MARK ANGELO D. SIAZON IV-ACSAD ELEC-3 ASS#5

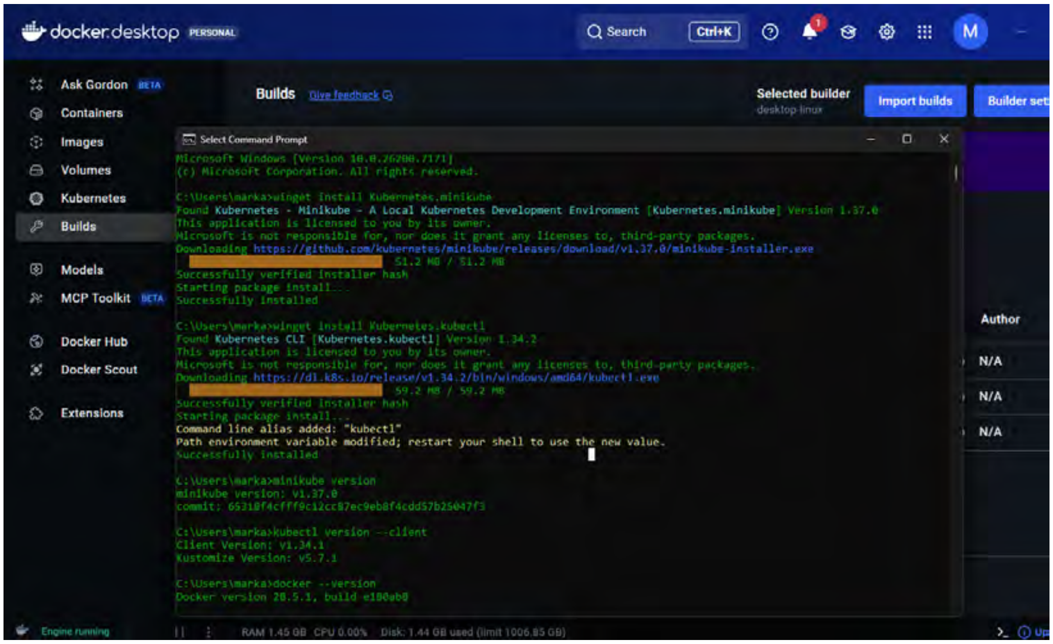
Assignment: Home Lab Activity

1. **Hello Minikube:**
    a. https://kubernetes.io/docs/tutorials/hello-minikube/
2. ***Get a Shell to a Running Container:***
    a. https://kubernetes.io/docs/tasks/debug/debug-application/get-shell-running-container/
3. ***Deploying Wordpress and MySQL with persistent volumes:***
    a. https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/

---

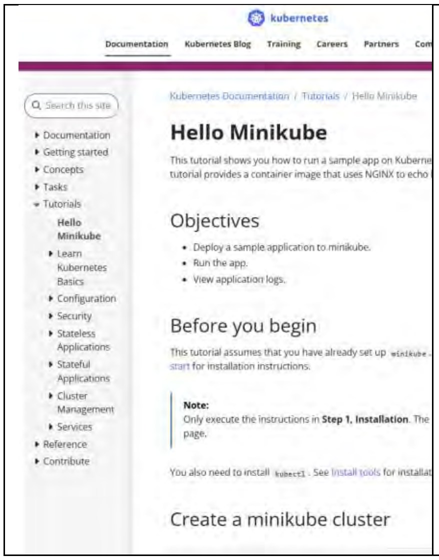
1. Installation Phase

- Install **Docker Desktop**
- Install **Minikube**
    - or type in cmd:
        - winget install Kubernetes.minikube
- **Install kubectl**
    - or type in cmd:
        - winget install Kubernetes.kubectl



```
Select Command Prompt
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\marka>winget install Kubernetes.minikube
Found Kubernetes - Minikube - A Local Kubernetes Development Environment [Kubernetes.minikube] Version 1.37.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/kubernetes/minikube/releases/download/v1.37.0/minikube-installer.exe
  ██████████████████████████████  51.2 MB / 51.2 MB
Successfully verified installer hash
Starting package install...
Successfully installed

C:\Users\marka>winget install Kubernetes.kubectl
Found Kubernetes CLI [Kubernetes.kubectl] Version 1.34.2
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://dl.k8s.io/release/v1.34.2/bin/windows/amd64/kubectl.exe
  ██████████████████████████████  59.2 MB / 59.2 MB
Successfully verified installer hash
Starting package install...
Command line alias added: "kubectl"
Path environment variable modified; restart your shell to use the new value.
Successfully installed

C:\Users\marka>minikube version
minikube version: v1.37.0
commit: 65318f4cfff9c12cc87ec9eb8f4cdd57b25047f3

C:\Users\marka>kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1

C:\Users\marka>docker --version
Docker version 28.5.1, build e180ab8
```

2. First Kubernetes Cluster



Kubernetes Documentation / Tutorials / Hello Minikube

Hello Minikube

This tutorial shows you how to run a sample app on Kuberne
tutorial provides a container image that uses NGINX to echo

Objectives

- Deploy a sample application to minikube.
- Run the app.
- View application logs.

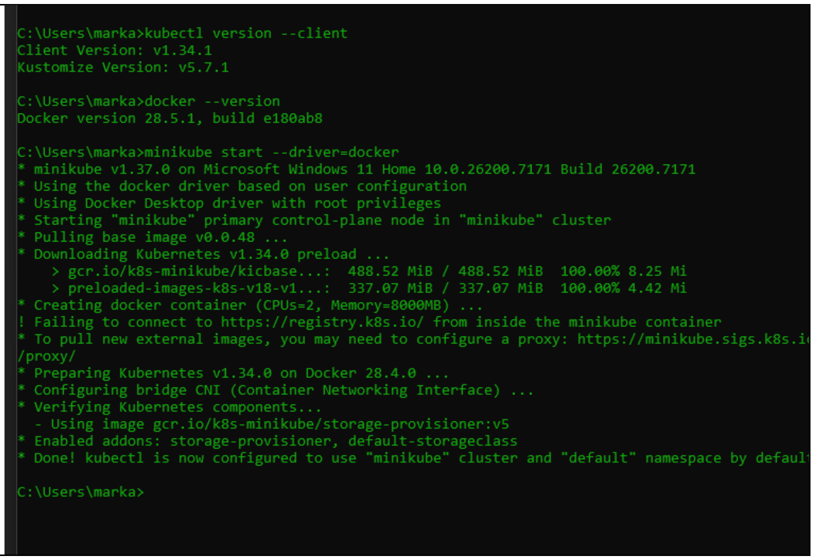
Before you begin

This tutorial assumes that you have already set up `minikube`
start for installation instructions.

Note:
Only execute the instructions in **Step 1, Installation**. The
page.

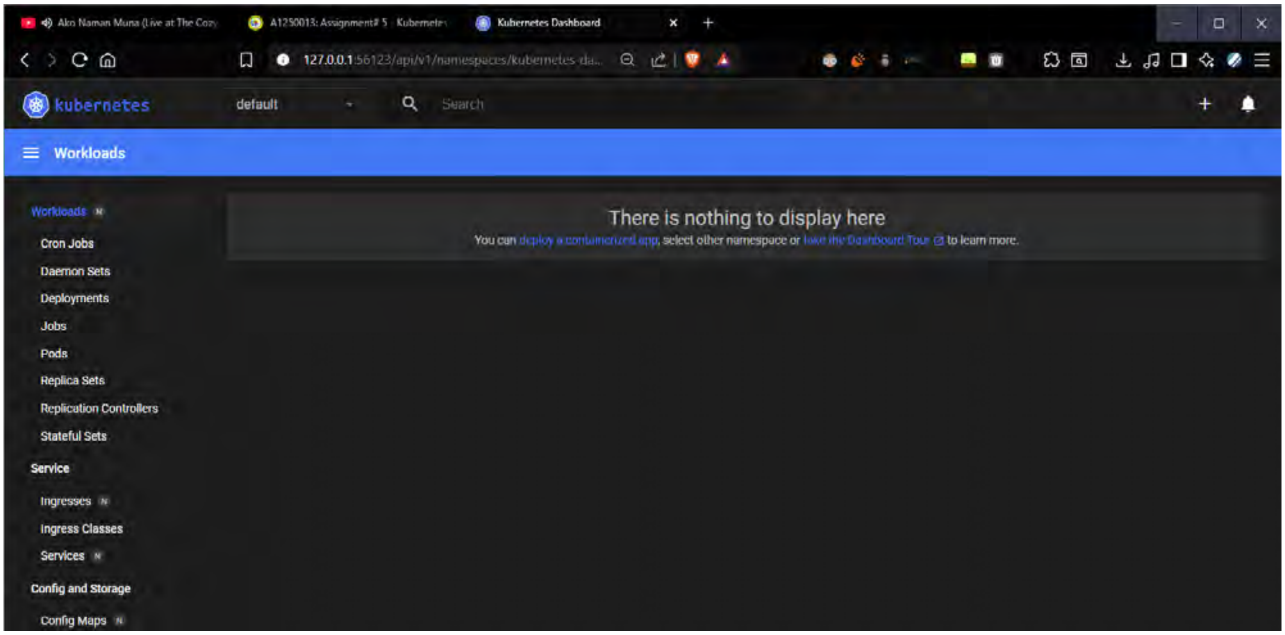
You also need to install `kubectl`. See Install tools for installat

Create a minikube cluster



```
C:\Users\marka>kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1

C:\Users\marka>docker --version
Docker version 28.5.1, build e180ab8

C:\Users\marka>minikube start --driver=docker
* minikube v1.37.0 on Microsoft Windows 11 Home 10.0.26200.7171 Build 26200.7171
* Using the docker driver based on user configuration
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.48 ...
* Downloading Kubernetes v1.34.0 preload ...
    > gcr.io/k8s-minikube/kicbase...:  488.52 MiB / 488.52 MiB  100.00% 8.25 Mi
    > preloaded-images-k8s-v18-v1...:  337.07 MiB / 337.07 MiB  100.00% 4.42 Mi
* Creating docker container (CPUs=2, Memory=8000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.i
/proxy/
* Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by defaul

C:\Users\marka>
```

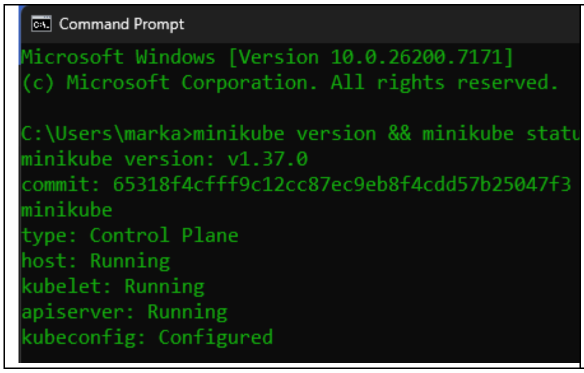
Kubernetes running



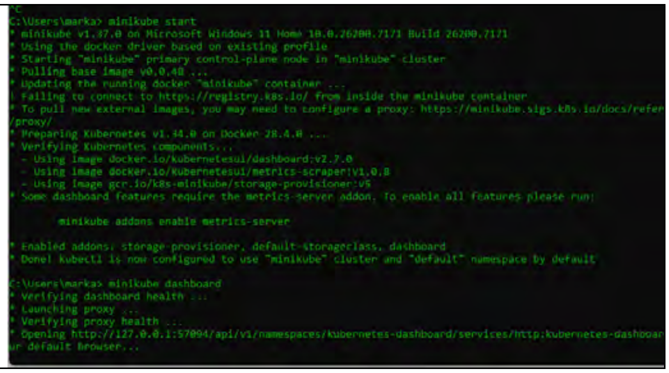
There is nothing to display here

You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.

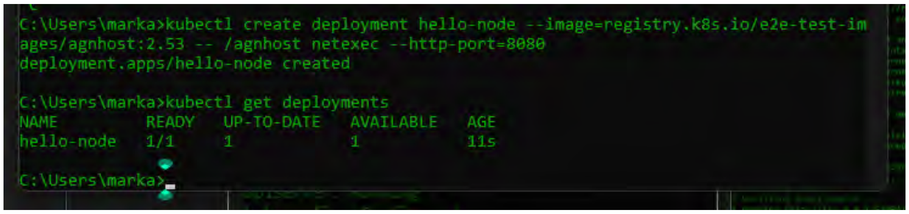
Create and View Deployment



```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\marka>minikube version && minikube statu
minikube version: v1.37.0
commit: 65318f4cfff9c12cc87ec9eb8f4cdd57b25047f3
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

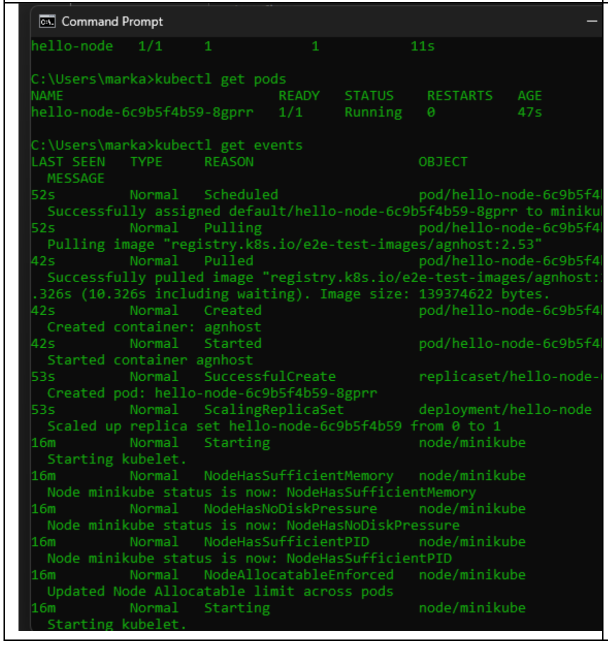


```
^C
C:\Users\marka> minikube start
* minikube v1.37.0 on Microsoft Windows 11 Home 10.0.26200.7171 Build 26200.7171
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.48 ...
* Updating the running docker "minikube" container ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/refer
/proxy/
* Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
* Verifying Kubernetes components...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

* Enabled addons: storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\marka> minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:57094/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboar
ur default browser...
```



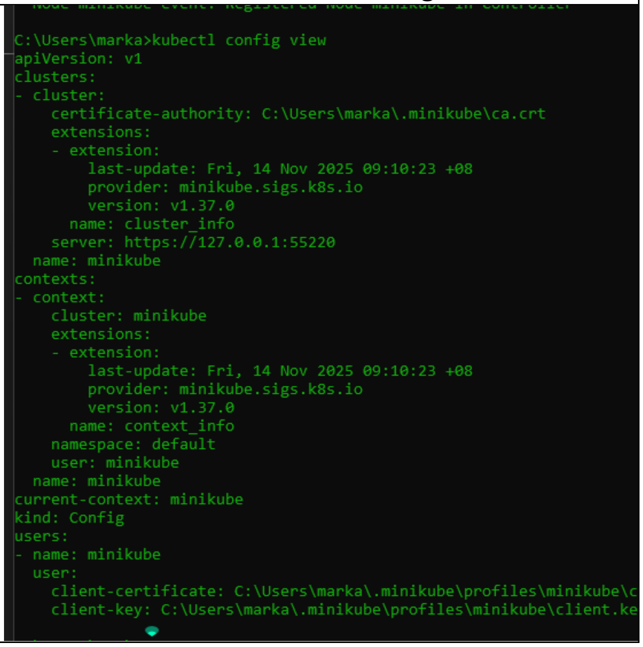
```
C:\Users\marka>kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-im
ages/agnhost:2.53 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created

C:\Users\marka>kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           11s

C:\Users\marka>_
```

View Pods & Cluster Events



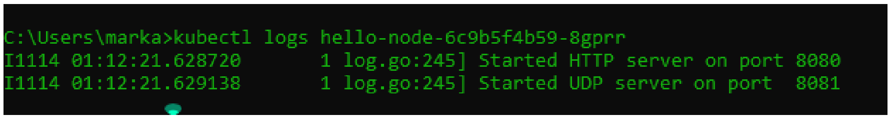
```
hello-node   1/1     1            1           11s

C:\Users\marka>kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-6c9b5f4b59-8gprr   1/1     Running   0          47s

C:\Users\marka>kubectl get events
LAST SEEN   TYPE     REASON                    OBJECT
  MESSAGE
52s         Normal   Scheduled                 pod/hello-node-6c9b5f4
  Successfully assigned default/hello-node-6c9b5f4b59-8gprr to miniku
52s         Normal   Pulling                   pod/hello-node-6c9b5f4
  Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.53"
42s         Normal   Pulled                    pod/hello-node-6c9b5f4
  Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:
.326s (10.326s including waiting). Image size: 139374622 bytes.
42s         Normal   Created                   pod/hello-node-6c9b5f4
  Created container: agnhost
42s         Normal   Started                   pod/hello-node-6c9b5f4
  Started container agnhost
53s         Normal   SuccessfulCreate          replicaset/hello-node-
  Created pod: hello-node-6c9b5f4b59-8gprr
53s         Normal   ScalingReplicaSet         deployment/hello-node
  Scaled up replica set hello-node-6c9b5f4b59 from 0 to 1
16m         Normal   Starting                  node/minikube
  Starting kubelet.
16m         Normal   NodeHasSufficientMemory   node/minikube
  Node minikube status is now: NodeHasSufficientMemory
16m         Normal   NodeHasNoDiskPressure     node/minikube
  Node minikube status is now: NodeHasNoDiskPressure
16m         Normal   NodeHasSufficientPID      node/minikube
  Node minikube status is now: NodeHasSufficientPID
16m         Normal   NodeAllocatableEnforced   node/minikube
  Updated Node Allocatable limit across pods
16m         Normal   Starting                  node/minikube
  Starting kubelet.
```

View kubectl Config



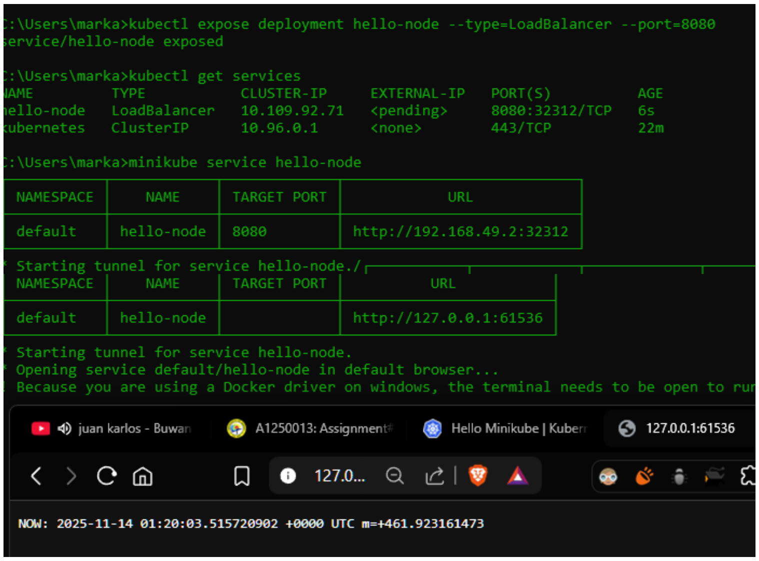
```
C:\Users\marka>kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: C:\Users\marka\.minikube\ca.crt
    extensions:
    - extension:
        last-update: Fri, 14 Nov 2025 09:10:23 +08
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: cluster_info
    server: https://127.0.0.1:55220
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Fri, 14 Nov 2025 09:10:23 +08
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
users:
- name: minikube
  user:
    client-certificate: C:\Users\marka\.minikube\profiles\minikube\c
    client-key: C:\Users\marka\.minikube\profiles\minikube\client.ke
```

View Pod Logs



```
C:\Users\marka>kubectl logs hello-node-6c9b5f4b59-8gprr
I1114 01:12:21.628720       1 log.go:245] Started HTTP server on port 8080
I1114 01:12:21.629138       1 log.go:245] Started UDP server on port  8081
```

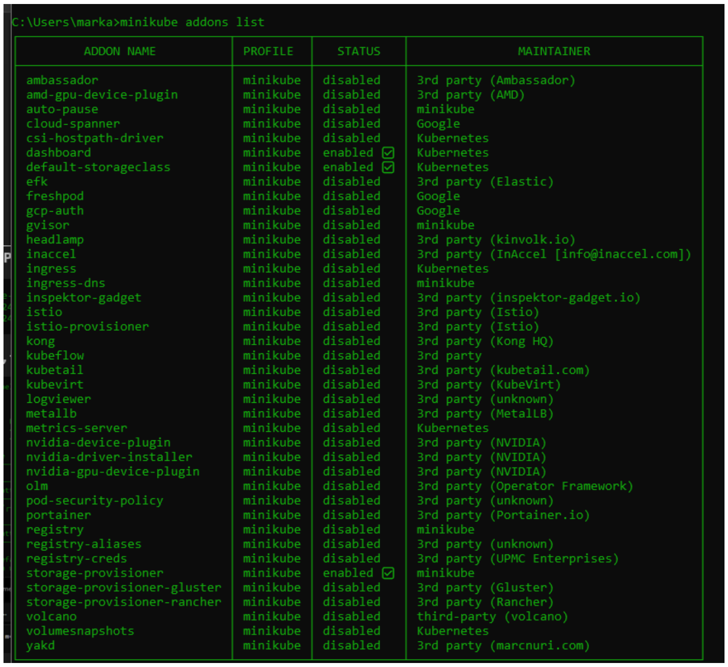
Create and View Services, then Access App in browser



```
:\Users\marka>kubectl expose deployment hello-node --type=LoadBalancer --port=8080
ervice/hello-node exposed

:\Users\marka>kubectl get services
AME          TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
ello-node    LoadBalancer   10.109.92.71   <pending>     8080:32312/TCP   6s
ubernetes    ClusterIP      10.96.0.1      <none>        443/TCP          22m

:\Users\marka>minikube service hello-node
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
| default   | hello-node | 8080        | http://192.168.49.2:32312 |
 Starting tunnel for service hello-node./
| NAMESPACE |    NAME    | TARGET PORT |           URL          |
| default   | hello-node |             | http://127.0.0.1:61536 |
 Starting tunnel for service hello-node.
 Opening service default/hello-node in default browser...
 Because you are using a Docker driver on windows, the terminal needs to be open to ru
```

```
NOW: 2025-11-14 01:20:03.515720902 +0000 UTC m=+461.923161473
```

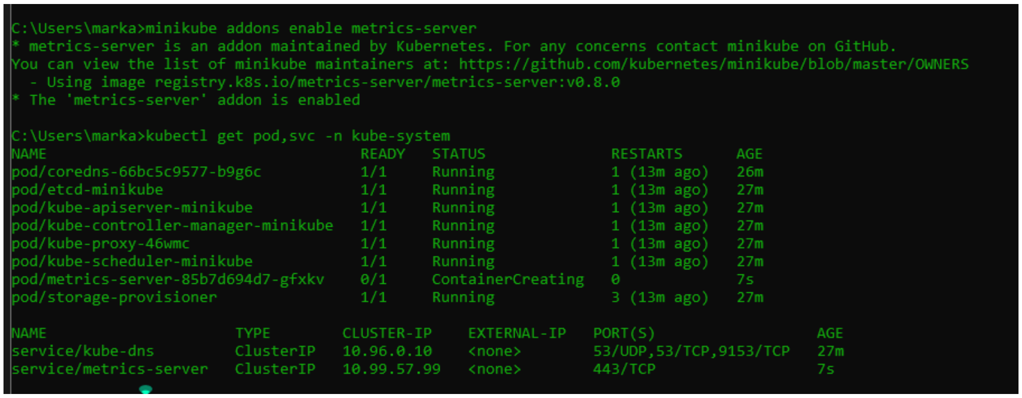
View Available Addons



```
C:\Users\marka>minikube addons list
┌─────────────────────────────┬──────────┬────────────┬──────────────────────────────────────────┐
│         ADDON NAME          │ PROFILE  │   STATUS   │                MAINTAINER                │
├─────────────────────────────┼──────────┼────────────┼──────────────────────────────────────────┤
│ ambassador                  │ minikube │ disabled   │ 3rd party (Ambassador)                   │
│ amd-gpu-device-plugin       │ minikube │ disabled   │ 3rd party (AMD)                          │
│ auto-pause                  │ minikube │ disabled   │ minikube                                 │
│ cloud-spanner               │ minikube │ disabled   │ Google                                   │
│ csi-hostpath-driver         │ minikube │ disabled   │ Kubernetes                               │
│ dashboard                   │ minikube │ enabled ✅ │ Kubernetes                               │
│ default-storageclass        │ minikube │ enabled ✅ │ Kubernetes                               │
│ efk                         │ minikube │ disabled   │ 3rd party (Elastic)                      │
│ freshpod                    │ minikube │ disabled   │ Google                                   │
│ gcp-auth                    │ minikube │ disabled   │ Google                                   │
│ gvisor                      │ minikube │ disabled   │ minikube                                 │
│ headlamp                    │ minikube │ disabled   │ 3rd party (kinvolk.io)                   │
│ inaccel                     │ minikube │ disabled   │ 3rd party (InAccel [info@inaccel.com])   │
│ ingress                     │ minikube │ disabled   │ Kubernetes                               │
│ ingress-dns                 │ minikube │ disabled   │ minikube                                 │
│ inspektor-gadget            │ minikube │ disabled   │ 3rd party (inspektor-gadget.io)          │
│ istio                       │ minikube │ disabled   │ 3rd party (Istio)                        │
│ istio-provisioner           │ minikube │ disabled   │ 3rd party (Istio)                        │
│ kong                        │ minikube │ disabled   │ 3rd party (Kong HQ)                      │
│ kubeflow                    │ minikube │ disabled   │ 3rd party                                │
│ kubetail                    │ minikube │ disabled   │ 3rd party (kubetail.com)                 │
│ kubevirt                    │ minikube │ disabled   │ 3rd party (KubeVirt)                     │
│ logviewer                   │ minikube │ disabled   │ 3rd party (unknown)                      │
│ metallb                     │ minikube │ disabled   │ 3rd party (MetalLB)                      │
│ metrics-server              │ minikube │ disabled   │ Kubernetes                               │
│ nvidia-device-plugin        │ minikube │ disabled   │ 3rd party (NVIDIA)                       │
│ nvidia-driver-installer     │ minikube │ disabled   │ 3rd party (NVIDIA)                       │
│ nvidia-gpu-device-plugin    │ minikube │ disabled   │ 3rd party (NVIDIA)                       │
│ olm                         │ minikube │ disabled   │ 3rd party (Operator Framework)           │
│ pod-security-policy         │ minikube │ disabled   │ 3rd party (unknown)                      │
│ portainer                   │ minikube │ disabled   │ 3rd party (Portainer.io)                 │
│ registry                    │ minikube │ disabled   │ minikube                                 │
│ registry-aliases            │ minikube │ disabled   │ 3rd party (unknown)                      │
│ registry-creds              │ minikube │ disabled   │ 3rd party (UPMC Enterprises)             │
│ storage-provisioner         │ minikube │ enabled ✅ │ minikube                                 │
│ storage-provisioner-gluster │ minikube │ disabled   │ 3rd party (Gluster)                      │
│ storage-provisioner-rancher │ minikube │ disabled   │ 3rd party (Rancher)                      │
│ volcano                     │ minikube │ disabled   │ third-party (volcano)                    │
│ volumesnapshots             │ minikube │ disabled   │ Kubernetes                               │
│ yakd                        │ minikube │ disabled   │ 3rd party (marcnuri.com)                 │
└─────────────────────────────┴──────────┴────────────┴──────────────────────────────────────────┘
```

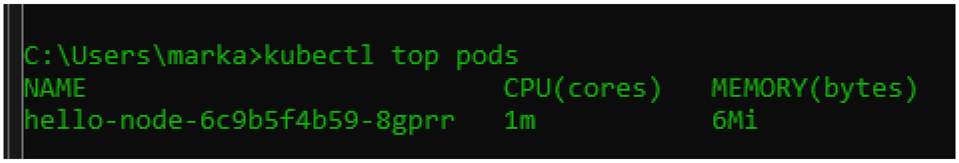
Enables the metrics server to check pod CPU/RAM usage. Then,

View Addon Pods & Services + *(metrics-server)*



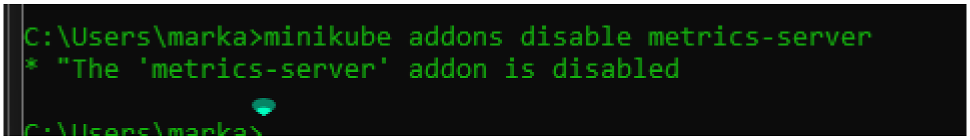
```
C:\Users\marka>minikube addons enable metrics-server
* metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  - Using image registry.k8s.io/metrics-server/metrics-server:v0.8.0
* The 'metrics-server' addon is enabled

C:\Users\marka>kubectl get pod,svc -n kube-system
NAME                                       READY   STATUS              RESTARTS      AGE
pod/coredns-66bc5c9577-b9g6c               1/1     Running             1 (13m ago)   26m
pod/etcd-minikube                          1/1     Running             1 (13m ago)   27m
pod/kube-apiserver-minikube                1/1     Running             1 (13m ago)   27m
pod/kube-controller-manager-minikube       1/1     Running             1 (13m ago)   27m
pod/kube-proxy-46wmc                       1/1     Running             1 (13m ago)   27m
pod/kube-scheduler-minikube                1/1     Running             1 (13m ago)   27m
pod/metrics-server-85b7d694d7-gfxkv        0/1     ContainerCreating   0             7s
pod/storage-provisioner                    1/1     Running             3 (13m ago)   27m

NAME                     TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)                  AGE
service/kube-dns         ClusterIP   10.96.0.10    <none>        53/UDP,53/TCP,9153/TCP   27m
service/metrics-server   ClusterIP   10.99.57.99   <none>        443/TCP                  7s
```

View Pod Metrics



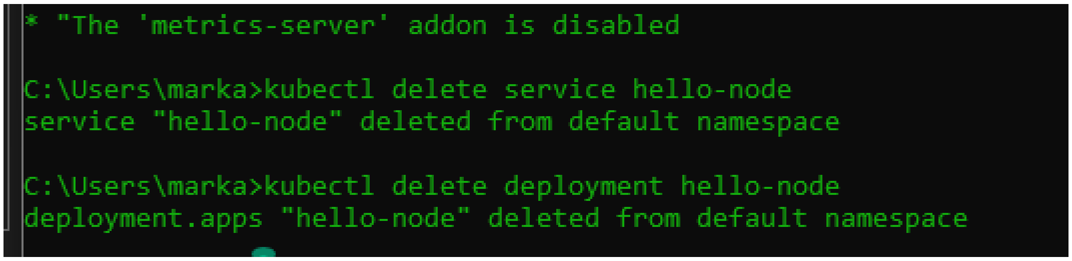
```
C:\Users\marka>kubectl top pods
NAME                         CPU(cores)   MEMORY(bytes)
hello-node-6c9b5f4b59-8gprr  1m           6Mi
```

Disable the Addon



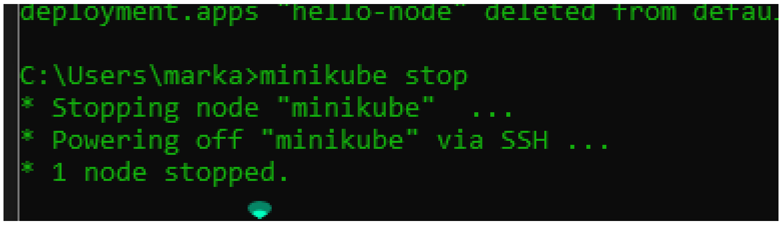
```
C:\Users\marka>minikube addons disable metrics-server
* "The 'metrics-server' addon is disabled

C:\Users\marka>
```

Clean Up Resources



```
* "The 'metrics-server' addon is disabled

C:\Users\marka>kubectl delete service hello-node
service "hello-node" deleted from default namespace

C:\Users\marka>kubectl delete deployment hello-node
deployment.apps "hello-node" deleted from default namespace
```

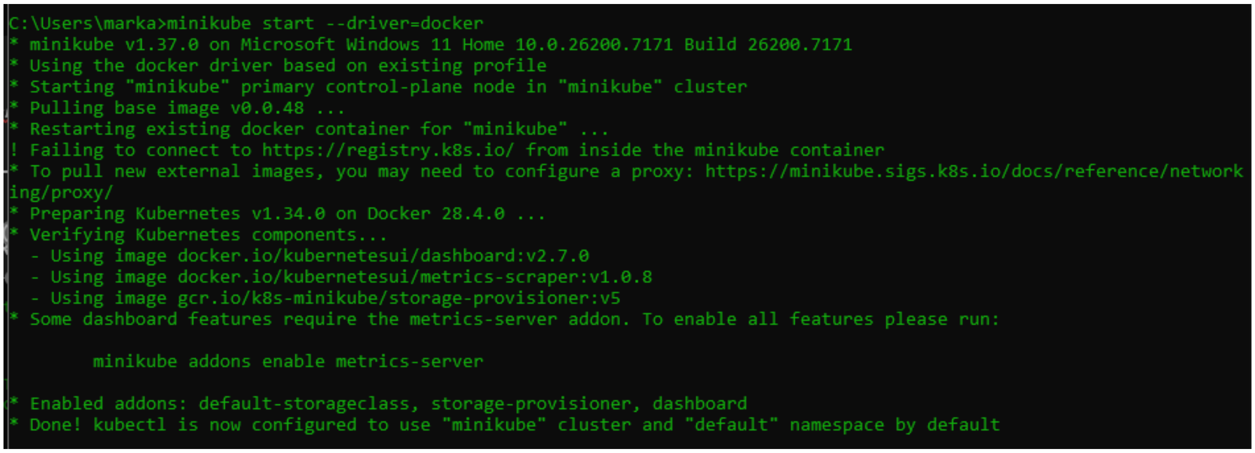
Stop the Minikube Cluster



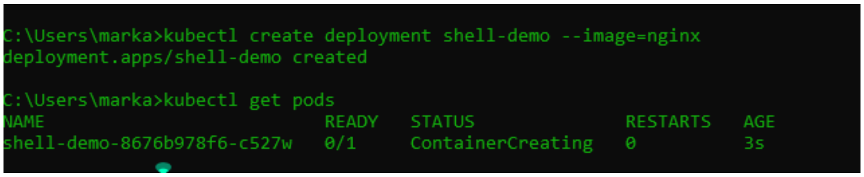
```
deployment.apps "hello-node" deleted from defau

C:\Users\marka>minikube stop
* Stopping node "minikube"  ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
```

*-- Hello Minikube **Done** --*

Get a Shell to a Running Container:

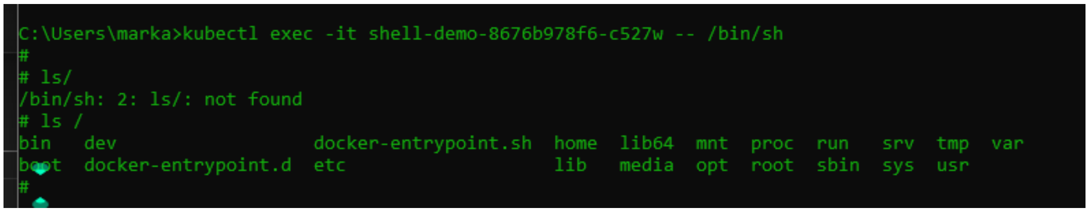


```
C:\Users\marka>minikube start --driver=docker
* minikube v1.37.0 on Microsoft Windows 11 Home 10.0.26200.7171 Build 26200.7171
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.48 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/network
ing/proxy/
* Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
* Verifying Kubernetes components...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

* Enabled addons: default-storageclass, storage-provisioner, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```



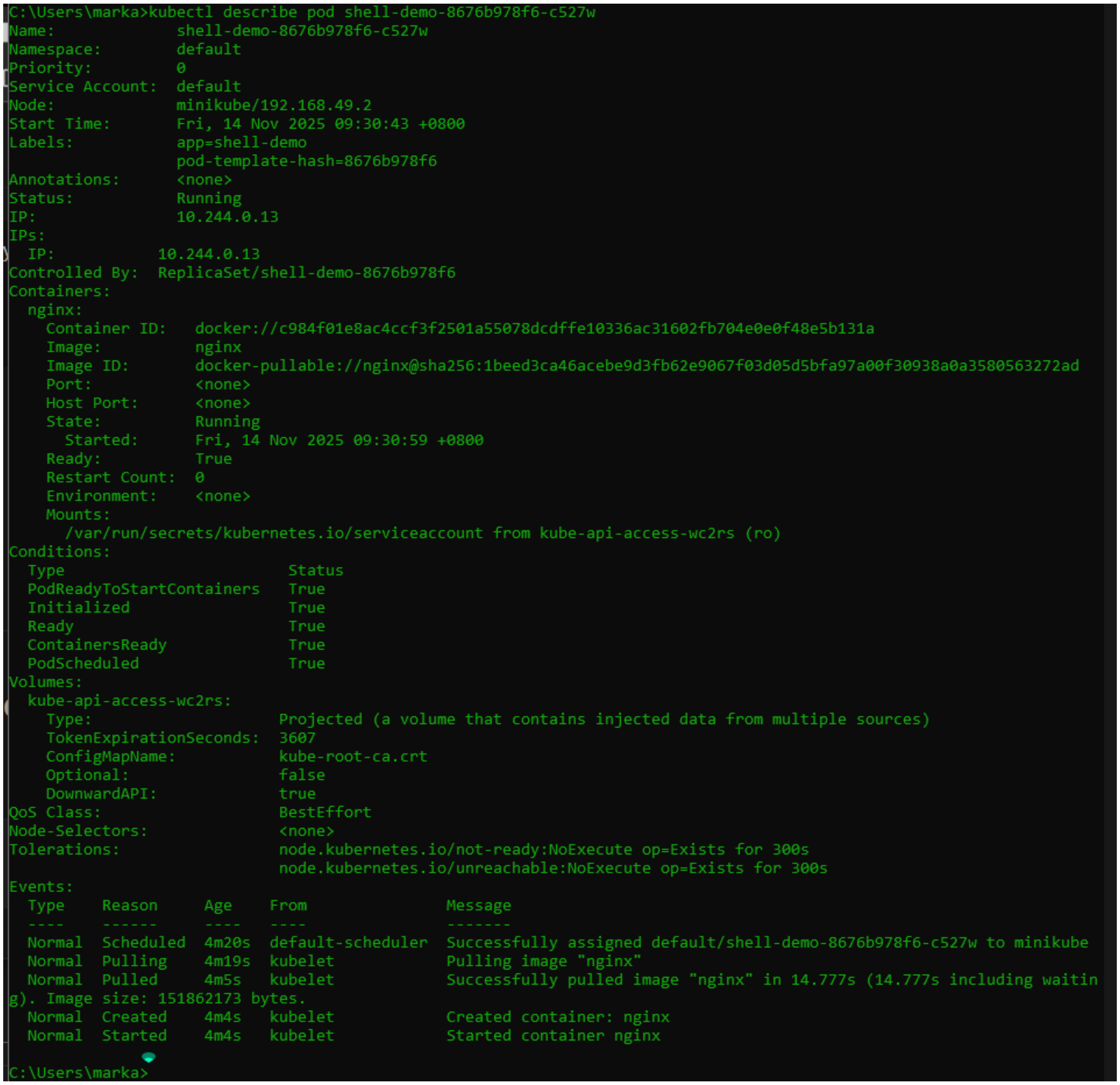
```
C:\Users\marka>kubectl create deployment shell-demo --image=nginx
deployment.apps/shell-demo created

C:\Users\marka>kubectl get pods
NAME                          READY   STATUS              RESTARTS   AGE
shell-demo-8676b978f6-c527w   0/1     ContainerCreating   0          3s
```

Execute into the Running Container



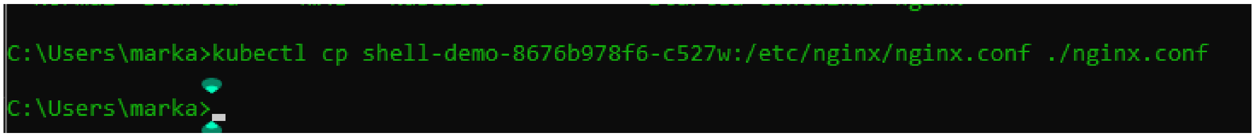
```
C:\Users\marka>kubectl exec -it shell-demo-8676b978f6-c527w -- /bin/sh
#
# ls/
/bin/sh: 2: ls/: not found
# ls /
bin   dev                   docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
#
```

Show container name *(For Debugging)*



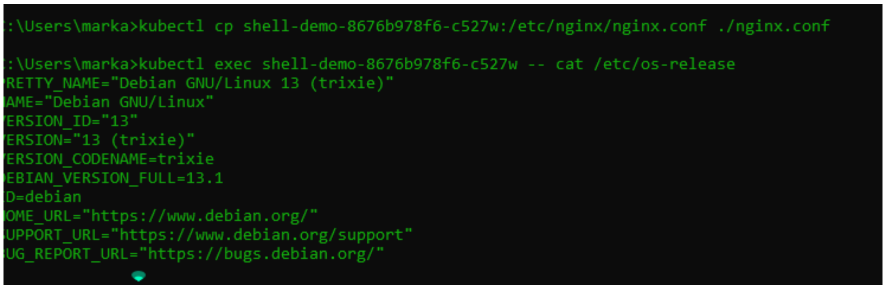
```
C:\Users\marka>kubectl describe pod shell-demo-8676b978f6-c527w
Name:             shell-demo-8676b978f6-c527w
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Fri, 14 Nov 2025 09:30:43 +0800
Labels:           app=shell-demo
                  pod-template-hash=8676b978f6
Annotations:      <none>
Status:           Running
IP:               10.244.0.13
IPs:
  IP:           10.244.0.13
Controlled By:  ReplicaSet/shell-demo-8676b978f6
Containers:
  nginx:
    Container ID:   docker://c984f01e8ac4ccf3f2501a55078dcdffe10336ac31602fb704e0e0f48e5b131a
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:1beed3ca46acebe9d3fb62e9067f03d05d5bfa97a00f30938a0a3580563272ad
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Fri, 14 Nov 2025 09:30:59 +0800
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-wc2rs (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-wc2rs:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  4m20s  default-scheduler  Successfully assigned default/shell-demo-8676b978f6-c527w to minikube
  Normal  Pulling    4m19s  kubelet            Pulling image "nginx"
  Normal  Pulled     4m5s   kubelet            Successfully pulled image "nginx" in 14.777s (14.777s including waitin
g). Image size: 151862173 bytes.
  Normal  Created    4m4s   kubelet            Created container: nginx
  Normal  Started    4m4s   kubelet            Started container nginx

C:\Users\marka>
```

kubectl cp (download files from inside the container)



```
C:\Users\marka>kubectl cp shell-demo-8676b978f6-c527w:/etc/nginx/nginx.conf ./nginx.conf

C:\Users\marka>
```

kubectl exec (show printed OS info**)**

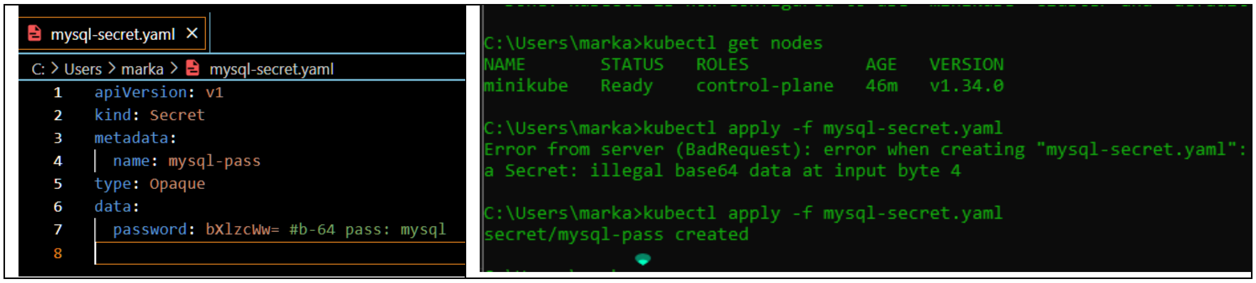


```
:\Users\marka>kubectl cp shell-demo-8676b978f6-c527w:/etc/nginx/nginx.conf ./nginx.conf

:\Users\marka>kubectl exec shell-demo-8676b978f6-c527w -- cat /etc/os-release
RETTY_NAME="Debian GNU/Linux 13 (trixie)"
AME="Debian GNU/Linux"
ERSION_ID="13"
ERSION="13 (trixie)"
ERSION_CODENAME=trixie
EBIAN_VERSION_FULL=13.1
D=debian
OME_URL="https://www.debian.org/"
UPPORT_URL="https://www.debian.org/support"
UG_REPORT_URL="https://bugs.debian.org/"
```

*-- Get a Shell to a Running Container: **Done** --*

Deploying Wordpress and MySQL with persistent volumes:

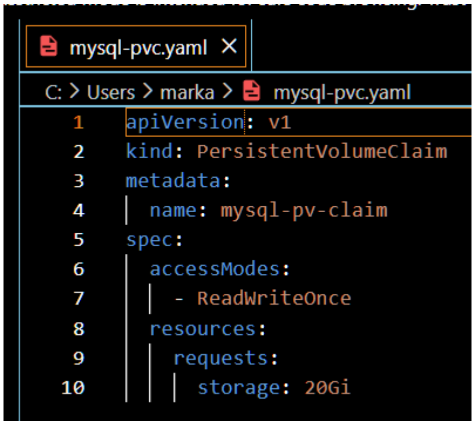
1. Create MySQL Secret



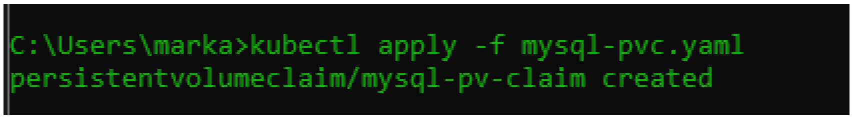
```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-pass
type: Opaque
data:
  password: bXlzcWw= #b-64 pass: mysql

```

```
C:\Users\marka>kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   46m   v1.34.0

C:\Users\marka>kubectl apply -f mysql-secret.yaml
Error from server (BadRequest): error when creating "mysql-secret.yaml":
a Secret: illegal base64 data at input byte 4

C:\Users\marka>kubectl apply -f mysql-secret.yaml
secret/mysql-pass created
```

2. Create the MySQL Persistent Volume Claim

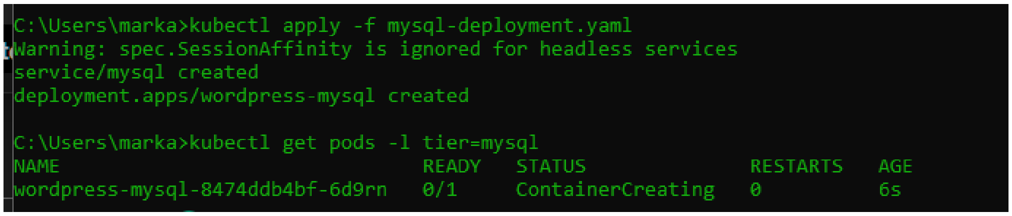


```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

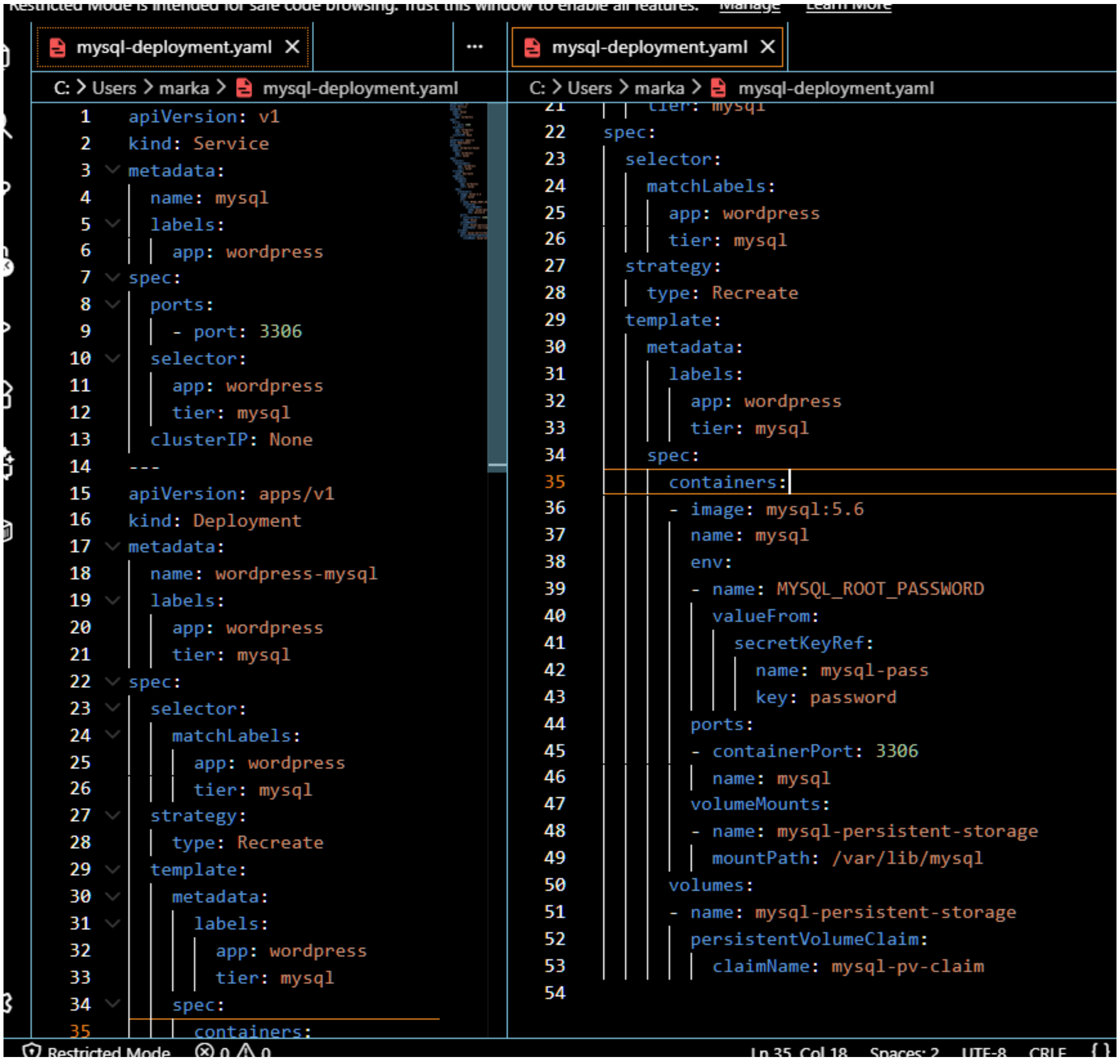


```
C:\Users\marka>kubectl apply -f mysql-pvc.yaml
persistentvolumeclaim/mysql-pv-claim created
```

3. Deploy MySQL (MySQL is Running)



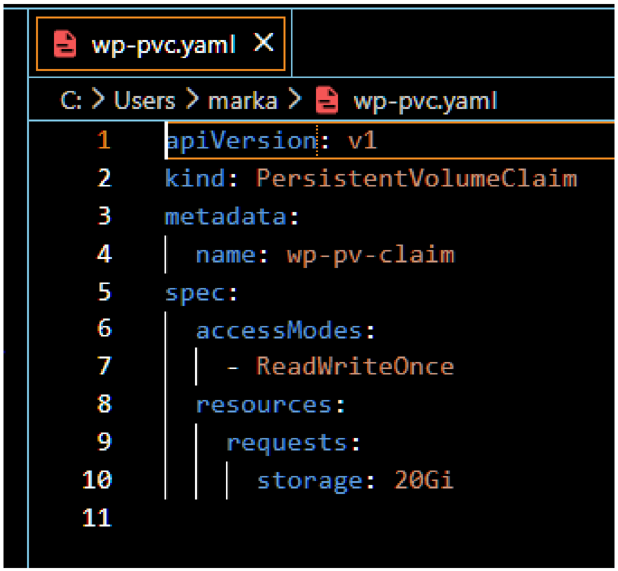
```
C:\Users\marka>kubectl apply -f mysql-deployment.yaml
Warning: spec.SessionAffinity is ignored for headless services
service/mysql created
deployment.apps/wordpress-mysql created

C:\Users\marka>kubectl get pods -l tier=mysql
NAME                               READY   STATUS              RESTARTS   AGE
wordpress-mysql-8474ddb4bf-6d9rn   0/1     ContainerCreating   0          6s
```


Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
mysql-deployment.yaml
mysql-deployment.yaml
C: > Users > marka > mysql-deployment.yaml
C: > Users > marka > mysql-deployment.yaml
Restricted Mode ⊗ 0 ⚠ 0
Ln 35, Col 18 Spaces: 2 UTF-8 CRLF {}

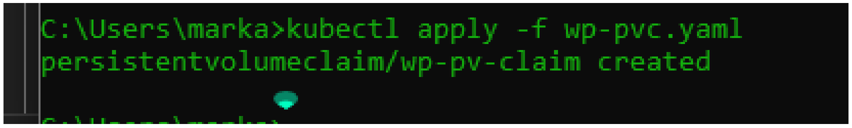
```yaml
apiVersion: v1
kind: Service
metadata:
  name: mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
    tier: mysql
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
      containers:
```

```yaml
    tier: mysql
spec:
  selector:
    matchLabels:
      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
          name: mysql
        volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: mysql-pv-claim
```

4. Create WordPress PVC (PVC Bound**)**

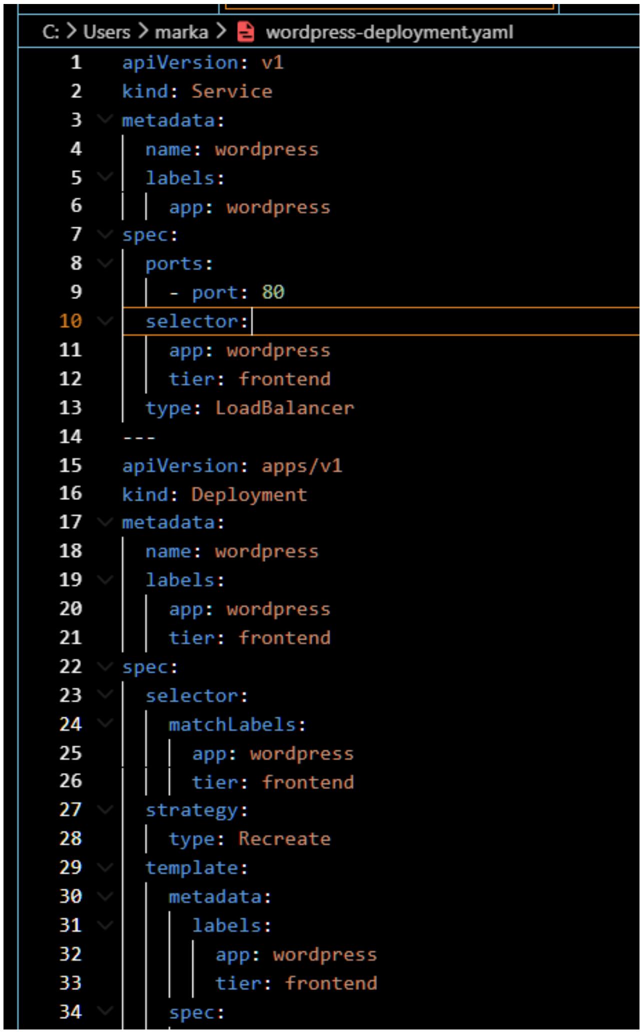


```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```



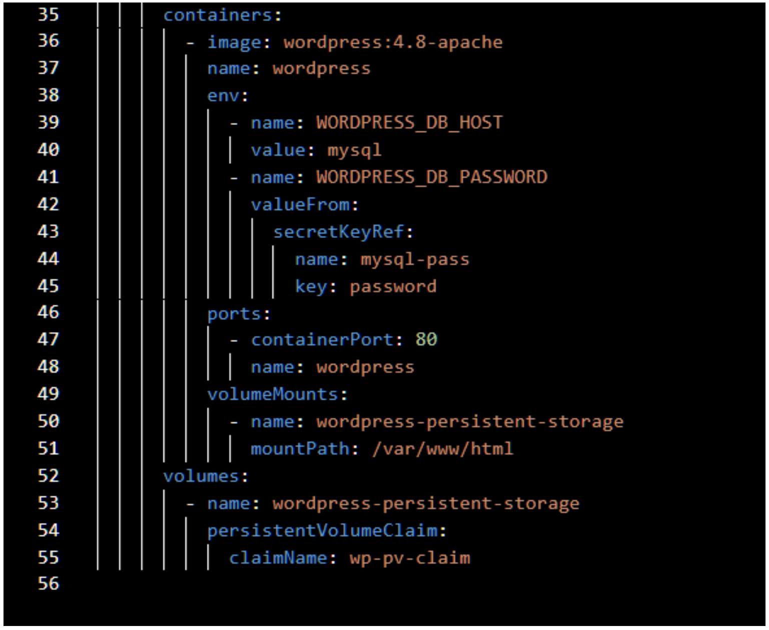
```
C:\Users\marka>kubectl apply -f wp-pvc.yaml
persistentvolumeclaim/wp-pv-claim created

C:\Users\marka>
```

5. Deploy WordPress (WordPress pod Running)

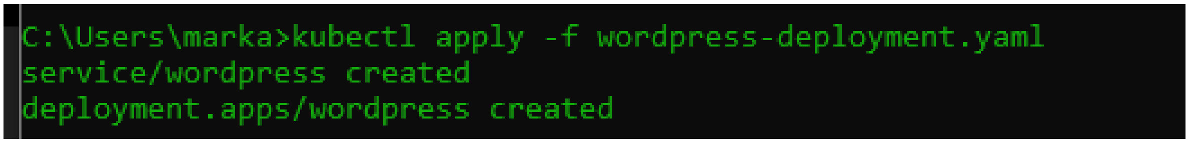


C: > Users > marka > wordpress-deployment.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
    tier: frontend
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: frontend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: frontend
    spec:
```

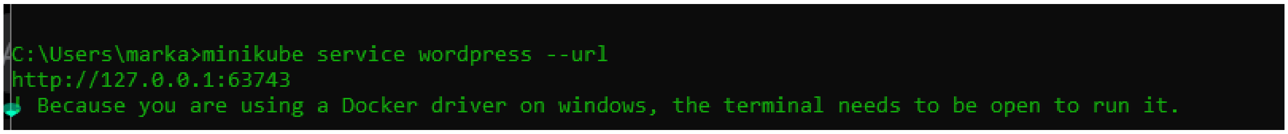


```yaml
      containers:
        - image: wordpress:4.8-apache
          name: wordpress
          env:
            - name: WORDPRESS_DB_HOST
              value: mysql
            - name: WORDPRESS_DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-pass
                  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
```

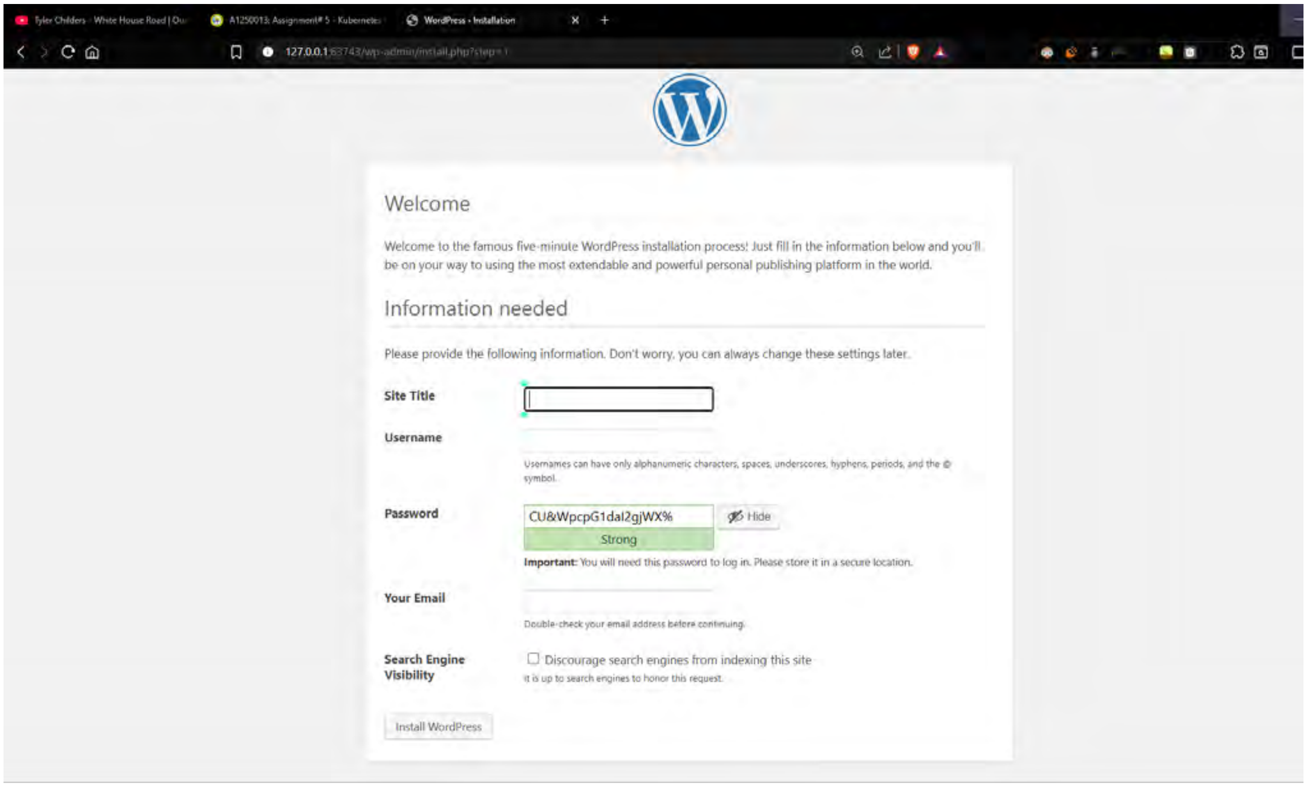


```
C:\Users\marka>kubectl apply -f wordpress-deployment.yaml
service/wordpress created
deployment.apps/wordpress created
```

6. Access WordPress (WordPress installation screen)



```
C:\Users\marka>minikube service wordpress --url
http://127.0.0.1:63743
❗ Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```



Tyler Childers - White House Road | Ou
A1250013: Assignment# 5 - Kubernetes
WordPress › Installation
127.0.0.1:63743/wp-admin/install.php?step=1

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

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: CU&WpcpG1dal2gjWX% Hide

Strong

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

Your Email

Double-check your email address before continuing.

Search Engine Visibility: □ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

-- Deploying Wordpress and MySQL with persistent volumes: Done --

Submitted by: ***Mark Angelo Siazon – IV-ACSAD***