

Kubernetes End to End project on EKS | EKS Install and app deploy with Ingress

AWS ELASTIC KUBERNETES SERVICE

EKS:

Kubeadm & KOPS are used to install the Kubernetes.

-If we use KOPS to install the Kubernetes cluster. We should be the reason for the below issues.

AWS ELASTIC KUBERNETES SERVICE



-master node went down.

-certificate expired.

-API server is down

-ETCD crashed.

-Scheduler is not working

To debug the above issues, it takes a lot of time and makes devops engineer complicated.

Solution for above all AWS provide.

-AWS provide high level control plane by EKS.

-EKS is managed control plan not on data plane, and provide and easy way to integrate with data plane or workers nodes.

-when we attach the workers nodes AWS provides and options.

1.EC2 instance 2.Fargate (AWS serverless compute).

AWS+Fargate = highly stable Kubernetes cluster.

AWS take care of control plane.

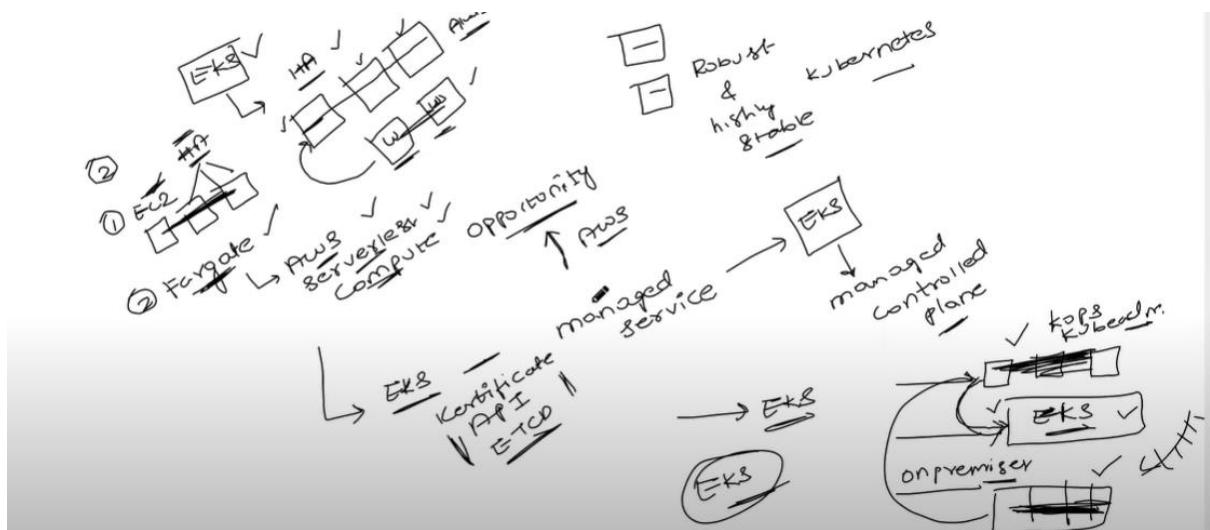
Fargate take care of worker node.

3Ways to use Kubernetes

1.KOPS & Kubeadm on EC2

2.EKS

3.Install Kubernetes on on primises



Agenda:

-Create an EKS cluster/ Deploy application in pod or deployment.

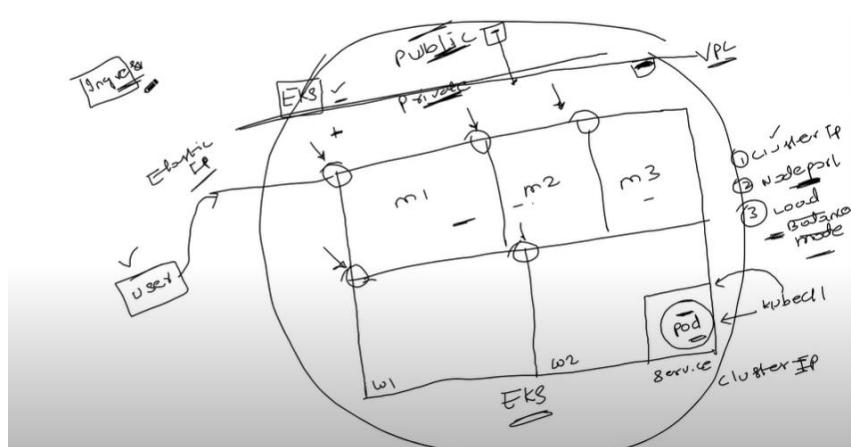
-To access User.

Create a services (Cluster IP/ Node port / Load balancer).

User->VPC->Load balancer ->elastic IP ->Public IP/Private IP ->Application.

But load balancer service type is so costly.

So we use Ingress.



Ingress basically route the traffic inside the cluster.

Devops engineer-> ingress.yaml->Allow user to access the application ->forward the request to services->From the service the request go to POD->write in ingress resource & deploy the ingress resource using Kubectl.

BUT...

User will only access the Public subnet. And app was in private subnet.

User request to load balancer in the public subnet->and come into private subnet.

Concept called **Ingress controller**

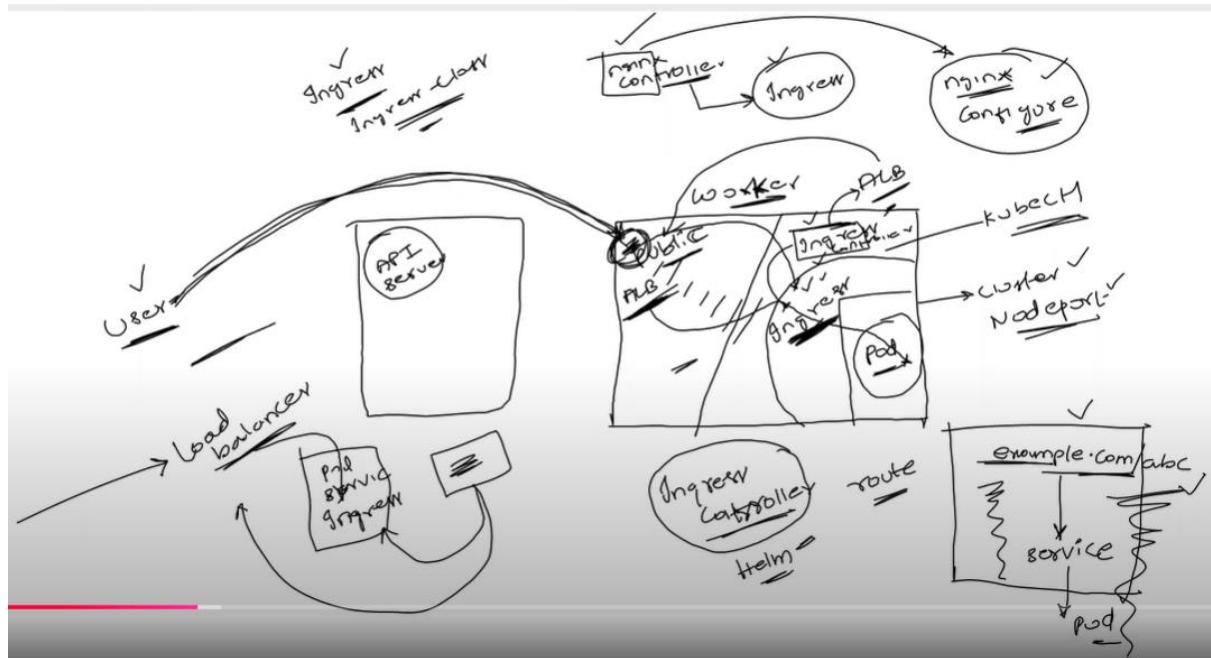
-For every load balancer have there own ingress controller.

All available in helmchart /plane yaml manifest we can deploy by using them.

Ingress controller looks for ->ingress resource-> Create an ALB (application load balance) -> By using them user request to ->ALB->application in pod.

Using ingress class we can define who has to watch the ingress resources->Ingress controller.

Devops engineer ->create ingress resources for every pod ->create on ingress controllers ->to create ALB for user to acces the POD.



Now we set up the environment.

prerequisites

kubectl – A command line tool for working with Kubernetes clusters. For more information, see [Installing or updating kubectl](#).

eksctl – A command line tool for working with EKS clusters that automates many individual tasks. For more information, see [Installing or updating](#).

AWS CLI – A command line tool for working with AWS services, including Amazon EKS. For more information, see [Installing, updating, and uninstalling the AWS CLI in the AWS Command Line Interface User Guide](#). After installing the AWS CLI, we recommend that you also configure it. For more information, see [Quick configuration with aws configure in the AWS Command Line Interface User Guide](#).

1.AWS CLI in our local Done.

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mahes>aws --version
aws-cli/2.17.49 Python/3.11.9 Windows/10 exe/AMD64

C:\Users\mahes>
```

Now connect the CLI to our AWS account.

The screenshot shows two screenshots of the AWS IAM console. The top screenshot displays the 'My security credentials' page for a root user named 'Mahesh23'. It shows the account name 'Mahesh23', email address 'kannegantimaheshbabu1998@gmail.com', and a note about best practices for root users. The bottom screenshot shows the 'Create access key' wizard, Step 1: Alternatives to root user access keys. It includes a warning message about not recommending root user access keys and suggesting IAM roles or IAM Identity Center instead. A checkbox at the bottom of the wizard asks if the user understands creating a root access key is not a best practice.

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access key

Access key	Secret access key
AKIAQIQANWR4ABDHOSI	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

Access keys (1)						Actions ▾	Create access key
Access key ID	Created on	Access key last used	Region last used	Service last used	Status		
AKIAQIQANWR4ABDHOSI	34 days ago	29 days ago	us-east-1	sts	Active		

AKIAQIQANWR4ABDHOSI

/UcXgSaY4phLjKKddy5TJeYPnz/e9EOiu8UfpauE

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mahes>aws --version
aws-cli/2.17.49 Python/3.11.9 Windows/10 exe/AMD64

C:\Users\mahes>aws configure
AWS Access Key ID [*****HOSI]: AKIAQIQANWR4ABDHOSI
AWS Secret Access Key [*****pauE]: /UcXgSaY4phLjKKddy5TJeYPnz/e9EOiu8UfpauE
Default region name [us-east-1]:
Default output format [json]:
```

C:\Users\mahes>

Our Local AWS CLI login into AWS account, As above.

2. Install Kubectl in our cmd.

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mahes>aws --version
aws-cli/2.17.49 Python/3.11.9 Windows/10 exe/AMD64

C:\Users\mahes>aws configure
AWS Access Key ID [*****HOSI]: AKIAJAXQIQUANWR4ABDHOSI
AWS Secret Access Key [*****pauE]: /UcXg5aY4phLjKKddy5TJeYPnz/e9EOiu8UfpauE
Default region name [us-east-1]:
Default output format [json]

C:\Users\mahes>curl -LO "https://dl.k8s.io/release/v1.28.0/bin/windows/amd64/kubectl.exe"
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
100  138  100  138    0     0  347      0 --:--:-- --:--:-- 350
100 48.2M  100 48.2M   0     0  9785k   0:00:05  0:00:05 --:--:-- 10.6M

C:\Users\mahes>kubectl version --client
Client Version: v1.28.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3

C:\Users\mahes>
```

```
C:\Users\mahes>kubectl version --client
Client Version: v1.28.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3

C:\Users\mahes>kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create      Create a resource from a file or from stdin
  expose      Take a replication controller, service, deployment or pod and expose it as a new Kubernetes service
  run         Run a particular image on the cluster
  set         Set specific features on objects

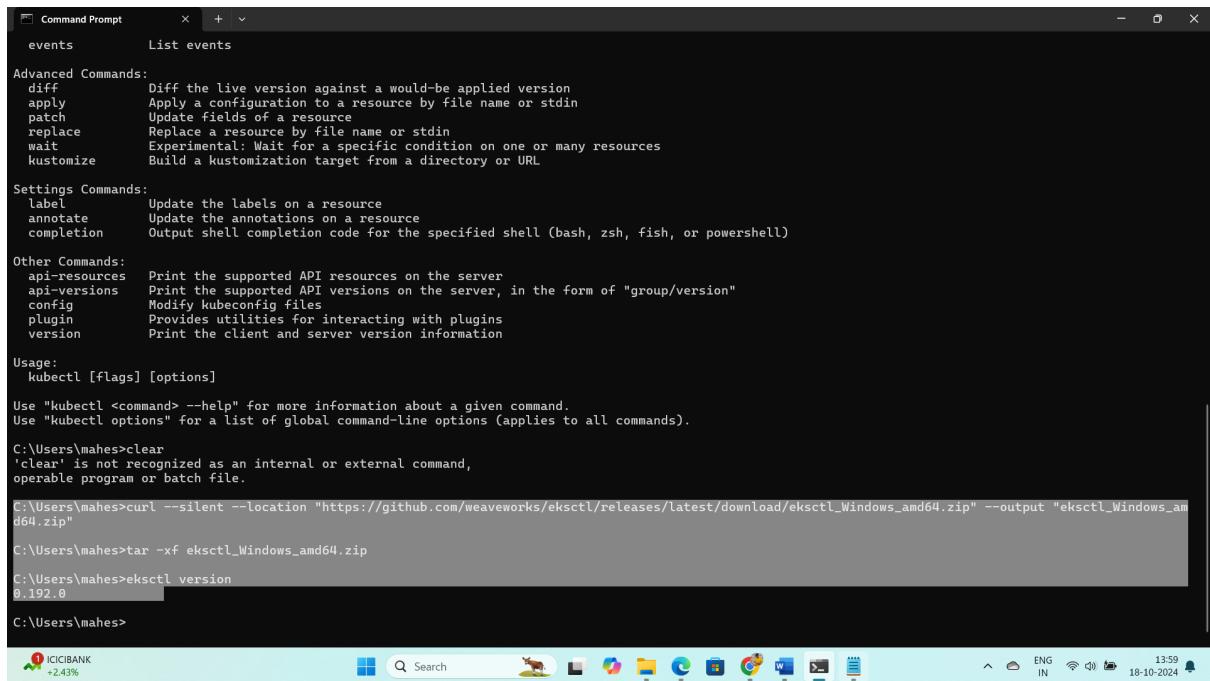
Basic Commands (Intermediate):
  explain     Get documentation for a resource
  get         Display one or many resources
  edit        Edit a resource on the server
  delete      Delete resources by file names, stdin, resources and names, or by resources and label selector

Deploy Commands:
  rollout     Manage the rollout of a resource
  scale       Set a new size for a deployment, replica set, or replication controller
  autoscale   Auto-scale a deployment, replica set, stateful set, or replication controller

Cluster Management Commands:
  certificate Modify certificate resources
  cluster-info Display cluster information
  top          Display resource (CPU/memory) usage
  cordon      Mark node as unschedulable
  uncordon    Mark node as schedulable
  drain       Drain node in preparation for maintenance
  taint       Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe    Show details of a specific resource or group of resources
  logs        Print the logs for a container in a pod
  attach      Attach to a running container
  exec        Execute a command in a container
  port-forward Forward one or more local ports to a pod
```

3. Install eksctl in our cmd



```
Command Prompt
events      List events

Advanced Commands:
diff        Diff the live version against a would-be applied version
apply       Apply a configuration to a resource by file name or stdin
patch       Update fields of a resource
replace     Replace a resource by file name or stdin
wait        Experimental: Wait for a specific condition on one or many resources
kustomize   Build a kustomization target from a directory or URL

Settings Commands:
label       Update the labels on a resource
annotate    Update the annotations on a resource
completion  Output shell completion code for the specified shell (bash, zsh, fish, or powershell)

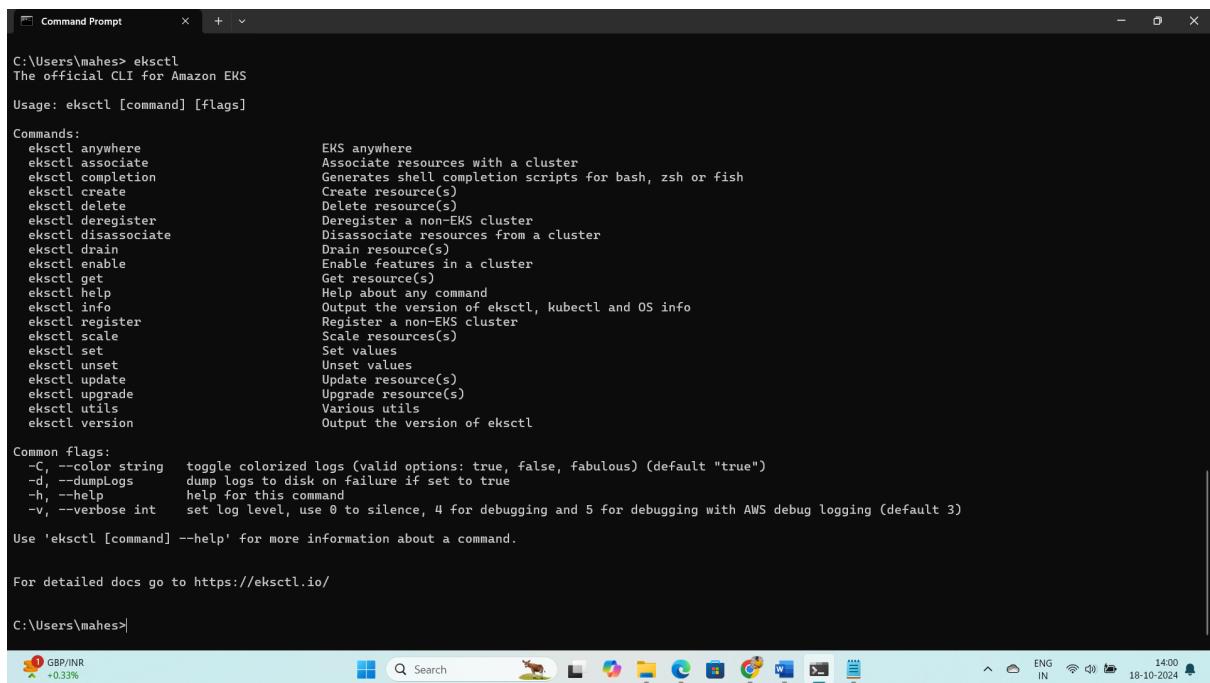
Other Commands:
api-resources Print the supported API resources on the server
api-versions  Print the supported API versions on the server, in the form of "group/version"
config       Modify kubeconfig files
plugin      Provides utilities for interacting with plugins
version      Print the client and server version information

Usage:
kubectl [flags] [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).

C:\Users\mahes>clear
'clear' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\mahes>curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Windows_amd64.zip" --output "eksctl_Windows_amd64.zip"
C:\Users\mahes>tar -xf eksctl_Windows_amd64.zip
C:\Users\mahes>eksctl version
0.192.0
C:\Users\mahes>
```



```
Command Prompt
C:\Users\mahes> eksctl
The official CLI for Amazon EKS

Usage: eksctl [command] [flags]

Commands:
eksctl anywhere          EKS anywhere
eksctl associate         Associate resources with a cluster
eksctl completion        Generates shell completion scripts for bash, zsh or fish
eksctl create             Create resource(s)
eksctl delete             Delete resource(s)
eksctl deregister        Deregister a non-EKS cluster
eksctl disassociate      Disassociate resources from a cluster
eksctl drain              Drain resource(s)
eksctl enable             Enable features in a cluster
eksctl get                Get resource(s)
eksctl help               Help about any command
eksctl info               Output the version of eksctl, kubectl and OS info
eksctl register           Register a non-EKS cluster
eksctl scale              Scale resources(s)
eksctl set                Set values
eksctl unset              Unset values
eksctl update             Update resource(s)
eksctl upgrade            Upgrade resource(s)
eksctl utils              Various utils
eksctl version            Output the version of eksctl

Common flags:
-C, --color string    toggle colorized logs (valid options: true, false, fabulous) (default "true")
-d, --dumplogs         dump logs to disk on failure if set to true
-h, --help              help for this command
-v, --verbose int     set log level, use 0 to silence, 4 for debugging and 5 for debugging with AWS debug logging (default 3)

Use 'eksctl [command] --help' for more information about a command.

For detailed docs go to https://eksctl.io/

C:\Users\mahes>
```

Now we will create EKS cluster using cmd.

eksctl create cluster --name demo-cluster --region us-east-1 --fargate

It takes some time to create EKS cluster

EKSCTL take care of all of control plane.

```
2024-10-18 17:01:39 [+] kubeconfig has been updated
2024-10-18 17:01:39 [+] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2024-10-18 17:01:43 [+] 1 task: { delete cluster control plane "demo-cluster" [async] }
2024-10-18 17:01:43 [+] will delete stack "eksctl-demo-cluster-cluster"
2024-10-18 17:01:44 [+] all cluster resources were deleted

C:\Users\mahesh\eksctl create cluster --name demo-cluster --region us-east-1 --fargate
2024-10-19 13:07:28 [+] eksctl version 0.192.0
2024-10-19 13:07:28 [+] using region us-east-1
2024-10-19 13:07:22 [+] setting availability zones to [us-east-1b us-east-1a]
2024-10-19 13:07:22 [+] subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
2024-10-19 13:07:22 [+] subnets for us-east-1a - public:192.168.32.0/19 private:192.168.96.0/19
2024-10-19 13:07:22 [+] using Kubernetes version 1.30
2024-10-19 13:07:22 [+] creating EKS cluster "demo-cluster" in "us-east-1" region with Fargate profile
2024-10-19 13:07:23 [+] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=demo-cluster'
2024-10-19 13:07:23 [+] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "demo-cluster" in "us-east-1"
2024-10-19 13:07:23 [+] CloudWatch Logging will not be enabled for cluster "demo-cluster" in "us-east-1"
2024-10-19 13:07:23 [+] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=demo-cluster'
2024-10-19 13:07:23 [+] default addons kube-proxy, coredns, vpc-cni were not specified, will install them as EKS addons
2024-10-19 13:07:23 [+] 2 sequential tasks: { create cluster control plane "demo-cluster",
  3 sequential sub-tasks: {
    1 task: { create addons },
    wait for control plane to become ready,
    create fargate profiles,
  }
}
2024-10-19 13:07:23 [+] building cluster stack "eksctl-demo-cluster-cluster"
2024-10-19 13:07:24 [+] deploying stack "eksctl-demo-cluster-cluster"
2024-10-19 13:07:24 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:08:25 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:09:26 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:10:27 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:11:28 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:12:29 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:13:30 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:14:31 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:15:32 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:14:31 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:15:32 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
```

29°C Partly sunny 13:19 ENG IN 19-10-2024

```
2024-10-19 13:07:23 [+] using Kubernetes version 1.30
2024-10-19 13:07:22 [+] creating EKS cluster "demo-cluster" in "us-east-1" region with Fargate profile
2024-10-19 13:07:23 [+] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=demo-cluster'
2024-10-19 13:07:23 [+] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "demo-cluster" in "us-east-1"
2024-10-19 13:07:23 [+] CloudWatch Logging will not be enabled for cluster "demo-cluster" in "us-east-1"
2024-10-19 13:07:23 [+] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=demo-cluster'
2024-10-19 13:07:23 [+] default addons kube-proxy, coredns, vpc-cni were not specified, will install them as EKS addons
2024-10-19 13:07:23 [+] 2 sequential tasks: { create cluster control plane "demo-cluster",
  3 sequential sub-tasks: {
    1 task: { create addons },
    wait for control plane to become ready,
    create fargate profiles,
  }
}
2024-10-19 13:07:23 [+] building cluster stack "eksctl-demo-cluster-cluster"
2024-10-19 13:07:24 [+] deploying stack "eksctl-demo-cluster-cluster"
2024-10-19 13:07:54 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:08:25 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:09:26 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:10:27 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:11:28 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:12:29 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:13:30 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:14:31 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:15:32 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:16:32 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:17:33 [+] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:17:38 [+] creating addon
2024-10-19 13:17:39 [+] successfully created addon
2024-10-19 13:17:39 [+] creating addon
2024-10-19 13:17:40 [+] successfully created addon
2024-10-19 13:17:41 [+] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2024-10-19 13:17:41 [+] creating addon
2024-10-19 13:17:41 [+] successfully created addon
```

29°C Partly sunny 13:19 ENG IN 19-10-2024

```

Command Prompt x + -
2024-10-19 13:16:32 [i] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:17:33 [i] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-19 13:17:38 [i] creating addon
2024-10-19 13:17:39 [i] successfully created addon
2024-10-19 13:17:39 [i] creating addon
2024-10-19 13:17:40 [i] successfully created addon
2024-10-19 13:17:41 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the request ed permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2024-10-19 13:17:41 [i] creating addon
2024-10-19 13:19:45 [i] creating Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-19 13:21:56 [i] created Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-19 13:22:28 [i] "coredns" is now schedulable onto Fargate
2024-10-19 13:23:08 [i] "coredns" is now scheduled onto Fargate
2024-10-19 13:23:35 [i] "coredns" pods are now scheduled onto Fargate
2024-10-19 13:23:35 [i] waiting for the control plane to become ready
2024-10-19 13:23:40 [i] saved kubeconfig as "C:\\Users\\mahes\\.kube\\config"
2024-10-19 13:23:40 [i] no tasks
2024-10-19 13:23:40 [i] all EKS cluster resources for "demo-cluster" have been created
2024-10-19 13:23:40 [i] created 0 nodegroup(s) in cluster "demo-cluster"
2024-10-19 13:23:40 [i] created 0 managed nodegroup(s) in cluster "demo-cluster"
2024-10-19 13:23:40 [x] kubectl not found; v1.10.0 or newer is required
2024-10-19 13:23:40 [i] cluster should be functional despite missing (or misconfigured) client binaries
2024-10-19 13:23:40 [i] EKS cluster "demo-cluster" in "us-east-1" region is ready

C:\Users\mahes>kubectl version --client
Client Version: v1.28.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3

C:\Users\mahes>curl -LO "https://dl.k8s.io/release/v1.31.0/bin/windows/amd64/kubectl.exe"
% Total    % Received % Xferd  Average Speed   Time   Time Current
          Dload  Upload Total   Spent   Left Speed
100  138 100  138    0     0      274      0 --:--:-- --:--:-- 277
100 55.2M 100 55.2M    0     0  9763k  0:00:05  0:00:05 --:--:-- 10.6M

C:\Users\mahes>kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
C:\Users\mahes>

```

EKS cluster created and kubectl also updated

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created
demo-cluster	Active	1.30	Standard support until July 28, 2025	Extended	30 minutes ago

A screenshot of the AWS EKS Cluster Overview page for a cluster named "demo-cluster". A prominent blue banner at the top states: "A new Kubernetes version is available for this cluster." Below this, a message indicates that standard support for Kubernetes version 1.30 ends on July 28, 2025, and provides a link to the pricing page. The "Upgrade now" button is highlighted. The "Cluster info" section shows the status as "Active", the Kubernetes version as "1.30", and the support period as "Standard support until July 28, 2025". The provider is listed as "EKS". The "Overview" tab is selected, and other tabs include Resources, Compute, Networking, Add-ons, Access, Observability, Upgrade insights, and Update history.

A screenshot of the AWS EKS Cluster Details page for the same "demo-cluster". The "Details" section displays the API server endpoint (<https://5E8188594CA3474DEA1DAC86A99B75AF.sk1.us-east-1.eks.amazonaws.com>) and the OpenID Connect provider URL (<https://oidc.eks.us-east-1.amazonaws.com/id/5E8188594CA3474DEA1DAC86A99B75AF>). Other details shown include the creation time ("31 minutes ago"), Cluster ARN, and Platform version ("eks.12"). The "Overview" tab is selected, and other tabs are visible.

A screenshot of the AWS EKS Cluster Details page for the "demo-cluster". The "Kubernetes version settings" section shows the upgrade policy as "Extended". The "Manage" button is visible. The "Overview" tab is selected, and other tabs are visible.

Here the open ID means we can integrate with external MFA like Octa, LDAP, IAM,.....

Amazon Elastic Kubernetes Service

Clusters

Amazon EKS Anywhere

Related services

Amazon ECR

AWS Batch

Console settings

Documentation

Submit feedback

Resource types

Workloads

PodTemplates

Pods

ReplicaSets

Deployments

StatefulSets

DaemonSets

Jobs

CronJobs

PriorityClasses

HorizontalPodAutoscalers

Cluster

Service and networking

Config and secrets

Workloads: Pods (0)

Pod is the smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster.

All Namespaces Filter Pods by name

Name Age

No Pods

This cluster does not have any Pods, or you don't have permission to view them.

CloudShell Feedback

30°C Partly sunny

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 13:39 19-10-2024

BY using the resource tap we can check the pod and resource details and all.

Amazon Elastic Kubernetes Service

Clusters

Amazon EKS Anywhere

Related services

Amazon ECR

AWS Batch

Console settings

Documentation

Submit feedback

Nodes (2) Info

Filter Nodes by property or value

Node name	Instance type	Node group	Created	Status
fargate-ip-192-168-122-118.ec2.internal	Fargate	-	Created 17 minutes ago	Ready
fargate-ip-192-168-81-84.ec2.internal	Fargate	-	Created 17 minutes ago	Ready

Node groups (0) Info

Edit Delete Add node group

Group name Desired size AMI release version Launch template Status

No node groups

This cluster does not have any node groups.

Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.

Add node group

CloudShell Feedback

30°C Partly sunny

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 13:40 19-10-2024

Here we have fargat we have serverless instance type

demo-cluster | Clusters | Elastic Kubernetes Service

us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters/demo-cluster?selectedTab=cluster-networking-tab

aws Services Search [Alt+S] N. Virginia Mahesh23

VPC Console Home

Amazon Elastic Kubernetes Service

Clusters

- Amazon EKS Anywhere
- Enterprise Subscriptions New

Related services

- Amazon ECR
- AWS Batch

Console settings Documentation Submit feedback

Cluster info

Status	Kubernetes version	Info	Support period	Provider
Active	1.30		Standard support until July 28, 2025	EKS

Networking

Manage VPC resources Manage endpoint access

VPC Info	Subnets	Cluster security group Info	API server endpoint access Info
vpc-0742ddbadc121bc64	subnet-010347a72f2b862c9 subnet-00d26087c4e9c7fd7 subnet-0513710816b7ac1b5 subnet-0b095ad7121231599	sg-0aba85fafd75ea0b0	Public
Cluster IP address family Info	IPv4	Additional security groups	Public access source allowlist
	Service IPv4 range Info	sg-09ff12f9706625f2	0.0.0.0/0 (open to all traffic)
	10.100.0.0/16		

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 30°C Partly sunny ENG IN 13:41 19-10-2024

demo-cluster | Clusters | Elastic Kubernetes Service

us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#/clusters/demo-cluster?selectedTab=cluster-logging-tab

aws Services Search [Alt+S] N. Virginia Mahesh23

VPC Console Home

Amazon Elastic Kubernetes Service

Clusters

- Amazon EKS Anywhere
- Enterprise Subscriptions New

Related services

- Amazon ECR
- AWS Batch

Console settings Documentation Submit feedback

Prometheus metrics

Collect and send cluster, infrastructure, and application Prometheus metrics to Amazon Managed Service for Prometheus.

Scrapers (0) Info

Scrapers collect and send metrics to Amazon Managed Service for Prometheus.

Delete Add scraper

Scraper ID	Scraper alias	Status	Created
No scrapers			

There are no scrapers associated with this cluster.

Add scraper

CloudWatch

Amazon CloudWatch delivers end-to-end observability for your applications, networks, and infrastructure. Use the CloudWatch Observability add-on to automatically instrument and collect standardized application metrics and detailed infrastructure telemetry for your Amazon EKS clusters. Use curated out-of-the-box dashboards to drill into application and infrastructure telemetry in CloudWatch Application Signals and CloudWatch Container Insights.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 30°C Partly sunny ENG IN 15:18 19-10-2024

Screenshot of the AWS EKS Cluster Control Plane Logging configuration page.

The left sidebar shows navigation options: Clusters, Amazon EKS Anywhere, Enterprise Subscriptions, Related services (Amazon ECR, AWS Batch), Console settings, Documentation, and Submit feedback.

The main content area displays the CloudWatch section, which includes a note about enabling CloudWatch Observability add-on. It also shows the Control plane logging section with the following configuration:

API server	Authenticator	Scheduler
off	off	off
Audit	Controller manager	
off	off	

Buttons for "Add scraper" and "Manage logging" are present.

Screenshot of the "Manage logging" configuration page for the demo-cluster.

The URL is <https://us-east-1.console.aws.amazon.com/eks/home?region=us-east-1#clusters/demo-cluster/manage-logging>.

The left sidebar shows the EKS navigation path: EKS > Clusters > demo-cluster > Control plane logging > Manage logging.

The main content area shows the "Control plane logging" configuration with the following options:

- API server: Logs pertaining to API requests to the cluster.
- Audit: Logs pertaining to cluster access via the Kubernetes API.
- Authenticator: Logs pertaining to authentication requests into the cluster.
- Controller manager: Logs pertaining to state of cluster controllers.
- Scheduler: Logs pertaining to scheduling decisions.

Buttons for "Cancel" and "Save changes" are at the bottom.

Screenshot of the CloudShell interface.

The URL is <https://us-east-1.console.aws.amazon.com/>.

The top bar shows the AWS logo, Services, Search, and user information (N. Virginia, Mahesh23).

The main area is labeled "CloudShell Feedback".

The screenshot shows the AWS EKS Cluster Logging interface. On the left, there's a sidebar with 'Amazon Elastic Kubernetes Service' and sections for 'Clusters' (Amazon EKS Anywhere, Enterprise Subscriptions), 'Related services' (Amazon ECR, AWS Batch), and 'Console settings'. The main area has a 'CloudWatch' section with a note about enabling observability through the CloudWatch Observability add-on. Below it is a 'Control plane logging' section with status for API server (on), Authenticator (off), Audit (off), Controller manager (off), and Scheduler (off). A 'Manage logging' button is also present.

Now we update the kubeconfig file as shown in the below.

```

ed permissions: the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2024-10-19 13:17:41 [+] creating addon
2024-10-19 13:17:41 [+] successfully created addon
2024-10-19 13:19:45 [+] creating Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-19 13:21:56 [+] created Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-19 13:22:28 [+] "coredns" is now scheduleable onto Fargate
2024-10-19 13:23:35 [+] "coredns" is now scheduled onto Fargate
2024-10-19 13:23:35 [+] "coredns" pods are now scheduled onto Fargate
2024-10-19 13:23:35 [+] waiting for the control plane to become ready
2024-10-19 13:23:40 [+] saved kubeconfig as "C:\\Users\\mahes\\kube\\config"
2024-10-19 13:23:40 [+] no tasks
2024-10-19 13:23:40 [+] all EKS cluster resources for "demo-cluster" have been created
2024-10-19 13:23:40 [+] created 0 nodegroup(s) in cluster "demo-cluster"
2024-10-19 13:23:40 [+] created 0 managed nodegroup(s) in cluster "demo-cluster"
2024-10-19 13:23:40 [+] kubectl not found, v1.18.0 or newer is required
2024-10-19 13:23:40 [+] cluster should be ready despite missing (or misconfigured) client binaries
2024-10-19 13:23:40 [+] EKS cluster "demo-cluster" in "us-east-1" region is ready

C:\\Users\\mahes>kubectl version --client
Client Version: v1.28.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3

C:\\Users\\mahes>curl -LO "https://dl.k8s.io/release/v1.31.0/bin/windows/amd64/kubectl.exe"
% Total    % Received % Xferd  Average Speed   Time   Time Current
          DLoad  Upload  Total Spent  Left Speed
100  138  100  138    0     274   0:--:-- --:--:-- 277
100 55.2M 100 55.2M    0     9763k  0:00:05 0:00:05 --:-- 10.6M

C:\\Users\\mahes>kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2

C:\\Users\\mahes>aws eks update-kubeconfig --name demo-cluster --region us-east-1
Added new context arn:aws:eks:us-east-1:515966528931:cluster/demo-cluster to C:\\Users\\mahes\\.kube\\config

C:\\Users\\mahes>

```

Now we go deploy with application pod

The screenshot shows the AWS EKS console interface for the 'demo-cluster'. On the left sidebar, there are sections for 'Amazon Elastic Kubernetes Service', 'Clusters', 'Amazon EKS Anywhere', 'Enterprise Subscriptions', 'Related services' (including Amazon ECR and AWS Batch), 'Console settings', 'Documentation', and 'Submit feedback'. The main content area shows two Fargate nodes: 'fargate-ip-192-168-122-118.ec2.internal' and 'fargate-ip-192-168-81-84.ec2.internal', both listed under the 'Fargate' provider. Below the nodes, there's a section for 'Node groups (0)' which states 'No node groups'. Under 'Fargate profiles (1)', there is one profile named 'fp-default' with status 'Active'. At the bottom, there's a 'CloudShell' feedback section and a Windows taskbar at the bottom.

Created fargate profile as shown in the below.

```

Command Prompt
Error: --cluster must be set
C:\Users\mahes> --cluster demo-cluster \
'--cluster' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\mahes> --region us-east-1 \
'--region' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\mahes> --name alb-sample-app \
'--name' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\mahes> eksctl create fargateprofile \ --cluster demo-cluster \ --region us-east-1 \ --name alb-sample-app \ --namespace game-2048
2024-10-19 15:41:40 [X] only one argument is allowed to be used as a name

C:\Users\mahes>eksctl create fargateprofile \
Error: --cluster must be set

C:\Users\mahes>eksctl create fargateprofile \ --cluster demo-cluster
Error: invalid Fargate profile: empty selector namespace

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app
Error: invalid Fargate profile: empty selector namespace

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app
Error: invalid Fargate profile: empty selector namespace

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app
Error: invalid Fargate profile: empty selector namespace

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace default
2024-10-19 15:49:23 [I] creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-19 15:50:30 [I] created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-10-19 15:54:43 [I] creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-19 15:55:17 [I] created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>

```

Tried second time

-Created a EKS cluster.

-Updated kube configure file.

-created fargate profile.

```

Command Prompt
2024-10-22 10:48:06 [!] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-22 10:49:07 [!] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-22 10:50:08 [!] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-22 10:51:09 [!] waiting for CloudFormation stack "eksctl-demo-cluster-cluster"
2024-10-22 10:51:14 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2024-10-22 10:51:14 [!] creating addon
2024-10-22 10:51:15 [!] creating addon
2024-10-22 10:51:16 [!] successfully created addon
2024-10-22 10:51:16 [!] creating addon
2024-10-22 10:51:17 [!] successfully created addon
2024-10-22 10:53:20 [!] creating Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-22 10:57:40 [!] created Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-22 10:58:12 [!] "coredns" is now schedulable onto Fargate
2024-10-22 10:59:18 [!] "coredns" pods are now scheduled onto Fargate
2024-10-22 10:59:18 [?] waiting for the control plane to become ready
2024-10-22 10:59:21 [?] saved kubeconfig as "C:\\Users\\mahes\\.kube\\config"
2024-10-22 10:59:21 [!] no tasks
2024-10-22 10:59:21 [?] all EKS cluster resources for "demo-cluster" have been created
2024-10-22 10:59:21 [?] created 0 nodegroup(s) in cluster "demo-cluster"
2024-10-22 10:59:21 [?] created 0 managed nodegroup(s) in cluster "demo-cluster"
2024-10-22 10:59:21 [X] kubectl not found, v1.10.0 or newer required
2024-10-22 10:59:21 [?] cluster should be functional despite missing (or misconfigured) client binaries
2024-10-22 10:59:21 [?] EKS cluster "demo-cluster" in "us-east-1" region is ready

C:\Users\mahes>kubectl version
Client Version: v1.31.0
Kustomize Version: v5.4.2
Server Version: v1.30.5-eks-celdseb

C:\Users\mahes>aws eks update-kubeconfig --name demo-cluster --region us-east-1
Updated context arn:aws:eks:us-east-1:515966528931:cluster/demo-cluster in C:\\Users\\mahes\\.kube\\config

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-10-22 11:06:30 [!] creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-22 11:06:48 [!] created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>

```

Fargate profiles (2)

Profile name	Namespaces	Status
alb-sample-app	game-2048	Active
fp-default	default, kube-system	Active

```
Command Prompt + - X
all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2024-10-22 10:51:14 [+] creating addon
2024-10-22 10:51:14 [+] successfully created addon
2024-10-22 10:51:15 [+] creating addon
2024-10-22 10:51:16 [+] successfully created addon
2024-10-22 10:51:16 [+] creating addon
2024-10-22 10:51:17 [+] successfully created addon
2024-10-22 10:53:20 [+] creating Fargate profile "fp-default" on EKS cluster "demo-cluster"
2024-10-22 10:58:12 [+] "coredns" is now schedulable onto Fargate
2024-10-22 10:59:18 [+] "coredns" pods are now scheduled onto Fargate
2024-10-22 10:59:18 [+] waiting for the control plane to become ready
2024-10-22 10:59:21 [+] saved kubeconfig as "C:\\Users\\mahes\\.kube\\config"
2024-10-22 10:59:21 [+] no tasks
2024-10-22 10:59:21 [+] all EKS cluster resources for "demo-cluster" have been created
2024-10-22 10:59:21 [+] created 0 nodegroup(s) in cluster "demo-cluster"
2024-10-22 10:59:21 [+] created 0 managed nodegroup(s) in cluster "demo-cluster"
2024-10-22 10:59:21 [+] kubectl not found, v1.10.0 or newer is required
2024-10-22 10:59:21 [+] cluster should be functional despite missing (or misconfigured) client binaries
2024-10-22 10:59:21 [+] EKS cluster "demo-cluster" in "us-east-1" region is ready

C:\Users\mahes>kubectl version
Client Version: v1.31.0
Kustomize Version: v5.4.2
Server Version: v1.30.5-eks-ce1d5eb

C:\Users\mahes>aws eks update-kubeconfig --name demo-cluster --region us-east-1
Updated context arn:aws:eks:us-east-1:515966528931:cluster/demo-cluster in C:\\Users\\mahes\\.kube\\config

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-10-22 11:06:30 [+] creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-22 11:06:48 [+] created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created

C:\Users\mahes>
```

As of now there is now ingress controller.

```
Command Prompt + - X
2024-10-22 10:59:18 [+] "coredns" pods are now scheduled onto Fargate
2024-10-22 10:59:18 [+] waiting for the control plane to become ready
2024-10-22 10:59:21 [+] saved kubeconfig as "C:\\Users\\mahes\\.kube\\config"
2024-10-22 10:59:21 [+] no tasks
2024-10-22 10:59:21 [+] all EKS cluster resources for "demo-cluster" have been created
2024-10-22 10:59:21 [+] created 0 nodegroup(s) in cluster "demo-cluster"
2024-10-22 10:59:21 [+] created 0 managed nodegroup(s) in cluster "demo-cluster"
2024-10-22 10:59:21 [+] kubectl not found, v1.10.0 or newer is required
2024-10-22 10:59:21 [+] cluster should be functional despite missing (or misconfigured) client binaries
2024-10-22 10:59:21 [+] EKS cluster "demo-cluster" in "us-east-1" region is ready

C:\Users\mahes>kubectl version
Client Version: v1.31.0
Kustomize Version: v5.4.2
Server Version: v1.30.5-eks-ce1d5eb

C:\Users\mahes>eks update-kubeconfig --name demo-cluster --region us-east-1
Updated context arn:aws:eks:us-east-1:515966528931:cluster/demo-cluster in C:\\Users\\mahes\\.kube\\config

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-10-22 11:06:30 [+] creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-22 11:06:48 [+] created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created

C:\Users\mahes>kubectl get pods
No resources found in default namespace.

C:\Users\mahes>kubectl get pods -n game-2048
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1    Running   0          2m5s
deployment-2048-85f8c7d69-8hgnb  1/1    Running   0          2m5s
deployment-2048-85f8c7d69-8pprl  1/1    Running   0          2m5s
deployment-2048-85f8c7d69-n2m99  1/1    Running   0          2m5s
deployment-2048-85f8c7d69-v45bf  1/1    Running   0          2m5s

C:\Users\mahes>
```

```

Command Prompt - kubectl : + -
Client Version: v1.31.0
Kustomize Version: v5.4.2
Server Version: v1.30.5-eks-ce1d5eb

C:\Users\mahes>aws eks update-kubeconfig --name demo-cluster --region us-east-1
Updated context arn:aws:eks:us-east-1:515966528931:cluster/demo-cluster in C:\Users\mahes\.kube\config

C:\Users\mahes>eksctl create fargateprofile --cluster demo-cluster --region us-east-1 --name alb-sample-app --namespace game-2048
2024-10-22 11:06:30 [I]  creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-22 11:06:48 [I]  created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created

C:\Users\mahes>kubectl get pods
No resources found in default namespace.

C:\Users\mahes>kubectl get pods -n game-2048
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-8gnhb  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-8pprl  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-n2m99  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-v45bf  1/1     Running   0          2m5s

C:\Users\mahes>kubectl get pods -n game-2048 -w
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-8gnhb  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-8pprl  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-n2m99  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-v45bf  1/1     Running   0          3m14s

C:\Users\mahes>
C:\Users\mahes>kubectl get svc -n game-2048 -w
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service-2048  NodePort  10.100.4.236 <none>        80:30463/TCP  6m32s
|
```

See here we have type as nodport and no external IP address means

Any one who have access to the VPC can access this applications.

Our aim is to make this access to external world.

```

Command Prompt - kubectl : + -
2024-10-22 11:06:30 [I]  creating Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"
2024-10-22 11:06:48 [I]  created Fargate profile "alb-sample-app" on EKS cluster "demo-cluster"

C:\Users\mahes>kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/examples/2048/2048_full.yaml
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created

C:\Users\mahes>kubectl get pods
No resources found in default namespace.

C:\Users\mahes>kubectl get pods -n game-2048
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-8gnhb  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-8pprl  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-n2m99  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-v45bf  1/1     Running   0          2m5s

C:\Users\mahes>kubectl get pods -n game-2048 -w
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-8gnhb  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-8pprl  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-n2m99  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-v45bf  1/1     Running   0          3m14s

C:\Users\mahes>
C:\Users\mahes>kubectl get svc -n game-2048 -w
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service-2048  NodePort  10.100.4.236 <none>        80:30463/TCP  6m32s
C:\Users\mahes>Kubectl get ingress -n gamee-2048
No resources found in gamee-2048 namespace.

C:\Users\mahes>Kubectl get ingress -n game-2048
NAME      CLASS   HOSTS   ADDRESS   PORTS   AGE
ingress-2048  alb     *       80       10m
C:\Users\mahes>
```

See here host can be any on * and no address means we don't have any ingress controller.

Ingress controller -> create ALB and configure all to take care of

Alb controller is nothing but a POD but to speak with other we should configure iam-oidc

```
Command Prompt
namespace/game-2048 created
deployment.apps/deployment-2048 created
service/service-2048 created
ingress.networking.k8s.io/ingress-2048 created

C:\Users\mahes>kubectl get pods
No resources found in default namespace.

C:\Users\mahes>kubectl get pods -n game-2048
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-8hgnb  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-8pprl  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-n2m99  1/1     Running   0          2m5s
deployment-2048-85f8c7d69-v45bf  1/1     Running   0          2m5s

C:\Users\mahes>kubectl get pods -n game-2048 -w
NAME          READY   STATUS    RESTARTS   AGE
deployment-2048-85f8c7d69-5hx4v  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-8hgnb  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-8pprl  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-n2m99  1/1     Running   0          3m14s
deployment-2048-85f8c7d69-v45bf  1/1     Running   0          3m14s

C:\Users\mahes>
C:\Users\mahes>kubectl get svc -n game-2048 -w
NAME        TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service-2048  NodePort  10.100.4.236 <none>       80:30463/TCP  6m32s

C:\Users\mahes>Kubectl get ingress -n gamee-2048
No resources found in gamee-2048 namespace.

C:\Users\mahes>Kubectl get ingress -n game-2048
NAME   CLASS  HOSTS  ADDRESS  PORTS  AGE
ingress-2048  alb   *       80      10m

C:\Users\mahes>eksctl utils associate-iam-oidc-provider --cluster demo-cluster --approve
2024-10-22 12:25:56 [+] will create IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"
2024-10-22 12:25:57 [+] created IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"

C:\Users\mahes>
```

abhishekveeramalla@aveerama-mac eks %
abhishekveeramalla@aveerama-mac eks % ALB controller -> Pod -> Access to AWS services such as ALB

Create an IAM policy and roles for this.

Download IAM policy/ Create IAM Policy

```
Command Prompt
deployment-2048-85f8c7d69-v45bf  1/1     Running   3m14s

C:\Users\mahes>
C:\Users\mahes>kubectl get svc -n game-2048 -w
NAME        TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service-2048  NodePort  10.100.4.236 <none>       80:30463/TCP  6m32s

C:\Users\mahes>Kubectl get ingress -n gamee-2048
No resources found in gamee-2048 namespace.

C:\Users\mahes>Kubectl get ingress -n game-2048
NAME   CLASS  HOSTS  ADDRESS  PORTS  AGE
ingress-2048  alb   *       80      10m

C:\Users\mahes>eksctl utils associate-iam-oidc-provider --cluster demo-cluster --approve
2024-10-22 12:25:56 [+] will create IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"
2024-10-22 12:25:57 [+] created IAM Open ID Connect provider for cluster "demo-cluster" in "us-east-1"

C:\Users\mahes>curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/install/iam_policy.json
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 8386 100 8386 0 0 23479 0 ---:--- ---:--- ---:--- 23756

C:\Users\mahes>aws iam create-policy --policy-name AWSLoadBalancerControllerIAMPoly --policy-document file://iam_policy.json
{
  "Policy": {
    "PolicyName": "AWSLoadBalancerControllerIAMPoly",
    "PolicyId": "ANPAXQIQANWRST7NG5IND",
    "Arn": "arn:aws:iam::515966528931:policy/AWSLoadBalancerControllerIAMPoly",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2024-10-22T07:01:48+00:00",
    "UpdateDate": "2024-10-22T07:01:48+00:00"
  }
}

C:\Users\mahes>
```

Create IAM Role

```

C:\Users\mahes>aws iam create-policy --policy-name AWSLoadBalancerControllerIAMPolicy --policy-document file:///iam_policy.json
{
  "Policy": {
    "PolicyName": "AWSLoadBalancerControllerIAMPolicy",
    "PolicyId": "ANPAXQIQANWR577NGSIND",
    "Arn": "arn:aws:iam::515966528931:policy/AWSLoadBalancerControllerIAMPolicy",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2024-10-22T07:01:48+00:00",
    "UpdateDate": "2024-10-22T07:01:48+00:00"
  }
}

C:\Users\mahes>eksctl create iamserviceaccount --cluster=demo-cluster --namespace=kube-system --name=aws-load-balancer-controller --role-name AmazonEKSLoadBalancerControllerRole
Error: --attach-policy-arn or --attach-role-arn must be set

C:\Users\mahes> --attach-policy-arn=arn:aws:iam::515966528931:policy/AWSLoadBalancerControllerIAMPolicy --approve
'--attach-policy-arn' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\mahes>eksctl create iamserviceaccount --cluster=demo-cluster --namespace=kube-system --name=aws-load-balancer-controller --role-name AmazonEKSLoadBalancerControllerRole --attach-policy-arn=arn:aws:iam::515966528931:policy/AWSLoadBalancerControllerIAMPolicy --approve
2024-10-22 12:39:13 [+] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2024-10-22 12:39:13 [+] serviceaccounts that exist in Kubernetes will be excluded, use --override-existing-serviceaccounts to override
2024-10-22 12:39:13 [+] 1 task:
  2 sequential sub-tasks:
    create IAM role for serviceaccount "kube-system/aws-load-balancer-controller",
    create serviceaccount "kube-system/aws-load-balancer-controller",
  } }2024-10-22 12:39:13 [+] building iamserviceaccount stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-10-22 12:39:14 [+] deploying stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-10-22 12:39:14 [+] waiting for CloudFormation stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-10-22 12:39:45 [+] waiting for CloudFormation stack "eksctl-demo-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-10-22 12:39:46 [+] created serviceaccount "kube-system/aws-load-balancer-controller"

C:\Users\mahes>

```

Deploy ALB controller

Add helm repo

```

C:\Users\mahes>helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories

C:\Users\mahes>

```

Install

```
Command Prompt + - X

--debug          enable verbose output
-h, --help        help for helm
--kube-apiserver string   the address and the port for the Kubernetes API server
--kube-as-group stringArray group to impersonate for the operation, this flag can be repeated to specify multiple groups.
--kube-as-user string    username to impersonate for the operation
--kube-ca-file string    the certificate authority file for the Kubernetes API server connection
--kube-context string    name of the kubeconfig context to use
--kube-insecure-skip-tls-verify if true, the Kubernetes API server's certificate will not be checked for validity. This will make your HTTPS connections insecure
--kube-tls-server-name string  server name to use for Kubernetes API server certificate validation. If it is not provided, the hostname used to contact the server is used
--kube-token string      bearer token used for authentication
--kubeconfig string      path to the kubeconfig file
-n, --namespace string   namespace scope for this request
--qps float32            queries per second used when communicating with the Kubernetes API, not including bursting
--registry-config string path to the registry config file (default "C:\\Users\\mahes\\AppData\\Roaming\\helm\\registry\\config.json")
--repository-cache string path to the directory containing cached repository indexes (default "C:\\Users\\mahes\\AppData\\Local\\Temp\\helm\\cache")
--repository-config string path to the file containing repository names and URLs (default "C:\\Users\\mahes\\AppData\\Roaming\\helm\\repositories.yaml")

Use "helm [command] --help" for more information about a command.

C:\Users\mahes>helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories

C:\Users\mahes>helm install aws-load-balancer-controller eks/aws-load-balancer-controller -n kube-system --set clusterName=demo-custer --set serviceAccount.create=false --set serviceAccount.name=aws-load-balancer-controller --set region=us-east-1 --set vpcId=vpc-0a8d161860e2517d0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Tue Oct 22 16:51:55 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!

C:\Users\mahes>
```

Verify that the deployments are running.

```
Command Prompt + - X

--kube-ca-file string    the certificate authority file for the Kubernetes API server connection
--kube-context string    name of the kubeconfig context to use
--kube-insecure-skip-tls-verify if true, the Kubernetes API server's certificate will not be checked for validity. This will make your HTTPS connections insecure
--kube-tls-server-name string  server name to use for Kubernetes API server certificate validation. If it is not provided, the hostname used to contact the server is used
--kube-token string      bearer token used for authentication
--kubeconfig string      path to the kubeconfig file
-n, --namespace string   namespace scope for this request
--qps float32            queries per second used when communicating with the Kubernetes API, not including bursting
--registry-config string path to the registry config file (default "C:\\Users\\mahes\\AppData\\Roaming\\helm\\registry\\config.json")
--repository-cache string path to the directory containing cached repository indexes (default "C:\\Users\\mahes\\AppData\\Local\\Temp\\helm\\cache")
--repository-config string path to the file containing repository names and URLs (default "C:\\Users\\mahes\\AppData\\Roaming\\helm\\repositories.yaml")

Use "helm [command] --help" for more information about a command.

C:\Users\mahes>helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories

C:\Users\mahes>helm install aws-load-balancer-controller eks/aws-load-balancer-controller -n kube-system --set clusterName=demo-custer --set serviceAccount.create=false --set serviceAccount.name=aws-load-balancer-controller --set region=us-east-1 --set vpcId=vpc-0a8d161860e2517d0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Tue Oct 22 16:51:55 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller  0/2     2           0           52s

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller  2/2     2           2           90s

C:\Users\mahes>
```

```

Command Prompt      + - X
--repository-config string      path to the file containing repository names and URLs (default "C:\Users\mahes\AppData\Roaming\helm\repositories.yaml")
Use "helm [command] --help" for more information about a command.

C:\Users\mahes>helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories

C:\Users\mahes>helm install aws-load-balancer-controller eks/aws-load-balancer-controller --set clusterName=demo-cluster --set serviceAccount.create=false --set serviceAccount.name=aws-load-balancer-controller --set region=us-east-1 --set vpcId=vpc-0a8d161860e2517d0
NAME: aws-load-balancer-controller
LAST DEPLOYED: Tue Oct 22 16:51:55 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller  0/2       2           0          52s

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller  2/2       2           2          90s

C:\Users\mahes>kubectl gte pods -n kube-system
error: unknown command "gte" for "kubectl"

Did you mean this?
  get

C:\Users\mahes>kubectl get pods -n kube-system
NAME                           READY   STATUS    RESTARTS   AGE
aws-load-balancer-controller-5d89446dd5-2l4p4  1/1     Running   0          4m1s
aws-load-balancer-controller-5d89446dd5-qkq6n  1/1     Running   0          4m1s
coredns-6659cb98f6-hlvks        1/1     Running   0          5h57m
coredns-6659cb98f6-mfcfg       1/1     Running   0          5h57m

C:\Users\mahes>

```



demo-cluster | Clusters | Elastic x Home | EC2 | us-east-1

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:

EC2 Dashboard

- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity
 - Reservations **New**
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes
 - Snapshots
- Life cycle Manager

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	1	Instances	0
Key pairs	1	Load balancers	1	Placement groups	0
Security groups	7	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

AWS Health Dashboard

EC2 Free Tier **Info**

Offers for all AWS Regions.

2 EC2 free tier offers in use

End of month forecast

⚠️ 0 offers forecasted to exceed free tier limit.

Exceeds free tier

⚠️ 0 offers exceeded and is now pay-as-you-go pricing.

Offer usage (monthly)

Linux EC2 Instances

648.646389 hours remaining

Storage space on EBS

28.59 GB remaining

[View all AWS Free Tier offers](#)

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers:

Rain to stop
In about 2 hours



The screenshot shows the AWS EC2 Load Balancers console. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Instances, Images, and Elastic Block Store. The main content area displays a table titled 'Load balancers (1)'. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. One row is visible, showing 'k8s-game2048-ingress...' as the Name, 'k8s-game2048-ingress2-3...', 'Active' as the State, 'vpc-0a8d161860e251...', '2 Availability Zones', and 'application' as the Type. Below the table, a message says '0 load balancers selected' and 'Select a load balancer above.' The bottom of the screen shows the Windows taskbar with various pinned icons.

See the ALB created as shown.

The screenshot shows the 'Load balancer details' page for the ALB 'k8s-game2048-ingress2-3d5239bee2'. The left sidebar is identical to the previous screenshot. The main content area is titled 'k8s-game2048-ingress2-3d5239bee2'. It features a 'Details' section with tabs for Load balancer type (Application), Status (Active), VPC (vpc-0a8d161860e2517d0), and Load balancer IP address type (IPv4). It also shows the Hosted zone (Z355XDOTRQ7X7K), Availability Zones (subnet-095956a7378608450, us-east-1d (use1-az2) and subnet-001918511e97a1815, us-east-1a (use1-az4)), and Date created (October 22, 2024, 16:53 (UTC+05:30)). Below this, the Load balancer ARN is listed as arn:aws:elasticloadbalancing:us-east-1:515966528931:loadbalancer/app/k8s-game2048-ingress2-3d5239bee2/2cdc24dac9daf457, and the DNS name is k8s-game2048-ingress2-3d5239bee2-1165673961.us-east-1.elb.amazonaws.com (A Record). At the bottom, there are tabs for Listeners and rules (1), Network mapping, Resource map - new, Security, Monitoring, Integrations, Attributes, and Tags. The bottom of the screen shows the Windows taskbar.

The screenshot shows the AWS Cloud Console interface. On the left, there's a navigation sidebar with options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area displays a load balancer named 'Z35SXDOTRQ7X7K' with the status 'Internet-facing'. It lists two subnets: 'subnet-095956a7378608430' (us-east-1d us-east-1az2) and 'subnet-001918511e97a1815' (us-east-1a us-east-1az4). The DNS name is 'k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com (A Record)'. Below this, a table titled 'Listeners and rules (1) Info' shows one rule for port 80: 'Return fixed response' with options for Response code: 404, Response body, and Response content type: text/plain.

```
C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2      2           2          16m

C:\Users\mahes>
```

The load balancer controller is created based on ingress resource.

```
C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2      2           2          16m

C:\Users\mahes>kubectl get pods -n kube-system
error: unknown command "gte" for "kubectl"

Did you mean this?
    get

C:\Users\mahes>kubectl get pods -n kube-system
NAME                           READY   STATUS    RESTARTS   AGE
aws-load-balancer-controller-5d89446dd5-2l4p4   1/1     Running   0          4m1s
aws-load-balancer-controller-5d89446dd5-qkq6n   1/1     Running   0          4m1s
coredns-6659cb98f6-hlvks   1/1     Running   0          5h57m
coredns-6659cb98f6-mfclg   1/1     Running   0          5h57m

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2      2           2          4m55s

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2      2           2          16m

C:\Users\mahes>kubectl get ingress -n game-2048
NAME        CLASS   HOSTS   ADDRESS          PORTS   AGE
ingress-2048 alb     *       k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com   80      5h14m

C:\Users\mahes>
```

See now the address was allocated as shown

k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com

Screenshot of the AWS Cloud Console showing the Load balancer details page for an Application Load Balancer named "k8s-game2048-ingress2-3d5239bee2".

The "Details" section displays the following information:

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-0a8d161860e2517d0	IPv4
Scheme	Internet-facing	Hosted zone Z355XDOTRQ7X7K	Availability Zones subnet-095956a7378608430 us-east-1d (use1-az2) subnet-001918511e97a1815 us-east-1a (use1-az4)
Load balancer ARN		DNS name info k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com (A Record)	Date created October 22, 2024, 16:53 (UTC+05:30)

The "Listeners and rules" tab is selected, showing one listener rule:

Listener rule	Action
Info	HTTP to HTTPS redirect

The "Network mapping" tab is also visible.

Screenshot of the AWS Cloud Console showing the Network mapping section for the same Application Load Balancer.

The "Network mapping" section displays the following information:

VPC	Load balancer IP address type
vpc-0a8d161860e2517d0	IPv4

The "Mappings" section shows traffic targets across two Availability Zones:

Zone	Subnet	IPv4 address	Private IPv4 address	IPv6 address
us-east-1d (use1-az2)	subnet-095956a7378608430	Assigned by AWS	Assigned from CIDR 192.168.32.0/19	Not applicable
us-east-1a (use1-az4)	subnet-001918511e97a1815	Assigned by AWS	Assigned from CIDR 192.168.0.0/19	Not applicable

Screenshot of the AWS CloudWatch Metrics console showing the Resource map - new view for a Load Balancer.

The Resource map displays the architecture of the load balancer, showing the flow from listeners to rules to target groups.

Listeners (1): HTTP:80, 2 rules.

Rules (2):

- Priority 1: Forward to target group. Conditions (If): Path Pattern is /*.
- Priority default: Fixed response. Conditions (If): If no other rule applies.

Target groups (1) Info: IP k8s-game2048-service2-cccab8960b, 5 targets.

IP	Port	Status
192.168.124.175	Port 80	Healthy
192.168.125.48	Port 80	Healthy
192.168.74.181	Port 80	Healthy
192.168.75.191	Port 80	Healthy
192.168.92.197	Port 80	Healthy

Screenshot of the AWS CloudWatch Metrics console showing the Resource map - new view for a Load Balancer.

The Resource map displays the architecture of the load balancer, showing the flow from listeners to rules to target groups.

Listeners (1): HTTP:80, 2 rules.

Rules (2):

- Priority 1: Forward to target group. Conditions (If): Path Pattern is /*.
- Priority default: Fixed response. Conditions (If): If no other rule applies.

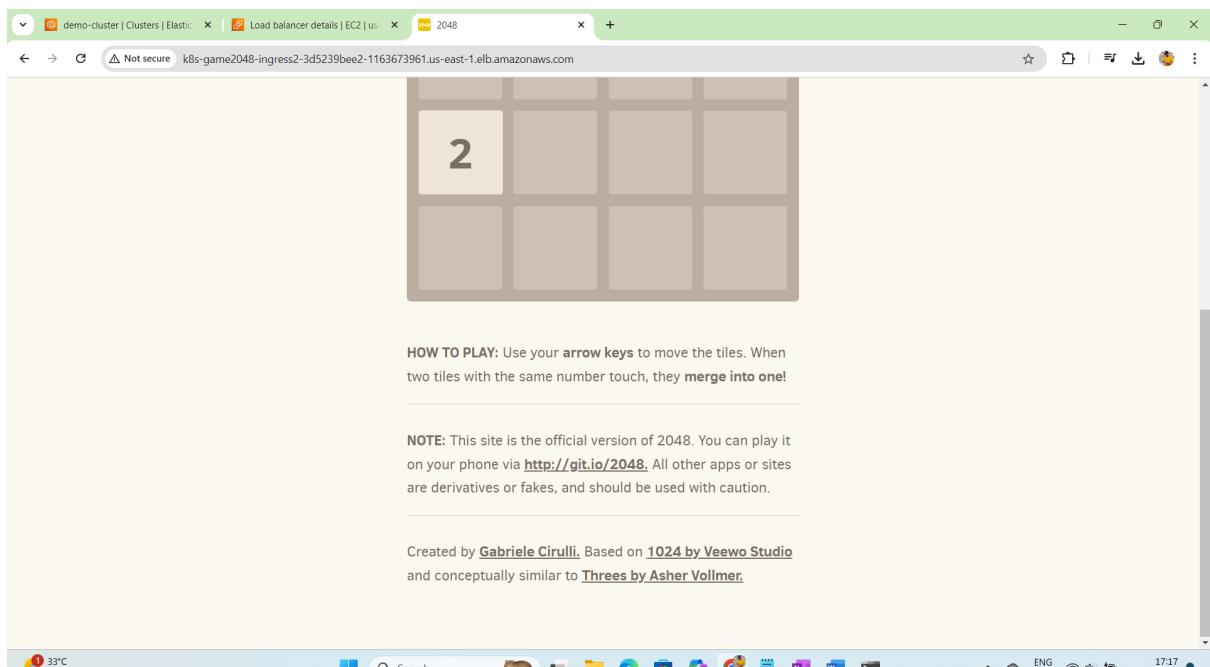
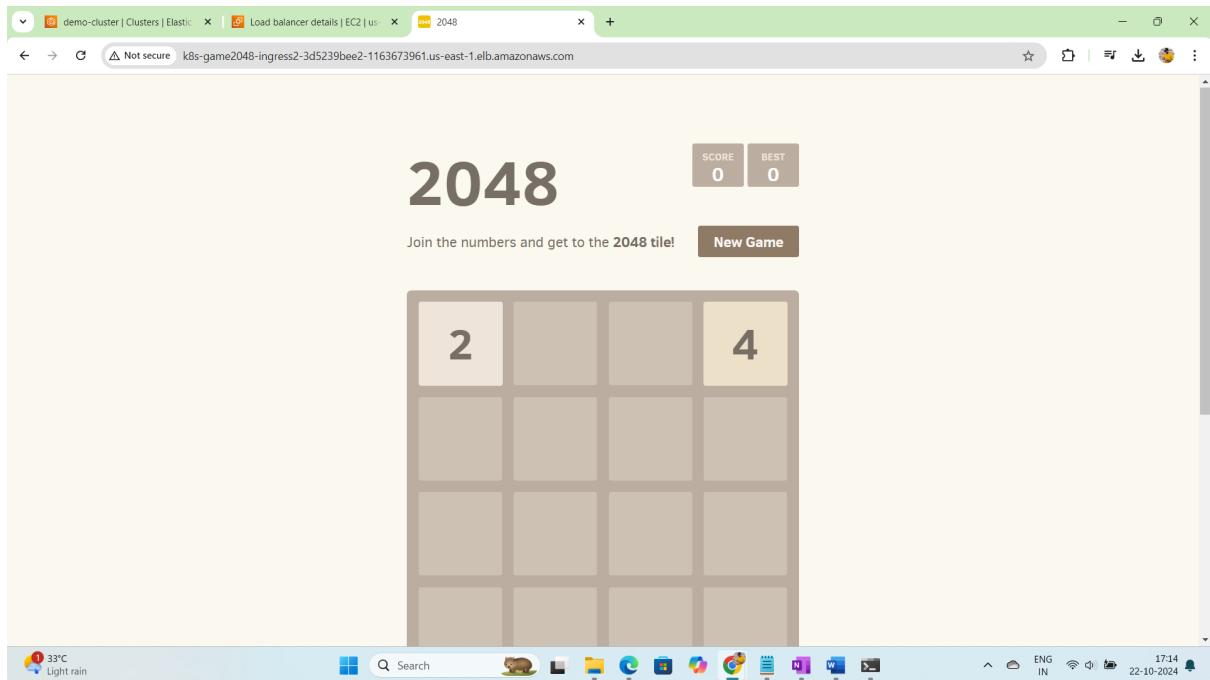
Target groups (1) Info: IP k8s-game2048-service2-cccab8960b, 5 targets.

IP	Port	Status
192.168.124.175	Port 80	Healthy
192.168.125.48	Port 80	Healthy
192.168.74.181	Port 80	Healthy
192.168.75.191	Port 80	Healthy
192.168.92.197	Port 80	Healthy

Screenshot of the AWS Cloud Console showing the Load balancer details for an Application Load Balancer (ALB). The ALB is Internet-facing, hosted in zone Z35SXDOTRQ7X7K, with Availability Zones subnet-095956a7378608430 (us-east-1d) and subnet-001918511e97a1815 (us-east-1a). It was created on October 22, 2024, at 16:53 UTC+05:30. The DNS name is k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com (A Record). The Tags section shows three tags: elbv2.k8s.aws/cluster (Value: demo-cluster), ingress.k8s.aws/resource (Value: LoadBalancer), and ingress.k8s.aws/stack (Value: game-2048/ingress-2048).

See here the DNS is also same with the address as shown above

Screenshot of a Microsoft Edge browser window. The address bar shows the URL <http://k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com>. The page content features a colorful musical illustration with a violin, piano, trumpet, and microphone. Below the illustration is a search bar with the placeholder "Search the web". At the bottom of the browser window, there are links for "Amazon", "Shopping Coupons", and "Add shortcut". The system tray at the bottom of the screen shows the date and time as 22-10-2024, 17:14, and the weather as 33°C Light rain.



Deleting..

```
Command Prompt + - X
C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2           2           90s

C:\Users\mahes>kubectl gte pods -n kube-system
error: unknown command "gte" for "kubectl"

Did you mean this?
    get

C:\Users\mahes>kubectl get pods -n kube-system
NAME          READY   STATUS    RESTARTS   AGE
aws-load-balancer-controller-5d89446dd5-2l4p4  1/1     Running   0          4m1s
aws-load-balancer-controller-5d89446dd5-qkq6n  1/1     Running   0          4m1s
coredns-6659cb98f6-hlvks  1/1     Running   0          5h57m
coredns-6659cb98f6-mfc1g  1/1     Running   0          5h57m

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2           2           4m55s

C:\Users\mahes>kubectl get deployment -n kube-system aws-load-balancer-controller
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2           2           16m

C:\Users\mahes>kubectl get ingress -n game-2048
NAME      CLASS  HOSTS   ADDRESS          PORTS  AGE
ingress-2048  alb    *       k8s-game2048-ingress2-3d5239bee2-1163673961.us-east-1.elb.amazonaws.com  80     5h14m

C:\Users\mahes>eksctl delete cluster --name demo-cluster --region us-east-1
2024-10-22 17:18:12 [+] deleting EKS cluster "demo-cluster"
2024-10-22 17:18:17 [+] deleting Fargate profile "alb-sample-app"
2024-10-22 17:20:27 [+] deleted Fargate profile "alb-sample-app"
2024-10-22 17:20:27 [+] deleting Fargate profile "fp-default"
2024-10-22 17:22:36 [+] deleted Fargate profile "fp-default"
2024-10-22 17:22:36 [+] deleted 2 Fargate profile(s)
2024-10-22 17:22:40 [+] kubeconfig has been updated
2024-10-22 17:22:40 [+] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
Error: deadline surpassed waiting for AWS load balancers to be deleted: k8s-game2048-ingress2-3d5239bee2

C:\Users\mahes>
```



