One of the necessary steps in setting up a new Kubernetes cluster from scratch is to assign permissions that allow the Kubernetes API to access various functionality within the worker kubelets. This lesson guides you through the process of creating a ClusterRole and binding it to the kubernetes user so that those permissions will be in place. After completing this lesson, your cluster will have the necessary role-based access control configuration to allow the cluster's API to access kubelet functionality such as logs and metrics.

You can configure RBAC for kubelet authorization with these commands. Note that these commands only need to be run on one control node.

Create a role with the necessary permissions:

```
cat << EOF | kubectl apply --kubeconfig admin.kubeconfig -f -
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: system:kube-apiserver-to-kubelet
rules:
  - apiGroups:
      - ""
    resources:
      - nodes/proxy
      - nodes/stats
      - nodes/log
      - nodes/spec
      - nodes/metrics
    verbs:
      - "*"
EOF
```

Bind the role to the kubernetes user:

```
cat << EOF | kubectl apply --kubeconfig admin.kubeconfig -f -
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: system:kube-apiserver
  namespace: ""
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: system:kube-apiserver-to-kubelet
subjects:
  - apiGroup: rbac.authorization.k8s.io
    kind: User
    name: kubernetes
EOF
```