The Kubernetes API server provides the primary interface for the Kubernetes control plane and the cluster as a whole. When you interact with Kubernetes, you are nearly always doing it through the Kubernetes API server. This lesson will guide you through the process of configuring the kube-apiserver service on your two Kubernetes control nodes. After completing this lesson, you should have a `systemd` unit set up to run kube-apiserver as a service on each Kubernetes control node.

You can configure the Kubernetes API server like so:

```
sudo mkdir -p /var/lib/kubernetes/

sudo cp ca.pem ca-key.pem kubernetes-key.pem kubernetes.pem \
  service-account-key.pem service-account.pem \
  encryption-config.yaml /var/lib/kubernetes/
```

Set some environment variables that will be used to create the `systemd` unit file. Make sure you replace the placeholders with their actual values:

```
INTERNAL_IP=$(curl http://169.254.169.254/latest/meta-data/local-ipv4)
CONTROLLER0_IP=<private ip of controller 0>
CONTROLLER1_IP=<private ip of controller 1>
```

Generate the kube-apiserver unit file for `systemd`:

```
cat << EOF | sudo tee /etc/systemd/system/kube-apiserver.service
[Unit]
Description=Kubernetes API Server
Documentation=https://github.com/kubernetes/kubernetes

[Service]
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --audit-log-maxage=30 \\
  --audit-log-maxbackup=3 \\
  --audit-log-maxsize=100 \\
  --audit-log-path=/var/log/audit.log \\
  --authorization-mode=Node,RBAC \\
  --bind-address=0.0.0.0 \\
  --client-ca-file=/var/lib/kubernetes/ca.pem \\
  --enable-admission-plugins=Initializers,NamespaceLifecycle,NodeRestriction,LimitRanger,ServiceAccount,
                    DefaultStorageClass,ResourceQuota \\
  --enable-swagger-ui=true \\
  --etcd-cafile=/var/lib/kubernetes/ca.pem \\
  --etcd-certfile=/var/lib/kubernetes/kubernetes.pem \\
  --etcd-keyfile=/var/lib/kubernetes/kubernetes-key.pem \\
  --etcd-servers=https://$CONTROLLER0_IP:2379,https://$CONTROLLER1_IP:2379 \\
  --event-ttl=1h \\
  --experimental-encryption-provider-config=/var/lib/kubernetes/encryption-config.yaml \\
  --kubelet-certificate-authority=/var/lib/kubernetes/ca.pem \\
  --kubelet-client-certificate=/var/lib/kubernetes/kubernetes.pem \\
  --kubelet-client-key=/var/lib/kubernetes/kubernetes-key.pem \\
  --kubelet-https=true \\
  --runtime-config=api/all \\
  --service-account-key-file=/var/lib/kubernetes/service-account.pem \\
  --service-cluster-ip-range=10.32.0.0/24 \\
  --service-node-port-range=30000-32767 \\
  --tls-cert-file=/var/lib/kubernetes/kubernetes.pem \\
  --tls-private-key-file=/var/lib/kubernetes/kubernetes-key.pem \\
  --v=2 \\
 --kubelet-preferred-address-types=InternalIP,InternalDNS,Hostname,ExternalIP,ExternalDNS
 Restart=on-failure
 RestartSec=5
```

```
[Install]
WantedBy=multi-user.target
EOF
```