

Multi-Tier Application Deployment



Kubernetes



WORDPRESS



MySQL®

Step 1: Creating a deployment YAML for MySQL and WordPress without actually applying it to the cluster.

```
root@master:~# kubectl create deployment mysql --image=docker.io/mysql --dry-run=client -o yaml > mysql.yaml
root@master:~# kubectl create deployment wordpress --image=docker.io/wordpress --dry-run=client -o yaml > wp.yaml
root@master:~#
```

Step 2: Edit both YAML file and add environment variables.

```
root@master:~# vi mysql.yaml
root@master:~# vi wp.yaml
root@master:~#
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: mysql
  name: mysql
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: mysql
    spec:
      containers:
      - image: docker.io/mysql
        name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: ajinkya
        - name: MYSQL_DATABASE
          value: K8s
      resources: {}
status: {}
```

◆ YAML file of the mysql

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: wordpress
  name: wordpress
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wordpress
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: wordpress
    spec:
      containers:
      - image: docker.io/wordpress
        name: wordpress
        env:
        - name: WORDPRESS_DB_HOST
          value: mysql
        - name: WORDPRESS_DB_NAME
          value: K8s
        - name: WORDPRESS_DB_PASSWORD
          value: ajinkya
        - name: WORDPRESS_DB_USER
          value: root
      resources: {}
status: {}
```

◆ YAML file of the WordPress

Step 3: Apply the yaml file of the mysql and wordpress

```
root@master:~# kubectl apply -f mysql.yaml
deployment.apps/mysql created
root@master:~# kubectl apply -f wp.yaml
deployment.apps/wordpress created
root@master:~#
```

Step 4: Fetching the details of pods using the following command.

Kubectl get pods

```
root@master:~# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
mysql-7d7cfc644-lcsd7              1/1     Running   0           48s
wordpress-68f6bc44cc-4cxdw        1/1     Running   0           42s
root@master:~# kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
mysql-7d7cfc644-lcsd7              1/1     Running   0           54s   10.36.0.1    worker    <none>            <none>
wordpress-68f6bc44cc-4cxdw        1/1     Running   0           48s   10.36.0.2    worker    <none>            <none>
root@master:~#
```

Step 5: Creating the service and give the port number to mysql and wordpress.

```
root@master:~# kubectl expose deployment mysql --port=3306
service/mysql exposed
root@master:~# kubectl expose deployment wordpress --port=80 --type=NodePort
service/wordpress exposed
root@master:~#
```

Step 6: Fetch the service of wordpress and mysql

```
root@master:~# kubectl get svc
NAME            TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes      ClusterIP     10.96.0.1       <none>           443/TCP          2d5h
mysql           ClusterIP     10.107.124.116  <none>           3306/TCP         9m13s
wordpress       NodePort      10.111.104.147  <none>           80:31837/TCP     8m43s
root@master:~#
```

Step 7: Fetch the deployments

```
root@master:~# kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
mysql         1/1     1            1           13m
wordpress     1/1     1            1           13m
root@master:~#
```

Step 8: Describe the service. Mysql and WordPress.

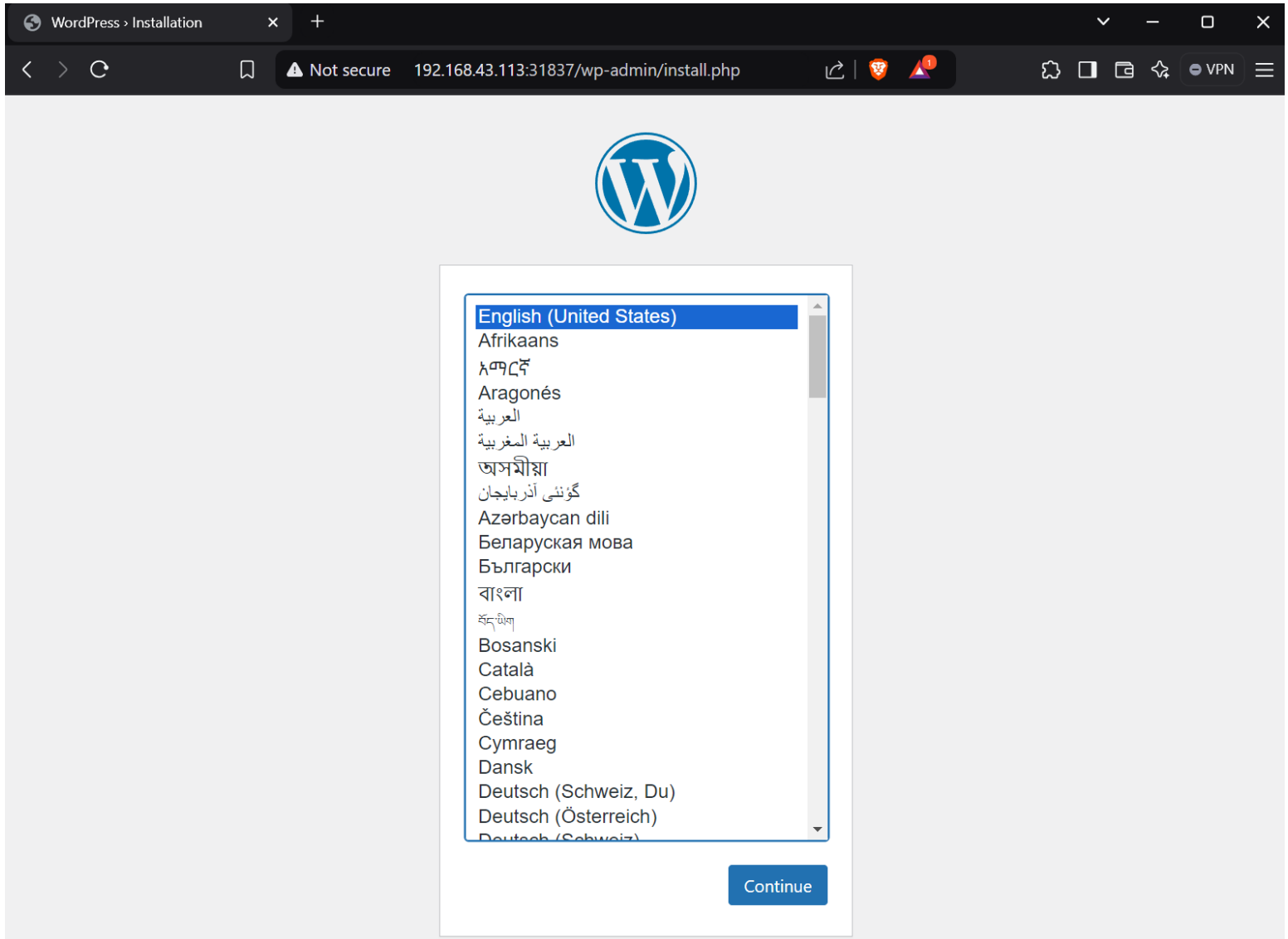
◆ Mysql

```
root@master:~# kubectl describe svc mysql
Name:          mysql
Namespace:     default
Labels:        app=mysql
Annotations:    <none>
Selector:      app=mysql
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.107.124.116
IPs:           10.107.124.116
Port:          <unset> 3306/TCP
TargetPort:    3306/TCP
Endpoints:     10.36.0.1:3306
Session Affinity: None
Events:        <none>
root@master:~#
```


◆ WordPress

```
root@master:~# kubectl describe svc wordpress
Name:          wordpress
Namespace:     default
Labels:        app=wordpress
Annotations:    <none>
Selector:      app=wordpress
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.111.104.147
IPs:           10.111.104.147
Port:          <unset> 80/TCP
TargetPort:    80/TCP
NodePort:      <unset> 31837/TCP
Endpoints:     10.36.0.2:80
Session Affinity: None
External Traffic Policy: Cluster
Events:        <none>
root@master:~#
```

Step 9: After entering the the system's IP with correct port number i.e NodePort described in the service.



◆ Enter the correct credential



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. Do not worry, you can always change these settings later.

Site Title

kubernetes

Username

ajinkya

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

Password

ajinkya@wp123

Hide

Medium

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

Your Email

ajinkyakale@gmail.com

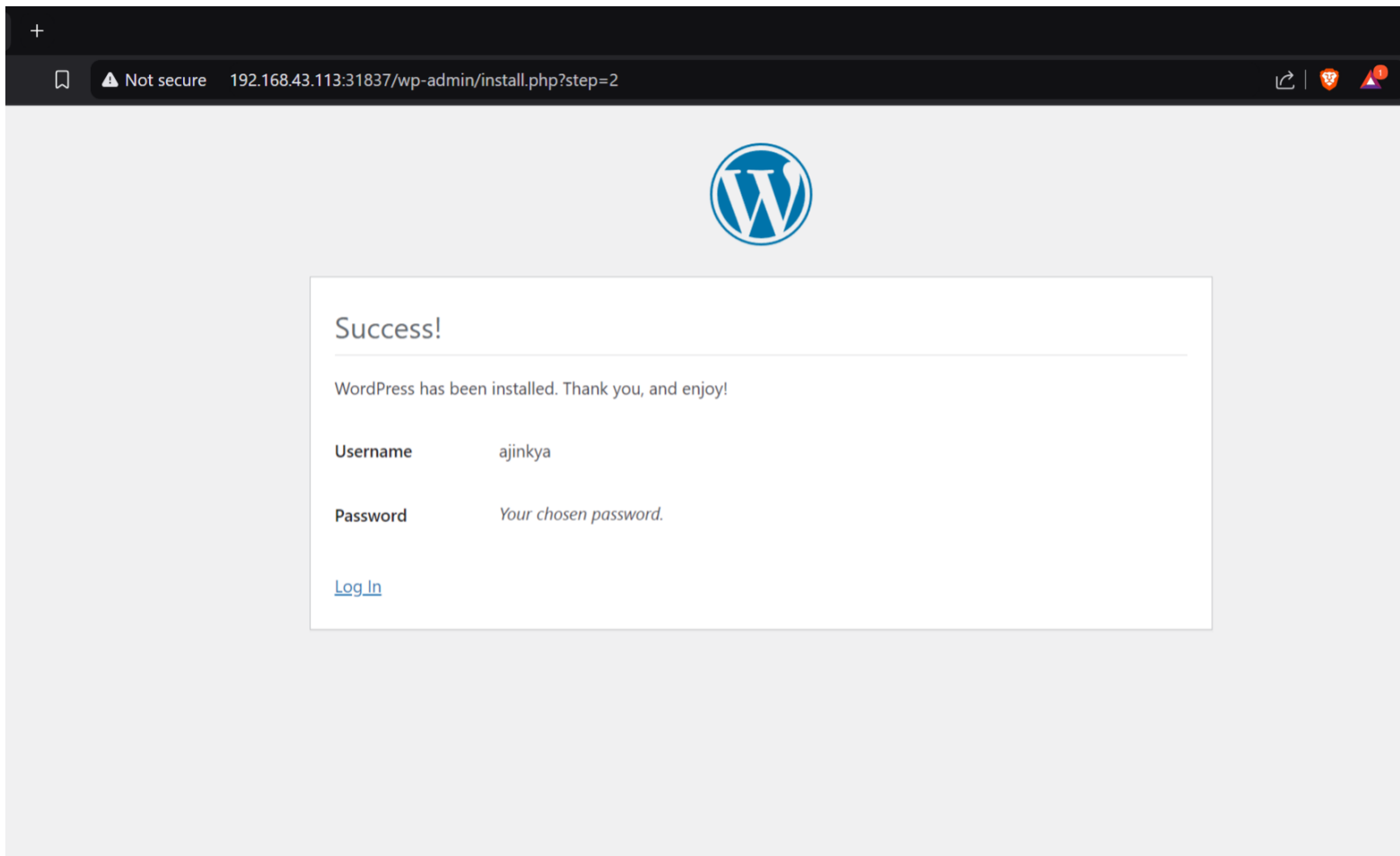
Double-check your email address before continuing.

Search engine visibility

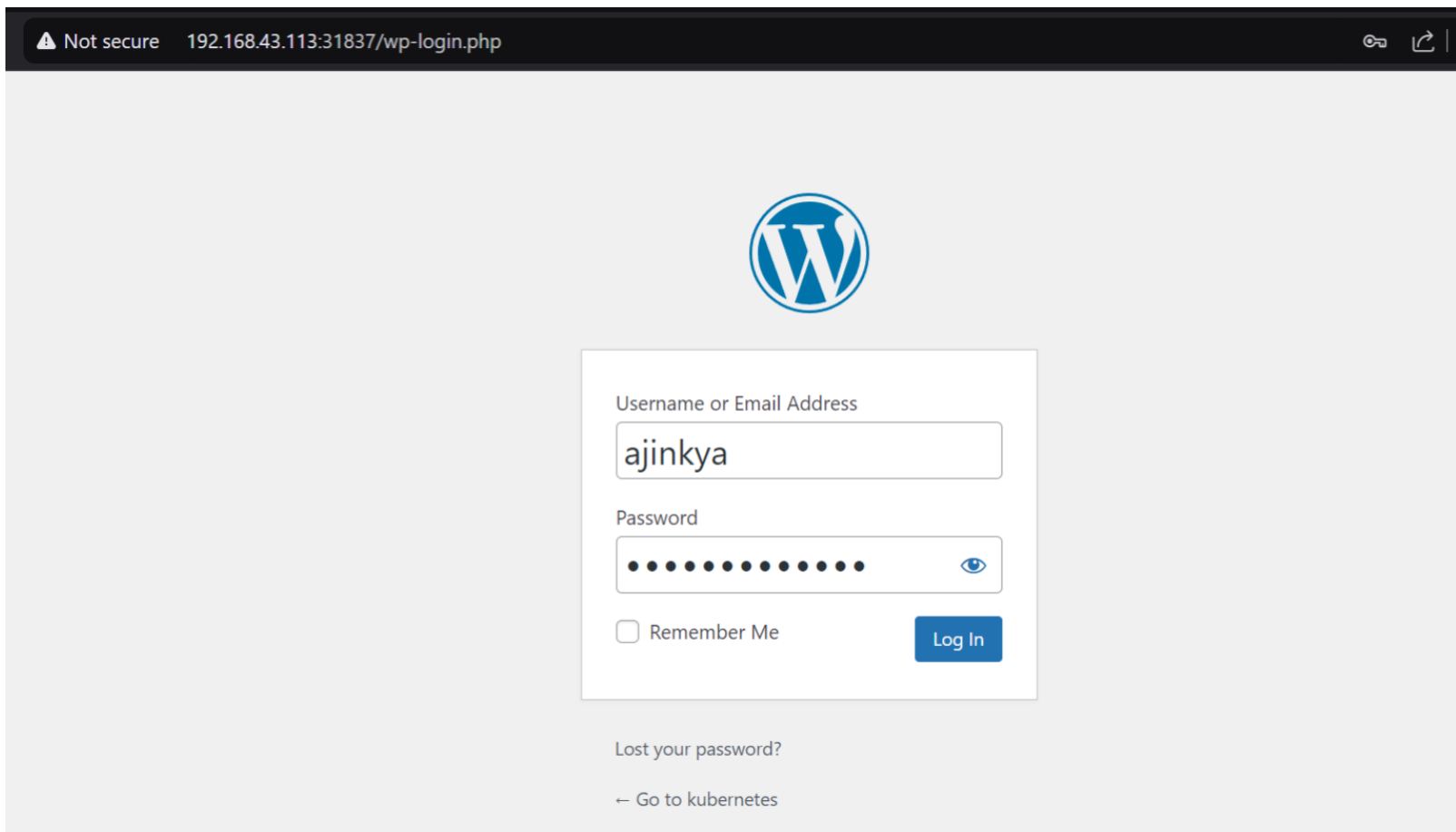
☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress



◆ Entering the correct login credentials.



◆ Welcome page of WordPress.

The screenshot shows the WordPress 6.6.2 admin dashboard. At the top, a navigation bar includes the WordPress logo, the site name 'kubernetes', and a 'New' button. Below this is a sidebar menu with options: Dashboard, Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and Collapse menu. The main content area features a large 'Welcome to WordPress!' banner with a link to 'Learn more about the 6.6.2 version.' Below the banner are three cards: 'Author rich content with blocks and patterns' (with a link to 'Add a new page'), 'Customize your entire site with block themes' (with a link to 'Open site editor'), and 'Switch up your site Styles' (with a link to 'Edit styles'). At the bottom, there are three widgets: 'Site Health Status' (showing 'No information yet...'), 'At a Glance' (showing '1 Post' and '1 Comment'), and 'Quick Draft' (with fields for 'Title' and 'Content', and a 'Save Draft' button). A dashed box on the right side of the dashboard indicates a drag-and-drop area for widgets.


Dashboard < kubernetes — WordPress 6.6.2

Not secure 192.168.43.113:31837/wp-admin/

Dashboard

Welcome to WordPress!


[Learn more about the 6.6.2 version.](#)



Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.


[Add a new page](#)



Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

[Open site editor](#)



Switch up your site Styles

Tweak your site, or give it a new look. Learn how about a new color palette.

[Edit styles](#)

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also [visit the Site Health screen](#) to gather information about your site now.

At a Glance

1 Post

1 Comment

Quick Draft

Title

Content

What's on your mind?

[Save Draft](#)

Drag & Drop Here

Ajinkya Kale