Giving Others Access to the Cluster

In this lesson, we will allow other users to access our cluster.

Creating a Distributable Configuration

Unless you're planning to be the only person in your organization with the access to the cluster, you'll need to create a `kubectl` configuration that you can distribute to your coworkers.

Let's see the steps.

```
cd cluster
mkdir -p config
export KUBECONFIG=$PWD/config/kubecfg.yaml
```

We went back to the `cluster` directory, created the sub-directory config, and exported `KUBECONFIG` variable with the path to the file where we'd like to store the configuration.

Now we can execute `kops export`.

```
kops export kubecfg --name ${NAME}
cat $KUBECONFIG
```

The **output** of the latter command is as follows.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: ...
    server: https://api-devops23-k8s-local-ivnbim-609446190.us-east-2.elb.amazonaws.com
```

```
    name: devops23.k8s.local
contexts:
- context:

    cluster: devops23.k8s.local
    user: devops23.k8s.local
  name: devops23.k8s.local
current-context: devops23.k8s.local
kind: Config
preferences: {}
users:
- name: devops23.k8s.local
  user:
    as-user-extra: {}
    client-certificate-data: ...
    client-key-data: ...
    password: oeezRbhG4yz3oBUO5kf7DSWcOwvjKZ6l
    username: admin
- name: devops23.k8s.local-basic-auth
  user:
    as-user-extra: {}
    password: oeezRbhG4yz3oBUO5kf7DSWcOwvjKZ6l
    username: admin
```

Now you can pass that configuration to one of your coworkers, and he'll have the same access as you.

Truth be told, you should create a new user and a password or, even better, an SSH key and let each user in your organization access the cluster with their own authentication.

You should also create RBAC permissions for each user or a group of users. We won't go into the steps how to do that since they are already explained in the *Securing Kubernetes Clusters* chapter.

In the next lesson, we will destroy the cluster we created.