

[Open in app](#)**Jonathan**[Follow](#)

29 Followers

[About](#)

# Certified Kubernetes Security Specialist (CKS) Preparation Part 2 — Create and Reset K8s Clusters



Jonathan Feb 10 · 2 min read

If you have not yet checked the first part of this series, please go ahead and check [Part1](#).

## Create K8s Clusters

Before we even start the learning journey of CKS, we would need to have an environment to test and break. For that, kubeadm is a good tool of rapidly bootstrapping everything together for new K8s clusters.

We would need one or more machines

- Running on Ubuntu or CentOS
- Each machine with the spec of at least 2 GiB of RAM and 2 CPUs
- Machines should be able to communicate with each other without issues through either private network or public network

Please check [this site](#) for the prerequisites information. [This Medium article](#) provides a pretty good explanation and demonstration on how you set up a K8s cluster with 1 master node and 2 worker nodes on Ubuntu machines.

[Open in app](#)

understand what CNI does is think of it as the service that allows Pods to communicate with each other regardless of what nodes they reside in and remove the Pod's network configuration when they are not in use. [This Medium article](#) provides a pretty thorough explanation.

## Reset K8s Clusters

Along the way of CKS learning, you would definitely require resetting K8s clusters for testing functions and features, here are some of the commands I use for that purpose.

#SSH into master nodes

- `sudo kubeadm reset -f`
- `sudo rm -rf /etc/kubernetes`
- `sudo iptables -F && iptables -t nat -F && iptables -t mangle -F && iptables -X`

Perform the same actions for worker nodes. After that, follow step 3 and further mentioned in [this Medium article](#) for re-initializing the K8s cluster.

If you would like to install a specific Kubernetes version for the cluster, you could use the following commands,

- `KUBE_VERSION= <desired K8s version>`
- `sudo apt-get update`

#I choose 0.8.7-00 for Kubernetes CNI, but any supported version should work.

- `sudo apt-get install -y docker.io kubelet=${KUBE_VERSION}-00  
kubeadm=${KUBE_VERSION}-00 kubectrl=${KUBE_VERSION}-00 kubernetes-  
cni=0.8.7-00`
- `sudo kubeadm init`

...

That is all for this article. Once following through the steps, we should be having a fully functional K8s cluster ready to perform anything!

Open in app



Ck

Kubernetes

Kubeadm

Create

Reset

About

Help

Legal

Get the Medium app

