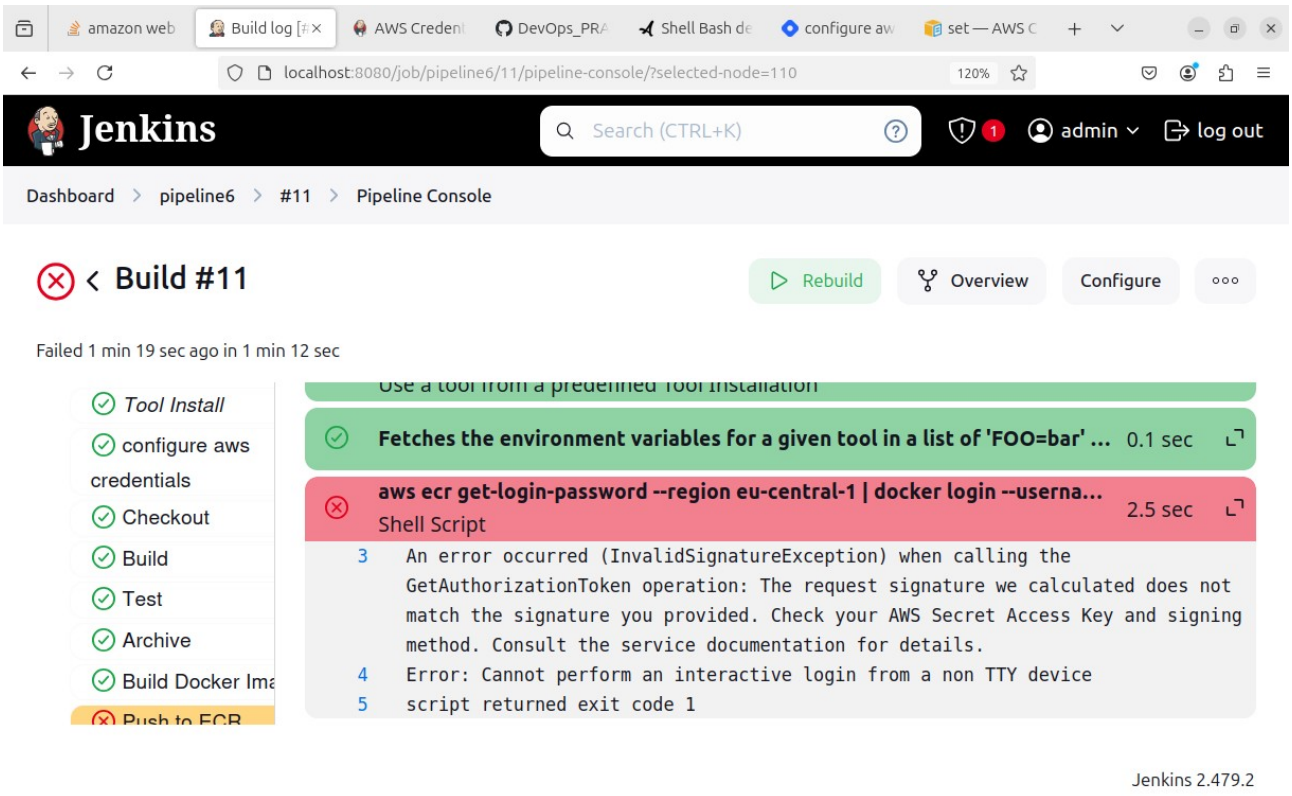
Hemos llegado hasta el pipeline del Jenkins pero este nos da error al intentar validar contra el AWS para subir la imagen creada. Hemos instalado el cliente de AWS en el bash del jenkins y entrado las credenciales y tambiénn instalado el plugin del AWS Credentials:




```
´pipeline {
  agent any

  environment {
    AWS_ACCOUNT_ID="AKIAWLRRXXWRFZ44RZ4K"
    AWS_DEFAULT_REGION="eu-central-1"
    IMAGE_REPO_NAME="spring-boot-app"
    IMAGE_TAG="${env.BUILD_ID}"
    REPOSITORY_URI = "${AWS_ACCOUNT_ID}.dkr.ecr.$
{AWS_DEFAULT_REGION}.amazonaws.com/${IMAGE_REPO_NAME}"
  }

  tools {
    maven 'M3'
    jdk 'JDK21'
  }

  stages {

    stage('configure aws credentials') {
      steps {
        withCredentials([[
        $class: 'AmazonWebServicesCredentialsBinding',
```

```
            accessKeyVariable: 'AWS_ACCESS_KEY_ID',
            secretKeyVariable: 'AWS_SECRET_ACCESS_KEY',
            credentialsId: 'credentials_aws']])
            {
               script
               {
                  sh 'aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID'
                  sh 'aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY'
                  sh 'aws configure set default.region eu-central-1'
                  //sh 'aws s3 ls'
               }
            }
         }
      }


      // Previous stages (build, test) here

      stage('Checkout') {
         steps {
            git 'https://github.com/AlbertProfe/BooksPageable.git'
         }
      }


      stage('Build') {
         steps {
            sh 'mvn clean package'
         }
      }

      stage('Test') {
         steps {
            sh 'mvn test'
         }
      }

      stage('Archive') {
         steps {
            archiveArtifacts artifacts: '**/target/*.jar', fingerprint: true
         }
      }

      stage('Build Docker Image') {
         steps {
            script {
               // Build the Docker image and tag it with both BUILD_NUMBER and latest
               sh "docker build -t ${IMAGE_REPO_NAME}:${IMAGE_TAG} -t $
{IMAGE_REPO_NAME}:latest ."
            }
         }
      }
```

```
        stage('Push to 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"
                    sh "docker tag ${IMAGE_REPO_NAME}:${IMAGE_TAG} ${REPOSITORY_URI}:${IMAGE_TAG}"
                    sh "docker push ${REPOSITORY_URI}:${IMAGE_TAG}"
                }
            }
        }

        stage('Deploy to ECS') {
            steps {
                script {
                    sh "aws ecs update-service --cluster your-cluster-name --service your-service-name --force-new-deployment"
                }
            }
        }
    }
}
```