**Name: Osama Bujuwayid**

**Email:** [**obujuwayid@gmail.com**](mailto:obujuwayid@gmail.com)

**Step 1: DevOps Certification training**

**Project 2: Automating Infrastructure using Terraform.**

DESCRIPTION

Use Terraform to provision infrastructure

**Description:**

Nowadays, infrastructure automation is critical. We tend to put the most emphasis on software development processes, but infrastructure deployment strategy is just as important. Infrastructure automation not only aids disaster recovery, but it also facilitates testing and development.

Your organization is adopting the DevOps methodology and in order to automate provisioning of infrastructure there's a need to setup a centralised server for Jenkins.

Terraform is a tool that allows you to provision various infrastructure components. Ansible is a platform for managing configurations and deploying applications. It means you'll use Terraform to build a virtual machine, for example, and then use Ansible to instal the necessary applications on that machine.

Considering the Organizational requirement you are asked to automate the infrastructure using Terraform first and install other required automation tools in it.

**Tools required:** Terraform, AWS account with security credentials, Keypair

**Expected Deliverables:**

Launch an EC2 instance using Terraform

Connect to the instance

Install Jenkins, Java and Python in the instance

1. **Install Terraform:** Write these command as following.

* sudo mkdir -p /opt/terraform
* sudo wget <https://releases.hashicorp.com/terraform/0.15.3/terraform_0.15.3_linux_amd64.zip>
* sudo apt-get install unzip -y
* sudo unzip terraform\_0.15.3\_linux\_amd64.zip
* sudo mv /opt/terraform/terraform /usr/bin/
* sudo terraform -version

Graphical user interface, text

Description automatically generated Graphical user interface, text

Description automatically generated

1. **Make Terraform Project**
   * sudo mkdir project-terraform
   * cd project-terraform
   * sudo touch aws.tf ceatec2.tf

Graphical user interface, text

Description automatically generated

* + sudo nano aws.tf
    1. **Put inside the file these information**

provider "aws" {

access\_key = "xx"

secret\_key = "xx"

region = "us-east-2"

}

* + 1. "XX" take it from our aws account for Access key and secret key from (My Security Credentials > Access keys > Create New Access Key > ).

Graphical user interface, text, application, email, website

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

* + sudo nano ceatec2.tf
    1. Put inside the file this text:

resource "aws\_instance" "myFirstInstance" {

ami = "ami-916f59f4"

key\_name = "myKey"

instance\_type = "t2.micro"

security\_groups= [ "security\_jenkins\_port"]

tags= {

Name = "jenkins\_instance"

}

}

#Create security group with firewall rules

resource "aws\_security\_group" "security\_jenkins\_port" {

name = "security\_jenkins\_port"

description = "security group for jenkins"

ingress {

from\_port = 8080

to\_port = 8080

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

# outbound from jenkis server

egress {

from\_port = 0

to\_port = 65535

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

tags= {

Name = "security\_jenkins\_port"

}

}

# Create Elastic IP address

resource "aws\_eip" "myFirstInstance" {

vpc = true

instance = aws\_instance.myFirstInstance.id

tags= {

Name = "jenkins\_elstic\_ip"

}

}

* + 1. Go to (EC2 > Instances > Lanch Instance > Ununtu Server > Take amNumber)

Graphical user interface, text, application, email

Description automatically generatedA screenshot of a computer

Description automatically generated

* + 1. Go to (Network and security > key pairs > Crreate new key pair)

1. **Start Terrafom Project**: enter these command
   * sudo terrafom plan && sudo terraform apply
   * Putty and Openssh