Create a Cluster

This lesson focuses on creating a cluster and the necessary requirements and gists for this chapter.

Pulling the code

You know the drill. We'll move into the directory with the vfarcic/k8s-specs repository, we'll pull the latest version of the code just in case I pushed something recently, and we'll create a new cluster unless you already have one at hand.

> 🔍 All the commands from this chapter are available in the 07-logging.sh Gist.

```
cd k8s-specs

git pull
```

Gists and specifications

Choose the flavor you want and run the commands from its `.sh` file to create the cluster and the required specifications needed in this chapter.

> **NOTE:** In the end, you will see a command to `DELETE` the cluster too. Don't execute that command. Use the `DELETE` command only when you need to delete the cluster, preferably at the end of the chapter.

This time, the requirements for the cluster changed. We need much more memory than before. The main culprit is ElasticSearch which is very resource hungry.

If you're using **Docker For Desktop** or **minikube**, you'll need to increase the memory dedicated to the cluster to **10 GB**. If that's too much for your laptop, you might choose to read the [Exploring Centralized Logging Through](#) without running the examples or you might have to switch to one of the Cloud providers (AWS, GCP, or Azure).

In the case of **EKS** and **AKS**, we'll need bigger nodes. For EKS we'll use **t2.large** and for AKS **Standard_B2ms**. Both are based on **2 CPUs** and **8 GB RAM**.

GKE requirements are the same as before.

On top of new requirements, it should be noted that we do NOT need `Prometheus` in this chapter, so I removed it from the Gists.

Feel free to use one of the Gists that follow to create a new cluster, or to validate that the one you're planning to use meets the requirements.

GKE

- [gke-monitor.sh](#): **GKE** with 3 n1-standard-1 worker nodes, **nginx Ingress**, and cluster IP stored in environment variable **LB_IP**
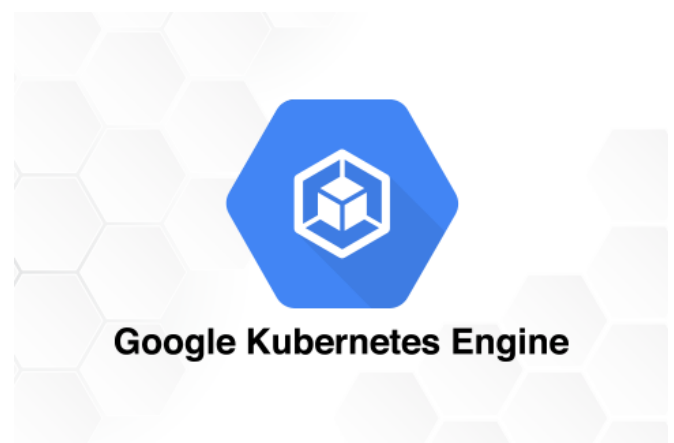
EKS

- [eks-logging.sh](#): **EKS** with 3 t2.large worker nodes, **nginx Ingress**, **Metrics Server**, **Cluster Autoscaler**, and cluster IP stored

in environment variable **LB_IP**

AKS

- aks-logging.sh: **AKS** with 3 Standard_B2ms worker nodes, **nginx Ingress**, and cluster IP stored in environment variable **LB_IP**

Docker for Desktop

- docker-logging.sh: **Docker for Desktop** with **2 CPUs, 10 GB RAM**, **nginx Ingress**, **Metrics Server**, and cluster IP stored in environment variable **LB_IP**

Minikube

- minikube-logging.sh: **minikube** with **2 CPUs, 10 GB RAM**, **ingress**, **storage-provisioner**, **default-storageclass**, and **metrics-server** addons enabled, and cluster IP stored in environment variable **LB_IP**

Now that we have a working cluster, we'll explore how to use logs through `kubectl`. That will provide a base for more comprehensive solutions that follow. Let's see that in the next lesson.