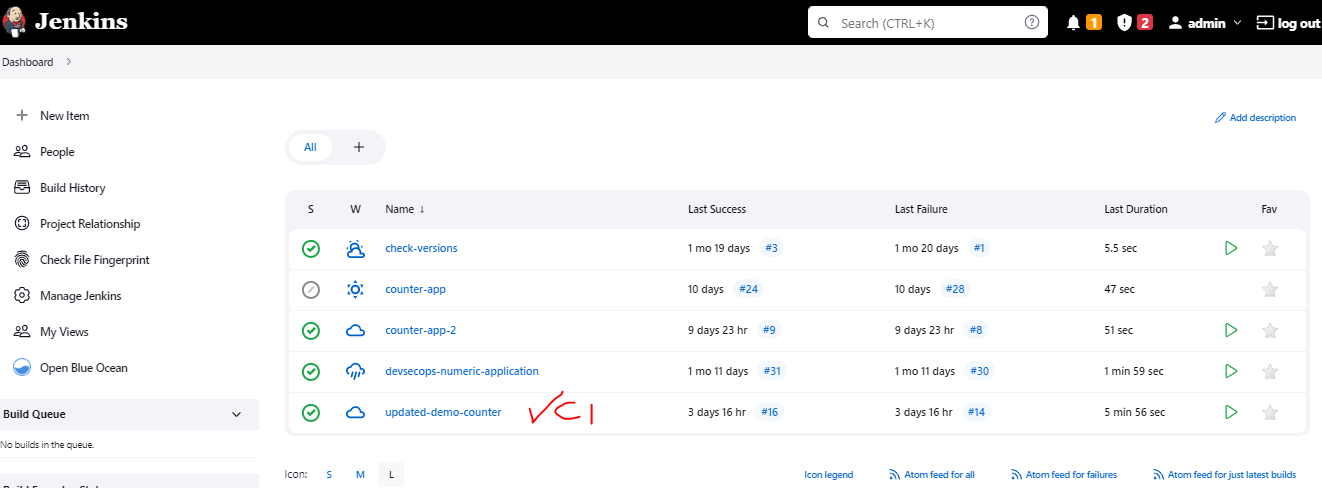
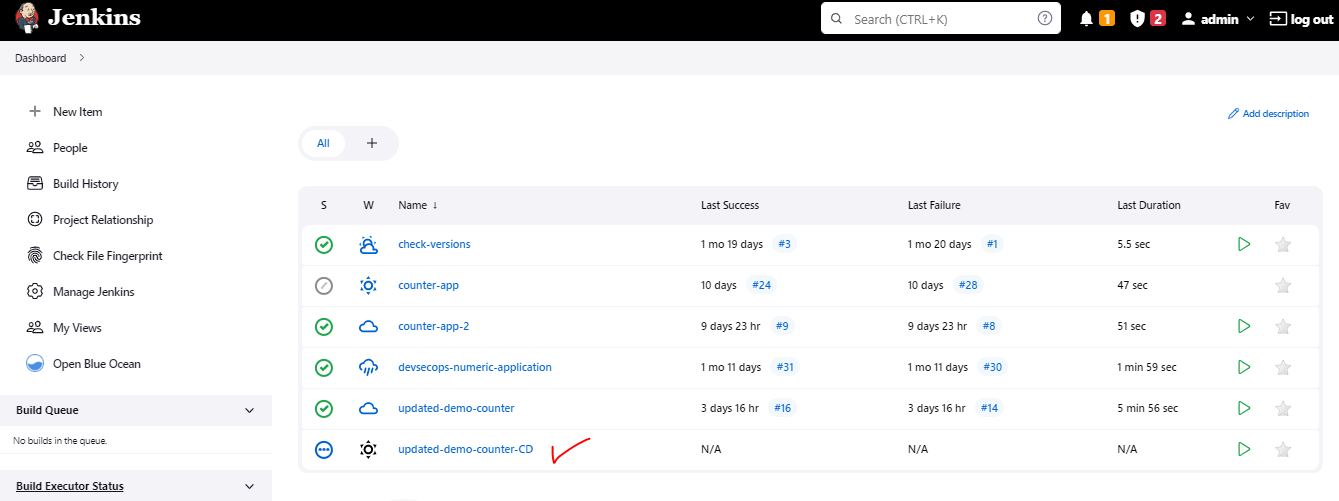
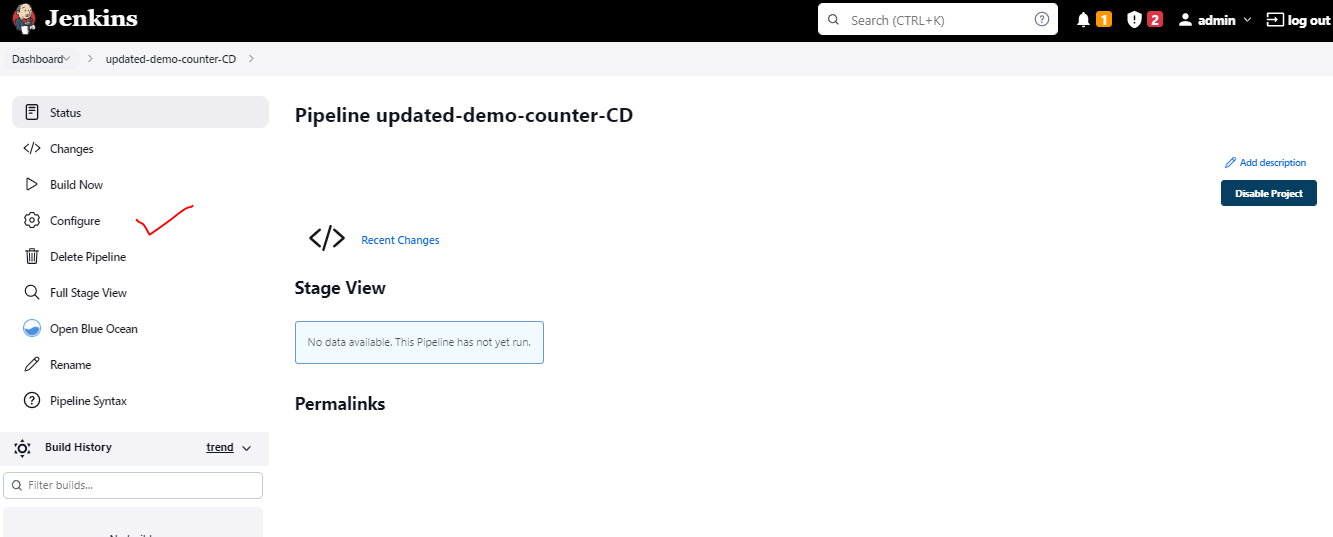
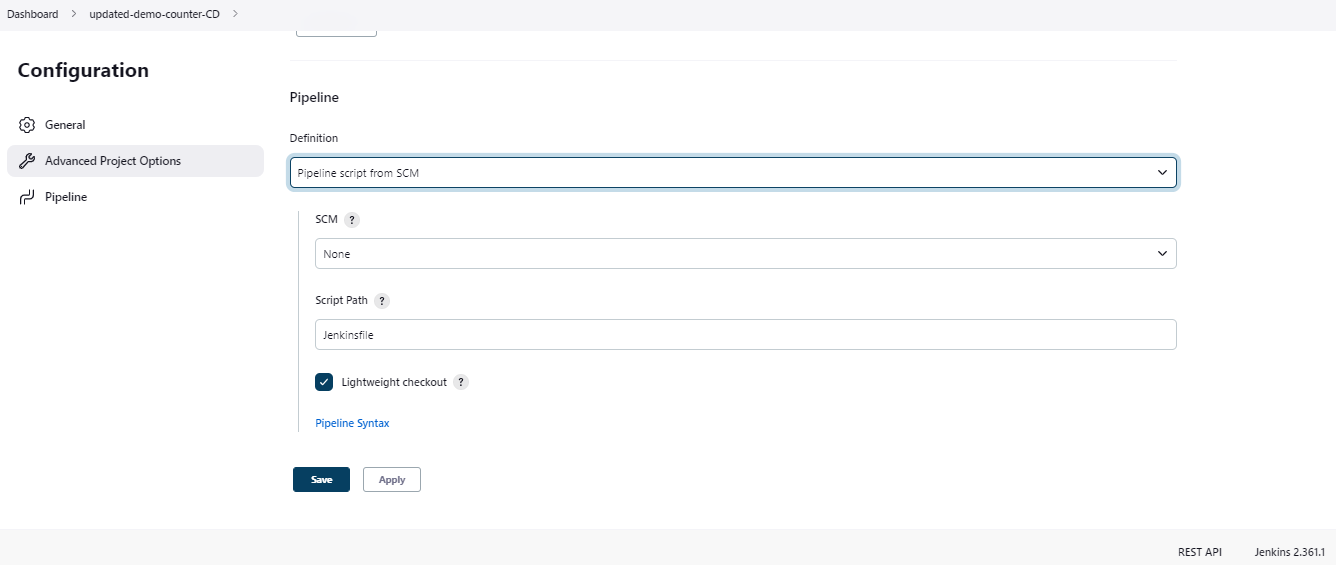
Login to Jenkins

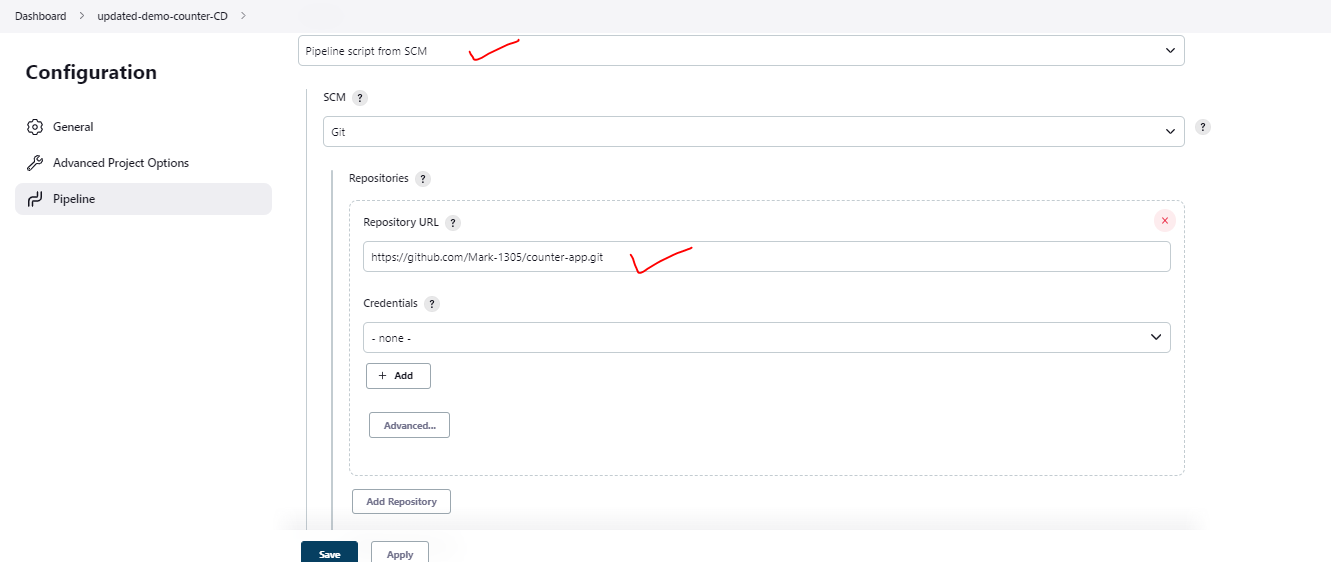


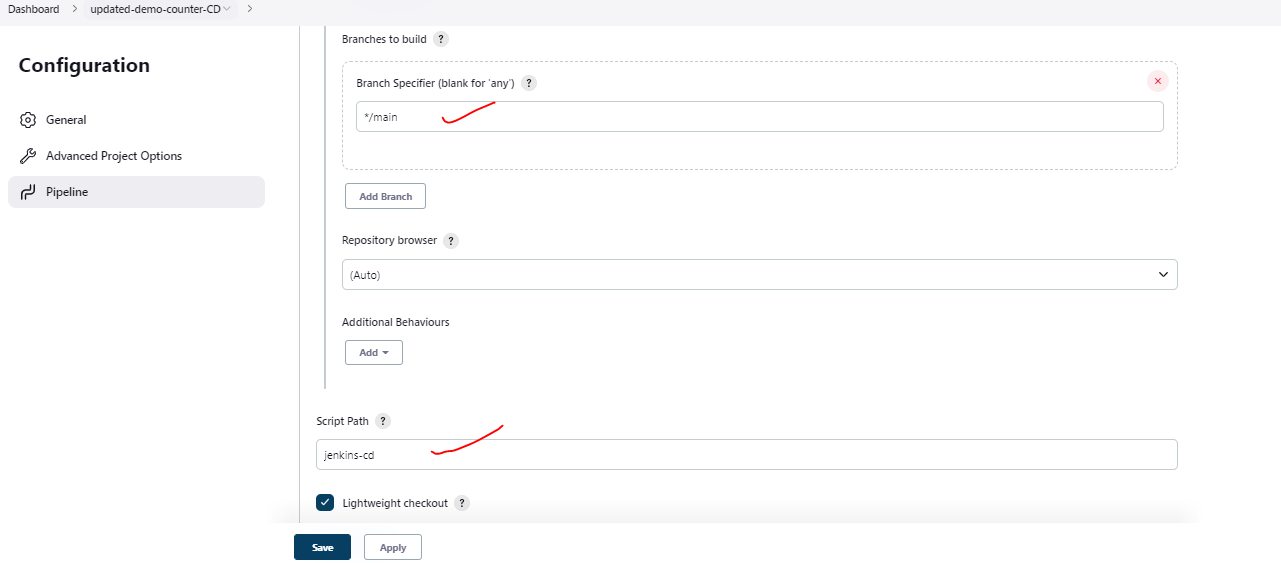
Create New Pipeline



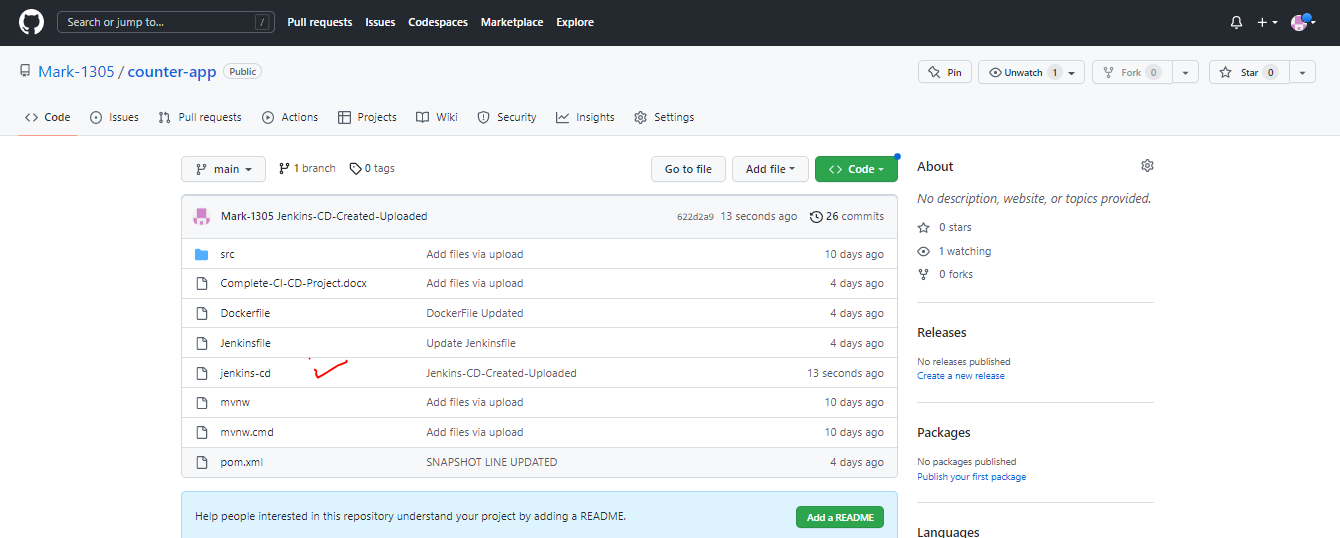








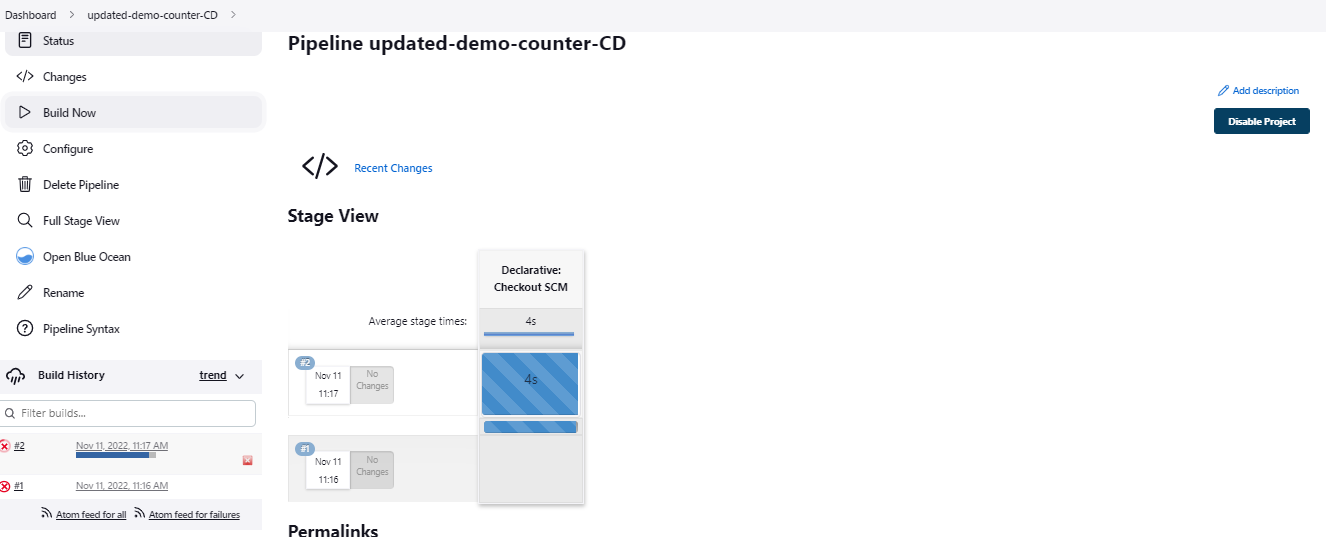
Jenkins CD file Created and Uploaded

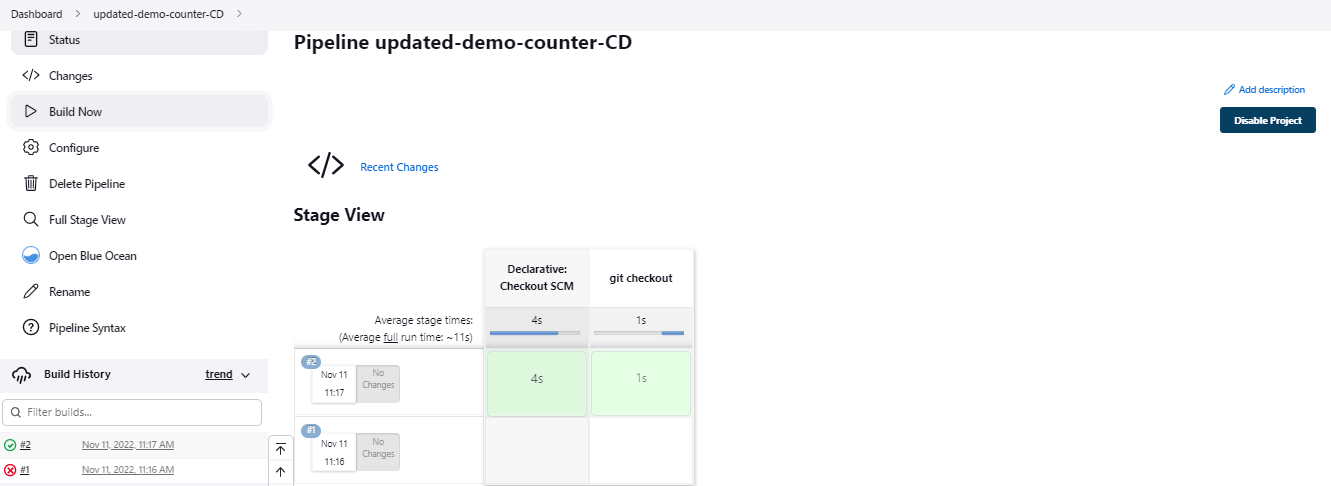


Code:

|  |
| --- |
| pipeline {  agent any  stages{  stage('git checkout'){  steps{  git branch 'main', url: "https://github.com/Mark-1305/counter-app.git"  }  }  }  } |

Let’s Run the Pipeline and See if the Git Checkout is happening



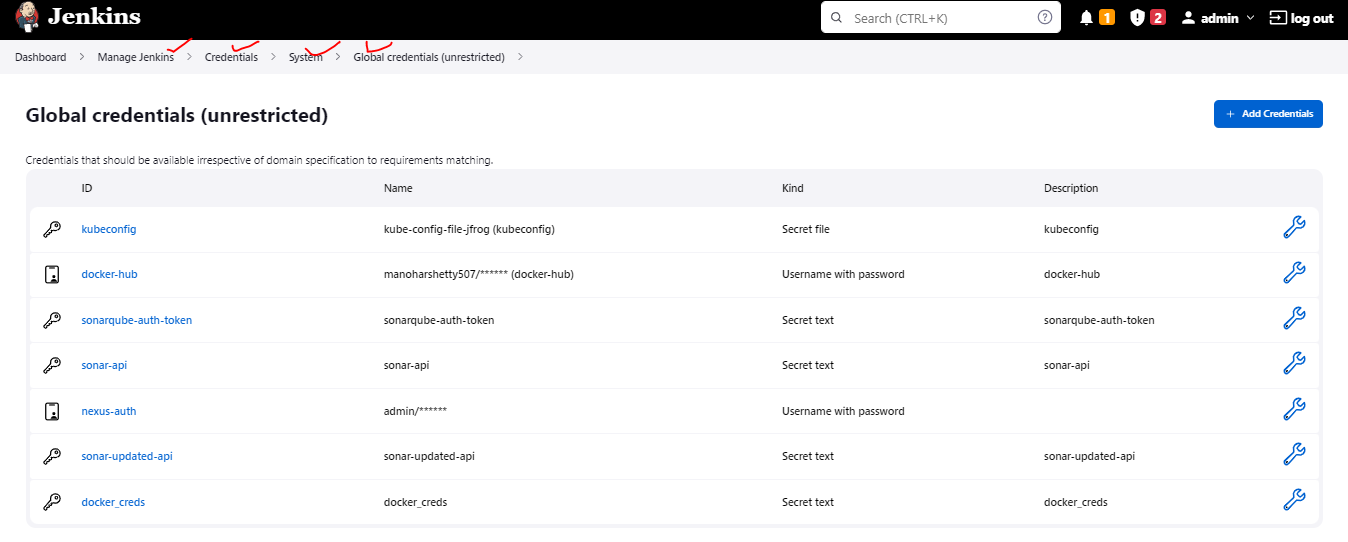


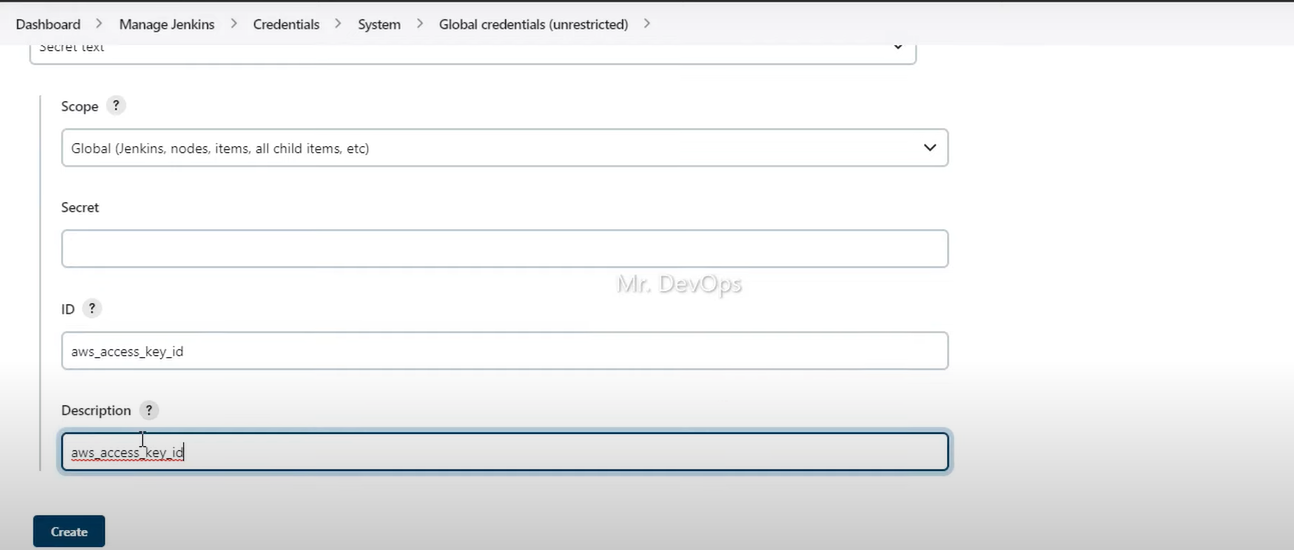
Generated Report

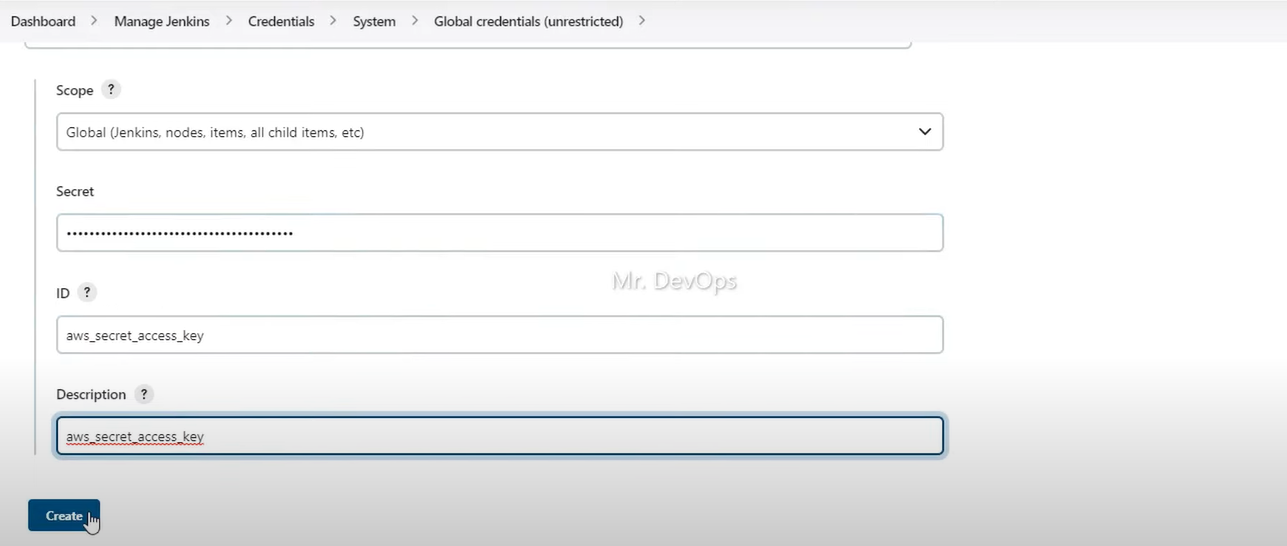
|  |
| --- |
| Started by user [admin](http://10.0.8.45:8080/user/admin)  Obtained jenkins-cd from git <https://github.com/Mark-1305/counter-app.git>  [Pipeline] Start of Pipeline  [Pipeline] node  Running on [Jenkins](http://10.0.8.45:8080/computer/(built-in)/) in /var/lib/jenkins/workspace/updated-demo-counter-CD  [Pipeline] {  [Pipeline] stage  [Pipeline] { (Declarative: Checkout SCM)  [Pipeline] checkout  Selected Git installation does not exist. Using Default  The recommended git tool is: NONE  No credentials specified  Cloning the remote Git repository  Cloning repository <https://github.com/Mark-1305/counter-app.git>  > git init /var/lib/jenkins/workspace/updated-demo-counter-CD # timeout=10  Fetching upstream changes from <https://github.com/Mark-1305/counter-app.git>  > git --version # timeout=10  > git --version # 'git version 2.17.1'  > git fetch --tags --progress -- <https://github.com/Mark-1305/counter-app.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10  > git config remote.origin.url <https://github.com/Mark-1305/counter-app.git> # timeout=10  > git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10  Avoid second fetch  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10  Checking out Revision 6923c6ddffd12ae19510659a771e1917c59b8c00 (refs/remotes/origin/main)  > git config core.sparsecheckout # timeout=10  > git checkout -f 6923c6ddffd12ae19510659a771e1917c59b8c00 # timeout=10  Commit message: "Colon-Added after Git Branch"  First time build. Skipping changelog.  [Pipeline] }  [Pipeline] // stage  [Pipeline] withEnv  [Pipeline] {  [Pipeline] stage  [Pipeline] { (git checkout)  [Pipeline] git  Selected Git installation does not exist. Using Default  The recommended git tool is: NONE  No credentials specified  > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/updated-demo-counter-CD/.git # timeout=10  Fetching changes from the remote Git repository  > git config remote.origin.url <https://github.com/Mark-1305/counter-app.git> # timeout=10  Fetching upstream changes from <https://github.com/Mark-1305/counter-app.git>  > git --version # timeout=10  > git --version # 'git version 2.17.1'  > git fetch --tags --progress -- <https://github.com/Mark-1305/counter-app.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10  Checking out Revision 6923c6ddffd12ae19510659a771e1917c59b8c00 (refs/remotes/origin/main)  > git config core.sparsecheckout # timeout=10  > git checkout -f 6923c6ddffd12ae19510659a771e1917c59b8c00 # timeout=10  > git branch -a -v --no-abbrev # timeout=10  > git checkout -b main 6923c6ddffd12ae19510659a771e1917c59b8c00 # timeout=10  Commit message: "Colon-Added after Git Branch"  [Pipeline] }  [Pipeline] // stage  [Pipeline] }  [Pipeline] // withEnv  [Pipeline] }  [Pipeline] // node  [Pipeline] End of Pipeline  Finished: SUCCESS |

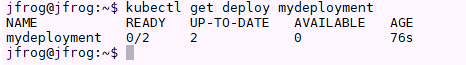
How to Connect Jenkins with EKS Cluster

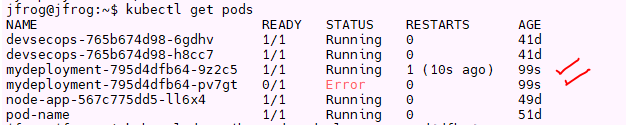
Pass Access Key and Secret Key











Common Issues in Pipeline:

java.lang.IllegalStateException: withKubeConfig step must be called with a body

1. Common issue – kubeconfig passing

Error: Have not used the sh command in bracket after passing credentials

|  |
| --- |
| stage('Check kubectl version') {           steps {                  withKubeConfig([credentialsId: 'kubeconfig'])                   sh 'kubectl version'            }        } |

Solution:

|  |
| --- |
| stage('Check kubectl version') {           steps {                  withKubeConfig([credentialsId: 'kubeconfig']){                   sh 'kubectl version'                  }            }        } |

2. Common Issue

+ kubectl apply -f Deployment.yaml

**Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fswagger-2.0.0.pb-v1%3Ftimeout%3D32s'/><script>window.location.replace('/login?from=%2Fswagger-2.0.0.pb-v1%3Ftimeout%3D32s');</script></head><body style='background-color:white; color:white;'>**

**Error: Authentication Failure**

**Solution: We have to pass the kubeconfig file in every stage if we have to run any commands on Kubernetes cluster**

|  |
| --- |
| stage('Kubernetes Login'){              steps{                  withKubeConfig([credentialsId: 'kubeconfig']){                      sh "echo 'Login Success - Now Deploy'"                  }                  }          }          stage('Check kubectl version') {           steps {                   sh 'kubectl version'            }        } |

**Though it logins and States the result on one step. It will fail authentication if we don’t pass cluster credentials**

**Solution**

|  |
| --- |
| stage('Kubernetes Login'){              steps{                  withKubeConfig([credentialsId: 'kubeconfig']){                      sh "echo 'Login Success - Now Deploy'"                  }                  }          }          stage('Check kubectl version') {           steps {                  withKubeConfig([credentialsId: 'kubeconfig']){                   sh 'kubectl version'                  }            }        } |

**:**