

# Terraform/EKS Exercise

1. Take any sample hello world application from the internet or build a simple one of your own.
  - a. Write a dockerfile and package it into a docker image. Build the docker image and push it to the docker hub **using terraform**.

```
→ q1 terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of kreuzwerker/docker from the dependency lock file
- Using previously-installed kreuzwerker/docker v3.0.2

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
→ q1 terraform apply -var "username=oartihsin" -var "password=dckr_pat_zKYwiplg0AGFvM-xz8r-i5dVwT0"
docker_image.image: Refreshing state... [id=sha256:6c352c440d71ddd6947b4ed8eb46e9fa0c7c9b1ef295f1911a64b64e727041e0oartihsin/hello-world-flask:v6]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
-/+ destroy and then create replacement

Terraform will perform the following actions:

# docker_image.image must be replaced
-/+ resource "docker_image" "image" {
  ~ id      = "sha256:6c352c440d71ddd6947b4ed8eb46e9fa0c7c9b1ef295f1911a64b64e727041e0oartihsin/hello-world-flask:v6" -> (known after apply)
  ~ image_id = "sha256:6c352c440d71ddd6947b4ed8eb46e9fa0c7c9b1ef295f1911a64b64e727041e0" -> (known after apply)
  ~ name     = "oartihsin/hello-world-flask:v6" -> "registry-1.docker.io/oartihsin/hello-world-flask:v6" # forces replacement
  + repo_digest = (known after apply)

  - build {
    - build_arg      = {} -> null
    - build_args     = {} -> null
    - cache_from     = [] -> null
    - context        = "" -> null
    - cpu_period     = 0 -> null
    - cpu_quota      = 0 -> null
    - cpu_shares     = 0 -> null
    - dockerfile     = "Dockerfile" -> null
    - extra_hosts    = [] -> null
    - force_remove   = false -> null
    - label          = {} -> null
    - labels         = {} -> null
    - memory         = 0 -> null
    - memory_swap    = 0 -> null
    - no_cache       = false -> null
    - pull_parent    = false -> null
    - remove         = true -> null
    - security_opt   = [] -> null
    - shm_size        = 0 -> null
    - squash          = false -> null
    - suppress_output = false -> null
    - tag             = [] -> null
  }
}
```

```

        }
    }

    + build {
        + cache_from   = []
        + context      = "."
        + dockerfile   = "Dockerfile"
        + extra_hosts  = []
        + remove       = true
        + security_opt = []
        + tag          = []
    }

}

# docker_registry_image.helloworld will be created
resource "docker_registry_image" "helloworld" {
    + id           = (known after apply)
    + insecure_skip_verify = false
    + keep_remotely = true
    + name         = "registry-1.docker.io/oartihsin/hello-world-flask:v6"
    + sha256_digest = (known after apply)
}

Plan: 2 to add, 0 to change, 1 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

docker_image.image: Destroying... [id=sha256:6c352c440d71ddd6947b4ed8eb46e9fa0c7c9b1ef295f1911a64b64e727041e0oartihsin/hello-world-flask:v6]
docker_image.image: Destruction complete after 0s
docker_image.image: Creating...
docker_image.image: Creation complete after 5s [id=sha256:b09c9074872e3290a880225fac1b1b5cb963afca3dcc4d8ae0c27caf93349c82registry-1.docker.io/oartihsin/hello-world-flask:v6]
docker_registry_image.helloworld: Creating...
docker_registry_image.helloworld: Still creating... [10s elapsed]

```

The screenshot shows a Docker Hub repository page for the user oartihsin. The repository name is hello-world-flask. The page includes sections for description, tags, and automated builds.

**Description:** A placeholder for a short description, with a note that it's used for indexing on Docker Hub and search engines. An 'Update' button is present.

**Docker commands:** A command box containing `docker push oartihsin/hello-world-flask:tagname`.

**Tags:** Displays two tags: v6 (Pushed 2 minutes ago) and v5 (Pushed 6 minutes ago). A 'See all' link is available.

**Automated Builds:** A section explaining the feature of automatically building and tagging images. It notes compatibility with GitHub and Bitbucket and availability with Pro, Team, and Business subscriptions. An 'Upgrade' button is provided.

- Build a 3-tier VPC with public and private subnets in each availability zone and deploy an EKS cluster in the above VPC.

Note: after running ‘terraform init’ -> error of ‘| An argument named "enable\_classiclink" is not expected here.’ at 3 places in .terraform file. Hence deleted those 3 lines(as these properties are not needed) manually to resolve the error

```

module.eks.kubernetes_config_map.aws_auth[0]: Creating...
module.eks.aws_autoscaling_group.workers[1]: Creating...
module.eks.aws_autoscaling_group.workers[0]: Creating...
module.eks.kubernetes_config_map.aws_auth[0]: Creation complete after 1s [id=kube-system/aws-auth]
module.eks.aws_autoscaling_group.workers[0]: Still creating... [10s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still creating... [10s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still creating... [20s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still creating... [20s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still creating... [30s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still creating... [30s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still creating... [40s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still creating... [40s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Creation complete after 42s [id=rattle-eks-Worker-Group-120240110113557623800000002]
module.eks.aws_autoscaling_group.workers[0] (deposed object a9befeb4): Destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a]
module.eks.aws_autoscaling_group.workers[1]: Creation complete after 42s [id=rattle-eks-Worker-Group-220240110113557618400000001]
module.eks.aws_autoscaling_group.workers[1] (deposed object a4ac8a25): Destroying... [id=rattle-eks-Worker-Group-220240110113003368900000009]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 10s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still destroying... [id=rattle-eks-Worker-Group-220240110113003368900000009, 10s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 20s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still destroying... [id=rattle-eks-Worker-Group-220240110113003368900000009, 20s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 30s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still destroying... [id=rattle-eks-Worker-Group-220240110113003368900000009, 30s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 40s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still destroying... [id=rattle-eks-Worker-Group-220240110113003368900000009, 40s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 50s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Still destroying... [id=rattle-eks-Worker-Group-220240110113003368900000009, 50s elapsed]
module.eks.aws_autoscaling_group.workers[1]: Destruction complete after 59s
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 1m0s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 1m10s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 1m20s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 1m30s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 1m40s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 1m50s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 2m0s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 2m10s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Still destroying... [id=rattle-eks-Worker-Group-12024011011300336890000000a, 2m20s elapsed]
module.eks.aws_autoscaling_group.workers[0]: Destruction complete after 2m22s

Apply complete! Resources: 3 added, 0 changed, 2 destroyed.

Outputs:

cluster_endpoint = "https://C36E68D0F51C5DEBA71EA2113B012ECA.gr7.us-east-1.eks.amazonaws.com"
cluster_id = "rattle-eks"
```

## VPC:

**VPC ID:** vpc-04e55b74da3cc97a4

**State:** Available

**Tenancy:** Default

**Default VPC:** No

**Network Address Usage metrics:** Disabled

**DNS hostnames:** Enabled

**DNS resolution:** Enabled

**Main route table:** rtb-0ecc02be163c14325

**Main network ACL:** acl-0e5e02f0436b4387c

**IPv6 pool:** -

**IPv6 CIDR (Network border group):** -

**Route 53 Resolver DNS Firewall rule groups:** -

**Owner ID:** 043240747508

**Resource map:** Subnets (6), Route tables (3)

## Subnets- in each AZ

Name	Subnet ID	State	VPC	IPv4 CIDR
Rattle-Private-Subnet	subnet-06bc5102a89abce5	Available	vpc-04e55b74da3cc97a4   Ratt...	10.0.3.0/24
Rattle-Private-Subnet	subnet-0ad268300674bf9f8	Available	vpc-04e55b74da3cc97a4   Ratt...	10.0.2.0/24
Rattle-Public-Subnet	subnet-00831fee458efcb49	Available	vpc-04e55b74da3cc97a4   Ratt...	10.0.5.0/24
Rattle-Public-Subnet	subnet-095e47e4691fff710	Available	vpc-04e55b74da3cc97a4   Ratt...	10.0.4.0/24
Rattle-Private-Subnet	subnet-00165f8b5saed06f43	Available	vpc-04e55b74da3cc97a4   Ratt...	10.0.1.0/24
Rattle-Public-Subnet	subnet-01647b7188fba5895	Available	vpc-04e55b74da3cc97a4   Ratt...	10.0.6.0/24

Screenshot of the AWS VPC Subnets page showing a list of 6 subnets.

**Subnets (1/6) Info**

State	VPC	IPv4 CIDR	Availability Zone	Route table	Net
Available	vpc-04e55b74da3cc97a4   Rattl...	10.0.3.0/24	us-east-1c	rtb-067be67ad4a2a83e9   Rattl...	acl...
Available	vpc-04e55b74da3cc97a4   Rattl...	10.0.2.0/24	us-east-1b	rtb-067be67ad4a2a83e9   Rattl...	acl...
Available	vpc-04e55b74da3cc97a4   Rattl...	10.0.5.0/24	us-east-1b	rtb-0c226c4ea73990354   Rattl...	acl...
Available	vpc-04e55b74da3cc97a4   Rattl...	10.0.4.0/24	us-east-1a	rtb-0c226c4ea73990354   Rattl...	acl...
Available	vpc-04e55b74da3cc97a4   Rattl...	10.0.1.0/24	us-east-1a	rtb-067be67ad4a2a83e9   Rattl...	acl...
Available	vpc-04e55b74da3cc97a4   Rattl...	10.0.6.0/24	us-east-1c	rtb-0c226c4ea73990354   Rattl...	acl...

No Hostname type IP name Disabled Resource name DNS AAAA record  
DNS64 Disabled Owner 043240747508

## EKS Cluster:

Screenshot of the AWS EKS Clusters page showing one cluster named "rattle-eks".

**EKS > Clusters**

New Kubernetes versions are available for 1 cluster.

**Clusters (1) Info**

Cluster name	Status	Kubernetes version	Support type	Provider
rattle-eks	Active	1.24 <a href="#">Update now</a>	Standard support until January 31, 2024	EKS

The screenshot shows the AWS EKS Cluster Details page for the cluster 'rattle-eks'. The cluster is in an Active state, running Kubernetes version 1.24. It has standard support until January 31, 2024, and is managed by the EKS provider. There are also notices about the Kubernetes version reaching end-of-life support.

3. Extend the terraform code or provide an automation solution to
  - a. Create a k8s namespace - "exercise",
  - b. Deploy 2 replicas of pods from the above built image in the "exercise" namespace, and
  - c. Expose the deployment using service type LoadBalancer and share the endpoint.

```
Enter a value: yes

kubernetes_namespace.Rattle_NS: Creating...
kubernetes_namespace.Rattle_NS: Creation complete after 1s [id=exercise]
kubernetes_deployment.Rattle_Deployment: Creating...
kubernetes_deployment.Rattle_Deployment: Creation complete after 10s [id=exercise/hello-world-deployment]
kubernetes_service.Rattle_Service: Creating...
kubernetes_service.Rattle_Service: Creation complete after 5s [id=exercise/hello-world-service]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

cluster_endpoint = "https://C36E6BD0F51C5DEBA71EA2113B012ECA.gr7.us-east-1.eks.amazonaws.com"
cluster_id = "rattle-eks"
```

## NameSpace : Exercise

The screenshot shows the AWS EKS console with the URL [us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters/rattle-eks?selectedTab=cl...](https://us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters/rattle-eks?selectedTab=cl...). The left sidebar is titled "Amazon Elastic Kubernetes Service" and includes sections for Clusters (New), Amazon EKS Anywhere (Enterprise Subscriptions New), Related services (Amazon ECR, AWS Batch), Documentation, and Submit feedback. The main content area is titled "Resource types" and shows "Cluster: Namespaces (5)". It includes a "Learn more" link and a search bar. A table lists five namespaces: default (Created 3 hours ago), exercise (Created 10 minutes ago), kube-node-lease (Created 3 hours ago), kube-public (Created 3 hours ago), and kube-system (Created 3 hours ago).

## Replica Sets: 2

The screenshot shows the AWS EKS console with the same URL as the previous screenshot. The left sidebar is identical. The main content area is titled "Resource types" and shows "Workloads: ReplicaSets (2)". It includes a "Learn more" link and a search bar. A table lists two replica sets: coredns-79989457d9 (kube-system, replicaset, 2 pods, 2 ready, 2 desired, created 3 hours ago) and hello-world-deployment-7bd4f5d879 (exercise, replicaset, 2 pods, 2 ready, 2 desired, created 9 minutes ago).

## Service : Loadbalancer

The screenshot shows the AWS EKS Service Details page for a service named 'hello-world-service'. The service was created 16 minutes ago and is associated with the namespace 'exercise'. It uses a LoadBalancer type and has a selector of 'app=MyTestApp'. The IP family policy is SingleStack, and the cluster IP is 172.20.95.216. The service has one port defined. The URL for the load balancer is [adf32ceb29ed2465ab5600848f45c1eb-113460504.us-east-1.elb.amazonaws.com](https://adf32ceb29ed2465ab5600848f45c1eb-113460504.us-east-1.elb.amazonaws.com).

## Script :

The screenshot shows a Slack channel named '# ssl\_expiry\_notificatoin'. A message from the 'incoming-webhook' app at 22:28 yesterday states: "The SSL certificate for {domain} will expire in {days\_until\_expiry} days." A message from the same app at 13:19 today states: "The SSL certificate for {domain} will NOT expire in {days\_until\_expiry} days." Below these messages, several other SSL expiration notifications are listed for various domains like www.google.com, www.titan.email, and www.goratlie.com.