DOCKER – JENKINS INTEGRATION – STORING IMAGES

Pre-Requisites:

Step1 – Create a VM Instance with Ubuntu as OS

Step2 – Service Account Permissions:

- Storage Admin
- Storage object user
- Artifact Registry Writer
- Artifact Registry Reader

Step3 – Download the JSON key for this service account

Step4 – Login to VM Instance SSH terminal Install Docker & Jenkins

Docker – Installation:

- sudo apt update -y
- sudo apt install docker.io -y
- docker -v

NOTE: make sure to work as sudo user.

Jenkins Installation: In below given GitHub repo you can find jenkins.sh file for installation.

https://github.com/SaravanaNani/Docker-GCP.git

Note: This GitHub Repo is used in all above given Scenarios

Step5 – Create Repository in Artifact registry.

Step6 – Create Bucket in Google Storage Bucket

Step7 – In VM Instance Console Terminal Run the Below Commands:

- sudo chmod 777 /var/run/docker.sock
- sudo systemctl daemon-reload
- sudo systemctl restart docker.service

1. Building and Pushing Docker Image from Repository to Artifact Registry

How do I write a Jenkins pipeline to build and push a Docker image to Google Cloud Artifact Registry using a Dockerfile in Local Repository?

Step1 – **In VM Instance Terminal - Set password to Jenkins-Change permissions to sudoers file:**

- passwd jenkins

enterpassword:

re-enterpassword:

- Sudoers file Permission:

- chmod 640 /etc/sudoers
- vim /etc/sudoers (add below line and save)
  - jenkins ALL=(ALL:ALL) ALL
- chmod 400 /etc/sudoers (run again after saving above line in the file)

Step2 – Write a sample Dockerfile in root path (Refer GitHub repo link for sample Dockerfile)

Step3 – Login to Jenkins Console and create global credential for Jenkins password and JSON key

Adding above Jenkins password in credentials:

Goto Jenkins Dashboard -> Manage Jenkins -> Credentials -> global -> add credentials -> Kind (secret text) -> Secret (Give password here) -> ID (Give ID) -> Description (Jenkins-password) -> create

Adding VM instance Service Account JSON KEY in credentials:

Goto Jenkins Dashboard -> Manage Jenkins -> Credentials -> global -> add credentials -> Kind (secret file) -> choose file (upload the JSON KEY file) -> ID (Give ID) -> Description (private-key) -> create

Pipeline: Find Pipeline in GitHub Repo or Refer Next page:

Pushing image to Artifact registry from Local (GitHub Repo file name | link)

```
pipeline {
  agent any
    environment {
     MY_PASSWORD = credentials('GIVE_JENKINS_PASSWORD_ID-HERE')
     GOOGLE_APPLICATION_CREDENTIALS = credentials('JSON-KEY-ID') // Jenkins secret FILE ID HERE - GCP service account key
   }
  stages {
    stage('configure') {
      steps {
        sh 'gcloud auth activate-service-account --key-file=$GOOGLE_APPLICATION_CREDENTIALS'
        sh 'gcloud auth configure-docker us-central1-docker.pkg.dev' // CHANGE THE REGION
      }
    }
    stage ('copy Dockerfile to jenkins workspace') {
      steps {
        sh 'echo $MY_PASSWORD | sudo -S cp /root/Dockerfile .'
      }
    }
    stage('build') {
      steps {
        sh 'docker build -t us-central1-docker.pkg.dev/$PROJECT_ID/$ARTIFACT-REPO-NAME/img:v${BUILD_NUMBER} .'  // Change region, repository, image
      }
    }
    stage ('pushing image to registry') {
      steps {
       sh 'docker push us-central1-docker.pkg.dev/$PROJECT_ID/$ARTIFACT-REPO-NAME/img:v${BUILD_NUMBER}'    // Change region, repository, image
      }
    }
  }
}
```

2. Checkout, Build & Push Docker Images from GitHub to Artifact Registry

How do I configure a Jenkins pipeline to check out a Dockerfile from GitHub, build the Docker image, and push it to Google Cloud Artifact Registry?

Step1 – Write a sample Dockerfile in GitHub Repository (Refer GitHub repo link for sample Dockerfile)

Step2 – Login to Jenkins Console and create global credential for JSON key

Adding VM instance Service Account JSON KEY in credentials:

Goto Jenkins Dashboard -> Manage Jenkins -> Credentials -> global -> add credentials -> Kind (secret file) -> choose file (upload the JSON KEY file) -> ID (Give ID) -> Description (private-key) -> create

Pipeline: Find Pipeline in GitHub Repo or Refer Next page:

Pushing Image to Artifact registry from GIT Repo (GitHub Repo file name | link)

```
pipeline {
   agent any

     environment {
       GOOGLE_APPLICATION_CREDENTIALS = credentials('JSON-KEY-ID') // Jenkins secret FILE ID HERE - 
GCP service account key
    }

   stages {
      stage('Checkout') {
         steps {
            git branch: 'main', url: 'https://github.com/SaravanaNani/Docker-GCP.git'
         }
      }

   stages {
      stage('configure') {
         steps {
            sh 'gcloud auth activate-service-account --key-file=$GOOGLE_APPLICATION_CREDENTIALS'
            sh 'gcloud auth configure-docker us-central1-docker.pkg.dev' // CHANGE THE REGION
         }
      }

      stage('build') {
         steps {
            sh 'docker build -t us-central1-docker.pkg.dev/$PROJECT_ID/$ARTIFACT-REPO-
NAME/img:v${BUILD_NUMBER} .'  // Change region, repository, image
         }
      }
      stage ('pushing image to registry'){
         steps {
         sh 'docker push us-central1-docker.pkg.dev/$PROJECT_ID/$ARTIFACT-REPO-
NAME/img:v${BUILD_NUMBER}'    // Change region, repository, image
         }
      }
   }
}
```

Note: For Clear Pipeline Syntax Refer the GitHub Repo

3. Pulling Docker Images from Artifact Registry Using Jenkins

What steps are involved in writing a Jenkins pipeline to pull a Docker image from Google Cloud Artifact Registry?

Pipeline: Find Pipeline in GitHub Repo or below in this page:

Pulling Image from Artifact Repo through Pipeline (GitHub Repo file name | link)

```
pipeline {
  agent any

  stages {
    stage ('Pulling image') {
      steps {
        sh 'docker pull us-central1-docker.pkg.dev/$PROJECT_ID/$REPO_NAME/$IMAGE:TAG' // Change
region, repository, image
      }
    }
    stage ('tagging image to small name') {
      steps {
        sh 'docker tag us-central1-docker.pkg.dev/$PROJECT_ID/$REPO_NAME/$IMAGE:TAG' $IMG:TAG' //
Here we are converting image name from big name to small name
      }
    }
    stage ('Deploying in container') {
      steps {
        sh 'docker run -itd --name $CONTAINER_NAME -p 8084:80 $IMAGE:TAG'
      }
    }
  }
}
```

4. Building and Storing Docker Images in Google Cloud Storage

How can I write a Jenkins pipeline to build a Docker image and store it in a Google Cloud Storage bucket?

Step1 – Install plugin (Google Cloud Storage)

- Goto Manage Jenkins -> Plugins -> Install **Google Cloud Storage**

Step2 – Login to Jenkins Console and create global credential for JSON key

Adding VM instance Service Account JSON KEY in credentials:

Goto Jenkins Dashboard -> Manage Jenkins -> Credentials -> global -> add credentials -> kind (Google Service Account from Private Key) -> Project ID -> Select - JSON Key. -> ID (Give ID) -> Description (private-key) -> create

Note: Credential Kind will be Changed to Storage Bucket

Stage (Push) - Pipeline syntax:

-> Sample Step (googleStorageUpload:Google Storage Classic Upload)
-> File Pattern (image*.tar) -> Storage Location (Give Storage Bucket Name)
-> Generate Pipeline and use in pipeline code (final stage)

Pipeline: Find Pipeline in GitHub Repo or Refer Next page:

Pushing Docker Image to Storage Bucket (GitHub Repo file name | link)

PIPELINE CODE EXPLAINATION:

- STAGE1: Checkout.
- STAGE2: Building image.
- STAGE3: Deleting all ".tar" files in Jenkins Workspace.
- STAGE4: Converting image to ".tar" file.
- STAGE5: Pushing to bucket:

```
pipeline {
  agent any

  stages {
    stage('checkout') {
      steps {
        git branch: 'main', url: 'https://github.com/SaravanaNani/Docker-GCP.git'
      }
    }
    stage('build') {
      steps {
        sh 'docker build -t img:v${BUILD_NUMBER} .'
      }
    }
    stage ('delete all .tar files') {
      steps {
        sh 'rm -rf *.tar'
      }
    }
    stage ('image to tar') {
      steps {
        sh 'docker save -o images${BUILD_NUMBER}.tar img:v${BUILD_NUMBER}'
      }

    }
    stage ('pushing to bucket') {
      steps {
        googleStorageUpload bucket: 'gs://BUCKET-NAME', credentialsId: 'JSON-KEY-ID', pattern:
'image*.tar'
      }
    }

  }
}
```

5. Pulling Docker Images from Google Cloud Storage

How do I create a Jenkins pipeline to pull a Docker image from a Google Cloud Storage bucket?

Step1 – Install plugin (Google Cloud Storage)

- Goto Manage Jenkins -> Plugins -> Install **Google Cloud Storage**

Step2 – Login to Jenkins Console and create global credential for JSON key

Adding VM instance Service Account JSON KEY in credentials:

Goto Jenkins Dashboard -> Manage Jenkins -> Credentials -> global -> add credentials -> kind (Google Service Account from Private Key) -> Project ID -> Select - JSON Key. -> ID (Give ID) -> Description (private-key) -> create

Note: Credential Kind will be Changed to Storage Bucket
Note: Pipeline Syntax Sample step will be changed for Pulling

Stage (Push) - Pipeline syntax:

-> Sample Step **(googleStorageDownload:Google Storage Download)**
-> Object to download (gsutil URL) -> Local Directory (.) -> Generate Pipeline Script

Pipeline: Find Pipeline in GitHub Repo or Refer Next page:

Pulling Image from Storage Bucket (GitHub Repo file name | link)

```
pipeline {
  agent any
    stages{
      stage('pulling image from GCS'){
        steps{
          googleStorageDownload bucketUri: '$OBJECT - gsutil URI', credentialsId: 'JSON-KEY-ID', localDirectory: '.'
        }
      }
      stage('Untar the image') {
        steps {
          sh 'docker load -i $IMAGE_FILE.tar' //IMAGE_FILE: Your file name (object name) in the bucket.
        }
      }
      stage('Deploying to Container'){
        steps{
          sh 'docker run -itd --name $CONTAINER_NAME -p 8083:8080 IMAGE:TAG' //IMAGE:TAG: Untar Image_name:tag
      }
    }
  }
}
```