Knowledge Check: WordPress Installation and troubleshooting

WordPress is a commonly used Blog and CMS engine that is an excellent introduction to a multi-tier application.

The WordPress example has two major components: A MySQL database to serve as the backing data store and the WordPress container that combines the Apache web server with PHP and the needed application dependencies to run an instance of the blog engine.

The manifests used in this example function as the bare minimum to provision an instance and should not be used in a production deployment. You are provided with 2 k8s manifest files (frontend.yaml and backend.yaml), use them to deploy WordPress and MySQL. Try to troubleshoot and fix the issue of why the pods may not be running.

---

1. Create the MySQL StatefulSet and its associated service.

$ kubectl create -f backend.yaml

.**NOTE:** The MySQL StatefulSet does not require a PVC to be created ahead of time for its storage. Instead, it uses the volumeClaimTemplates StatefulSet feature in combination with the default StorageClass provided by Docker Desktop to dynamically provision a volume.

2. Wait for the Pod to be up and running:

$ kubectl get pods --watch

3. With MySQL up and running, WordPress can now be provisioned.

$ kubectl create -f frontend.yaml

4. Wait for the Pods to be up and running:

$ kubectl get pods --watch

5. Now try to troubleshoot, correct the problem, and access the WordPress starting page

---

Clean Up

Cleanup when you are done.

$ kubectl delete -f frontend.yaml

$ kubectl delete -f backend.yaml