R. Da Costa

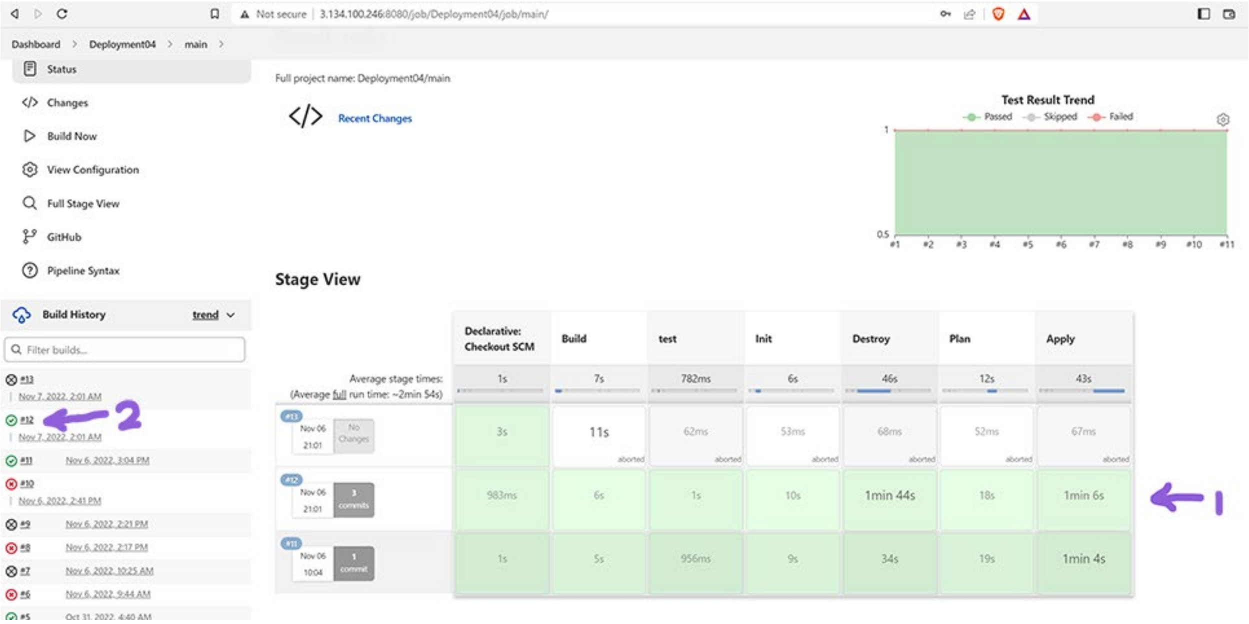
Kura Labs – Deployment 04 **2022-11-05**

Project Outline:

Using a Jenkins CI/CD Pipeline in conjunction with Terraform, to build a VPC and deploy a python application to it.

Note: The details of the setup of the Jenkins server that will run the pipeline is not defined in this document

1. Jenkins runs Build, Test stages on the application. Then the Terraform stages [Init, Destroy, Plan and Apply] are run to build the VPC and deploy the application
    1) See successful completion of all stages
    2) Navigate to Pipline results



Not secure | 3.134.100.246:8080/job/Deployment04/job/main/

Dashboard > Deployment04 > main >

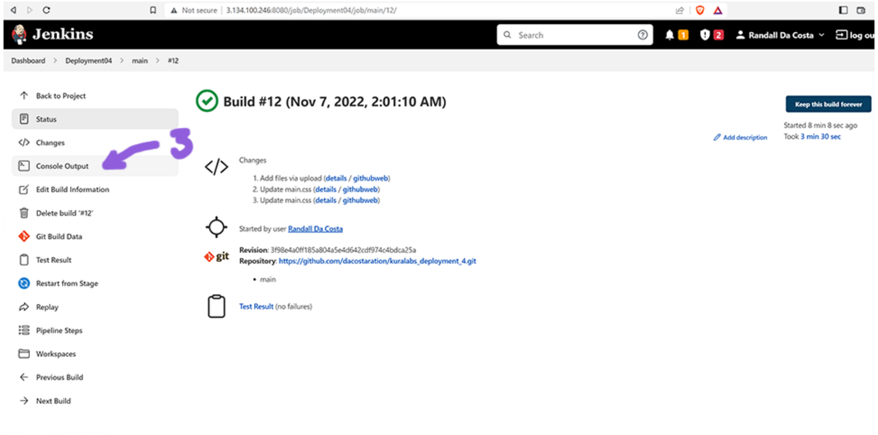
Full project name: Deployment04/main

#13 Nov 7, 2022, 2:01 AM
#12 Nov 7, 2022, 2:01 AM
#11 Nov 6, 2022, 3:04 PM
#10 Nov 6, 2022, 2:41 PM
#9 Nov 6, 2022, 2:21 PM
#8 Nov 6, 2022, 2:17 PM
#7 Nov 6, 2022, 10:25 AM
#6 Nov 6, 2022, 9:44 AM
#5 Oct 31, 2022, 4:40 AM

Stage View

	Declarative: Checkout SCM	Build	test	Init	Destroy	Plan	Apply
Average stage times: (Average full run time: ~2min 54s)	1s	7s	782ms	6s	46s	12s	43s
#13 Nov 06 21:01 No Changes	3s	11s aborted	62ms aborted	53ms aborted	68ms aborted	52ms aborted	67ms aborted
#12 Nov 06 21:01 3 commits	983ms	6s	1s	10s	1min 44s	18s	1min 6s
#11 Nov 06 10:04 1 commit	1s	5s	956ms	9s	34s	19s	1min 4s

2. Review “Console Output”
   3) Click to view “Console Output” to view the newly created VPC’s elastic IP.



Build #12 (Nov 7, 2022, 2:01:10 AM)

Started 8 min 8 sec ago
Took 3 min 30 sec

Changes

1. Add files via upload (details / githubweb)
2. Update main.css (details / githubweb)
3. Update main.css (details / githubweb)

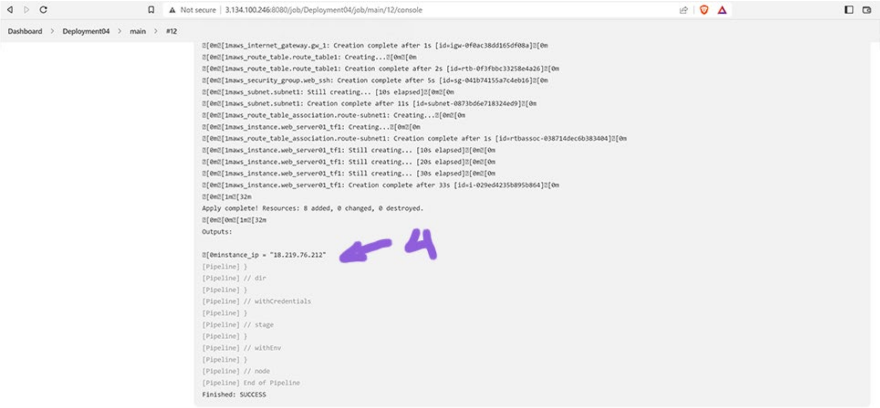
Started by user Randall Da Costa

Revision: 3f98e4a0ff185a804a5e4d642cdf974c4bdca25a
Repository: https://github.com/dacostaration/kuralabs_deployment_4.git

- main

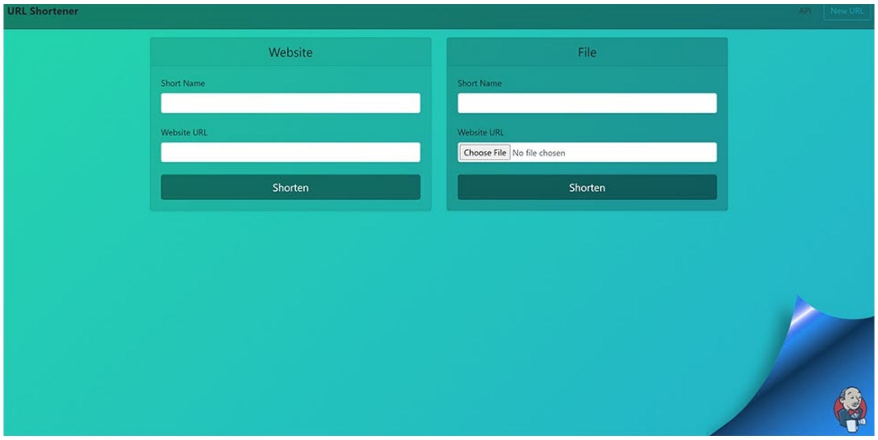
Test Result (no failures)

3. Copy the VPC’s IP address. This is where the application will be accessible [port 8000]:
   4) http://xx.xx.xx.xx:8000

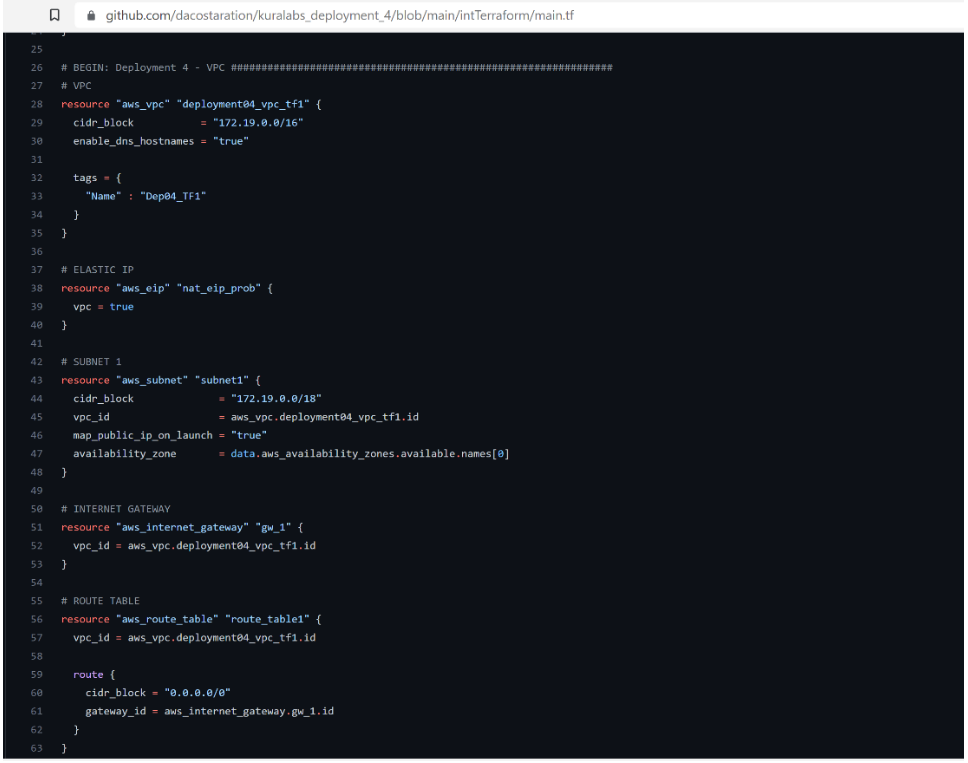

Not secure | 3.134.100.246:8080/job/Deployment04/job/main/12/console
Dashboard > Deployment04 > main > #12
4

```
▯[0m▯[1maws_internet_gateway.gw_1: Creation complete after 1s [id=igw-0f0ac38dd165df08a]▯[0m
▯[0m▯[1maws_route_table.route_table1: Creating...▯[0m▯[0m
▯[0m▯[1maws_route_table.route_table1: Creation complete after 2s [id=rtb-0f3fbbc33258e4a26]▯[0m
▯[0m▯[1maws_security_group.web_ssh: Creation complete after 5s [id=sg-041b74155a7c4eb16]▯[0m
▯[0m▯[1maws_subnet.subnet1: Still creating... [10s elapsed]▯[0m▯[0m
▯[0m▯[1maws_subnet.subnet1: Creation complete after 11s [id=subnet-0873bd6e718324ed9]▯[0m
▯[0m▯[1maws_route_table_association.route-subnet1: Creating...▯[0m▯[0m
▯[0m▯[1maws_instance.web_server01_tf1: Creating...▯[0m▯[0m
▯[0m▯[1maws_route_table_association.route-subnet1: Creation complete after 1s [id=rtbassoc-038714dec6b383404]▯[0m
▯[0m▯[1maws_instance.web_server01_tf1: Still creating... [10s elapsed]▯[0m▯[0m
▯[0m▯[1maws_instance.web_server01_tf1: Still creating... [20s elapsed]▯[0m▯[0m
▯[0m▯[1maws_instance.web_server01_tf1: Still creating... [30s elapsed]▯[0m▯[0m
▯[0m▯[1maws_instance.web_server01_tf1: Creation complete after 33s [id=i-029ed4235b895b864]▯[0m
▯[0m▯[1m▯[32m
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
▯[0m▯[0m▯[1m▯[32m
Outputs:

▯[0minstance_ip = "18.219.76.212"
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4. URL-shortener application display



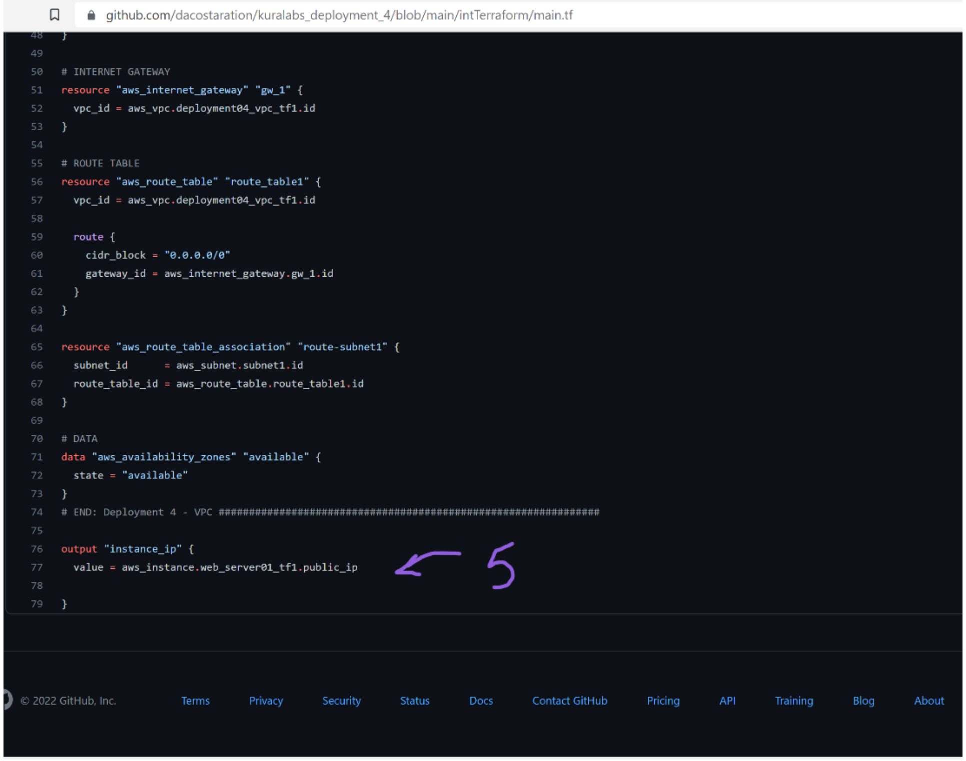
5. Terraform VPC definition



github.com/dacostaration/kuralabs_deployment_4/blob/main/intTerraform/main.tf

```
# BEGIN: Deployment 4 - VPC ################################################################
# VPC
resource "aws_vpc" "deployment04_vpc_tf1" {
  cidr_block           = "172.19.0.0/16"
  enable_dns_hostnames = "true"

  tags = {
    "Name" : "Dep04_TF1"
  }
}

# ELASTIC IP
resource "aws_eip" "nat_eip_prob" {
  vpc = true
}

# SUBNET 1
resource "aws_subnet" "subnet1" {
  cidr_block              = "172.19.0.0/18"
  vpc_id                  = aws_vpc.deployment04_vpc_tf1.id
  map_public_ip_on_launch = "true"
  availability_zone       = data.aws_availability_zones.available.names[0]
}

# INTERNET GATEWAY
resource "aws_internet_gateway" "gw_1" {
  vpc_id = aws_vpc.deployment04_vpc_tf1.id
}

# ROUTE TABLE
resource "aws_route_table" "route_table1" {
  vpc_id = aws_vpc.deployment04_vpc_tf1.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw_1.id
  }
}
```

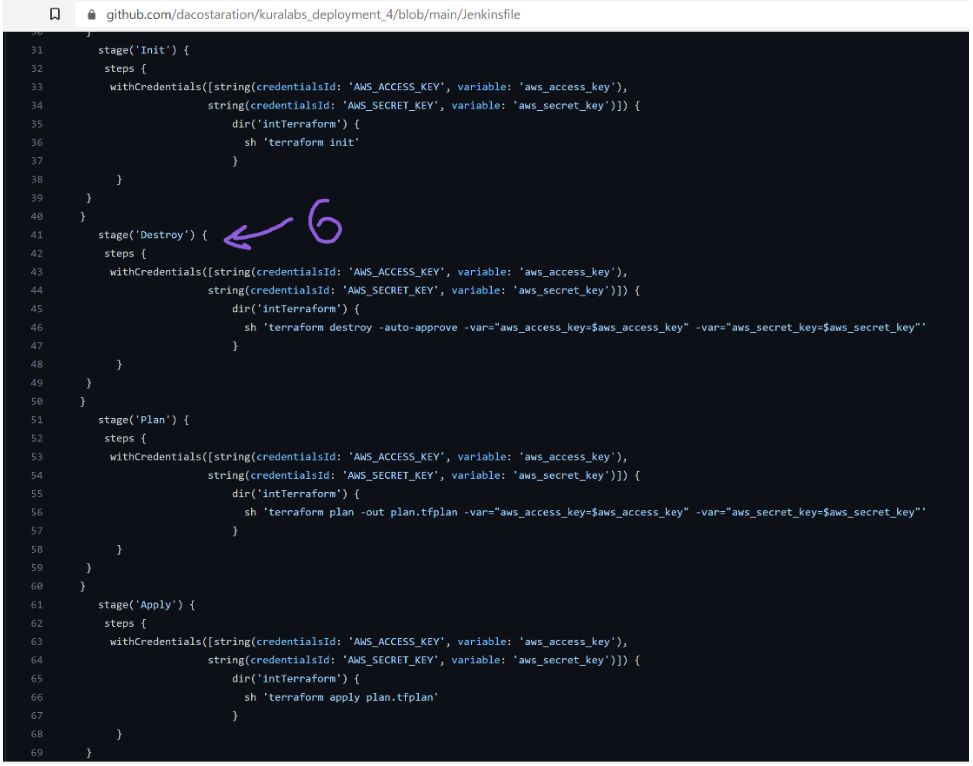
6. Terraform VPC definition (continued)
   5) "output" segment that will print the VPC IP to the console log (see #3)



github.com/dacostaration/kuralabs_deployment_4/blob/main/intTerraform/main.tf

```
}

# INTERNET GATEWAY
resource "aws_internet_gateway" "gw_1" {
  vpc_id = aws_vpc.deployment04_vpc_tf1.id
}

# ROUTE TABLE
resource "aws_route_table" "route_table1" {
  vpc_id = aws_vpc.deployment04_vpc_tf1.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw_1.id
  }
}

resource "aws_route_table_association" "route-subnet1" {
  subnet_id      = aws_subnet.subnet1.id
  route_table_id = aws_route_table.route_table1.id
}

# DATA
data "aws_availability_zones" "available" {
  state = "available"
}
# END: Deployment 4 - VPC ################################################################

output "instance_ip" {
  value = aws_instance.web_server01_tf1.public_ip

}
```

7. “Destroy” stage in Jenkinsfile
    6) Needed to terminate any instances and relationships to subnets before attempting to spin up new instances during subsequent pipeline runs



github.com/dacostaration/kuralabs_deployment_4/blob/main/Jenkinsfile

```
   stage('Init') {
    steps {
     withCredentials([string(credentialsId: 'AWS_ACCESS_KEY', variable: 'aws_access_key'),
                     string(credentialsId: 'AWS_SECRET_KEY', variable: 'aws_secret_key')]) {
                         dir('intTerraform') {
                           sh 'terraform init'
                         }
      }
 }
}
   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"'
                         }
      }
 }
}
   stage('Plan') {
    steps {
     withCredentials([string(credentialsId: 'AWS_ACCESS_KEY', variable: 'aws_access_key'),
                     string(credentialsId: 'AWS_SECRET_KEY', variable: 'aws_secret_key')]) {
                         dir('intTerraform') {
                           sh 'terraform plan -out plan.tfplan -var="aws_access_key=$aws_access_key" -var="aws_secret_key=$aws_secret_key"'
                         }
      }
 }
}
   stage('Apply') {
    steps {
     withCredentials([string(credentialsId: 'AWS_ACCESS_KEY', variable: 'aws_access_key'),
                     string(credentialsId: 'AWS_SECRET_KEY', variable: 'aws_secret_key')]) {
                         dir('intTerraform') {
                           sh 'terraform apply plan.tfplan'
                         }
      }
 }
```

Issues:

1. Without the “Destroy” stage, the terraform apply stage would likely fail after your first successful run due to to the fact that some elements that would need to be created, would already exist. E.g. security groups, subnets, etc.

Possible Improvements:

1. Instead of having the user navigate the Jenkins console to find the newly created VPC’s IP address, the pipeline could notify the user via email