**NFJS Beyond Kubernetes Extended Workshop – Required preparation before session**

Version 1.2 - 7/3/22

Provided by Tech Skills Transformations

Thank you for signing up for the Beyond Kubernetes workshop. We're excited that you'll be joining us and want to help you get the most out of your session. With that in mind, please read through the preparation steps below to ensure you are ready to go on the day of the session. The WIFI setup at the conference location will NOT allow you to do all the prep at that location, so you need to ensure that if you are attending at the location, the setup of the environment is done prior to your arrival where you can download large images and come to the workshop without needing to do that.

This workshop is a combination of lecture and hands-on labs. It assumes you have a basic working knowledge of Kubernetes. (There will not be an opportunity to provide substantial foundational Kubernetes knowledge given the time constraints.) To do the labs, you will need a modern laptop to use during the workshop. Since the workshop will be all day, you should make sure to bring your charger. In advance of the workshop, you should ensure that your laptop is setup to use one of the environments below - either the virtual machine image with VirtualBox or an environment that you configure yourself with Docker, Kubernetes, Helm, Tekton, ArgoCD and Istio as outlined.

TO-DO:

* Ensure that you have a laptop (with charger) that is powerful enough to run either VirtualBox or an environment you setup according to the specifications.
* Prior to coming to the class, setup either the VirtualBox environment or the User-supplied environment. Details are below. You only need one - not both.

**Option 1: VirtualBox environment**

**Note: This environment is the simplest to use/setup if you are able to run VirtualBox and load/run the VM. Note that if you encounter problems with that, often it may be due to running another application on your machine that is already using the virtualization subsystem, such as Docker for Desktop or Parallels. You can sometimes get things to work by stopping those applications and any part of them that is running on your system. In some cases, a reboot afterwards may help - remembering to stop/not restart the other applications that may use the virtualization subsystems. You can also Google for particular error messages you get when trying to run VirtualBox.** [**This document**](https://github.com/skilldocs/extra/blob/main/vbtroubleshoot.pdf) **has some other pointers that may help if you are running into issues.**

**Alternatively, if you cannot get VirtualBox or the VM to run well on your system, then you can use option 2, a user-supplied environment, as detailed further down in this doc.**

1. Install the [VirtualBox](https://www.virtualbox.org/) application from VirtualBox.org

2. Ensure that the VirtualBox application starts and runs on your system.

Note: If you encounter an error starting VirtualBox, solutions to many common problems can be found via Googling the error along with the Operating System you're running on. Some troubleshooting tips are also included in a doc at <https://github.com/brentlaster/safaridocs/blob/main/vbtroubleshoot.pdf>

3. Download the pre-configured VirtualBox image (.ova file) for the class from either of the two locations below. (Note that this file is over 3G in size, so it may take a while.):

* + - <https://www.dropbox.com/s/sia45gdlsfg9gma/k8s-ps.ova?dl=0>
    - <https://bclconf.s3.us-west-2.amazonaws.com/k8s-ps.ova>

4. Reference the setup doc at <https://github.com/skilldocs/beyond-k8s/blob/main/beyond-k8s-setup.pdf> for details on how to proceed with loading the file into VirtualBox if you need those.

5. Start the virtual machine in VirtualBox (see setup doc) and enable networking via the "arrows" icon in the upper right (see screenshot below).

Graphical user interface, text, application

Description automatically generated

6. Open a "Terminal Emulator" session on the desktop and start the virtual machine with the command below. Wait for this to complete.

$ ﻿sudo minikube start --vm-driver=none

7. Get the latest workshop files.

$ cd ~/beyond-k8s

$ git pull

8. Now run the script below. This will setup ArgoCD, Tekton, Helm, Istio, etc. and pre-pull images needed for the workshop, so they are present already on the machine and don't have to be downloaded again later.

$ ﻿~/beyond-k8s/extra/setup.sh

(This will also output the ArgoCD password at the end of the script. You will need this in one of the labs. If you don't see it, the process may not be ready and you can run the script again and it will likely show up. (You can also get this again later if you need it.))

8. After running the script, you should see a set of namespaces like the list below. (On this machine, we have aliased "kubectl" to just "k".

﻿$ k get ns

NAME STATUS AGE

argocd Active 3m3s

default Active 27m

istio-system Active 3m27s

kube-node-lease Active 27m

kube-public Active 27m

kube-system Active 27m

tekton-pipelines Active 3m

9. After going through these steps, you can Close and Power Off the machine, but leave it loaded in VirtualBox in prep for the class.

10. NOTE: As an additional check, you can power it back on once to make sure it comes back up as expected.

11. OPTIONAL: If the machine seems to be slow or taking up too many resources on your system, you can adjust the amount of memory, etc. via right-clicking on the machine in the list in VirtualBox, byselecting Settings and then adjusting memory, etc. as shown in the example below.

Graphical user interface, text, application, email

Description automatically generated

**Option 2: User-Supplied Environment**

**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.**

This option assumes you are doing all of 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)\*.

(\* If you are using Linux as your platform, you may be able to utilize the setup script from the VM at <https://github.com/skillrepos/beyond-k8s/blob/main/extra/setup.sh> to get some pieces installed, but it is not guaranteed and will not install all of the applications.)

1. Install a version of [Git](http://git-scm.org/). (If you are installing on a Windows system, it is recommended to also install the Git Bash Shell for Windows.)

2. Install [Docker](https://docs.docker.com/get-docker/) (any edition is fine). (Note on Windows or Mac, you may need to install the [Docker desktop](https://www.docker.com/products/docker-desktop/).)

3. Install and run a Kubernetes cluster using any package/application you want such as [minikube](https://minikube.sigs.k8s.io/docs/start/), [kind](https://kind.sigs.k8s.io/), 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.

4. You only need a single node for your Kubernetes instance.

5. The workshop was designed for at least **Kubernetes version 1.21**, though other versions may work.

6. Install the Kubernetes command line tool, [kubectl](https://kubernetes.io/docs/tasks/tools/).

7. Install [Helm 3](https://helm.sh/docs/intro/install/). Helm 2 will not work.

8. Install [Kustomize](https://kubectl.docs.kubernetes.io/installation/kustomize/binaries/)

9. Install [Istio](https://istio.io/latest/docs/setup/install/) in your cluster.

(See also https://istio.io/latest/docs/setup/getting-started/)

10. Install [ArgoCD](https://argo-cd.readthedocs.io/en/stable/operator-manual/installation/) in your cluster.

11. Install [Tekton](https://tekton.dev/docs/getting-started/tasks/) core pieces in your cluster.

12. Install the [Tekton dashboard](https://tekton.dev/docs/dashboard/install/) in your cluster.

13. Install the [Tekton CLI](https://github.com/tektoncd/cli).

14. Clone down the workshop git repository below to your system.

$ git clone <https://github.com/skillrepos/beyond-k8s.git>

15. Execute the following script to prepull images that will be needed during the class.

$ ./beyond-k8s/extra/image-prepull.sh

16. Make sure that you can run both the “tkn” and “argocd” command line applications from a terminal session.

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

18. Optional - install [meld](https://meldmerge.org/) 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.

19. When you are done, you should have namespaces in your cluster for "istio-system", "tekton-pipelines", and "argocd" with the various kubernetes resources for each application running in them.

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.](https://github.com/skilldocs/beyond-k8s/blob/main/beyond-k8s-labs.pdf)