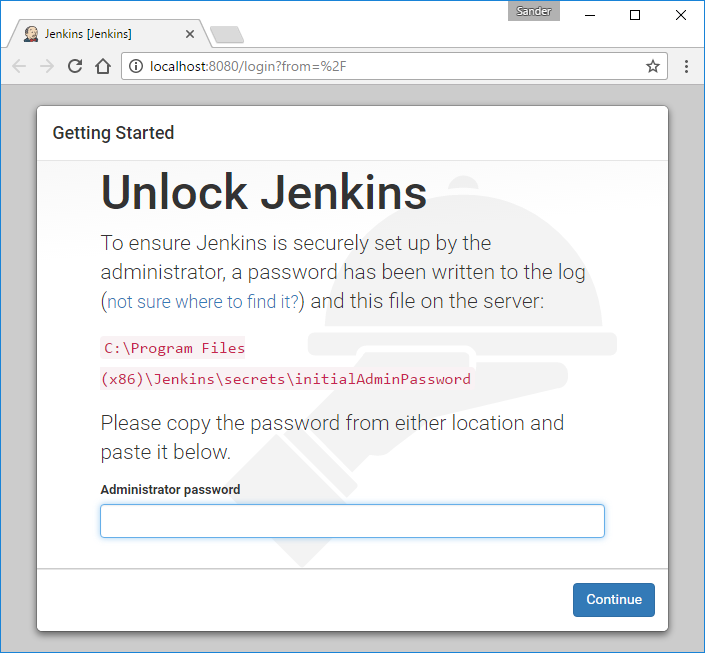
**PUSH DOCKER IMAGE TO THE ECR USING JENKINS**

**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



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:**

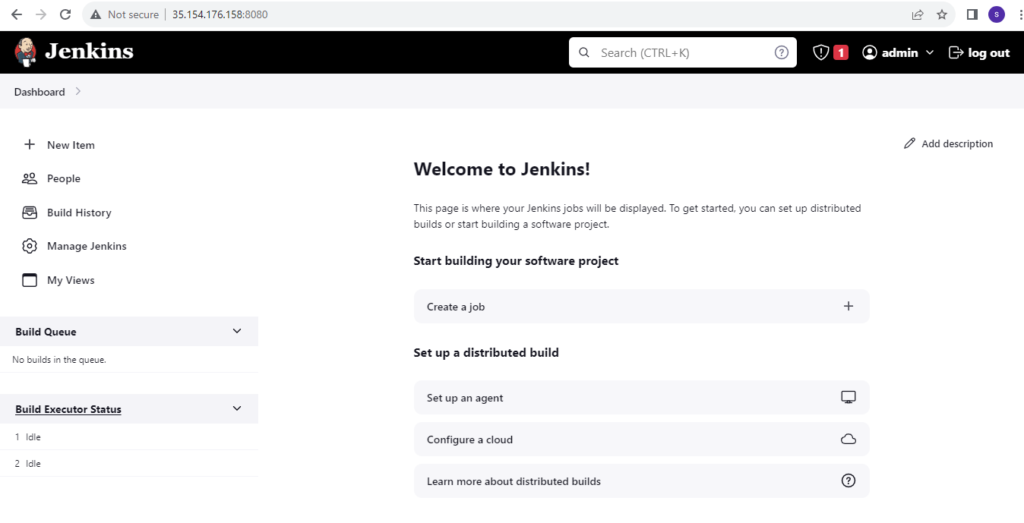
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

sudo yum install -y unzip

sudo unzip awscliv2.zip

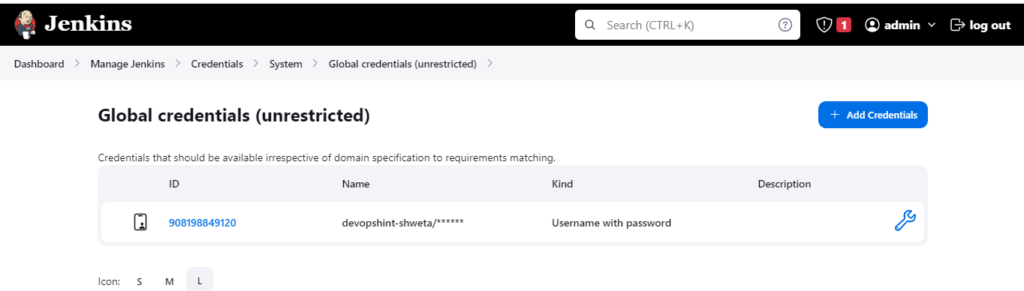
sudo ./aws/install

**Jenkins Console :**



**Add AWS Credentials in Jenkins:**

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



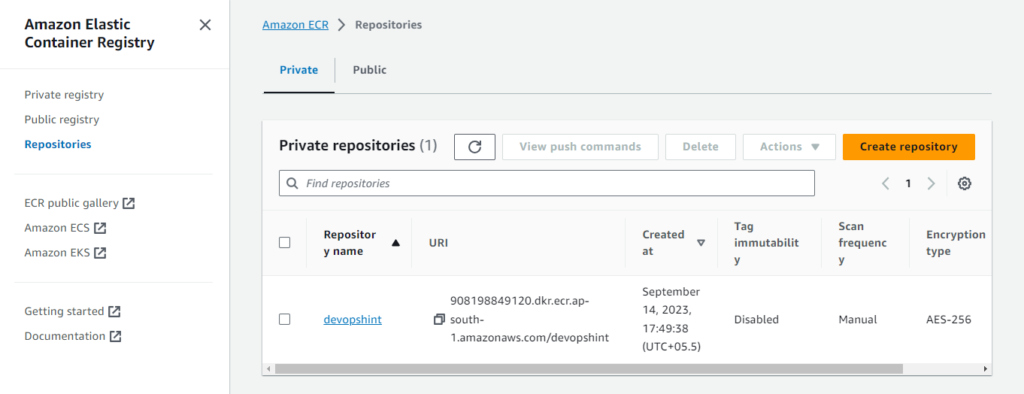
Then add credentials and here add AWS username and password and account ID

**Download the plugins in Jenkins :**

* Docker Plugin
* Docker pipeline
* Amazon ECR plugin

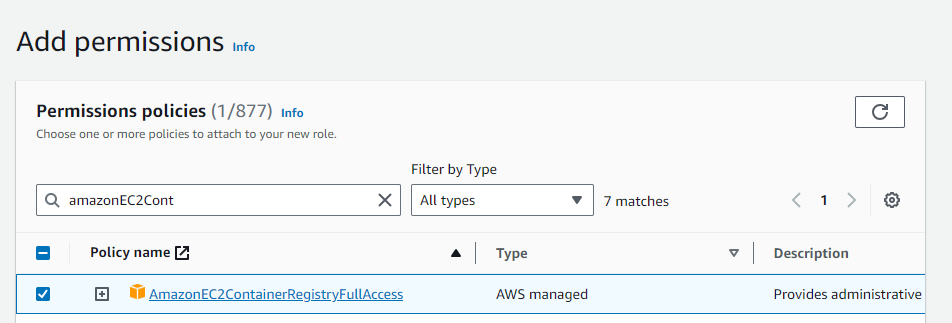
**Create ECR Repository in AWS Account:**

Go to AWS Account>>search “ECR”>>create



**Create IAM Role in AWS:**

Create IAM Role with the Permissions “AmazonEC2ContainerRegistryFullAccess”



**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:**

