

1. Launch an EKS cluster using terraform
2. Deploy a sample nginx/tomcat/react service on it.
3. Attach a LB and create R53 endpoint pointing to lab, service should be accessible from the endpoint.
4. Variablize all parameters and pass values as env.tfvars file.

Import vpc configuration

Import subnets configuration

Import role for cluster and nodes

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat deployment.yml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ayush@ayush:~/github/ttnbootcamp-tothenew$
```

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat loadbalancer.yml
apiVersion: v1
kind: Service
metadata:
  name: loadbalancer
spec:
  type: LoadBalancer
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
ayush@ayush:~/github/ttnbootcamp-tothenew$
```

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ cat main.tf
provider "aws" {
  region = "us-east-1"
}

resource "aws_vpc" "ayush-vpc" {
  # (resource arguments)
  cidr_block = "192.168.0.0/16"
  enable_dns_hostnames = true
  enable_dns_support   = true
}

resource "aws_subnet" "subnet-1" {
  map_public_ip_on_launch = true
  vpc_id                  = aws_vpc.ayush-vpc.id
  cidr_block = "192.168.64.0/18"
  tags = {
    "kubernetes.io/cluster/cluster" = "shared"
  }
}

resource "aws_subnet" "subnet-2" {
  map_public_ip_on_launch = true
  vpc_id                  = aws_vpc.ayush-vpc.id
  cidr_block = "192.168.128.0/18"
  tags = {
    "kubernetes.io/cluster/cluster" = "shared"
  }
}
```

```

ayush@ayush:~/github/ttnbootcamp-tothenew$ cat variable.tf
variable "default_tags" {
  type = map(string)
  default = {
    Name: "ayush-tf",
    owner: "ayush",
    purpose: "ayush-tf",
  }
}

variable "cluster-name" {
  default = "cluster"
  type    = string
}
ayush@ayush:~/github/ttnbootcamp-tothenew$

```

```

ayush@ayush:~/github/ttnbootcamp-tothenew$ cat run.sh
terraform apply
aws eks --region us-east-1 update-kubeconfig --name cluster
sleep 5
kubectl apply -f deployment.yml
sleep 5
kubectl apply -f loadbalancer.yml
sleep 5
kubectl get svc
ayush@ayush:~/github/ttnbootcamp-tothenew$

```

```

ayush@ayush:~/github/ttnbootcamp-tothenew$ ./run.sh
data.aws_region.current: Refreshing state...
aws_iam_role.role-eks-node: Refreshing state... [id=EKSNodeInstanceRole]
aws_iam_role.role-eks-master: Refreshing state... [id=eksServiceRole]
aws_vpc.ayush-vpc: Refreshing state... [id=vpc-0b061c711cd6ec803]
aws_subnet.subnet-1: Refreshing state... [id=subnet-0a5a6b106347d1b70]
aws_subnet.subnet-2: Refreshing state... [id=subnet-033003c92989d26d9]
aws_security_group.node-sg: Refreshing state... [id=sg-0c17617fb945c9e4a]
aws_security_group.master-sg: Refreshing state... [id=sg-0ffc419d032895ec7]
aws_security_group_rule.node-ingress-self: Refreshing state... [id=sgrule-4218791901]
aws_security_group_rule.node-ingress-cluster: Refreshing state... [id=sgrule-3495080785]
aws_security_group_rule.demo-cluster-ingress-node-http: Refreshing state... [id=sgrule-2171574775]
aws_eks_cluster.cluster: Refreshing state... [id=cluster]
data.aws_ami.eks-worker: Refreshing state...
aws_launch_configuration.cluster-lc: Refreshing state... [id=terraform-eks-launch-config20200415182903224200000001]
aws_autoscaling_group.cluster-asg: Refreshing state... [id=terraform-eks-asg]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
~ update in-place

```

```
- name: aws
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: aws-iam-authenticator
      args:
        - "token"
        - "-i"
        - var.cluster-name
Updated context arn:aws:eks:us-east-1:187632318301:cluster/cluster in /home/ayush/.kube/config
deployment.apps/nginx-deployment created
service/loadbalancer created
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP
kubernetes           ClusterIP           10.100.0.1      <none>
loadbalancer         LoadBalancer       10.100.117.253  aa2ca9ae848694460b3973111fb905d1-1387151283.us-east-1.elb.amazonaws.com
```

← → ↻ ⓘ Not secure | aa2ca9ae848694460b3973111fb905d1-1387151283.us-east-1.elb.amazonaws.com

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.