Jenkins Capstone Project #1 - Creating a Jenkins Continuous Integration (CI) Pipeline To Trigger Automatically on GitHub Pull Requests

1. **Install Jenkins on AWS : follow documentation step https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/**
2. **Install Github pull request builder plugin and configure it as described https://plugins.jenkins.io/ghprb/**
3. **Add Github repo webhook for your repo**
4. **Add branch protection**
5. **Create your pipeline in Jenkins and add your Jenkins file.**

```
pipeline{

  agent{
     label 'aws-agent'
  }

  stages{

    stage('build'){
      steps{
        script{
          sh 'mvn clean package'
        }
      }
    }

    stage(test){
      steps{
        script{
          sh 'mvn test'
        }
      }
    }
}
}
```

Jenkins Capstone Project #2 - Creating a Jenkins Continuous Delivery (CD) Pipeline To Deploy A Java App on AWS EKS

1. **Make sure that your application has : deployment files and Docker file**
2. **Create credentials for Dockerhub in Jenkins**
3. **Launch EKS and get your credentials Access and secret Key**
4. **Install pipeline : aws steps plugin in Jenkins**
5. **Setup AWS CLI credentials in Jenkins**
6. **Create CD pipline with steps in Jenkins and add your steps for deployment in Jenkins file https://plugins.jenkins.io/pipeline-aws/#plugin-content-withaws**
7. **Make sure your agent has Docker, AWSCLI and aws-iam-authenticator installed**

```
pipeline{
   agent{
      label 'aws-agent'
   }
   stages{
      stage('build'){
         steps{
            script{
               sh 'docker build -t java-app .'
            }
         }
      }

 agent{
 }

 stages{

   stage('build'){
     steps{
       script{
         sh 'mvn clean package'
       stage('push'){
          steps{
             script{
                withCredentials([usernamePassword(credentialsId: 'docker-hub',
passwordVariable: 'Password', usernameVariable: 'Username')]) {
                sh 'docker login --username $Username --password $Password'
                sh 'docker tag java-app $Username/java-app'
                sh 'docker push $Username/java-app'
```

```
                }
            }
        }
    }

    stage('deploy'){
        steps{
            script{
                withAWS(credentials: 'aws-cli', region: 'us-east-2') {
                sh 'aws eks update-kubeconfig --region us-east-2 --name eks'
                sh 'kubectl apply -f ./k8s/deployment.yaml'
                }
            }
        }
    }
  }
}
```

Note:

Kubectl Issue in session:

- Ref: https://github.com/aws/aws-cli/issues/6920, Issue was related to aws cli version after uninstalling it and install version >= 2.11 it works