

To Create a Cluster or Not to Create a Cluster?

In this lesson, we will discuss if we should create a cluster with jx or install Jenkins X inside an existing cluster.

We'll cover the following

- Create a cluster using jx or install Jenkins in an existing cluster?
- `jx create cluster help`
- `jx install --help`

As I mentioned already, Jenkins X is much more than just a tool for *continuous integration* or *continuous delivery*. One of the many features it has is to create a **fully operational Kubernetes cluster** and install the **tools** we might need to operate it efficiently. On the other hand, `jx` allows us to install Jenkins X inside an existing Kubernetes cluster as well.

So, we need to make a choice.

Create a cluster using `jx` or install Jenkins in an existing cluster?

The decision will largely depend on your current situation, as well as the purpose of the cluster.

If you plan to create a cluster only for the purpose of the exercises in this course, I recommend using `jx` to create a cluster, assuming that your favorite hosting vendor is one of the supported ones. Another reason for letting `jx` handle the creation of the cluster lies in the potential desire to have a dedicated cluster for continuous delivery. In both cases, we can use the `jx create cluster` command.

On the other hand, you might already have a cluster where other applications are running and might want to add Jenkins X to it. In that case, all we have to do is install Jenkins X by executing `jx install`.

Let's go through both the `jx create cluster` and `jx install` commands and see what we've got.

what we've got.

jx create cluster help

Let's run the following command first:

```
jx create cluster help
```

Judging from the output, we can see that Jenkins X works with quite a few different providers. We can use it inside **Azure AKS**, **AWS with kops**, **AWS EKS**, **Google GKE**, **Oracle OKE**, **IBM ICP**, **IBM IKS**, **Minikube**, **Minishift**, **OpenShift**, and **Kubernetes**. Now, the last provider I mentioned is curious.

Aren't all other providers just different flavors of Kubernetes? If they are, why do we have a provider called `kubernetes` on top of more specific providers (e.g., GKE)?

The `kubernetes` provider allows us to run Jenkins X in (almost) any Kubernetes flavor. The difference between the `kubernetes` provider and all the others lies in additions that are useful only for those providers. Nevertheless, you can run Jenkins X in (almost) any Kubernetes flavor. If your provider is on the list, use it. Otherwise, pick the `kubernetes` provider instead.



As a side note, do not trust the list I presented as being final. By the time you read this, Jenkins X might have added more providers to the list.

It is worth mentioning that `jx` cannot create a cluster in all those providers, but that it can run there. A few of those cannot be created dynamically, namely, OKE, ICP, and OpenShift. If you prefer one of those, you'll have to wait until we reach the part with the instructions to install Jenkins X in an existing cluster.

You'll also notice that `jx` will install some of the local dependencies if you do not have them already on your laptop. Which ones will be installed depends on your choice of the provider. For example, `gcloud` is installed only if you choose GKE as your provider. On the other hand, `kubectl` will be installed no matter the choice, as long as you do not have it already on your laptop.

So, if you do choose to use one of the cloud providers, `jx create cluster` is an excellent option, unless you already have a cluster where you'd like to install

Jenkins X. If that's the case, or if you cannot use one of the cloud providers, you should be exploring the `jx install` command instead. It is a subset of `create cluster`. If we take a look at the supported providers in the `install` command, we'll see that they are the same as those we saw in `create cluster`.

```
jx install --help #
```

Now let's execute the following command:

```
jx install --help | grep "provider="
```

The output is as follows.

```
--provider='': Cloud service providing the Kubernetes cluster. Supported providers: aks, aws, eks
```

I'll show you how to use Jenkins X to create a GKE, EKS, and AKS cluster. If you do have access to one of those providers, I suggest you follow the instructions. Even if you're already planning to install Jenkins X inside an existing cluster, it would be helpful to see the benefits we get with the `jx create cluster` command. Further on, I'll show you how to install Jenkins X inside any existing cluster. It's up to you to choose which path you'd like to follow. Ideally, you might want to try them all, and get more insight into the differences between cloud providers and Kubernetes flavors.

No matter the choice, I will make sure that all four combinations are supported through the rest of the chapters. We'll always start with the instructions on how to create a new cluster in GKE, EKS, and AKS, as well as how to install Jenkins X and everything else we might need in an existing cluster.

Before we proceed, please note that we'll specify most of the options through arguments. We could have skipped them and let `jx` ask us questions like how many nodes do you want? Nevertheless, I believe that using arguments is a better way since it results in a documented and reproducible process to create something. Ideally, `jx` should not ask us any questions. We can indeed accomplish that by running in the batch mode. I'll reserve that for the next chapter, and our first attempt will use a combination of arguments and questions.

Also, before we dive into the actual setup, we'll create a file `myvalues.yaml`.

```
echo "nexus:"
```

Among other things, Jenkins X installs Nexus where we can store our libraries. While that is useful for many projects, we will not need it in the examples that follow. By disabling it, we'll avoid reserving resources we won't use.

For your convenience, bookmarks to the relevant sub-chapters are as follows.

- [Creating A Google Kubernetes Engine \(GKE\) Cluster With jx](#)
- [Creating An Amazon Elastic Container Service for Kubernetes \(EKS\) Cluster With jx](#)
- [Creating An Azure Kubernetes Service \(AKS\) Cluster With jx](#)
- [Is Your Existing Cluster Ready For Jenkins X?](#)



You'll notice that some of the text in the previous sections is repeated. That's intentional since I wanted each set of instructions to be self-sufficient, even though such repetition might annoy those who try multiple Kubernetes flavors or want to experience both the process of creating the cluster as well as installing Jenkins X inside an existing one.

If you prefer to create the cluster in one of the other providers, I recommend reading the instructions for one of the “big three” (AWS, Azure, or Google), since the requirements and the steps are very similar.

In the next lesson, we will look at the method to create a **GKE** cluster.