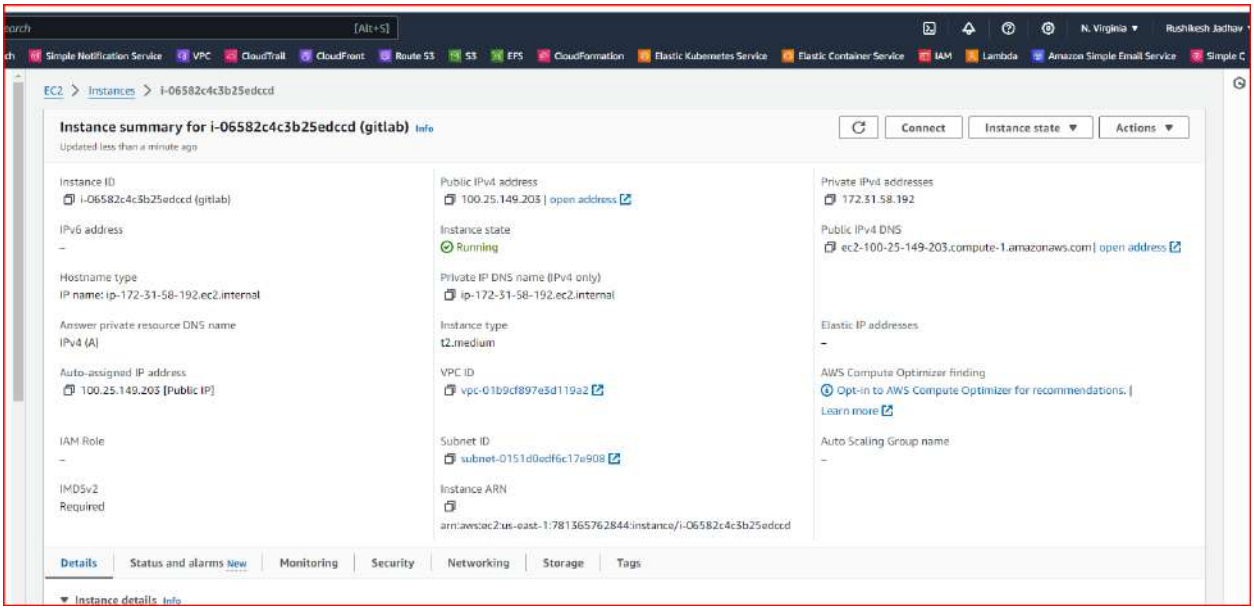
GITLAB POC

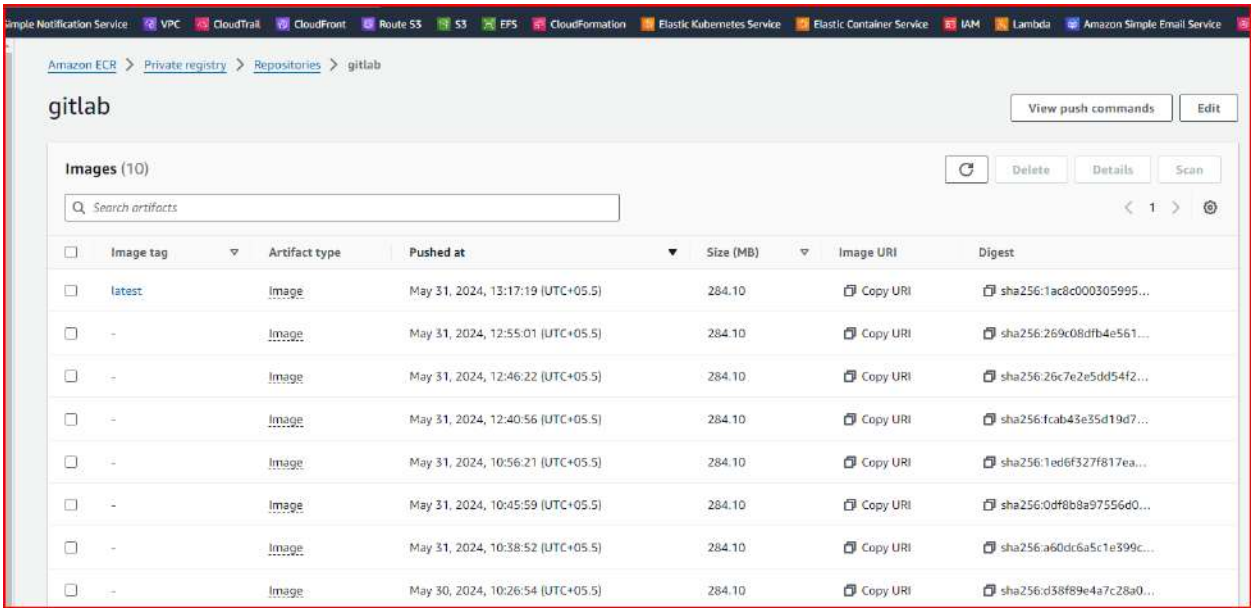
For running Gitlab pipeline we need to do some configurations on AWS



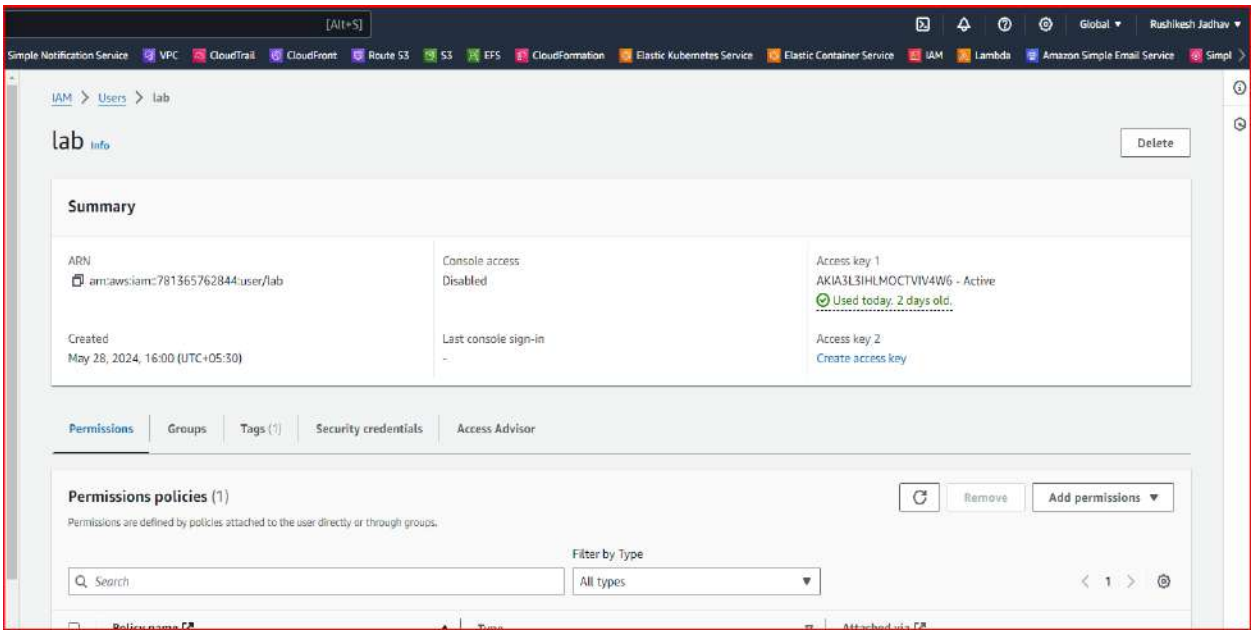
Security group



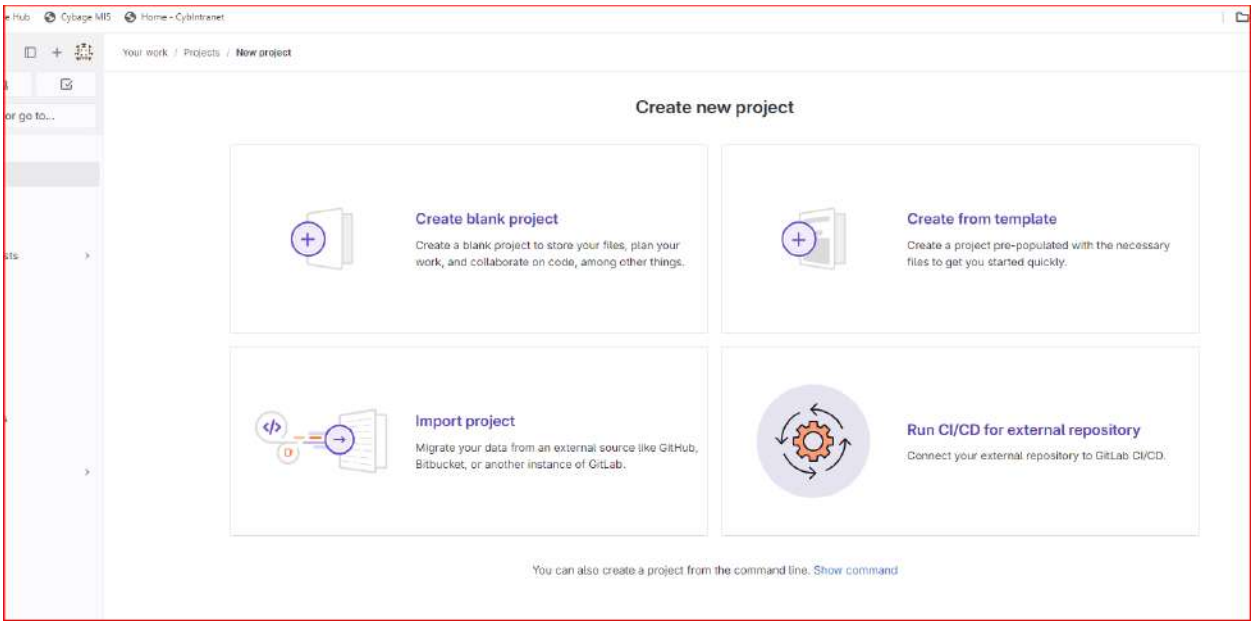
ECR


Amazon ECR > Private registry > Repositories > gitlab
gitlab
View push commands
Edit
Images (10)
Delete
Details
Scan
Search artifacts
Image tag
Artifact type
Pushed at
Size (MB)
Image URI
Digest
latest
Image
May 31, 2024, 13:17:19 (UTC+05:5)
284.10
Copy URI
sha256:1ac8c000305995...
Image
May 31, 2024, 12:55:01 (UTC+05:5)
284.10
Copy URI
sha256:269c08dfb4e561...
Image
May 31, 2024, 12:46:22 (UTC+05:5)
284.10
Copy URI
sha256:26c7e2e5dd54f2...
Image
May 31, 2024, 12:40:56 (UTC+05:5)
284.10
Copy URI
sha256:fcab43e35d19d7...
Image
May 31, 2024, 10:56:21 (UTC+05:5)
284.10
Copy URI
sha256:1ed6f327f817ea...
Image
May 31, 2024, 10:45:59 (UTC+05:5)
284.10
Copy URI
sha256:0df8b8a97556d0...
Image
May 31, 2024, 10:38:52 (UTC+05:5)
284.10
Copy URI
sha256:a60dc6a5c1e399c...
Image
May 30, 2024, 10:26:54 (UTC+05:5)
284.10
Copy URI
sha256:d38f89e4a7c28a0...

IAM

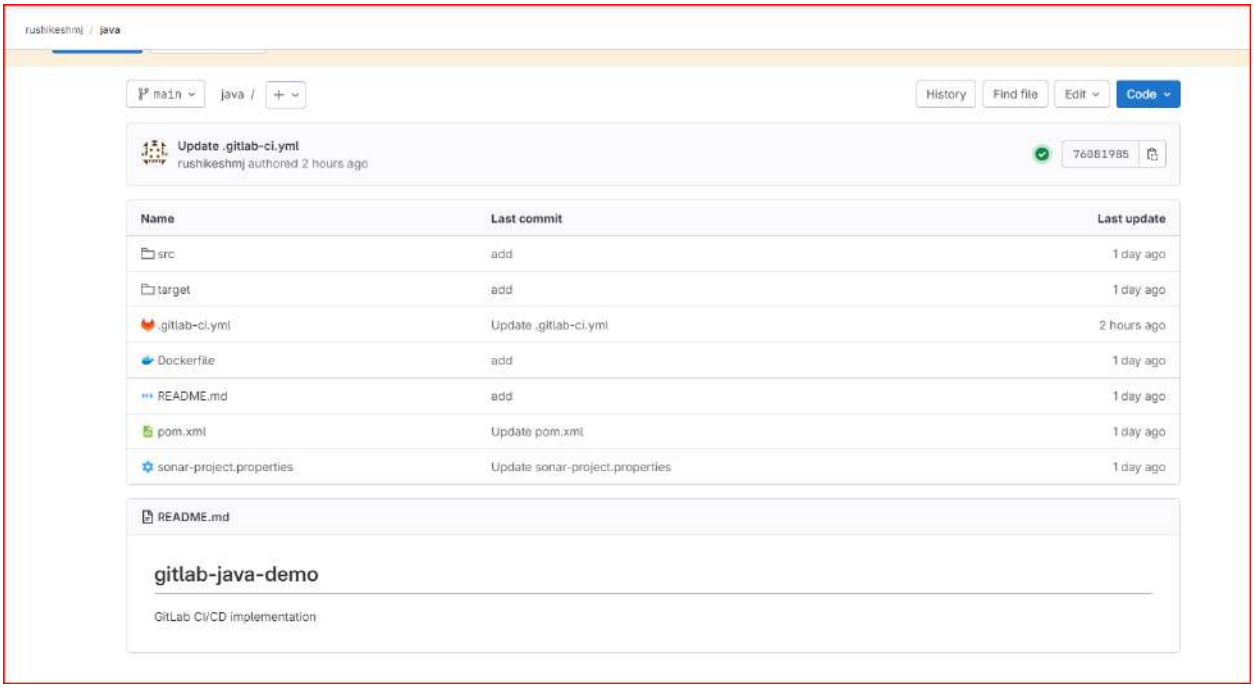

IAM > Users > lab
lab Info
Delete
Summary
ARN
arn:aws:iam::781365762844:user/lab
Console access
Disabled
Access key 1
AKIA3L3IHLMOCTVIV4W6 - Active
Used today. 2 days old.
Created
May 28, 2024, 16:00 (UTC+05:30)
Last console sign-in
-
Access key 2
Create access key
Permissions
Groups
Tags (1)
Security credentials
Access Advisor
Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types

Import project from GITHUB



After that go to project tab on left side and select your project

Create .gitlab-ci.yml file inside the project and start writing yml file.



After doing this install some application on EC2 Console

sudo apt-get update

sudo apt-get install docker.io

sudo su

sudo curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64

sudo chmod +x /usr/local/bin/gitlab-runner

sudo chmod +x /usr/local/bin/gitlab-runner

sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/bash

sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab-runner

sudo gitlab-runner start

sudo gitlab-runner register

sudo vi /etc/gitlab-runner/config.toml

sudo gitlab-runner restart

docker run -d --name sonarcontainer -p 9000:9000 sonarqube

docker ps

docker run -d --name jenkinscontainer -p 8080:8080 jenkins/jenkins

docker ps

sudo apt-get install -y curl unzip

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

aws --version

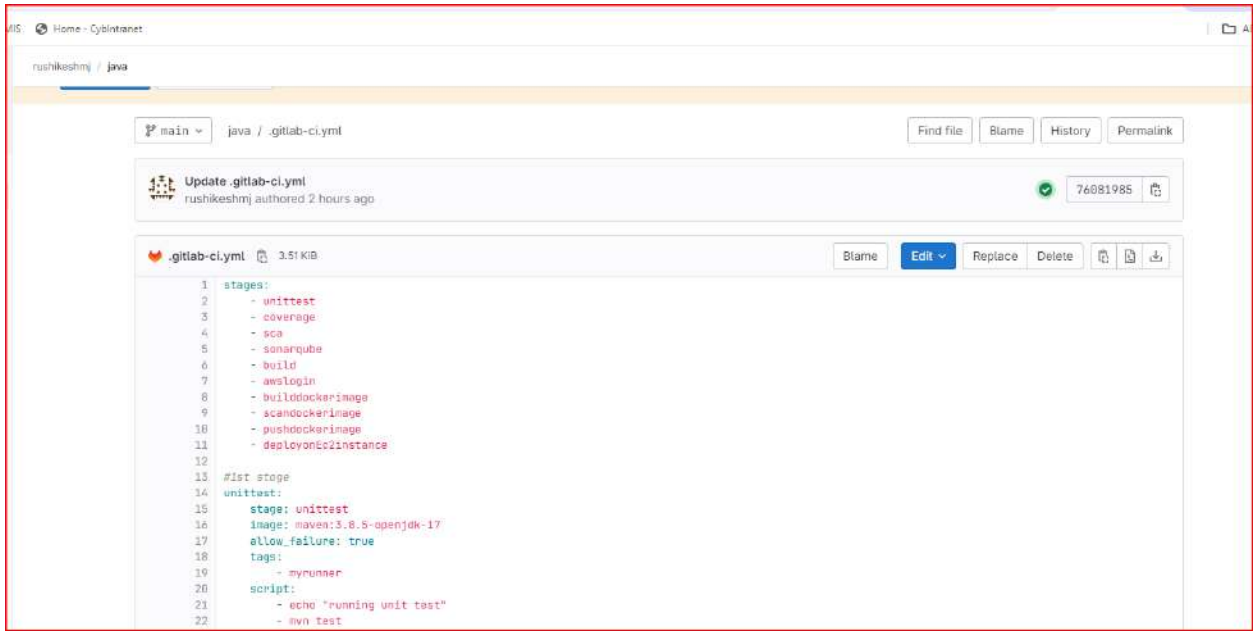
aws configure

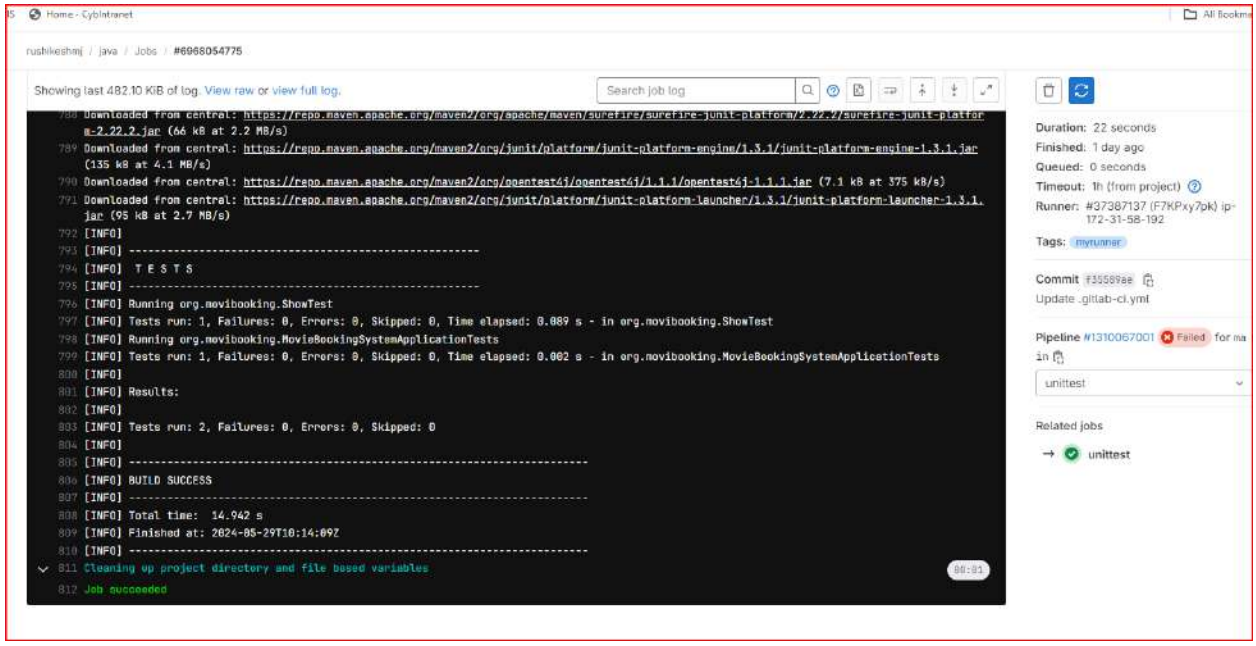
aws login

1. CI steps for GitLab workflow which includes (Use Self Hosted Runner)

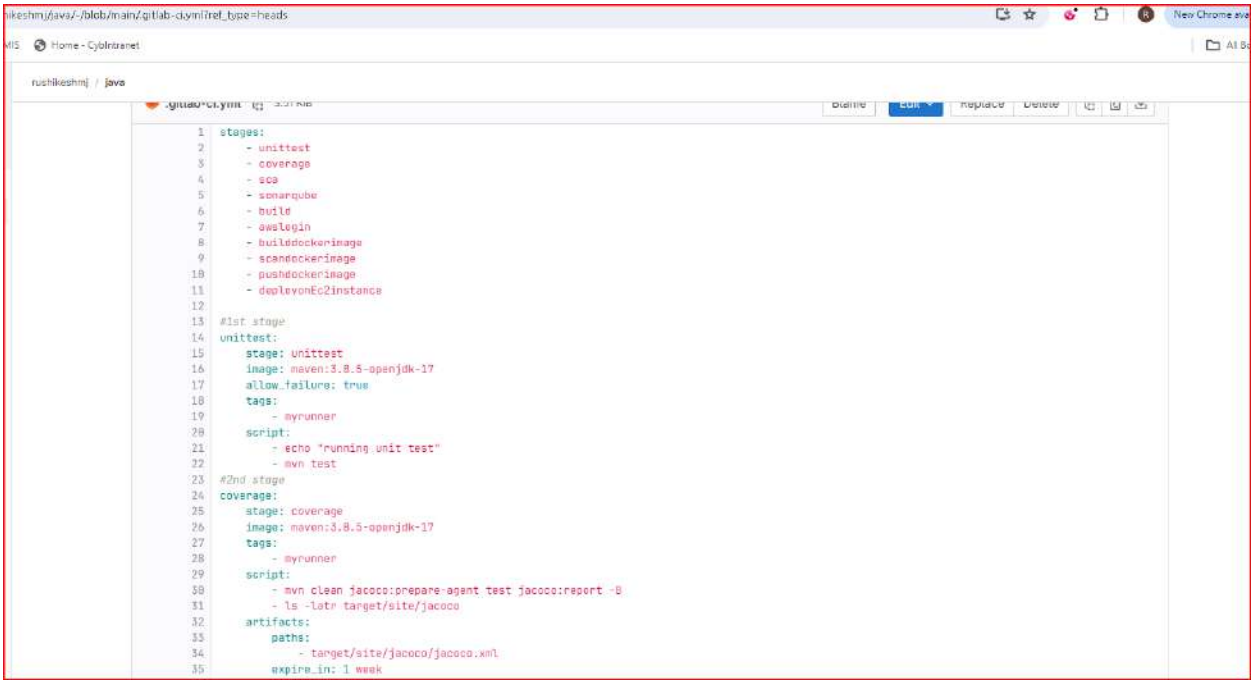
I. Checkout code to repo

II. Unit testing

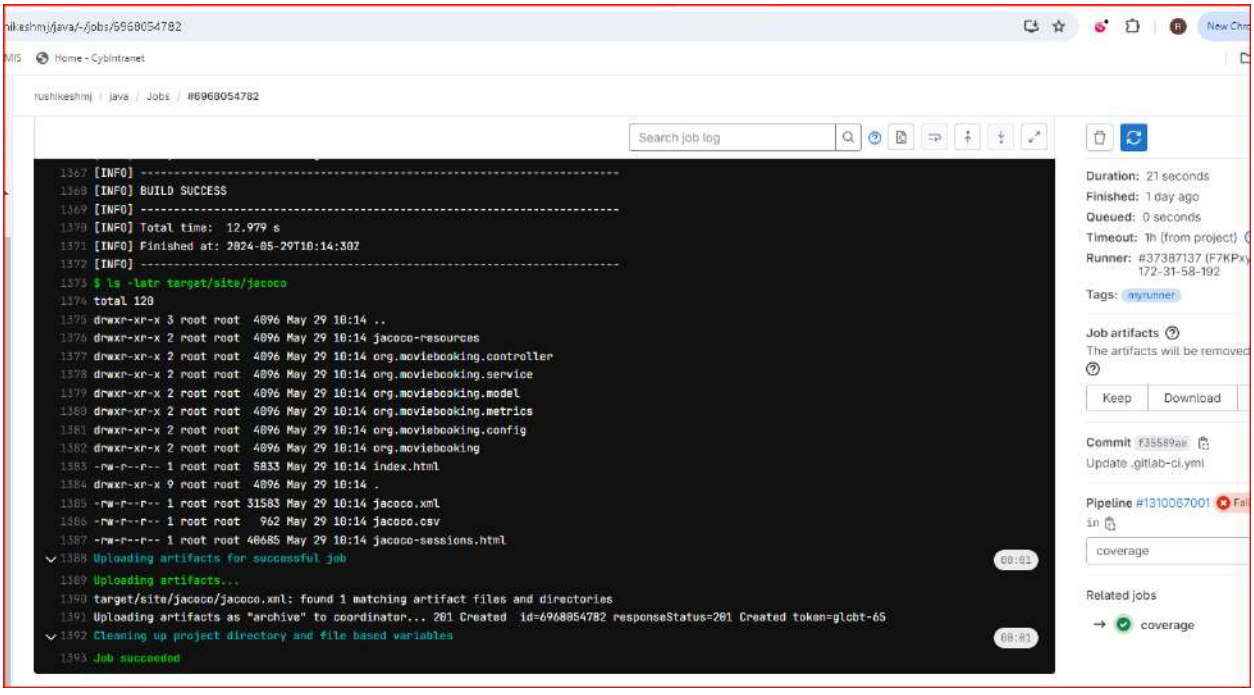




III. code coverage using jacoco

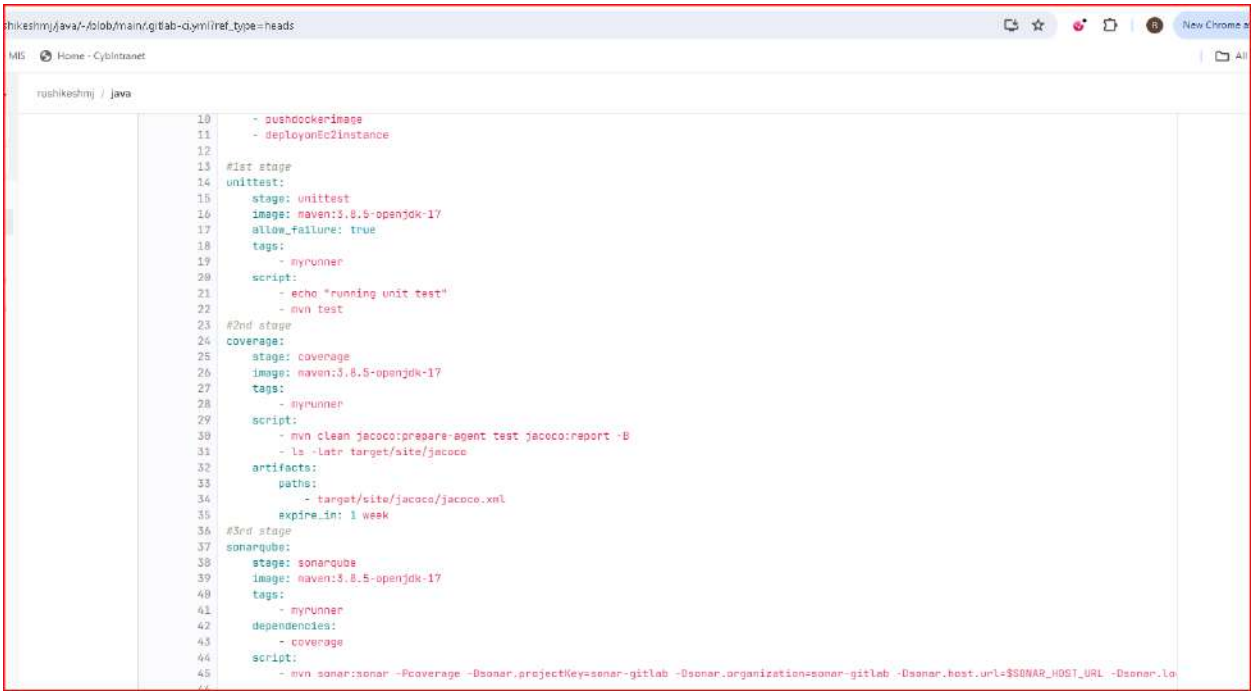


```
stages:
    - unittest
    - coverage
    - sca
    - sonarqube
    - build
    - awslogin
    - builddockerimage
    - scandockerimage
    - pushdockerimage
    - deployonEc2instance

#1st stage
unittest:
    stage: unittest
    image: maven:3.8.5-openjdk-17
    allow_failure: true
    tags:
        - myrunner
    script:
        - echo "running unit test"
        - mvn test
#2nd stage
coverage:
    stage: coverage
    image: maven:3.8.5-openjdk-17
    tags:
        - myrunner
    script:
        - mvn clean jacoco:prepare-agent test jacoco:report -B
        - ls -latr target/site/jacoco
    artifacts:
        paths:
            - target/site/jacoco/jacoco.xml
        expire_in: 1 week
```

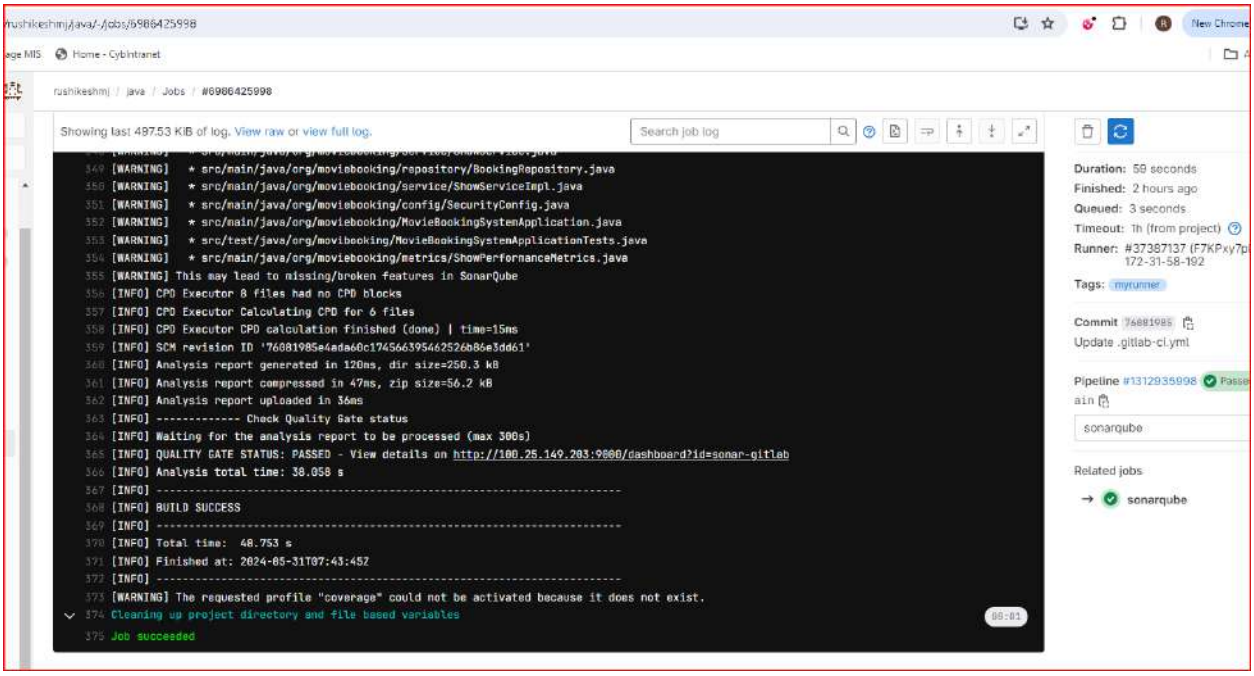


```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.979 s
[INFO] Finished at: 2024-05-29T10:14:30Z
[INFO] ------------------------------------------------------------------------
$ ls -latr target/site/jacoco
total 120
drwxr-xr-x 3 root root  4096 May 29 10:14 ..
drwxr-xr-x 2 root root  4096 May 29 10:14 jacoco-resources
drwxr-xr-x 2 root root  4096 May 29 10:14 org.moviebooking.controller
drwxr-xr-x 2 root root  4096 May 29 10:14 org.moviebooking.service
drwxr-xr-x 2 root root  4096 May 29 10:14 org.moviebooking.model
drwxr-xr-x 2 root root  4096 May 29 10:14 org.moviebooking.metrics
drwxr-xr-x 2 root root  4096 May 29 10:14 org.moviebooking.config
drwxr-xr-x 2 root root  4096 May 29 10:14 org.moviebooking
-rw-r--r-- 1 root root  5833 May 29 10:14 index.html
drwxr-xr-x 9 root root  4096 May 29 10:14 .
-rw-r--r-- 1 root root 31583 May 29 10:14 jacoco.xml
-rw-r--r-- 1 root root   962 May 29 10:14 jacoco.csv
-rw-r--r-- 1 root root 40685 May 29 10:14 jacoco-sessions.html
Uploading artifacts for successful job
Uploading artifacts...
target/site/jacoco/jacoco.xml: found 1 matching artifact files and directories
Uploading artifacts as "archive" to coordinator... 201 Created  id=6968054782 responseStatus=201 Created token=glcbt-65
Cleaning up project directory and file based variables
Job succeeded
```

IV. SCA using SonarQubes along with quality gates

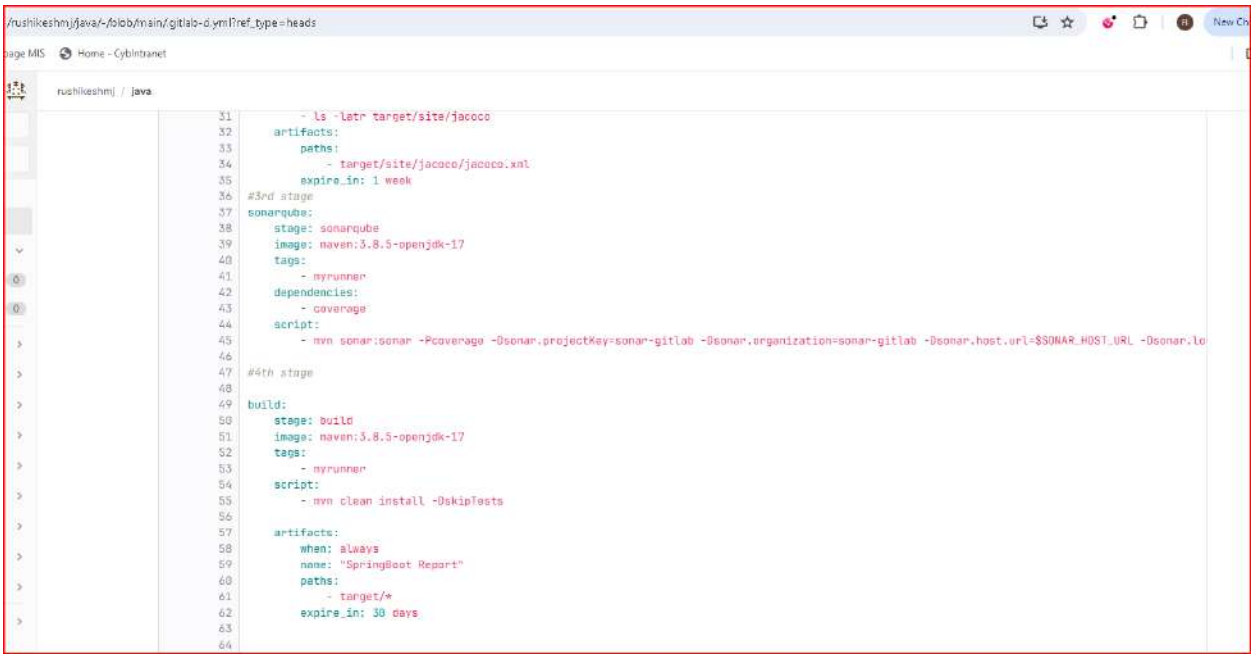


```
    - pushdockerimage
    - deployonEc2instance

#1st stage
unittest:
    stage: unittest
    image: maven:3.8.5-openjdk-17
    allow_failure: true
    tags:
        - myrunner
    script:
        - echo "running unit test"
        - mvn test
#2nd stage
coverage:
    stage: coverage
    image: maven:3.8.5-openjdk-17
    tags:
        - myrunner
    script:
        - mvn clean jacoco:prepare-agent test jacoco:report -B
        - ls -latr target/site/jacoco
    artifacts:
        paths:
            - target/site/jacoco/jacoco.xml
        expire_in: 1 week
#3rd stage
sonarqube:
    stage: sonarqube
    image: maven:3.8.5-openjdk-17
    tags:
        - myrunner
    dependencies:
        - coverage
    script:
        - mvn sonar:sonar -Pcoverage -Dsonar.projectKey=sonar-gitlab -Dsonar.organization=sonar-gitlab -Dsonar.host.url=$SONAR_HOST_URL -Dsonar.lo
```

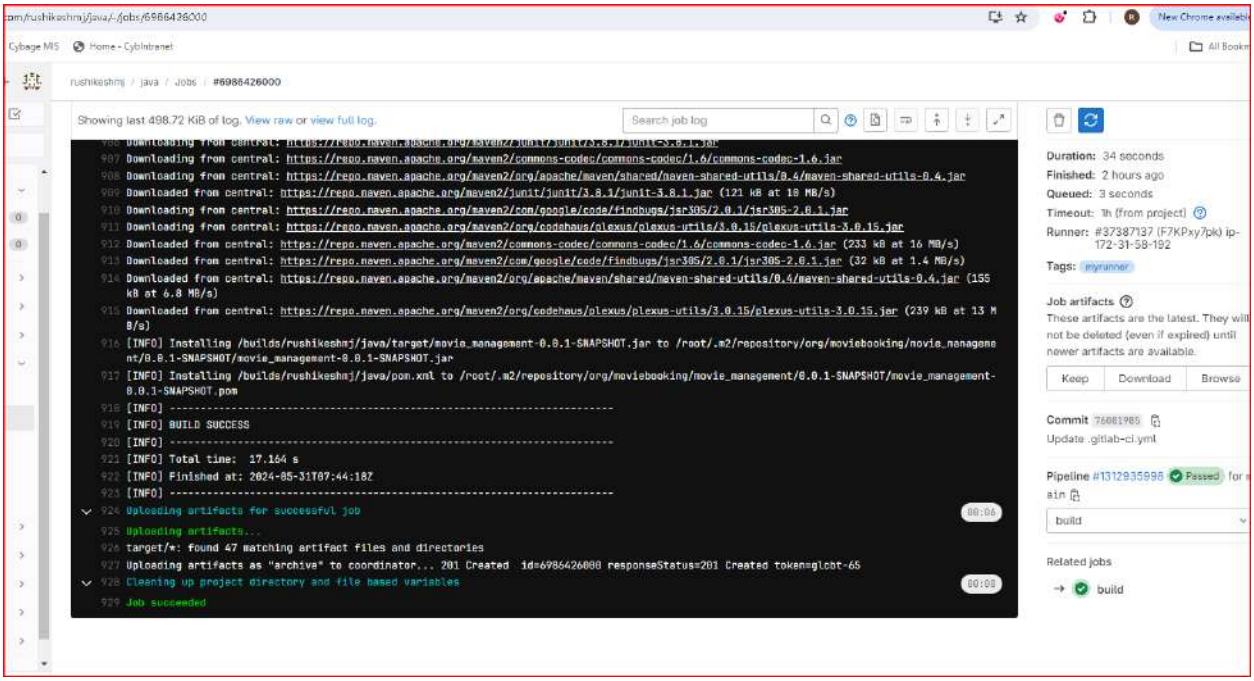


```
[WARNING]   * src/main/java/org/moviebooking/repository/BookingRepository.java
[WARNING]   * src/main/java/org/moviebooking/service/ShowServiceImpl.java
[WARNING]   * src/main/java/org/moviebooking/config/SecurityConfig.java
[WARNING]   * src/main/java/org/moviebooking/MovieBookingSystemApplication.java
[WARNING]   * src/test/java/org/moviebooking/MovieBookingSystemApplicationTests.java
[WARNING]   * src/main/java/org/moviebooking/metrics/ShowPerformanceMetrics.java
[WARNING] This may lead to missing/broken features in SonarQube
[INFO] CPD Executor 8 files had no CPD blocks
[INFO] CPD Executor Calculating CPD for 6 files
[INFO] CPD Executor CPD calculation finished (done) | time=15ms
[INFO] SCM revision ID '76081985e4ada60c174566395462526b86e3dd61'
[INFO] Analysis report generated in 120ms, dir size=250.3 kB
[INFO] Analysis report compressed in 47ms, zip size=56.2 kB
[INFO] Analysis report uploaded in 36ms
[INFO] ------------ Check Quality Gate status
[INFO] Waiting for the analysis report to be processed (max 300s)
[INFO] QUALITY GATE STATUS: PASSED - View details on http://100.25.149.203:9000/dashboard?id=sonar-gitlab
[INFO] Analysis total time: 38.058 s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  48.753 s
[INFO] Finished at: 2024-05-31T07:43:45Z
[INFO] ------------------------------------------------------------------------
[WARNING] The requested profile "coverage" could not be activated because it does not exist.
Cleaning up project directory and file based variables
Job succeeded
```

V. Build and package JAR

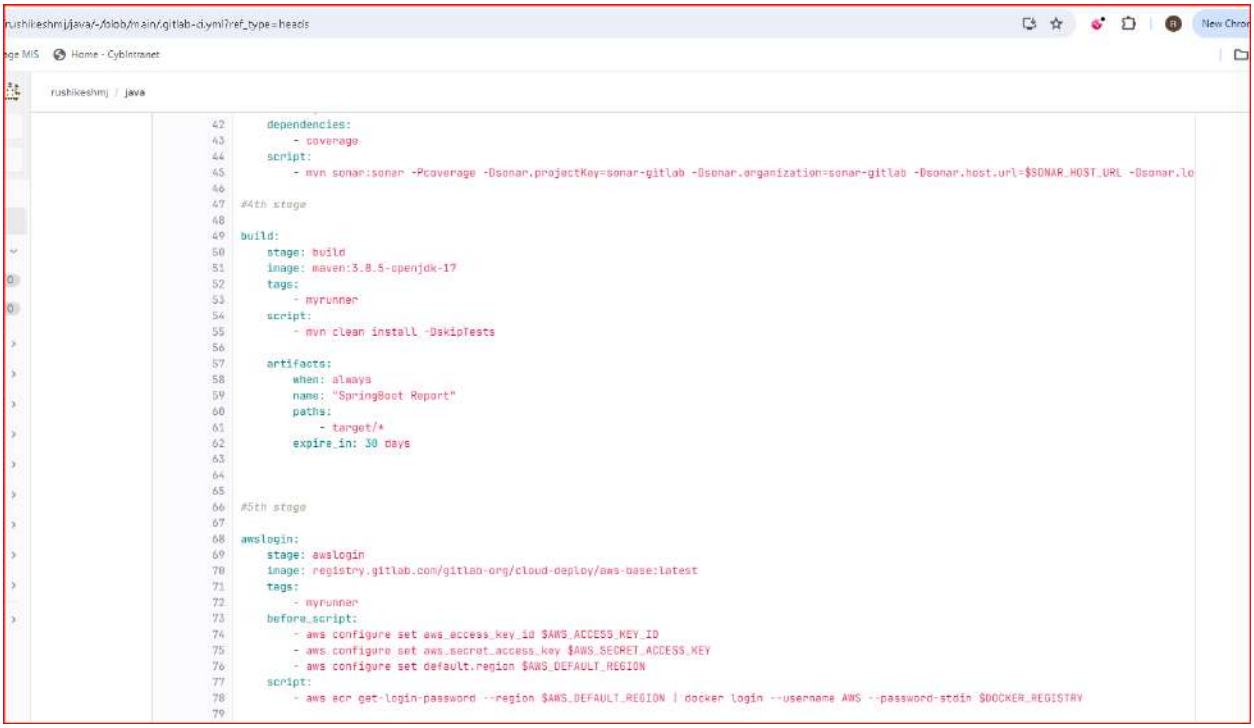


```
31          - ls -latr target/site/jacoco
32      artifacts:
33          paths:
34              - target/site/jacoco/jacoco.xml
35          expire_in: 1 week
36  #3rd stage
37  sonarqube:
38      stage: sonarqube
39      image: maven:3.8.5-openjdk-17
40      tags:
41          - myrunner
42      dependencies:
43          - coverage
44      script:
45          - mvn sonar:sonar -Pcoverage -Dsonar.projectKey=sonar-gitlab -Dsonar.organization=sonar-gitlab -Dsonar.host.url=$SONAR_HOST_URL -Dsonar.lo
46
47  #4th stage
48
49  build:
50      stage: build
51      image: maven:3.8.5-openjdk-17
52      tags:
53          - myrunner
54      script:
55          - mvn clean install -DskipTests
56
57      artifacts:
58          when: always
59          name: "SpringBoot Report"
60          paths:
61              - target/*
62          expire_in: 30 days
63
64
```

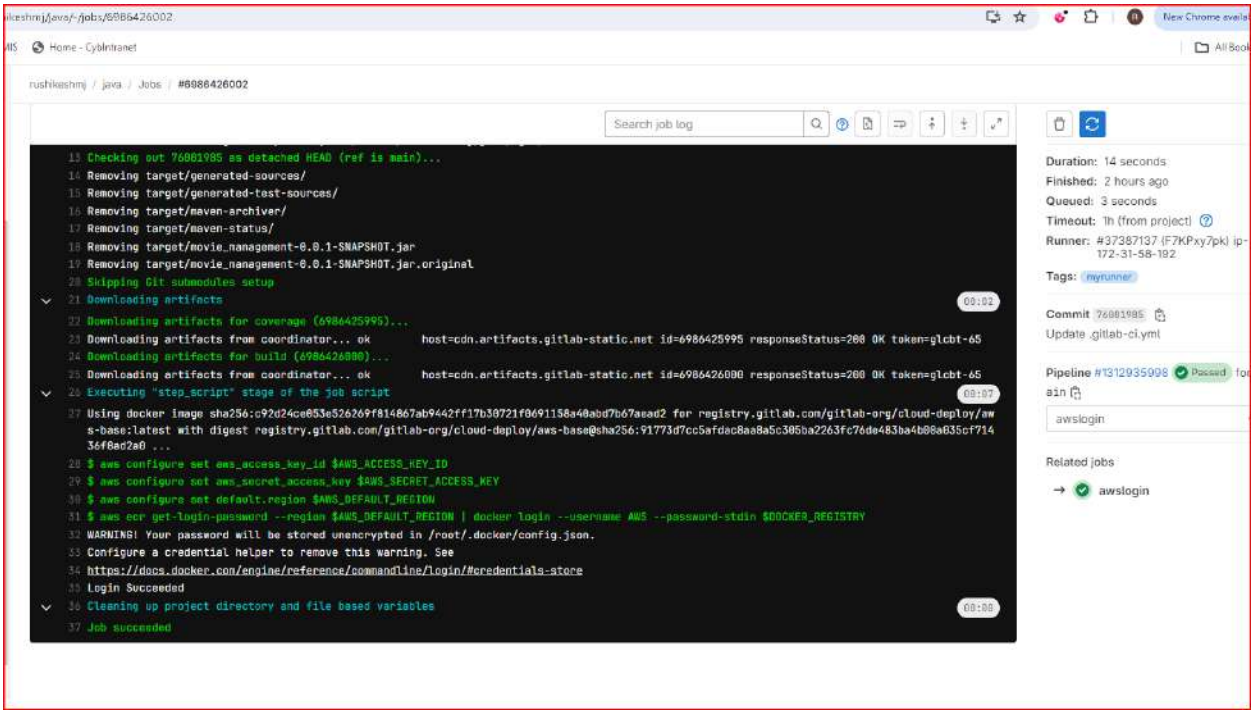


```
Showing last 498.72 KiB of log. View raw or view full log.
Search job log

907 Downloading from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.6/commons-codec-1.6.jar
908 Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.4/maven-shared-utils-0.4.jar
909 Downloaded from central: https://repo.maven.apache.org/maven2/junit/junit/3.8.1/junit-3.8.1.jar (121 kB at 10 MB/s)
910 Downloading from central: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.1/jsr305-2.0.1.jar
911 Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar
912 Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.6/commons-codec-1.6.jar (233 kB at 16 MB/s)
913 Downloaded from central: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.1/jsr305-2.0.1.jar (32 kB at 1.4 MB/s)
914 Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.4/maven-shared-utils-0.4.jar (155
    kB at 6.8 MB/s)
915 Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar (239 kB at 13 M
    B/s)
916 [INFO] Installing /builds/rushikeshmj/java/target/movie_management-0.0.1-SNAPSHOT.jar to /root/.m2/repository/org/moviebooking/movie_managame
    nt/0.0.1-SNAPSHOT/movie_management-0.0.1-SNAPSHOT.jar
917 [INFO] Installing /builds/rushikeshmj/java/pom.xml to /root/.m2/repository/org/moviebooking/movie_management/0.0.1-SNAPSHOT/movie_management-
    0.0.1-SNAPSHOT.pom
918 [INFO] ------------------------------------------------------------------------
919 [INFO] BUILD SUCCESS
920 [INFO] ------------------------------------------------------------------------
921 [INFO] Total time:  17.164 s
922 [INFO] Finished at: 2024-05-31T07:44:18Z
923 [INFO] ------------------------------------------------------------------------
924 Uploading artifacts for successful job                                    00:04
925 Uploading artifacts...
926 target/*: found 47 matching artifact files and directories
927 Uploading artifacts as "archive" to coordinator... 201 Created  id=6986426000 responseStatus=201 Created token=glcbt-65
928 Cleaning up project directory and file based variables                   00:00
929 Job succeeded

Duration: 34 seconds
Finished: 2 hours ago
Queued: 3 seconds
Timeout: 1h (from project)
Runner: #37387137 (F7KPxy7pk) ip-172-31-58-192
Tags: myrunner

Job artifacts
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.
Keep  Download  Browse

Commit 76081985
Update .gitlab-ci.yml

Pipeline #1312935998 Passed for main
build

Related jobs
build
```

VI. Configuration of AWS



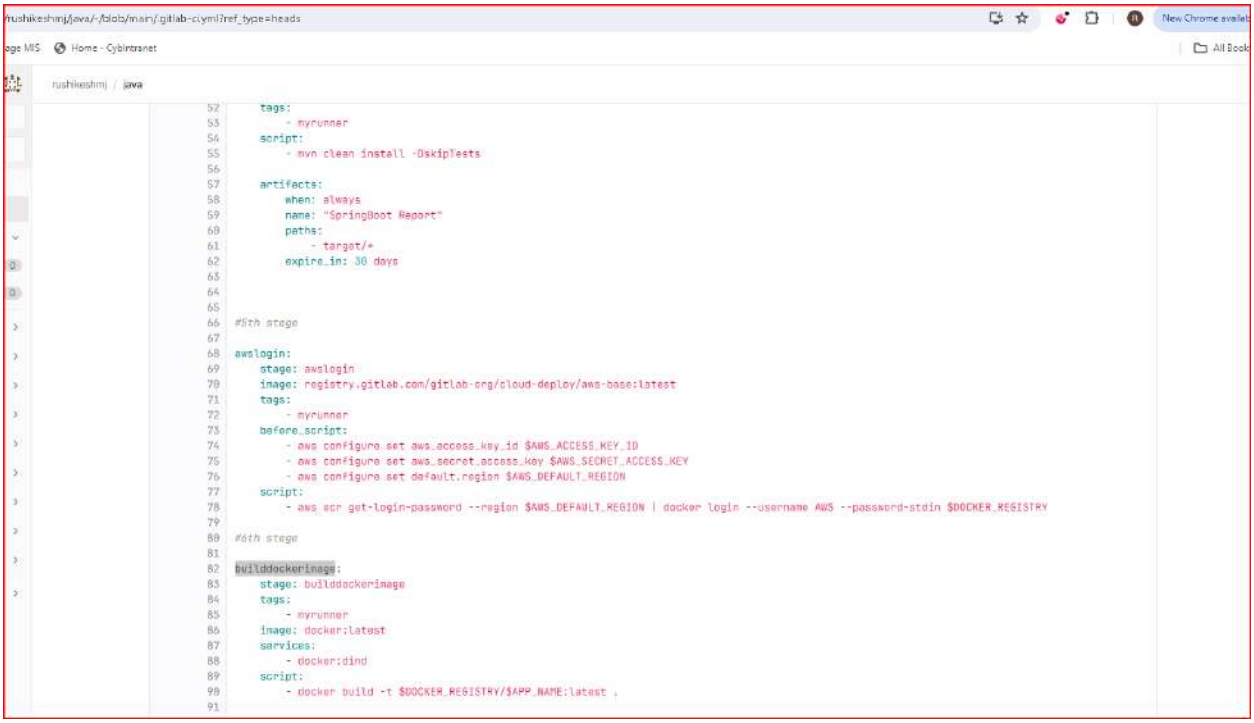
```
    dependencies:
        - coverage
    script:
        - mvn sonar:sonar -Pcoverage -Dsonar.projectKey=sonar-gitlab -Dsonar.organization=sonar-gitlab -Dsonar.host.url=$SONAR_HOST_URL -Dsonar.lo

#4th stage

build:
    stage: build
    image: maven:3.8.5-openjdk-17
    tags:
        - myrunner
    script:
        - mvn clean install -DskipTests

    artifacts:
        when: always
        name: "SpringBoot Report"
        paths:
            - target/*
        expire_in: 30 days


#5th stage

awslogin:
    stage: awslogin
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    before_script:
        - aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
        - aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
        - aws configure set default.region $AWS_DEFAULT_REGION
    script:
        - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY
```



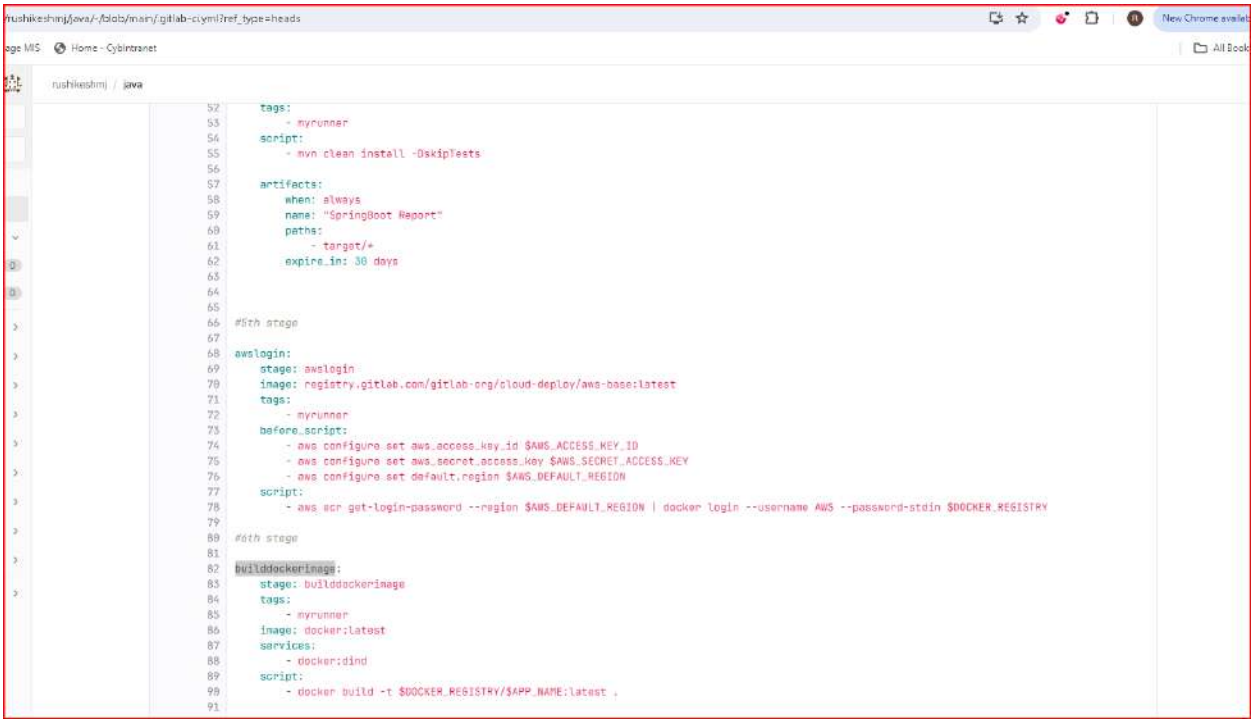
```
Checking out 76001985 as detached HEAD (ref is main)...
Removing target/generated-sources/
Removing target/generated-test-sources/
Removing target/maven-archiver/
Removing target/maven-status/
Removing target/movie_management-0.0.1-SNAPSHOT.jar
Removing target/movie_management-0.0.1-SNAPSHOT.jar.original
Skipping Git submodules setup
Downloading artifacts                                                    00:02
Downloading artifacts for coverage (6986425995)...
Downloading artifacts from coordinator... ok        host=cdn.artifacts.gitlab-static.net id=6986425995 responseStatus=200 OK token=glcbt-65
Downloading artifacts for build (6986426000)...
Downloading artifacts from coordinator... ok        host=cdn.artifacts.gitlab-static.net id=6986426000 responseStatus=200 OK token=glcbt-65
Executing "step_script" stage of the job script                          00:07
Using docker image sha256:c92d24ce653e526269f814867ab9442ff17b30721f0691158a40abd7b67aead2 for registry.gitlab.com/gitlab-org/cloud-deploy/aw
s-base:latest with digest registry.gitlab.com/gitlab-org/cloud-deploy/aws-base@sha256:91773d7cc5afdac8aa8a5c305ba2263fc76da483ba4b80a635cf714
36f8ad2a0 ...
$ aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
$ aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
$ aws configure set default.region $AWS_DEFAULT_REGION
$ aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
Cleaning up project directory and file based variables                    00:00
Job succeeded
```

```
Duration: 14 seconds
Finished: 2 hours ago
Queued: 3 seconds
Timeout: 1h (from project)
Runner: #37387137 (F7KPxy7pk) ip-
172-31-58-192
Tags: myrunner
Commit 76001985
Update .gitlab-ci.yml
Pipeline #1312935908 Passed
awslogin
Related jobs
awslogin
```

VII. builddockerimage

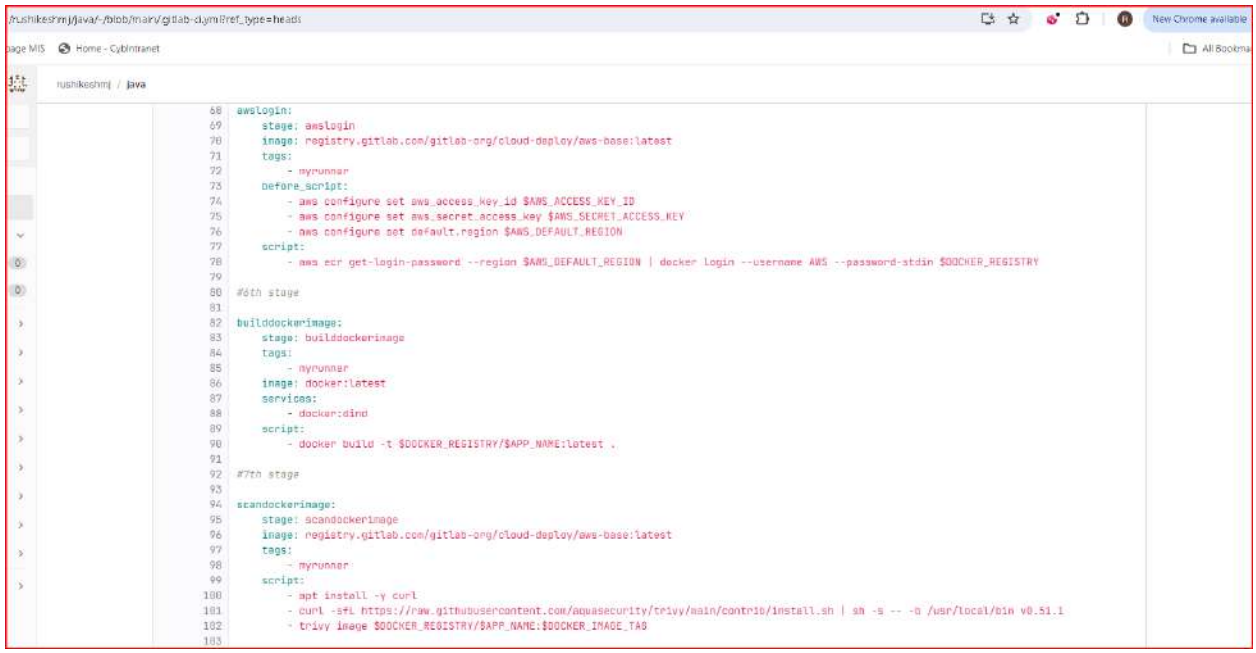


```
    tags:
        - myrunner
    script:
        - mvn clean install -DskipTests

    artifacts:
        when: always
        name: "SpringBoot Report"
        paths:
            - target/*
        expire_in: 30 days



#5th stage

awslogin:
    stage: awslogin
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    before_script:
        - aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
        - aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
        - aws configure set default.region $AWS_DEFAULT_REGION
    script:
        - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY

#6th stage

builddockerimage:
    stage: builddockerimage
    tags:
        - myrunner
    image: docker:latest
    services:
        - docker:dind
    script:
        - docker build -t $DOCKER_REGISTRY/$APP_NAME:latest .
```

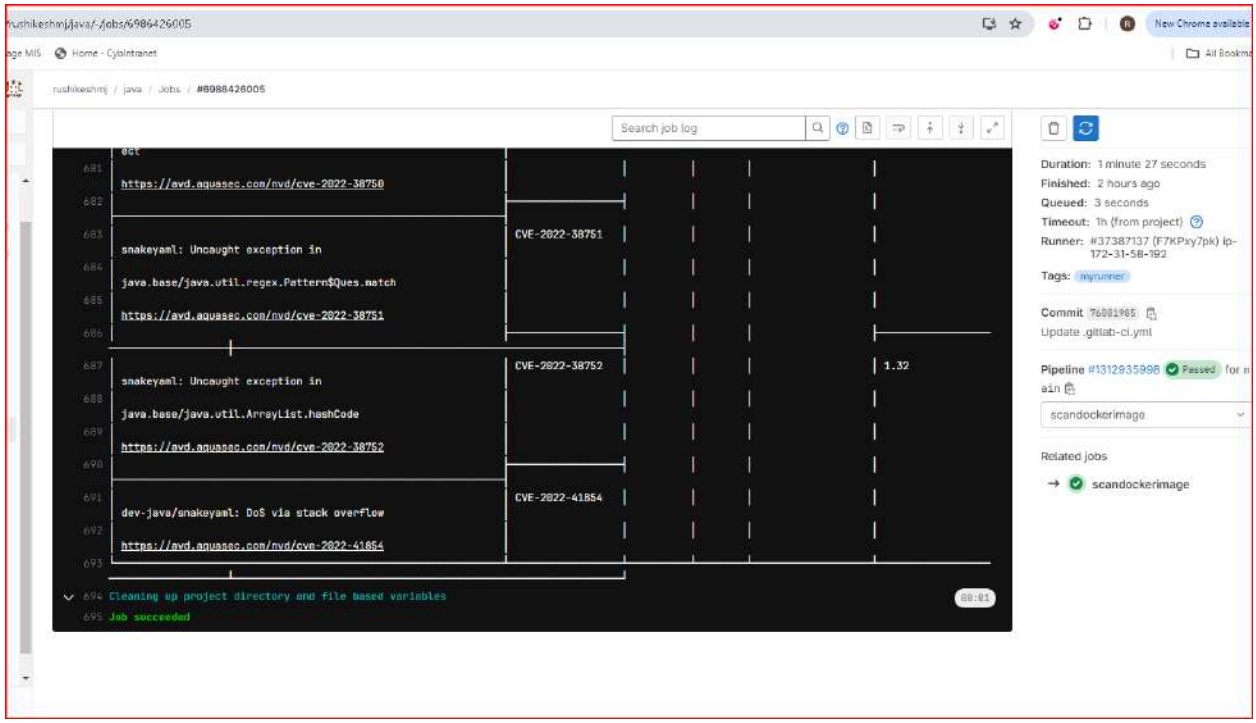


```
    tags:
        - myrunner
    script:
        - mvn clean install -DskipTests

    artifacts:
        when: always
        name: "SpringBoot Report"
        paths:
            - target/*
        expire_in: 30 days



#5th stage

awslogin:
    stage: awslogin
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    before_script:
        - aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
        - aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
        - aws configure set default.region $AWS_DEFAULT_REGION
    script:
        - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY

#6th stage

builddockerimage:
    stage: builddockerimage
    tags:
        - myrunner
    image: docker:latest
    services:
        - docker:dind
    script:
        - docker build -t $DOCKER_REGISTRY/$APP_NAME:latest .
```

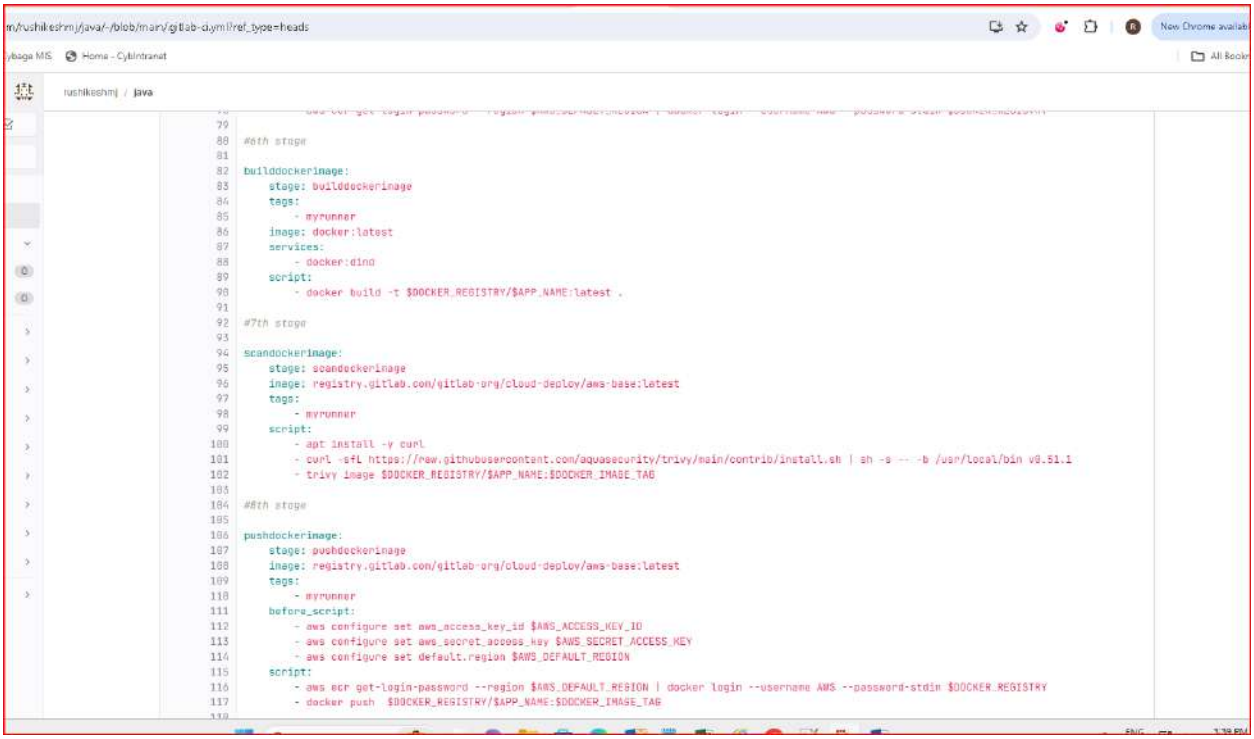
VIII. scandockerimage



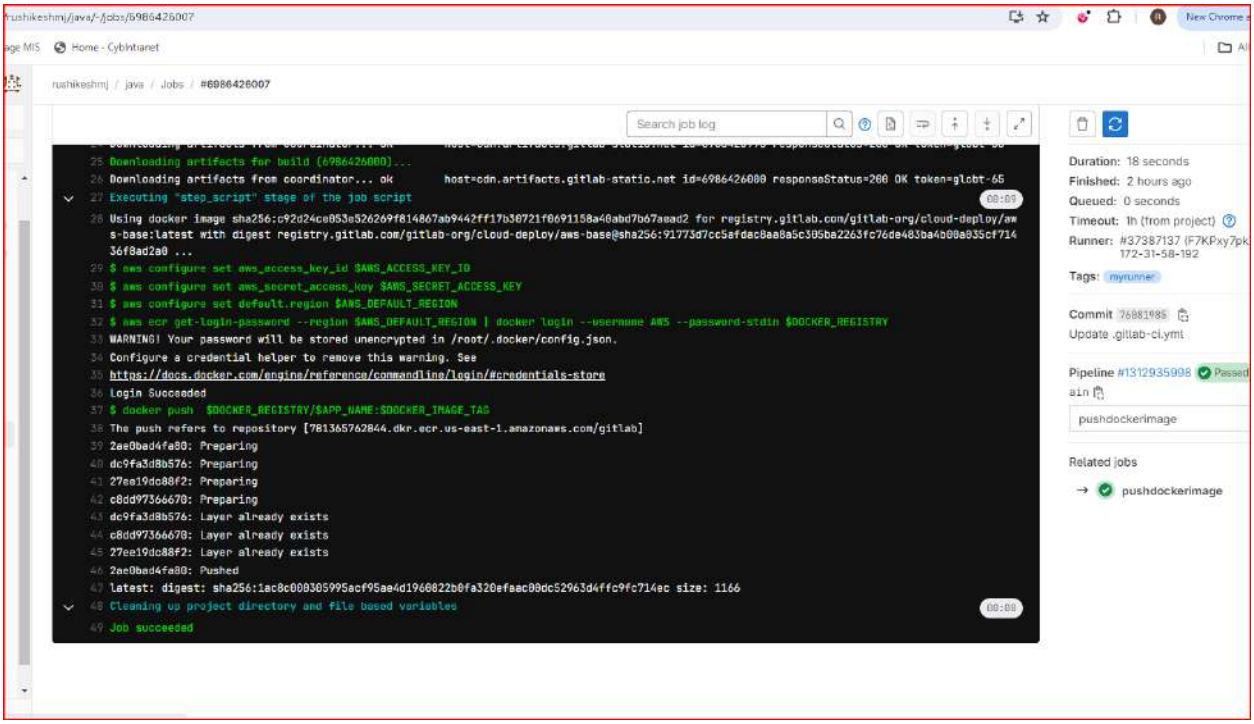
```
awslogin:
    stage: awslogin
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    before_script:
        - aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
        - aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
        - aws configure set default.region $AWS_DEFAULT_REGION
    script:
        - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY

#6th stage

builddockerimage:
    stage: builddockerimage
    tags:
        - myrunner
    image: docker:latest
    services:
        - docker:dind
    script:
        - docker build -t $DOCKER_REGISTRY/$APP_NAME:latest .

#7th stage

scandockerimage:
    stage: scandockerimage
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    script:
        - apt install -y curl
        - curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sh -s -- -b /usr/local/bin v0.51.1
        - trivy image $DOCKER_REGISTRY/$APP_NAME:$DOCKER_IMAGE_TAG
```



IX. pushdockerimage



```
#8th stage

builddockerimage:
    stage: builddockerimage
    tags:
        - myrunner
    image: docker:latest
    services:
        - docker:dind
    script:
        - docker build -t $DOCKER_REGISTRY/$APP_NAME:latest .

#7th stage

scandockerimage:
    stage: scandockerimage
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    script:
        - apt install -y curl
        - curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sh -s -- -b /usr/local/bin v0.51.1
        - trivy image $DOCKER_REGISTRY/$APP_NAME:$DOCKER_IMAGE_TAG

##8th stage

pushdockerimage:
    stage: pushdockerimage
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    before_script:
        - aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
        - aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
        - aws configure set default.region $AWS_DEFAULT_REGION
    script:
        - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY
        - docker push  $DOCKER_REGISTRY/$APP_NAME:$DOCKER_IMAGE_TAG
```

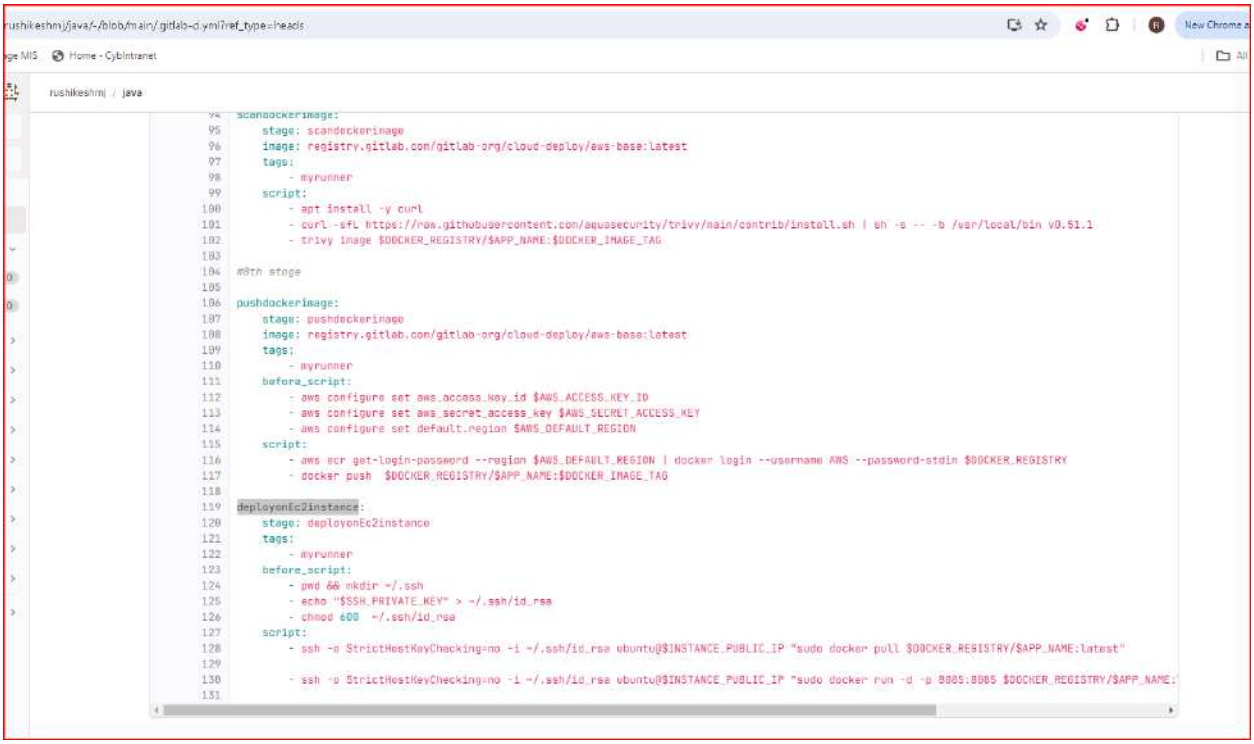


```
Downloading artifacts for build (6986426008)...
Downloading artifacts from coordinator... ok      host=cdn.artifacts.gitlab-static.net id=6986426008 responseStatus=200 OK token=glcbt-65
Executing "step_script" stage of the job script                                                                                    00:09
Using docker image sha256:c92d24ce053e526269f814867ab9442ff17b30721f0691158a40abd7b67aead2 for registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest with digest registry.gitlab.com/gitlab-org/cloud-deploy/aws-base@sha256:91773d7cc5afdac8ae8a5c305ba2263fc76de483ba4b00a835cf71436f8ad2a0 ...
$ aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
$ aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
$ aws configure set default.region $AWS_DEFAULT_REGION
$ aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
$ docker push  $DOCKER_REGISTRY/$APP_NAME:$DOCKER_IMAGE_TAG
The push refers to repository [781365762844.dkr.ecr.us-east-1.amazonaws.com/gitlab]
2ae0bad4fa80: Preparing
dc9fa3d8b576: Preparing
27ee19dc88f2: Preparing
c8dd97366670: Preparing
dc9fa3d8b576: Layer already exists
c8dd97366670: Layer already exists
27ee19dc88f2: Layer already exists
2ae0bad4fa80: Pushed
latest: digest: sha256:1ac8c000305995acf95ae4d1960822b0fa320efaac00dc52963d4ffc9fc714ec size: 1166
Cleaning up project directory and file based variables                                                                              00:00
Job succeeded
```

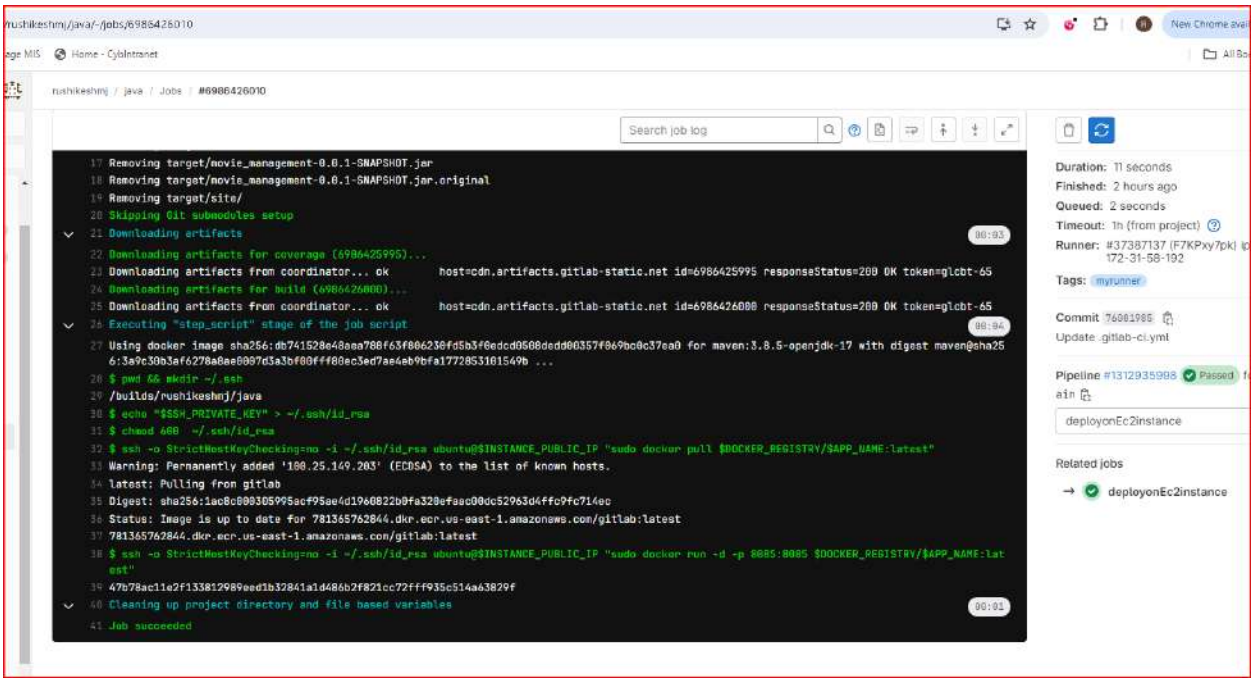
Duration: 18 seconds
Finished: 2 hours ago
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #37387137 (F7KPxy7pk)
172-31-58-192
Tags: myrunner
Commit 76881985
Update .gitlab-ci.yml
Pipeline #1312935998 Passed
pushdockerimage
Related jobs
pushdockerimage

2. CD steps includes

1. Deploy application on EC2 using SSH Private Key

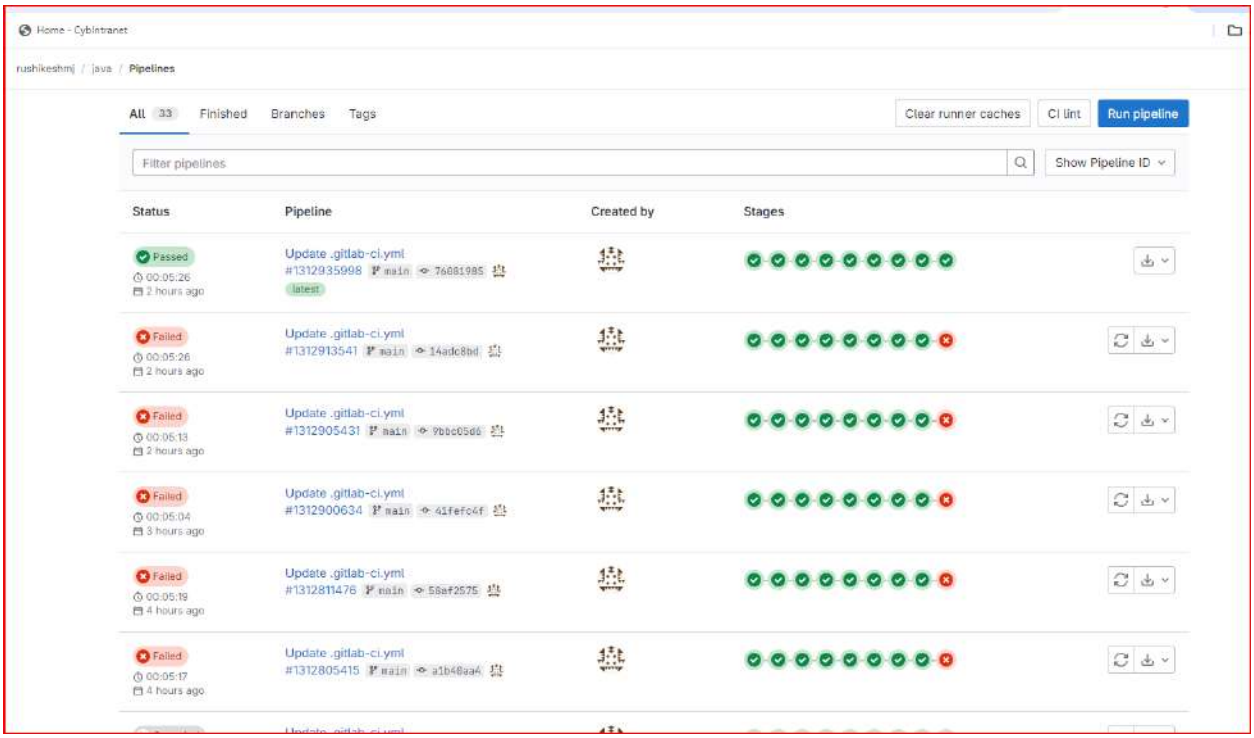


```
scandockerimage:
    stage: scandockerimage
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    script:
        - apt install -y curl
        - curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sh -s -- -b /var/local/bin v0.51.1
        - trivy image $DOCKER_REGISTRY/$APP_NAME:$DOCKER_IMAGE_TAG

#8th stage

pushdockerimage:
    stage: pushdockerimage
    image: registry.gitlab.com/gitlab-org/cloud-deploy/aws-base:latest
    tags:
        - myrunner
    before_script:
        - aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID
        - aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY
        - aws configure set default.region $AWS_DEFAULT_REGION
    script:
        - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $DOCKER_REGISTRY
        - docker push  $DOCKER_REGISTRY/$APP_NAME:$DOCKER_IMAGE_TAG

deployonEc2instance:
    stage: deployonEc2instance
    tags:
        - myrunner
    before_script:
        - pwd && mkdir ~/.ssh
        - echo "$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa
        - chmod 600  ~/.ssh/id_rsa
    script:
        - ssh -o StrictHostKeyChecking=no -i ~/.ssh/id_rsa ubuntu@$INSTANCE_PUBLIC_IP "sudo docker pull $DOCKER_REGISTRY/$APP_NAME:latest"

        - ssh -o StrictHostKeyChecking=no -i ~/.ssh/id_rsa ubuntu@$INSTANCE_PUBLIC_IP "sudo docker run -d -p 8085:8085 $DOCKER_REGISTRY/$APP_NAME:
```

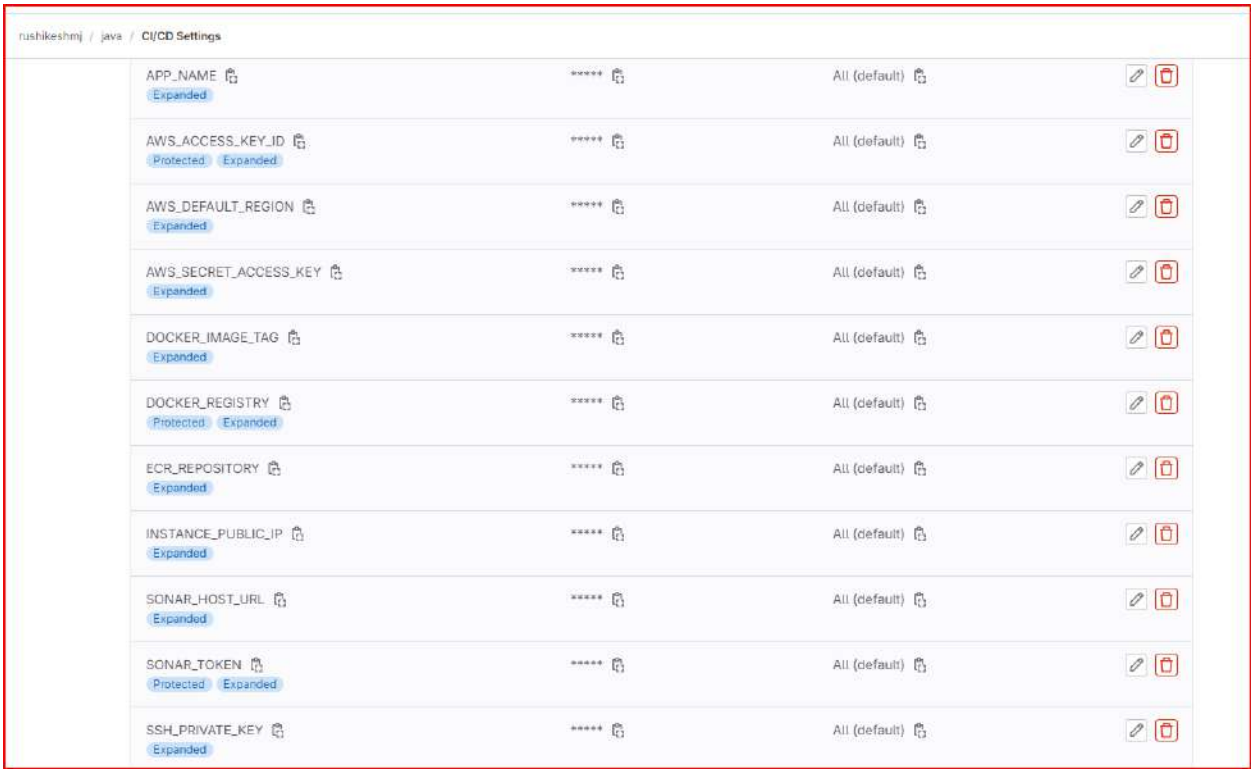


```
Removing target/movie_management-0.0.1-SNAPSHOT.jar
Removing target/movie_management-0.0.1-SNAPSHOT.jar.original
Removing target/site/
Skipping Git submodules setup
Downloading artifacts                                                    00:03
Downloading artifacts for coverage (6986425995)...
Downloading artifacts from coordinator... ok       host=cdn.artifacts.gitlab-static.net id=6986425995 responseStatus=200 OK token=glcbt-65
Downloading artifacts for build (6986426008)...
Downloading artifacts from coordinator... ok       host=cdn.artifacts.gitlab-static.net id=6986426008 responseStatus=200 OK token=glcbt-65
Executing "step_script" stage of the job script                          00:04
Using docker image sha256:db741528e48aaa708f63f80623ffd5b3f6edcd0508dedd00357f869bc0c37ea0 for maven:3.8.5-openjdk-17 with digest maven@sha25
6:3a9c30b3af6278a8ae0097d3a3bf00fff08ec3ed7ae4eb9bfa1772853101549b ...
$ pwd && mkdir ~/.ssh
/builds/rushikeshmj/java
$ echo "$SSH_PRIVATE_KEY" > ~/.ssh/id_rsa
$ chmod 600  ~/.ssh/id_rsa
$ ssh -o StrictHostKeyChecking=no -i ~/.ssh/id_rsa ubuntu@$INSTANCE_PUBLIC_IP "sudo docker pull $DOCKER_REGISTRY/$APP_NAME:latest"
Warning: Permanently added '180.25.149.203' (ECDSA) to the list of known hosts.
latest: Pulling from gitlab
Digest: sha256:1ac8c80030595995acf95ae4d1960822b0fa328efaac00dc52963d4ffc9fc714ec
Status: Image is up to date for 781365762844.dkr.ecr.us-east-1.amazonaws.com/gitlab:latest
781365762844.dkr.ecr.us-east-1.amazonaws.com/gitlab:latest
$ ssh -o StrictHostKeyChecking=no -i ~/.ssh/id_rsa ubuntu@$INSTANCE_PUBLIC_IP "sudo docker run -d -p 8085:8085 $DOCKER_REGISTRY/$APP_NAME:lat
est"
47b78ac11e2f133812989eed1b32841a1d486b2f821cc72fff935c514a63829f
Cleaning up project directory and file based variables                    00:01
Job succeeded

Duration: 11 seconds
Finished: 2 hours ago
Queued: 2 seconds
Timeout: 1h (from project)
Runner: #37387137 (F7KPxy7pk)
172-31-58-192
Tags: myrunner
Commit 76081985
Update .gitlab-ci.yml
Pipeline #1312935988 Passed
deployonEc2instance
Related jobs
deployonEc2instance
```

Complete pipeline



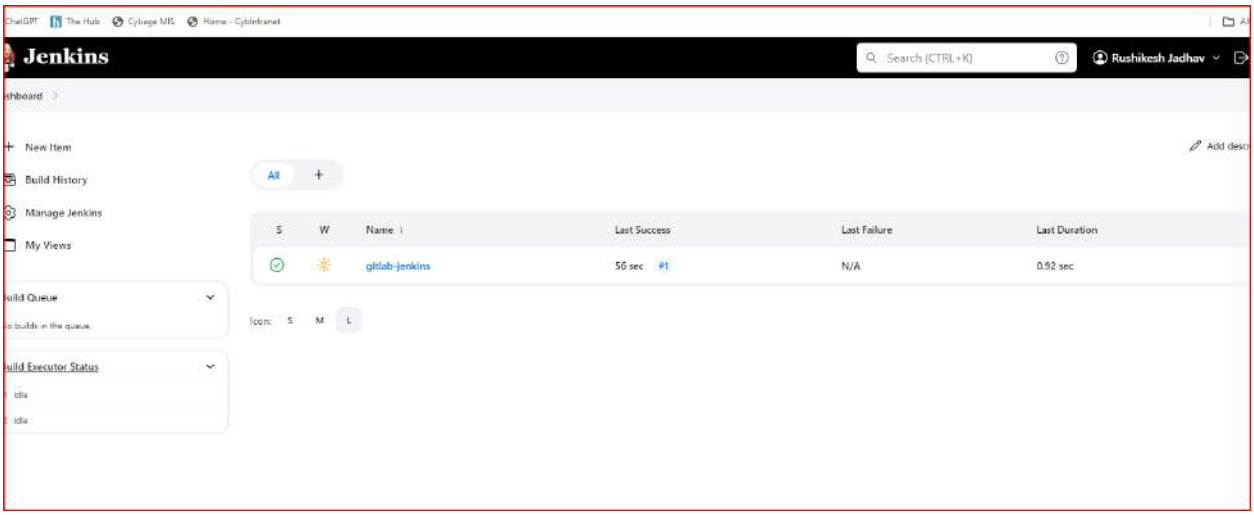
For above pipeline code we need to set variables in settings--CI/CD-- variables



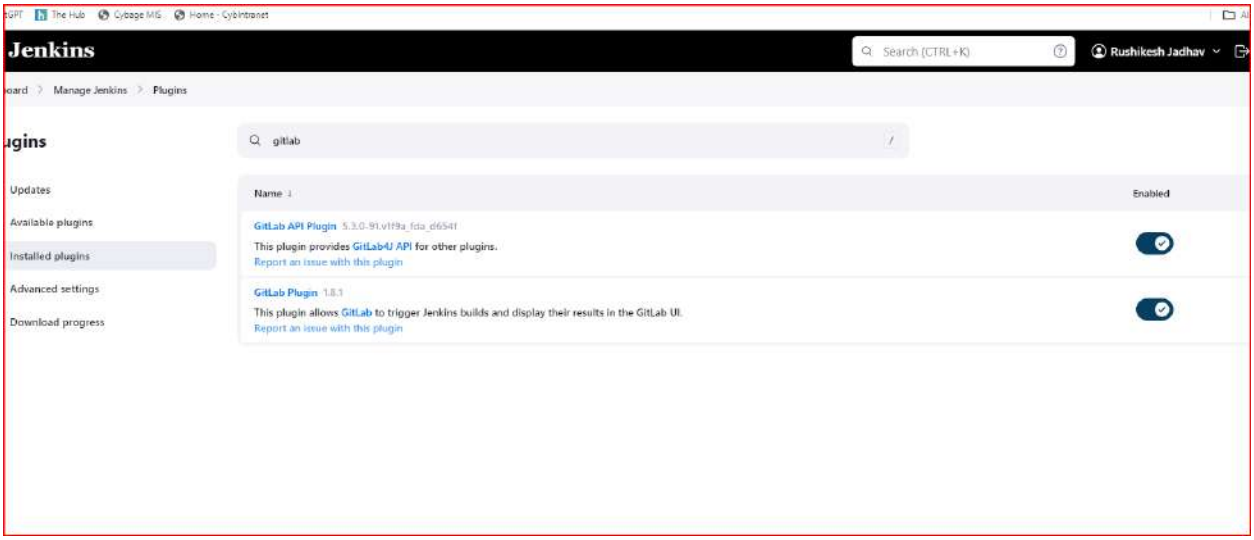
3. Integration of tools

Jenkins GitLab Integration using same GitLab Repo

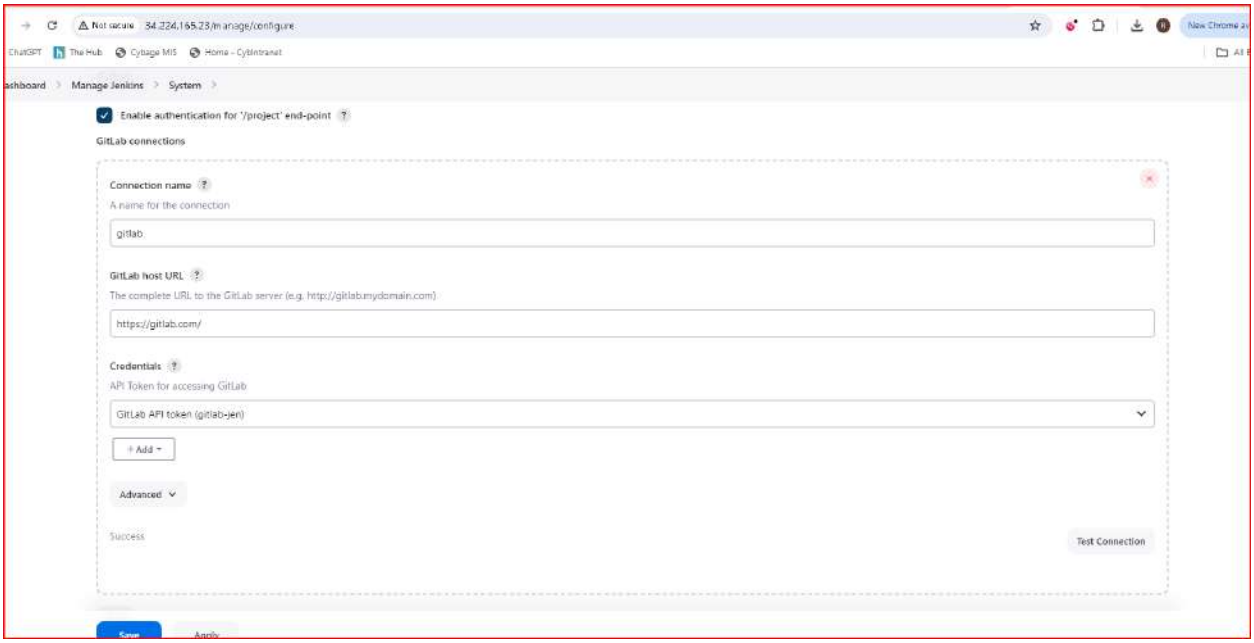
Create Job



Download plugins of gitlab

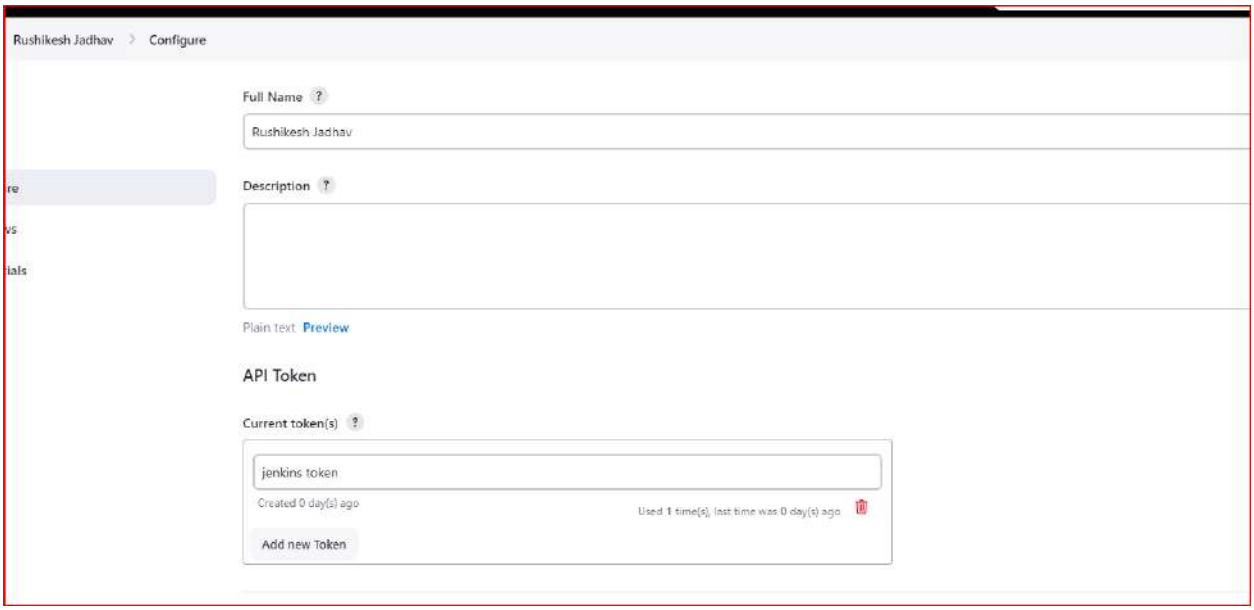


Add Gitlab credentials

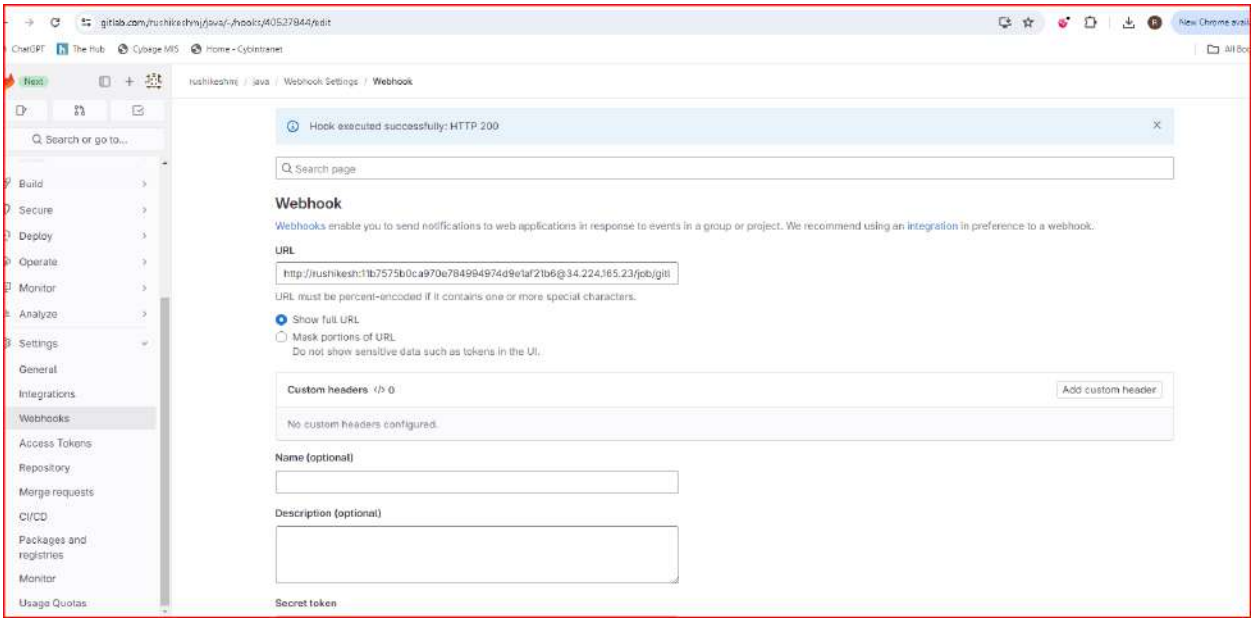
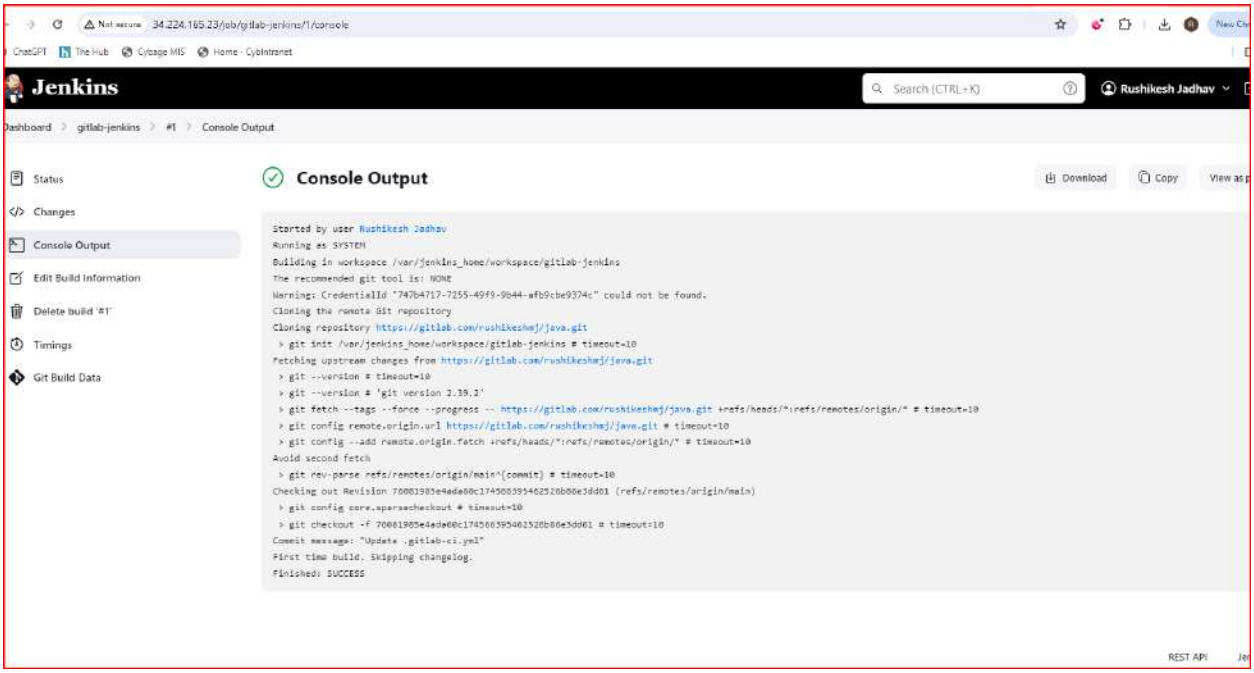


We can see added credentials here

API Token from Jenkins to access webhook



GitLab events trigger webhooks



Console output