

Installation Raspberry Pi Ubuntu Server 64-bit

1. install control plane/-master
controller1\$: curl -sL https://get.k3s.io | INSTALL_K3S_EXEC="C="--tls-san 192.168.1.10 --node-taint CriticalAddonsOnly=true:NoExecute" sh -
2. generate a k3s token
controller1\$: sudo cat /var/lib/rancher/k3s/server/node-token > node-token.txt
3. ssh into worker[x]
worker1\$: scp ubuntu@192.168.1.10:/home/ubuntu/node-token.txt /tmp/node-token.txt
4. install worker and join controller
worker1\$: curl -sL https://get.k3s.io | \ K3S_URL=https://192.168.1.7:6443 \ K3S_TOKEN=\$(cat /tmp/node-token.txt) \ sh -
5. check node installed and in 'Ready' state
controller\$: kubectl get nodes

Repeat Step 3 for every worker

Worker Nodes after install

- add a config.yaml file
/etc/rancher/k3s/config.yaml
ig.yaml on worker1 node-ip: 192.168.1.8 node-name: "worker1" node-label: ["kubernetes.io/role=worker", "project=homelab"]
- restart agent
sudo systemctl restart k3s-agent

Change IP Address of K3S nodes

- Remove any containers in the node
controller1\$: kubectl drain worker1 --ignore-daemonsets --delete-emptydir-data
- Remove the node from controller
controller1\$: kubectl delete node worker1
- Stop the worker k3s
worker1\$: sudo systemctl stop k3s-agent
- cleanup
worker\$: sudo rm -rf /etc/rancher /var/lib/rancher

Regular Install steps

Add on Additional Control Plane / Master

- Install
curl -sL https://get.k3s.io | \ INSTALL_K3S_EXEC="server \ --server https://192.168.1.10:6443 \ --token <token> \ --tls-san 192.168.1.11 \ --node-taint CriticalAddonsOnly=true:NoExecute \ --cluster-init=false" \ sh -

Controller After Install

- create config.yaml file
/etc/rancher/k3s/config.yaml
ig.yaml server: https://192.168.1.7:6443 name: default
/etc/rancher/k3s/config.yaml # Equivalent to advertiseAddress / LocalAPIEndpoint bind-address: 192.168.1.7 # Equivalent to nodeRegistration: name node-name: "controller" # Required so the SSL certificate matches the IP you advertised
tls-san: - 192.168.1.7 # Optional: ensures your ubuntu user can read the k3s.yaml without sudo
write-kubeconfig-mode: "-0644"

MacOS Administration Machine

1. MacOS Install Kubectl
brew install kubernetes-cli
2. MacOS verify
kubectl version --client
3. MacOS SSH into K3s control plane
\$> ssh ubuntu@192.x.xx.xxx
4. LINUX: Copy config to home drive
sudo cat /etc/rancher/k3s/k3s.yaml > k3s2.yaml
5. MacOS
scp ubuntu@192.168.1.7:/k3s2.yaml /Downloads/k3s.yaml

The kubeconfig file contains the credentials and endpoint information needed to connect securely to your K3s cluster. This file resides on your K3s server node

Development Workflow

- MACOS install Rancher Desktop
brew install --cask rancher
- MACOS Create project add docker file
FROM payara/micro:latest
COPY target/my-jsf-app.war /opt/payara/micro/deployments/
CMD ["--clustermode", "kubernetes"]

Troubleshooting

- view system error logs
sudo journalctl -u k3s-agent -n 50 --no-pager



By Slobodan Lohja (slohja)
cheatography.com/slohja/

www.linkedin.com/in/slobodanlohja/

Not published yet.

Last updated 27th January, 2026.

Page 1 of 1.

Sponsored by [ApolloPad.com](https://apollopod.com)

Everyone has a novel in them. Finish Yours!

<https://apollopod.com>