Nodes setting

Basic setting on each node

I’m going to log first to my primary control node (one of three dedicated as "control plane" for my Kubernetes cluster)

First, we should confirm that all our nodes are up:

```
#switch to root
sudo -s
#install nmap
apt install nmap
#scan local network range to see who is up
nmap -sP 192.168.0.1-254
```

This confirmed all my nodes are up and connected to the network. I’m going to use this primary node to connect to everything within this network. However, before we can leverage some automation with Ansible, there are a couple of manual tasks to do.

First, prepare /etc/hosts file on the one control node you are no right now:

```
# Edit /etc/hosts with your favorite editor, mine looks like:
127.0.0.1 localhost

192.168.0.101 control01 control01.local
192.168.0.109 control02 control02.local
192.168.0.108 control03 control03.local

192.168.0.102 cube01 cube01.local
192.168.0.103 cube02 cube02.local
192.168.0.104 cube03 cube03.local
192.168.0.105 cube04 cube04.local
192.168.0.106 cube05 cube05.local
192.168.0.107 cube06 cube06.local
```

Set hostname:

```
#hostnamectl set-hostname <hostname>
hostnamectl set-hostname control01

# check with status
hostnamectl status
```

Next, SSH log in manually to every node, and change the password when asked (keep it the same for simplicity, for now). So, you will have the username ubuntu and your password on each node.

Making life easier with Ansible

I’m going to use Ansible in a very simple manner so that everybody can understand what is going on. Of course, all of the following steps can be done via playbook, that you would run and leave for coffee…

Install Ansible

```
apt install ansible
```

Next, we need to create a file /etc/ansible/hosts (or edit it), and add our hosts. In essence, here we define hosts and groups of hosts that Ansible will try to manage.

```
# Edit file /etc/ansible/hosts
[control]
control01  ansible_connection=local var_hostname=control01
control02  ansible_connection=ssh   var_hostname=control02
control03  ansible_connection=ssh   var_hostname=control03

[workers]
cube01  ansible_connection=ssh  var_hostname=cube01
cube02  ansible_connection=ssh  var_hostname=cube02
cube03  ansible_connection=ssh  var_hostname=cube03
cube04  ansible_connection=ssh  var_hostname=cube04
cube05  ansible_connection=ssh  var_hostname=cube05
cube06  ansible_connection=ssh  var_hostname=cube06

[cube:children]
control
workers
```

Above, you can see I have added 3 groups: “control“, “workers” and “cube“. This was split so that if I want to execute some actions only on control servers, I use the “control” group. Group “cube” has children. This basically means that it’s a group of groups, and when I’m using “cube” I’m targeting every single node from the listed groups.

Another thing is “var_hostname”. This is a variable set individually per host, so that I can reference it later and change each nodes “hostname“ to its correct one with one command.

Next, “ansible_connection”: we are telling Ansible how to connect to that host. The primary method is ssh, but I specified “local” for `control01`, because this is the node that we are running Ansible from. This way, it won’t try to ssh to itself.

Lastly, we are going to make it so that user `ubuntu` will be able to log in to other nodes from `contro01` without the password using an ssh key. This step is optional, but after this you won’t need to type the password every time you run Ansible.

```
# Make sure you are user ubuntu
cd
mkdir -p ~/.ssh
chmod 700 ~/.ssh
ssh-keygen -t rsa
# Copy keys to each node, for example:
ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@control02
```

After this, we are ready for some mass settings with Ansible, but that will be in the next article. I’m trying to keep them shorter so that you don’t get overwhelmed by a walls of text.

Your very first Ansible command

This is the last thing before we head on to the next article; we are going to check if Ansible is working fine and can connect to all nodes:

```
# Run following as ubuntu user
# We are going to execute ping via ansible, the "cube" is group we specified in /etc/ansible/hosts
# And if you rememeber this will execute the command on all nodes.
# -m mean we are going to use module, in our case module: ping
ansible cube -m ping

#Result should be:
ubuntu@ubuntu:~$ ansible cube -m ping
control01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
control02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
control03 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
cube02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
cube01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
cube03 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
cube04 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
cube05 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
cube06 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Liked it ? Buy me a drink :)

Last update: October 20, 2021

Comments

What do you think?

1 Response

👍 Upvote 😝 Funny 😍 Love 😯 Surprised 😤 Angry 😥 Sad

2 Comments https://rpi4cluster.com Disqus' Privacy Policy 1 Login

♡ Favorite Tweet Share Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

DfUSE DfU • 10 months ago

hosts file seems to have a typo as all cube nodes are cube01.local

```
192.168.0.102 cube01 cube01.local
192.168.0.103 cube02 cube01.local
192.168.0.104 cube03 cube01.local
192.168.0.105 cube04 cube01.local
192.168.0.106 cube05 cube01.local
192.168.0.107 cube06 cube01.local
```

Thanks!