

Deployment #4

Welcome to Deployment 4!! Time to deploy with Terraform. You will need to follow the steps below and then add to the pipeline.

1. Install Jenkins on an EC2 if you haven't already:

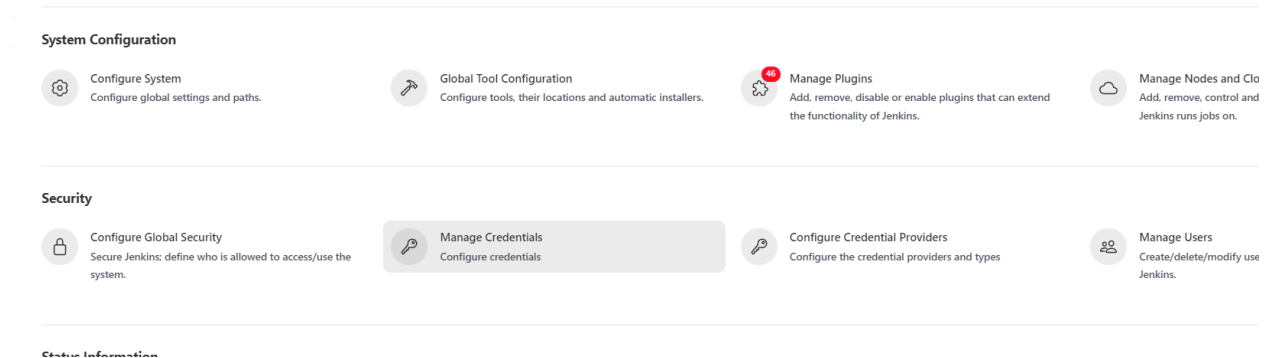
- You do not have to recreate a Jenkins server in your VPC. Highly recommended you use your Jenkins server from the default VPC!!

2. Install Terraform on the Jenkins Server:

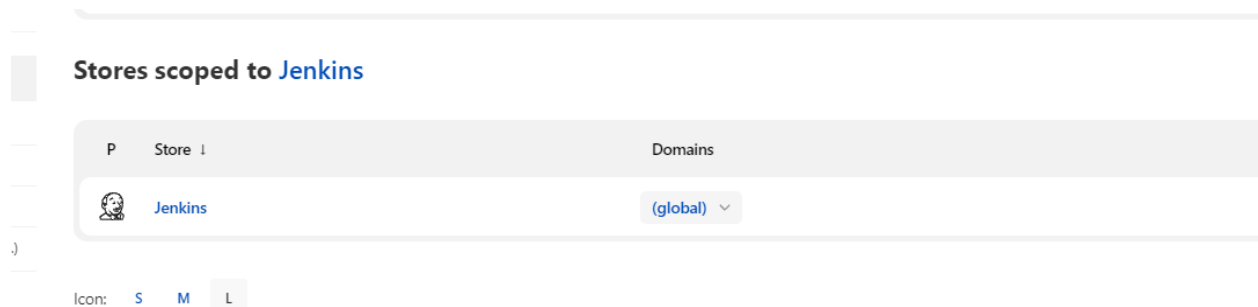
- Link to install Terraform:
<https://www.terraform.io/downloads>

3. Configure credentials on Jenkins:

- In the Jenkins Dashboard, click on manage Jenkins and then select Manage Credentials:



- Now select Global.



- On the right hand side, select Add Credentials:

Imported Bookmarks bar CYS Personal AWS Documentation ClassNotes DMV

Jenkins







Dashboard > Credentials > System > Global credentials (unrestricted) >

↑ Back to credential domains

+ Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain spec

ID
 38017438-4189-4602-9846-752ac40c0fbd
 aws-key
 aws2
 aws3
 AWS-EBuser
 SSH-key

Icons: S M I

- Now enter the **First** credentials:
 - Select “Secret text” for Kind
 - Scope should be Global
 - **Secret:** Copy and Paste your aws access key
 - **ID:** AWS_ACCESS_KEY
 - Select Create

New credentials

Kind
Secret text

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Secret

ID ?

Description ?

Create

- Now enter the **Second** credentials:
 - Select “Secret text” for Kind
 - Scope should be Global
 - **Secret:** Copy and Paste your aws secret key
 - **ID:** AWS_SECRET_KEY
 - Select Create

4. Create a Pipeline build in Jenkins:

- Before you build your pipeline and execute, observe the Jenkinsfile, initTerraform folder and all files in the initTerraform folder.
- Once you have successfully run your deployment and check your application. Add a destroy stage to the Jenkinsfile:

```
stage('Destroy') {
  steps {
    withCredentials([string(credentialsId: 'AWS_ACCESS_KEY', variable: 'aws_access_key'),
                    string(credentialsId: 'AWS_SECRET_KEY', variable: 'aws_secret_key')]) {
      dir('intTerraform') {
        sh 'terraform destroy -auto-approve -var="aws_access_key=$aws_access_key"
        -var="aws_secret_key=$aws_secret_key"'
      }
    }
  }
}
```

1. Now create a VPC with Terraform and deploy your application to it!!

2. Add your additions from Deployment 3 to the Pipeline!!

3. Diagram the new pipeline!!!

- Must have a diagram of pipeline and VPC

- Must included any additions to the pipeline to the diagram

4.Create documentation!!!

Note: Please submit your work by uploading your work to a repo or the forked repo. Then submit the link to the repo via LMS.