

## OS setting

We are going to install some helpful software, set up hostnames, and do some other configuration that we need before we install Kubernetes. Of course, we are going to use Ansible. Don't get scared its simple 🐼. If you have followed my guides, you should not have any issues.

### Set hostname to every node

If you remember, [Here](#), I have set variable `var_hostname` in file `/etc/ansible/hosts` for each node, and this little nugget comes into play right now. On your main control node execute the following:

```
ansible cube -b -m shell -a "hostnamectl set-hostname {{ var_hostname }}"
```

Notable parameters are `"cube"`, as mentioned before, it means it will execute commands on every single node. Next, `-b` is short for `-become`. What this does is sudo by default, as we should have an ubuntu user with sudo already set up. This is all we need to execute our commands as root.

Next in line, `-m shell`. we are telling Ansible to use the shell module. This in essence mean we are going to execute commands via shell, and the commands are in `-a " <commands> "`.

Last in line is `{{ var_hostname }}`. This gets replaced for each node with a variable we set up in `/etc/ansible/hosts`. Setting variables ahead can be an easy way to customize some nodes, while still letting Ansible do all the heavy lifting.

You can check manually if everything worked by logging in to any of the nodes and running:

```
hostnamectl status
# or from main control node run Ansible command and get result from all nodes 🐼 Isn't Ansible useful ?
ansible cube -b -m shell -a "hostnamectl status | grep hostname"
```

### Remove "snap"

Don't get me wrong. Ubuntu's snap is not a bad idea, just useless for us. We are going to remove this feature and save some resources.

```
# All nodes should be the same so running this "list" should be enough on the main control node
ubuntu@control01:~$ snap list
Name      Version  Rev  Tracking      Publisher  Notes
core18    28280929 1936 latest/stable canonical/ base
lxd       4.7      18140 latest/stable/ canonical/ -
snapd     2.47.1   9730 latest/stable canonical/ snapd

# Remove the snaps from all nodes ( order matters )
ansible cube -b -m shell -a "snap remove lxd && snap remove core18 && snap remove snapd"
```

Remove snapd demon:

```
ansible cube -b -m shell -a "apt purge snapd -y"
ansible cube -b -m shell -a "apt autoremove -y"
```

### Update the OS

Simple as stated, we are going to update OS packages to the latest ones:

```
ansible cube -m apt -a "upgrade=yes update_cache=yes" --become
```

### Edit /boot/firmware/cmdline.txt

Required stuff for running containers.

```
ansible cube -b -m shell -a "sed -i 's/s/ cgroup_enable=cpuset cgroup_enable=memory cgroup_memory=1 swappiness=1/' /boot/firmware/cmdline.txt"
```

### Iptables to see bridged traffic

Create file `k3s.conf` in your home directory

```
# add following to ~/k3s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
# We are going to use Ansible again to copy it to place, in this case /etc/sysctl.d/k3s.conf so this will load to kernel on reboot
ansible cube -b -m copy -a "src=~/.k3s.conf dest=/etc/sysctl.d/k3s.conf owner=root group=root mode=0644"
```

### Disable green led

Since we do not have an SD card in our Raspberry Pi, the green LED is going nuts. We can disable it (turn it off) with a simple command. If anybody has a way to show disk activity with the green LED related to USB disk, please let me know!

```
# Disable the green led ( changing led0 to led1 you can disable power led 🐼 ) also enabling it back on, just changed echo 0 to echo 1
sudo sh -c 'echo 0 > /sys/class/leds/led0/brightness'

# Disable it for good after reboot adding paramters to /boot/firmware/config.txt
ansible cube -b -m shell -a "echo 'dtparam=act_led_trigger=none' >> /boot/firmware/config.txt && echo 'dtparam=act_led_activelow=off' >> /boot/firmware/config.txt"
```

My `/boot/firmware/config.txt` looks like this:

```
ubuntu@control01:~$ cat /boot/firmware/config.txt
[p14]
max_framebuffers=2

[all]
arm_64bit=1
kernel=vmlinux
cmdline=cmdline.txt
initramfs initrd.img followkernel

# Enable the audio output, I2C and SPI interfaces on the GPIO header
dtparam=audio=on
dtparam=i2c_arm=on
dtparam=spi=on

# Enable the serial pins
enable_uart=1

# Comment out the following line if the edges of the desktop appear outside
# the edges of your display
disable_overscan=1

# If you have issues with audio, you may try uncommenting the following line
# which forces the HDMI output into HDMI mode instead of DVI (which doesn't
# support audio output)
#hdmi_drive=2

# If you have a CM4, uncomment the following line to enable the USB2 outputs
# on the IO board (assuming your CM4 is plugged into such a board)
#dtoverlay=dwc2,dr_mode=host
dtparam=act_led_trigger=none
dtparam=act_led_activelow=off
```

There are other options that green light can do, just none that I found for USB disk :/

```
# Options for dtparam=act_led_trigger=
none          No trigger
kbd-scrolllock    Keyboard scroll lock
kbd-numlock       Keyboard num lock
kbd-capslock      Keyboard caps lock
kbd-kanaLock      Keyboard kana lock
kbd-shiftlock     Keyboard shift
kbd-ctrllock      Keyboard ctrl
kbd-altlock       Keyboard alt
kbd-shiftlock     Keyboard left shift
kbd-shiftlock     Keyboard right shift
kbd-ctrllock      Keyboard left ctrl
kbd-ctrllock      Keyboard right ctrl
timer           Flash at 1 second intervals
oneshot         Flash only once
heartbeat        Flash like a heartbeat (1-0-1-00000)
backlight        Always on
gpio            Flash when a certain GPIO is high???
cpu0            Flash on cpu0 usage
cpu1            Flash on cpu1 usage
cpu2            Flash on cpu2 usage
cpu3            Flash on cpu3 usage
default-on       Always on
[input]          Default state
panic           Flash on kernel panic
mmc0            Flash on mmc0 (primary SD Card interface) activity
mmc1            Flash on mmc1 (secondary SD Card interface) activity
rfkill0         Flash on wifi activity
rfkill1         Flash on bluetooth activity
```

### Remove IPv6

I don't like it, so it's gone.

```
# Edit /etc/sysctl.d/99-sysctl.conf and add to the bottom:
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
net.ipv6.conf.eth0.disable_ipv6 = 1

# If you do not have that file, create it and copy content of /etc/sysctl.conf into it first
```

Reboot

Do a reboot of all nodes. We can speed it up a little again with Ansible, but I will target just workers. If I did a reboot on all, I might kill my main node faster than it can be executed on others.. So, Ansible reboot for workers, and manually log in and reboot for control01-3.

```
ansible workers -b -m reboot
```

I hope you liked this part and got something useful. Take a break, grab some beverage and maybe for me too.

🍷 Liked it ? Buy me a drink :)

Last update: October 20, 2021

Comments

What do you think?  
3 Responses

👍

Upvote

😂

Funny

😍

Love

😲

Surprised

😡


Angry

😞

Sad

2 Comments   <https://rpi4cluster.com>   [Disqus' Privacy Policy](#)   [Login](#)

[Favorite](#)   [Tweet](#)   [Share](#)   [Sort by Best](#)

Join the discussion...

LOG IN WITH   OR SIGN UP WITH DISQUS ?

- DRUSE DRU • 10 months ago • edited

Shouldnt it be # add following to ~/k3s.conf instead of # add following to ~/k8s.conf ?

And also ansible workers -b -m reboot instead of "cube"?

Thank you for this!

⌵ | ⌵ • Reply • Share ⌵
- viadoportos Mod → DRUSE DRU • 10 months ago

1. ah yes the k3s and k8s it should be just one does not really matter which one. But since I'm using k3s I have made it consistent. ( might take minute to push