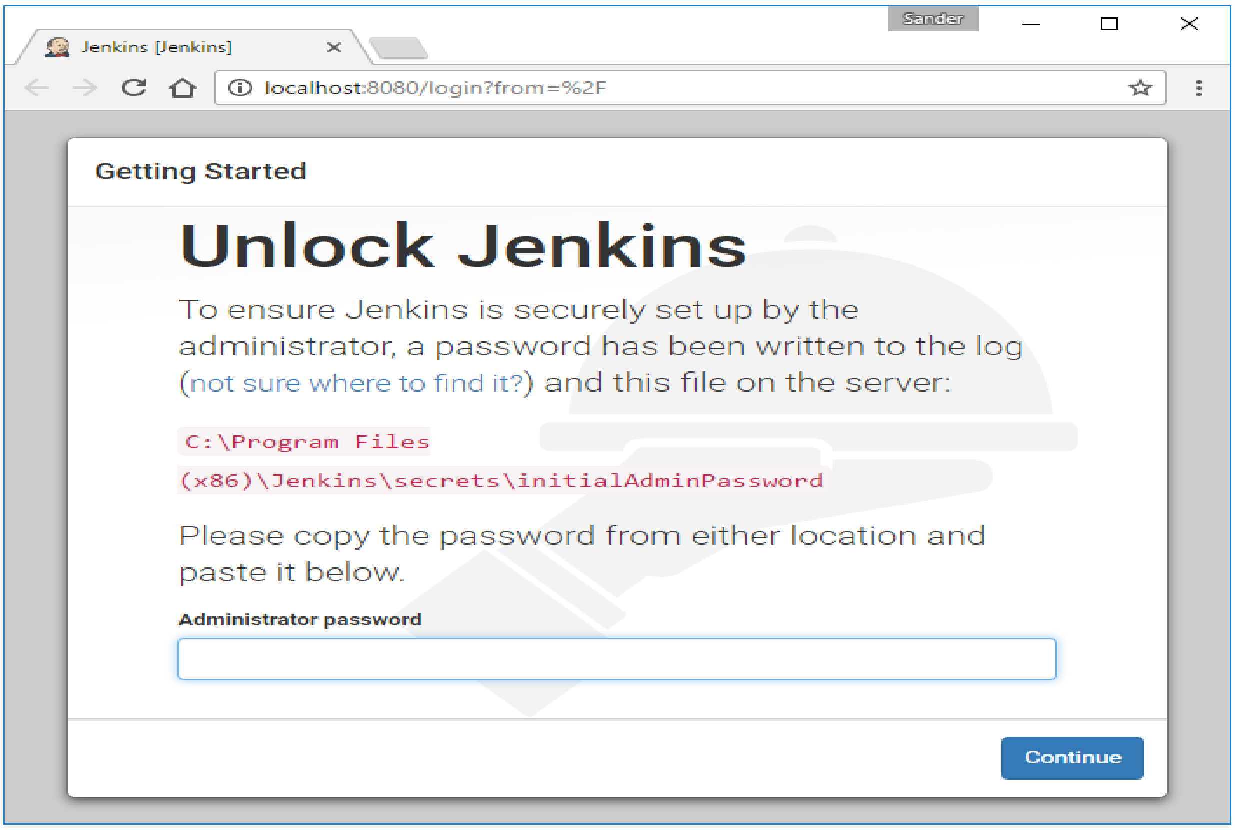
PUSH DOCKER IMAGE TO THE ECR USING JENKINS

Pre-Requirements :

In server(Security groups) port 8080 has to be enabled

1. sudo yum update
2. sudo yum install java-17* -y
3. sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
4. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
5. sudo yum install Jenkins
6. jenkins –version
7. sudo systemctl enable Jenkins
8. sudo systemctl start Jenkins
9. sudo systemctl status Jenkins
10. sudo cat /var/lib/jenkins/secrets/initialAdminPassword



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword
```

Please copy the password from either location and paste it below.

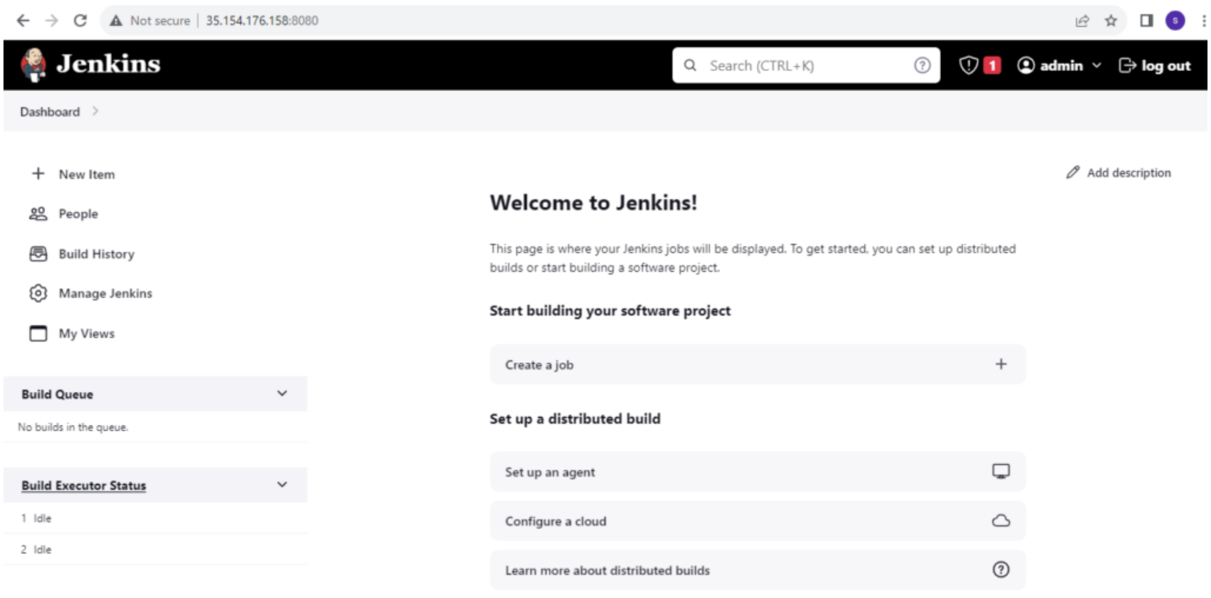
Administrator password

Copy the password from the Linux server and paste it in the above page (Administrator password).

Configure the Jenkins setup (name, mail ID, password etc.)

1. **Install Docker:** sudo yum install docker –y
2. **Start Docker Service:** sudo service docker start
3. **Enable Docker Service:** sudo systemctl enable docker
4. **Add the User to the Docker Group:** sudo usermod -aG docker ec2-user
5. **Adjust Docker Socket Permissions (Optional):** sudo chmod 666 /var/run/docker.sock
6. **Restart Jenkins:** sudo systemctl restart Jenkins
7. **Check Jenkins Status:** sudo systemctl status Jenkins
8. **Download and Install AWS CLI Version 2:**
   curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
   sudo yum install -y unzip
   sudo unzip awscliv2.zip
   sudo ./aws/install

Jenkins Console :



Not secure | 35.154.176.158:8080

Jenkins

Dashboard

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

Build Queue

No builds in the queue.

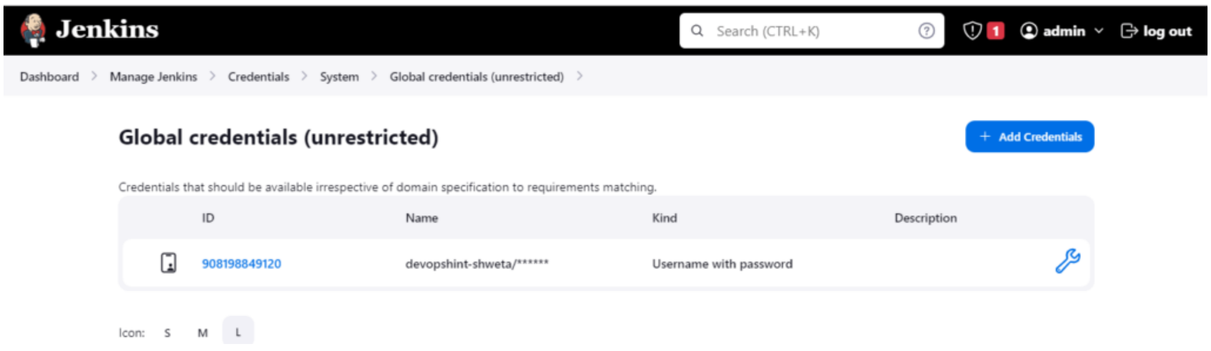
Build Executor Status

1 Idle

2 Idle

Add AWS Credentials in Jenkins:

GO to the Manage Jenkins>>Credentials>>system>>Global credentials


Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
908198849120	devopshint-shweta/******	Username with password	

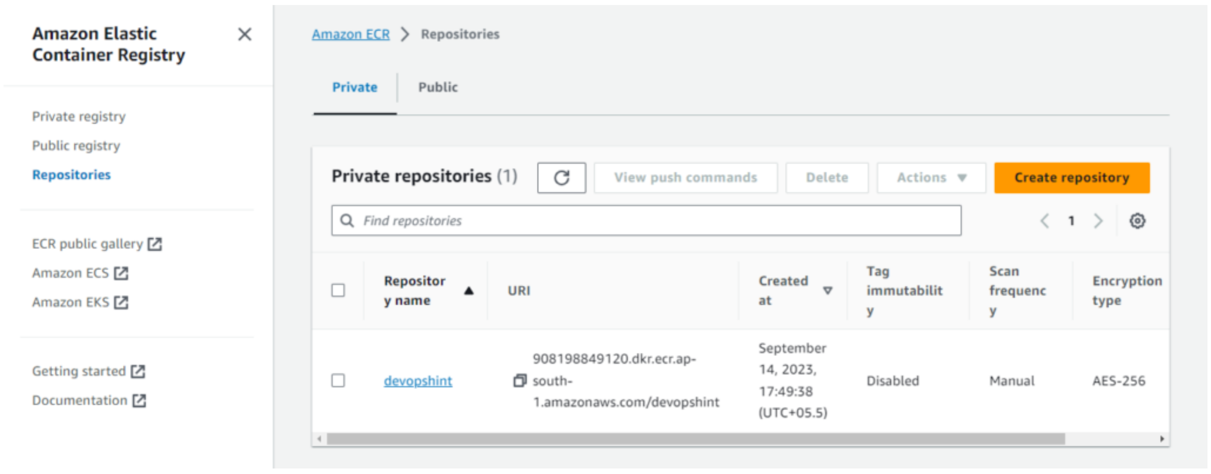
Then add credentials and here add AWS username and password and account ID

Download the plugins in Jenkins :

- Docker Plugin
- Docker pipeline
- Amazon ECR plugin

Create ECR Repository in AWS Account:

Go to AWS Account>>search "ECR">>create

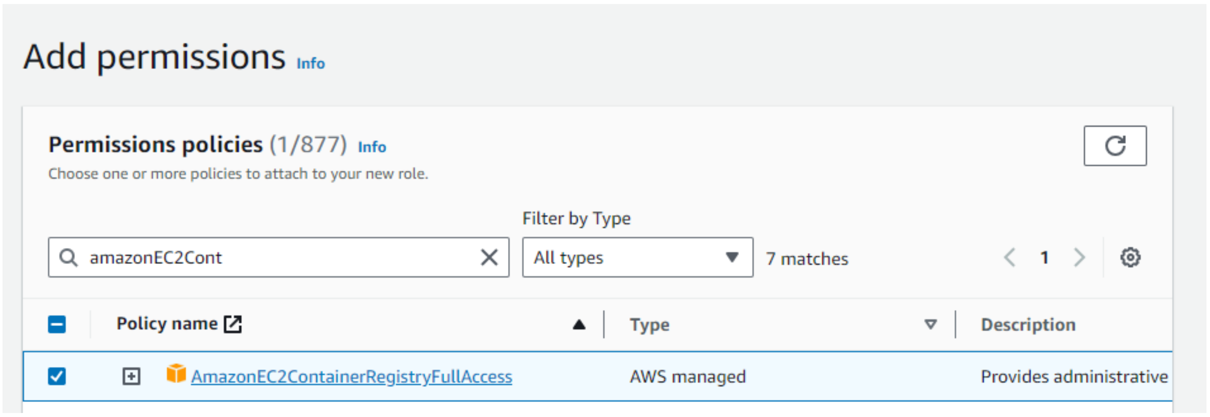

Amazon ECR > Repositories

Private repositories (1)

Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type
devopshint	908198849120.dkr.ecr.ap-south-1.amazonaws.com/devopshint	September 14, 2023, 17:49:38 (UTC+05.5)	Disabled	Manual	AES-256

Create IAM Role in AWS:

Create IAM Role with the Permissions "AmazonEC2ContainerRegistryFullAccess"



Add permissions Info

Permissions policies (1/877) Info

Choose one or more policies to attach to your new role.

Filter by Type: All types

amazonEC2Cont — 7 matches

Policy name	Type	Description
✓ AmazonEC2ContainerRegistryFullAccess	AWS managed	Provides administrative

Build Pipeline code in Jenkins:

To create jenkins pipeline go to the Jenkins Dashboard>> “new Item”>> give name and select “Pipeline”>> paste the below code in “pipeline script”

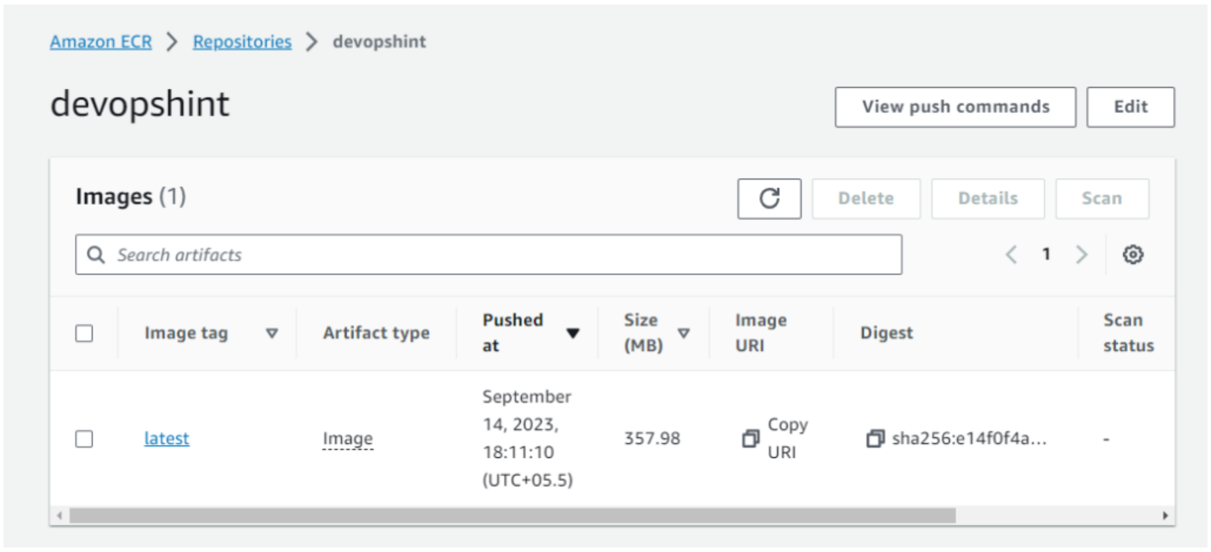
CODE:

```
pipeline {

  agent any

  environment {

    AWS_ACCOUNT_ID="891377261650"

    AWS_DEFAULT_REGION="ap-southeast-2"

    IMAGE_REPO_NAME="devopshint"

    IMAGE_TAG="v1"

    REPOSITORY_URI = "891377261650.dkr.ecr.ap-southeast-2.amazonaws.com/devopshint"

  }


  stages {


     stage('Logging into AWS ECR') {

       steps {

         script {
                sh """aws ecr get-login-password --region ${AWS_DEFAULT_REGION} | docker login --
username AWS --password-stdin
${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_DEFAULT_REGION}.amazonaws.com"""

                }



            }

        }



        stage('Cloning Git') {

            steps {

                git branch: 'main', url: 'https://github.com/premchandkakke/reactjs-app.git'

            }

        }



    // Building Docker images

    stage('Building image') {

      steps{

        script {

          dockerImage = docker.build "${IMAGE_REPO_NAME}:${IMAGE_TAG}"

        }

      }

    }



    // Uploading Docker images into AWS ECR

    stage('Pushing to ECR') {

     steps{

         script {

                sh """docker tag ${IMAGE_REPO_NAME}:${IMAGE_TAG}
${REPOSITORY_URI}:$IMAGE_TAG"""
```

```
            sh """docker push
${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_DEFAULT_REGION}.amazonaws.com/${IMAGE_REPO_NAME}:
${IMAGE_TAG}"""

        }

       }

     }

   }

}
```

Note: Modify the credentials and “Apply” and “save” >>”Build”

Check ECR Repository weather image is pushed or not:



devopshint

Images (1)

	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status
	latest	Image	September 14, 2023, 18:11:10 (UTC+05.5)	357.98	Copy URI	sha256:e14f0f4a...	-