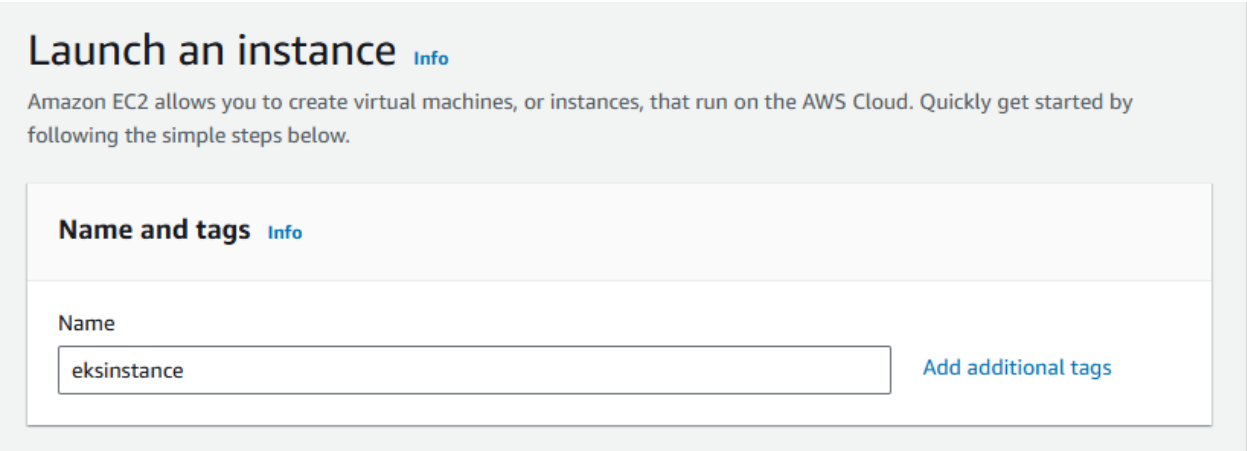
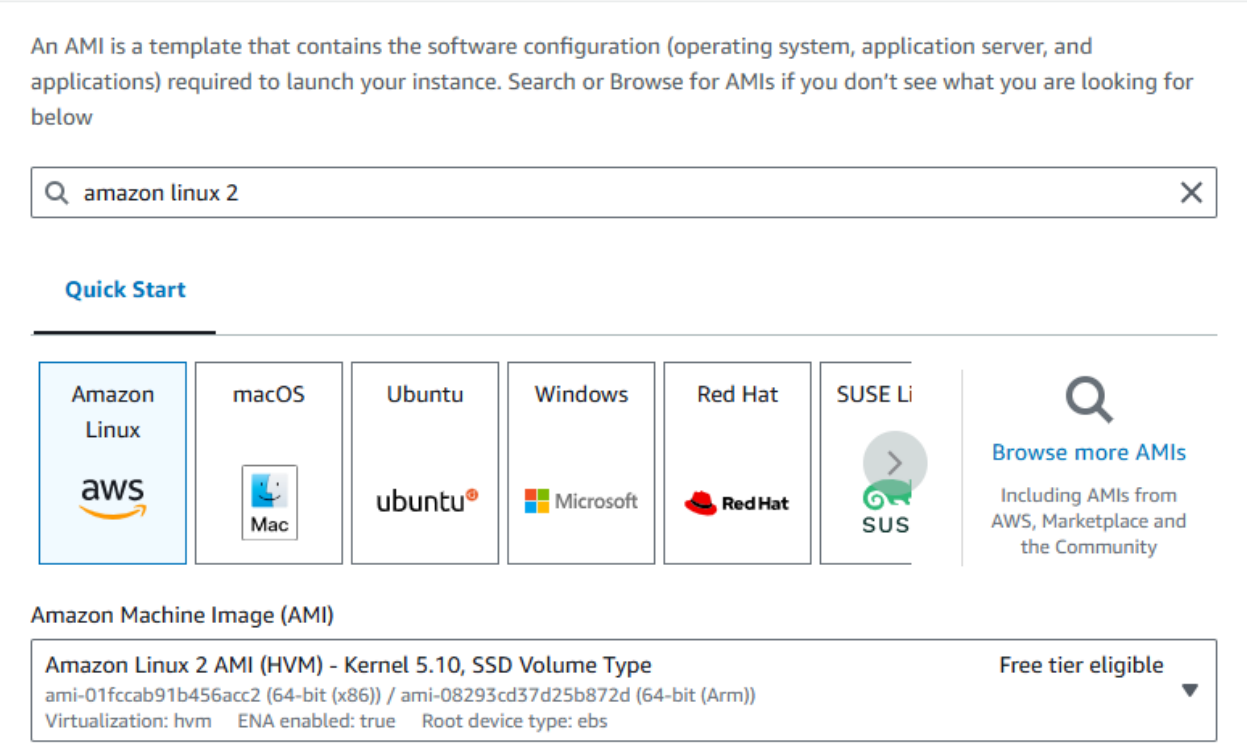
Step 1: Creating an EC2 Instance





Key pair name

Key pairs allow you to connect to your instance securely.

mykeypair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

- (selected) **RSA**
  RSA encrypted private and public key pair
- ( ) **ED25519**
  ED25519 encrypted private and public key pair

Private key file format

- (selected) **.pem**
  For use with OpenSSH
- ( ) **.ppk**
  For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Network Info

vpc-0978b75104d1238f0

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- (•) Create security group
- ( ) Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

- [x] Allow SSH traffic from
  Helps you connect to your instance
  Anywhere
  0.0.0.0/0
- [ ] Allow HTTPS traffic from the internet
  To set up an endpoint, for example when creating a web server
- [x] Allow HTTP traffic from the internet
  To set up an endpoint, for example when creating a web server

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-03557776ce2ff426c)

Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

1 2 3 4 5 6

Instances (1) Info

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

Instance ID = i-03557776ce2ff426c | Clear filters

1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabili
eksinstance	i-03557776ce2ff426c	Running	t2.micro	Initializing	View alarms +	us-east-1c

```
$ ssh -i "veenakey.pem" ec2-user@ec2-52-91-99-65.compute-1.amazonaws.com
Last login: Wed Oct 18 08:47:43 2023 from 103.170.220.184
       ,     #_
       ~\_  ####_        Amazon Linux 2
      ~~  \_#####\
      ~~     \###|       AL2 End of Life is 2025-06-30.
      ~~       \#/ ___
       ~~       V~' '->
        ~~~         /    A newer version of Amazon Linux is available!
          ~~._.   _/
             _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
           _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-24-84 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                | 3.6 kB     00:00
No packages marked for update
[ec2-user@ip-172-31-24-84 ~]$ sudo yum install unzip
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package unzip-6.0-57.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-24-84 ~]$ sudo unzip OCI.zip
Archive:  OCI.zip
  inflating: Dockerfile
```

```
[ec2-user@ip-172-31-24-84 ~]$ sudo amazon-linux-extras install docker
Installing docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10
21 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                              | 3.6 kB     00:00
amzn2extra-docker                                       | 2.9 kB     00:00
amzn2extra-kernel-5.10                                  | 3.0 kB     00:00
(1/7): amzn2-core/2/x86_64/group_gz                       | 2.7 kB   00:00
(2/7): amzn2-core/2/x86_64/updateinfo                     | 719 kB   00:00
(3/7): amzn2extra-kernel-5.10/2/x86_64/primary            | 8.7 MB   00:00
(4/7): amzn2extra-docker/2/x86_64/primary                 |  25 kB   00:00
(5/7): amzn2extra-docker/2/x86_64/updateinfo              |  13 kB   00:00
(6/7): amzn2extra-kernel-5.10/2/x86_64/updateinfo         |  39 kB   00:00
(7/7): amzn2-core/2/x86_64/primary_db                     |  67 MB   00:01
Resolving Dependencies
--> Running transaction check
---> Package docker.x86_64 0:20.10.23-1.amzn2.0.1 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Processing Dependency: pigz for package: docker-20.10.23-1.amzn2.0.1.x86_64
--> Running transaction check
---> Package containerd.x86_64 0:1.6.19-1.amzn2.0.3 will be installed
---> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
---> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
---> Package runc.x86_64 0:1.1.7-3.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```
[ec2-user@ip-172-31-24-84 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-24-84 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-24-84 ~]$ exit
logout
```

```
      A newer version of Amazon Linux is available!
        Amazon Linux 2023, GA and supported until 2028-03-15.
          https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-24-84 ~]$ sudo yum install awscli -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package awscli-1.18.147-1.amzn2.0.2.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-24-84 ~]$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-172-31-24-84 ~]$ docker build -t lambda_ecr .
Sending build context to Docker daemon  10.30MB
Step 1/4 : FROM python:alpine
alpine: Pulling from library/python
8626ea774ef: Pull complete
875af95346d: Pull complete
9ba8f7e671f: Pull complete
e93e141e36c: Pull complete
0688e5d3611: Pull complete
Digest: sha256:ae35274f417fc81ba6ee1fc84206e8517f28117566ee6e04a64f004c1409bdac
Status: Downloaded newer image for python:alpine
 ---> 3ca908b949b4
Step 2/4 : COPY ./context .
 ---> 12e1ef557247
Step 3/4 : RUN pip install -r requirements.txt
 ---> Running in 5e300f27f03d
Collecting boto3>=1.16.19 (from -r requirements.txt (line 1))
  Obtaining dependency information for boto3>=1.16.19 from https://files.pythonhosted.org/packages/c7/dd/4fe47b2cec8731ec26d7410e659c4f0c4cd36baa835e2312c80ec5383b07/boto3-1.28.85-py3-none-any
.whl.metadata
  Downloading boto3-1.28.85-py3-none-any.whl.metadata (6.7 kB)
Collecting requests>=2.25.0 (from -r requirements.txt (line 2))
  Obtaining dependency information for requests>=2.25.0 from https://files.pythonhosted.org/packages/70/8e/0e2d847013cb52cd35b38c009bb167a1a26b2ce6cd6965bf26b471c08f44/requests-2.31.0-py3-none
-any.whl.metadata
  Downloading requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)
Collecting botocore<1.32.0,>=1.31.65 (from boto3>=1.16.19->-r requirements.txt (line 1))
  Obtaining dependency information for botocore<1.32.0,>=1.31.65 from https://files.pythonhosted.org/packages/63/c6/8e29a2b4dffa188d07c26d19ae578a26d8063834e4d844bf22c2a8028229/botocore-1.31.6
5-py3-none-any.whl.metadata
  Downloading botocore-1.31.65-py3-none-any.whl.metadata (6.1 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3>=1.16.19->-r requirements.txt (line 1))
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.8.0,>=0.7.0 (from boto3>=1.16.19->-r requirements.txt (line 1))
  Obtaining dependency information for s3transfer<0.8.0,>=0.7.0 from https://files.pythonhosted.org/packages/5a/4b/fec9ca18f8874a96c5061422625ba86c3ee1e6587ccd92ff9f5bf7bd91b2/s3transfer-0.7.0
-py3-none-any.whl.metadata
  Downloading s3transfer-0.7.0-py3-none-any.whl.metadata (1.8 kB)
Collecting charset-normalizer<4,>=2 (from requests>=2.25.0->-r requirements.txt (line 2))
  Obtaining dependency information for charset-normalizer<4,>=2 from https://files.pythonhosted.org/packages/96/88/b69aa474040ae8f16df6689fe042884f015956998050d1c38b389fb16981/charset_normaliz
```

```
Successfully built c81056ea8ef9
Successfully tagged lambda_ecr:latest
```

```
[ec2-user@ip-172-31-24-84 ~]$ docker images
REPOSITORY                                                  TAG       IMAGE ID       CREATED          SIZE
627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr     latest    c81056ea8ef9   25 minutes ago   161MB
lambda_ecr                                                  latest    c81056ea8ef9   25 minutes ago   161MB
python                                                      alpine    3ca908b949b4   2 weeks ago      51.1MB
```

```
Downloading botocore-1.31.65-py3-none-any.whl (11.3 MB)
   ---------------------------------------- 11.3/11.3 MB 67.6 MB/s eta 0:00:00
Downloading certifi-2023.7.22-py3-none-any.whl (158 kB)
   ---------------------------------------- 158.3/158.3 kB 33.2 MB/s eta 0:00:00
Downloading charset_normalizer-3.3.0-cp312-cp312-musllinux_1_1_x86_64.whl (138 kB)
   ---------------------------------------- 138.8/138.8 kB 26.6 MB/s eta 0:00:00
Downloading s3transfer-0.7.0-py3-none-any.whl (79 kB)
   ---------------------------------------- 79.8/79.8 kB 19.8 MB/s eta 0:00:00
Downloading urllib3-2.0.7-py3-none-any.whl (124 kB)
   ---------------------------------------- 124.2/124.2 kB 27.2 MB/s eta 0:00:00
Installing collected packages: urllib3, six, jmespath, idna, charset-normalizer, certifi, requests, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.28.85 botocore-1.31.65 certifi-2023.7.22 charset-normalizer-3.3.0 idna-3.4 jmespath-1.0.1 python-dateutil-2.8.2 requests-2.31.0 s3transfer-0.7.0 six-1.16.0 ur
llib3-2.0.7
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: ht
tps://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 23.3
[notice] To update, run: pip install --upgrade pip
Removing intermediate container 5e300f27f03d
 ---> d3a63f1e8445
Step 4/4 : CMD python3 bootstrap.py
 ---> Running in 730e5990a0a1
Removing intermediate container 730e5990a0a1
 ---> c81056ea8ef9
Successfully built c81056ea8ef9
Successfully tagged lambda_ecr:latest
[ec2-user@ip-172-31-24-84 ~]$ client_loop: send disconnect: Connection reset by peer

Windows 10Ahutush MINGW64 ~/Downloads
$ ssh -i "veetakey.pem" ec2-user@ec2-52-91-99-65.compute-1.amazonaws.com
Last login: Wed Oct 18 09:19:01 2023 from 103.170.221.79
       __|  __|_  )
       _|  (     /   Amazon Linux 2
      ___|\___|___|
https://aws.amazon.com/linux/amazon-linux-2/
[ec2-user@ip-172-31-24-84 ~]$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 627416124954.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
Installing collected packages: urllib3, six, jmespath, idna, charset-normalizer, certifi, requests, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.28.65 botocore-1.31.65 certifi-2023.7.22 charset-normalizer-3.3.0 idna-3.4 jmespath-1.0.1 python-dateutil-2.8.2 requests-2.31.0 s3transfer-0.7.0 six-1.16.0 urll
b3-2.0.7
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: http
s://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 23.3
[notice] To update, run: pip install --upgrade pip
Removing intermediate container 5e800f27f00d
 ---> d3e83f1e9445
Step 4/4 : CMD python3 bootstrap.py
 ---> Running in 730e5990abal
Removing intermediate container 730e5990abal
 ---> c81056ea8ef9
Successfully built c81056ea8ef9
Successfully tagged lambda_ecr:latest
[ec2-user@ip-172-31-24-84 ~]$ client_loop: send disconnect: Connection reset by peer

Windows 10@Abhatosh MINGW64 ~/Downloads
$ ssh -i "veerakey.pem" ec2-user@ec2-52-91-99-65.compute-1.amazonaws.com
Last login: Wed Oct 18 09:19:01 2023 from 103.170.221.79
       ,     #_
       ~\_  ####_        Amazon Linux 2
      ~~  \_#####\
      ~~     \###|       AL2 End of Life is 2025-06-30.
      ~~       \#/ ___
       ~~       V~' '->
        ~~~         /    A newer version of Amazon Linux is available!
          ~~._.   _/
             _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
           _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-24-84 ~]$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 627416124954.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-24-84 ~]$ docker tag lambda_ecr:latest 627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr:latest
[ec2-user@ip-172-31-24-84 ~]$ docker push 627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr:latest
The push refers to repository [627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr]
6fa88d08052f: Pushed
e2cd8f667263: Pushed
8355721b9cf2: Pushed
c69f1c6d46dc: Pushed
33df02257737: Pushed
a1c2f658ec5f: Pushed
cc2447e1835a: Pushed
latest: digest: sha256:2e6b72ec41b774bc247b629360a0b686e9a9b23015bba0f21f4c9bb1c98cc167 size: 1787
[ec2-user@ip-172-31-24-84 ~]$ |
```

```
[ec2-user@ip-172-31-24-84 ~]$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 627416124954.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-24-84 ~]$ docker tag lambda_ecr:latest 627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr:latest
[ec2-user@ip-172-31-24-84 ~]$ docker push 627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr:latest
The push refers to repository [627416124954.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr]
6fa88d08052f: Pushed
e2cd8f667263: Pushed
8355721b9cf2: Pushed
c69f1c6d46dc: Pushed
33df02257737: Pushed
a1c2f658ec5f: Pushed
cc2447e1835a: Pushed
latest: digest: sha256:2e6b72ec41b774bc247b629360a0b686e9a9b23015bba0f21f4c9bb1c98cc167 size: 1787
[ec2-user@ip-172-31-24-84 ~]$ |
```

HELM

HELM :- package manager for Kubernetes & is indeed a templating tool & you define Kubernetes object that you need & can use templating language to populate little bits of those objects with the information that is configurable
In CI/CD, helm chart can be used to deploy on Kubernetes

```
kubectl version --short
helm version --short
helm create Tourism Management System
mkdir charts
cd charts
mkdir my-nginx
cd my-nginx
vi Chart.yaml
→ apiVersion: v1
  name: my-nginx
  version: 0.1.0
  appversion: 5.6
  description: My custom nginx chart
  :wq

mkdir templates
tree

>> kubectl create deploy my-nginx --image nginx
>> kubectl get all
>> Kubectl create deploy my-nginx --image nginx --dry run -o
>> Kubectl create deploy my-nginx --image nginx --dry run -o yaml
```

capital

It contains information about the chart

← chart version

← App's version

>> kubectl create deploy my-nginx --image nginx --dry-run -o yaml > charts/my-nginx/templates

>> kubectl create deploy my-nginx --image nginx --dry run -o yaml > charts/my-nginx/templates/deployment.yaml

>> kubectl delete deploy my-nginx

clear

>> kubectl get all

→ Now we have a chart i.e. chart.yaml [information about our chart] & then we also have the templates directory where we have our deployment file.

→ Now we are then good to proceed with helming install command to install the chart

>> watch -x kubectl get all

→ now we see this chart getting deployed in our cluster

>> helm install --name my-nginx

>> clear

>> helm list

>> clear

>> kubectl expose deploy my-nginx --port 80 --dry-run -o yaml

>> kubectl expose deploy my-nginx --port 80 --dry-run -o yaml > templates/service.yaml

Now we have:-

```
my-
 ├→ Chart.yaml
 ├→ templates.yaml
 ├→ deployment.yaml
 └→ service.yaml
```

Date
Page

Now, we need to upgrade the installed application.

>> helm list

>> vi chart.yaml

```
apiVersion v1
name my-nginx
version 0.2.0
apiVersion 5.6
description My custom nginx chart
:wq
```

>> helm upgrade my-nginx .

(Here, service -> my-nginx got created)

>> (So, we have created deployment, service)

>> clear

>> helm list [This is now with the upgraded version [0.2.0]]

>> helm rollback my-nginx 1

>> helm rollback my-nginx 2

>> helm list

>> helm delete --purge my-nginx

Containerizing the application

```yaml
version: '3.8'
services:
  app:
     build: ./path_to_php_application
     ports:
       - "8080:80"
     links:
       - db
  db:
     build: ./path_to_database
     ports:
        - "3306:3306"
     environment:
     MYSQL_ROOT_PASSWORD: rootpassword
     MYSQL_DATABASE: tms
adminer:
     image: adminer
     ports:
       - 8081:8080
```

Deployment for PHP application to ensure its availability and scalability

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
   name: php-app
spec:
   replicas: 3
   selector:
    matchlabels:
       app: php-app
    template:
    metadata:
       labels:
           app: php-app
    spec:
      containers:
          -name : php-container
    image: your-docker-image- for- php-app p
    ports:
     - containerPort: 80
```

Configuring an ingress resource for exposing the application externally

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
name: php-app-ingress
spec:
   rules:
   - host: your-app-domain.com
     http:
        paths:
        - path: /
        pathType: Prefix
        backend:
          service:
             name: php-app
             port:
                number: 80
```

Creating PersistentVolume for providing storage for the application and database

```yaml
apiVersion: v1 2
kind: PersistentVolume
metadata:
   name: pv-volume
   labels:
      type: local
spec:
  storageClassName: manual
  capacity:
      storage: 1Gi
accessModes :
      - ReadWriteOnce
hostPath:
     path: "/mnt/data"
```

Creating PersistentVolumeClaim for providing storage for the application and database

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
   name: pv-claim
spec:
   storageClassName: manual
   accessModes:
      - ReadWriteOnce
   resources:
     requests:
        storage: 1Gi
```

Setting up NFS (standard protocol that allow to mount a storage device on a local drive) server for providing storage for the application

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: nfs
provisioner: example.com/nfs
parameters:
  server: nfs-server.example.com
  path: /exported/path
```