

```
apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: "iti-devops"
5  spec:
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

 sudo      

```
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl apply -f nameSpace.yml
namespace/iti-devops created
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl get namespaces
NAME                STATUS    AGE
default             Active   29d
iti-devops          Active   14s
kube-node-lease     Active   29d
kube-public         Active   29d
kube-system         Active   29d
kubernetes-dashboard Active   26d
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab#
```

```
1 apiVersion: v1
2 kind: ServiceAccount
3 metadata:
4   name: iti-sa-devops
5   namespace: iti-devops
6 automountServiceAccountToken: true
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  sudo      

```
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl apply -f serviceAccount.yml
serviceaccount/iti-sa-devops created
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl get ServiceAccount -n iti-devops
NAME          SECRETS  AGE
default       0        8m37s
iti-sa-devops  0        32s
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab#
```

```
2 kind: ClusterRole
3 metadata:
4   # "namespace" omitted since ClusterRoles are not namespaced
5   name: cluster-role-devops
6 rules:
7 - apiGroups: [""]
8   resources: ["configMaps", "secrets", "endpoints", "nodes", "pods", "services", "namespaces", "events", "serviceAccounts"]
9   verbs: ["get", "watch", "list", "create", "patch", "update"]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

sudo + - [] [] ^ x

```
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl apply -f clusterRole.yml
```

```
clusterrole.rbac.authorization.k8s.io/cluster-role-devops created
```

```
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl get ClusterRole
```

NAME	CREATED AT
admin	2022-12-31T11:20:20Z
cluster-admin	2022-12-31T11:20:20Z
cluster-role-devops	2023-01-29T15:00:20Z
edit	2022-12-31T11:20:20Z
kubeadm:get-nodes	2022-12-31T11:20:22Z
kubernetes-dashboard	2023-01-03T00:30:42Z
system:aggregate-to-admin	2022-12-31T11:20:20Z
system:aggregate-to-edit	2022-12-31T11:20:20Z
system:aggregate-to-view	2022-12-31T11:20:20Z
system:auth-delegator	2022-12-31T11:20:20Z
system:basic-user	2022-12-31T11:20:20Z
system:certificates.k8s.io:certificatesigningrequests:nodeclient	2022-12-31T11:20:20Z
system:certificates.k8s.io:certificatesigningrequests:selfnodeclient	2022-12-31T11:20:20Z
system:certificates.k8s.io:kube-apiserver-client-approver	2022-12-31T11:20:20Z
system:certificates.k8s.io:kube-apiserver-client-kubelet-approver	2022-12-31T11:20:20Z
system:certificates.k8s.io:kubelet-serving-approver	2022-12-31T11:20:20Z
system:certificates.k8s.io:legacy-unknown-approver	2022-12-31T11:20:20Z
system:controller:attachdetach-controller	2022-12-31T11:20:20Z
system:controller:certificate-controller	2022-12-31T11:20:20Z
system:controller:clusterrole-aggregation-controller	2022-12-31T11:20:20Z
system:controller:cronjob-controller	2022-12-31T11:20:20Z
system:controller:daemon-set-controller	2022-12-31T11:20:20Z
system:controller:deployment-controller	2022-12-31T11:20:20Z
system:controller:disruption-controller	2022-12-31T11:20:20Z
system:controller:endpoint-controller	2022-12-31T11:20:20Z
system:controller:endpointlice-controller	2022-12-31T11:20:20Z

```
2 # This cluster role binding allows anyone in the "manager" group to read secrets in any namespace.
3 kind: ClusterRoleBinding
4 metadata:
5   name: cluster-role-binding-devops
6   namespace: iti-devops
7 subjects:
8 - kind: ServiceAccount
9   name: iti-sa-devops
10  namespace: iti-devops
11 roleRef:
12   kind: ClusterRole
13   name: cluster-role-devops
14   apiGroup: rbac.authorization.k8s.io
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

sudo

```
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl apply -f binding.yml
clusterrolebinding.rbac.authorization.k8s.io/cluster-role-binding-devops created
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# kubectl get ClusterRoleBinding -n iti-devops
```

NAME	ROLE	AGE
cluster-admin	ClusterRole/cluster-admin	29d
cluster-role-binding-devops	ClusterRole/cluster-role-devops	28s
kubeadm:get-nodes	ClusterRole/kubeadm:get-nodes	29d
kubeadm:kubelet-bootstrap	ClusterRole/system:node-bootstrapper	29d
kubeadm:node-autoapprove-bootstrap	ClusterRole/system:certificates.k8s.io:certificatesigningrequests:nodeclient	29d
kubeadm:node-autoapprove-certificate-rotation	ClusterRole/system:certificates.k8s.io:certificatesigningrequests:selfnodeclient	29d
kubeadm:node-proxier	ClusterRole/system:node-proxier	29d
kubernetes-dashboard	ClusterRole/cluster-admin	26d
minikube-rbac	ClusterRole/cluster-admin	29d
storage-provisioner	ClusterRole/system:persistent-volume-provisioner	29d
system:basic-user	ClusterRole/system:basic-user	29d
system:controller:attachdetach-controller	ClusterRole/system:controller:attachdetach-controller	29d
system:controller:certificate-controller	ClusterRole/system:controller:certificate-controller	29d
system:controller:clusterrole-aggregation-controller	ClusterRole/system:controller:clusterrole-aggregation-controller	29d
system:controller:cronjob-controller	ClusterRole/system:controller:cronjob-controller	29d
system:controller:daemon-set-controller	ClusterRole/system:controller:daemon-set-controller	29d
system:controller:deployment-controller	ClusterRole/system:controller:deployment-controller	29d
system:controller:disruption-controller	ClusterRole/system:controller:disruption-controller	29d
system:controller:endpoint-controller	ClusterRole/system:controller:endpoint-controller	29d
system:controller:endpointslice-controller	ClusterRole/system:controller:endpointslice-controller	29d
system:controller:endpointslicemirroring-controller	ClusterRole/system:controller:endpointslicemirroring-controller	29d
system:controller:ephemeral-volume-controller	ClusterRole/system:controller:ephemeral-volume-controller	29d
system:controller:expand-controller	ClusterRole/system:controller:expand-controller	29d

```
1 What Is Kubernetes Deployment?
2
3 A Deployment is a Kubernetes resource object used for declarative application updates. Deployments allow you to define the lifecycle of applications, including the container images they use, the number
  of pods and the manner of updating them. Deployments are fully managed by the backend in Kubernetes, with the entire update process being server side, with no client involvement. They ensure that a
  specified number of pods are always running and available. The entire update process is recorded, with versioning to provide options for pausing, resuming or rolling back to previous version
4
5
6
7 What Is Kubernetes StatefulSet?
8
9 A StatefulSet is a workload API object for managing stateful applications. Usually, Kubernetes users are not concerned with how pods are scheduled, although they do require pods to be deployed in
  order, to be attached to persistent storage volumes, and to have unique, persistent network IDs that are retained through rescheduling. StatefulSets can help achieve these objectives. Like Deployments,
  StatefulSets manage the pods based on the same container specifications. However, they differ from deployments in that they maintain sticky identities for each pod. Pods may be created from an
  identical spec, but they are not interchangeable and are thus assigned unique identifiers that persist through rescheduling.
```

I can't set up ingress on minikube and I can't solve this problem

```
root@ahmed-Lenovo-Y520-15IKBN: /home/ahmed/Downloads/Sprints/k8s/day-5/lab
root@ahmed-Lenovo-Y520-15IKBN:/home/ahmed/Downloads/Sprints/k8s/day-5/lab# minikube addons enable ingress
💡 ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
■ Using image k8s.gcr.io/ingress-nginx/controller:v1.2.1
■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
🔍 Verifying ingress addon...

❌ Exiting due to MK_ADDON_ENABLE: run callbacks: running callbacks: [waiting for app.kubernetes.io/name=ingress-nginx pods: timed out waiting for the condition]
```

🐱 If the above advice does not help, please let us know:
👉 <https://github.com/kubernetes/minikube/issues/new/choose>

Please run `minikube logs --file=logs.txt` and attach logs.txt to the GitHub issue.
Please also attach the following file to the GitHub issue:
- /tmp/minikube_addons_5e63cb8ceb2e7e06e33b8dd7a9dcbebc0ad5ec5a_0.log