Demo:

This demo will connect a Raspberry Pi(as edge node) to the Virtual Machine (which works as the master node). You can make an Kubernetes cluster using more Raspberry Pis if available.

Steps Involved:
The major steps involved are:

1 – Getting the Raspberry Pi ready.
2 – Installing k3s-server on the master node.
3 – Installing k3s-agent on the edge(worker) node.

Let's take a look at each of them.

Step 1 – Getting the Raspberry Pi ready.

Note: You'll need a microSD card reader to install Raspbian.
There multiple ways to do this. I'm listing three which I am sure about. Two of these use a display screen and the last one is a headless install.

a. ***Using a pre-installed microSD card:***
   1. If you have a microSD card with pre-installed NOOBS, the easiest way to do this is to connect the Raspberry Pi board to a display monitor using HDMI, connect usb mouse and keyboard to the Pi, insert the card and power it on.
   2. Using a GUI you'll be able to install Raspbian by following directions on the screen. This is the simplest of ways.

b. ***Using an existing/new microSD card:***
   1. If you have an empty microSD card(or format your microSD card after backup), partition it as FAT32 to enable NOOBS to be installed on it.
   2. You can use a tool called SDCardFormatter which is available at: https://www.sdcard.org/downloads/formatter/ . Choose the appropriate OS; it works for both macOS and Windows.
   3. To install NOOBS, go to https://www.raspberrypi.org/downloads/ , select NOOBS and then copy the contents of the downloaded folder into your partitioned microSD card. Then repeat the steps in 'a'.

c. ***Headless install using Ether:***
   1. You'll need to download the Raspbian image. You can find it here - https://www.raspberrypi.org/downloads/raspbian/ .
   2. Download Etcher to flash the image onto the microSD card. You can find Etcher here - https://www.balena.io/etcher/ . It is available for macOS, Windows and Linux. Run the installer. The next steps are pretty straightforward.
   3. Put the microSD card into your machine. This doesn't need to be formatted, you can use a new SD card out of the box.

- Select image – this will be the Raspbian image you have downloaded.
- Select the drive.
- Click 'Flash'.

4. After flashing the image, if you can't see it on the system, safely eject the external volume and put back the card again into the reader. This should be seen as 'boot' on Mac and I'm pretty sure on Windows it is boot followed by a letter drive.
5. Next, we need to enable SSH. Since this is a headless installation, simply create an empty file at the root of the boot disk. Do not add an extension to the file.
6. Next, create another file in the root called wpa_supplicant.conf and place the following contents:

   country= ISO 3166 alpha-2 country code
   ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev update_config=1
   network={
   ssid="network_name"
   psk="network_password"
   }
7. Safely eject the card, put it into the Raspberry Pi and it should boot.

Step 2 – Installing k3s-server on the master node

K3s is not natively available for macOS. I tried to install a virtual machine and provisioned it using Vagrant. The Vagrant file can be found in the github folder.

1. Do 'vagrant up'. Alternatively, use 'vagrant up –provider=virtualbox'.

2. For a public network, you will be asked to choose which network you want to use for your external IP address.

3. Once the vm is up and running, type the following commands to get the k3s server running.

4. Since K3s server is already running in a vm, during my efforts to connect the Raspberry pi node to the Vm(master), I figured out that the agent uses the internal IP of the vm and runs into an error where it doesn't receive any tcp response. As a result, I had to figure out a workaround so that Raspberry Pi can connect to the vm using the node token. To do this: (You can run this as root)

   a. export K3S_NODE_NAME=${HOSTNAME_}
   b. export K3S_EXTERNAL_IP=xx.xx.xx.xx (Do an ifconfig and you would find the ip address)
   c. export INSTALL_K3S_EXEC="--docker --write-kubeconfig ~/.kube/config --write-kubeconfig-mode 666 --tls-san $K3S_EXTERNAL_IP --kube-apiserver-arg service-

node-port-range=1-65000 --kube-apiserver-arg advertise-address=$K3S_EXTERNAL_IP --kube-apiserver-arg external-hostname=$K3S_EXTERNAL_IP"

d. curl -sfL https://docs.rancher.com/k3s/k3s-install.sh | sh -
e. To check if installation was correct and the master is active, run – $(sudo) kubectl get nodes. This should display one running node as master.

5. Once we have ensured that the master is active, the next thing ot do is grab the token required for a worker node to connect to the cluster. To do this, execute the following command: $cat /var/lb/rancher/k3s/server/node-token

Step 3 – Installing k3s-agent on the worker node.

1. To install the k3s agent on the rpi – first ssh into the rpi

2. To ssh, you can grab the IP address from the Rasbian OS GUI using the dispay screen. Or you can try using nmap to find the address(I haven't used the nmap method). You'll be asked to enter your password. If you haven't set a password for your Raspberry Pi, the default username = pi and password = raspberry.

3. After successful ssh, we need to disable swap in the Pi. To do this, enter: dphys-swapfile swapoff && dphys-swapfile uninstall && update-rc.d dphys-swapfile remove

4. Then, append this to the /boot/cmdline.txt - cgroup_enable=cpuset cgroup_memory=1 cgroup_enable=memory. DO NOT put this before the text already in the file.

5. Reboot the Pi.

6. Again, we'll have to configure k3s using the external IP address of the vm. Do the following: (You can run this as root)
    a. export K3S_TOKEN="TOKEN_GRABBED_IN_STEP_2.5"
    b. export K3S_URL=https://xx.xx.xx.xx:6443 (This is the same IP used in Step2.4b)
    c. export INSTALL_K3S_EXEC="--docker --token $K3S_TOKEN --server $K3S_URL"
    d. export K3S_NODE_NAME=${HOSTNAME_}
    e. curl -sfL https://docs.rancher.cn/k3s/k3s-install.sh | sh -

7. On successful installation, check if the node is shown on the master. Do $kubectl get nodes. If it doesn't show immediately, give it a few more seconds and try again.

That is how we have deployed a 2-node cluster using Raspberry Pi 3 B+ as the edge node and Ubuntu VM as the master.

You can find the demo video and the Vagrantfle along with this document in the github folder.