Terraform

Documentation by Erika Rai and Love Bansal

Terraform version 1

Step 1: Create Key pair

Step 2: Create security group

Step 3: Create EC2 Instance

Step 4: Open command prompt run following commamds

- Cd .ssh
- Ssh –I key_pair_file_name.pem ubuntu@public_IP_address
- Sudo su
- Visit www.terraform/download and copy following commands and run
- wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
- 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 terraform
- make directory using mkdir directory_name

  Example 1
- cd Example1/

- ls –ltra
- Now create main.tf using vi editor
- Vi main.tf
- Vi editor is in reading mode by default to change it to writing press ‘I’
- Now write the following content into the file
- Resource “local_file” “myfile”{
  content = “This is my text”
  filename = “%mytextfile.txt”
  }
- Press esc to go to read mode
- Press shift + ‘:’ and now write mode to quit “wq!”
- Now to look into main. Tf run “cat main.tf”
- Aias tf = terraform, aliasing to call terraform with short variable name
- Tf init, to initialize the terraform and get required files
- Tf plan
- Tf apply
- Now it run whatever written in main.tf file
- Come out of the folder and create new folder using command

  Example 2
- “cd .. && mkdir Ex-2 && cd Ex-2”
- Cat > main.tf , > sign is used to write a new file
- Write
  Terraform {
  Required_version = “>=0.12.26”
  }
  Output “Welcome_world” {

```
Value = “Welcome, world!”
}
```

- Press enter
- To stop, save and exit from editor we press ctrl + Z
- Tf init
- Tf plan
- Tf apply
- Terraform will run main.tf and give output

Example 3

- Cd .. && Mkdir Ex-3
- Cd Ex-3
- cat<<EOF > main.tf

```
terraform {
 required_version = ">= 0.12.26"
 }
provider "aws" {
access_key = " "
secret_key = " "
region = "us-east-1"
}
resource "aws_s3_bucket" "b" {
 bucket = "my-tf-your-18augbucket22"
acl = "private"
 versioning {
enabled = true
}
}
```

- EOF
- Versioning maintains history of our data
- Tf init

- Tf plan
- Tf apply, you can check the bucket is created
- Now to destroy the bucket we will run following command
- Tf destroy

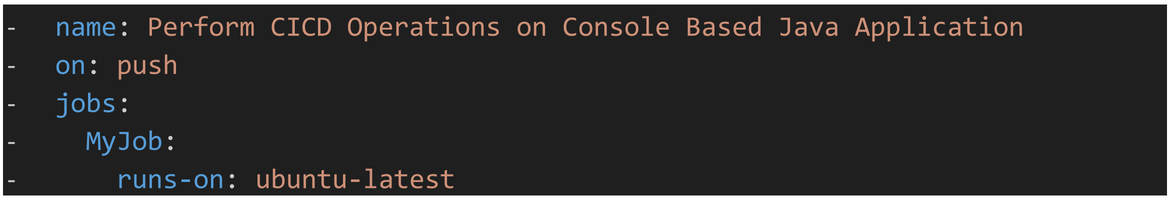
Terraform version 2

-Create main.tf file and add the following content

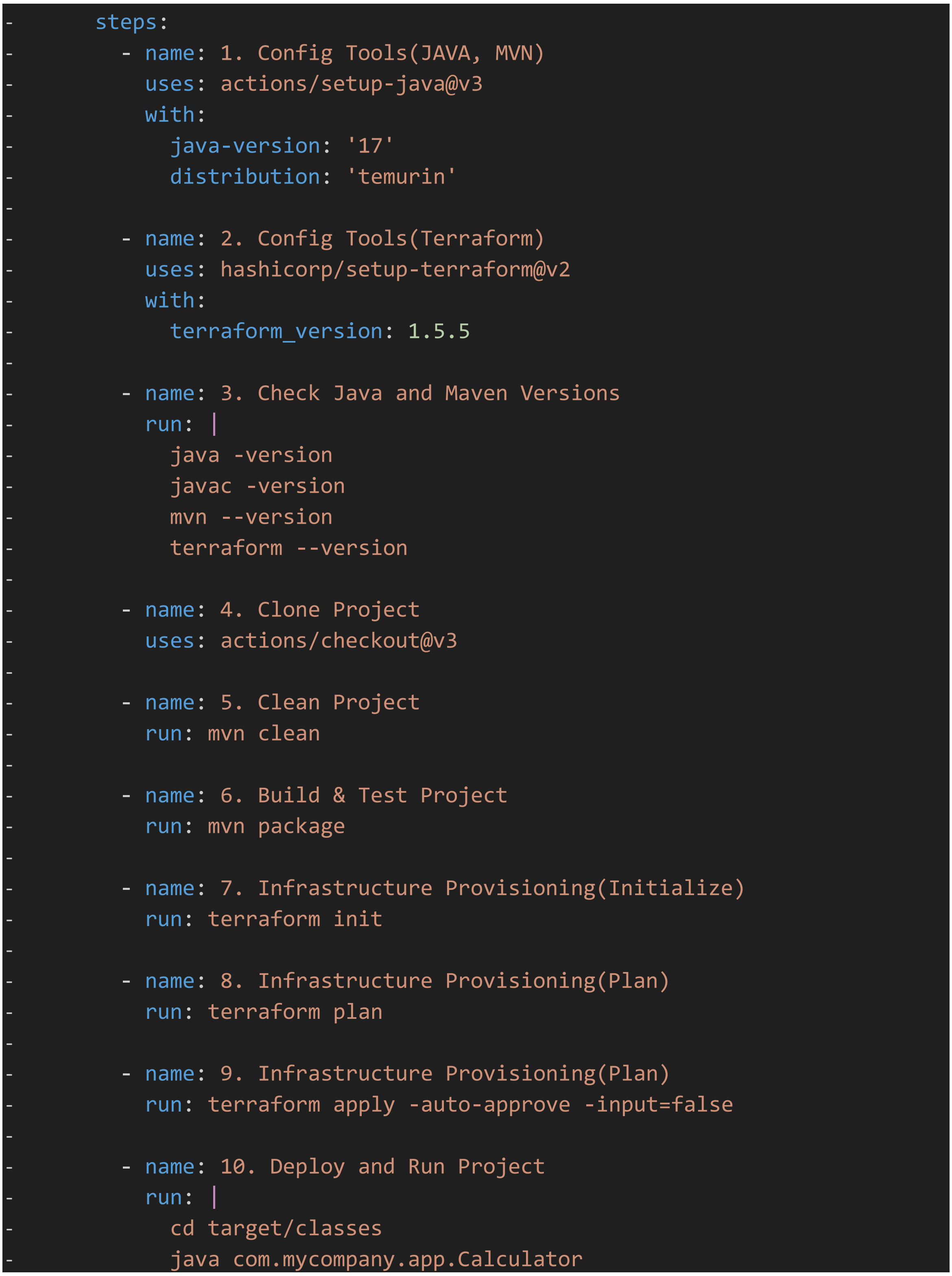
```
terraform {
         required_version = ">= 0.12.26"
         }
provider "aws" {
        access_key = " "
        secret_key = " "
        region = "us-east-1"
        }
resource "aws_s3_bucket" "b" {
         bucket = "my-tf-your-18augbucket22"
        acl = "private"
         versioning {
        enabled = true
        }
}
```

Now make a .github folder, make new workflows folder in it and make new file .YAML file with .yml or .yaml

- Add the following content into YAML file



```
-   name: Perform CICD Operations on Console Based Java Application
-   on: push
-   jobs:
-     MyJob:
-       runs-on: ubuntu-latest
```



```
      steps:
        - name: 1. Config Tools(JAVA, MVN)
          uses: actions/setup-java@v3
          with:
            java-version: '17'
            distribution: 'temurin'

        - name: 2. Config Tools(Terraform)
          uses: hashicorp/setup-terraform@v2
          with:
            terraform_version: 1.5.5

        - name: 3. Check Java and Maven Versions
          run: |
            java -version
            javac -version
            mvn --version
            terraform --version

        - name: 4. Clone Project
          uses: actions/checkout@v3

        - name: 5. Clean Project
          run: mvn clean

        - name: 6. Build & Test Project
          run: mvn package

        - name: 7. Infrastructure Provisioning(Initialize)
          run: terraform init

        - name: 8. Infrastructure Provisioning(Plan)
          run: terraform plan

        - name: 9. Infrastructure Provisioning(Plan)
          run: terraform apply -auto-approve -input=false

        - name: 10. Deploy and Run Project
          run: |
            cd target/classes
            java com.mycompany.app.Calculator
```

name in above file is optional

on is for the command on which you want the file to be executed

In **jobs** you can write different task you want to execute and write following commands

-Now, open terminal in vs code and do “git add .” to add everything

-the run “git commit –m “any _message””

-now to push changes to github run “git push –u origin main”

When you press enter the yaml file gets executed and the job will be performed which will create an s3 bucket and run the java maven file