Revise and Test the Concepts

In this lesson, we will first revise and test our concepts through a short quiz.

Short recap of the chapter

We explored the simplest way to scale our Deployments and StatefulSets. It's simple because the mechanism is baked into Kubernetes. All we had to do is define a `HorizontalPodAutoscaler` with target memory and CPU. While this method for auto-scaling is commonly used, it is often not sufficient. Not all applications increase memory or CPU usage when under stress. Even when they do, those two metrics might not be enough.

In one of the following chapters, we'll explore how to extend `HorizontalPodAutoscaler` to use a custom source of metrics. For now, we'll destroy what we created, and we'll start the next chapter fresh.

Remove the previous resources

If you are planning to keep the cluster running, please execute the commands that follow to remove the resources we created.

```
# If NOT GKE or AKS

helm delete metrics-server \
    --namespace metrics


kubectl delete namespace metrics
```

```
kubectl delete ns go-demo-5
```

Otherwise, please delete the whole cluster if you created it only for the purpose of this chapter and you're not planning to dive into the next chapter right away.

Test your concepts with quiz

1   `HorizontalPodAutoscaler` collects information about used resources (memory and CPU) of nodes and Pods.

COMPLETED 0%   1 of 9   ‹   ›

In the next chapter, we will see how to Auto-Scale the nodes of Kubernetes Cluster.