

**Kubernetes Dev Workshop – Required preparation before session**  
**Setup for User-Supplied Environment Option**

Version 1.3 - 5/21/23

Provided by Tech Skills Transformations

**Note: This option is provided as an alternative to running VirtualBox and using the pre-configured VM. Because of the variability of individual environments, there is no guarantee that all labs will work as expected with a user-supplied environment. If you prefer to use a pre-configured VM via VirtualBox, see the doc for setting up that environment here.**

This option assumes you are doing all the installs for the applications needed in the workshop (again in advance of the workshop). **In each of the bullet items below pertaining to an application, there are hyperlinks to install information for the application. You will still need to pick the right install packages and follow instructions in those links for your platform (Windows, Mac, Linux).**

1. Install a version of [Git](#). (If you are installing on a Windows system, it is recommended to also install the Git Bash Shell for Windows.)
2. Install [Docker](#) (any edition is fine). (Note on Windows or Mac, you may need to install the [Docker desktop](#).)
3. Install and run a Kubernetes cluster using any package/application you want such as [minikube](#), [kind](#), etc. (It is not recommended to rely on a cloud instance of Kubernetes due to possible internet bandwidth limitations.) You will need to be able to create namespaces, delete resources, etc. in this cluster so ensure you have this type of admin access to the cluster you will use. To complete lab 4, you'll also need to be able to have a persistent volume named "**mysql-pv**" created. Details on how to do that will vary depending on the underlying cluster implementation.
4. You only need a single node for your Kubernetes instance.
5. The workshop was designed for at least **Kubernetes version 1.23**, though other versions may work.
6. Install the Kubernetes command line tool, [kubectl](#).
7. Install [Helm 3](#). Helm 2 will not work.
8. Install [Kustomize](#)
9. Clone down the workshop git repository below to your system.

```
$ git clone https://github.com/skillrepos/k8s-dev.git
```

10. Install the monitoring pieces.

There is a script in the k8s-dev/monitoring directory called setup-monitoring.sh that may do this for you. Run it.

There are two pieces of information it will output that you will need to capture for use later:

- a. A token to use for logging into the Kubernetes dashboard. It will be shown in the output after this line: " ---- TOKEN to use for logging into dashboard follows ----"
- b. The initial password to use to log into Grafana. It will be shown in the output after this line: " --- Grafana initial password follows ---"

After this runs, you should have a namespace named "kubernetes-dashboard" with the dashboard components running in it and a "monitoring" namespace with the Prometheus and Grafana pieces running in it.

If the script does not work, you can look at the script to see the commands it is trying to run and work through them or reference the alternative methods below. You only need to use the alternative methods if the script does not work.

Alternative ways to install monitoring pieces:

a. Kubernetes dashboard:

1. . Use the following command: (ref <https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>)

```
$ kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy
/recommended.yaml
```

2. We need to supply some rbac setup (service account, etc.) to allow access. In the k8s-dev/monitoring directory is the rbac spec that you can just apply as follows:

```
$ cd ~/k8s-dev/monitoring
$ k apply -f dashboard-rbac.yaml
```

b. Prometheus and Kubernetes

To be able to access these, we need to install the Prometheus and Grafana stack. It's fairly easy to do with Helm. Run the commands below:

```
$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

```
$ helm repo update
```

```
$ helm install -n monitoring monitoring --version="38.0.3" prometheus-community/kube-prometheus-stack
```

11. Ensure you have access to a text editor that you are comfortable with and know how to use and that you can access easily.

12. Optional - install [meld](#) for your platform if running MacOS or Linux and you don't already have it. Meld is a visual diffing and merging tool that we will be using in some labs. You can install a different tool if you prefer or, if you have to, rely on standard OS tools like diff and using an editor to merge.

You should now be setup for the class!

If you would like to have a printed copy of the labs, you can print them out and bring them with you. They are available from [this github location](#).