In order to make use of Kubernetes' ability to encrypt sensitive data at rest, you need to provide Kubernetes with an encrpytion key using a data encryption config file. This lesson walks you through the process of creating a encryption key and storing it in the necessary file, as well as showing how to copy that file to your Kubernetes controllers. After completing this lesson, you should have a valid Kubernetes data encryption config file, and there should be a copy of that file on each of your Kubernetes controller servers.

Here are the commands used in the demo.

Generate the Kubernetes Data encrpytion config file containing the encrpytion key:

```
ENCRYPTION_KEY=$(head -c 32 /dev/urandom | base64)

cat > encryption-config.yaml << EOF
kind: EncryptionConfig
apiVersion: v1
resources:
  - resources:
      - secrets
    providers:
      - aescbc:
          keys:
            - name: key1
              secret: ${ENCRYPTION_KEY}
      - identity: {}
EOF
```

Copy the file to both controller servers:

```
scp encryption-config.yaml cloud_user@<controller 1 public ip>:~/
scp encryption-config.yaml cloud_user@<controller 2 public ip>:~/
```