

Day 2:

The screenshot shows a Microsoft Edge browser window with the URL <https://eksctl.io/usage/spot-instances/>. The page title is "Spot instances". On the left, there's a sidebar with links like "Custom AMI support", "VPC Networking", "GPU Support", "Spot instances" (which is the active section), "IAM", "Customizing kubelet configuration", "CloudWatch logging", "Windows Worker Nodes", "EKS Managed Nodegroups", "EKS Fargate Support", "gitops", "Config file schema", "Troubleshooting", "Minimum IAM policies", "FAQ", "GitOps Quickstart", "Examples", and "Community". The main content area has a heading "Spot instances" and a paragraph about support for spot instances through MixedInstancesPolicy. It includes a code snippet for nodeGroups:

```
nodeGroups:
- name: ng-1
  minSize: 2
  maxSize: 5
  instancesDistribution:
    maxPrice: 0.017
    instanceTypes: ["t3.small", "t3.medium"] # At least one instance type should
    onDemandBaseCapacity: 0
    onDemandPercentageAboveBaseCapacity: 50
  spotInstancePools: 2
```

Below the code snippet, a note says: "Note that the nodeGroups.X.instanceType field shouldn't be set when using the instancesDistribution field." A footer at the bottom of the page says "This example uses GPU instances:".

<https://eksctl.io/usage/eks-managed-nodes/>

```
cluster1 - Notepad
File Edit Format View Help
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: vishcluster
  region: ap-south-1

nodeGroups:
  - name: ng1
    desiredCapacity: 2
    instanceType: t2.micro
    ssh:
      publicKeyName: mykey1111.pem
  - name: ng2
    desiredCapacity: 1
    instanceType: t2.small
    ssh:
      publicKeyName: mykey1111.pem
nodeGroups:
  - name: ng-mixed
```

```
cluster1 - Notepad
File Edit Format View Help
ssh:
  publicKeyName: mykey1111.pem
- name: ng2
  desiredCapacity: 1
  instanceType: t2.small
  ssh:
    publicKeyName: mykey1111.pem
nodeGroups:
  - name: ng-mixed
    minSize: 2
    maxSize: 5
    instancesDistribution:
      maxPrice: 0.017
      instanceTypes: ["t3.small", "t3.medium"]
      onDemandBaseCapacity: 0
      onDemandPercentageAboveBaseCapacity: 50
      spotInstancePools: 2
    ssh:
      publicKeyName: mykey1111.pem
```

Annotations as shown below.

```
apiVersion: v1
kind: Service
metadata:
  name: example
  namespace: kube-system
  labels:
    run: example
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-ssl-cert: arn:aws:acm:xx-xxxx-x:...
    service.beta.kubernetes.io/aws-load-balancer-backend-protocol: http
spec:
  type: LoadBalancer
  ports:
  - port: 443
    targetPort: 5556
    protocol: TCP
  selector:
    app: example
```

Different settings can be applied to a load balancer service in AWS using *annotations*. The

kubeadm
AWS
Node Name
Load Balancers
Azure
Node Name
CloudStack
Node Name
GCE
Node Name
HUAWEI CLOUD
Node Name
OpenStack
Node Name
Services

<https://kubernetes.io/docs/concepts/cluster-administration/cloud-providers/>

Google

kubernetes load balancer providers

All Images News Videos Shopping More Settings Tools

About 6,45,000 results (0.45 seconds)

Ad · aws.amazon.com/ ▾
Fault-Tolerant and Flexible | Run Fault-Tolerant Workloads
EC2 Spot Instances offer spare compute capacity in the AWS cloud at steep discounts. Compute at Up to 90% Off the On-Demand Price and Get Faster Results. Try Now! Download Our Mobile App. Try For Free. View Pricing Details.

Ad · www.ibm.com/load/balancer ▾
IBM® Cloud Load Balancer | Flexible Features & Pricing
Management is Easier with a Graphical Interface and APIs to Monitor Server Health. Get Intelligent Traffic Distribution and SSL Offloading from IBM Cloud **Load Balancer**. Chat with an Expert. Sign Up Today. Discover the Offerings. Explore Key Features.

Ad · www.cloudflare.com/ ▾
Cloudflare Load Balancer | Easy, Scalable, and Secure
Balance traffic across multiple servers or geolocation regions. Start today! Get intelligent global traffic control that optimizes implementation time and protection. Flat-rate Pricing

serverless architecture:

Without managing the worker node , master node
I will manage everything for u

Google search results for "fargate aws":

About 5,66,000 results (0.51 seconds)

aws.amazon.com › fargate ▾

AWS Fargate - Run containers without having to manage ...

AWS Fargate is a compute engine for Amazon ECS and EKS that allows you to run containers without having to manage servers or clusters.

Amazon ECS on AWS Fargate
AWS Fargate is a technology that you can use with Amazon ECS ...

AWS Fargate FAQs
AWS Fargate is a compute engine for Amazon ECS and EKS that ...

AWS Fargate Pricing
AWS Fargate pricing is calculated based on the vCPU and ...

Introducing AWS Fargate
AWS Fargate is an easy way to deploy your containers on AWS.

User Guide for AWS Fargate
AWS Fargate is a technology that you can use with Amazon ECS ...

AWS Fargate Getting Started
AWS Fargate works with services like Amazon SageMaker and ...

More results from amazon.com »

Waiting for www.google.com...

AWS Management Console - Services - Resource Groups - Mumbai - Support

Amazon Elastic Container Service (ECS)

Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.

Get started

The screenshot shows the Amazon ECS homepage. At the top, there are three main sections with icons and text:

- Run containers at scale**: Features an icon of a stack of containers with a plus sign. Below it, text says: "Amazon ECS makes it easy to use containers as a foundational building block for your applications by eliminating the need for you to install, operate, and scale your own cluster management infrastructure."
- Flexible container placement**: Features an icon of a server rack with a gear. Below it, text says: "Amazon ECS lets you schedule long-running applications, services, and batch processes. Amazon ECS maintains application availability and allows you to scale your containers up or down to meet your application's capacity requirements."
- Integrated and extensible**: Features an icon of a server rack with a circular arrow. Below it, text says: "Amazon ECS is integrated with familiar features like Elastic Load Balancing, EBS volumes, VPC, and IAM. Simple APIs let you integrate and use your own schedulers or connect Amazon ECS into your existing software delivery process."

At the bottom of the page, there is a footer with links to "Elastic Container Service documentation and support", "Documentation", "Support", "Forums", and "Contact us". The browser status bar at the bottom shows the URL "ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/getStarted".

fargat is the sub service of the ecs

CNI(container network interface)

To get the connectivity between two different container running in isolated system

In eks they have their own aws cnf plugin which helps the most

About 17,400 results (0.37 seconds)

[docs.aws.amazon.com › eks › latest › userguide › cni-u...](https://docs.aws.amazon.com/eks/latest/userguide/cni-u...)

Amazon VPC CNI plugin for Kubernetes upgrades - AWS ...

When you launch an Amazon EKS cluster, we apply a recent version of the Amazon VPC CNI plugin for Kubernetes to your cluster. The absolute latest version of ...

[docs.aws.amazon.com › eks › latest › userguide › pod-...](https://docs.aws.amazon.com/eks/latest/userguide/pod-...)

Pod networking (CNI) - Amazon EKS - AWS Documentation

Amazon EKS supports native VPC networking via the Amazon VPC Container Network Interface (CNI) plugin for Kubernetes. Using this CNI plugin allows ...

[Amazon VPC CNI plugin for ... · CNI custom networking · CNI configuration variables](https://docs.aws.amazon.com/eks/latest/userguide/cni-c...)

[docs.aws.amazon.com › userguide › cni-custom-network](https://docs.aws.amazon.com/eks/latest/userguide/cni-c...)

CNI custom networking - Amazon EKS - AWS Documentation

The use cases discussed in this topic require the Amazon VPC CNI plugin for Kubernetes version 1.4.0 or later. To check your CNI version, and upgrade if ...

<https://docs.aws.amazon.com/eks/latest/userguide/cni-c...>

CNI custom networking

By default, when new network interfaces are allocated for pods, ipamD uses the worker node's primary elastic network interface's (ENI) security groups and subnet. However, there are use cases where your pod network interfaces should use a different security group or subnet, within the same VPC as your control plane security group. For example:

- There are a limited number of IP addresses available in a subnet. This limits the number of pods that can be created in the cluster. Using different subnets for pod groups allows you to increase the number of available IP addresses.
- For security reasons, your pods must use different security groups or subnets than the node's primary network interface.
- The worker nodes are configured in public subnets and you want the pods to be placed in private subnets using a NAT Gateway. For more information, see [External source network address translation \(SNAT\)](#).

Note

ENI (elastic networking interface)

We can launch the network card as well as ip in a given cidr range

(availability ipv4 range)

using docker ps cmd we can check which all container has outside connectivity

In aws we have limitation like in t2.micro we can't add more than 4 network cards

And in t2.small we can't add more than 11 and so on

AWS EKS SERVICE - Google Docs | Day 2 - Google Docs | AWS EKS - Google Drive | Coronavirus Outbreak in India | elastic network interface - Google Search

Apps New Tab Search G Inbox (157) - 2019p... M Inbox (1,144) - vish... G Inbox (252) - 2019p... M Gmail YouTube Maps 29th APRIL - Googl...

Google elastic network interface

All Shopping Videos Images News More Settings Tools

About 3,23,00,000 results (0.42 seconds)

docs.aws.amazon.com › latest › UserGuide › using-eni ▾

Elastic network interfaces - Amazon Elastic Compute Cloud

An **elastic network interface** is a logical networking component in a VPC that represents a virtual network card. It can include the following attributes: A primary ...

Network interface basics · IP addresses per network ... · Detaching a network ...

People also ask

- What is an elastic network interface?
- What is AWS Eni used for?
- What is Eni in AWS VPC?
- How do I add a network interface to ec2?

Feedback

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html>

Type here to search

AWS EKS SERVICE - Google Docs | Day 2 - Google Docs | AWS EKS - Google Drive | Coronavirus Outbreak in India | Elastic network interfaces - Google Search

Apps New Tab Search G Inbox (157) - 2019p... M Inbox (1,144) - vish... G Inbox (252) - 2019p... M Gmail YouTube Maps 29th APRIL - Googl...

aws Search in this guide English Sign In to the Console

AWS Documentation update Full dark mode is now available for text and code. To switch modes, choose Preferences.

Amazon Elastic Compute Cloud X User Guide for Linux Instances

Feedback Preferences

On this page

| Instance type | Maximum network interfaces | Private IPv4 addresses per interface | IPv6 addresses per interface |
|---------------|----------------------------|--------------------------------------|------------------------------|
| a1.medium | 2 | 4 | 4 |
| a1.large | 3 | 10 | 10 |
| a1.xlarge | 4 | 15 | 15 |
| a1.2xlarge | 4 | 15 | 15 |
| a1.4xlarge | 8 | 30 | 30 |
| a1.metal | 8 | 30 | 30 |
| c1.medium | 2 | 6 | IPv6 not supported |

addresses. For more information about IPv6 in VPC, see IP Addressing in Your VPC in the *Amazon VPC User Guide*.

Network interface basics

IP addresses per network interface per instance type

Scenarios for network interfaces

Best practices for configuring network interfaces

Working with network interfaces

Type here to search

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html>

Maximum network interface * private ipv4 = pods (in which we have 1 ip reserved for the os)

```
fargate - Notepad
File Edit Format View Help
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: f-lwcluster
  region: ap-south-1

fargateProfiles:
- name: fargate-default
```

coredns are those ports which r running in the slave part

```
fargate - Notepad
File Edit Format View Help
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: f-lwcluster
  region: ap-south-1

fargateProfiles:
- name: fargate-default
  selectors:
    - namespace: kube-system
    - namespace: default
```

if a service req to communicate with the other service we use the roles

- Cloud formation
- cluster->client->eksctl

In kubernetes master check the login and password

For sso u can create a identity provider

The screenshot displays two stacked views of the AWS IAM Management Console.

Top View (Main Dashboard):

- Left Sidebar:** Shows the navigation menu under "Identity and Access Management (IAM)".
 - Dashboard** (selected)
 - Access management**
 - Groups
 - Users
 - Roles
 - Policies
 - Identity providers**
 - Account settings
 - Access reports**
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
 - Credential report

Middle Section: Welcome to Identity and Access Management.

- IAM users sign-in link:
<https://410914255776.signin.aws.amazon.com/console>
- IAM Resources**
 - Users: 1 Roles: 8
 - Groups: 0 Identity Providers: 0
 - Customer Managed Policies: 1
- Security Status**
 - 2 out of 5 complete.
 - Checkmarks: Delete your root access keys, Create individual IAM users.
 - Warnings: Activate MFA on your root account, Use groups to assign permissions, Apply an IAM password policy.
- Additional Information**
 - IAM best practices
 - IAM documentation
 - Web Identity Federation Playground
 - Policy Simulator
 - Videos, IAM release history and additional resources

Bottom Bar: © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Bottom View (Provider Management):

- Left Sidebar:** Same as the top view.
- Middle Section:** Shows the "Identity providers" section.
 - Create Provider** and **Delete Providers** buttons.
 - A search bar labeled "Search".
 - A table with columns: **Provider Name**, **Type**, and **Creation Time**. A note says "Showing 0 results".
 - No records found.
- Bottom Bar:** © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the AWS IAM Management Console. On the left, there's a sidebar with navigation links like Dashboard, Access management, Policies, and Access reports. The main area displays a table titled "Showing 8 results" with columns for Role name, Trusted entities, and Last activity. The table lists several AWS service roles, such as AWSCodePipelineServiceRole-ap-south-1..., AWSserviceRoleForAmazonEKS, and AWSserviceRoleForAmazonElasticFileSys... The last activity column shows various timeframes like "21 days", "Today", and "None".

<https://kubernetes.io/docs/concepts/cluster-administration/cloud-providers/>

The screenshot shows the Kubernetes documentation page for cloud providers. The left sidebar includes links for Home, Getting started, Concepts, Overview, Cluster, Architecture, Containers, Workloads, Services, Load Balancing, and Networking, Storage, Configuration, and Security. The main content area discusses annotations supported on AWS ELBs, listing items like `service.beta.kubernetes.io/aws-load-balancer-access-log-emit-interval`, `service.beta.kubernetes.io/aws-load-balancer-access-log-enabled`, and `service.beta.kubernetes.io/aws-load-balancer-access-log-s3-bucket-name`. To the right, a vertical list of cloud provider names is shown with their corresponding node names: kubeadm (AWS), Node Name (Load Balancers), Azure, Node Name (CloudStack), Node Name (GCE), Node Name (HUAWEI CLOUD), Node Name (OpenStack), and Node Name (Services).

Google Search results for "kubernetes annotations elb":

About 26,000 results (0.40 seconds)

kubernetes.io › docs › concepts › cluster-administration › Cloud Providers | Kubernetes

Jun 22, 2020 - Different settings can be applied to a load balancer service in AWS using **annotations**. The following describes the **annotations** supported on AWS ELBs:

- service.beta.kubernetes.io/aws-load-balancer-access-log-emit-interval : Used to specify access log emit interval.
- kubeadm · AWS · OpenStack · IBM Cloud Kubernetes ...

People also search for:

- aws-cloud-controller-manager aws-load-balancer-eip-allocations
- kubernetes/cloud-provider github elb logs kubernetes
- kubernetes nlb aws-load-balancer-connection-idle-timeout

kubernetes.io › docs › concepts › services-networking › Service | Kubernetes

May 30, 2020 - There are several **annotations** to manage access logs for **ELB** Services on

Github Gist: AWS ELB-related annotations

```

• service.beta.kubernetes.io/aws-load-balancer-connection-idle-timeout (in seconds, default 60)
• service.beta.kubernetes.io/aws-load-balancer-cross-zone-load-balancing-enabled (true|false)
• service.beta.kubernetes.io/aws-load-balancer-extra-security-groups (comma-separated list)
• service.beta.kubernetes.io/aws-load-balancer-healthcheck-threshold
• service.beta.kubernetes.io/aws-load-balancer-healthcheck-interval
• service.beta.kubernetes.io/aws-load-balancer-healthcheck-timeout
• service.beta.kubernetes.io/aws-load-balancer-healthcheck-unhealthy-threshold
• service.beta.kubernetes.io/aws-load-balancer-internal (true|false)
• service.beta.kubernetes.io/aws-load-balancer-proxy-protocol: '*'
• service.beta.kubernetes.io/aws-load-balancer-ssl-cert (IAM or ACM ARN)
• service.beta.kubernetes.io/aws-load-balancer-ssl-negotiation-policy
• service.beta.kubernetes.io/aws-load-balancer-ssl-ports (default '*')
• service.beta.kubernetes.io/aws-load-balancer-type: nlb

```

KIVagant commented on May 18, 2017 • edited

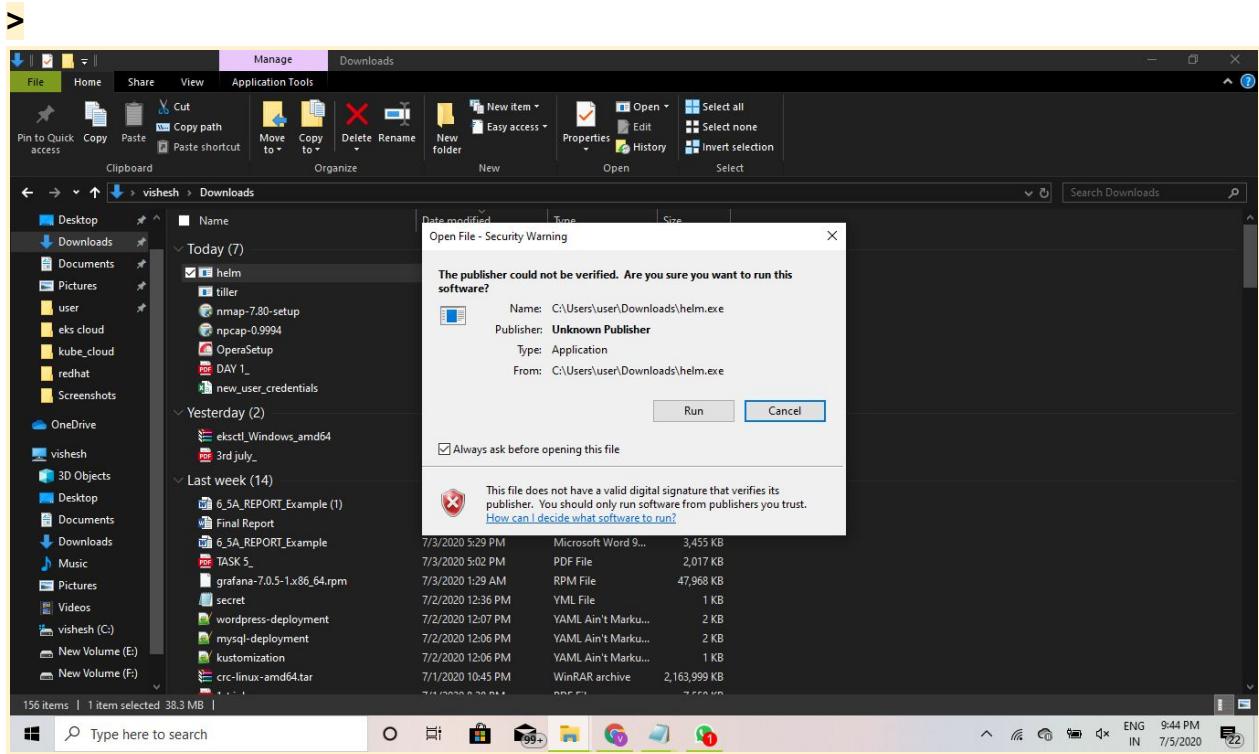
Thank you, @mgoodness

BTW, actual variables are here:

<https://github.com/kubernetes/kubernetes/blob/master/pkg/cloudprovider/providers/aws/aws.go#L72>

<https://gist.github.com/mgoodness/1a2926f3b02d8e8149c224d25cc57dc1>

HELM:



> all the major application create a package ...when we run the package they create the entire application
> helm is the package(chart) manager

Screenshot of a Google search results page for "helm charts". The search bar shows the query "helm charts". The results page indicates about 97,70,000 results found in 0.34 seconds. The top result is a blog post titled "Helm Charts" from Coveros, dated Aug 6, 2019. The post describes Helm as a Kubernetes package and operations manager, mentioning that a Helm chart typically contains a Deployment and a Service, but can also include an Ingress, Persistent Volume Claims, or other Kubernetes objects. A screenshot of the blog post content is shown.

Helm Charts

Helm is a Kubernetes package and operations manager. ... A **Helm chart** will usually contain at least a Deployment and a Service, but it can also contain an Ingress, Persistent Volume Claims, or any other Kubernetes object. **Helm charts** are used to deploy an application, or one component of a larger application. Aug 6, 2019

www.coveros.com/what-is-a-helm-chart-a-beginners-...

[What Is A Helm Chart? – A Beginner's Guide - Coveros](#)

Screenshot of a Microsoft Edge browser window showing the Helm Hub interface. The address bar shows the URL "https://www.google.com/search?q=helm+charts&sxsrf=ALeKk014yhHWlQow3Db...". The main content area displays the "Discover & launch great Kubernetes-ready apps" heading, a search bar labeled "Search charts...", and a count of "1356 charts ready to deploy". Below this, four chart cards are visible:

- gabibbo97/389ds** (fedora-32) by **gabibbo97**
- fypipe/Fypipe** by **fypipe**
- ckotzbauer/access-manager** (0.1.0) by **ckotzbauer**
- mogaal/adminer** (4.7.3) by **mogaal**

Helm Hub: Discover & launch great Kubernetes applications

hub.helm.sh

Apps New Tab Search G Inbox (157) - 2019p... M Inbox (1,144) - vish... M Inbox (252) - 2019p... M Gmail YouTube Maps 29th APRIL - Googl...

cetic/adminer 4.7.6 stable/aerospike v4.5.0.5 aerospike/aerospike 5.0.0.4 aerospike/aerospike-enterprise 5.0.0.4

aerospike/aerospike-rest-client 1.16.0 buildkite/buildkite-agent 3.17.0 jitterbit/jitterbit 10 choerodon/agile-service 0.22.1

agones/agones 1.7.0-rc franzinc/agraphmmr 7.0.0 stable/airflow 1.10.10 bitnami/airflow 1.10.10

helm helm charts?q=jenkins

Search charts... Charts About

Charts

Repository

- all
- stable
- incubator
- jfrog
- kremers
- linkerd2
- linkerd2-edge
- rimusz
- buildkite
- cloudposse
- cockroachdb
- keel
- webhookrelay
- appscode

jenkins

cloudbees/cloudbees-core 2.235.1.2 cloudbees/cloudbees-jenkins-distribution 2.222.2.1 choerodon/jenkins 2.60.3-alpine stable/jenkins lts

This chart will do the following:

- 1 x Jenkins Master with port 8080 exposed on an external LoadBalancer
- All using Kubernetes Deployments

Installing the Chart

To install the chart with the release name `my-release`:

```
$ helm install my-release stable/jenkins
```

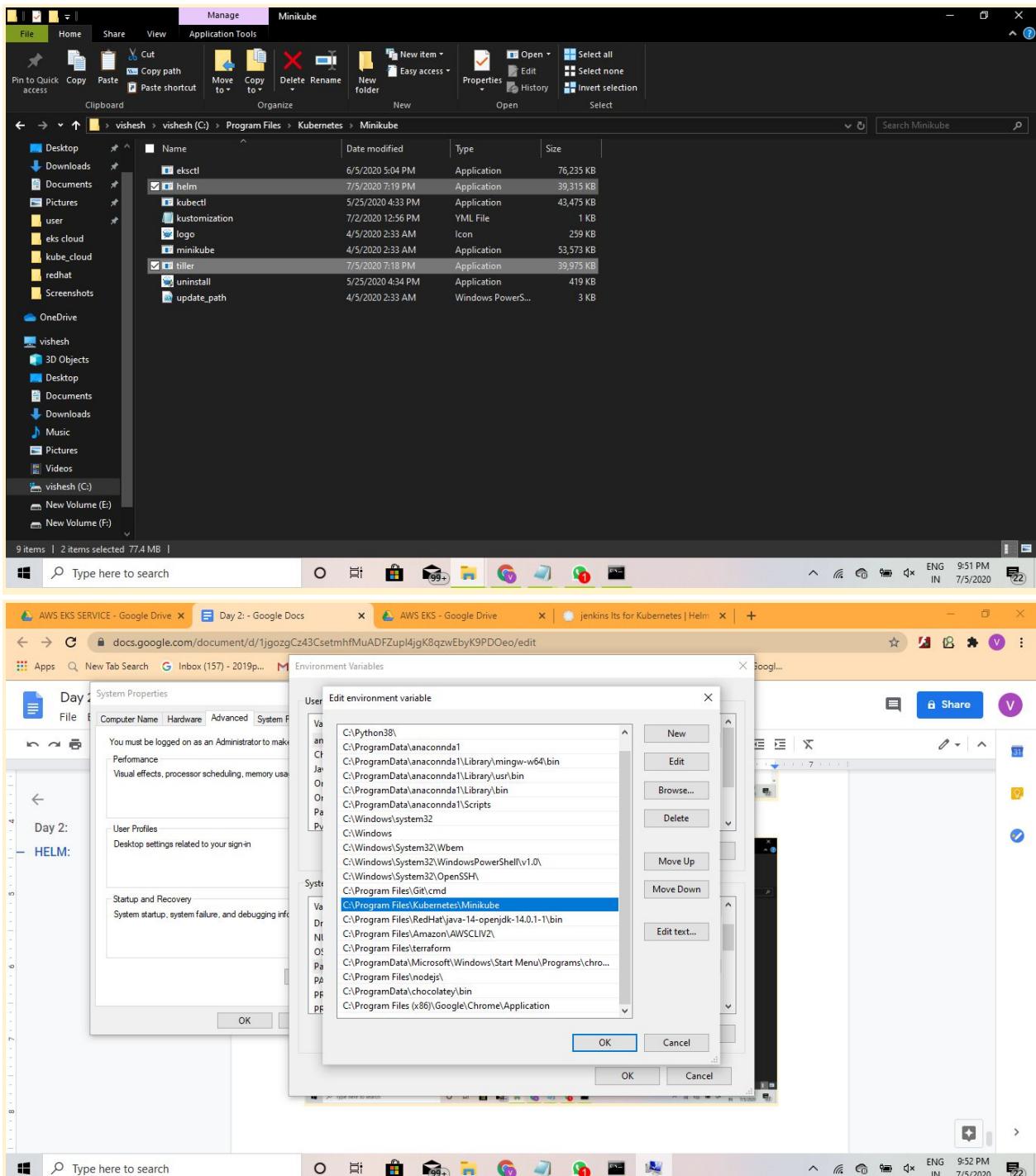
Upgrading an existing Release to a new major version

Chart Versions

2.1.2 - Jul 5, 2020
2.1.1 - Jul 3, 2020
2.1.0 - Jul 2, 2020
2.0.1 - Jun 16, 2020
2.0.0 - Jun 6, 2020
[show all](#)

client program: helm

Server program: thriller



```
C:\Users\user\Desktop\eks cloud>aws eks list-clusters
{
  "clusters": []
}
```

The screenshot displays three separate browser tabs within a single window, all sharing the same address bar URL: `ap-south-1.console.aws.amazon.com`.

Top Tab: Shows the "Elastic IP addresses | EC2 Manager" page. It lists one entry: "13.234.244.85" with allocation ID "eipalloc-07feec4b93d65bf77". There is an "Allocate Elastic IP address" button.

Middle Tab: Shows the "Amazon Container Services" page under the "Amazon EKS" section. It displays a cluster named "vishcluster" with Kubernetes version 1.16 and status "Creating". The "Compute" tab is selected in the navigation bar. Below it, the "Node Groups (0)" section shows "No Managed Node Groups".

Bottom Tab: Shows the "Dashboard | EC2 Manager" page. It displays a summary of instance types and AMIs. The "Compute" tab is selected in the navigation bar.

| Name | Security group ID | Security group name | VPC ID | Description |
|--|----------------------|--|---|----------------------------|
| eksctl-vishcluster-cl... <input checked="" type="checkbox"/> | sg-00f6355b2b691ef30 | eksctl-vishcluster-clust... Communication | vpc-08435fe4542f1c1f2 Edit | EKS created security group |
| eks-cluster-sg-vishclu... <input checked="" type="checkbox"/> | sg-028f8f6e0788db307 | eks-cluster-sg-vishclus... Communication | vpc-08435fe4542f1c1f2 Edit | EKS created security group |
| <input type="checkbox"/> - | sg-0a5b180560f799dc0 | default | vpc-08435fe4542f1c1f2 Edit | default VPC security group |
| eksctl-vishcluster-cl... <input checked="" type="checkbox"/> | sg-0c01d9075e9fc13f6 | eksctl-vishcluster-clust... Communication | vpc-08435fe4542f1c1f2 Edit | EKS created security group |
| <input type="checkbox"/> - | sg-850e7ee7 | default | vpc-15f8e57d Edit | default VPC security group |

C:\Users\user\Desktop\eks cloud>eksctl get cluster

| NAME | REGION |
|-------------|------------|
| vishcluster | ap-south-1 |

C:\Users\user\Desktop\eks cloud>eksctl get cluster -h

Get cluster(s)

Usage: eksctl get cluster [flags]

Aliases: cluster, clusters

General flags:

| | |
|---------------------|---|
| -n, --name string | EKS cluster name |
| -A, --all-regions | List clusters across all supported regions |
| -r, --region string | AWS region |
| --chunk-size int | return large lists in chunks rather than all at once, pass 0 to disable (default 100) |
| -o, --output string | specifies the output format (valid option: table, json, yaml) (default "table") |
| --timeout duration | maximum waiting time for any long-running operation (default 25m0s) |

AWS client flags:

-p, --profile string AWS credentials profile to use (overrides the AWS_PROFILE environment variable)

Common flags:

-C, --color string toggle colorized logs (valid options: true, false, fabulous) (default "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 get cluster [command] --help' for more information about a command.

```
C:\Users\user\Desktop\eks cloud>eksctl get -h  
Get resource(s)
```

Usage: eksctl get [flags]

Commands:

eksctl get cluster Get cluster(s)

eksctl get nodegroup Get nodegroup(s)

eksctl get iamserviceaccount Get iamserviceaccount(s)

eksctl get iamidentitymapping Get IAM identity mapping(s)

eksctl get labels Get nodegroup labels

eksctl get fargateprofile Get Fargate profile(s)

Common flags:

-C, --color string toggle colorized logs (valid options: true, false, fabulous) (default "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 get [command] --help' for more information about a command.

```
C:\Users\user\Desktop\eks cloud>eksctl get fargateprofile --cluster vishcluster
```

Error: failed to get Fargate profile(s) for cluster "vishcluster":

AccessDeniedException:

status code: 403, request id: b651ce68-e1fc-42fe-b2ad-a42cb09d3fbf

```
C:\Users\user\Desktop\eks cloud>eksctl get nodegroup --cluster vishcluster
```

| CLUSTER | NODEGROUP | CREATED | MIN SIZE | MAX SIZE |
|------------------|-----------------------|----------------------|----------|----------|
| DESIRED CAPACITY | INSTANCE TYPE | IMAGE ID | | |
| vishcluster | ng1 | 2020-07-06T19:47:29Z | 2 | 2 |
| t2.micro | ami-073969767527f7306 | | | |
| vishcluster | ng2 | 2020-07-06T19:47:28Z | 1 | 1 |
| t2.small | ami-073969767527f7306 | | | |

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main area displays a table of instances:

| NAME | Instance ID | Instance Type | Availability Zone | Instance State | Status |
|-------------------|---------------------|---------------|-------------------|----------------|--------|
| vishcluster-ns... | i-0ff5648081078deac | t2.small | ap-south-1b | running | 2/2 |
| vishcluster-ns... | i-065a4e6029f5e3d6f | t2.micro | ap-south-1a | running | 2/2 |
| vishcluster-ns... | i-06e4e75a920f5e9a5 | t2.micro | ap-south-1b | running | 2/2 |

```
C:\Users\user\Desktop\eks cloud>eksctl scale nodegroup --cluster vishcluster  
--name gp1 -- nodes=3
```

>> if we have set the limit

```
C:\Users\user\Desktop\eks cloud>eksctl scale nodegroup --cluster vishcluster  
--name gp2 -- nodes=3 --nodes-max=5
```

```
[!] scaling nodegroup stack "eksctl-lwcluster-nodegroup-ng2" in cluster eksctl-lwcluster-cluster
[!] the desired nodes 3 is greater than current nodes-max/maxSize 1
Error: failed to scale nodegroup for cluster "lwcluster", error the desired nodes 3 is greater than current nodes-max/maxSize 1

C:\Users\Vimal Daga\Desktop\eks_class_code>eksctl get nodegroup --cluster lwcluster
CLUSTER      NODEGROUP      CREATED          MIN SIZE    MAX SIZE    DESIRED CAPACITY
INSTANCE TYPE IMAGE ID
lwcluster    ng-mixed     2020-07-05T13:27:27Z   2           5           0
lwcluster    ng1          2020-07-05T13:27:28Z   2           2           2
lwcluster    ng2          2020-07-05T13:27:28Z   1           1           1
lwcluster    t2.small     ami-073969767527f7306
```

```
[?] scaling nodegroup stack "eksctl-lwcluster-nodegroup-ng2" in cluster eksctl-lwcluster-cluster
[?] scaling nodegroup, desired capacity from 1 to 3, max size from 1 to 5
```

C:\Users\user>aws eks update-kubeconfig --name vishcluster

Updated context arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster in
C:\Users\user\.kube\config

The screenshot shows the AWS Management Console with the EC2 Instances page open. The left sidebar shows navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main content area displays a table of instances with columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status. The table shows three instances with the following details:

| Name | Instance ID | Type | Zone | Status |
|------------------|---------------------|----------|-------------|---------|
| vishcluster-n... | i-0ff5648081078deac | t2.small | ap-south-1b | running |
| vishcluster-n... | i-065a4e6029f5e3d6f | t2.micro | ap-south-1a | running |
| vishcluster-n... | i-06e4e75a920f5e9a5 | t2.micro | ap-south-1b | running |

Below the table, there are sections for Secondary private IPs (192.168.10.233, 192.168.2.149, 192.168.24.135), VPC ID (vpc-08435fe4542f1c1f2), Subnet ID (subnet-02c316dba4d2fd30c), Network interfaces (eth0), IAM role (eksctl-vishcluster-nodegroup-ng2), Scheduled events (No scheduled events), AMI ID (amazon-eks-node-1.16-v20200618 (ami-073969767527f7306)), Platform details (Linux/UNIX), Usage operation (RunInstances), Source/dest. check (True), and a note about ClusterSharedNodeSecurityGroup F110ZVIIW8RK8.

C:\Users\user>cd C:\Users\user\Desktop\eks cloud

C:\Users\user\Desktop\eks cloud>ssh -i mykey1111.pem -l ec2-user 13.235.48.193

The authenticity of host '13.235.48.193 (13.235.48.193)' can't be established.

ECDSA key fingerprint is

SHA256:VxFWM7bSyakJRSLS4UfB+18H8Yn3+qqobfJcWADCdic.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '13.235.48.193' (ECDSA) to the list of known hosts.

Last login: Thu Jun 18 01:20:30 2020 from 205.251.233.50

Last login: Thu Jun 18 01:20:30 2020 from 205.251.233.50

_| _|_)

_| (/ Amazon Linux 2 AMI

__|__|_|

<https://aws.amazon.com/amazon-linux-2/>

4 package(s) needed for security, out of 10 available

Run "sudo yum update" to apply all updates.

```
[ec2-user@ip-192-168-21-219 ~]$ sudo su
[root@ip-192-168-21-219 ec2-user]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 192.168.21.219 netmask 255.255.224.0 broadcast 192.168.31.255
        inet6 fe80::80a:d3ff:febe:7256 prefixlen 64 scopeid 0x20<link>
            ether 0a:0a:d3:be:72:56 txqueuelen 1000 (Ethernet)
            RX packets 147359 bytes 205412075 (195.8 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 12503 bytes 1376946 (1.3 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 1795 bytes 129988 (126.9 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1795 bytes 129988 (126.9 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
[root@ip-192-168-21-219 ec2-user]# free -m
              total        used      free    shared  buff/cache   available
Mem:       1991         264       696          0       1030       1587
Swap:          0          0          0
```

```
[root@ip-192-168-21-219 ec2-user]# systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor
  preset: disabled)
  Active: active (running) since Mon 2020-07-06 19:51:29 UTC; 12min ago
    Docs: https://docs.docker.com
  Main PID: 4010 (dockerd)
    Tasks: 12
   Memory: 598.1M
     CGroup: /system.slice/docker.service
             └─4010 /usr/bin/dockerd -H fd://
--containerd=/run/containerd/containerd.sock
```

```
Jul 06 19:51:28 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:
time="2020-07-06T19:51:28.936285645Z" ...pc
```

```
Jul 06 19:51:28 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:51:28.936299133Z" ...pc  
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:51:29.032658106Z" ...."  
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:51:29.348565863Z" ...."  
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:51:29.430319754Z" ...ce  
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:51:29.430882379Z" ...n"  
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:51:29.516533335Z" ...k"  
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal systemd[1]:  
Started Docker Application Container Engine.  
Jul 06 19:52:38 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:52:38.056924889Z" ...e"  
Jul 06 19:52:38 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]:  
time="2020-07-06T19:52:38.101362855Z" ...e"  
Hint: Some lines were ellipsized, use -l to show in full.  
[root@ip-192-168-21-219 ec2-user]# ps aux | grep kubelet  
root 4581 0.9 4.3 856268 88708 ? Ssl 19:51 0:06 /usr/bin/kubelet  
--node-ip=192.168.21.219  
--node-labels=alpha.eksctl.io/cluster-name=vishcluster, alpha.eksctl.io/nodegroup  
-name=ng2, alpha.eksctl.io/instance-id=i-0ff5648081078deac --max-pods=11  
--register-node=true --register-with-taints= --cloud-provider=aws  
--container-runtime=docker --network-plugin=cni --cni-bin-dir=/opt/cni/bin  
--cni-conf-dir=/etc/cni/net.d  
--pod-infra-container-image=602401143452.dkr.ecr.ap-south-1.amazonaws.com/ek  
s/pause-amd64:3.1 --kubeconfig=/etc/eksctl/kubeconfig.yaml  
--config=/etc/eksctl/kubelet.yaml  
root 13473 0.0 0.0 119420 864 pts/0 S+ 20:04 0:00 grep --color=auto  
kubelet
```

A screenshot of a Windows desktop environment. In the center is a terminal window titled 'kubelet' showing log output from a Docker daemon. The logs include entries like:

```

Jul 06 19:51:28 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:28.936285645Z" ...
Jul 06 19:51:28 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:28.936299132Z" ...
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:29.032658106Z" ...
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:29.348565863Z" ...
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:29.430319754Z" ...
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:29.430882379Z" ...
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:51:29.516533335Z" ...
Jul 06 19:51:29 ip-192-168-21-219.ap-south-1.compute.internal systemd[1]: Started Docker Application Container Engine.
Jul 06 19:52:38 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:52:38.056924889Z" ...
Jul 06 19:52:38 ip-192-168-21-219.ap-south-1.compute.internal dockerd[4010]: time="2020-07-06T19:52:38.101362855Z" ...

```

Below the terminal, the command line shows:

```
[root@ip-192-168-21-219 ec2-user]# exit
```

```
exit
[ec2-user@ip-192-168-21-219 ~]$ exit
logout
Connection to 13.235.48.193 closed.
```

```
C:\Users\user\Desktop\eks cloud>ssh -i mykey1111.pem -l ec2-user 13.126.18.68
The authenticity of host '13.126.18.68 (13.126.18.68)' can't be established.
ECDSA key fingerprint is
SHA256:UNBo0dN/vJp5yAFXqMg9Rm/rhA6cw9u8lMBnvnGrB48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.126.18.68' (ECDSA) to the list of known hosts.
Last login: Thu Jun 18 01:20:30 2020 from 205.251.233.50
Last login: Thu Jun 18 01:20:30 2020 from 205.251.233.50
```

```
_ _| _|_ )
_| ( / Amazon Linux 2 AMI
__| \__|_|
```

<https://aws.amazon.com/amazon-linux-2/>
4 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.

```
[ec2-user@ip-192-168-42-210 ~]$ sudo su
[root@ip-192-168-42-210 ec2-user]# ps aux | grep kubelet
root 4556 1.1 8.9 872660 90588 ? Ssl 19:51 0:09 /usr/bin/kubelet
--node-ip=192.168.42.210
--node-labels=alpha.eksctl.io/cluster-name=vishcluster,alpha.eksctl.io/nodegroup
-name=ng1,alpha.eksctl.io/instance-id=i-065a4e6029f5e3d6f --max-pods=4
--register-node=true --register-with-taints= --cloud-provider=aws
--container-runtime=docker --network-plugin=cni --cni-bin-dir=/opt/cni/bin
--cni-conf-dir=/etc/cni/net.d
--pod-infra-container-image=602401143452.dkr.ecr.ap-south-1.amazonaws.com/ek
s/pause-amd64:3.1 --kubeconfig=/etc/eksctl/kubeconfig.yaml
--config=/etc/eksctl/kubelet.yaml
root 18230 0.0 0.0 119420 908 pts/0 S+ 20:05 0:00 grep --color=auto
kubelet
[root@ip-192-168-42-210 ec2-user]# ifconfig
eni28274e53c7c: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
inet6 fe80::d420:f0ff:fe78:477c prefixlen 64 scopeid 0x20<link>
ether d6:20:f0:78:47:7c txqueuelen 0 (Ethernet)
RX packets 1749 bytes 144969 (141.5 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1751 bytes 566433 (553.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eni500ff986a34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
inet6 fe80::5c6e:3eff:fe0d:2abd prefixlen 64 scopeid 0x20<link>
ether 5e:6e:3e:0d:2a:bd txqueuelen 0 (Ethernet)
RX packets 1713 bytes 142033 (138.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1713 bytes 553760 (540.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
inet 192.168.42.210 netmask 255.255.224.0 broadcast 192.168.63.255
inet6 fe80::fa:4aff:fed8:b126 prefixlen 64 scopeid 0x20<link>
ether 02:fa:4a:d8:b1:26 txqueuelen 1000 (Ethernet)
RX packets 159173 bytes 220367938 (210.1 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 15636 bytes 1574639 (1.5 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 192.168.63.242 netmask 255.255.224.0 broadcast 192.168.63.255
    inet6 fe80::d9:84ff:fe4e:19c4 prefixlen 64 scopeid 0x20<link>
        ether 02:d9:84:4e:19:c4 txqueuelen 1000 (Ethernet)
        RX packets 982 bytes 471330 (460.2 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 952 bytes 69480 (67.8 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 3267 bytes 242184 (236.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 3267 bytes 242184 (236.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
[root@ip-192-168-42-210 ec2-user]# exit
exit
[ec2-user@ip-192-168-42-210 ~]$ exit
logout
Connection to 13.126.18.68 closed.
C:\Users\user\Desktop\eks cloud>ssh -i mykey1111.pem -l ec2-user 13.233.251.16
The authenticity of host '13.233.251.16 (13.233.251.16)' can't be established.
ECDSA key fingerprint is
SHA256:FA8rOulqleGUz4PAN6kTpQFP36GZgE4f5FvB6hGc160.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.233.251.16' (ECDSA) to the list of known hosts.
Last login: Thu Jun 18 01:20:30 2020 from 205.251.233.50
```

```
_) )
_ ( / Amazon Linux 2 AMI
___\_\_|__|
```

```
https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.
```

```
[ec2-user@ip-192-168-12-162 ~]$ sudo su
[root@ip-192-168-12-162 ec2-user]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 192.168.12.162 netmask 255.255.224.0 broadcast 192.168.31.255
        inet6 fe80::89c:cbff:fe24:7c26 prefixlen 64 scopeid 0x20<link>
            ether 0a:9c:cb:24:7c:26 txqueuelen 1000 (Ethernet)
                RX packets 147936 bytes 205590153 (196.0 MiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 14604 bytes 1491627 (1.4 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
            RX packets 2620 bytes 189718 (185.2 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2620 bytes 189718 (185.2 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
[root@ip-192-168-12-162 ec2-user]# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 192.168.12.162 netmask 255.255.224.0 broadcast 192.168.31.255
        inet6 fe80::89c:cbff:fe24:7c26 prefixlen 64 scopeid 0x20<link>
            ether 0a:9c:cb:24:7c:26 txqueuelen 1000 (Ethernet)
                RX packets 148240 bytes 205675963 (196.1 MiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 14894 bytes 1528003 (1.4 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
            RX packets 2725 bytes 197320 (192.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2725 bytes 197320 (192.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
[root@ip-192-168-12-162 ec2-user]# exit
```

exit

[ec2-user@ip-192-168-12-162 ~]\$ exit

logout

Connection to 13.233.251.16 closed.

A screenshot of a Windows desktop environment. On the left, there is a terminal window titled "Administrator Command Prompt" showing a command-line session with EKS commands like `eksctl` and `kubectl`. On the right, there is a "YAMLInt - The YAM..." application window displaying a YAML configuration file. The taskbar at the bottom shows various icons for common Windows applications like File Explorer, Edge, and File History. The system tray indicates the date and time as 7/7/2020, 1:38 AM, with language set to ENG IN.

```
C:\Users\user\Desktop\eks cloud>eks cloud>eksctl get fargateprofile --cluster vishcluster
Error: failed to get Fargate profile(s) for cluster "vishcluster": AccessDeniedException:
status code: 403, request id: b651ce68-e1fc-42fe-b2ad-a42cb89d3fbf

C:\Users\user\Desktop\eks cloud>eksctl get nodegroup --cluster vishcluster
CLUSTER      NODEGROUP      CREATED          MIN SIZE   MAX SIZE   DESIRED CAPACITY   INSTANCE
vishcluster  ng1           2020-07-06T19:47:29Z  2           2           2           t2.micro
vishcluster  ng1           ami-073969767527f7306
vishcluster  ng2           2020-07-06T19:47:28Z  1           1           1           t2.small
vishcluster  ng2           ami-073969767527f7306

C:\Users\user\Desktop\eks cloud>kubectl get pods
No resources found in default namespace.

C:\Users\user\Desktop\eks cloud>kubectl get nodes
NAME                  STATUS  ROLES   AGE   VERSION
ip-192-168-12-162.ap-south-1.compute.internal  Ready  <none>  16m  v1.16.8-eks-fd1ea7
ip-192-168-21-219.ap-south-1.compute.internal  Ready  <none>  15m  v1.16.8-eks-fd1ea7
ip-192-168-42-210.ap-south-1.compute.internal  Ready  <none>  16m  v1.16.8-eks-fd1ea7

C:\Users\user\Desktop\eks cloud>
```

```
exit
[ec2-user@ip-192-168-12-162 ~]$ exit
logout
Connection to 13.233.251.16 closed.
```

C:\Users\user\Desktop\eks cloud>kubectl get pods

No resources found in default namespace.

C:\Users\user\Desktop\eks cloud>kubectl get nodes

| NAME | STATUS | ROLES | AGE | VERSION |
|---|--------|--------|-----|--------------------|
| ip-192-168-12-162.ap-south-1.compute.internal | Ready | <none> | 16m | v1.16.8-eks-fd1ea7 |
| ip-192-168-21-219.ap-south-1.compute.internal | Ready | <none> | 15m | v1.16.8-eks-fd1ea7 |
| ip-192-168-42-210.ap-south-1.compute.internal | Ready | <none> | 16m | v1.16.8-eks-fd1ea7 |

Screenshot of the AWS VPC console showing the list of VPCs and a detailed view of a specific VPC.

VPC Dashboard

| Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR | DHCP options set | Main Route table |
|------------------|-----------------------|-----------|----------------|-----------|------------------|------------------|
| eksctl-vishcl... | vpc-08435fe4542f1c1f2 | available | 192.168.0.0/16 | - | dopt-57758d3c | rtb-0af6a5fd2236 |
| | vpc-15f8e57d | available | 172.31.0.0/16 | - | dopt-57758d3c | rtb-2c6dbe47 |

Detailed View of VPC: vpc-08435fe4542f1c1f2

| | | | |
|--|--|-----------|------|
| Description | CIDR Blocks | Flow Logs | Tags |
| VPC ID: vpc-08435fe4542f1c1f2 State: available IPv4 CIDR: 192.168.0.0/16 IPv6 Pool: - Network ACL: acl-0c6a7913c13650f94 DHCP options set: dopt-57758d3c Owner: 410914255776 | Tenancy: default Default VPC: No IPv6 CIDR: - DNS resolution: Enabled DNS hostnames: Enabled Route table: rtb-0af6a5fd2236b1208 | | |

Feedback English (US)

Type here to search

© 2008 - 2020, Amazon Internet Services Private Ltd, or its affiliates. All rights reserved. Privacy Policy Terms of Use

ENG 1:40 AM IN 7/7/2020

Screenshot of the AWS Subnets console showing the list of subnets and a detailed view of a specific subnet.

Subnets

| Name | Subnet ID | State | VPC | IPv4 CIDR | Available IPv4 | IPv6 CIDR | Available |
|------------------|--------------------------|-----------|--|------------------|----------------|-----------|-----------|
| eksctl-vishcl... | subnet-0c17f65ac501d0a7d | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster-cluster/VPC | 192.168.96.0/... | 8187 | - | ap... |
| eksctl-vishcl... | subnet-0470d32104d202d5b | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster-cluster/VPC | 192.168.64.0/... | 8186 | - | ap... |
| eksctl-vishcl... | subnet-066cf2920fdd9c095 | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster-cluster/VPC | 192.168.32.0/... | 8182 | - | ap... |
| eksctl-vishcl... | subnet-04f311a3dfa5fc1a7 | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster-cluster/VPC | 192.168.160... | 8187 | - | ap... |

Detailed View of Subnet: subnet-0c17f65ac501d0a7d

| | | | | | |
|---|--|---|--|---------------------|---------|
| Description | Flow Logs | Route Table | Network ACL | Tags | Sharing |
| Subnet ID: subnet-0c17f65ac501d0a7d VPC: vpc-08435fe4542f1c1f2 eksctl-vishcluster-cluster/VPC Available IPv4 Addresses: 8187 Availability Zone: ap-south-1b (aps1-az3) Network ACL: acl-0c6a7913c13650f94 Auto-assign public IPv4 address: No Outpost ID: - | State: available IPv4 CIDR: 192.168.96.0/19 | IPv6 CIDR: - Route Table: rtb-0238f196a89fb7d68 eksctl-vishcluster-cluster/PrivateRouteTableAPPSOUTH1B | Default subnet: No Auto-assign IPv6 address: No | Owner: 410914255776 | |

Feedback English (US)

Type here to search

© 2008 - 2020, Amazon Internet Services Private Ltd, or its affiliates. All rights reserved. Privacy Policy Terms of Use

ENG 1:40 AM IN 7/7/2020

Screenshot of the AWS VPC Subnets page showing a list of subnets and their details.

Subnet List:

| Name | Subnet ID | State | VPC | IPv4 CIDR | Available IPv4 | IPv6 CIDR |
|------------------|---------------------------|-----------|---|------------------|----------------|-----------|
| eksctl-vishcl... | subnet-0c17f65ac501d0a7d | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | 192.168.96.0/... | 8187 | - |
| eksctl-vishcl... | subnet-0470d32104d202d5b | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | 192.168.64.0/... | 8186 | - |
| eksctl-vishcl... | subnet-086cf2920fdd9c095 | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | 192.168.32.0/... | 8182 | - |
| eksctl-vishcl... | subnet-04f341a3dfa5fc1a7 | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | 192.168.160.... | 8187 | - |
| eksctl-vishcl... | subnet-07fbaf58f8549f60c9 | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | 192.168.128.... | 8187 | - |
| eksctl-vishcl... | subnet-02c316dba4d2fd30c | available | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | 192.168.0.0/19 | 8180 | - |
| | subnet-d7ead0bf | available | vpc-15f8e57d | 172.31.32.0/20 | 4091 | - |
| | subnet-881daff3 | available | vpc-15f8e57d | 172.31.16.0/20 | 4091 | - |
| | subnet-2f3ee5563 | available | vpc-15f8e57d | 172.31.0.0/20 | 4091 | - |

Selected Subnet Details:

| | | | |
|-----------|---|-----------|-----------------|
| Subnet ID | subnet-0c17f65ac501d0a7d | State | available |
| VPC | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | IPv4 CIDR | 192.168.96.0/19 |

Subnet Configuration:

| | | | |
|---------------------------------|---|--------------------------|---|
| Subnet ID | subnet-0c17f65ac501d0a7d | State | available |
| VPC | vpc-08435fe4542f1c1f2 eksctl-vishcluster- | IPv4 CIDR | 192.168.96.0/19 |
| Available IPv4 Addresses | 8187 | IPv6 CIDR | - |
| Availability Zone | ap-south-1b (aps1-az3) | Route Table | rtb-0238f196a89fb7d68 eksctl-vishcluster- |
| | | | cluster/PrivateRouteTableAPSOUTH1B |
| Network ACL | acl-0c6a7913c13650f94 | Default subnet | No |
| Auto-assign public IPv4 address | No | Auto-assign IPv6 address | No |
| Outpost ID | - | Owner | 410914255776 |

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main content area displays a table of instances. The table columns include NAME, Instance ID, Instance Type, Availability Zone, Instance State, and Status. There are three rows of data:

| NAME | Instance ID | Instance Type | Availability Zone | Instance State | Status |
|------------------|---------------------|---------------|-------------------|----------------|--------|
| vishcluster-n... | i-0ff5648081078deac | t2.small | ap-south-1b | running | 2/2 |
| vishcluster-n... | i-065a4e6029f5e3d6f | t2.micro | ap-south-1a | running | 2/2 |
| vishcluster-n... | i-06e4a75a920f5e9a5 | t2.micro | ap-south-1b | running | 2/2 |

Below the table, there are details for one instance, including Private DNS (ip-192-168-42-210.ap-south-1.compute.internal), Private IPs (192.168.63.242, 192.168.42.210), Secondary private IPs (192.168.63.12, 192.168.43.57), VPC ID (vpc-08435fe4542f1c1f2 (eksctl-vishcluster-cluster/VPC)), Subnet ID (subnet-086cf2920fd9c095 (eksctl-)), Availability zone (ap-south-1a), Security groups (eksctl-vishcluster-nodegroup-ng1-SG-13DLYUVBET782, eksctl-vishcluster-cluster-ClusterSharedNodeSecurityGroup-F110ZVIW8RK8), Scheduled events (No scheduled events), AMI ID (amazon-eks-node-1.16-v20200618 (ami-073969767527f7306)), and Platform details (Linux/UNIX).

```
C:\Users\user\Desktop\eks cloud>ssh -i mykey1111.pem -l ec2-user 13.235.48.193
```

```
Last login: Mon Jul  6 20:03:03 2020 from 157.37.235.251
```

```
Last login: Mon Jul  6 20:03:03 2020 from 157.37.235.251
```

```
_ _| _ |_ )
_ | ( / Amazon Linux 2 AMI
__| \__|_|_ |
```

<https://aws.amazon.com/amazon-linux-2/>

4 package(s) needed for security, out of 10 available

Run "sudo yum update" to apply all updates.

[ec2-user@ip-192-168-21-219 ~]\$ sudo su

[root@ip-192-168-21-219 ec2-user]# docker ps

| CONTAINER ID | IMAGE | COMMAND | |
|---|----------------|---------------|-------|
| CREATED | STATUS | PORTS | NAMES |
| bd7565a6daf3 | | | |
| 602401143452.dkr.ecr.ap-south-1.amazonaws.com/amazon-k8s-cni | | "/bin/sh -c | |
| /app/ent..." | 21 minutes ago | Up 21 minutes | |
| k8s_aws-node aws-node-xp8g8_kube-system_0e043691-92f4-4fd1-a136-59de498 | | | |
| 7575f_0 | | | |
| a0d04a59d781 | | | |
| 602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/kube-proxy | | | |

```

" kube-proxy --v=2 --..." 21 minutes ago Up 21 minutes
k8s_kube-proxy_kube-proxy-rplkh_kube-system_990e88de-9176-46ee-8e7f-064a6
d120c45_0
7af095bcffe4
602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/pause-amd64:3.1
"/pause" 21 minutes ago Up 21 minutes
k8s_POD_kube-proxy-rplkh_kube-system_990e88de-9176-46ee-8e7f-064a6d120c4
5_0
162ed492c65e
602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/pause-amd64:3.1
"/pause" 21 minutes ago Up 21 minutes
k8s_POD_aws-node-xp8g8_kube-system_0e043691-92f4-4fd1-a136-59de4987575f
_0
[root@ip-192-168-21-219 ec2-user]# docker images
REPOSITORY TAG IMAGE ID
CREATED SIZE
602401143452.dkr.ecr.ap-south-1.amazonaws.com/amazon-k8s-cni v1.6.3
19115dae207d 2 weeks ago 332MB
602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/kube-proxy v1.16.8
df19d4cdbd99 3 months ago 96.7MB
602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/pause-amd64 3.1
9e462c010bf3 2 years ago 742kB
C:\Users\user\Desktop\eks cloud>kubectl get ns
NAME STATUS AGE
default Active 29m
kube-node-lease Active 29m
kube-public Active 29m
kube-system Active 29m
C:\Users\user\Desktop\eks cloud>kubectl get pods -n kube-system
NAME READY STATUS RESTARTS AGE
aws-node-hnx26 1/1 Running 0 23m
aws-node-v6vz6 1/1 Running 0 23m
aws-node-xp8g8 1/1 Running 0 23m
coredns-6856799b8d-4bb5j 1/1 Running 0 29m
coredns-6856799b8d-8hv7j 1/1 Running 0 29m
kube-proxy-98hbl 1/1 Running 0 23m
kube-proxy-n9jcr 1/1 Running 0 23m
kube-proxy-rplkh 1/1 Running 0 23m

```

A screenshot of a Windows desktop environment. In the foreground, a command prompt window titled "Select Administrator: Command Prompt" is open, showing the output of several Kubernetes commands. The commands include "eksctl get [command] --help", "kubectl get ns", and "kubectl get pods -n kube-system". The output lists various pods and their status across different nodes. In the background, there is a YAMLLint tool window titled "YAMLLint - The YAM...". The taskbar at the bottom shows other open applications like "AWS EKS", "CloudForm", and "Elastic File". A notification bar on the right indicates the user is Vishesh Garg and it's 1:42 AM Today, with a timestamp of 7/7/2020.

```
C:\Users\user\Desktop\eks cloud>eksctl get [command] --help
Common flags:
  -C, --color string    toggle colorized logs (valid options: true, false, fabulous) (default "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 get [command] --help' for more information about a command.

Error: unknown resource type "ns"

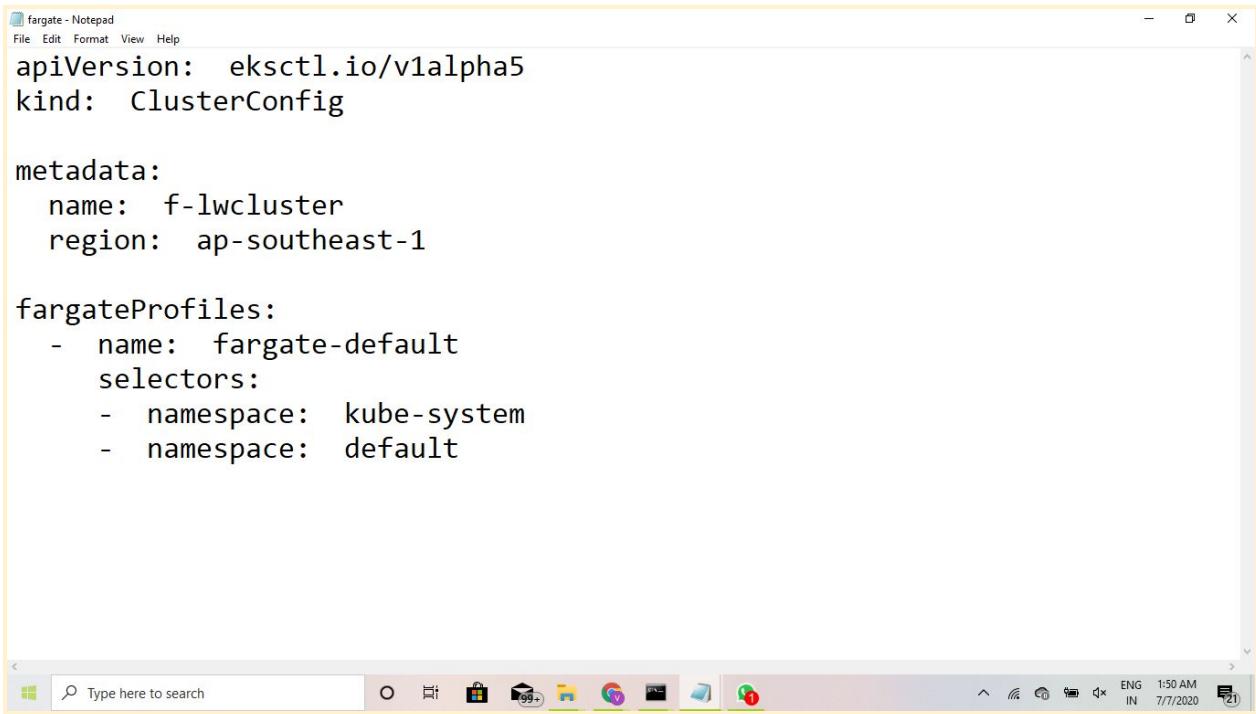
C:\Users\user\Desktop\eks cloud>kubectl get ns
NAME      STATUS   AGE
default   Active   29m
kube-node-lease  Active   29m
kube-public  Active   29m
kube-system  Active   29m

C:\Users\user\Desktop\eks cloud>kubectl get pods -n kube-system
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE
aws-node-hnx26  1/1    Running   0          24m   192.168.12.162
ip-192-168-12-162.ap-south-1.compute.internal <none>        <none>
aws-node-v6vz6  1/1    Running   0          24m   192.168.42.210
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
aws-node-xp8g8  1/1    Running   0          23m   192.168.21.219
ip-192-168-21-219.ap-south-1.compute.internal <none>        <none>
coredns-6856799b8d-4bb5j  1/1    Running   0          30m   192.168.63.12
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
coredns-6856799b8d-8hv7j  1/1    Running   0          30m   192.168.43.57
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
kube-proxy-98hbl  1/1    Running   0          24m   192.168.12.162
ip-192-168-12-162.ap-south-1.compute.internal <none>        <none>
kube-proxy-n9jcr  1/1    Running   0          24m   192.168.42.210
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
kube-proxy-rplkh  1/1    Running   0          23m   192.168.21.219
ip-192-168-21-219.ap-south-1.compute.internal <none>        <none>
```

```
C:\Users\user\Desktop\eks cloud>kubectl get pods -n kube-system -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE
NOMINATED NODE  READINESS GATES
aws-node-hnx26  1/1    Running   0          24m   192.168.12.162
ip-192-168-12-162.ap-south-1.compute.internal <none>        <none>
aws-node-v6vz6  1/1    Running   0          24m   192.168.42.210
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
aws-node-xp8g8  1/1    Running   0          23m   192.168.21.219
ip-192-168-21-219.ap-south-1.compute.internal <none>        <none>
coredns-6856799b8d-4bb5j  1/1    Running   0          30m   192.168.63.12
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
coredns-6856799b8d-8hv7j  1/1    Running   0          30m   192.168.43.57
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
kube-proxy-98hbl  1/1    Running   0          24m   192.168.12.162
ip-192-168-12-162.ap-south-1.compute.internal <none>        <none>
kube-proxy-n9jcr  1/1    Running   0          24m   192.168.42.210
ip-192-168-42-210.ap-south-1.compute.internal <none>        <none>
kube-proxy-rplkh  1/1    Running   0          23m   192.168.21.219
ip-192-168-21-219.ap-south-1.compute.internal <none>        <none>

C:\Users\user\Desktop\eks cloud>kubectl config view
apiVersion: v1
clusters:
```

```
- cluster:
  certificate-authority-data: DATA+OMITTED
  server:
    https://9D3D3D718E9E2331830AA8B1D33A22CC.yl4.ap-south-1.eks.amazonaws.com
    name: arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster
  contexts:
  - context:
    cluster: arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster
    user: arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster
    name: arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster
  current-context: arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster
  kind: Config
  preferences: {}
  users:
  - name: arn:aws:eks:ap-south-1:410914255776:cluster/vishcluster
    user:
      exec:
        apiVersion: client.authentication.k8s.io/v1alpha1
        args:
        - --region
        - ap-south-1
        - eks
        - get-token
        - --cluster-name
        - vishcluster
      command: aws
      env: null
```



```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: f-lwcluster
  region: ap-southeast-1

fargateProfiles:
  - name: fargate-default
    selectors:
      - namespace: kube-system
      - namespace: default
```

The following configuration variables are supported in the config

C:\Users\user\Desktop\eks cloud>aws configure

AWS Access Key ID [*****X6R6]:

AWS Secret Access Key [*****P6qU]:

Default region name [ap-south-1]: ap-southeast-1

Default output format [json]:

C:\Users\user\Desktop\eks cloud>eksctl create cluster -f fargate.yml

[i] eksctl version 0.21.0

[i] using region ap-southeast-1

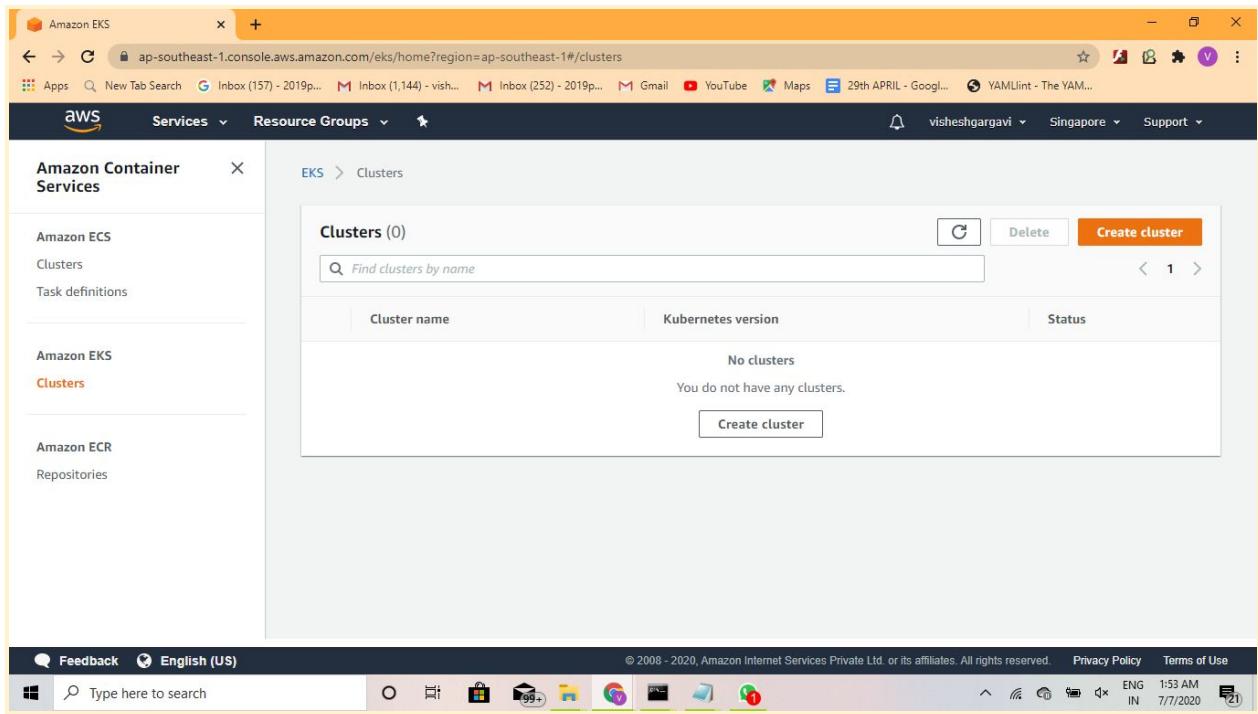
[i] setting availability zones to [ap-southeast-1a ap-southeast-1b

ap-southeast-1c]

[i] subnets for ap-southeast-1a - public:192.168.0.0/19 private:192.168.96.0/19

[i] subnets for ap-southeast-1b - public:192.168.32.0/19 private:192.168.128.0/19

[i] subnets for ap-southeast-1c - public:192.168.64.0/19 private:192.168.160.0/19



```
C:\Users\user\Desktop\eks cloud>kubectl create -k .
secret/mysql-pass-ctm2f4889c created
service/wordpress-mysql created
service/wordpress created
deployment.apps/wordpress-mysql created
deployment.apps/wordpress created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
```

```
k customization - Notepad
File Edit Format View Help
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
secretGenerator:
- name: mysql-pass
  literals:
    - password=redhat
resources:
- mysql-deployment.yml
- wordpress-deployment.yml

mysql-deployment - Notepad
File Edit Format View Help
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
```

```
mysql-deployment - Notepad
File Edit Format View Help
labels:
  app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: mysql
  strategy:
```

```
mysql-deployment - Notepad
File Edit Format View Help
  app: wordpress
  tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
      containers:
        - image: mysql:5.6
          name: mysql
          env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-pass
                  key: password
          ports:
            - containerPort: 3306
```

```
mysql-deployment - Notepad
File Edit Format View Help [Minimize]
containers:
- image: mysql:5.6
  name: mysql
  env:
  - name: MYSQL_ROOT_PASSWORD
    valueFrom:
      secretKeyRef:
        name: mysql-pass
        key: password
  ports:
  - containerPort: 3306
    name: mysql
  volumeMounts:
  - name: mysql-persistent-storage
    mountPath: /var/lib/mysql
volumes:
- name: mysql-persistent-storage
  persistentVolumeClaim:
    claimName: mysql-pv-claim

< Type here to search O Windows Taskbar ENG 1:56 AM IN 7/7/2020 (21)
wordpress-deployment - Notepad
File Edit Format View Help [Minimize]
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
  - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
```

```
wordpress-deployment - Notepad
File Edit Format View Help
labels:
  app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: frontend
  strategy:
    type: Recreate
```

```
wordpress-deployment - Notepad
File Edit Format View Help
  app: wordpress
  tier: frontend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: frontend
    spec:
      containers:
        - image: wordpress:4.8-apache
          name: wordpress
          env:
            - name: WORDPRESS_DB_HOST
              value: wordpress-mysql
            - name: WORDPRESS_DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-pass
                  key: password
      ports:
```

```
wordpress-deployment - Notepad
File Edit Format View Help
- image: wordpress:4.8-apache
  name: wordpress
  env:
    - name: WORDPRESS_DB_HOST
      value: wordpress-mysql
    - name: WORDPRESS_DB_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mysql-pass
          key: password
  ports:
    - containerPort: 80
      name: wordpress
  volumeMounts:
    - name: wordpress-persistent-storage
      mountPath: /var/www/html
  volumes:
    - name: wordpress-persistent-storage
      persistentVolumeClaim:
        claimName: wp-pv-claim
```

The screenshot shows the AWS EKS console interface. On the left, there's a sidebar with 'Amazon Container Services' and 'Amazon EKS' sections. Under 'Amazon EKS', 'Clusters' is selected. The main area is titled 'Clusters (1)' and shows a table with one row. The table has columns for 'Cluster name', 'Kubernetes version', and 'Status'. The data in the table is:

| Cluster name | Kubernetes version | Status |
|--------------|--------------------|----------|
| f-lwcluster | 1.16 | Creating |

```
C:\Users\user\Desktop\eks cloud>kubectl get sc
NAME      PROVISIONER      AGE
gp2 (default)  kubernetes.io/aws-ebs  41m
```

```
C:\Users\user\Desktop\eks cloud>kubectl get pvc
```

| NAME | STATUS | VOLUME | CAPACITY | ACCESS |
|----------------|--------------|--|----------|--------|
| MODES | STORAGECLASS | AGE | | |
| mysql-pv-claim | Bound | pvc-c433420b-2311-4809-aae4-d11bf0aa2814 | 1Gi | |
| RWO | gp2 | 2m33s | | |
| wp-pv-claim | Bound | pvc-e1dca128-12ca-47b4-8ba5-ab8df50e7128 | 1Gi | |
| RWO | gp2 | 2m32s | | |

C:\Users\user\Desktop\eks cloud>kubectl get pv

| NAME | CAPACITY | ACCESS | MODES | RECLAIM POLICY |
|--------|--|--------------|--------|----------------|
| STATUS | CLAIM | STORAGECLASS | REASON | AGE |
| Bound | pvc-c433420b-2311-4809-aae4-d11bf0aa2814 | 1Gi | RWO | Delete |
| Bound | default/mysql-pv-claim | gp2 | | 2m33s |
| Bound | pvc-e1dca128-12ca-47b4-8ba5-ab8df50e7128 | 1Gi | RWO | Delete |
| Bound | default/wp-pv-claim | gp2 | | 2m33s |

C:\Users\user\Desktop\eks cloud>kubectl get deploy

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|-----------------|-------|------------|-----------|-------|
| wordpress | 1/1 | 1 | 1 | 2m46s |
| wordpress-mysql | 1/1 | 1 | 1 | 2m46s |

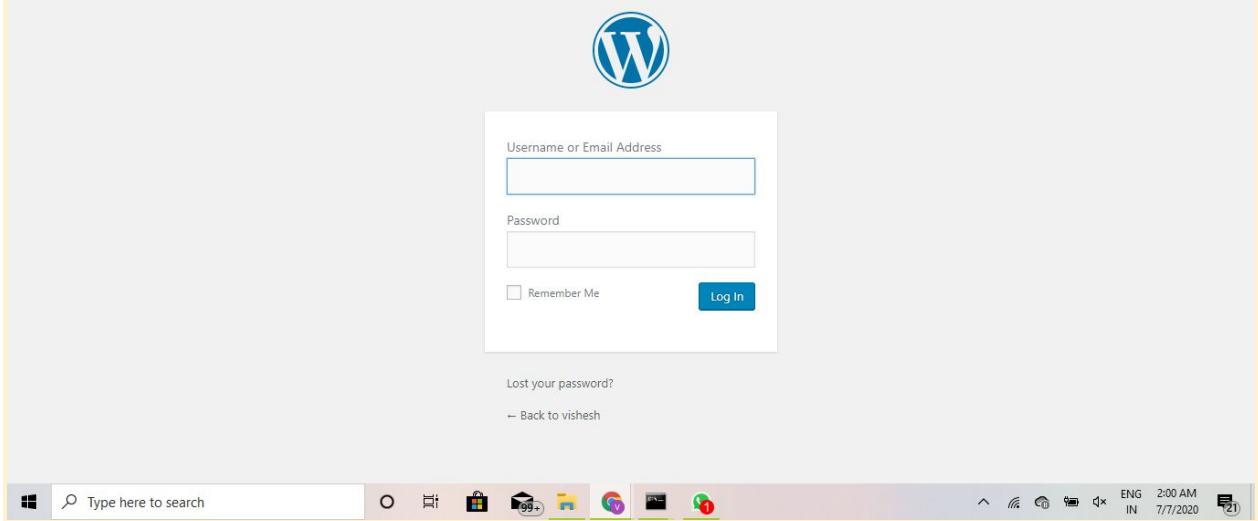
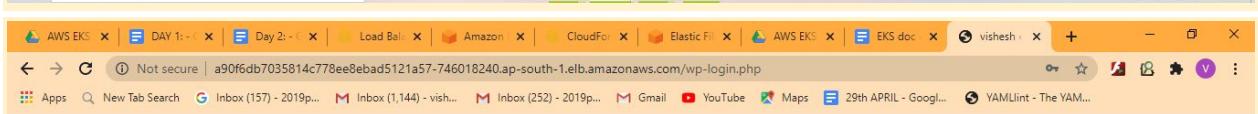
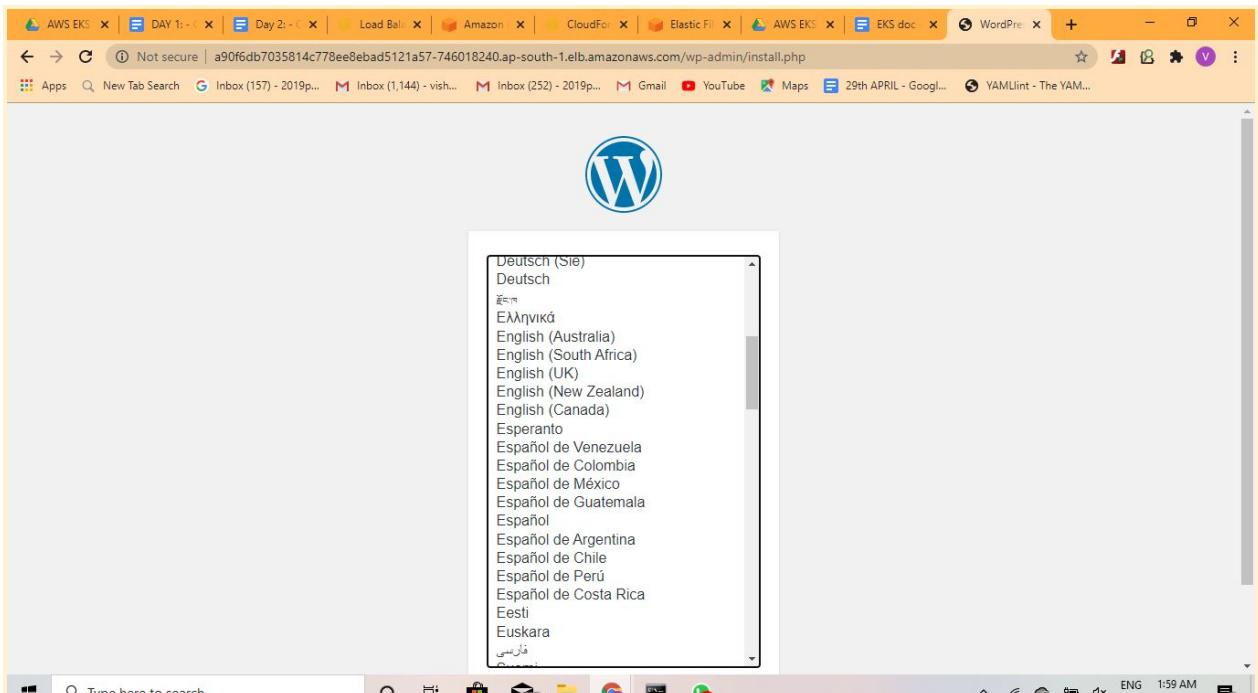
C:\Users\user\Desktop\eks cloud>kubectl get all

| NAME | READY | STATUS | RESTARTS | AGE |
|--------------------------------------|-------|---------|----------|-------|
| pod/wordpress-88cb86b9b-v5cs6 | 1/1 | Running | 0 | 2m52s |
| pod/wordpress-mysql-66b4cc9ccb-rwts6 | 1/1 | Running | 0 | 2m52s |

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP |
|-------------------------|--------------|--------------|---|
| PORT(S) | AGE | | |
| service/kubernetes | ClusterIP | 10.100.0.1 | <none> |
| 443/TCP | 42m | | |
| service/wordpress | LoadBalancer | 10.100.0.143 | a90f6db7035814c778ee8ebad5121a57-746018240.ap-south-1.elb.amazonaws.com |
| 80:30288/TCP | 2m52s | | |
| service/wordpress-mysql | ClusterIP | None | <none> |
| 3306/TCP | 2m52s | | |

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|---------------------------------|-------|------------|-----------|-------|
| deployment.apps/wordpress | 1/1 | 1 | 1 | 2m53s |
| deployment.apps/wordpress-mysql | 1/1 | 1 | 1 | 2m53s |

| NAME | DESIRED | CURRENT | READY | AGE |
|--|---------|---------|-------|-------|
| replicaset.apps/wordpress-88cb86b9b | 1 | 1 | 1 | 2m53s |
| replicaset.apps/wordpress-mysql-66b4cc9ccb | 1 | 1 | 1 | 2m53s |



JULY 6, 2020 BY VISHESH

my project

hiii vishesh here

Edit

RECENT POSTS

- my project
- Hello world!

RECENT COMMENTS

- A WordPress Commenter on Hello world!

```
C:\Users\user\Desktop\eks cloud>kubectl delete pod
wordpress-mysql-66b4cc9ccb-rwts6
pod "wordpress-mysql-66b4cc9ccb-rwts6" deleted
```

| NAME | READY | STATUS | RESTARTS | AGE |
|------|-------|--------|----------|-----|
|------|-------|--------|----------|-----|

| | | | | |
|--------------------------------------|-----|---------|---|------|
| pod/wordpress-88cb86b9b-v5cs6 | 1/1 | Running | 0 | 7m2s |
| pod/wordpress-mysql-66b4cc9ccb-gpkdj | 1/1 | Running | 0 | 13s |

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP |
|-------------------------|--------------|--------------|---|
| PORT(S) | AGE | | |
| service/kubernetes | ClusterIP | 10.100.0.1 | <none> |
| 443/TCP | 46m | | |
| service/wordpress | LoadBalancer | 10.100.0.143 | a90f6db7035814c778ee8ebad5121a57-746018240.ap-south-1.elb.amazonaws.com |
| 80:30288/TCP | 7m3s | | |
| service/wordpress-mysql | ClusterIP | None | <none> |
| 3306/TCP | 7m3s | | |

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|---------------------------------|-------|------------|-----------|------|
| deployment.apps/wordpress | 1/1 | 1 | 1 | 7m3s |
| deployment.apps/wordpress-mysql | 1/1 | 1 | 1 | 7m3s |

| NAME | DESIRED | CURRENT | READY | AGE |
|--|---------|---------|-------|------|
| replicaset.apps/wordpress-88cb86b9b | 1 | 1 | 1 | 7m3s |
| replicaset.apps/wordpress-mysql-66b4cc9ccb | 1 | 1 | 1 | 7m3s |

The image shows three stacked screenshots of a Microsoft Edge browser window.

Top Screenshot: A WordPress blog post titled "my project". The post content is "hiii vishesh here". On the right side, there are "RECENT POSTS" and "RECENT COMMENTS" sections. The "RECENT POSTS" section includes links to "my project" and "Hello world!". The "RECENT COMMENTS" section shows a comment from "A WordPress Commenter" on the "Hello world!" post.

Middle Screenshot: A GitHub gist page with the URL gist.github.com/mgoodness/1a2926f3b02d8e8149c224d25cc57dc1. The content of the gist is a list of AWS Lambda function configuration variables:

- service.beta.kubernetes.io/aws-load-balancer-connection-idle-timeout (in seconds, default 60)
- service.beta.kubernetes.io/aws-load-balancer-cross-zone-load-balancing-enabled (true|false)
- service.beta.kubernetes.io/aws-load-balancer-extra-security-groups (comma-separated list)
- service.beta.kubernetes.io/aws-load-balancer-healthcheck-threshold
- service.beta.kubernetes.io/aws-load-balancer-healthcheck-interval
- service.beta.kubernetes.io/aws-load-balancer-healthcheck-timeout
- service.beta.kubernetes.io/aws-load-balancer-healthcheck-unhealthy-threshold
- service.beta.kubernetes.io/aws-load-balancer-internal (true|false)
- service.beta.kubernetes.io/aws-load-balancer-proxy-protocol: '*' (true|false)
- service.beta.kubernetes.io/aws-load-balancer-ssl-cert (IAM or ACM ARN)
- service.beta.kubernetes.io/aws-load-balancer-ssl-negotiation-policy
- service.beta.kubernetes.io/aws-load-balancer-ssl-ports (default '*')
- service.beta.kubernetes.io/aws-load-balancer-type: nlb

A comment by "KIVagant" is visible, thanking the author and providing a link to the Kubernetes code: <https://github.com/kubernetes/kubernetes/blob/master/pkg/cloudprovider/providers/aws/aws.go#L72>.

Bottom Screenshot: A blank Microsoft Edge browser window with a search bar containing "Type here to search". The taskbar at the bottom shows various pinned and open application icons.

wordpress-deployment - Notepad

```
File Edit Format View Help
apiVersion: v1
kind: Service
Annotation:
  - service.beta.kubernetes.io/aws-load-balancer-type: nlb
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
---
apiVersion: v1
kind: PersistentVolumeClaim
```

The screenshot shows the AWS Lambda console with the function configuration page open. The left sidebar lists 'Functions' (vishcluster), 'Actions', 'Logs', and 'Metrics'. The main area displays the function configuration with tabs for 'Handler', 'Runtime', 'Code', 'Environment', 'Tracing', and 'Logs'. The 'Handler' tab shows 'index.handler' and 'Runtime' shows 'Node.js 12.x'. The 'Code' tab indicates the code is 'Up-to-date' from GitHub. The 'Environment' tab shows environment variables like 'AWS_LAMBDA_FUNCTION_NAME' and 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE'. The 'Tracing' tab shows 'AWS X-Ray' is enabled. The 'Logs' tab shows log groups for the function.

Launch instance wizard | EC2 Management Console

ap-southeast-1.console.aws.amazon.com/ec2/v2/home?region=ap-southeast-1#LaunchInstanceWizard:

Subnet: IN0 preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open Create new Capacity Reservation

IAM role: None Create new IAM role

None
eksctl-vishcluster-nodegroup-ng1-NodeInstanceProfile-QNM8EJN4RMHY
eksctl-vishcluster-nodegroup-ng2-NodeInstanceProfile-14061F57C6THO

Shutdown behavior: Stop - Hibernate behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring Additional charges apply.

Tenancy: Shared - Run a shared hardware instance Additional charges may apply when launching Dedicated instances.

Cancel Previous Review and Launch Next: Add Storage

Create EFS system

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Type here to search

ap-south-1.console.aws.amazon.com/efs/home?region=ap-south-1#/wizard/

Step 1: Configure network access

Step 2: Configure file system settings

Step 3: Configure client access

Step 4: Review and create

Configure network access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC: vpc-08435fe4542f1c1f2... vpc-15f8e57d (default)

Create mount target: vpc-08435fe4542f1c1f2 - eksctl-vishcluster-cluster/VPC

Instances connect to the file system through the mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across the VPC can access the file system.

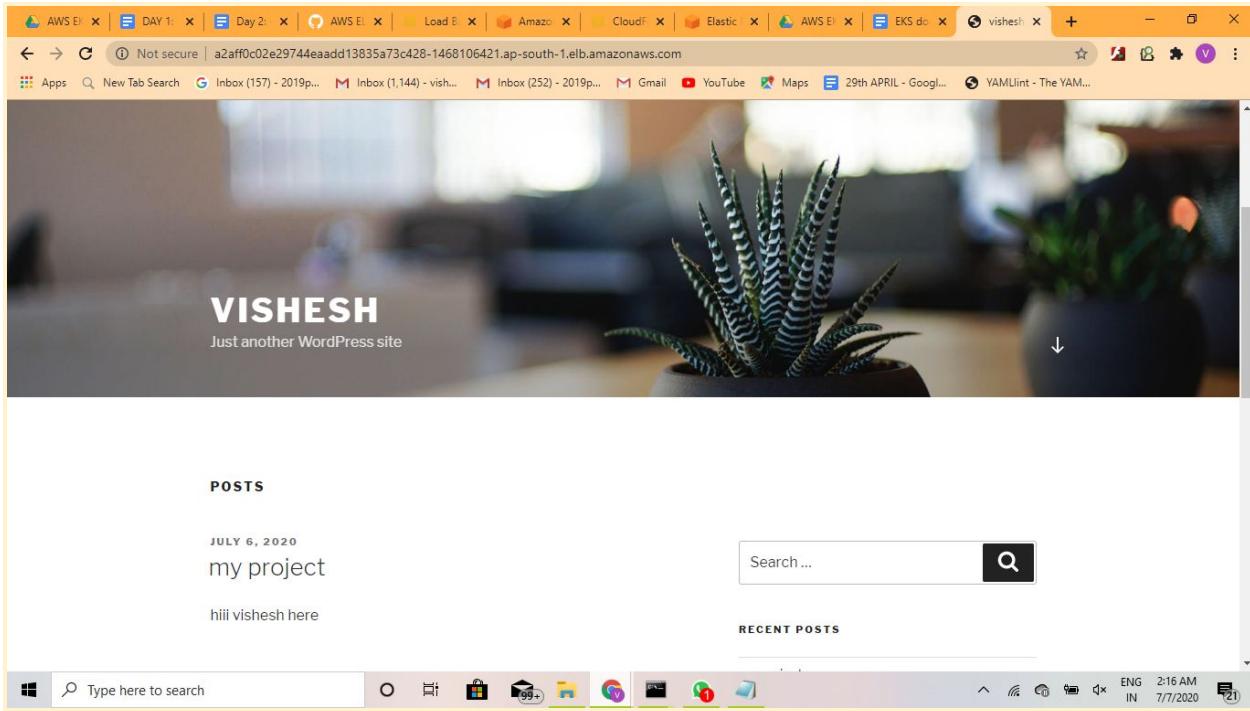
| | Availability Zone | Subnet | IP address | Security groups |
|-------------|--|-----------|---------------------------------|-----------------|
| ap-south-1a | subnet-07fab58f8549f60c9 - eksctl-vishcluster-cluster/SubnetPrivateAPSOUTH1A | Automatic | sg-0a5b180560f799dcdd - default | |
| ap-south-1b | subnet-02c316dba4d2fd30c - eksctl-vishcluster-cluster/SubnetPublicAPSOUTH1B | Automatic | sg-0a5b180560f799dcdd - default | |

Type here to search

A screenshot of the AWS EFS console. The top navigation bar shows various AWS services like AWS EKS, AWS ELB, Load Balancer, Amazon, CloudFront, Elastic File, AWS EKS, EKS doc, and more. The main content area is titled "File systems". It features a "Create file system" button and an "Actions" dropdown. A table lists one file system: Name: fs-68bc37b9, File system ID: fs-68bc37b9, Metered size: 6.0 Kib, Number of mount targets: 3, Creation date: 07/06/2020, 20:42:07 UTC.

```
C:\Users\user\Desktop\eks cloud>kubectl create --validate=false -k .
secret/mysql-pass-ctm2f4889c created
service/wordpress-mysql created
service/wordpress created
deployment.apps/wordpress-mysql created
C:\Users\user\Desktop\eks cloud>kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP
PORT(S)    AGE
kubernetes   ClusterIP   10.100.0.1    <none>
443/TCP    96s
wordpress   LoadBalancer 10.100.206.20
a2aff0c02e29744eaadd13835a73c428-1468106421.ap-south-1.elb.amazonaws.com
80:32540/TCP 51s
wordpress-mysql ClusterIP  None        <none>
3306/TCP    51s
```

```
C:\Users\user\Desktop\eks cloud>kubectl get secret
NAME          TYPE           DATA  AGE
default-token-7mgdj  kubernetes.io/service-account-token  3    60m
mysql-pass-ctm2f4889c  Opaque          1    57s
```



```
C:\Users\user\Desktop\eks cloud>kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
wordpress-88cb86b9b-h9b6c   1/1     Running   0          96s
wordpress-mysql-66b4cc9ccb-dqk76  1/1     Running   0          96s
C:\Users\user\Desktop\eks cloud>kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP             NODE
NOMINATED NODE  READINESS GATES
wordpress-88cb86b9b-h9b6c   1/1     Running   0      2m14s  192.168.24.135
ip-192-168-21-219.ap-south-1.compute.internal <none>        <none>
wordpress-mysql-66b4cc9ccb-dqk76  1/1     Running   0      2m14s
192.168.10.233 ip-192-168-21-219.ap-south-1.compute.internal <none>
<none>
C:\Users\user\Desktop\eks cloud>helm init
$HELM_HOME has been configured at C:\Users\user\.helm.
```

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users' policy.

To prevent this, run `helm init` with the --tiller-tls-verify flag.

For more information on securing your installation see:

https://v2.helm.sh/docs/securing_installation/

C:\Users\user\Desktop\eks cloud>helm repo add stable

<https://kubernetes-charts.storage.googleapis.com/>

"stable" has been added to your repositories

C:\Users\user\Desktop\eks cloud> helm repo list

NAME URL

stable <https://kubernetes-charts.storage.googleapis.com/>

local <http://127.0.0.1:8879/charts>

C:\Users\user\Desktop\eks cloud>

C:\Users\user\Desktop\eks cloud>helm repo update

Hang tight while we grab the latest from your chart repositories...

...Skip local chart repository

...Successfully got an update from the "stable" chart repository

Update Complete.

C:\Users\user\Desktop\eks cloud>kubectl create ns lw1

namespace/lw1 created

C:\Users\user\Desktop\eks cloud>helm install --name my-release stable/jenkins -n lw1

Error: release lw1 failed: namespaces "default" is forbidden: User "system:serviceaccount:kube-system:default" cannot get resource "namespaces" in API group "" in the namespace "default"

C:\Users\user\Desktop\eks cloud>kubectl get pods -n kube-system

| NAME | READY | STATUS | RESTARTS | AGE |
|--------------------------------|-------|---------|----------|------|
| aws-node-hnx26 | 1/1 | Running | 0 | 58m |
| aws-node-v6vz6 | 1/1 | Running | 0 | 58m |
| aws-node-xp8g8 | 1/1 | Running | 0 | 58m |
| coredns-6856799b8d-4bb5j | 1/1 | Running | 0 | 65m |
| coredns-6856799b8d-8hv7j | 1/1 | Running | 0 | 65m |
| kube-proxy-98hbl | 1/1 | Running | 0 | 58m |
| kube-proxy-n9jcr | 1/1 | Running | 0 | 58m |
| kube-proxy-rplkh | 1/1 | Running | 0 | 58m |
| tiller-deploy-6974685dbc-p59g2 | 1/1 | Running | 0 | 3m7s |

C:\Users\user\Desktop\eks cloud>kubectl -n kube-system create serviceaccount tiller

serviceaccount/tiller created

```
C:\Users\user\Desktop\eks cloud>kubectl create clusterrolebinding tiller  
--clusterrole cluster-admin --serviceaccount=kube-system:tiller  
clusterrolebinding.rbac.authorization.k8s.io/tiller created
```

```
C:\Users\user\Desktop\eks cloud>helm init --service-account tiller  
$HELM_HOME has been configured at C:\Users\user\.helm.  
Warning: Tiller is already installed in the cluster.  
(Use --client-only to suppress this message, or --upgrade to upgrade Tiller to the  
current version.)
```

```
C:\Users\user\Desktop\eks cloud>kubectl get pods --namespace kube-system  
NAME          READY   STATUS    RESTARTS   AGE  
aws-node-hnx26   1/1     Running   0          59m  
aws-node-v6vz6   1/1     Running   0          59m  
aws-node-xp8g8   1/1     Running   0          59m  
coredns-6856799b8d-4bb5j   1/1     Running   0          66m  
coredns-6856799b8d-8hv7j   1/1     Running   0          66m  
kube-proxy-98hbl   1/1     Running   0          59m  
kube-proxy-n9jcr   1/1     Running   0          59m  
kube-proxy-rplkh   1/1     Running   0          59m  
tiller-deploy-6974685dbc-p59g2 1/1     Running   0          4m8s  
C:\Users\user\Desktop\eks cloud>helm init --service-account tiller --upgrade  
$HELM_HOME has been configured at C:\Users\user\.helm.
```

Tiller (the Helm server-side component) has been updated to
gcr.io/kubernetes-helm/tiller:v2.16.9 .

```
C:\Users\user\Desktop\eks cloud>helm install --name myjn stable/jenkins  
Error: release myjn failed: namespaces "default" is forbidden: User  
"system:serviceaccount:kube-system:default" cannot get resource  
"namespaces" in API group "" in the namespace "default"
```

```
C:\Users\user\Desktop\eks cloud>helm init --service-account tiller --upgrade  
$HELM_HOME has been configured at C:\Users\user\.helm.
```

Tiller (the Helm server-side component) has been updated to
gcr.io/kubernetes-helm/tiller:v2.16.9 .

```
C:\Users\user\Desktop\eks cloud>helm install my-release stable/jenkins  
Error: This command needs 1 argument: chart name
```

```
C:\Users\user\Desktop\eks cloud>helm install --name my-release stable/jenkins  
NAME: my-release  
LAST DEPLOYED: Tue Jul  7 02:30:01 2020  
NAMESPACE: default  
STATUS: DEPLOYED
```

RESOURCES:

```
==> v1/ConfigMap
```

| NAME | DATA | AGE |
|---|------|-----|
| my-release-jenkins | 2 | 2s |
| my-release-jenkins-jenkins-jcasc-config | 1 | 2s |
| my-release-jenkins-tests | 1 | 2s |

```
==> v1/Deployment
```

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|--------------------|-------|------------|-----------|-----|
| my-release-jenkins | 0/1 | 1 | 0 | 1s |

```
==> v1/PersistentVolumeClaim
```

| NAME | STATUS | VOLUME | CAPACITY | ACCESS MODES | AGE |
|--------------------|---------|--------|----------|--------------|-----|
| my-release-jenkins | Pending | | | | 2s |

```
==> v1/Pod(related)
```

| NAME | READY | STATUS | RESTARTS | AGE |
|-------------------------------------|-------|---------|----------|-----|
| my-release-jenkins-79d79fb448-zkmp2 | 0/2 | Pending | 0 | 1s |

```
==> v1/Role
```

| NAME | AGE |
|------------------------------------|-----|
| my-release-jenkins-casc-reload | 1s |
| my-release-jenkins-schedule-agents | 1s |

```
==> v1/RoleBinding
```

| NAME | AGE |
|-------------------------------------|-----|
| my-release-jenkins-watch-configmaps | 1s |
| my-release-jenkins-schedule-agents | 1s |

==> v1/Secret

| NAME | TYPE | DATA | AGE |
|--------------------|--------|------|-----|
| my-release-jenkins | Opaque | 2 | 2s |

==> v1/Service

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|--------------------------|-----------|----------------|-------------|-----------|-----|
| my-release-jenkins | ClusterIP | 10.100.82.240 | <none> | 8080/TCP | 1s |
| my-release-jenkins-agent | ClusterIP | 10.100.219.230 | <none> | 50000/TCP | 1s |

==> v1/ServiceAccount

| NAME | SECRETS | AGE |
|--------------------|---------|-----|
| my-release-jenkins | 1 | 2s |

NOTES:

1. Get your 'admin' user password by running:

```
printf $(kubectl get secret --namespace default my-release-jenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode);echo
```

2. Get the Jenkins URL to visit by running these commands in the same shell:

```
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/component=jenkins-master" -l "app.kubernetes.io/instance=my-release" -o jsonpath="{.items[0].metadata.name}")  
echo http://127.0.0.1:8080  
kubectl --namespace default port-forward $POD_NAME 8080:8080
```

3. Login with the password from step 1 and the username: admin

4. Use Jenkins Configuration as Code by specifying configScripts in your values.yaml file, see documentation: <http://configuration-as-code> and examples: <https://github.com/jenkinsci/configuration-as-code-plugin/tree/master/demos>

For more information on running Jenkins on Kubernetes, visit:

<https://cloud.google.com/solutions/jenkins-on-container-engine>

For more information about Jenkins Configuration as Code, visit:

<https://jenkins.io/projects/jcasc/>

C:\Users\user\Desktop\eks cloud>kubectl get pods

| NAME | READY | STATUS | RESTARTS | AGE |
|------|-------|--------|----------|-----|
| | | | | |

```
my-release-jenkins-79d79fb448-zkmp2 0/2 Pending 0 66s
```

```
C:\Users\user\Desktop\eks cloud>kubectl get pods
```

| NAME | READY | STATUS | RESTARTS | AGE |
|-------------------------------------|-------|---------|----------|-----|
| my-release-jenkins-79d79fb448-zkmp2 | 0/2 | Pending | 0 | 92s |

```
C:\Users\user\Desktop\eks cloud>kubectl get svc
```

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|--------------------------|-----------|----------------|-------------|-----------|------|
| kubernetes | ClusterIP | 10.100.0.1 | <none> | 443/TCP | 8m2s |
| my-release-jenkins | ClusterIP | 10.100.82.240 | <none> | 8080/TCP | 2m4s |
| my-release-jenkins-agent | ClusterIP | 10.100.219.230 | <none> | 50000/TCP | 2m4s |

The image displays two side-by-side screenshots of a Microsoft Edge browser window.

Top Screenshot: The title bar shows multiple tabs open, including AWS, DAY, AWS, Load, Cloud, AWS, EKS, vishe, Kube, and others. The active tab is "hub.helm.sh/charts?q=prometheus". The main content area is titled "Charts" and shows a search bar with "prometheu". Below it is a sidebar titled "Repository" listing various chart sources: all, stable, incubator, jfrog, kremers, linkerd2, linkerd2-edge, rimusz, buildkite, cloudposse, cockroachdb, keel, webhookrelay, and others. To the right, there are four chart cards displayed:

- aerospike/aerospike** version 5.0.0.4
- aerospike/aerospike-enterprise** version 5.0.0.4
- datawire/ambassador** version 1.5.5
- aws/appmesh-prometheus** version 2.13.1

Bottom Screenshot: The title bar shows multiple tabs open, including AWS, DAY, AWS, Load, Cloud, AWS, EKS, vishe, helm, and others. The active tab is "google.com/search?...". The main content area shows a Google search result for "helm prometheus chart". The snippet includes a link to "Prometheus Helm - GitHub" and a brief description: "No information is available for this page. Learn why". Below this, another snippet is shown: "hub.helm.sh › charts › stable › prometheus" followed by "prometheus 2.19.0 for Kubernetes | Helm Hub | Monocular". It describes the chart as bootstrapping a Prometheus deployment on a Kubernetes cluster using the Helm package manager. The snippet ends with "To install the chart with the release name my-release : \$ helm install opsgoodness/prometheus --name my-release. The command deploys Prometheus on the ...".

Introduction

This chart bootstraps a [Prometheus](#) deployment on a [Kubernetes](#) cluster using the [Helm](#) package manager.

Prerequisites

- Kubernetes 1.3+ with Beta APIs enabled

Installing the Chart

To install the chart with the release name `my-release`:

```
$ helm install --name my-release stable/prometheus
```

The command deploys Prometheus on the Kubernetes cluster in the default configuration. The [configuration](#) section lists the parameters that can be configured during installation.

Tip: List all releases using `helm list`

Uninstalling the Chart

To uninstall/delete the `my-release` deployment:

C:\Users\user\Desktop\eks cloud>`kubectl create namespace prometheus`
namespace/prometheus created

C:\Users\user\Desktop\eks cloud>`helm install stable/prometheus --namespace prometheus --set alertmanager.persistentVolume.storageClass="gp2" --set server.persistentVolume.storageClass="gp2"`
NAME: youthful-antelope
LAST DEPLOYED: Tue Jul 7 02:38:09 2020
NAMESPACE: prometheus
STATUS: DEPLOYED

RESOURCES:

==> v1/ConfigMap

| NAME | DATA AGE |
|--|-------------|
| youthful-antelope-prometheus-alertmanager | 1 1s |
| youthful-antelope-prometheus-server | 5 1s |

==> v1/DaemonSet

| NAME | DESIRED | CURRENT | READY | UP-TO-DATE |
|---|----------|----------|----------|------------|
| AVAILABLE NODE SELECTOR AGE | | | | |
| youthful-antelope-prometheus-node-exporter | 3 | 3 | 0 | 3 0 |

<none> 1s

==> v1/Deployment

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|---|-------|------------|-----------|-----|
| youthful-antelope-kube-state-metrics | 0/1 | 1 | 0 | 1s |
| youthful-antelope-prometheus-alertmanager | 0/1 | 1 | 0 | 1s |
| youthful-antelope-prometheus-pushgateway | 0/1 | 1 | 0 | 1s |
| youthful-antelope-prometheus-server | 0/1 | 1 | 0 | 1s |

==> v1/PersistentVolumeClaim

| NAME | STATUS | VOLUME | CAPACITY | ACCESS MODES |
|---|---------|--------|----------|--------------|
| STORAGECLASS | AGE | | | |
| youthful-antelope-prometheus-alertmanager | Pending | gp2 | 1s | |
| youthful-antelope-prometheus-server | Pending | gp2 | 1s | |

==> v1/Pod(related)

| NAME | READY | STATUS | RESTARTS | AGE |
|--|-------|---------|----------|-----|
| youthful-antelope-kube-state-metrics-7c779dbccc-6nfqp | 0/1 | | | |
| ContainerCreating | 0 | 1s | | |
| youthful-antelope-prometheus-alertmanager-659bc644fc-br2dz | 0/2 | Pending | | |
| 0 | 1s | | | |
| youthful-antelope-prometheus-node-exporter-crgqn | 0/1 | | | |
| ContainerCreating | 0 | 1s | | |
| youthful-antelope-prometheus-node-exporter-r9d9m | 0/1 | Pending | 0 | |
| 1s | | | | |
| youthful-antelope-prometheus-node-exporter-tnvgq | 0/1 | | | |
| ContainerCreating | 0 | 1s | | |
| youthful-antelope-prometheus-pushgateway-788f6b97d7-dtzcs | 0/1 | | | |
| ContainerCreating | 0 | 1s | | |
| youthful-antelope-prometheus-server-76b6fbdf59-s9mb6 | 0/2 | Pending | | |
| 0 | 1s | | | |

==> v1/Service

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) |
|---|-----------|---------------|-------------|---------|
| AGE | | | | |
| youthful-antelope-kube-state-metrics | ClusterIP | 10.100.56.221 | <none> | |
| 8080/TCP | 1s | | | |
| youthful-antelope-prometheus-alertmanager | ClusterIP | 10.100.107.84 | <none> | |
| 80/TCP | 1s | | | |

```
youthful-antelope-prometheus-node-exporter ClusterIP None <none>
9100/TCP 1s
youthful-antelope-prometheus-pushgateway ClusterIP 10.100.195.151 <none>
9091/TCP 1s
youthful-antelope-prometheus-server ClusterIP 10.100.242.20 <none>
80/TCP 1s
```

==> v1/ServiceAccount

| NAME | SECRETS | AGE |
|--|---------|-----|
| youthful-antelope-kube-state-metrics | 1 | 1s |
| youthful-antelope-prometheus-alertmanager | 1 | 1s |
| youthful-antelope-prometheus-node-exporter | 1 | 1s |
| youthful-antelope-prometheus-pushgateway | 1 | 1s |
| youthful-antelope-prometheus-server | 1 | 1s |

==> v1beta1/ClusterRole

| NAME | AGE |
|---|-----|
| youthful-antelope-prometheus-server | 2s |
| youthful-antelope-kube-state-metrics | 2s |
| youthful-antelope-prometheus-alertmanager | 2s |
| youthful-antelope-prometheus-pushgateway | 2s |

==> v1beta1/ClusterRoleBinding

| NAME | AGE |
|---|-----|
| youthful-antelope-prometheus-server | 2s |
| youthful-antelope-kube-state-metrics | 2s |
| youthful-antelope-prometheus-alertmanager | 2s |
| youthful-antelope-prometheus-pushgateway | 2s |

NOTES:

The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:

`youthful-antelope-prometheus-server.prometheus.svc.cluster.local`

Get the Prometheus server URL by running these commands in the same shell:

```
export POD_NAME=$(kubectl get pods --namespace prometheus -l
"app=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
```

```
kubectl --namespace prometheus port-forward $POD_NAME 9090
```

The Prometheus alertmanager can be accessed via port 80 on the following DNS name from within your cluster:

youthful-antelope-prometheus-alertmanager.prometheus.svc.cluster.local

Get the Alertmanager URL by running these commands in the same shell:

```
export POD_NAME=$(kubectl get pods --namespace prometheus -l  
"app=prometheus,component=alertmanager" -o  
jsonpath="{.items[0].metadata.name}")  
kubectl --namespace prometheus port-forward $POD_NAME 9093  
#####  
#####  
##### WARNING: Pod Security Policy has been moved to a global property.  
#####  
#####      use .Values.podSecurityPolicy.enabled with pod-based #####  
#####      annotations           #####  
#####      (e.g. .Values.nodeExporter.podSecurityPolicy.annotations) #####  
#####  
#####
```

The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:

youthful-antelope-prometheus-pushgateway.prometheus.svc.cluster.local

Get the PushGateway URL by running these commands in the same shell:

```
export POD_NAME=$(kubectl get pods --namespace prometheus -l  
"app=prometheus,component=pushgateway" -o  
jsonpath="{.items[0].metadata.name}")  
kubectl --namespace prometheus port-forward $POD_NAME 9091
```

For more information on running Prometheus, visit:

<https://prometheus.io/>

C:\Users\user\Desktop\eks cloud>kubectl get pods -n prometheus

| NAME | READY | STATUS | RESTARTS | AGE |
|------|-------|--------|----------|-----|
|------|-------|--------|----------|-----|

```

youthful-antelope-kube-state-metrics-7c779dbccc-6nfqp      1/1  Running  0
46s
youthful-antelope-prometheus-alertmanager-659bc644fc-br2dz  0/2  Pending   0
46s
youthful-antelope-prometheus-node-exporter-crgqn          1/1  Running  0
46s
youthful-antelope-prometheus-node-exporter-r9d9m           0/1  Pending   0
46s
youthful-antelope-prometheus-node-exporter-tnvgq           1/1  Running  0
46s
youthful-antelope-prometheus-pushgateway-788f6b97d7-dtzcs  1/1  Running
0      46s
youthful-antelope-prometheus-server-76b6fbdf59-s9mb6       0/2  Pending   0
46s
C:\Users\user\Desktop\eks cloud>kubectl get svc -n prometheus
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP PORT(S)
AGE
youthful-antelope-kube-state-metrics    ClusterIP  10.100.56.221 <none>
8080/TCP  75s
youthful-antelope-prometheus-alertmanager ClusterIP  10.100.107.84 <none>
80/TCP   75s
youthful-antelope-prometheus-node-exporter ClusterIP  None        <none>
9100/TCP  75s
youthful-antelope-prometheus-pushgateway  ClusterIP  10.100.195.151
<none>   9091/TCP 75s
youthful-antelope-prometheus-server       ClusterIP  10.100.242.20 <none>
80/TCP   75s

```

Screenshot of a Windows desktop showing two Command Prompt windows and a Google Chrome browser.

The top Command Prompt window shows the output of running `kubectl` commands to list pods and services in the default namespace of an EKS cluster named "cloud".

```

C:\Users\user\Desktop\eks cloud>kubectl get pods
No resources found in default namespace.

C:\Users\user\Desktop\eks cloud>kubectl get pods -n prometheus
NAME                                     READY   STATUS    RESTARTS   AGE
youthful-antelope-kube-state-metrics-7c779dbccc-6nfqp   1/1    Running   0          46s
youthful-antelope-prometheus-alertmanager-659bc644fc-br2dz 0/2    Pending   0          46s
youthful-antelope-prometheus-node-exporter-crgqn         1/1    Running   0          46s
youthful-antelope-prometheus-node-exporter-r9d9m         0/1    Pending   0          46s
youthful-antelope-prometheus-node-exporter-tnvgg         1/1    Running   0          46s
youthful-antelope-prometheus-pushgateway-788f6b97d7-dtzcs 1/1    Running   0          46s
youthful-antelope-prometheus-server-76b6fbdf59-s9mb6      0/2    Pending   0          46s

C:\Users\user\Desktop\eks cloud>kubectl get svc -n prometheus
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
youthful-antelope-kube-state-metrics   ClusterIP  10.100.56.221 <none>     8080/TCP  75s
youthful-antelope-prometheus-alertmanager   ClusterIP  10.100.107.84  <none>     80/TCP   75s
youthful-antelope-prometheus-node-exporter   ClusterIP  None         <none>     9100/TCP  75s
youthful-antelope-prometheus-pushgateway   ClusterIP  10.100.195.151 <none>     9091/TCP  75s
youthful-antelope-prometheus-server   ClusterIP  10.100.242.20  <none>     80/TCP   75s
  
```

The bottom Command Prompt window shows the output of running `kubectl` commands to list pods and services in the "prometheus" namespace of the same EKS cluster.

```

C:\Users\user\Desktop\eks cloud>kubectl get pods
No resources found in default namespace.

C:\Users\user\Desktop\eks cloud>kubectl get pods -n prometheus
NAME                                     READY   STATUS    RESTARTS   AGE
youthful-antelope-kube-state-metrics-7c779dbccc-6nfqp   1/1    Running   0          46s
youthful-antelope-prometheus-alertmanager-659bc644fc-br2dz 0/2    Pending   0          46s
youthful-antelope-prometheus-node-exporter-crgqn         1/1    Running   0          46s
youthful-antelope-prometheus-node-exporter-r9d9m         0/1    Pending   0          46s
youthful-antelope-prometheus-node-exporter-tnvgg         1/1    Running   0          46s
youthful-antelope-prometheus-pushgateway-788f6b97d7-dtzcs 1/1    Running   0          46s
youthful-antelope-prometheus-server-76b6fbdf59-s9mb6      0/2    Pending   0          46s

C:\Users\user\Desktop\eks cloud>kubectl get svc -n prometheus
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)   AGE
youthful-antelope-kube-state-metrics   ClusterIP  10.100.56.221 <none>     8080/TCP  75s
youthful-antelope-prometheus-alertmanager   ClusterIP  10.100.107.84  <none>     80/TCP   75s
youthful-antelope-prometheus-node-exporter   ClusterIP  None         <none>     9100/TCP  75s
youthful-antelope-prometheus-pushgateway   ClusterIP  10.100.195.151 <none>     9091/TCP  75s
youthful-antelope-prometheus-server   ClusterIP  10.100.242.20  <none>     80/TCP   75s
  
```

The browser window shows a Google Doc titled "aws eks" which contains a link to a YML file for a Jenkins pipeline named "jenkins".

The taskbar at the bottom shows several pinned icons, including AWS EKS, CloudFormation, and Jenkins.

C:\Users\user\Desktop\eks cloud>**kubectl -n prometheus port-forward svc/youthful-antelope-prometheus-server 8888:80**

All Unhealthy

kubernetes-apiservers (2/2 up) show less

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|-------|---|-------------|-----------------|-------|
| https://192.168.114.229:443/metrics | UP | instance="192.168.114.229:443" job="kubernetes-apiservers" | 21.117s ago | 100.3ms | |
| https://192.168.169.224:443/metrics | UP | instance="192.168.169.224:443" job="kubernetes-apiservers" | 13.862s ago | 75.74ms | |

kubernetes-nodes (7/7 up) show less

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|--|-------|--|-------------|-----------------|-------|
| https://kubernetes.default.svc:443 | UP | alpha_eksctl_io_cluster_name="lwcluster" | 56.924s ago | 12.06ms | |

kube_pod_container_info

Resolution: 14s
Total time series: 33

Execute - insert metric at cursor ↴

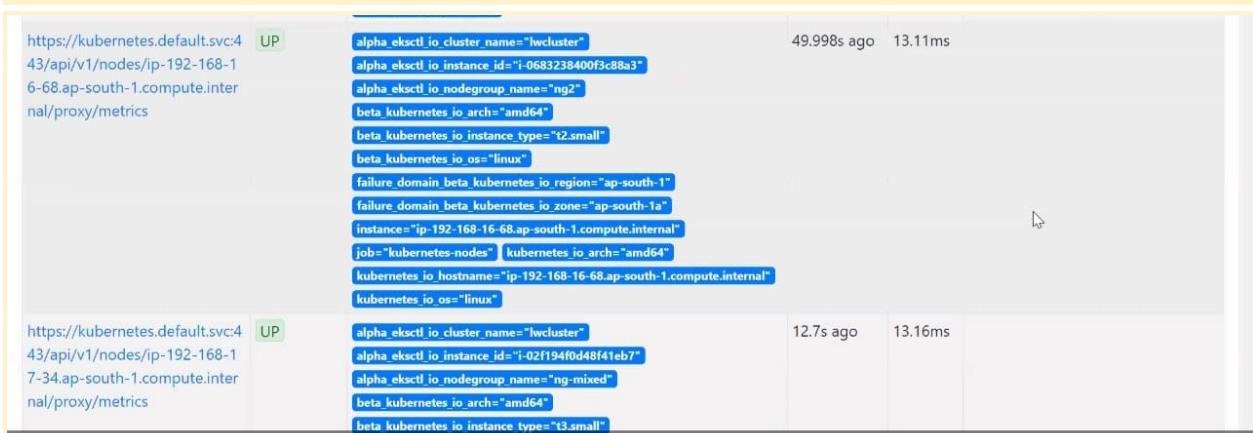
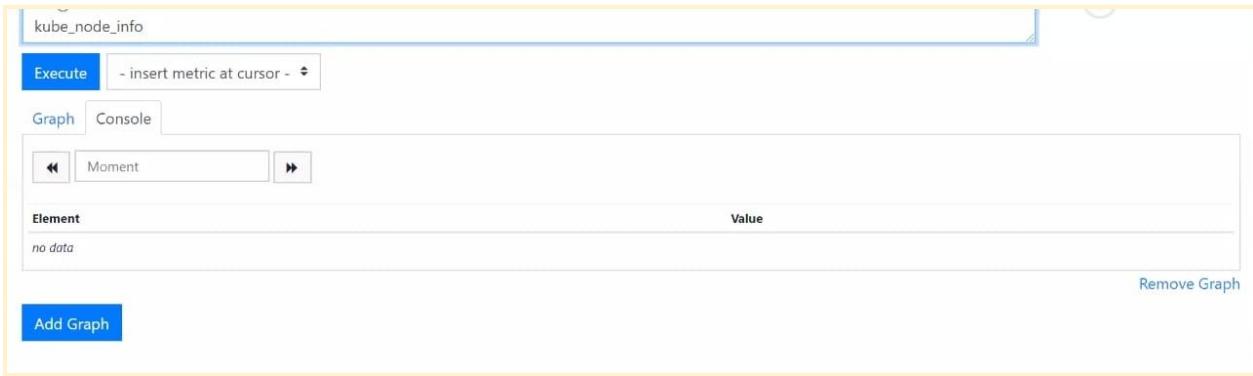
Graph Console

◀ Moment ▶

I

| Element | Value |
|--|-------|
| kube_pod_container_info{app_kubernetes_io_instance="independent-beetle",app_kubernetes_io_managed_by="Tiller",app_kubernetes_io_name="kube-state-metrics",container="aws-node",container_id="docker://9e376322b2459e7d2306da5fae08f8dc26c752172ab0c5f99e7b133874c0a384",helm_sh_chart="kube-state-metrics-2.8.11",image_id="602401143452.dkr.ecr.ap-south-1.amazonaws.com/amazon-k8s-cni:v1.6.3",image_id="docker-pullable://602401143452.dkr.ecr.ap-south-1.amazonaws.com/amazon-k8s-cni@sha256:8c42794a579e904b0b7014b7df64bb26d88011ab168125c844b96d916904d072",instance="192.168.26.71:8080",job="kubernetes-service-endpoints",kubernetes_name="independent-beetle-kube-state-metrics",kubernetes_namespace="prometheus",kubernetes_node="ip-192-168-16-68.ap-south-1.compute.internal",namespace="kube-system",pod="aws-node-gkbz4"} | 1 |
| kube_pod_container_info{app_kubernetes_io_instance="independent-beetle",app_kubernetes_io_managed_by="Tiller",app_kubernetes_io_name="kube-state-metrics",container="kube-proxy",container_id="docker://6650bf81991b608855de0f8dc4a93001bd7a3fd9a7a6d5cc786c566395fc72e7",helm_sh_chart="kube-state-metrics-2.8.11",image_id="602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/kube-proxy:v1.16.8",image_id="docker-pullable://602401143452.dkr.ecr.ap-south-1.amazonaws.com/eks/kube-proxy@sha256:dec3e952d45de3c3eeb9a981a6724a82b83e9620585fa521fc1b6dc42bd2fb8",instance="192.168.26.71:8080",job="kubernetes-service-endpoints",kubernetes_name="independent-beetle-kube-state-metrics",kubernetes_namespace="prometheus",kubernetes_node="ip-192-168-16-68.ap-south-1.compute.internal",namespace="kube-system",pod="kube-proxy-bgdgq"} | 1 |

| | beta_kubernetes_io_os="linux" | | | | |
|---|---|--|------------|--------|--|
| | failure_domain_beta_kubernetes_io_region="ap-south-1" failure_domain_beta_kubernetes_io_zone="ap-south-1b" instance="ip-192-168-89-245.ap-south-1.compute.internal" job="kubernetes-nodes-cadvisor" kubernetes_io_arch="amd64" kubernetes_io_hostname="ip-192-168-89-245.ap-south-1.compute.internal" kubernetes_io_os="linux" | | | | |
| https://kubernetes.default.svc:443/api/v1/nodes/ip-192-168-92-70.ap-south-1.compute.internal/proxy/metrics/cadvisor | UP | alpha_eksctl_io_cluster_name="lwcluster" alpha_eksctl_io_instance_id="i-01805899c6238a55e" alpha_eksctl_io_nodegroup_name="ng1" beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="t2.micro" beta_kubernetes_io_os="linux" failure_domain_beta_kubernetes_io_region="ap-south-1" failure_domain_beta_kubernetes_io_zone="ap-south-1b" instance="ip-192-168-92-70.ap-south-1.compute.internal" job="kubernetes-nodes-cadvisor" kubernetes_io_arch="amd64" | 7.223s ago | 34.2ms | |



The screenshot shows a browser window with the URL 'github.com/helm/charts/tree/master/stable/grafana'. The page contains instructions for installing the Grafana chart using Helm. It includes sections for 'TL;DR;', 'Installing the Chart', 'Uninstalling the Chart', and 'Upgrading an existing Release to a new major version'. It also notes that a major chart version change requires manual actions. The browser's address bar and taskbar are visible at the bottom.

<https://github.com/helm/charts/tree/master/stable/grafana>

```
C:\Users\user\Desktop\eks cloud>kubectl delete all --all -n prometheus
pod "youthful-antelope-kube-state-metrics-7c779dbccc-6nfqp" deleted
pod "youthful-antelope-prometheus-alertmanager-659bc644fc-br2dz" deleted
```

```
pod "youthful-antelope-prometheus-node-exporter-crgqn" deleted  
C:\Users\user\Desktop\eks cloud>kubectl create namespace grafana  
namespace/grafana created
```

```
C:\Users\user\Desktop\eks cloud>helm install stable/grafana --namespace  
grafana --set persistence.storageClassName="gp2" --set  
adminPassword='redhat' --set service.type=LoadBalancer  
NAME: lumpy-seal  
E0707 02:48:16.074232 10724 portforward.go:372] error copying from remote  
stream to local connection: readfrom tcp4 127.0.0.1:60327->127.0.0.1:60330: write  
tcp4 127.0.0.1:60327->127.0.0.1:60330: wsasend: An established connection was  
aborted by the software in your host machine.  
LAST DEPLOYED: Tue Jul 7 02:48:13 2020  
NAMESPACE: grafana  
STATUS: DEPLOYED
```

RESOURCES:

```
==> v1/ClusterRole
```

| NAME | AGE |
|--------------------------------|-----|
| lumpy-seal-grafana-clusterrole | 1s |

```
==> v1/ClusterRoleBinding
```

| NAME | AGE |
|---------------------------------------|-----|
| lumpy-seal-grafana-clusterrolebinding | 1s |

```
==> v1/ConfigMap
```

| NAME | DATA | AGE |
|-------------------------|------|-----|
| lumpy-seal-grafana | 1 | 1s |
| lumpy-seal-grafana-test | 1 | 1s |

```
==> v1/Deployment
```

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|--------------------|-------|------------|-----------|-----|
| lumpy-seal-grafana | 0/1 | 1 | 0 | 1s |

```
==> v1/Pod(related)
```

| NAME | READY | STATUS | RESTARTS | AGE |
|-------------------------------------|-------|-------------------|----------|-----|
| lumpy-seal-grafana-7766755c7b-sv7wt | 0/1 | ContainerCreating | 0 | 1s |

```
==> v1/Role
```

NAME AGE
lumpy-seal-grafana-test 1s

==> v1/RoleBinding
NAME AGE
lumpy-seal-grafana-test 1s

==> v1/Secret
NAME TYPE DATA AGE
lumpy-seal-grafana Opaque 3 1s

==> v1/Service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
lumpy-seal-grafana LoadBalancer 10.100.94.239 <pending> 80:32701/TCP 1s

==> v1/ServiceAccount
NAME SECRETS AGE
lumpy-seal-grafana 1 1s
lumpy-seal-grafana-test 1 1s

==> v1beta1/PodSecurityPolicy
NAME PRIV CAPS SELINUX RUNASUSER FSGROUP SUPGROUP READONLYROOTFS VOLUMES
lumpy-seal-grafana false RunAsAny RunAsAny RunAsAny RunAsAny
false configMap,emptyDir,projected,secret,downwardAPI,persistentVolumeClaim
lumpy-seal-grafana-test false RunAsAny RunAsAny RunAsAny RunAsAny
false configMap,downwardAPI,emptyDir,projected,secret

==> v1beta1/Role
NAME AGE
lumpy-seal-grafana 1s

==> v1beta1/RoleBinding
NAME AGE
lumpy-seal-grafana 1s

NOTES:

1. Get your 'admin' user password by running:

```
kubectl get secret --namespace grafana lumpy-seal-grafana -o  
jsonpath="{.data.admin-password}" | base64 --decode ; echo
```

2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:

lumpy-seal-grafana.grafana.svc.cluster.local

Get the Grafana URL to visit by running these commands in the same shell:

NOTE: It may take a few minutes for the LoadBalancer IP to be available.

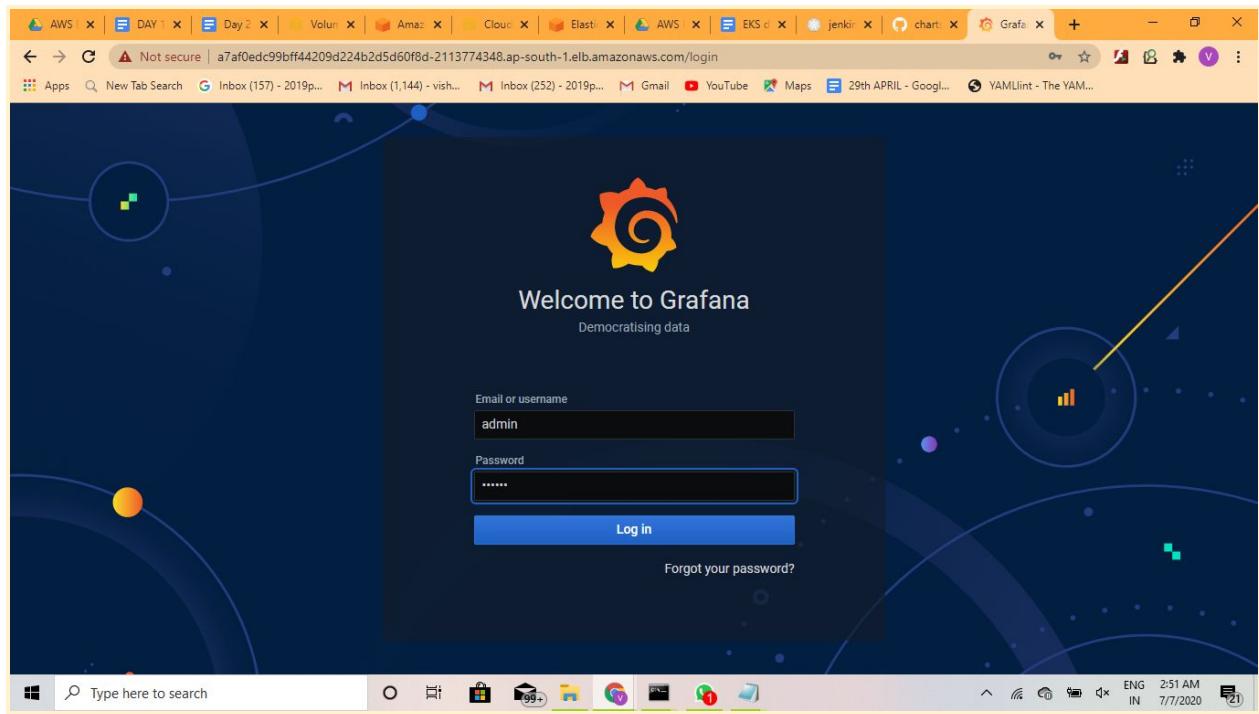
You can watch the status of by running 'kubectl get svc --namespace grafana -w lumpy-seal-grafana'

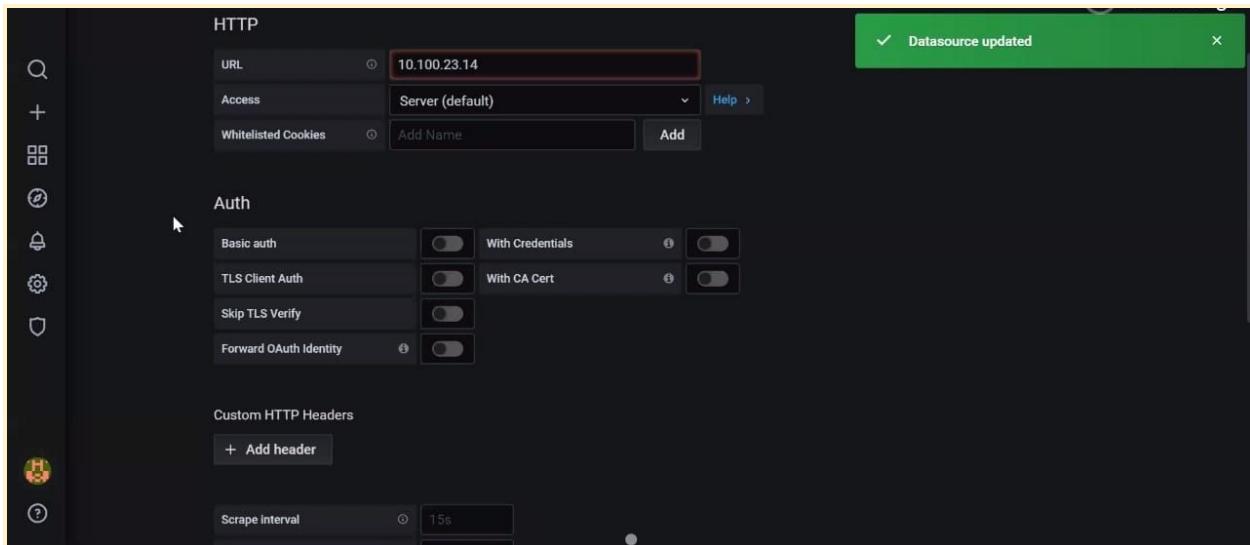
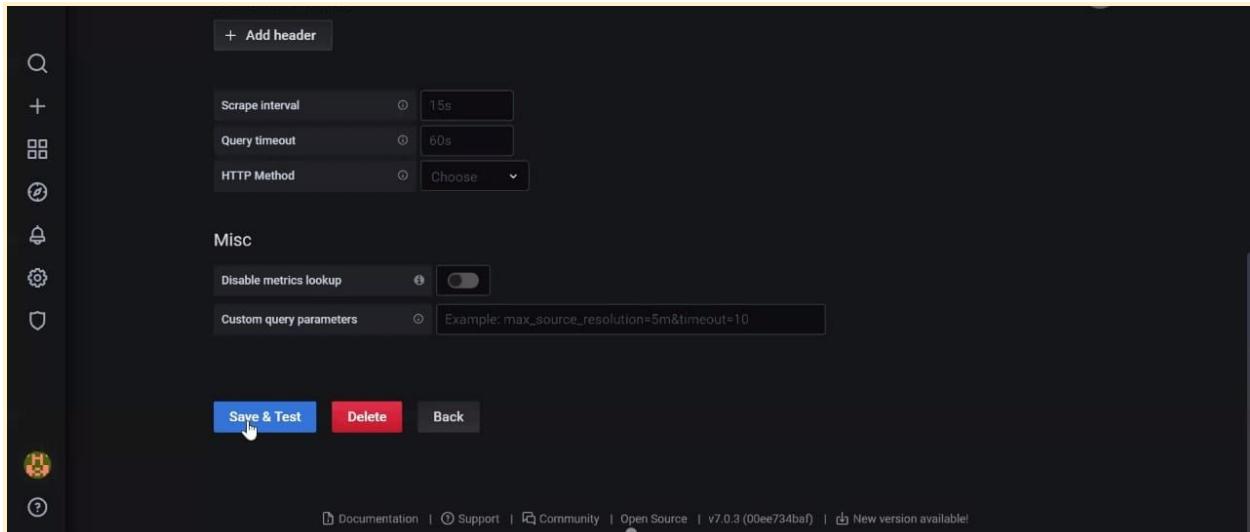
```
export SERVICE_IP=$(kubectl get svc --namespace grafana  
lumpy-seal-grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip}')  
http://$SERVICE_IP:80
```

3. Login with the password from step 1 and the username: admin

```
#####  
#####  
##### WARNING: Persistence is disabled!!! You will lose your data when  
#####  
#####           the Grafana pod is terminated.           #####  
#####  
#####
```

```
C:\Users\user\Desktop\eks cloud>  
C:\Users\user\Desktop\eks cloud>kubectl get pods -n grafana  
NAME          READY   STATUS    RESTARTS   AGE  
lumpy-seal-grafana-7766755c7b-sv7wt  1/1     Running   0      18s
```





Add Data Source

Choose a data source type

Filter by name or type

Cancel

Time series databases

- Prometheus**
Open source time series database & alerting
Core Core plugin that is bundled with Grafana
- Graphite**
Open source time series database
Core
- OpenTSDB**
Open source time series database
Core

Options

Name: Cluster Monitoring for Kubernetes

Folder: General

Unique identifier (uid): JABGX-mz

Prometheus: Prometheus

Import Cancel

Documentation | Support | Community | Open Source | v7.0.3 (00ee734ba) | New version available!

Import

Import dashboard from file or Grafana.com

Upload .json file

Import via grafana.com

10000 Load

Import via panel json

Data Sources / Prometheus
Type: Prometheus

Settings **Dashboards**

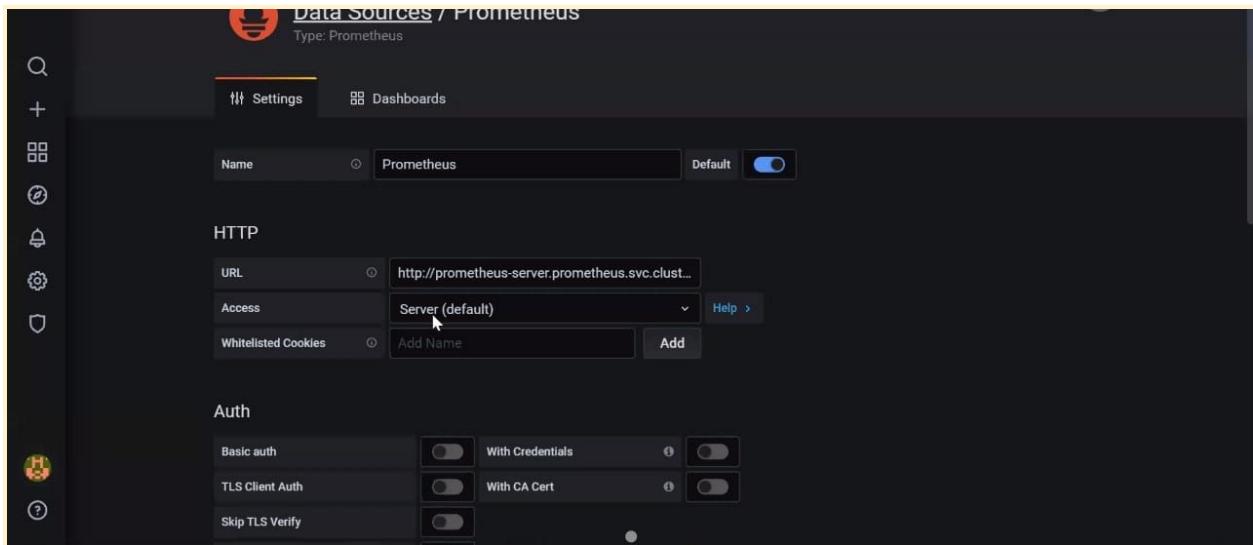
Name: **Prometheus** Default: **On**

HTTP

URL: **http://prometheus-server.prometheus.svc.clust...**
Access: **Server (default)** Help >
Whitelisted Cookies: **Add Name** Add

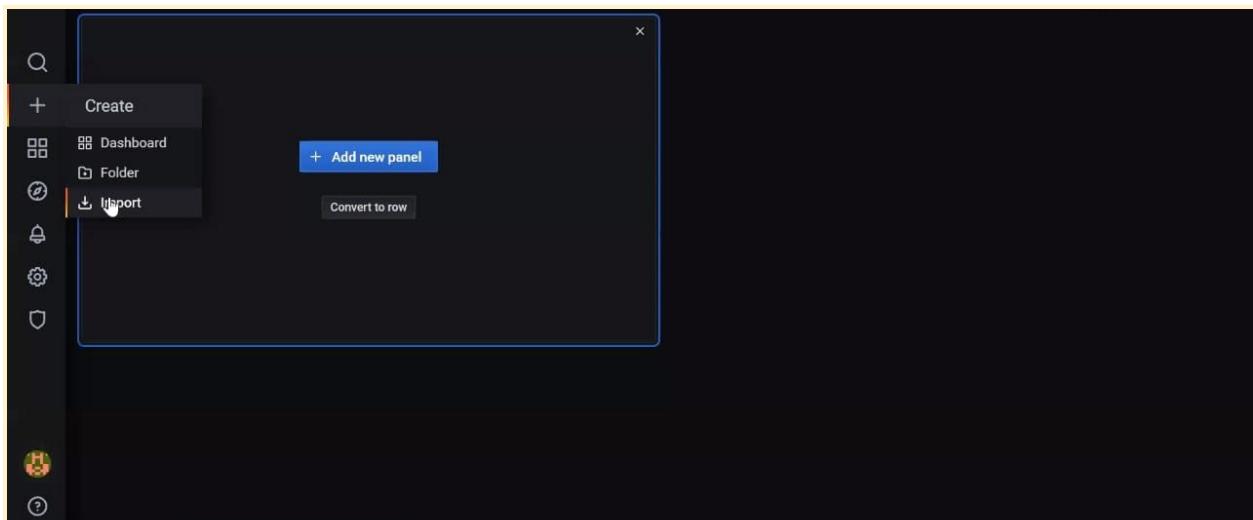
Auth

Basic auth: **On** With Credentials: **On**
TLS Client Auth: **On** With CA Cert: **On**
Skip TLS Verify: **On**



Create

+ **Dashboard** + **Add new panel**
Folder
Import
Convert to row



Add data source
Choose a data source type

Filter by name or type

Cancel

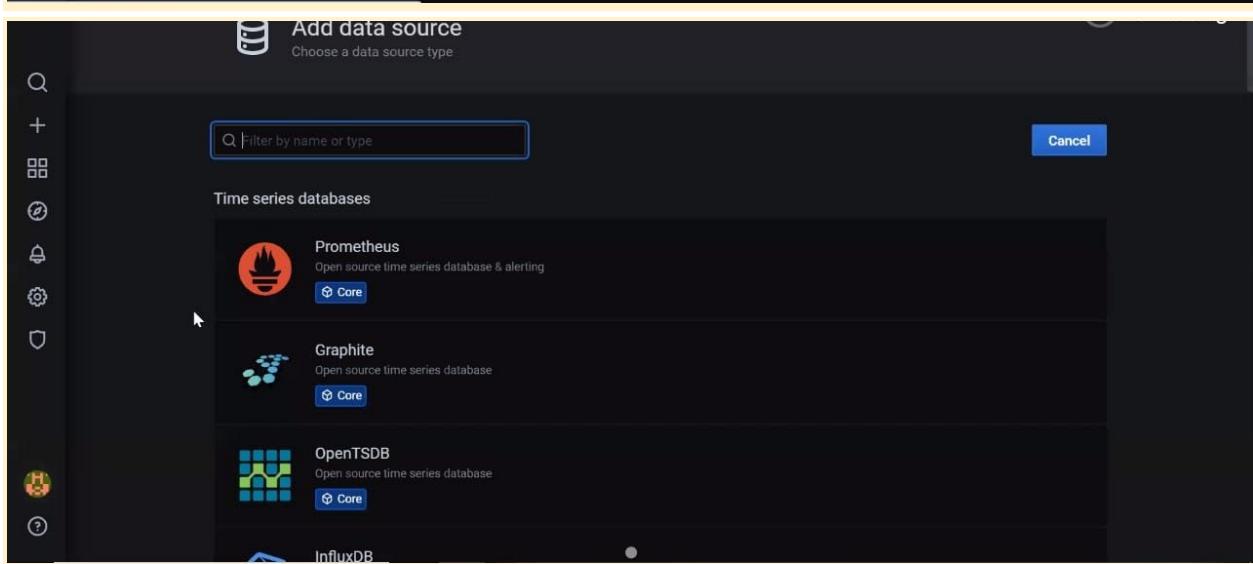
Time series databases

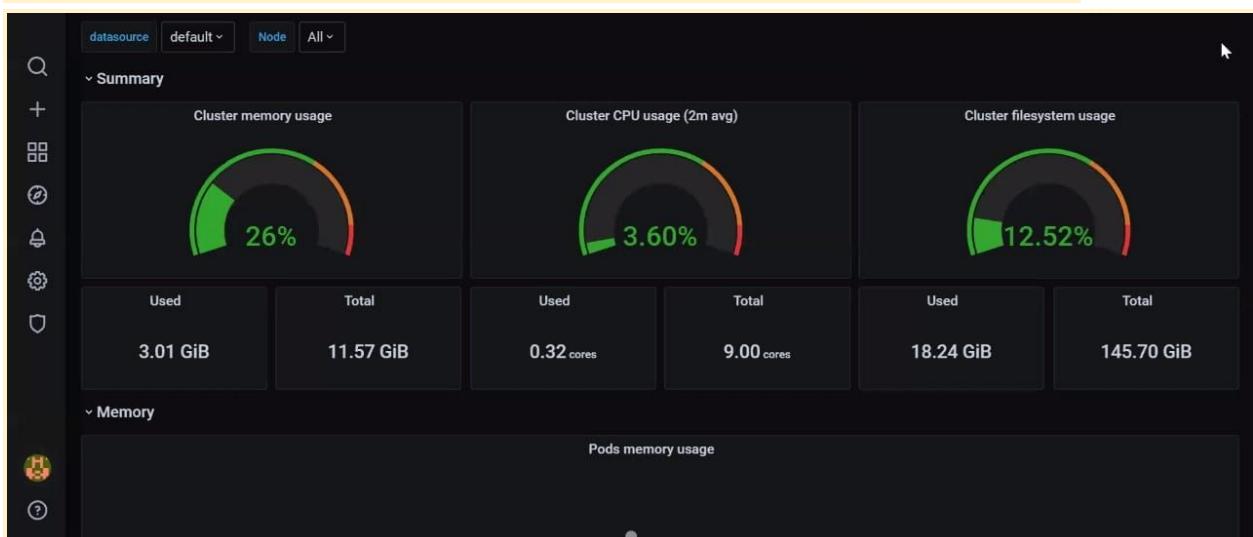
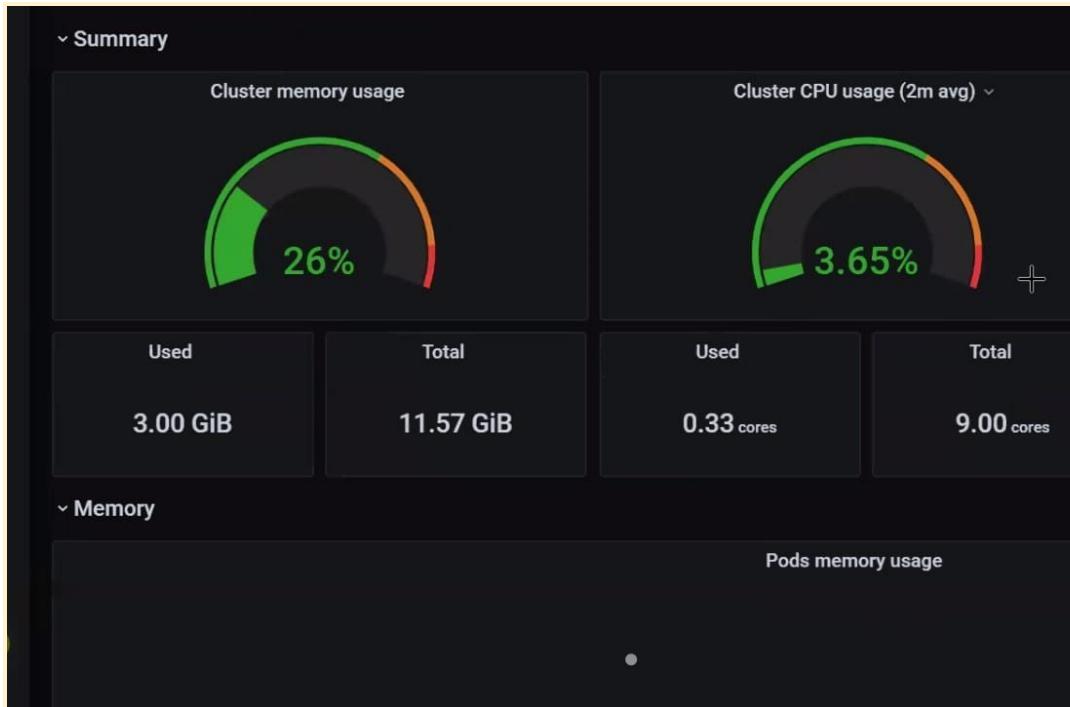
Prometheus
Open source time series database & alerting
Core

Graphite
Open source time series database
Core

OpenTSDB
Open source time series database
Core

InfluxDB





Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL DATA SOURCE AND DASHBOARDS

Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

DATA SOURCES Add your first data source

DASHBOARDS Create your first dashboard

Remove this panel

Learn how in the docs

Learn how in the docs

Latest from the blog

Getting started with the Grafana Cloud Agent, a remote_write

Jul 02

```
C:\Users\user\Desktop\eks cloud>kubectl describe
svc/unrealistic-liger-prometheus-server -n prometheus
Name:      unrealistic-liger-prometheus-server
Namespace:  prometheus
Labels:    app=prometheus
          chart=prometheus-11.6.1
          component=server
          heritage=Tiller
          release=unrealistic-liger
Annotations: <none>
Selector:   app=prometheus,component=server,release=unrealistic-liger
Type:      ClusterIP
IP:       10.100.122.133
Port:      http 80/TCP
```

TargetPort: 9090/TCP

Endpoints: <none>

Session Affinity: None

Events: <none>

The image consists of two screenshots of a web browser. The top screenshot shows a Google search results page for 'grafana dashboard'. The search bar contains 'grafana dashboard'. Below it, there are tabs for 'All', 'Images', 'Videos', 'News', 'Books', and 'More'. The main content area shows a snippet from the official Grafana website: 'About 10,90,000 results (0.34 seconds)'. It includes a link to 'grafana.com' and a brief description of what Grafana Dashboards are. A sidebar titled 'People also ask' lists four questions: 'What is a Grafana dashboard?', 'What is Grafana used for?', 'How do I access my Grafana dashboard?', and 'Is Grafana free to use?'. The bottom screenshot shows the Grafana Dashboard marketplace. The URL is 'grafana.com/grafana/dashboards?search=kubernetes&dataSource=prometheus'. The page has a navigation bar with 'Features', 'Contribute', 'Dashboards' (which is selected), 'Plugins', and 'Download'. On the left, there are filters for 'Name / Description' (set to 'kubernetes'), 'Data Source' (set to 'Prometheus'), 'Panel Type' (set to 'All'), 'Category' (set to 'All'), 'Collector' (set to 'All'), and 'Sort By' (set to 'Name'). The main content area displays four dashboards: 1. Kubernetes cluster overview (by dguyhasnoname), 1. Kubernetes Deployment Statefulset Daemonset metrics (by prat0318), 1. Kubernetes Deployment Statefulset Daemonset metrics (by Steven Beauge), and 1. Kubernetes Deployment Statefulset Daemonset metrics (by stupidhome836). Each listing includes a preview icon, the dashboard name, the author, download count, review count, and a star rating.

Screenshot of the Grafana plugin store on the Grafana website, showing various Kubernetes dashboards available for download.

The top navigation bar includes links for Features, Contribute, Dashboards (selected), Plugins, and Download.

The main content area displays five dashboards:

- Cluster cost & utilization metrics** by kubecost: A dashboard to help manage Kubernetes cluster costs and resources. Downloads: 33576, Reviews: 0.
- Cluster Monitoring for Kubernetes** by Pivotal Observability: This dashboard provides cluster admins with the ability to monitor nodes and identify workload bottlenecks. Downloads: 25819, Reviews: 4.
- Detailed Node Memory** by nitinjain999: Gives Detailed Node Memory Metrics of Kubernetes Cluster 1) Used Memory 2) SWAP 3) Cache 4)... Downloads: 359, Reviews: 0.
- Detailed pods resources** by fredgrnd: Get your Kubernetes pods resources usage by namespaces. Downloads: 2466, Reviews: 0.
- Easy RabbitMQ (K8S)** by nbigot: RabbitMQ stats (easy to use in kubernetes). Downloads: 1289, Reviews: 1.

The URL https://grafana.com/grafana/dashboards/10000 is highlighted in the address bar of the browser window.

The bottom section shows the "Cluster Monitoring for Kubernetes" dashboard details:

- Cluster Monitoring for Kubernetes** by Pivotal Observability
- DASHBOARD**
- This dashboard provides cluster admins with the ability to monitor nodes and identify workload bottlenecks. It can be deployed with PSPs enabled using the following helm chart - <https://github.com/pivotal-cf/charts-grafana>
- Last updated: a year ago
- Downloads: 25819**, **Reviews: 4**
- Add your review!**

Navigation tabs for the dashboard include Overview (selected), Revisions, and Reviews.

Get this dashboard options include a "10000" button and a "Copy ID to Clipboard" button.

Deployment information: This dashboard and config can be deployed via these [helm-charts](#).

Download options: Download JSON, How do I import this dashboard?

The screenshot shows two windows side-by-side. The left window is the AWS EFS console in the 'File systems' section, displaying a single file system named 'fs-68bc37b9'. The right window is a Notepad application showing a YAML configuration for an EFS provisioner.

AWS EFS Console (Left Window):

- Header:** AWS | DAY | Day | Vol. | Amaz | Cloud | Elasti | AWS | EKS | Jenkins | char | Inbo | Clus | +
- URL:** ap-south-1.console.aws.amazon.com/efs/home?region=ap-south-1#filesystems
- Services:** Services ▾ Resource Groups ▾
- File systems:** File systems, AWS DataSync ▾, AWS Backup ▾
- Alert:** Reduce your storage price to \$0.08/GB-month* with EFS Lifecycle Management. Learn more ↗
- Table:** Shows the file system 'fs-68bc37b9' with details: Name (fs-68bc37b9), File system ID (fs-68bc37b9), Metered size (6.0 KiB), Number of mount targets (3), and Creation date (07/06/2020, 20:42:07 UTC).
- Other details:** Owner ID (410914255776), File system state (Available), Performance mode (General Purpose), Throughput mode (Bursting), Encrypted (No), Lifecycle policy (None).
- Tags:** No tags added.

Notepad (Right Window):

```
File Edit Format View Help
type: Recreate
template:
  metadata:
    labels:
      app: efs-provisioner
spec:
  containers:
    - name: efs-provisioner
      image: quay.io/external_storage/efs-provisioner:v0.1.0
      env:
        - name: FILE_SYSTEM_ID
          value: fs-68bc37b9
        - name: AWS_REGION
          value: ap-south-1
        - name: PROVISIONER_NAME
          value: lw-course/aws-efs
  volumeMounts:
    - name: pv-volume
      mountPath: /persistentvolumes
  volumes:
    - name: pv-volume
      nfs:
        server: fs-c2e66d13.efs.ap-south-1.amazonaws.com
        path: /
```

The Notepad window also displays the same EFS lifecycle management alert and the AWS Storage Blog link as the EFS console.

```
create-efs-provisioner - Notepad
File Edit Format View Help
matchLabels:
  app: efs-provisioner
replicas: 1
strategy:
  type: Recreate
template:
  metadata:
    labels:
      app: efs-provisioner
spec:
  containers:
    - name: efs-provisioner
      image: quay.io/external_storage/efs-provisioner:v0.1.0
      env:
        - name: FILE_SYSTEM_ID
          value: fs-68bc37b9
        - name: AWS_REGION
          value: ap-southeast-1
        - name: PROVISIONER_NAME
          value: lw-course/aws-efs
      volumeMounts:
        - name: pv-volume
          mountPath: /persistentvolumes
  volumes:
    - name: pv-volume
  nfs:
    server: fs-c2e66d13.efs.ap-south-1.amazonaws.com
    path: /
```

Screenshot of AWS EFS console showing file system configuration and mount targets.

File system access

- DNS name: fs-68bc37b9.efs.ap-south-1.amazonaws.com
- Amazon EC2 mount instructions (from local VPC)
- Amazon EC2 mount instructions (across VPC peering connection)
- On-premises mount instructions

Mount targets

| VPC | Availability Zone | Subnet | IP address | Mount target ID | Network interface ID | Security groups | Mount target state |
|---|-------------------|---|-----------------|-----------------|-----------------------|----------------------------|--------------------|
| vpc-08435fe4542f1c12 - eksctl-vishcluster-cluster/VPC | ap-south-1c | subnet-04f341a3dfa5fc1a7 - eksctl-vishcluster-cluster/SubnetPrivateAPPSOUTH1C | 192.168.189.155 | fsmr-4ace189b | eni-071e1547a20da7ed6 | sg-0a5b180560f799dcdefault | Available |
| | ap-south-1b | subnet-02c316dba4d2fd30c - eksctl-vishcluster-cluster/SubnetPublicAPPSOUTH1B | 192.168.16.97 | fsmr-4cce189d | eni-02111728d61bbf794 | sg-0a5b180560f799dcdefault | Available |
| | ap-south-1a | subnet-07fab58f8549f60c9 - eksctl-vishcluster-cluster/SubnetPublicAPPSOUTH1A | 192.168.149.82 | fsmr-4cce189e | eni-02111728d61bbf795 | sg-0a5b180560f799dcdefault | Available |

Notepad Content:

```

*create-efs-provisioner - Notepad
File Edit Format View Help
matchLabels:
  app: efs-provisioner
replicas: 1
strategy:
  type: Recreate
template:
  metadata:
    labels:
      app: efs-provisioner
spec:
  containers:
    - name: efs-provisioner
      image: quay.io/external_storage/efs-provisioner:v0.1.0
      env:
        - name: FILE_SYSTEM_ID
          value: fs-68bc37b9
        - name: AWS_REGION
          value: ap-southeast-1
        - name: PROVISIONER_NAME
          value: lw-course/aws-efs
      volumeMounts:
        - name: pv-volume
          mountPath: /persistentvolumes
  volumes:
    - name: pv-volume
      nfs:
        server: fs-68bc37b9.efs.ap-south-1.amazonaws.com
        path: /

```

```
create-rbac - Notepad
File Edit Format View Help
---  
apiVersion: rbac.authorization.k8s.io/v1beta1  
kind: ClusterRoleBinding  
metadata:  
  name: nfs-provisioner-role-binding  
subjects:  
  - kind: ServiceAccount  
    name: default  
    namespace: lwns  
roleRef:  
  kind: ClusterRole  
  name: cluster-admin  
  apiGroup: rbac.authorization.k8s.io
```



```
create-storage - Notepad
File Edit Format View Help
---  
kind: StorageClass  
apiVersion: storage.k8s.io/v1  
metadata:  
  name: aws-efs  
provisioner: lw-course/aws-efs  
---  
kind: PersistentVolumeClaim  
apiVersion: v1  
metadata:  
  name: efs-wordpress  
  annotations:  
    volume.beta.kubernetes.io/storage-class: "aws-efs"  
spec:  
  accessModes:  
    - ReadWriteMany  
  resources:  
    requests:  
      storage: 10Gi  
---  
kind: PersistentVolumeClaim  
apiVersion: v1
```

```
create-storage - Notepad
File Edit Format View Help
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  ...
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: efs-mysql
  annotations:
    volume.beta.kubernetes.io/storage-class: "aws-efs"
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

C:\Users\user\Desktop\eks cloud>kubectl create ns lwns
namespace/lwns created

C:\Users\user\Desktop\eks cloud>kubectl create -f create-efs-provisioner.yml -n lwns
deployment.apps/efs-provisioner created

C:\Users\user\Desktop\eks cloud>kubectl create -f create-rbac.yml -n lwns
clusterrolebinding.rbac.authorization.k8s.io/nfs-provisioner-role-binding created

C:\Users\user\Desktop\eks cloud>kubectl create -f create-storage.yml -n lwns
storageclass.storage.k8s.io/aws-efs created

persistentvolumeclaim/efs-wordpress created

persistentvolumeclaim/efs-mysql created

C:\Users\user\Desktop\eks cloud>kubectl get sc -n lwns

| NAME | PROVISIONER | AGE |
|------|-------------|-----|
|------|-------------|-----|

| | | |
|---------|-------------------|-----|
| aws-efs | lw-course/aws-efs | 56s |
|---------|-------------------|-----|

C:\Users\user\Desktop\eks cloud>kubectl get pvc -n lwns

| NAME | STATUS | VOLUME | CAPACITY | ACCESS MODES |
|------|--------|--------|----------|--------------|
|------|--------|--------|----------|--------------|

| | |
|--------------|-----|
| STORAGECLASS | AGE |
|--------------|-----|

| | | | |
|-----------|---------|---------|-----|
| efs-mysql | Pending | aws-efs | 77s |
|-----------|---------|---------|-----|

| | | | |
|---------------|---------|---------|-----|
| efs-wordpress | Pending | aws-efs | 78s |
|---------------|---------|---------|-----|

```
C:\Users\user\Desktop\eks cloud>kubectl get pv -n lwns
```

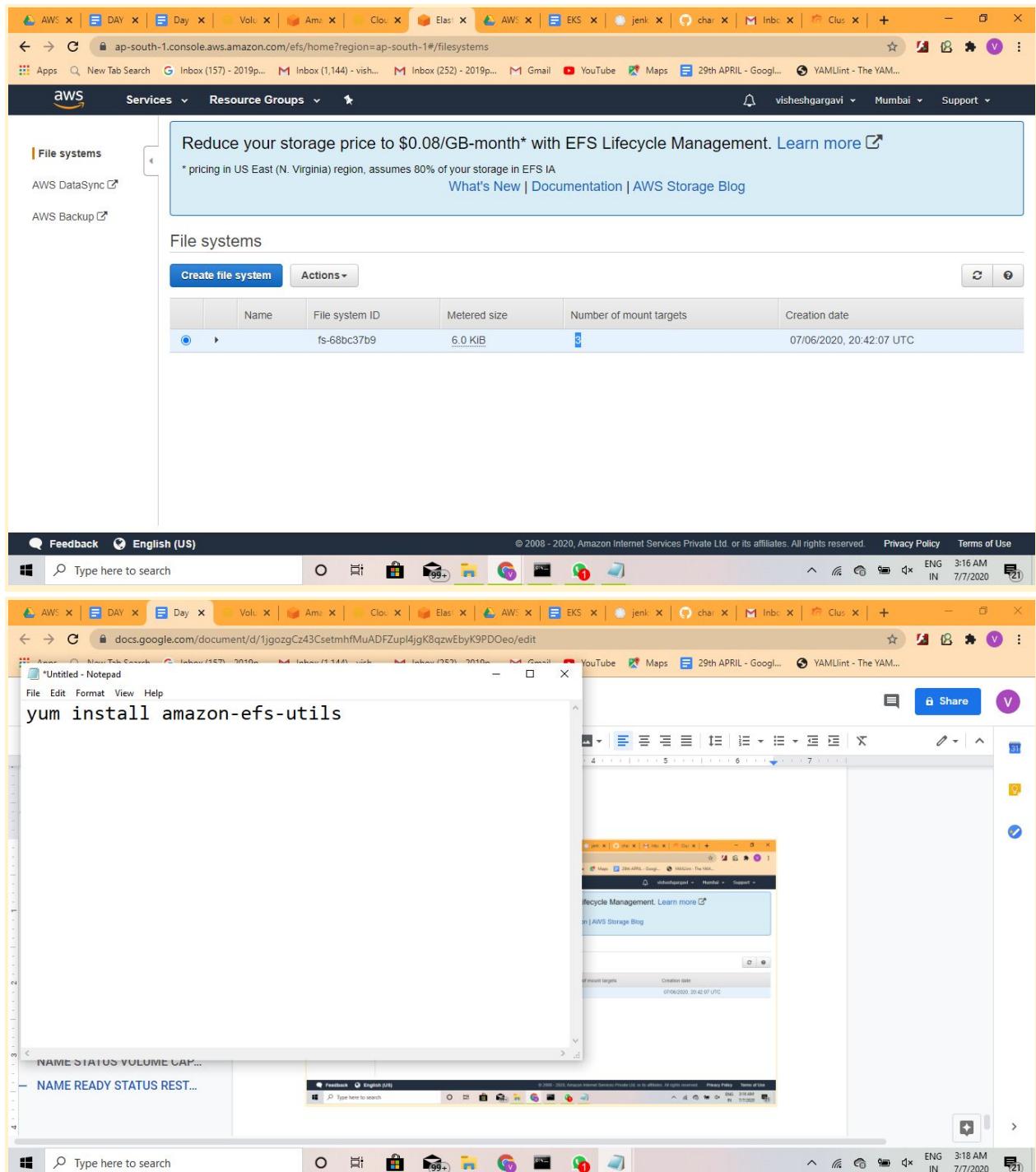
The screenshot shows a Google Sheets document with a terminal window open in a sidebar. The terminal output is as follows:

```
Day 2: [Select Command Prompt]
File Edit View Insert Format Tools Add-ons Help Last edit was 2 minutes ago
C:\Users\user\Desktop\eks cloud>dir
Volume in drive C is vishesh
Volume Serial Number is 1CF6-F84B
NAME AGE
NAME AGE
Directory of C:\Users\user\Desktop\eks cloud
07/07/2020 03:14 AM <DIR> .
07/07/2020 03:14 AM <DIR> ..
07/07/2020 12:56 AM 378 cluster1.yml
07/07/2020 03:05 AM 867 create-efs-provisioner.yml
07/07/2020 03:06 AM 297 create-rbac.yml
07/07/2020 03:07 AM 605 create-storage.yml
07/07/2020 03:13 AM 1,088 deploy-mysql.yml
07/07/2020 03:14 AM 1,185 deploy-wordpress.yml
07/07/2020 01:24 AM 0 eksctl
07/07/2020 01:50 AM 247 fargate.yml
07/05/2020 01:05 AM 10 index.php
NOTES: 07/06/2020 02:37 PM 204 kustomization.yml
06/19/2020 01:29 AM 1,670 mykey1111.pem
NAME READY STATUS RESTARTS AGE
07/07/2020 01:10 AM 1,382 mysql-deployment.yml
07/05/2020 01:30 AM 198 pvc1.yml
IP: 10.100.122.1 07/05/2020 01:29 AM 170 storage1.yml
07/07/2020 02:04 AM 1,463 wordpress-deployment.yml
NAME PROVISIONER 15 File(s) 9,684 bytes
NAME STATUS VOLUME 2 Dir(s) 140,223,918,080 bytes free
NAME STATUS VOLUME
C:\Users\user\Desktop\eks cloud>
```

```
C:\Users\user\Desktop\eks cloud>kubectl create -f deploy-mysql.yml -n lwns
service/wordpress-mysql created
deployment.apps/wordpress-mysql created
```

```
C:\Users\user\Desktop\eks cloud>kubectl create -f deploy-wordpress.yml -n lwns
service/wordpress created
deployment.apps/wordpress created
```

```
C:\Users\user\Desktop\eks cloud>kubectl get pods -n lwns
NAME READY STATUS RESTARTS AGE
efs-provisioner-7db9d6f9cf-hwxbs 0/1 ContainerCreating 0 5m9s
wordpress-659785fd8b-4478w 0/1 Pending 0 37s
wordpress-mysql-6c8fb78df4-fsptv 0/1 Pending 0 49s
```



```
C:\Users\user\Desktop\eks cloud>kubectl create secret generic mysql-pass
--from-literal=password=redhat
secret/mysql-pass created
C:\Users\user\Desktop\eks cloud>eksctl get cluster --region ap-southeast-1
NAME      REGION
f-lwcluster  ap-southeast-1
```

```
C:\Users\user\Desktop\eks cloud>aws eks --region ap-southeast-1  
update-kubeconfig --name f-lwcluster  
Added new context arn:aws:eks:ap-southeast-1:410914255776:cluster/f-lwcluster  
to C:\Users\user\.kube\config  
C:\Users\user\Desktop\eks cloud>eks config view  
'eks' is not recognized as an internal or external command,  
operable program or batch file.
```

```
C:\Users\user\Desktop\eks cloud>  
C:\Users\user\Desktop\eks cloud>kubectl get pods  
No resources found in default namespace.
```

```
C:\Users\user\Desktop\eks cloud>kubectl get ns  
NAME      STATUS AGE  
default    Active 79m  
kube-node-lease Active 79m  
kube-public Active 79m  
kube-system Active 79m
```

```
C:\Users\user\Desktop\eks cloud>kubectl get nodes  
NAME                  STATUS ROLES AGE VERSION  
fargate-ip-192-168-146-188.ap-southeast-1.compute.internal Ready <none>  
71m v1.16.8-eks-e16311  
fargate-ip-192-168-179-167.ap-southeast-1.compute.internal Ready <none>  
71m v1.16.8-eks-e16311
```

The screenshot shows the AWS EKS (Amazon Container Services) console. On the left, there's a sidebar with links for Amazon ECS (Clusters, Task definitions), Amazon EKS (Clusters), and Amazon ECR (Repositories). The main area displays a table titled "Clusters (1)".

| Cluster name | Kubernetes version | Status |
|--------------|--------------------|--------|
| f-lwcluster | 1.16 | Active |

At the bottom of the page, there are links for Privacy Policy and Terms of Use.

The screenshot shows the AWS EC2 Management Console. The left sidebar includes links for New EC2 Experience, EC2 Dashboard (Events, Tags, Reports, Limits), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts), and Images (AMIs). The main area displays a message: "You do not have any running instances in this region." It also includes links for Getting Started Guide and Launch Instance, and a search bar.

At the bottom of the page, there are links for Privacy Policy and Terms of Use.

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' page open. The left sidebar shows navigation options like EC2 Dashboard, Instances, and Images. The main content area displays a table titled 'Elastic IP addresses (1/1)'. The table has columns for Name, Public IPv4 address, Allocation ID, and Associated instance ID. One row is listed with the following values: Name (empty), Public IPv4 address (54.179.91.31), Allocation ID (eipalloc-0d5559aaa102ec480), and Associated instance ID (empty). There are buttons for 'Actions' and 'Allocate Elastic IP address' at the top right of the table.

```
C:\Users\user\Desktop\eks cloud>kubectl get nodes --all-namespaces
NAME STATUS ROLES AGE VERSION
fargate-ip-192-168-146-188.ap-southeast-1.compute.internal Ready <none>
72m v1.16.8-eks-e16311
fargate-ip-192-168-179-167.ap-southeast-1.compute.internal Ready <none>
72m v1.16.8-eks-e16311
```