



Exercise 3.3: Finish Cluster Setup

1. View the available nodes of the cluster. It can take a minute or two for the status to change from NotReady to Ready. The NAME field can be used to look at the details. Your node name may be different.

```
student@cp:~$ kubectl get node
```

	NAME	STATUS	ROLES	AGE	VERSION
1	k8scp	Ready	control-plane,master	28m	v1.21.1
2	worker	Ready	<none>	50s	v1.21.1

2. Look at the details of the node. Work line by line to view the resources and their current status. Notice the status of Taints. The cp won't allow non-infrastructure pods by default for security and resource contention reasons. Take a moment to read each line of output, some appear to be an error until you notice the status shows False.

```
student@cp:~$ kubectl describe node k8scp
```

```

1 Name: k8scp
2 Roles: control-plane,master
3 Labels: beta.kubernetes.io/arch=amd64
4         beta.kubernetes.io/os=linux
5         kubernetes.io/arch=amd64
6         kubernetes.io/hostname=cp
7         kubernetes.io/os=linux
8         node-role.kubernetes.io/control-plane=
9         node-role.kubernetes.io/master=
10 Annotations: kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
11              node.alpha.kubernetes.io/ttl: 0
12              projectcalico.org/IPv4Address: 10.142.0.3/32
13              projectcalico.org/IPv4IPIPTunnelAddr: 192.168.242.64
14              volumes.kubernetes.io/controller-managed-attach-detach: true
15 CreationTimestamp: Wed, 26 May 2021 22:04:03 +0000
16 Taints: node-role.kubernetes.io/master:NoSchedule
17 <output_omitted>
```

3. Allow the cp server to run non-infrastructure pods. The cp node begins tainted for security and performance reasons. We will allow usage of the node in the training environment, but this step may be skipped in a production environment. Note the **minus sign (-)** at the end, which is the syntax to remove a taint. As the second node does not have the taint you will get a not found error.

```
student@cp:~$ kubectl describe node | grep -i taint
```

```

1 Taints: node-role.kubernetes.io/master:NoSchedule
2 Taints: <none>
```

```
student@cp:~$ kubectl taint nodes --all node-role.kubernetes.io/master-
```

```

1 node/k8scp untainted
2 error: taint "node-role.kubernetes.io/master:" not found
```

4. Determine if the DNS and Calico pods are ready for use. They should all show a status of Running. It may take a minute or two to transition from Pending.

```
student@cp:~$ kubectl get pods --all-namespaces
```

```

1 NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
2 kube-system    calico-node-jlgwr                         1/1     Running   0           6m
3 kube-system    calico-kube-controllers-74b888b647-wlqf5  1/1     Running   0           6m
4 kube-system    calico-node-tpvnr                         2/2     Running   0           6m
5 kube-system    coredns-78fcdf6894-nc5cn                 1/1     Running   0           17m
6 kube-system    coredns-78fcdf6894-xs96m                 1/1     Running   0           17m
7 <output_omitted>

```

5. **Only if** you notice the `coredns-` pods are stuck in `ContainerCreating` status you may have to delete them, causing new ones to be generated. Delete both pods and check to see they show a `Running` state. Your pod names will be different.

```
student@cp:~$ kubectl get pods --all-namespaces
```

```

1 NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AGE
2 kube-system    calico-node-qkvzh                       2/2     Running   0           59m
3 kube-system    calico-node-vndn7                       2/2     Running   0           12m
4 kube-system    coredns-576cbf47c7-rn6v4               0/1     ContainerCreating 0           3s
5 kube-system    coredns-576cbf47c7-vq5dz               0/1     ContainerCreating 0           94m
6 <output_omitted>

```

```
student@cp:~$ kubectl -n kube-system delete \
  pod coredns-576cbf47c7-vq5dz coredns-576cbf47c7-rn6v4
```

```

1 pod "coredns-576cbf47c7-vq5dz" deleted
2 pod "coredns-576cbf47c7-rn6v4" deleted

```

6. When it finished you should see a new tunnel, `tunl0`, interface. It may take up to a minute to be created. As you create objects more interfaces will be created, such as `cali` interfaces when you deploy pods, as shown in the output below.

```
student@cp:~$ ip a
```

```

1 <output_omitted>
2 4: tunl0@NONE: <NOARP,UP,LOWER_UP> mtu 1440 qdisc noqueue state
3 UNKNOWN group default qlen 1000
4   link/ipip 0.0.0.0 brd 0.0.0.0
5   inet 192.168.0.1/32 brd 192.168.0.1 scope global tunl0
6       valid_lft forever preferred_lft forever
7 6: calib0b93ed4661@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
8 1440 qdisc noqueue state UP group default
9   link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netnsid 1
10   inet6 fe80::ecee:eeff:ffff:eeee/64 scope link
11       valid_lft forever preferred_lft forever
12 <output_omitted>

```