

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

Execute run.sh

```
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"
  }
}
```

```

resource "aws_elb" "bar"{
  count = var.enable_route53 ? 1 : 0

  listener {
    instance_port      = var.protocol
    instance_protocol = "TCP"
    lb_port            = 80
    lb_protocol        = "TCP"
  }

  tags = {
    "kubernetes.io/cluster/${var.cluster-name}" = "owned"
    "kubernetes.io/service-name"               = "default/loadbalancer"
  }
cross_zone_load_balancing = false
}

resource "aws_route53_zone" "private" {
  count = var.enable_route53 ? 1 : 0
  name = "ttn-internal.com"

  vpc {
    vpc_id = aws_vpc.ayush-vpc.id
  }
}

```

```
resource "aws_route53_zone" "private" {
  count = var.enable_route53 ? 1 : 0
  name = "ttn-internal.com"

  vpc {
    vpc_id = aws_vpc.ayush-vpc.id
  }
}

resource "aws_route53_record" "www" {
  count = var.enable_route53 ? 1 : 0
  zone_id = aws_route53_zone.private[count.index].zone_id
  name     = "loadbalancer.com"
  type     = "A"

  alias {
    name           = aws_elb.bar[count.index].dns_name
    zone_id        = aws_elb.bar[count.index].zone_id
    evaluate_target_health = true
  }
}
```

```
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  
}
```

```
variable "enable_route53" {  
    description = "enable route 53"  
    type        = bool  
    default = "1"  
}
```

```
variable "protocol" {  
    description = "find protcol of LB"  
    type        = number  
    default = "3000"  
}
```

```
ayush@ayush:~/github/ttnbootcamp-tothenew$
```



```

ayush@ayush:~/github/ttnbootcamp-tothenew$ cat run.sh
terraform apply -var="enable_route53=0"
aws eks --region us-east-1 update-kubeconfig --name cluster
sleep 5
kubectl delete -f deployment.yml
kubectl apply -f deployment.yml
sleep 5
kubectl delete -f loadbalancer.yml
kubectl apply -f loadbalancer.yml
sleep 5
kubectl get pods
kubectl get svc
lb_dns=$( kubectl get svc | awk 'NR == 3 {print $4}' )
echo $lb_dns | cut -d "-" -f "1"
lb_name=$( echo $lb_dns | cut -d "-" -f "1" )
terraform import aws_elb.bar $lb_name
protocol=$( kubectl get svc | awk 'NR == 3 {print $5}' | cut -d ":" -f "2" | cut -d "/" -f "1")
echo "protocol " $protocol
terraform apply -var="protocol=$protocol" -var="enable_route53=1"
echo "FINISH "
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	READY	STATUS	RESTARTS	AGE
nginx-deployment-7fd6966748-5th9t	1/1	Running	0	21s
nginx-deployment-7fd6966748-xv2jk	1/1	Running	0	21s
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	
kubernetes	ClusterIP	10.100.0.1	<none>	
loadbalancer	LoadBalancer	10.100.30.80	a350ae1d2827e41f0bcea6942ecd7daf-1072721664.us-east-1.elb.amazonaws.com	
protocol	31305			

```

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-2: Refreshing state... [id=subnet-033003c92989d26d9]
aws_security_group.master-sg: Refreshing state... [id=sg-034a25366400ca1f5]
aws_subnet.subnet-1: Refreshing state... [id=subnet-0a5a6b106347d1b70]
aws_security_group.node-sg: Refreshing state... [id=sg-0974168d5fa9fc866]
aws_security_group_rule.node-ingress-self: Refreshing state... [id=sgrule-26256054]
aws_security_group_rule.node-ingress-cluster: Refreshing state... [id=sgrule-1478019970]
aws_security_group_rule.demo-cluster-ingress-node-http: Refreshing state... [id=sgrule-39331243]
aws_eks_cluster.cluster: Refreshing state... [id=cluster]
aws_eks_node_group.node: Refreshing state... [id=cluster:ayush-ng]
data.aws_ami.eks-worker: Refreshing state...
aws_launch_configuration.cluster-lc: Refreshing state... [id=terraform-eks-launch-config20200415182903224200000001]

```

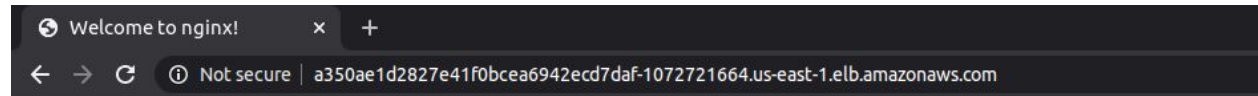
Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
aws_route53_zone.private[0]: Creating...
aws_route53_zone.private[0]: Still creating... [10s elapsed]
aws_route53_zone.private[0]: Still creating... [20s elapsed]
aws_route53_zone.private[0]: Still creating... [30s elapsed]
aws_route53_zone.private[0]: Still creating... [40s elapsed]
aws_route53_zone.private[0]: Still creating... [50s elapsed]
aws_route53_zone.private[0]: Creation complete after 56s [id=Z097084929YA38Y5NQDNI]
aws_route53_record.www[0]: Creating...
aws_route53_record.www[0]: Still creating... [10s elapsed]
aws_route53_record.www[0]: Still creating... [20s elapsed]
aws_route53_record.www[0]: Still creating... [30s elapsed]
aws_route53_record.www[0]: Still creating... [40s elapsed]
aws_route53_record.www[0]: Still creating... [50s elapsed]
aws_route53_record.www[0]: Still creating... [1m0s elapsed]
aws_route53_record.www[0]: Still creating... [1m10s elapsed]
aws_route53_record.www[0]: Creation complete after 1m17s [id=Z097084929YA38Y5NQDNI_loadbalancer.com_A]
```



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.

Back to Hosted Zones

Create Record Set

Import Zone File

Delete Record Set

Record Set Name

X

Any Type

☐ Aliases Only
 ☐ Weighted Only

⏪

<

Displaying 1 to 3 out of 3 Record Sets

>

⏩

<input type="checkbox"/>	Name	Type	Value	Evaluate
<input type="checkbox"/>	ttn-internal.com.	NS	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.	-
<input type="checkbox"/>	ttn-internal.com.	SOA	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amaz	-
<input checked="" type="checkbox"/>	loadbalancer.com.ttn-internal.com.	A	ALIAS a350ae1d2827e41f0bcea6942ecd7daf-1072i	Yes

Edit Record Set

Name:

loadbalancer.com .ttn-internal.com.

Type:

A – IPv4 address

Alias:

☒ Yes
 ☐ No

Alias Target:

a350ae1d2827e41f0bcea6942ecd7daf

Alias Hosted Zone ID:

Z35SXDOTRQ7X7K

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-2.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com
- VPC endpoint: example.us-east-2.vpce.amazonaws.com
- API Gateway custom regional API: d-abcde12345.execute-api.us-west-2.amazonaws.com
- Global Accelerator DNS name: a012345abc.awsglobalaccelerator.com

[Learn More](#)

Routing Policy:

Simple

Route 53 responds to queries based only on the values in this record.

[Learn More](#)

Save Record Set