

# **EKS Training at LinuxWorld Information Pvt Ltd.**

By: Shivangi Saxena

## **EKS**

Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane. Amazon EKS is certified Kubernetes conformant, so existing applications running on upstream Kubernetes are compatible with Amazon EKS.

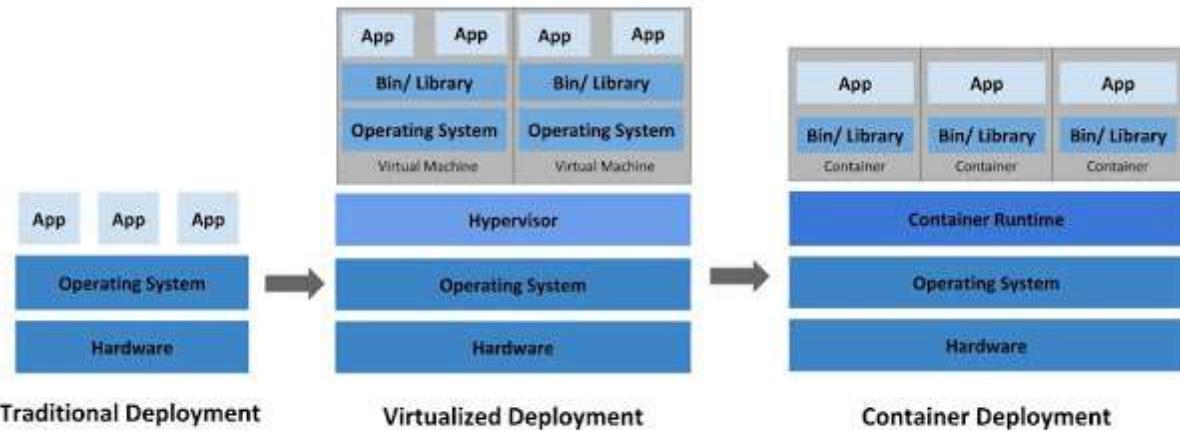
Amazon EKS automatically manages the availability and scalability of the Kubernetes control plane nodes that are responsible for starting and stopping containers, scheduling containers on virtual machines, storing cluster data, and other tasks. Amazon EKS automatically detects and replaces unhealthy control plane nodes for each cluster.

## **What is Kubernetes?**

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.



## TASK :

1. *Create Kubernetes Cluster using AWS EKS.*
2. *Integrate of EKS with Services like Ec2, EBS, EFS, LB.*
3. *Create a Multi-tier Architecture of Wordpress & MySQL on the top of EKS*
4. *Using Helm to install Prometheus and Grafana*
5. *Create a Server less Architecture by Fargate Cluster EKS(Elastic Kubernetes Services).*

## Steps :-

1. Create Kubernetes Cluster using AWS EKS

```
Administrator: Command Prompt - chocolatey install eksctl
Microsoft Windows [Version 10.0.18362.980]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>chocolatey install eksctl
Chocolatey v0.10.15
[Pending] Removing incomplete install for 'eksctl'
Installing the following packages:
eksctl
By installing you accept licenses for the packages.

eksctl v0.23.0 [Approved]
eksctl package files install completed. Performing other installation steps.
The package eksctl wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script? ([Y]es/[A]ll - yes to all/[N)o/[P]rint): yes

eksctl is going to be installed in 'C:\ProgramData\chocolatey\lib\eksctl\tools'
File appears to be downloaded already. Verifying with package checksum to determine if it needs to be redownloaded.
Error - hashes do not match. Actual value was 'CF757A205ABAEC2CFCF59645B7C3A4C0602B4C06FA1E774C65BED29D3396DE1E'.
Downloading eksctl 64 bit
  from 'https://github.com/weaveworks/eksctl/releases/download/0.23.0/eksctl_Windows_amd64.zip'
Progress: 64% - Saving 12.7 MB of 19.65 MB
```

```
C:\Users\USER\Desktop\eks_class_cloud>eksctl create cluster -f cluster.yml
[::] eksctl version 0.23.0
[::] using region ap-south-1
[::] setting availability zones to [ap-south-1c ap-south-1a ap-south-1b]
[::] subnets for ap-south-1c - public:192.168.0.0/19 private:192.168.96.0/19
[::] subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19
[::] subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19
[::] nodegroup "ng1" will use "ami-0f9be4bb82b6005cd" [AmazonLinux2/1.16]
[::] nodegroup "ng2" will use "ami-0f9be4bb82b6005cd" [AmazonLinux2/1.16]
[::] using Kubernetes version 1.16
[::] creating EKS cluster "shivangicloud" in "ap-south-1" region with un-managed nodes
[::] 2 nodegroups (ng1, ng2) were included (based on the include/exclude rules)
[::] will create a CloudFormation stack for cluster itself and 2 nodegroup stack(s)
[::] will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
[::] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1
--cluster=shivangicloud'
[::] CloudWatch logging will not be enabled for cluster "shivangicloud" in "ap-south-1"
[::] you can enable it with 'eksctl utils update-cluster-logging --region=ap-south-1 --cluster=shivangicloud'
[::] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "shivangicloud" in "ap-south-1"
[::] 2 sequential tasks: { create cluster control plane "shivangicloud", 2 sequential sub-tasks: { no tasks, 2 parallel
sub-tasks: { create nodegroup "ng1", create nodegroup "ng2" } } }
[::] building cluster stack "eksctl-shivangicloud-cluster"
[::] deploying stack "eksctl-shivangicloud-cluster"
```

The screenshot shows the AWS EKS service dashboard under the Amazon Container Services section. The main content area displays a table titled 'Clusters (1)'. It contains one row with the following columns: Cluster name (shivangicloud3), Kubernetes version (1.16), and Status (Active). A button labeled 'Update now' is present in the Kubernetes version column. A notification bar at the top states 'New Kubernetes versions are available for 1 cluster.'

Cluster name	Kubernetes version	Status
shivangicloud3	1.16 <a href="#">Update now</a>	Active

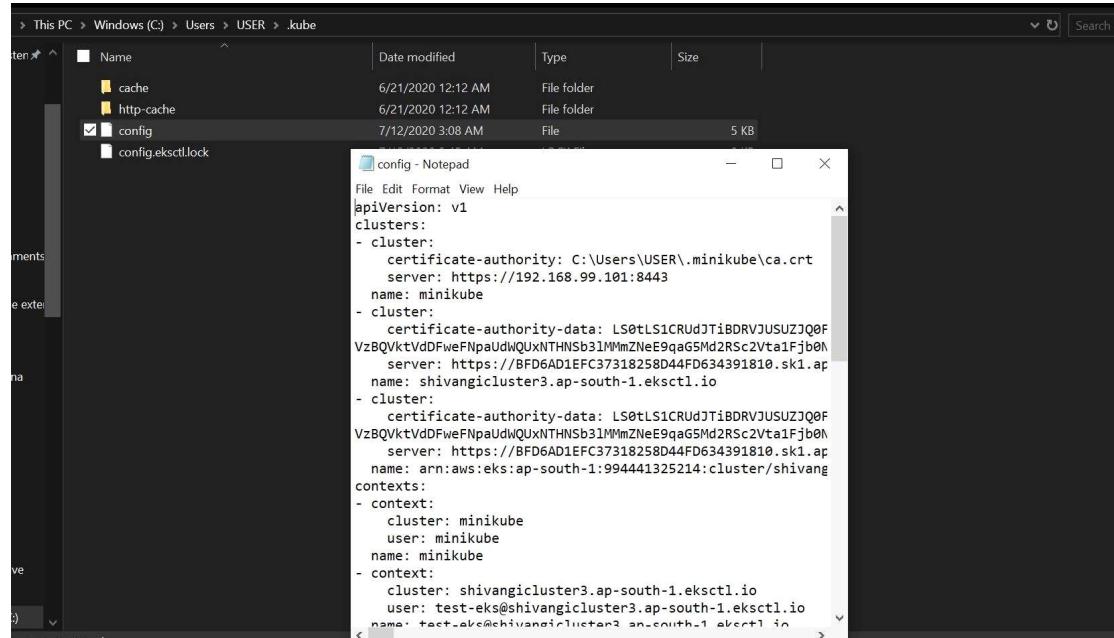
The screenshot shows the AWS EC2 service dashboard under the New EC2 Experience section. The main content area displays a table titled 'Instances (3)' showing three running instances. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. The instances are: shivangicloud3 (t2.micro, ap-south-1b, running, 2/2 checks, None, ec2-35-154-38), shivangicloud3 (t2.micro, ap-south-1a, running, 2/2 checks, None, ec2-13-232-24), and shivangicloud3 (t2.micro, ap-south-1a, running, 2/2 checks, None, ec2-13-233-34).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (I)
shivangicloud3	i-0b46fde9a966541ac	t2.micro	ap-south-1b	running	2/2 checks	None	ec2-35-154-38
shivangicloud3	i-0c9fa094f8f24898d	t2.micro	ap-south-1a	running	2/2 checks	None	ec2-13-232-24
shivangicloud3	i-0fe65b68ae29594d	t2.micro	ap-south-1a	running	2/2 checks	None	ec2-13-233-34

The screenshot shows a Windows Command Prompt window with the title 'C:\WINDOWS\system32\cmd.exe'. The command 'kubectl config view' is run, displaying the current configuration. The output includes sections for apiVersion, clusters, contexts, and users. Key details include the 'clusters' section listing a 'minikube' cluster and an 'eksctl.io' cluster, and the 'users' section listing 'minikube' and 'test-eks' users with their respective certificates and keys.

```
C:\Users\USER\.kube>kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: C:\Users\USER\.minikube\ca.crt
    server: https://192.168.99.101:8443
    name: minikube
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://BFD6AD1EFC37318258D44FD634391810.sk1.ap-south-1.eks.amazonaws.com
    name: shivangicloud3.ap-south-1.eksctl.io
contexts:
- context:
    cluster: minikube
    user: minikube
    name: minikube
- context:
    cluster: shivangicloud3.ap-south-1.eksctl.io
    user: test-eks@shivangicloud3.ap-south-1.eksctl.io
    name: test-eks@shivangicloud3.ap-south-1.eksctl.io
current-context: test-eks@shivangicloud3.ap-south-1.eksctl.io
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: C:\Users\USER\.minikube\profiles\minikube\client.crt
    client-key: C:\Users\USER\.minikube\profiles\minikube\client.key
- name: test-eks@shivangicloud3.ap-south-1.eksctl.io
```

```
C:\Users\USER\.kube>
C:\Users\USER\.kube>aws eks update-kubeconfig --name shivangicloud
Added new context arn:aws:eks:ap-south-1:994441325214:cluster/shivangicloud to C:\Users\USER\.kube\config
C:\Users\USER\.kube>
```



```
C:\WINDOWS\system32\cmd.exe
C:\Users\USER\Desktop\eks_class_cloud>eksctl get cluster
NAME      REGION
shivangicloud3    ap-south-1

C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods
No resources found in default namespace.

C:\Users\USER\Desktop\eks_class_cloud>kubectl get nodes
NAME           STATUS  ROLES   AGE   VERSION
ip-192-168-18-29.ap-south-1.compute.internal  Ready   <none>  28m   v1.16.12-eks-904af05
ip-192-168-67-182.ap-south-1.compute.internal  Ready   <none>  28m   v1.16.12-eks-904af05
ip-192-168-73-168.ap-south-1.compute.internal  Ready   <none>  27m   v1.16.12-eks-904af05

C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\WINDOWS\system32\cmd.exe
ip-192-168-67-182.ap-south-1.compute.internal    Ready    <none>    28m    v1.16.12-eks-904af05
ip-192-168-73-168.ap-south-1.compute.internal    Ready    <none>    27m    v1.16.12-eks-904af05

C:\Users\USER\Desktop\eks_class_cloud>kubectl describe nodes ip-192-168-67-182.ap-south-1.compute.internal
Name:           ip-192-168-67-182.ap-south-1.compute.internal
Roles:          <none>
Labels:         alpha.eksctl.io/cluster-name=shivangicloud3
                alpha.eksctl.io/instance-id=i-0c9fa094f8f24896d
                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-1a
                kubernetes.io/arch=amd64
                kubernetes.io/hostname=ip-192-168-67-182.ap-south-1.compute.internal
                kubernetes.io/os=linux
Annotations:   node.alpha.kubernetes.io/ttl: 0
                volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp: Sun, 12 Jul 2020 02:45:37 +0530
Taints:         <none>
Unschedulable:  false
Lease:
  HolderIdentity: ip-192-168-67-182.ap-south-1.compute.internal
  AcquireTime:    <unset>
  RenewTime:     Sun, 12 Jul 2020 03:16:28 +0530
Conditions:
```

```
C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>kubectl get ns
NAME      STATUS  AGE
default   Active  41m
kube-node-lease  Active  41m
kube-public  Active  41m
kube-system  Active  41m

C:\Users\USER\Desktop\eks_class_cloud>kubectl create namespace shivangi
namespace/shivangi created

C:\Users\USER\Desktop\eks_class_cloud>kubectl get ns
NAME      STATUS  AGE
default   Active  42m
kube-node-lease  Active  42m
kube-public  Active  42m
kube-system  Active  42m
shivangi   Active  14s

C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\USER\Desktop\eks_class_cloud>kubectl cluster-info
Kubernetes master is running at https://BF6AD1EFC37318258D44FD634391810.sk1.ap-south-1.eks.amazonaws.com
coreDNS is running at https://BF6AD1EFC37318258D44FD634391810.sk1.ap-south-1.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

C:\Users\USER\Desktop\eks_class_cloud>kubectl get ns
NAME      STATUS  AGE
default   Active  43m
kube-node-lease  Active  43m
kube-public  Active  43m
kube-system  Active  43m
shivangi   Active  107s

C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl create deployment myweb1 --image=vimal13/apache-webserver-php  
deployment.apps/myweb1 created
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods  
NAME READY STATUS RESTARTS AGE  
myweb1-7b885b4cbb-xx2hn 1/1 Running 0 30s
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods  
NAME READY STATUS RESTARTS AGE  
myweb1-7b885b4cbb-xx2hn 1/1 Running 0 30s  
  
C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods -o wide  
NAME READY STATUS RESTARTS AGE IP NODE  
N NOMINATED NODE READINESS GATES  
myweb1-7b885b4cbb-xx2hn 1/1 Running 0 5ms 192.168.94.151 ip-192-168-73-168.ap-south-1.compute.int  
ernal <none> <none>  
  
C:\Users\USER\Desktop\eks_class_cloud>kubectl scale deployment myweb1 --replicas=3  
deployment.apps/myweb1 scaled  
  
C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods  
NAME READY STATUS RESTARTS AGE  
myweb1-7b885b4cbb-xx2hn 1/1 Running 0 30s  
  
C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods -o wide  
NAME READY STATUS RESTARTS AGE IP NODE  
N NOMINATED NODE READINESS GATES  
myweb1-7b885b4cbb-xx2hn 1/1 Running 0 5ms 192.168.94.151 ip-192-168-73-168.ap-south-1.compute.int  
ernal <none> <none>  
  
C:\Users\USER\Desktop\eks_class_cloud>kubectl scale deployment myweb1 --replicas=3  
deployment.apps/myweb1 scaled  
  
C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods  
NAME READY STATUS RESTARTS AGE  
myweb1-7b885b4cbb-78t46 1/1 Running 0 40s  
myweb1-7b885b4cbb-xx2hn 1/1 Running 0 6m24s  
myweb1-7b885b4cbb-xzjgr 1/1 Running 0 40s  
  
C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl describe service/myweb1
Name:           myweb1
Namespace:      default
Labels:         app=myweb1
Annotations:    <none>
Selector:       app=myweb1
Type:          LoadBalancer
IP:            10.100.221.92
LoadBalancer Ingress: a9b59fb616b2e48b7bd0bfb868c2a38d-2006377750.ap-south-1.elb.amazonaws.com
Port:          <unset>  80/TCP
TargetPort:     80/TCP
NodePort:      <unset>  31647/TCP
Endpoints:     192.168.21.143:80,192.168.71.69:80,192.168.94.151:80
Session Affinity: None
External Traffic Policy: Cluster
Events:
  Type  Reason             Age   From           Message
  ----  -----            ----  --  -----
  Normal  EnsuringLoadBalancer  2m21s  service-controller  Ensuring load balancer
  Normal  EnsuredLoadBalancer  2m18s  service-controller  Ensured load balancer

C:\Users\USER\Desktop\eks_class_cloud>
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'New EC2 Experience' and links for 'EC2 Dashboard', 'Events', 'Tags', and 'Limits'. The main area has a search bar and a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. Three instances are listed:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
shivangiclust...	i-0b46fde9a906541ac	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-35-154-38
shivangiclust...	i-0c9fa094fb24896d	t2.micro	ap-south-1a	running	2/2 checks ...	None	ec2-13-232-24
shivangiclust...	i-0fe65b68ae29594d	t2.micro	ap-south-1a	running	2/2 checks ...	None	ec2-13-233-34

The screenshot shows the AWS Security Groups page. At the top, it says 'Security Groups (19) Info' and has a 'Create security group' button. Below is a search bar with 'eks' and a table with columns: Name, Security group..., Security grou..., and VPC ID. Four security groups are listed:

Name	Security group...	Security grou...	VPC ID
-	sg-042d31d2b26...	launch-wizard-6	vpc-19fce171
eks-cluster-sg...	sg-05b02f0fd429...	eks-cluster-sg-s...	vpc-04bc0487365b...
-	sg-05d8d039469...	launch-wizard-2	vpc-19fce171

EC2 > Security Groups

**Security Groups (4) Info**

<input type="checkbox"/>	Name	Security group ID	Security group description	VPC ID
<input type="checkbox"/>	eksctl-shivan...	sg-037cba8248c7...	eksctl-shivangicl...	vpc-04bc0487365b...
<input type="checkbox"/>	eksctl-shivan...	sg-060009b2455...	eksctl-shivangicl...	vpc-04bc0487365b...
<input type="checkbox"/>	eksctl-shivan...	sg-0bbc04d3a98...	eksctl-shivangicl...	vpc-04bc0487365b...

DWS Services Resource Groups Mumbai Support

**Elastic Block Store**

**Volumes**

Snapshots Lifecycle Manager

<input type="checkbox"/>	Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Z
<input type="checkbox"/>	vol-077d441...	80 GiB	gp2	240	snap-0115874...	July 12, 2020 at 2:4...	ap-south-1b	
<input type="checkbox"/>	vol-06bc083...	80 GiB	gp2	240	snap-0115874...	July 12, 2020 at 2:4...	ap-south-1a	
<input type="checkbox"/>	vol-0944327...	80 GiB	gp2	240	snap-0115874...	July 12, 2020 at 2:4...	ap-south-1a	

Watch Getting GitHub Chocolatey Chocolatey Google How to WhatsApp LAUNCH Slack EC2 Manager

Apps Princeton Lynda: Online Courses Disney Professional... Disney Imagination... gre prep (86) 1 Simple Exercise exercise Winter Internships...

aws Services Resource Groups Mumbai Support

**Security Groups** New

**Elastic IPs** New

**Placement Groups** New

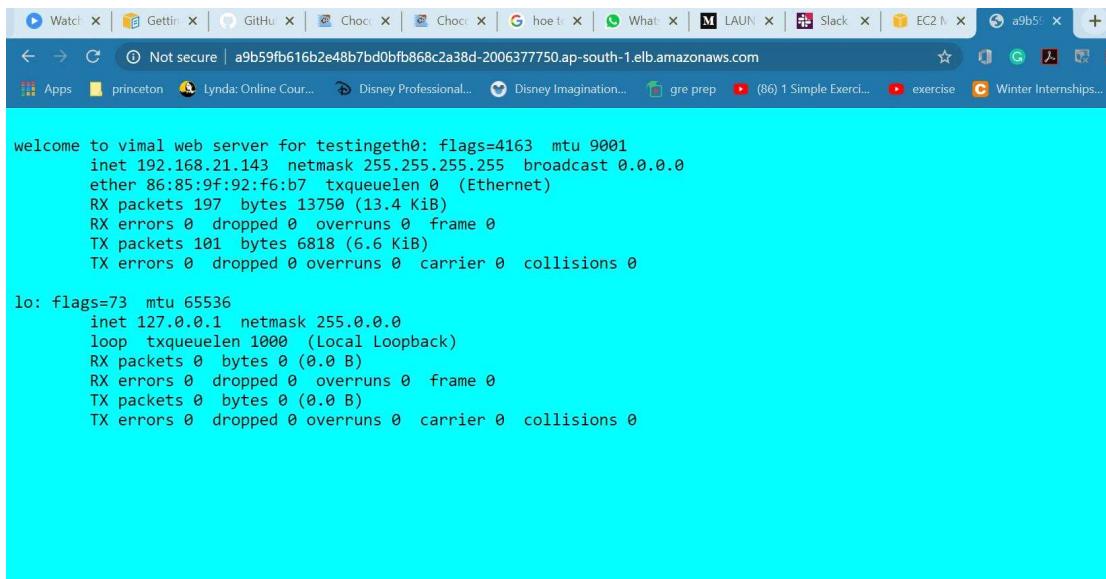
**Key Pairs** New

**Network Interfaces**

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Z
<input type="checkbox"/>	a9b59fb616b2e48b7bd0fbf8...	a9b59fb616b2e48b7bd0fbf8...	available	vpc-04bc0487365b59dc2	ap-south-1a

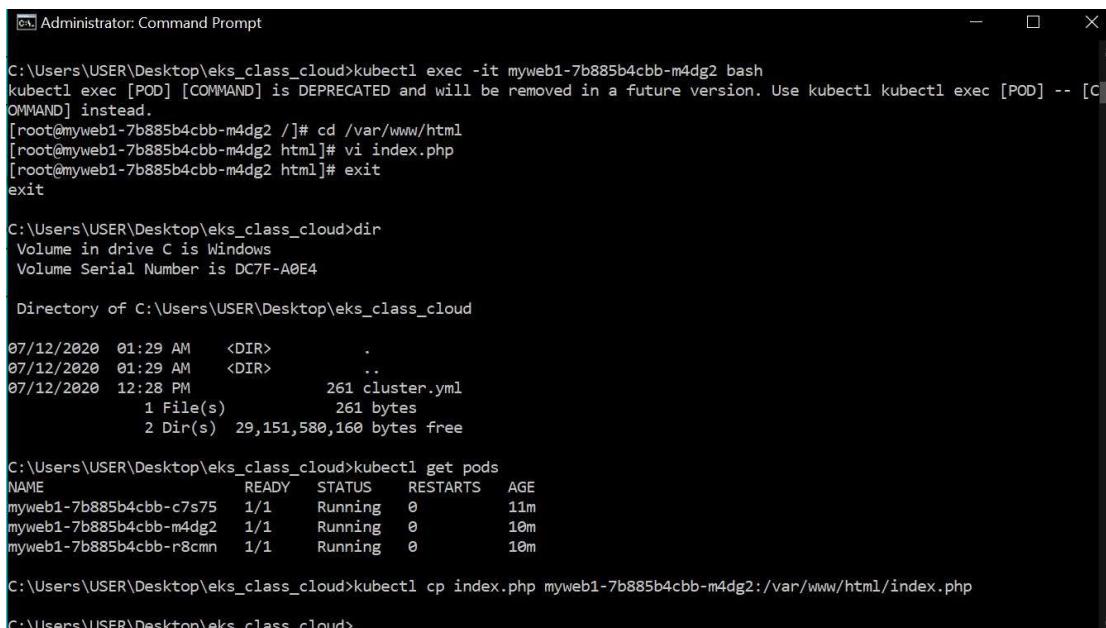
Load balancer: a9b59fb616b2e48b7bd0fbf868c2a38d

Description Instances Health check Listeners Monitoring Tags Migration



welcome to vimal web server for testingeth0: flags=4163 mtu 9001  
inet 192.168.21.143 netmask 255.255.255.255 broadcast 0.0.0.0  
ether 86:85:9f:92:f6:b7 txqueuelen 0 (Ethernet)  
RX packets 197 bytes 13750 (13.4 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 101 bytes 6818 (6.6 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
loop txqueuelen 1000 (Local Loopback)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0



```
C:\Users\USER\Desktop\eks_class_cloud>kubectl exec -it myweb1-7b885b4cbb-m4dg2 bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl kubectl exec [POD] -- [COMMAND] instead.
[root@myweb1-7b885b4cbb-m4dg2 /]# cd /var/www/html
[root@myweb1-7b885b4cbb-m4dg2 html]# vi index.php
[root@myweb1-7b885b4cbb-m4dg2 html]# exit
exit

C:\Users\USER\Desktop\eks_class_cloud>dir
 Volume in drive C is Windows
 Volume Serial Number is DC7F-A0E4

 Directory of C:\Users\USER\Desktop\eks_class_cloud

07/12/2020  01:29 AM    <DIR>      .
07/12/2020  01:29 AM    <DIR>      ..
07/12/2020  12:28 PM           261 cluster.yml
               1 File(s)      261 bytes
               2 Dir(s)  29,151,580,160 bytes free

C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myweb1-7b885b4cbb-c7s75  1/1     Running   0          11m
myweb1-7b885b4cbb-m4dg2  1/1     Running   0          10m
myweb1-7b885b4cbb-r8cmn  1/1     Running   0          10m

C:\Users\USER\Desktop\eks_class_cloud>kubectl cp index.php myweb1-7b885b4cbb-m4dg2:/var/www/html/index.php
C:\Users\USER\Desktop\eks_class_cloud>
```

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

C:\Users\USER\Desktop\eks_class_cloud>kubectl describe sc gp2
Name:          gp2
IsDefaultClass: Yes
Annotations:   kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"storage.k8s.io/v1","kind":"StorageClass","metadata":{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"},"name":"gp2"},"parameters":{"fsType":"ext4","type":"gp2"}},provisioner:"kubernetes.io/aws-ebs",volumeBindingMode:"WaitForFirstConsumer"},storageclass.kubernetes.io/is-default-class=true
Provisioner:    kubernetes.io/aws-ebs
Parameters:    fsType=ext4,type=gp2
AllowVolumeExpansion: <unset>
MountOptions:   <none>
ReclaimPolicy: Delete
VolumeBindingMode: WaitForFirstConsumer
Events:        <none>

C:\Users\USER\Desktop\eks_class_cloud>
```

```
Administrator: Command Prompt

C:\Users\USER\Desktop\eks_class_cloud>kubectl get sc gp2
NAME      PROVISIONER      AGE
gp2 (default)  kubernetes.io/aws-ebs  50m

C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>kubectl create -f sc.yml
storageclass.storage.k8s.io/lwsc1 created

C:\Users\USER\Desktop\eks_class_cloud>kubectl create -f pvc.yml
persistentvolumeclaim/lwpvc1 created

C:\Users\USER\Desktop\eks_class_cloud>
```

## 2. Integrate of EKS with Services like Ec2, EBS, EFS, LB

```
Administrator: Command Prompt

C:\Users\USER\Desktop\eks_class_cloud>kubectl get sc gp2
NAME      PROVISIONER      AGE
gp2 (default)  kubernetes.io/aws-ebs  50m

C:\Users\USER\Desktop\eks_class_cloud>sc1.yml - Notepad
File Edit Format View Help
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: lwsc123
  provisioner: kubernetes.io/aws-ebs
  parameters:
    type: io1
  reclaimPolicy: Retain
The StorageClass "lwsc123" has invalid parameters: "type": "io1" must be either case alphanumeric or a valid regex, and the regex used for validation must match the provided value.
C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl create -f sc1.yml
storageclass.storage.k8s.io/lwsc123 created

C:\Users\USER\Desktop\eks_class_cloud>kubectl get sc1 gp2
error: the server doesn't have a resource type "sc1"

C:\Users\USER\Desktop\eks_class_cloud>kubectl get sc gp2
NAME      PROVISIONER      AGE
gp2 (default)  kubernetes.io/aws-ebs  59m

C:\Users\USER\Desktop\eks_class_cloud>kubectl get pvc
NAME    STATUS    VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  AGE
lwpvc1  Bound    pvc-c6ea591d-ae9d-498b-8114-af27e849b99a  10Gi     RWO           lwsc1        10m

C:\Users\USER\Desktop\eks_class_cloud>kubectl get pv
NAME    REASON    AGE
pvc-c6ea591d-ae9d-498b-8114-af27e849b99a  10Gi     RWO
                                                Retain
                                                Bound
                                                default/lwpvc1  lwsc1
                                                10m

C:\Users\USER\Desktop\eks_class_cloud>kubectl get sc
NAME      PROVISIONER      AGE
gp2 (default)  kubernetes.io/aws-ebs  61m
lwsc1      kubernetes.io/aws-ebs  10m
lwsc123    kubernetes.io/aws-ebs  2m42s

C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\Users\USER\Desktop\eks_class_cloud>kubectl create -f sc1.yml
storageclass.storage.k8s.io/shivangi1 created

C:\Users\USER\Desktop\eks_class_cloud>kubectl get sc
NAME      PROVISIONER      AGE
gp2 (default)  kubernetes.io/aws-ebs  66m
lwsc1      kubernetes.io/aws-ebs  16m
lwsc123    kubernetes.io/aws-ebs  7m51s
shivangi1    kubernetes.io/aws-ebs  7s
shivangijsc1  kubernetes.io/aws-ebs  116s

C:\Users\USER\Desktop\eks_class_cloud>
```

```
C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>aws configure
AWS Access Key ID [*****DX2S]: AKIA6PCKSNKPFQ6YDX2S
AWS Secret Access Key [*****Zcr6]: NnuRNuuZJ/wuXpIjuF80TiZFif5iFLQWHZXZcr6
Default region name [ap-south-1]:
Default output format [None]

C:\Users\USER\Desktop\eks_class_cloud>eksctl get cluster
NAME      REGION
shivangicloud4  ap-south-1

C:\Users\USER\Desktop\eks_class_cloud>eksctl get nodegroup --cluster shivangicloud4
CLUSTER      NODEGROUP      CREATED      MIN SIZE      MAX SIZE      DESIRED CAPACITY
INSTANCE TYPE  IMAGE ID
shivangicloud4  ng1      2020-07-12T07:11:14Z  2            2            2
                t2.micro  ami-0f9be4bb82b6005cd
shivangicloud4  ng2      2020-07-12T07:11:14Z  1            1            1
                t2.micro  ami-0f9be4bb82b6005cd

C:\Users\USER\Desktop\eks_class_cloud>
```

```

Administrator: Command Prompt

C:\Users\USER\Desktop\eks_class_cloud>eksctl scale nodegroup --cluster shivangicloud --name ng2 --nodes=3
[!] scaling nodegroup stack "eksctl-shivangicloud-nodegroup-ng2" in cluster eksctl-shivangicloud-cluster
[!] the desired nodes 3 is greater than current nodes-max/maxSize 1
Error: failed to scale nodegroup for cluster "shivangicloud", error the desired nodes 3 is greater than current nodes -max/maxSize 1

C:\Users\USER\Desktop\eks_class_cloud>eksctl get nodegroup --cluster shivangicloud
CLUSTER          NODEGROUP      CREATED           MIN SIZE    MAX SIZE   DESIRED CAPACITY
INSTANCE TYPE    IMAGE ID
shivangicloud    ng1           2020-07-12T07:11:14Z  2           2           2
t2.micro         ami-0f9be4bb82b6005cd
shivangicloud    ng2           2020-07-12T07:11:14Z  1           1           1
t2.micro         ami-0f9be4bb82b6005cd

C:\Users\USER\Desktop\eks_class_cloud>eksctl scale nodegroup --cluster shivangicloud --name ng2 --nodes=3 --nodes-max=5
[!] scaling nodegroup stack "eksctl-shivangicloud-nodegroup-ng2" in cluster eksctl-shivangicloud-cluster
[!] scaling nodegroup, desired capacity from 1 to 3, max size from 1 to 5

C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>
C:\Users\USER\Desktop\eks_class_cloud>eksctl scale nodegroup --cluster shivangicloud --name ng2 --nodes=3 --nodes-max=5
[!] scaling nodegroup stack "eksctl-shivangicloud-nodegroup-ng2" in cluster eksctl-shivangicloud-cluster
[!] no change for nodegroup "ng2" in cluster "eksctl-shivangicloud-cluster": nodes-min 1, desired 3, nodes-max 5

C:\Users\USER\Desktop\eks_class_cloud>
```

```

Administrator: Command Prompt - eksctl delete cluster -f cluster.yml

ssh: connect to host 13.234.67.226 port 22: Connection timed out

C:\Users\USER\Desktop\eks_class_cloud>kubectl edit sc gp2

C:\Users\USER\Desktop\eks_class_cloud>eksctl delete cluster -f cluster.yml
[!] eksctl version 0.23.0
[!] using region ap-south-1
[!] deleting EKS cluster "shivangicloud"
[!] either account is not authorized to use Fargate or region ap-south-1 is not supported. Ignoring error
[!] kubeconfig has been updated
[!] cleaning up LoadBalancer services
[!] 2 sequential tasks: { 2 parallel sub-tasks: { delete nodegroup "ng2", delete nodegroup "ng1" }, delete
role plane "shivangicloud4" [async] }
[!] will delete stack "eksctl-shivangicloud-nodegroup-nginx"
[!] waiting for stack "eksctl-shivangicloud-nodegroup-nginx" to get deleted
[!] will delete stack "eksctl-shivangicloud-nodegroup-ng2"
[!] waiting for stack "eksctl-shivangicloud-nodegroup-ng2" to get deleted
```

```

ec2-user@ip-192-168-55-220:~ 
C:\Users\USER\Desktop\eks_class_cloud>eksctl get nodegroup --cluster shivangicloud
CLUSTER          NODEGROUP      CREATED           MIN SIZE    MAX SIZE   DESIRED CAPACITY   INSTANCE TYPE    IMAGE ID
shivangicloud    ng-mixed     2020-07-12T09:17:36Z  2           5           0           t3.small      ami-0f9be4bb82b6005cd
shivangicloud    ng1          2020-07-12T09:17:36Z  2           2           2           t2.micro      ami-0f9be4bb82b6005cd
shivangicloud    ng2          2020-07-12T09:17:36Z  1           5           3           t2.small      ami-0f9be4bb82b6005cd

C:\Users\USER\Desktop\eks_class_cloud>ssh -i mykey11.pem -i ec2-user 35.154.225.96
The authenticity of host '35.154.225.96' (35.154.225.96) can't be established.
EDSA key fingerprint is SHA256:bo3Yd8Ht2Th/o8%OLEpxA4nidede4b8pnYFKLZujl1o0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.154.225.96' (EDSA) to the list of known hosts.
Last login: Thu Jul  9 05:47:55 2020 from 205.251.233.48
[ec2-user@ip-192-168-55-220 ~]$ 

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-55-220 ~]$
```

```

root@ip-192-168-55-220:~#
C:\Users\USER\Desktop\eks_class_cloud>ssh -i mykey11.pem -l ec2-user 35.154.225.96
The authenticity of host '35.154.225.96' (35.154.225.96) can't be established.
EDSA key fingerprint is SHA256:b03Vd8htI7h/o8M0EpXAAnid0Hb8pnYFKLZUj1l00.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.154.225.96' (EDSA) to the list of known hosts.
Last login: Thu Jul  9 05:47:55 2020 from 205.251.233.48
[ec2-user@ip-192-168-55-220 ~]$ sudo su -
[root@ip-192-168-55-220 ~]# free -m
              total        used       free  buff/cache  available
Mem:      1955         349        391          0       1414       1456
Swap:          0          0          0
[root@ip-192-168-55-220 ~]# 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 Sun 2020-07-12 09:21:19 UTC; 1h 0min ago
       Docs: https://docs.docker.com
      Main PID: 3194 (dockerd)
        Tasks: 13
         Memory: 967.4M
        CGroup: /system.slice/docker.service
                  └─3194 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 12 09:21:18 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:18.204557281Z" level=info msg="cResolverWrapper: sending update...legRPC"
Jul 12 09:21:18 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:18.204572756Z" level=info msg="ClientConn switching balancer t...legRPC"
Jul 12 09:21:18 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:18.249721474Z" level=info msg="Loading containers: start."
Jul 12 09:21:18 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:18.739647787Z" level=info msg="Loading containers: done."
Jul 12 09:21:19 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:19.589689646Z" level=info msg="Docker daemon" commit=369ce74 g...03.6-ce
Jul 12 09:21:19 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:19.591696896Z" level=info msg="Daemon has completed initialization"
Jul 12 09:21:19 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:19.64239422Z" level=info msg="Started Docker Application Container Engine.
Jul 12 09:21:19 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:21:19.64239422Z" level=info msg="API listen on /var/run/docker.sock"
Jul 12 09:23:46 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:23:46.941650450Z" level=warning msg="Published ports are discarded...k mode"
Jul 12 09:23:47 ip-192-168-55-220.ap.south-1.compute.internal dockerd[3194]: time="2020-07-12T09:23:47.072744965Z" level=warning msg="Published ports are discarded...k mode"
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-192-168-55-220 ~]#

```

```

root@ip-192-168-55-220:~#
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-192-168-55-220 ~]# ifconfig
eni47c8270ad43: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
    inet6 fe80::8ca9:a2ff:fe5:f16b  prefixlen 64  scopeid 0x20<link>
        ether 8e:a9:a2:f5:f1:6b  txqueuelen 0  (Ethernet)
        RX packets 657 bytes 45826 (43.9 Kib)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1279 bytes 86194 (84.1 Kib)
        TX errors 0 dropped 0 overruns 0 carrier 0  collisions 0

eni827eb090f3c: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
    inet6 fe80::88dd:faff:feff:73f1  prefixlen 64  scopeid 0x20<link>
        ether ba:0d:f8:f7:73:f1  txqueuelen 0  (Ethernet)
        RX packets 1368 bytes 93978 (91.7 Kib)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2660 bytes 179814 (174.8 Kib)
        TX errors 0 dropped 0 overruns 0 carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
    inet 192.168.55.220  netmask 255.255.255.0  broadcast 192.168.63.255
    inet6 fe80::a5:81ff:fe35:1e60  prefixlen 64  scopeid 0x20<link>
        ether 02:a5:81:35:1e:60  txqueuelen 1000  (Ethernet)
        RX packets 287473 bytes 365484680 (348.5 MiB)

```

```

C:\Users\USER\Desktop\eks_class_cloud>kubectl get pods -n kube-system
NAME           READY   STATUS    RESTARTS   AGE
aws-node-24qv8  1/1    Running   0          69m
aws-node-4r5bx  1/1    Running   0          61m
aws-node-9nm9l  1/1    Running   0          69m
aws-node-jtxbs  1/1    Running   0          71m
aws-node-q5gjt  1/1    Running   0          70m
aws-node-tkwcb  1/1    Running   0          60m
aws-node-wwlq9  1/1    Running   0          71m
coredns-6856799b8d-9qkt8  1/1    Running   0          79m
coredns-6856799b8d-swj24  1/1    Running   0          79m
kube-proxy-4smml  1/1    Running   0          69m
kube-proxy-82v2h  1/1    Running   0          69m
kube-proxy-fdbkm  1/1    Running   0          71m
kube-proxy-h2pgd  1/1    Running   0          71m
kube-proxy-k5smn  1/1    Running   0          61m
kube-proxy-m452m  1/1    Running   0          60m
kube-proxy-s9czf  1/1    Running   0          70m

C:\Users\USER\Desktop\eks_class_cloud>

```

### 3. Create a Multi-tier Architecture of Wordpress & MySQL on the top of EKS

#### WordPress

It is the simplest, most popular way to create your own website or blog.

WordPress is an open-source content management system licensed under GPLv2

which means that anyone can use or modify the WordPress software for free.

#### MySQL

MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.

The most common use for MySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store.

#### EFS

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.

Amazon EFS offers two storage classes: the Standard storage class, and the [Infrequent Access storage class](#) (EFS IA). EFS IA provides price/performance that's cost-optimized for files not accessed every day. By simply enabling EFS Lifecycle Management on your file system, files not accessed according to the lifecycle policy you choose will be automatically and transparently moved into EFS IA. The EFS IA storage class costs only \$0.025/GB-month\*.

```
C:\Administrator: Command Prompt

C:\Users\USER\Desktop>cd kube_cloud

C:\Users\USER\Desktop\kube_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

C:\Users\USER\Desktop\kube_cloud>
```

```
C:\Administrator: Command Prompt

persistentvolumeclaim/wp-pv-claim created

C:\Users\USER\Desktop\kube_cloud>kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
myweb1-7b885b4ccb-5lqnq   1/1    Running   0          76m
myweb1-7b885b4ccb-crgqr   1/1    Running   0          77m
myweb1-7b885b4ccb-kbdx4   1/1    Running   0          76m
wordpress-88cb86b9b-h51sv  1/1    Running   0          75s
wordpress-mysql-66b4cc9ccb-69khq  1/1    Running   0          75s

C:\Users\USER\Desktop\kube_cloud>kubectl get secret
NAME          TYPE           DATA   AGE
default-token-qpblx  kubernetes.io/service-account-token  3      106m
mysql-pass-ctm2f4889c  Opaque          1      84s

C:\Users\USER\Desktop\kube_cloud>kubectl get pvc
NAME        STATUS   VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  AGE
mysql-pv-claim  Bound   pvc-1b60d3f0-55ce-48c6-af33-4cdf5cfa8095  20Gi       RWO          gp2          98s
shivipvc1     Bound   pvc-dfcf2c7b-b7c9-46d2-8636-8394b9d83347  10Gi      RWO          shivsc1      62m
wp-pv-claim   Bound   pvc-a4a7254c-c077-434e-88ca-a98baa072a46  20Gi      RWO          gp2          98s

C:\Users\USER\Desktop\kube_cloud>
```

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

**Information needed**

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	shivangi123
Username	shivangi1234
Password	***** <span>Show</span>

**Important:** You will need this password to log in. Please store it in a secure location.

File systems

**Success!**

You have created a file system. You can mount your file system from an EC2 instance with an NFSv4.1 client installed. You can also mount your file system from an on-premises server over an AWS Direct Connect or AWS VPN connection. Click [here](#) for EC2 mount instructions, and [here](#) for on-premises mount instructions.

Name	File system ID	Metered size	Number of mount targets	Creation date
fs-42179d93	6.0 KiB	3	07/12/2020, 11:59:10 UTC	

Other details

Owner ID: 994441325214  
File system state: Available

Tags

No tags added

## 4. Using Helm to install Prometheus and Grafana.

### Helm

Helm helps you manage Kubernetes applications — Helm Charts help you define, install, and upgrade even the most complex Kubernetes application.

We will use helm further to install Prometheus and grafna.

```
Administrator: Command Prompt
C:\Users\USER\Desktop\kube_cloud>helm init
Creating C:\Users\USER\.helm
Creating C:\Users\USER\.helm\repository
Creating C:\Users\USER\.helm\repository\cache
Creating C:\Users\USER\.helm\repository\local
Creating C:\Users\USER\.helm\plugins
Creating C:\Users\USER\.helm\starters
Creating C:\Users\USER\.helm\cache\archive
Creating C:\Users\USER\.helm\repository\repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$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\kube_cloud>helm init
$HELM_HOME has been configured at C:\Users\USER\.helm.
Warning: Tiller is already installed in the cluster.
```

```
Administrator: Command Prompt

C:\Users\USER\Desktop\kube_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\kube_cloud>
```

```
Administrator: Command Prompt
serviceaccount/tiller created

C:\Users\USER\Desktop\kube_cloud>kubectl -n kube-system create clusterrolebinding tiller --clusterrole cluster-admin --serviceaccount=kube-system:tiller
clusterrolebinding.rbac.authorization.k8s.io/tiller created

C:\Users\USER\Desktop\kube_cloud>helm init --service-account tiller-upgrade
$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\kube_cloud>helm init --service-account tiller-upgrade
$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\kube_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\kube_cloud>
```

```
C:\Users\USER\Desktop\kube_cloud>helm install stable/prometheus --namespace prometheus --set alertmanager.persistentVolume.storageClass="gp2" --set server.persistentVolume.storageClass="gp2"
NAME: tailored-waterbuffalo
LAST DEPLOYED: Sun Jul 12 17:57:42 2020
NAMESPACE: prometheus
STATUS: DEPLOYED

RESOURCES:
--> V1/ConfigMap
NAME          DATA   AGE
tailored-waterbuffalo-prometheus-alertmanager  1      0s
tailored-waterbuffalo-prometheus-server        5      0s

--> V1/DaemonSet
NAME           DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
tailored-waterbuffalo-prometheus-node-exporter  7        7        0       7           0          <(none)>    0s

--> V1/Deployment
NAME           READY  UP-TO-DATE  AVAILABLE  AGE
tailored-waterbuffalo-kube-state-metrics  0/1   1          0          0s
tailored-waterbuffalo-prometheus-alertmanager  0/1   1          0          0s
tailored-waterbuffalo-prometheus-pushgateway  0/1   1          0          0s
tailored-waterbuffalo-prometheus-server        0/1   1          0          0s

--> V1/PersistentVolumeClaim
NAME           STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
tailored-waterbuffalo-prometheus-alertmanager  Pending  gp2     0s
tailored-waterbuffalo-prometheus-server        Pending  gp2     0s

--> V1/Pod(related)
NAME           READY  STATUS      RESTARTS  AGE
sappy-lion-kube-state-metrics-8675fcrdbd-1fsc6  1/1   Running    0          22s
tailored-waterbuffalo-kube-state-metrics-6ff5b5b767-sg488  0/1   ContainerCreating  0          0s
tailored-waterbuffalo-prometheus-alertmanager-65499c4b9f-nt77m  0/2   Pending    0          0s
tailored-waterbuffalo-prometheus-node-exporter-65zfv  0/1   Pending    0          0s
tailored-waterbuffalo-prometheus-node-exporter-d96fx  0/1   Pending    0          0s
tailored-waterbuffalo-prometheus-node-exporter-m8n4f  0/1   Pending    0          0s
tailored-waterbuffalo-prometheus-node-exporter-qg58n  0/1   Pending    0          0s
tailored-waterbuffalo-prometheus-node-exporter-rms8p  0/1   Pending    0          0s
```

```
C:\Users\USER\Desktop\kube_cloud>
C:\Users\USER\Desktop\kube_cloud>kubectl get pods -n prometheus
NAME           READY  STATUS      RESTARTS  AGE
sappy-lion-kube-state-metrics-8675fcrdbd-1fsc6  1/1   Running    0          2m58s
sappy-lion-prometheus-alertmanager-667dbd95f6-bxz5q  2/2   Running    0          2m58s
sappy-lion-prometheus-node-exporter-55hhn  1/1   Running    0          2m58s
sappy-lion-prometheus-node-exporter-5f6tk  1/1   Running    0          2m58s
sappy-lion-prometheus-node-exporter-6j1r5r  1/1   Running    0          2m58s
sappy-lion-prometheus-node-exporter-cxst6  1/1   Running    0          2m58s
sappy-lion-prometheus-node-exporter-dpngg  1/1   Running    0          2m58s
sappy-lion-prometheus-node-exporter-tp2v5  0/1   Pending    0          2m58s
sappy-lion-prometheus-node-exporter-vbrsv  1/1   Running    0          2m58s
sappy-lion-prometheus-pushgateway-7cfdb9868b6-tmps6  1/1   Running    0          2m58s
sappy-lion-prometheus-server-6f85795c7-szsf  2/2   Running    0          2m58s
tailored-waterbuffalo-kube-state-metrics-6ff5b5b767-sg488  1/1   Running    0          2m36s
tailored-waterbuffalo-prometheus-alertmanager-65499c4b9f-nt77m  2/2   Running    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-65zfv  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-d96fx  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-m8n4f  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-qg58n  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-rms8p  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-w8lwt  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-node-exporter-zjf2c  0/1   Pending    0          2m36s
tailored-waterbuffalo-prometheus-pushgateway-7fcd7c9d85-4dgsh  1/1   Running    0          2m36s
tailored-waterbuffalo-prometheus-server-646fcfc6d8b-wmb5t  2/2   Running    0          2m36s
```

The screenshot shows the Prometheus Graph interface at the URL `127.0.0.1:8888/graph`. The top navigation bar includes links for New Tab, Elastic, EC2 Metrics, vpcs, AWS ElastiCache, AWS ElastiFile, service, (4) Whales, 23rdJUL, Edit Profile, and a Plus icon. Below the navigation is a toolbar with icons for Apps, princeton, Lynda: Online Courses, Disney Professional..., Disney Imagination..., gre prep, (86) 1 Simple Exercise, exercise, and Winter Internships... A dropdown menu for Prometheus, Alerts, Graph, Status, and Help is visible.

The main area has a "Try experimental React UI" link and a checkbox for "Enable query history". A search bar contains the placeholder "Expression (press Shift+Enter for newlines)". Below it is a "Execute" button and a dropdown menu set to "- insert metric at cursor".

The interface is divided into "Graph" and "Console" tabs, with "Graph" selected. A search bar for "Moment" is centered between two arrows pointing left and right. The results table has columns for "Element" and "Value". A single row is present with the value "no data". On the right side, there is a "Remove Graph" button.

```
Administrator: Command Prompt
C:\Users\USER\Desktop\kube_cloud>helm install --name my-release stable/grafana
NAME: my-release
LAST DEPLOYED: Sun Jul 12 18:19:44 2020
NAMESPACE: shivangi
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME          AGE
my-release-grafana-clusterrole  1s

==> v1/ClusterRoleBinding
NAME          AGE
my-release-grafana-clusterrolebinding  1s

==> v1/ConfigMap
NAME      DATA AGE
my-release-grafana  1  1s
my-release-grafana-test  1  1s

==> v1/Deployment
NAME      READY UP-TO-DATE AVAILABLE AGE
my-release-grafana  0/1   1        0       0s

==> v1/Pod(related)
NAME      READY STATUS      RESTARTS AGE
my-release-grafana-786b7765f8-swrbb2  0/1  ContainerCreating  0       0s

==> v1/Role
NAME          AGE
my-release-grafana-test  1s

==> v1/RoleBinding
NAME          AGE
my-release-grafana-test  1s

==> v1/Secret
NAME      TYPE  DATA AGE
my-release-grafana  Opaque  3    1s

==> v1/Service
```

```
Administrator: Command Prompt

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

==> v1beta1/Role
NAME          AGE
my-release-grafana  1s

==> v1beta1/RoleBinding
NAME          AGE
my-release-grafana  1s

NOTES:
1. Get your 'admin' user password by running:
   kubectl get secret --namespace shivangi my-release-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:
   my-release-grafana.shivangi.svc.cluster.local

   Get the Grafana URL to visit by running these commands in the same shell:
   export POD_NAME=$(kubectl get pods --namespace shivangi -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=my-release" -o jsonpath=".items[0].metadata.name")
   kubectl --namespace shivangi port-forward $POD_NAME 3000

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\kube_cloud>
```

```
Administrator: Command Prompt

Run: helm ls --all my-release; to check the status of the release
Or run: helm del --purge my-release; to delete it

C:\Users\USER\Desktop\kube_cloud>helm install --name my-release1 stable/grafana
NAME: my-release1
LAST DEPLOYED: Sun Jul 12 18:23:46 2020
NAMESPACE: shivangi
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME          AGE
my-release1-grafana-clusterrole  1s

==> v1/ClusterRoleBinding
NAME          AGE
my-release1-grafana-clusterrolebinding  1s

==> v1/ConfigMap
NAME          DATA  AGE
my-release1-grafana  1    1s
my-release1-grafana-test  1    1s

==> v1/Deployment
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
my-release1-grafana  0/1    1        0         0s

==> v1/Pod(related)
NAME          READY  STATUS      RESTARTS  AGE
my-release1-grafana-8644697c6d-z4r2h  0/1    ContainerCreating  0         0s

==> v1/Role
NAME          AGE
my-release1-grafana-test  1s

==> v1/RoleBinding
NAME          AGE
```

```
Administrator: Command Prompt

==> v1/ServiceAccount
NAME      SECRETS  AGE
my-release1-grafana   1      1s
my-release1-grafana-test 1      1s

==> vibeta1/PodSecurityPolicy
NAME          PRIV  CAPS  SELINUX  RUNASUSER  FSGROUP  SUPGROUP  READONLYROOTFS  VOLUMES
my-release1-grafana  false  RunAsAny  RunAsAny  RunAsAny  false    configMap,emptyDir,projected,secret,downwardAPI,persistentVolumeClaim
my-release1-grafana-test  false  RunAsAny  RunAsAny  RunAsAny  RunAsAny  false    configMap,downwardAPI,emptyDir,projected,secret

==> vibeta1/Role
NAME      AGE
my-release1-grafana  0s

==> vibeta1/RoleBinding
NAME      AGE
my-release1-grafana  0s

NOTES:
1. Get your 'admin' user password by running:
   kubectl get secret --namespace shivangi my-release1-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:
   my-release1-grafana.shivangi.svc.cluster.local

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

   export POD_NAME=$(kubectl get pods --namespace shivangi -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=my-release1" -o jsonpath=".items[0].metadata.name")
   kubectl --namespace shivangi port-forward $POD_NAME 3000

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.
#####

Administrator: Command Prompt
```

```
Administrator: Command Prompt

C:\Users\USER\Desktop\kube_cloud>helm install --name my-release2 stable/grafana
NAME: my-release2
LAST DEPLOYED: Sun Jul 12 18:25:30 2020
NAMESPACE: shivangi
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME      AGE
my-release2-grafana-clusterrole  0s

==> v1/ClusterRoleBinding
NAME      AGE
my-release2-grafana-clusterrolebinding  0s

==> v1/ConfigMap
NAME      DATA  AGE
my-release2-grafana   1      0s
my-release2-grafana-test 1      0s

==> v1/Deployment
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
my-release2-grafana  0/1    1          0          0s

==> v1/Pod(related)
NAME      READY  STATUS      RESTARTS  AGE
my-release2-grafana-55d7bcb557-s546z  0/1  ContainerCreating  0      0s

==> v1/Role
NAME      AGE
my-release2-grafana-test  0s

==> v1/RoleBinding
NAME      AGE
my-release2-grafana-test  0s

==> v1/Secret
NAME      TYPE  DATA  AGE
my-release2-grafana  Opaque  3      0s

Administrator: Command Prompt
```

```
Administrator: Command Prompt
C:\Users\USER\Desktop\kube_cloud>helm install stable/grafana --namespace grafna --set persistent.storageClassName="gp2"
--set adminPassword='redhat' --set service.type=LoadBalancer
NAME: worn-coral
LAST DEPLOYED: Sun Jul 12 18:39:58 2020
NAMESPACE: grafna
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME          AGE
worn-coral-grafana-clusterrole  1s

==> v1/ClusterRoleBinding
NAME          AGE
worn-coral-grafana-clusterrolebinding  1s

==> v1/ConfigMap
NAME          DATA  AGE
worn-coral-grafana   1     1s
worn-coral-grafana-test 1     1s

==> v1/Deployment
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
worn-coral-grafana  0/1    1          0         1s

==> v1/Pod(related)
NAME      READY  STATUS    RESTARTS  AGE
worn-coral-grafana-7465bbd645-42vc1  0/1    Running  0         1s

==> v1/Role
```



