**Prerequisite:**

* AWS cloud access with aws admin key & token (**ONLY** **Programmatic access**) ( I used aws free tier & IAM role with AdministratorAccess )
* Bash installed
* SSH-keygen installed
* Terraform installed

**How to run application:**

1. Please extract zip file
2. Navigate to folder
3. Setup your AWS aws\_access\_key & aws\_secret\_key (.either in default field in variable.tf (not recommended ) or with following system variables commands on terminal

**export AWS\_ACCESS\_KEY\_ID="youraccesskey"**

**export AWS\_SECRET\_ACCESS\_KEY="yoursecretkey"**

1. After setting credential please run following command on shell script to run whole setup:

“**bash setup.sh**”

**Important note:** I have user default t2.micro ubuntu AMI so might be it will prompt you to accept T&C, IT will also show URL

**What is include:**

Shell script tasks

1. Download library & Build project
2. create new key pair for Aws machines SSH
3. Trigger Terraform and setup new AWS infrastructure consists following AWS products

* VPN
* Subnets
* route tables
* security group
* application load balancer
* 2 ec2 instances

Insides: After creating infrastructure terraform copy all files from project build folder to end point ec2 machines and then run one ansible task to run all java application on each machines, you can see application by browsing DNS of **Load balancer URL** which you get at the end of terraform scripts. I created 2 application machines to perform A/B testing. You can ssh machine with **Ubuntu** user and IP shown by terraform output. (SSH only allowed from machine terraform initiated, restricted in security group)

**Next steps for maturity & Improvements.**

1. Setup Continues infrastructure/CI/CD pipeline for deployment (tools in aws - code commit, code pipeline & code build)
2. Setup automated tests for continues infrastructure also integrated in pipeline.
3. For better resource utilization we can use containerization.
4. I used local terraform state file it should be on remote location so setup back-ground also requires
5. Use baston server for deployment and ssh.
6. Setup monitoring service & scaling mechanism for production infrastructure.

**My development tools:**

I used Visual studio code and ubuntu vagrant machine to create this Infrastructure as code demo.

Thanks & Regards:

Hari