**Jenkins**

[**https://github.com/sundarp1438/azure-terraform-kubernetes.git**](https://github.com/sundarp1438/azure-terraform-kubernetes.git)

A diagram of two people

Description automatically generated

**Jenkins Installation:**

1. Please login to the Linux machine.
2. Please download the Jenkins.war file in any location.

Cd /opt

Wget <https://get.jenkins.io/war-stable/2.440.2/jenkins.war>

A screenshot of a computer

Description automatically generated

A black and white screen with white text

Description automatically generated

1. Run the below command to run the Jenkins.

java -jar jenkins.war

A black screen with white text

Description automatically generated

1. Browse to” http://localhost:8080” and wait until the Unlock Jenkins page appears.

A screenshot of a computer

Description automatically generated

1. Please get the initial password from the below path for first time.

A screenshot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

1. Please proceed with suggested plug-ins installation.

A screenshot of a computer

Description automatically generated

1. Plugins are going to install.

A screenshot of a computer

Description automatically generated

1. Initial screen, click on the below option to use as admin.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Now, we are landing on welcome page of Jenkins.

A screenshot of a computer

Description automatically generated

1. Change the password for admin.

Go to Admin 🡪 click on Configure.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white and blue rectangle with black lines

Description automatically generated

1. It is going to log off and re-login with new password.

A screenshot of a login page

Description automatically generated

**azure-cli installation:**

Import the Microsoft repository key:

sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc

Add the Azure CLI repository:

sudo tee /etc/yum.repos.d/azure-cli.repo <<EOF

[azure-cli]

name=Azure CLI

baseurl=https://packages.microsoft.com/yumrepos/azure-cli

enabled=1

gpgcheck=1

gpgkey=https://packages.microsoft.com/keys/microsoft.asc

EOF

Install the Azure CLI:

sudo dnf install azure-cli -y

Verify the installation:

az --version

**Terraform installation:**

curl -o terraform.zip https://releases.hashicorp.com/terraform/1.0.11/terraform\_1.0.11\_linux\_amd64.zip

unzip terraform.zip

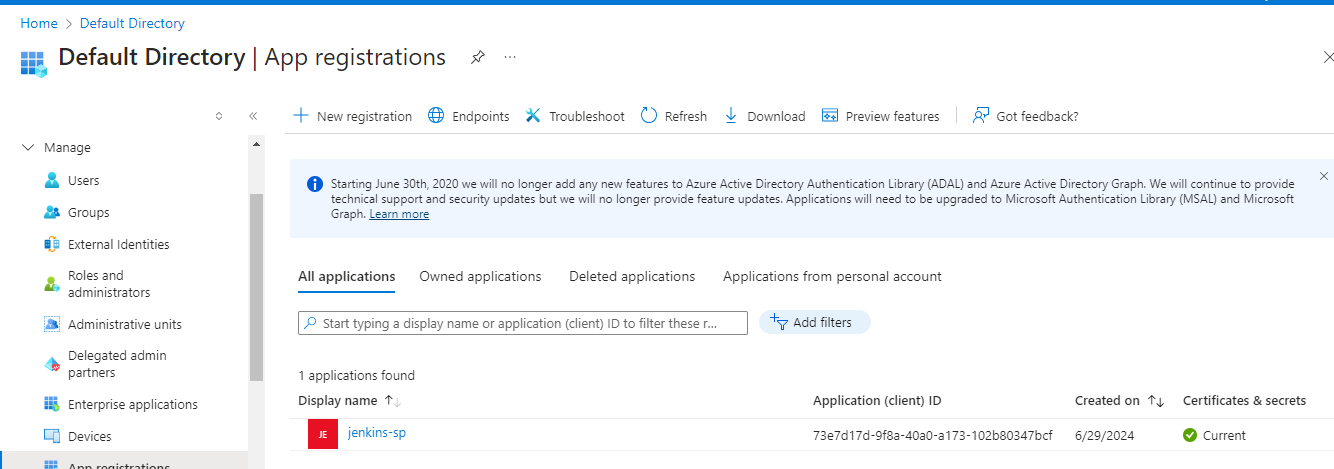
sudo mv terraform /usr/local/bin/

**Azure App registration Creation:**

1. Login to the Azure Portal.

<https://portal.azure.com/>

1. Look for the Microsoft Entra ID service.
2. Go to App registrations 🡪 click new registration.



1. Save the client\_id and password and tenant\_id, subscription\_Id for Jenkins configuration.

**Plugins Installation:**

1. Go to manage Jenkins 🡪 plugins 🡪 install the below plugins.

Azure CLI Plugin

Azure Credentials

Terraform Plugin

1. Go to manage Jenkins 🡪 credentials manager 🡪 add system azure credentials under system.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Go to manage Jenkins 🡪 Tools🡪 Add the terraform tool.

A screenshot of a computer

Description automatically generated

**Jenkinsfile:**

pipeline {

agent any

tools{

"org.jenkinsci.plugins.terraform.TerraformInstallation" "TERRAFORM\_HOME"

}

environment {

PATH = "$PATH:/usr/local/bin"

}

stages {

stage('Workspace Cleaning'){

steps{

cleanWs()

}

}

stage('Git Checkout'){

steps{

git branch: 'main', url: 'https://github.com/sundarp1438/azure-terraform-kubernetes.git'

}

}

stage('Terraform init'){

steps{

withCredentials([azureServicePrincipal(

credentialsId: 'azure-credentials',

subscriptionIdVariable: 'ARM\_SUBSCRIPTION\_ID',

clientIdVariable: 'ARM\_CLIENT\_ID',

clientSecretVariable: 'ARM\_CLIENT\_SECRET',

tenantIdVariable: 'ARM\_TENANT\_ID'

)]) {

sh " terraform init"

}

}

}

stage('Terraform Format'){

steps{

withCredentials([azureServicePrincipal(

credentialsId: 'azure-credentials',

subscriptionIdVariable: 'ARM\_SUBSCRIPTION\_ID',

clientIdVariable: 'ARM\_CLIENT\_ID',

clientSecretVariable: 'ARM\_CLIENT\_SECRET',

tenantIdVariable: 'ARM\_TENANT\_ID'

)]) {

sh " terraform fmt"

}

}

}

stage('Terraform Validate'){

steps{

withCredentials([azureServicePrincipal(

credentialsId: 'azure-credentials',

subscriptionIdVariable: 'ARM\_SUBSCRIPTION\_ID',

clientIdVariable: 'ARM\_CLIENT\_ID',

clientSecretVariable: 'ARM\_CLIENT\_SECRET',

tenantIdVariable: 'ARM\_TENANT\_ID'

)]) {

sh " terraform validate"

}

}

}

stage('Terraform plan'){

steps{

withCredentials([azureServicePrincipal(

credentialsId: 'azure-credentials',

subscriptionIdVariable: 'ARM\_SUBSCRIPTION\_ID',

clientIdVariable: 'ARM\_CLIENT\_ID',

clientSecretVariable: 'ARM\_CLIENT\_SECRET',

tenantIdVariable: 'ARM\_TENANT\_ID'

)]) {

sh " terraform plan -out=tfplan -var client\_id=$ARM\_CLIENT\_ID -var client\_secret=$ARM\_CLIENT\_SECRET -var subscription\_id=$ARM\_SUBSCRIPTION\_ID -var tenant\_id=$ARM\_TENANT\_ID"

}

}

}

stage('Approval') {

steps {

script {

def userInput = input(id: 'Confirm', message: 'Apply Terraform?', parameters: [ [$class: 'BooleanParameterDefinition', defaultValue: false, description: 'Apply terraform', name: 'Confirm'] ])

}

}

}

stage('Terraform apply'){

steps{

withCredentials([azureServicePrincipal(

credentialsId: 'azure-credentials',

subscriptionIdVariable: 'ARM\_SUBSCRIPTION\_ID',

clientIdVariable: 'ARM\_CLIENT\_ID',

clientSecretVariable: 'ARM\_CLIENT\_SECRET',

tenantIdVariable: 'ARM\_TENANT\_ID'

)]) {

sh "terraform apply --auto-approve -var client\_id=$ARM\_CLIENT\_ID -var client\_secret=$ARM\_CLIENT\_SECRET -var subscription\_id=$ARM\_SUBSCRIPTION\_ID -var tenant\_id=$ARM\_TENANT\_ID"

}

}

}

}

}

**Terraform Files:**

**Main.tf**

# Create a resource group

resource "azurerm\_resource\_group" "aks" {

name = "aksResourceGroup"

location = "East US"

}

# Create a virtual network

resource "azurerm\_virtual\_network" "aks" {

name = "aksVnet"

address\_space = ["10.1.0.0/16"]

location = azurerm\_resource\_group.aks.location

resource\_group\_name = azurerm\_resource\_group.aks.name

}

# Create a subnet

resource "azurerm\_subnet" "aks" {

name = "aksSubnet"

resource\_group\_name = azurerm\_resource\_group.aks.name

virtual\_network\_name = azurerm\_virtual\_network.aks.name

address\_prefixes = ["10.1.1.0/24"]

}

# Create an AKS cluster

resource "azurerm\_kubernetes\_cluster" "aks" {

name = "aksCluster"

location = azurerm\_resource\_group.aks.location

resource\_group\_name = azurerm\_resource\_group.aks.name

dns\_prefix = "aksdns"

default\_node\_pool {

name = "default"

node\_count = 2

vm\_size = "Standard\_DS2\_v2"

vnet\_subnet\_id = azurerm\_subnet.aks.id

}

identity {

type = "SystemAssigned"

}

tags = {

Environment = "Development"

}

}

**Variable.tf**

# client ID

variable "client\_id" {

type = string

description = "Client ID"

}

# client secret

variable "client\_secret" {

type = string

description = "Client secret"

}

# Subscription ID

variable "subscription\_id" {

type = string

description = "Subscription ID"

}

# Tenant ID

variable "tenant\_id" {

type = string

description = "Tenant ID"

}

**Provider.tf**

# Azure Provider

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = "~>3.0"

}

}

}

# configure the Microsoft Azure Provider

provider "azurerm" {

client\_id = var.client\_id

client\_secret = var.client\_secret

subscription\_id = var.subscription\_id

tenant\_id = var.tenant\_id

features {}

}

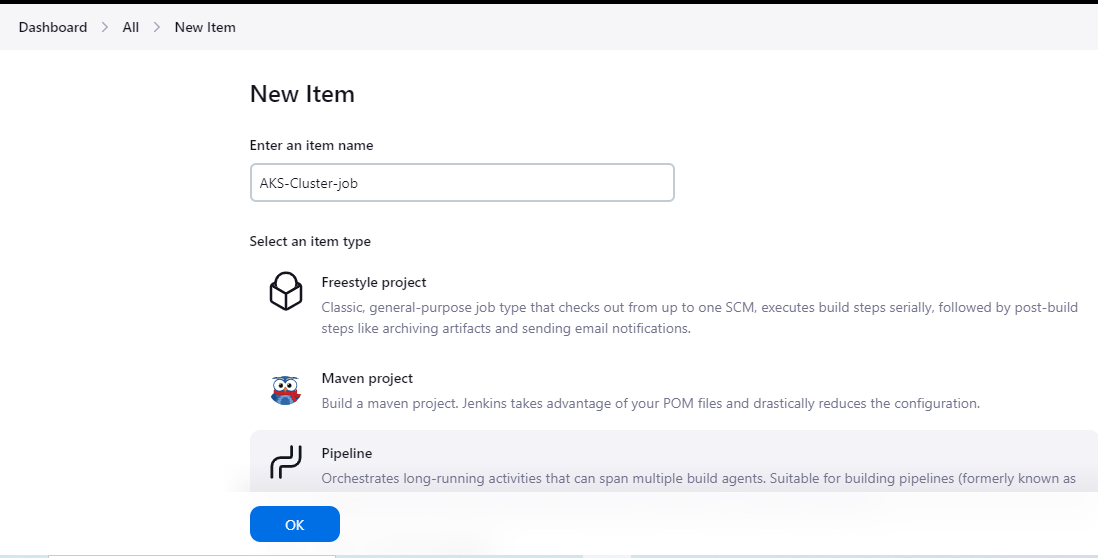
**Jenkins Pipeline Job Creation:**

1. Go to Jenkins dashboard 🡪 click on New item.

A close-up of a computer screen

Description automatically generated

1. Provide name of the job and select pipeline job.



1. Select pipeline from SCM and provide the GitHub URL.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A long rectangular object with a long rectangular object in the middle

Description automatically generated

1. Click on build now.

A screenshot of a computer

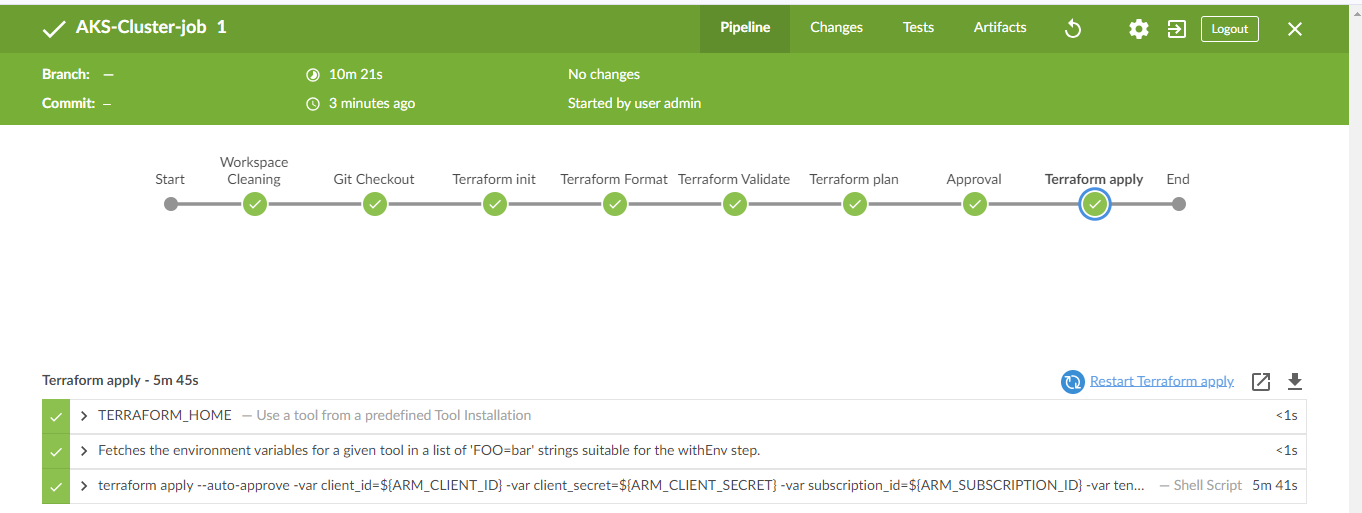
Description automatically generated

1. Build will trigger now.

A screenshot of a computer

Description automatically generated

1. My build is a success now.



1. Go to azure portal and validate the AKS ,Vnet and Resource groups.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Connect to AKS cluster.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated