**CI/CD IaC Pipeline Using Github, Docker Jenkins and Terraform**

**Create the Jenkins server using below docker compose**

|  |
| --- |
| version: '3.7'  services:    jenkins:      image: jenkins/jenkins:lts      privileged: true      user: root      ports:        - 8081:8080        - 50000:50000      container\_name: jenkin-sample      volumes:        - /home/${myname}/jenkins\_compose/jenkins\_configuration:/var/jenkins\_home        - /var/run/docker.sock:/var/run/docker.sock |

Now access Jenkins server –

<http://localhost:8081/>

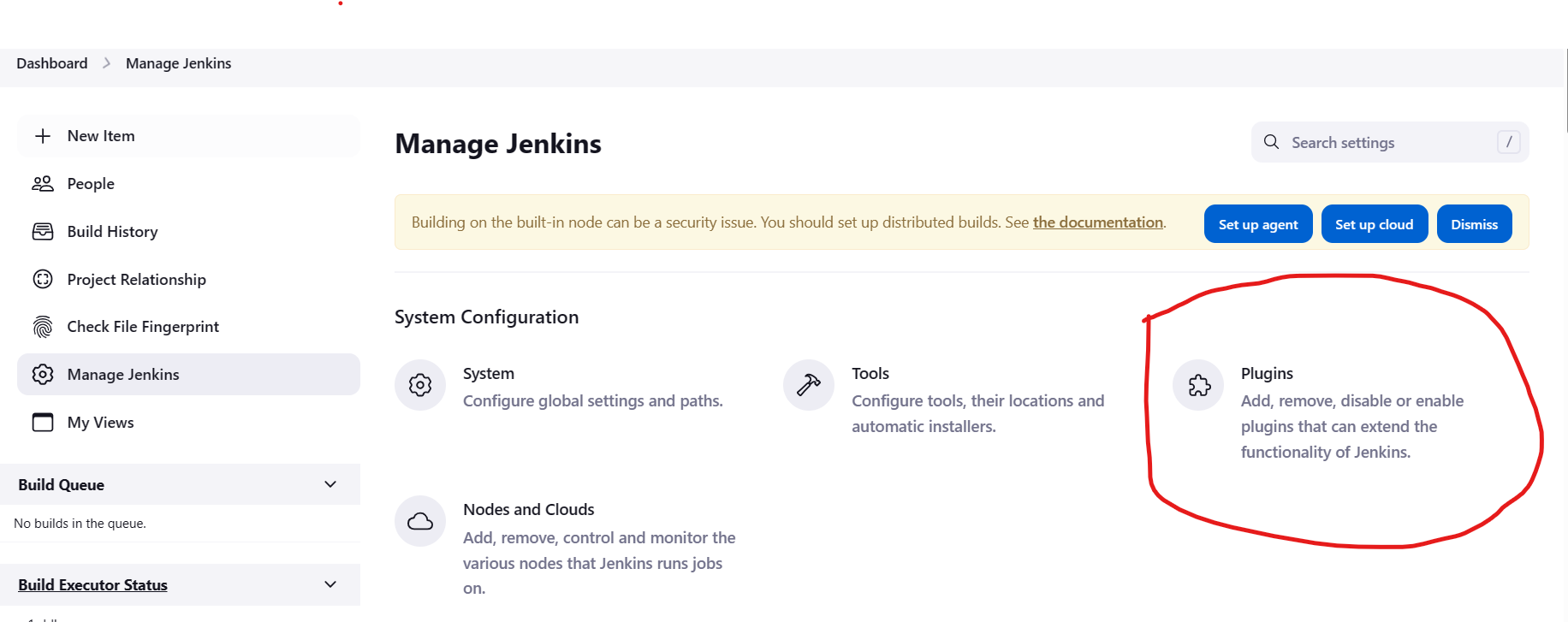
setup Jenkins password and login to Jenkins

Jenkins Prerequisites:

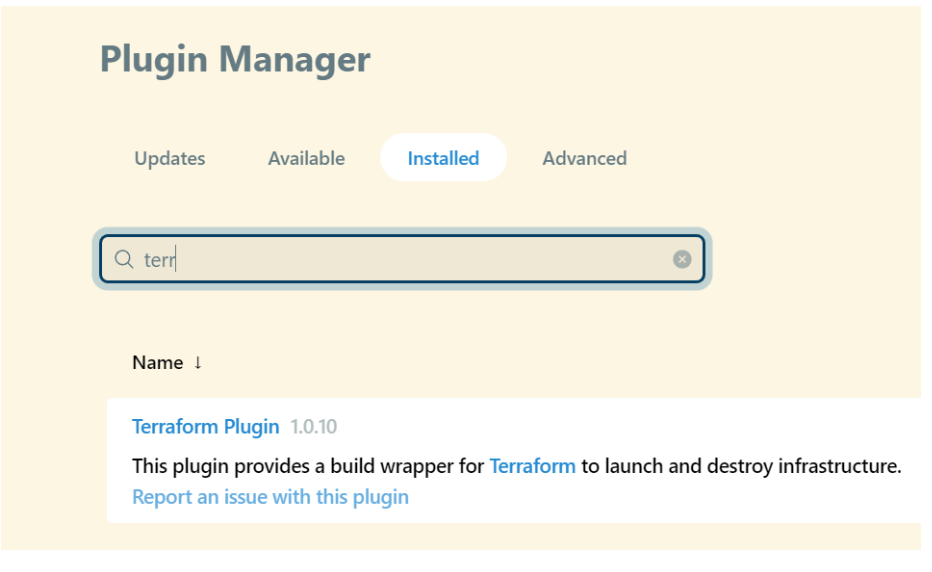
|  |
| --- |
| wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gp  echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list  sudo apt update && sudo apt install terraformg |

### **Jenkins Plugins Required to Build**

* **Click on Manage Jenkins**



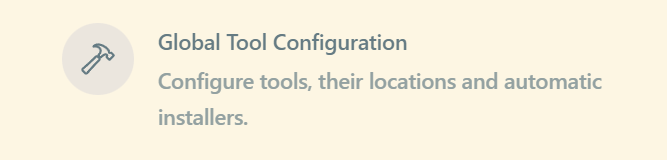
### **Installing the Terraform Plugin**

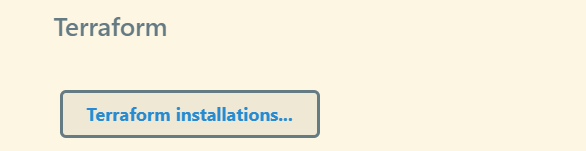


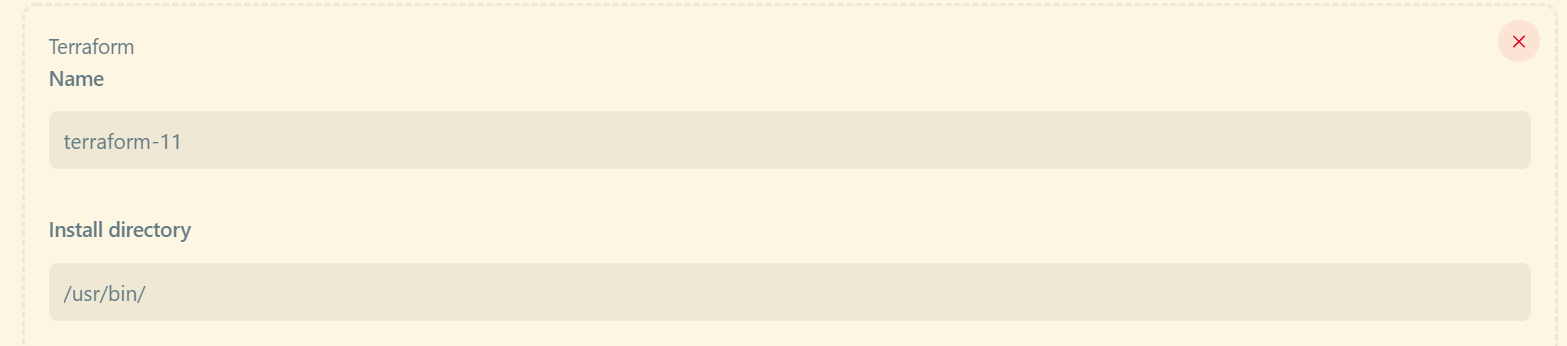
After setup terraform the plugin we need to do some configuring.

### **Configuring Terraform in Jenkins Configuration**

Manage Jenkins -> Global Tool Configuration -> Terraform







1. Click on Terraform Installations
2. Enter name (e.g. Terraform-11)
3. Uncheck automatic install and enter install directory (based on your OS e.g. /usr/bin/ )
4. **Note:** You don't need to included the application itself e.g. terraform.
5. Great now that is done we need to setup the AWS Credentials in Jenkins (This requires the AWS Credentials plugin to be installed.)

### **AWS Credentials Setup**

1. **Note:**Install the AWS Credential Plugin if you don't have it (Use similar steps to Terraform Plugin)
2. Manage Jenkins -> Manage Credentials
3. Click on manage Jenkins, then click manage credentials, then click on the Jenkins store, the click the Global credentials link, then add credentials.



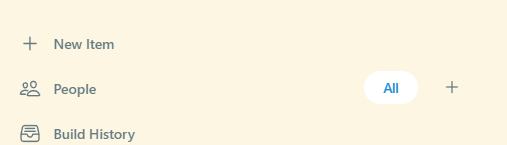
Note: Your can also use IAM role instead of credentials.

Now we need to setup Jenkins pipeline

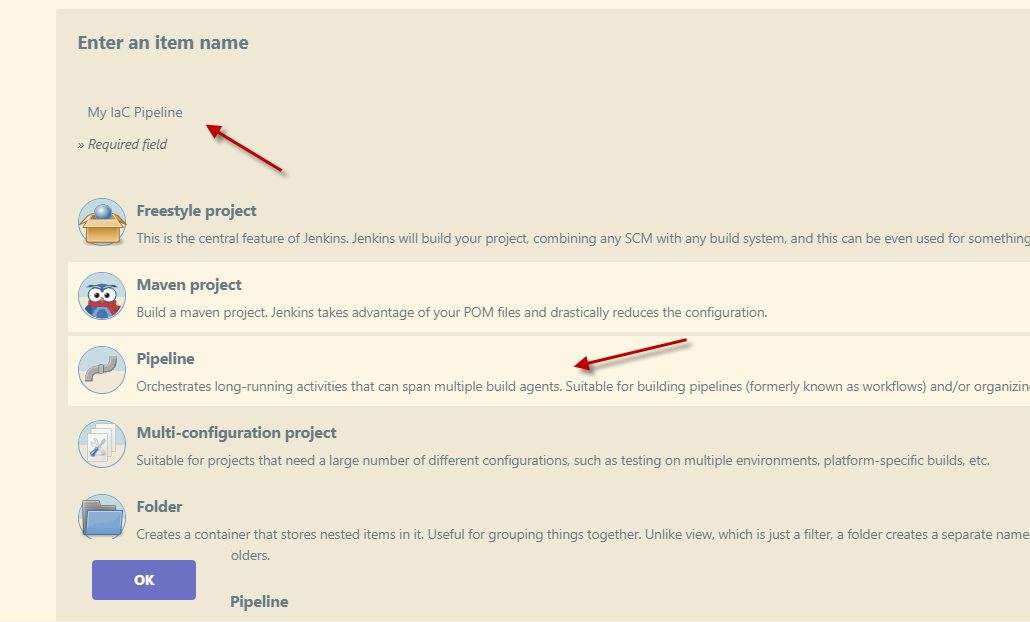
Creating a Jenkins Pipeline

### **Create the Pipeline**

**1. Click on create item**



**2. Enter Pipeline Name (e.g. My Terraform IaC Pipeline)**

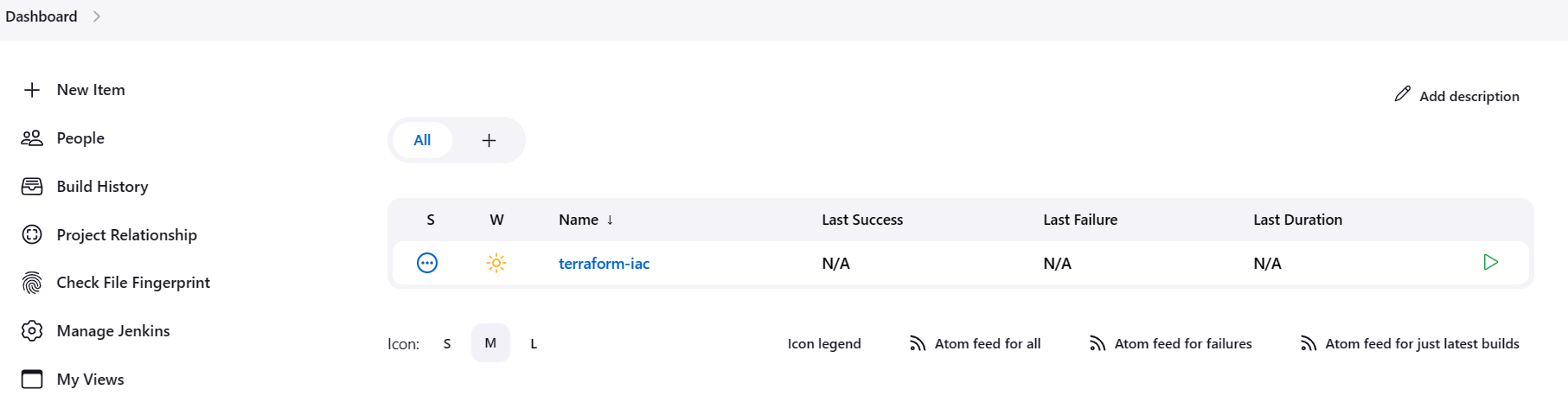


3. Enter your pipeline script:

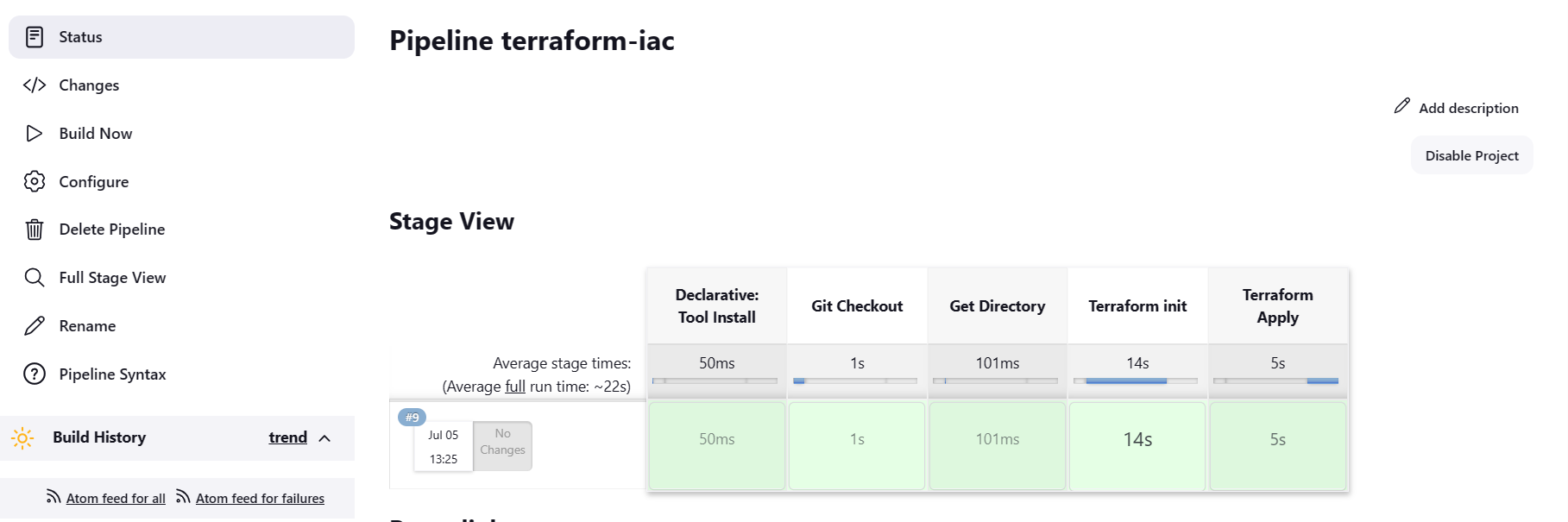
**Note:**You need to change the credentialsId: to the name you gave for the id of the AWS Credentials earlier and replace " ec0125a7-4849-4ba2-8af6-97e236380eed " with your own.

|  |
| --- |
| pipeline{  agent {  label 'default'  }  tools {  terraform 'terraform-11'  }    stages{  stage('Git Checkout'){  steps{  git '<github-project>  }  }  stage('Get Directory') {  steps{  println(WORKSPACE)  }  }  stage('Terraform init'){  steps{  sh 'terraform init'  }  }  stage('Terraform Apply'){  steps{  withCredentials([[  $class: 'AmazonWebServicesCredentialsBinding',  credentialsId: "ec0125a7-4849-4ba2-8af6-97e236380eed",  accessKeyVariable: 'AWS\_ACCESS\_KEY\_ID',  secretKeyVariable: 'AWS\_SECRET\_ACCESS\_KEY']]) {  sh 'terraform apply --auto-approve'  }  }  }  }  } |

After creating pipeline you can see now , below pipeline in dashboard.



Now click on build and check the status -



Now check the console output-  


After the successfully pipeline job run, you can check if all the aws resources created successfully –