Experiment: Working K8S Local Dev With Kind

In the Foundation experiment/lab we installed Docker and KIND, validate our Docker installation, and now we'll use KIND.

Open a Windows Command Prompt

C:\> mkdir c:\projects

C:\> mkdir c:\projects\kind

C:\> cd c:\projects\kind

C:\projects\kind> kind create cluster

Administrator: Command Prompt

```
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
C:\Windows\system32>cd \projects\kind
C:\projects\kind>kind create cluster
Creating cluster "kind" ...
• Ensuring node image (kindest/node:v1.18.2) DD ...

☑ Ensuring node image (kindest/node:v1.18.2) ☑
• Preparing nodes 2 22

☑ Preparing nodes ☑ ☑
• Writing configuration 2 22 ...

☑ Writing configuration ☑ ☑☑
• Starting control-plane DDD

☑ Starting control-plane ☑☑☑
• Installing CNI 🛭 🕮 ...
☑ Installing CNI ☑ ☑
• Installing StorageClass 🛭 🕮 ...

☑ Installing StorageClass ☑ ☑
Set kubectl context to "kind-kind"
You can now use your cluster with:
kubectl cluster-info --context kind-kind
Have a nice day! 🛭 💵
C:\projects\kind>
```

Super simple, execute "kind create cluster", and go grab a refill on your favorite beverage while it spins for a few minutes depending on your network speed. Any delay in our initial cluster loading is in grabbing the images we need to build our cluster.

C:\projects\kind> kind get clusters

Kind

Check the kubectl client version

C:\>kubectl version --client

Client Version: version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.0", GitCommit:"e19964183377d0ec2052d1f1fa930c4d7575bd50", GitTreeState:"clean", BuildDate:"2020-08-26T14:30:33Z", GoVersion:"go1.15", Compiler:"gc", Platform:"windows/amd64"}

Check the kind cluster we just created

C:\> kubectl cluster-info --context kind-kind

Kubernetes master is running at https://127.0.0.1:51089 KubeDNS is running at https://127.0.0.1:51089/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

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

Notice we need to use the --context or set our default context to kind-kind

The newest installation of KIND automagically creates the ~/.kube/config file targeted at kind-kind, so you won't have to create that file. If your ~/.kube/config file is not created you will have to do that.

C:\> cd %USERPROFILE%

C:\Users\wcsadmin> mkdir .kube

C:\Users\wcsadmin> cd .kube

C:\Users\wcsadmin\.kube> notepad config

The file should look similar to below, and of course the server port reference will be from the information that was returned from your "kubectl cluster-info" apiVersion: v1

clusters:

- cluster:

certificate-authority-data:

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0I CQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSm wKY201bGRHVnpNQjRYRFRJd01Ea3dPREF3TWpJME0xb1hEVE13TURrd05qQXdN akkwTTFvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdE UVIKS29aSWh2Y05BUUVCQIFBRGdnRVBBRENDQVFvQ2dnRUJBTHJJCk9oYXpaZ FJNamx1WFh5aTQ0cFRBbkxPZEZ5QIJVb0FlRmZHUTIzRzd3UkJWMzNMaU5DUDR CMINvUTZUb20zejMKK2pydUZsUWhZK0grQ1A3UjByRGMwUmozbXdHM0dPckJYcn RBNzB6N2MrTUdMMm5QaEt0OTZ4Q0INdVIvZ1NUbgpIMVFuY0pkM3I1T3RGd1I2czd GbWxyb1RRNkFoVE5GZktBMTV2UHZ4Mmw2bEV3a0JGRIMrNjZ3dERPT2dJZy9CCkl wQkIzcWZ3bzYwbFRFUGorOEczbnBEYTdPNkVTNXNmZFIBZWJGTndlT21DL0cvdnN 2Yks4MXV0anNNM1oyWmgKNUExNkxnUlpNOXk2Rzl2blluYjNlekM2UWNMVG9YVm9 4YU95d0I0MFJIaXBIVkN3U2hKRG9qcjU0ZTBTWXIncQo5MlFpeE1GWEphUndVc1M0 eVU4Q0F3RUFBYU1qTUNFd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd 0VCCi93UUZNQU1CQWY4d0RRWUpLb1pJaHZjTkFRRUxCUUFEZ2dFQkFMVUxZMk

NXTVINSIJUb0dQZGVWSmQ3MmZoL24KblFGWHRYT3dQbldmMEISR0grYTVwOVcy ZytKRzQ4QjUwSUtZWFErVEZOTVZ0QUYrMDFWTFJRbzFQUEhudTM5NQppYlpHSn NybTBhWGVwOHFISWF2Lzl0MUxHcWk5dThkdFV6cjYvVEkwK2dUK2lEZlZ3bHRYSlZ iVXN4Q2U1VE5UCmdQTW9GWjBOdDVCTHUzRkdrV0x3OXRNSlkyU0JNckNZZ0ZhZ Hg5bkVSMjd4L0NsdjEvR3picEhWRmJmYVlubmQKUFNZbVRiWncxYlgyZzJBdC9pQ1 BSbDJGbXYzbXBTdENjaEppVDNkK0U5aEtpV2p4T0kvVDFsN2E5UWxJTXdpdApldjZo MlJkUTQvZVdkeTFxY2hDYkZ6N0UyZEtvSXRONFFseFhQcVlMQ1VnYzB4TEFGK2kx dkV1bDM3Zz0KLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQ0=

server: https://127.0.0.1:51089

name: kind-kind

contexts: - context:

cluster: kind-kind user: kind-kind name: kind-kind

current-context: kind-kind

kind: Config preferences: {}

users:

- name: kind-kind

user:

client-certificate-data:

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM4akNDQWRxZ0F3SUJBZ0IJ U3VyaXQxdGJKNnN3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQ XhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TURBNU1EZ3dNREI5TkROYUZ3MHINVE E1TURnd01ESXIORFJhTURReApGekFWQmdOVkJBb1REbk41YzNSbGJUcHRZWE4 wWlhKek1Sa3dGd1IEVIFRREV4QnJkV0psY201bGRHVnpMV0ZrCmJXbHVNSUICSW pBTkJna3Foa2lHOXcwQkFRRUZBQU9DQVE4QU1JSUJDZ0tDQVFFQXo1K1d0L242 RGtEbm5QblcKMFVmb3kxM1ArSldjQi9BMXdBTTU3L2RSREpIZ1g4N1lqOFRibkRDdU NhcElxVFY0ZiBqdW9JdDRDZmNKQUxMVqpXTWxhYVhncmljSXqxejdBdzlEUDVNRH M2VkdJcWRKaGc1Z0ROQWVYdTIzZUNuaUgxVDBnUFJxNjdKWHY4YW9VCi9YbIY3 UFdXNDVGbDlvd0NwSEc5Vlp4MUNIaVVOd1RVb09pd29oYzhpWmh6QzNHdS9CZ3F RbDR5K3hMVU0wTDkKWXZEY0NBU2dYdEhCQkU3TWN2SWxIQWFFdmdKVVk0c1 MvMUhZMvtZQmZVcCtzUzZjcHBKWFpveXhaTGdaMm51YgpTYmVhQ0J0Wk9wbGR3 YTgvV29SbGtrRkFpK0RURkt3cEZmTC9DYkxgOHd3QTVXdnNtWVdyWkFRZGE2aTB KQ2k5CmdaMzhCUUIEQVFBQm95Y3dKVEFPQmdOVkhROEJBZjhFQkFNQ0JhQXdF d1IEVIIwbEJBd3dDZ1IJS3dZQkJRVUgKQXdJd0RRWUpLb1pJaHZjTkFRRUxCUUFEZ 2dFQkFCOS9FL3o1cnNBeDNoaDVxVlo3ZUswTy95bUtES05SOFVDTApwc21kaVl2dm 1qZEVkc0l2WFhWM2dMeWQ1TmF0SSt2WU5rbS9YODlqNTlURXNVeGhtS2V2SEorQ mRZWXBPcEQ5CmY3Y3dBNnNIS2IEcGJUMVVuaiRoeHJoZvt6VEFXN3hleHZEam8v dG5xazg1dDR1bzVLdFVrRjRFNEN1QXQ3TFUKUjJaTlBta2UwdTRsUWgzaFU5Slg1bz Y5N3FHb21EMkZTeDdvb212eXdDMS8rbFBIVkNURk95U2pPbXEya001VApiZCtRbmh oeVVTRXJheC9sbnV5SDhkUlJYZkZWelk1WElLNS81RnBBMFFUUkltRXJUVExSem5 mcktNQXVaS3VTCjVFL0d4Z083MU1vZGN3ajM2NWZKWE9JT1c0UIZ4TXJabEpnQVZ la0lGUitQSUV0blErRT0KLS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQ0=

client-key-data:

LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUIFcEFJQkFBS0NBUUVB ejUrV3QvbjZEa0RublBuVzBVZm95MTNQK0pXY0IvQTF3QU01Ny9kUkRKSGdYODdZ Cmo4VGJuREN1Q2FwSXFUVjRmMGp1b0l0NENmY0pBTExWV01sYWFYZ3JpY0l4M Xo3QXc5RFA1TURzNIZHSXFkSmgKZzVnRE5BZVh1MjNIQ25pSDFUMGdQUnE2N0p Ydjhhb1UvWG5WN1BXVzQ1Rmw5b3dDcEhHOVZaeDFDSGIVTndUVQpvT2l3b2hjOG laaHpDM0d1L0JncVFsNHkreExVTTBMOVl2RGNDQVNnWHRIQkJFN01jdklsSEFhRXZ nSIVZNHNTCi8xSFkzK1lCZIVwK3NTNmNwcEpYWm95eFpMZ1oybnViU2JIYUNCdFp PcGxkd2E4L1dvUmxra0ZBaStEVEZLd3AKRmZML0NiTGo4d3dBNVd2c21ZV3JaQVFk YTZpMEpDaTlnWjM4QIFJREFRQUJBb0lCQUVzYmxFNWhvOC9jTXUxYQpoQmVaUit HcHdqNVBBTzd1T3NPSFowSWoyYklPWTNqRlB4cGpRSDYwTFlGWmxJZUJ6R0Zm Wk5PM0lHbWFjQ3RNCmhsbGtlY3pocC81aHZkMzcyWWY4MWZnT3dxVjAxVmQ1djh UM0ROR1puWTQwSklydEoxWkFrcFVJUW02cm51MXgKZGI1c2dMTUQ5TjNHRDNpd EZaZWZmYnFtcXIreEYvN2tTQ0ZKSEpyR0JoMk9GT3JSTXdxUXREaFBJWS9DTzN6 VQoybVBOTXN2Rk5JQjBoK2lKVXEvWEpwOHAvd3hmYkYva3FuRXBEYTZhaTFXT21 NdzhuQjFtRFZqSjR3TWtZdEkwCjZrTWFTdHYvVVBBeVIvZzFvV0ZTeGFGcGs4OEh6 UUp1ZnpXamZ6c2Z4bDN4ZnNwY1FKNzNZUIZzb29kRFdWTEUKRkwxVFRBRUNnW UVBODVVWkIMcHp3M0ISa21rQkJOT0VLYk5wWWNOZEYyOUNDOGNuYWJRMFdM NVIqUWEzdjBYMgo5YXdiVIIQVXBCck1wZUY0VINPalhSTWpIazkwZDIvM1dwRlkxdjJV UjBSTXdrYy9MVXNIVjJ6S2xUczZmanZFCi8vU0VTRm9FS2ZmWmkzbjAzUm81R3gvd WZxK1YwUnl6L0pNMzlKeDl5N0o5NHVHdE1FSEVCY1VDZ1lFQTJqVXoKK1VzSnRB Q2NwNkJ3TDNOV1h3U29aMk9nbzZnYjBOclpTUW9YbnNwdGt0UmF0R3NER3JTWG 1CeGJNRUloVkIERQpKbDBNTGc3QnFEaDcwVkIBbmNJWmQzQkhNd0NYMmgzNDN 6NmJZSTZ1RFBSU1k0amd4WHB0djBaVDdIV1ZDVFgwCk9TWnArYkpuMFpJb3gwSz BHc0FlcDk2WThIZnRReUoyU2FOendVRUNnWUVBbytzWFVONEluMk01RGdVWnIXe XEKQ1FJU2pkYll0NzVjZk42VjJGMkx5Smkzc0pmdnVZbFV5emo1NEE5cVh0RW1IUTI oVWpJOGNwczVTK0YrYUEzeQpNSUdWZm9DQmM0QTBBNTI4bHpkaGhtVFE0Nkp MRjc0VE1ZZ1VLVGhpaXZIZTcyeXY2c2NGM1FvZERpWU5OUDhTCjVJc1I0Y3dWaDF VWHdFSE1zYWZnU1YwQ2dZRUFpUmY2RW5zeG1uVHo5MkVXZXNsMUQzZW1zbV ptcTh2WHhnMXAKakxrWmcvdGNnbTZHbW1uTlpuN2w2M3lOVWpHS0xVUkZlSERuV VJ5V1VURkRvWXhxdExNWk92QkEyMnRZL0lldQpOWnhwRkc0d0xoVm1tZ0NLYjZmd Xdjald1MjVZZDVQOVg5YWhxRzZOU1o4Um5iZHlzbzZkZ0x1YXpTSWI0QjBMCndsSSt UTUVDZ1ICSTFrek5EeHhWUFVucERleXNnWmxlS3QwcDltTzhKWjlGcU5UM044eXY1 OEZnbitkSFU2WmYKWXZUNDdkK3FxdHRYREJEa1ZlWkJ4KzVSdW5kWlgxM3NKUk g2NUhVeE5QczBkdGVLa3RObnh5Zm9ubEFZTU13QgorWXBIMFVUT2xaK3FOT3d1U zFkZHM2MU14SVZWRIJxTDVzVGIVZGdkU3VLenU3ZW41R2hIY3c9PQotLS0tLUVOR CBSU0EgUFJJVkFURSBLRVktLS0tLQo=

Create a folder under your projects or profile folder and switch directory to our new folder

C:\projects\kind> mkdir kind-wp

C:\projects\kind> cd kind-wp

Create kustomization.yaml with the following content

C:\projects\kind\kind-wp> notepad kustomization.yaml

~/projects/kind/kind-wp \$ vi kustomization.yaml

Paste the following content into our file and save.

secretGenerator:

- name: mysql-pass

literals:

- password=VerySecure2020

resources:

- mysql-deployment.yaml
- wordpress-deployment.yaml

Alternatively you can clone the git repo for the class which has this file in the labs subfolder

\$> git clone https://github.com/GeorgeNiece/DevOpsForMicroservicesWithKubernetes-3day.git

The formatting of a YAML file is key to the stanzas. The indenting in a YAML file is how it groups the information, similar to Ruby coding.

Download the following two resource files to our c:\projects\kind\kind-wp or ~/projects/kind/kind-wp folders

\$ wget https://kubernetes.io/examples/application/wordpress/mysql-deployment.yaml

\$ wget https://kubernetes.io/examples/application/wordpress/wordpress-deployment.yaml

Copy or move the two resource YAML files that we downloaded to the c:\projects\kind\kind-wp folder, if it wasn't save there.

From the kind-wp folder run the kubectl apply

C:\projects\kind\kind-wp> kubectl apply -k ./

Run the following commands to inspect the deployment

C:\projects\kind\kind-wp> kubectl get secrets

C:\projects\kind\kind-wp> kubectl get pvc

C:\projects\kind\kind-wp> kubectl get pods

Once the "get pods" shows status as Running, rather than ContainerCreating we'll check the services for wordpress

C:\projects\kind\kind-wp> kubectl get services wordpress

Try accessing the normal default wordpress, this will result in a 404, and we'll correct that gap in the next update

open http://localhost:8080/

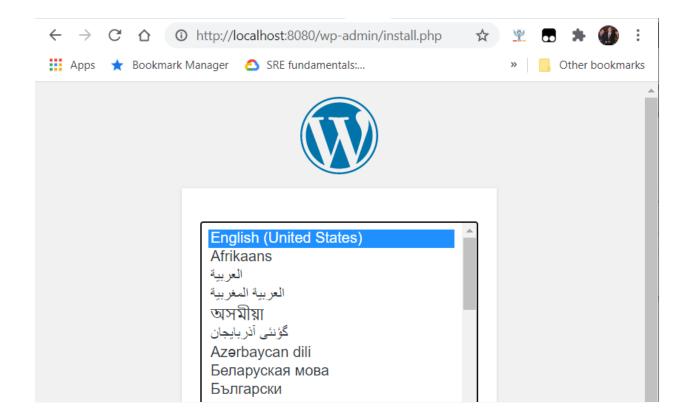
Note: We see that this is refused but recall that we see our service, but have not enabled a port forward to get to that WordPress instance

Create a port forward to allow us to access our Wordpress environment

~/projects/kind/kind-wp\$ kubectl port-forward svc/wordpress 8080:80

Now reopen or refresh

http://localhost:8080/



For MacOS

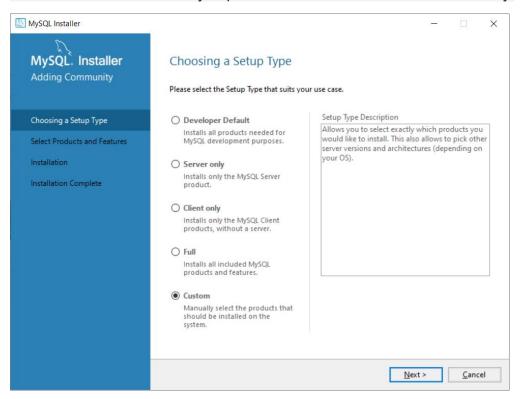
\$ brew install mysql-client

For Windows: Download the MySQL Shell if you don't already have it from the below URL

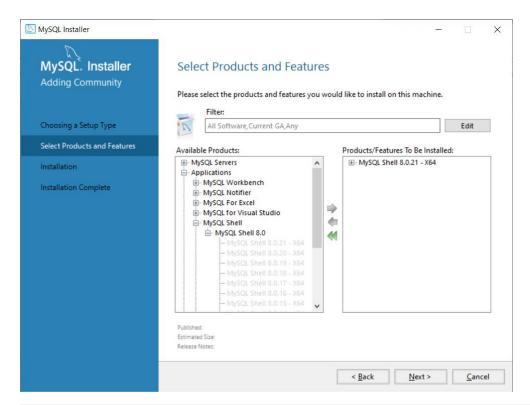
https://dev.mysql.com/downloads/shell/



If you have an Oracle account, you can login or you can select "No thanks, just download" Install the MySQL Shell, by running the downloaded installer Select the "Custom" unless you prefer to install the server and such locally.

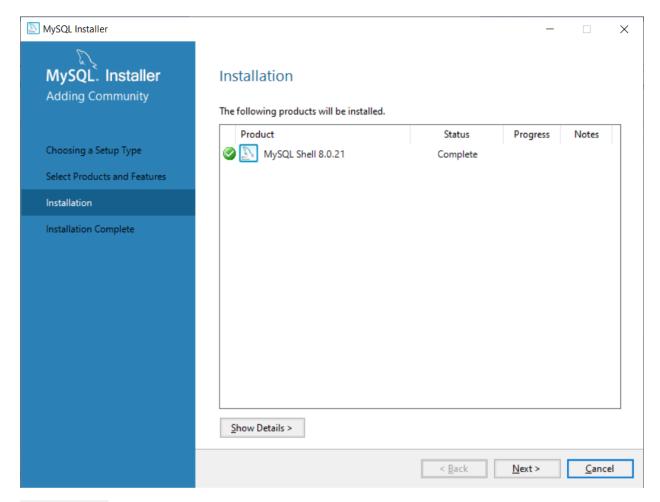


Select "Next"

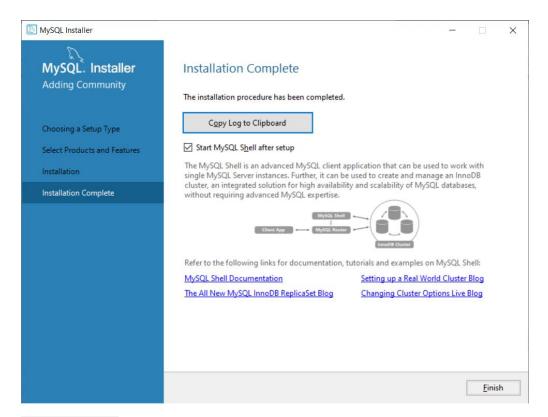


Expand the options under Applications -> MySQL Shell -> MySQL Shell 8.0 and then select MySQL Shell 8.0.21, selecting the top arrow to add that to the Products/Features To Be Installed list.

Select "Execute" to download the MySQL Shell



Select "Next"



Select "Finish"

Connectivity into our containerized DB

In a command shell check your pods and create a port forward for MySQL to connect to port 3306

~/projects/kind/kind-wp \$ kubectl get pods

C:\projects\kind\kind-wp> kubectl port-forward wordpress-mysql-65b8f6b6bd-bmf81

```
Administrator: Command Prompt - kubectl port-forward wordpress-mysql-65b8f6b6bd-bmf8l 3306:3306
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Windows\system32>kubectl get pods
                                   READY
                                           STATUS
                                                      RESTARTS
                                                                 AGE
NAME
wordpress-74984574d4-rw525
                                            Running
                                                                  36m
                                            Running 0
wordpress-mysql-65b8f6b6bd-bmf8l 1/1
                                                                 36m
::\Windows\system32>kubectl port-forward wordpress-mysql-65b8f6b6bd-bmf8l 3306:3306
Forwarding from 127.0.0.1:3306 -> 3306
Forwarding from [::1]:3306 -> 3306
```

From your MySQL Shell

Enter \connect root@localhost:3306

C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe

MySQL Shell 8.0.21

Copyright (c) 2016, 2020, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.

MySQL JS > \connect root@localhost:3306'

Creating a session to 'root@localhost:3306'

Please provide the password for 'root@localhost:3306': *************

Save password for 'root@localhost:3306'? [Y]es/[N]o/Ne[v]er (default No): Y

Fetching schema names for autocompletion... Press ^C to stop.

Your MySQL connection id is 5

Server version: 5.6.49 MySQL Community Server (GPL)

No default schema selected; type \use <schema> to set one.

You can review your **kustomization.yaml** to find the MySQL password set or if you used the default noted in this lab, it will be **VerySecure2020**

Enter \sql show databases;

The list of databases in MySQL is listed, including our WP DB

Enter \sql use wordpress;

This selects the WordPress DB that we've created in our deployment

Enter \sql show tables:

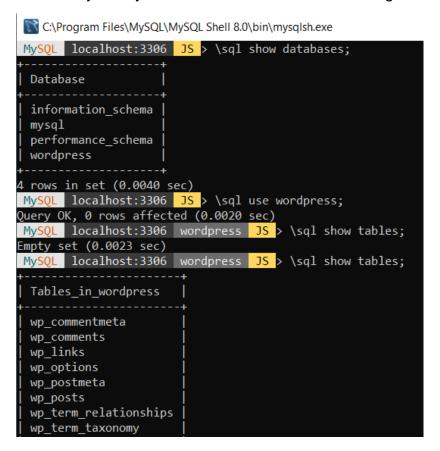
You'll notice that no tables are created, since we've not created our WorkPress instance, as yet.

Go back to the WordPress admin screen, fill in the information and select "Install WordPress"

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

Site Title	KIND-WP
Username	georgeniece Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the
Password	VerySecure2020 Medium Medium
	Important: You will need this password to log in. Please store it in a secure location.
Your Email	george.niece@digitaltransform 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	

Return to your MySQL Shell and look at the tables again in our WordPress DB



Lab complete, kill the port-forwarding session by Ctrl-C

Run the following command to delete your Secret, Deployments, Services and PersistentVolumeClaims:

From our kind-wp folder in a command prompt

C:\projects\kind\kind-wp> kubectl delete -k ./

```
Handling connection for 8080

Handling connection for 8080

C:\projects\kind\k8s-wp>kubectl delete -k ./
secret "mysql-pass-d2dmhcd6k7" deleted
service "wordpress-mysql" deleted
service "wordpress" deleted
deployment.apps "wordpress-mysql" deleted
deployment.apps "wordpress" deleted
persistentvolumeclaim "mysql-pv-claim" deleted
persistentvolumeclaim "wp-pv-claim" deleted

C:\projects\kind\k8s-wp>__
```

```
C:\projects\kind\kind-wp> kind help delete Deletes one of [cluster]
```

Usage:

kind delete [command]

Available Commands:

cluster Deletes a cluster

clusters Deletes one or more clusters

Flags:

-h, --help help for delete

Global Flags:

--loglevel string DEPRECATED: see -v instead

-q, --quiet silence all stderr output

-v, --verbosity int32 info log verbosity

Use "kind delete [command] --help" for more information about a command.

C:\projects\kind\kind-wp>kind delete cluster --help Deletes a resource

Usage:

kind delete cluster [flags]

Flags:

-h, --help help for cluster

--kubeconfig string sets kubeconfig path instead of \$KUBECONFIG or

\$HOME/.kube/config

--name string the cluster name (default "kind")

Global Flags:

--loglevel string DEPRECATED: see -v instead

-q, --quiet silence all stderr output-v, --verbosity int32 info log verbosity

C:\projects\kind\kind-wp> kind delete cluster --name Kind Deleting cluster "Kind" ...