Ansible

### *Build 3 instances t2.micro on amazon linux, controller, host-manager-1 and host-manager-2*

- hostnamectl set-hostname controller
- bash
- ssh-keygen
- ip a s
- cd .ssh
- vim authorized_keys
- cat id_rsa.pub

#### *Here incase of controller copy paste the public key of other two instances and paste the keys in the vim file and likewise do the same for other 2 instances to establish connection*

### *Controller*

- systemctl start sshd
- systemctl enable sshd

#### *Use ping command to check if the connection is established between the instances and allow ICMP port in network security*

### *Controller - Install ansible*

- yum install ansible* -y
- ansible --version
- cd /etc/ansible/
- vim ansible.cfg

### *Paste the below content inside the vim file*

```cfg

# config file for ansible -- http://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

hostfile       = /etc/ansible/hosts
library        = /usr/share/ansible
remote_tmp     = $HOME/.ansible/tmp
pattern        = *
forks          = 5
poll_interval  = 15
sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
transport      = smart
remote_port    = 22
```

```
# additional paths to search for roles in, colon seperated
#roles_path    = /etc/ansible/roles

# uncomment this to disable SSH key host checking
#host_key_checking = False

# change this for alternative sudo implementations
sudo_exe = sudo

# what flags to pass to sudo
#sudo_flags = -H

# SSH timeout
timeout = 10

# default user to use for playbooks if user is not specified
# (/usr/bin/ansible will use current user as default)
#remote_user = root

# logging is off by default unless this path is defined
# if so defined, consider logrotate
#log_path = /var/log/ansible.log

# default module name for /usr/bin/ansible
#module_name = command

# use this shell for commands executed under sudo
# you may need to change this to bin/bash in rare instances
# if sudo is constrained
#executable = /bin/sh

# if inventory variables overlap, does the higher precedence one win
# or are hash values merged together?  The default is 'replace' but
# this can also be set to 'merge'.
#hash_behaviour = replace

# How to handle variable replacement - as of 1.2, Jinja2 variable syntax is
# preferred, but we still support the old $variable replacement too.
# Turn off ${old_style} variables here if you like.
#legacy_playbook_variables = yes

# list any Jinja2 extensions to enable here:
#jinja2_extensions = jinja2.ext.do,jinja2.ext.i18n

# if set, always use this private key file for authentication, same as
# if passing --private-key to ansible or ansible-playbook
#private_key_file = /path/to/file

# format of string {{ ansible_managed }} available within Jinja2
# templates indicates to users editing templates files will be replaced.
# replacing {file}, {host} and {uid} and strftime codes with proper values.
ansible_managed = Ansible managed: {file} modified on %Y-%m-%d %H:%M:%S by {uid} on {host}

# by default, ansible-playbook will display "Skipping [host]" if it determines a task
# should not be run on a host.  Set this to "False" if you don't want to see these "Skipping"
# messages. NOTE: the task header will still be shown regardless of whether or not the
# task is skipped.
#display_skipped_hosts = True

# by default (as of 1.3), Ansible will raise errors when attempting to dereference
# Jinja2 variables that are not set in templates or action lines. Uncomment this line
# to revert the behavior to pre-1.3.
#error_on_undefined_vars = False

# set plugin path directories here, seperate with colons
action_plugins     = /usr/share/ansible_plugins/action_plugins
callback_plugins   = /usr/share/ansible_plugins/callback_plugins
connection_plugins = /usr/share/ansible_plugins/connection_plugins
lookup_plugins     = /usr/share/ansible_plugins/lookup_plugins
vars_plugins       = /usr/share/ansible_plugins/vars_plugins
filter_plugins     = /usr/share/ansible_plugins/filter_plugins

# don't like cows?  that's unfortunate.
# set to 1 if you don't want cowsay support or export ANSIBLE_NOCOWS=1
#nocows = 1

# don't like colors either?
# set to 1 if you don't want colors, or export ANSIBLE_NOCOLOR=1
#nocolor = 1

# the CA certificate path used for validating SSL certs. This path
# should exist on the controlling node, not the target nodes
# common locations:
# RHEL/CentOS: /etc/pki/tls/certs/ca-bundle.crt
# Fedora     : /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
# Ubuntu     : /usr/share/ca-certificates/cacert.org/cacert.org.crt
#ca_file_path =

# the http user-agent string to use when fetching urls. Some web server
# operators block the default urllib user agent as it is frequently used
# by malicious attacks/scripts, so we set it to something unique to
# avoid issues.
#http_user_agent = ansible-agent

[paramiko_connection]

# uncomment this line to cause the paramiko connection plugin to not record new host
# keys encountered.  Increases performance on new host additions.  Setting works independently of the
# host key checking setting above.
#record_host_keys=False

# by default, Ansible requests a pseudo-terminal for commands executed under sudo. Uncomment this
# line to disable this behaviour.
#pty=False

[ssh_connection]

# ssh arguments to use
# Leaving off ControlPersist will result in poor performance, so use
# paramiko on older platforms rather than removing it
#ssh_args = -o ControlMaster=auto -o ControlPersist=60s

# The path to use for the ControlPath sockets. This defaults to
# "%(directory)s/ansible-ssh-%%h-%%p-%%r", however on some systems with
# very long hostnames or very long path names (caused by long user names or
# deeply nested home directories) this can exceed the character limit on
# file socket names (108 characters for most platforms). In that case, you
# may wish to shorten the string below.
#
# Example:
# control_path = %(directory)s/%%h-%%r
#control_path = %(directory)s/ansible-ssh-%%h-%%p-%%r

# Enabling pipelining reduces the number of SSH operations required to
# execute a module on the remote server. This can result in a significant
# performance improvement when enabled, however when using "sudo:" you must
# first disable 'requiretty' in /etc/sudoers
#
# By default, this option is disabled to preserve compatibility with
# sudoers configurations that have requiretty (the default on many distros).
#
#pipelining = False

# if True, make ansible use scp if the connection type is ssh
# (default is sftp)
#scp_if_ssh = True
```

```
[accelerate]

accelerate_port = 5099

accelerate_timeout = 30

accelerate_connect_timeout = 5.0

```

- ansible --version

#### *Create vim host under etc/ansible*

- vim hosts

paste ip of the two hosts in format

[us-server]

172.31.1.23

- ansible all --list-hosts
- ansible all -m ping

type yes 2 times

#### *When the output is green command is executed, when gold that means ansible did changes in the remote machine and when in red it indicates error*

### *Configure web server*

- cd /etc/ansible
- vim configure.yml

#### *Paste the below content inside vim file*

```yaml

  ---

- name: configure apache server
 hosts: all

 tasks:

    - name: installed httpd pkg
     dnf:
         name: httpd
         state: latest
    - name: copy index.html file
     copy:
         src: index.html
         dest: /var/www/html/index.html
    - name: started apache
     systemd:
          name: httpd
          state: started
          enabled: true

```

- ansible-playbook configure.yml --syntax-check

- cat >index.html

- ansible-playbook configure.yml

#### *paste the public ip of the host*

### *To make a group and a user using script*

vim playbook.yml

#### *Paste the below content inside the vim file*

```yaml
---

- name: creating some user & group
  hosts: all

  tasks:

    - name: create a group
      group:
          name: devops
          state: present
    - name: create an user thor
      user:
          name: thor
      uid: 1200
      shell: /bin/bash
      home: /home/india
      groups: devops
      state: present
  - name: install smb pkg
    yum:
      name: cifs-utils
      state: present
  - name: install ftp
    yum:
      name: nfs-utils
      state: present
```

- ansible-playbook playbook.yml

#### *Go to host and run the below command to check the user created inside group*

- cat /etc/group

### *Handler*

- vim configure-appache.yml   (*Inside ansible directory*)

(*Paste the below content*)

```yaml
 ---
- name: configure apache with handler
```

```
hosts: all
tasks:
 - name: installed httpd
   dnf:
     name: httpd
     state: latest

 - name: copied httpd.conf file on target machine
   copy:
     src: httpd.conf
     dest: /etc/httpd/conf/httpd.conf

 - name: copied index.html
   copy:
     src: index.html
     dest: /var/www/html/index.html

 - name: restart the httpd service
   systemd:
     name: httpd
     state: restarted
     enabled: true
   notify: restart_httpd

handlers:
 - name: restart_httpd
   service:
     name: httpd
```

```
      state: restarted
      enabled: true

  - name: restart_firewalld
    service:
      name: firewalld
      state: restarted
      enabled: true
```

(*Now make an index file*)

- cat > index.html
- yum install httpd -y
- cd /etc/ansible
- vim /etc/httpd/conf/httpd.conf   (#*Add a comment at start #Mahek Shetty and then save it*)
- ll   (#*Check for index.html file*)
- ansible-playbook configure-appache.yml --syntax-check   (#*Handler and task should have same indentation*)
- ansible-playbook configure-appache.yml

#### (*Now go to host and check if the files are reflecting*)

(*To check if httpd is installed*)

- rpmquery httpd

(*To check if the index html file is available*)

- cd /var/www/html
- ll

(*To check conf file*)

- cd /etc/httpd/conf
- ll

(*Check if the changes made in the /etc/httpd/conf/httpd.conf is reflecting*)

- cat httpd.conf
- ip a s
- curl http://172.31.34.36

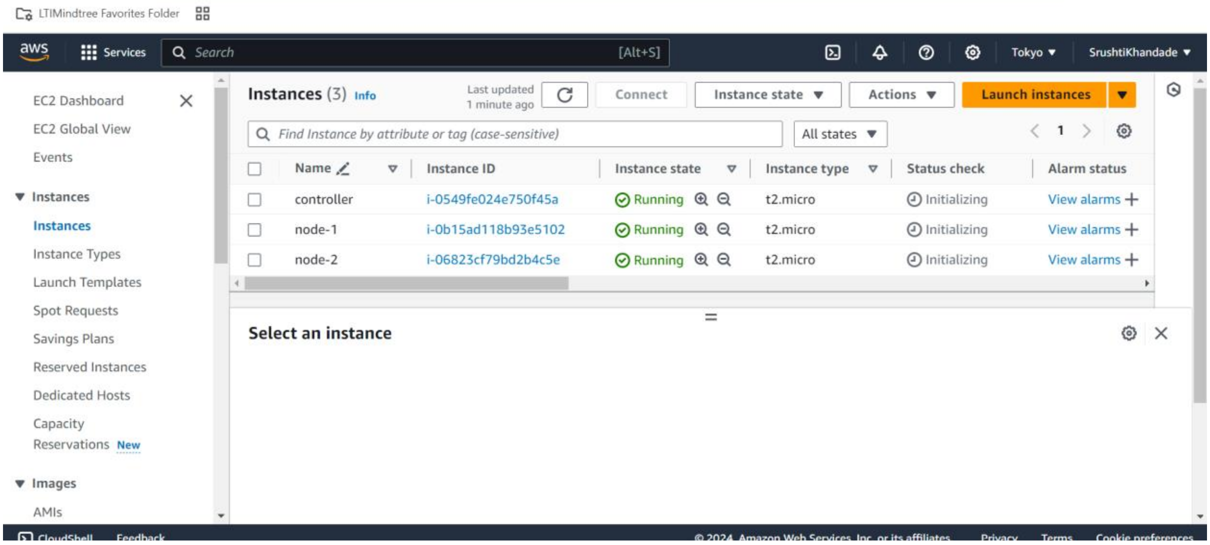
### *Configure AWS with ansible*

- cd /etc/ansible
- vim creds.yml (#*Paste the aws_access_key: and aws_secret_key: *)
- vim ec2.yml

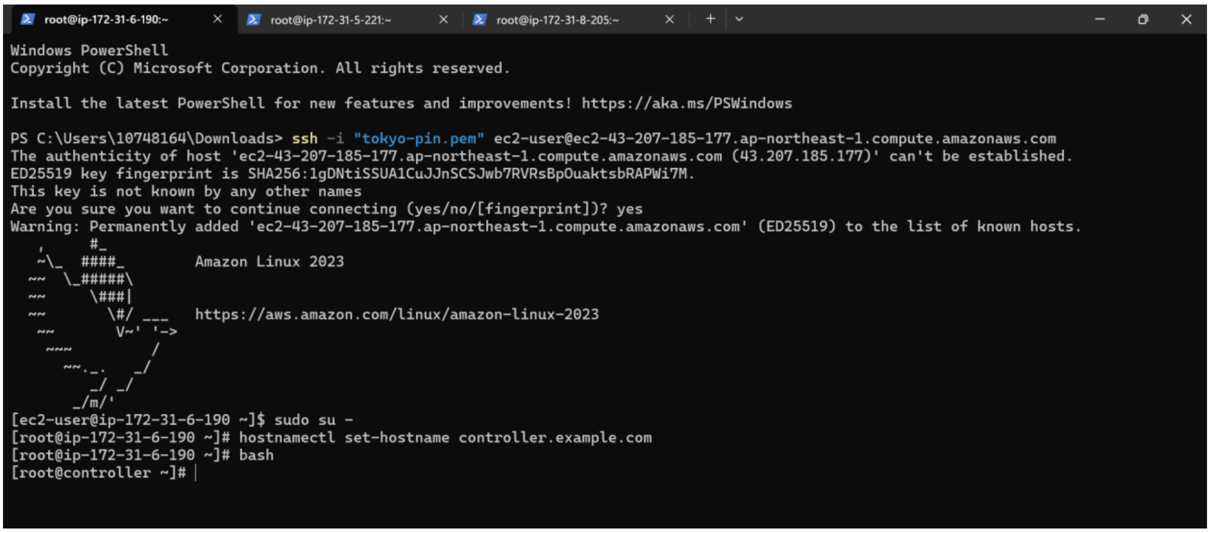
#### *Here change the ami, region, security group, key name*

```yaml
 ---
- name: create an ec2 instance
  hosts: all
  vars_files:
      - creds.yml
  tasks:
    - name: install pip
      yum:
        name: pip
      state: present
  - name: install boto3
    pip:
      name: boto3
      state: present

  - name: create an ec2 instance using ansible
    amazon.aws.ec2_instance:
      aws_access_key: "{{ aws_access_key }}"
      aws_secret_key: "{{ aws_secret_key }}"
      key_name: "ans-key"
      instance_type: t2.micro
      security_group: sg-0891eca823714b7e5
      region: ap-southeast-2
      count: 1
      image_id: ami-0e8fd5cc56e4d158c
      tags:
```

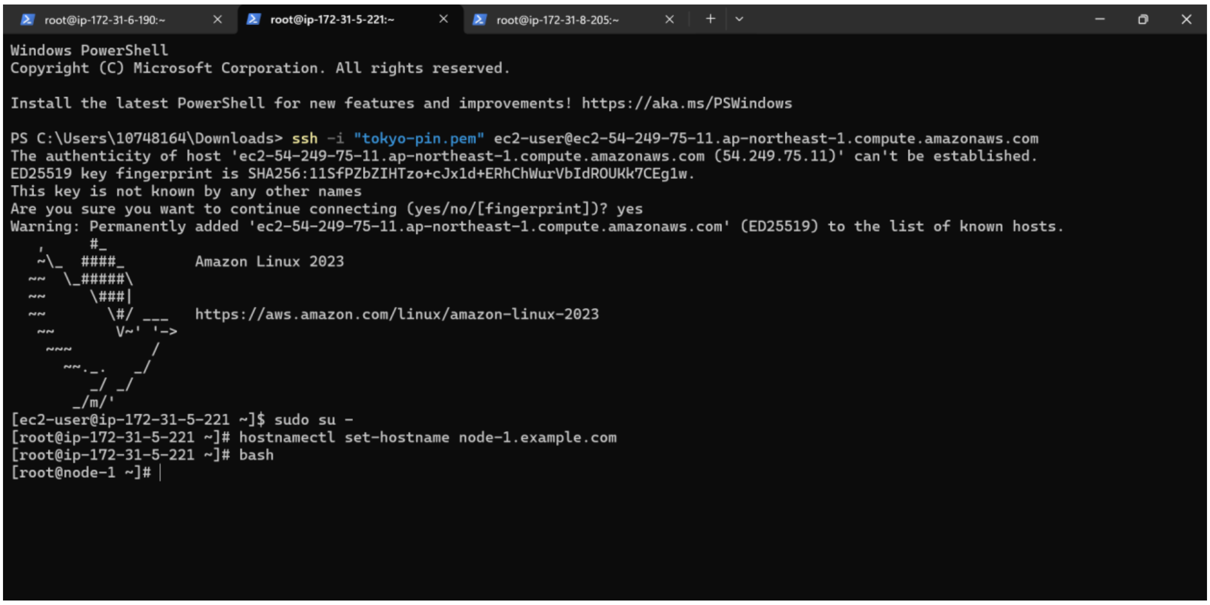
```
ansible-playbook ec2.yml
```



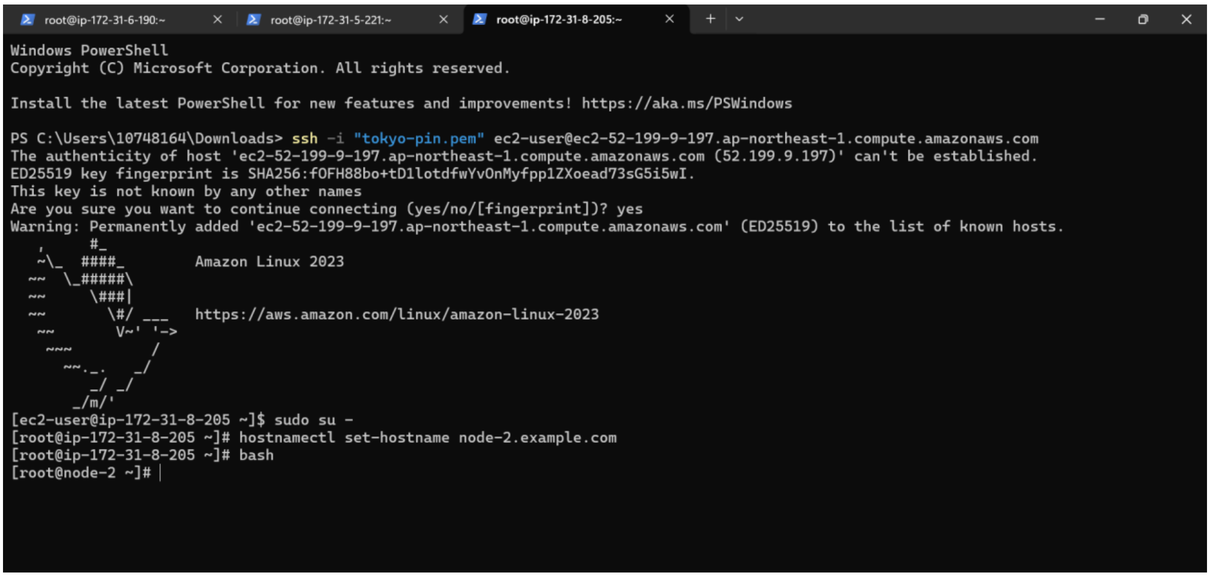
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	controller	i-0549fe024e750f45a	✓ Running	t2.micro	Initializing	View alarms +
□	node-1	i-0b15ad118b93e5102	✓ Running	t2.micro	Initializing	View alarms +
□	node-2	i-06823cf79bd2b4c5e	✓ Running	t2.micro	Initializing	View alarms +



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10748164\Downloads> ssh -i "tokyo-pin.pem" ec2-user@ec2-43-207-185-177.ap-northeast-1.compute.amazonaws.com
The authenticity of host 'ec2-43-207-185-177.ap-northeast-1.compute.amazonaws.com (43.207.185.177)' can't be established.
ED25519 key fingerprint is SHA256:1gDNtiSSUA1CuJJnSCSJwb7RVRsBpOuaktsbRAPWi7M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-43-207-185-177.ap-northeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-6-190 ~]$ sudo su -
[root@ip-172-31-6-190 ~]# hostnamectl set-hostname controller.example.com
[root@ip-172-31-6-190 ~]# bash
[root@controller ~]#
```

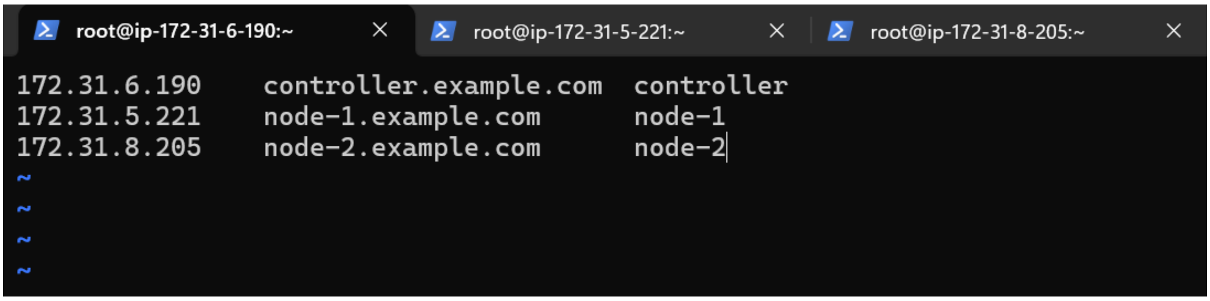


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10748164\Downloads> ssh -i "tokyo-pin.pem" ec2-user@ec2-54-249-75-11.ap-northeast-1.compute.amazonaws.com
The authenticity of host 'ec2-54-249-75-11.ap-northeast-1.compute.amazonaws.com (54.249.75.11)' can't be established.
ED25519 key fingerprint is SHA256:11SfPZbZIHTzo+cJx1d+ERhChWurVbIdROUKk7CEg1w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-249-75-11.ap-northeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-5-221 ~]$ sudo su -
[root@ip-172-31-5-221 ~]# hostnamectl set-hostname node-1.example.com
[root@ip-172-31-5-221 ~]# bash
[root@node-1 ~]#
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10748164\Downloads> ssh -i "tokyo-pin.pem" ec2-user@ec2-52-199-9-197.ap-northeast-1.compute.amazonaws.com
The authenticity of host 'ec2-52-199-9-197.ap-northeast-1.compute.amazonaws.com (52.199.9.197)' can't be established.
ED25519 key fingerprint is SHA256:fOFH88bo+tD1lotdfwYvOnMyfpp1ZXoead73sG5i5wI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-199-9-197.ap-northeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-8-205 ~]$ sudo su -
[root@ip-172-31-8-205 ~]# hostnamectl set-hostname node-2.example.com
[root@ip-172-31-8-205 ~]# bash
[root@node-2 ~]#
```

In controller



```
172.31.6.190    controller.example.com  controller
172.31.5.221    node-1.example.com      node-1
172.31.8.205    node-2.example.com      node-2
~
~
~
~
```