Creating a Cluster: Creating S3 Bucket and Installing kops

In this lesson, we will take our first steps towards creating a cluster by creating an S3 bucket and installing kops.

WE'LL COVER THE FOLLOWING

Naming the Cluster

We'll start by deciding the name of our soon to be created cluster. We'll choose to call it `devops23.k8s.local`. The latter part of the name (`.k8s.local`) is mandatory if we do not have a DNS at hand. It's a naming convention kops uses to decide whether to create a gossip-based cluster or to rely on a publicly available domain.

If this would be a "real" production cluster, you would probably have a DNS for it, or if you have your own DNS and you are planning to use it to set-up cluster, there will be minor additional steps to the things we have done so far, please refer to Kops for exact steps.

However, since we cannot be sure whether you do have one for the exercises in this course, we'll play it safe, and proceed with the gossip mode.

We'll store the name in an environment variable so that it is easily accessible.

```
export NAME=devops23.k8s.local
```

When we create the cluster, kops will store its state in a location we're about to configure. If you have previously used Terraform, you'll notice that kops uses a very similar approach.

kops uses the state it generates when creating the cluster for all subsequent operations. If we want to change any aspect of a cluster, we'll have to change the desired state first, and then apply those changes to the cluster.

Creating an S3 Bucket

At the moment, when creating a cluster in AWS, the only option for storing the state are Amazon S3 buckets. We can expect availability of additional stores soon. For now, S3 is our only option.

The command that creates an S3 bucket in our region is as follows.

```
export BUCKET_NAME=devops23-$(date +%s)

aws s3api create-bucket \
    --bucket $BUCKET_NAME \
    --create-bucket-configuration \
    LocationConstraint=$AWS_DEFAULT_REGION
```

We created a bucket with a unique name and the output is as follows.

```
{
    "Location": "http://devops23-1519993212.s3.amazonaws.com/"
}
```

For simplicity, we'll define the environment variable `KOPS_STATE_STORE`. Kops will use it to know where we store the state. Otherwise, we'd need to use `--store` argument with every `kops` command.

```
export KOPS_STATE_STORE=s3://$BUCKET_NAME
```

Installing kops

There's only one thing missing before we create the cluster. We need to install kops.

MacOS

If you are a **MacOS user**, the easiest way to install `kops` is through Homebrew.

```
brew update && brew install kops
```

As an alternative, we can download a release from GitHub.

```
curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.gith

chmod +x ./kops

sudo mv ./kops /usr/local/bin/
```

Linux

If, on the other hand, you're a **Linux user**, the commands that will install `kops` are as follows.

```
wget -O kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.githu

chmod +x ./kops

sudo mv ./kops /usr/local/bin/
```

Windows

Finally, if you are a **Windows user**, there is no kops installer. Please download `kops-windows-amd64` from kops releases, rename it to `kops.exe`, and make sure that the directory with the binary is in `PATH` variable. You'll know that it works if you can execute `kops version`.

In the next lesson, we'll spend a bit of time discussing the specifications we might want our cluster to have.