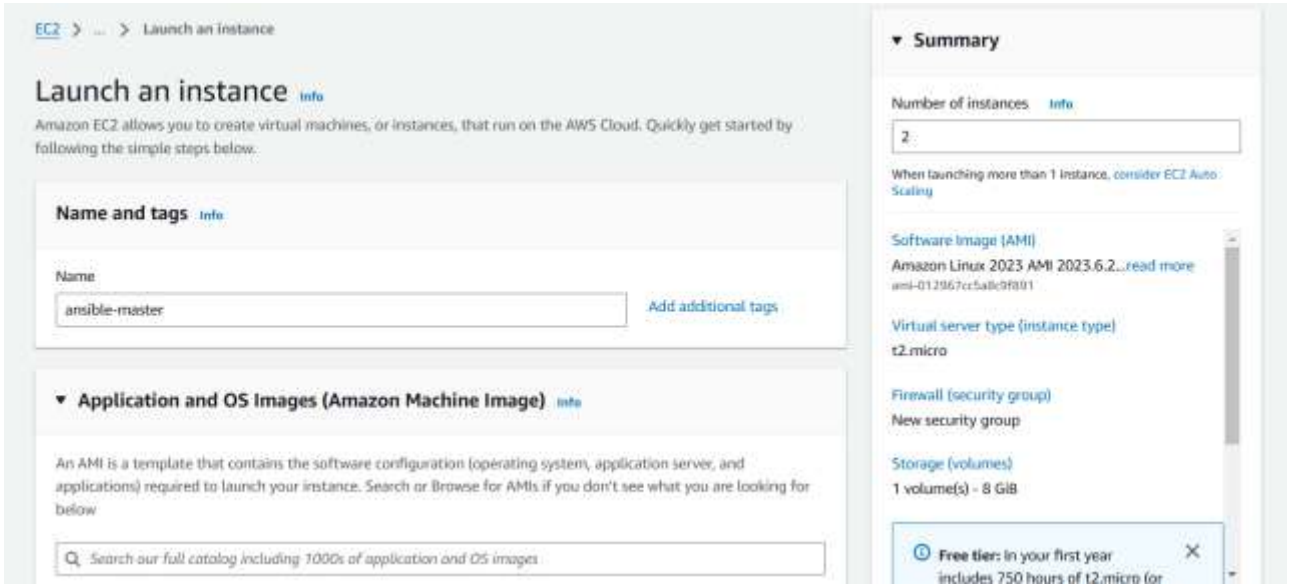
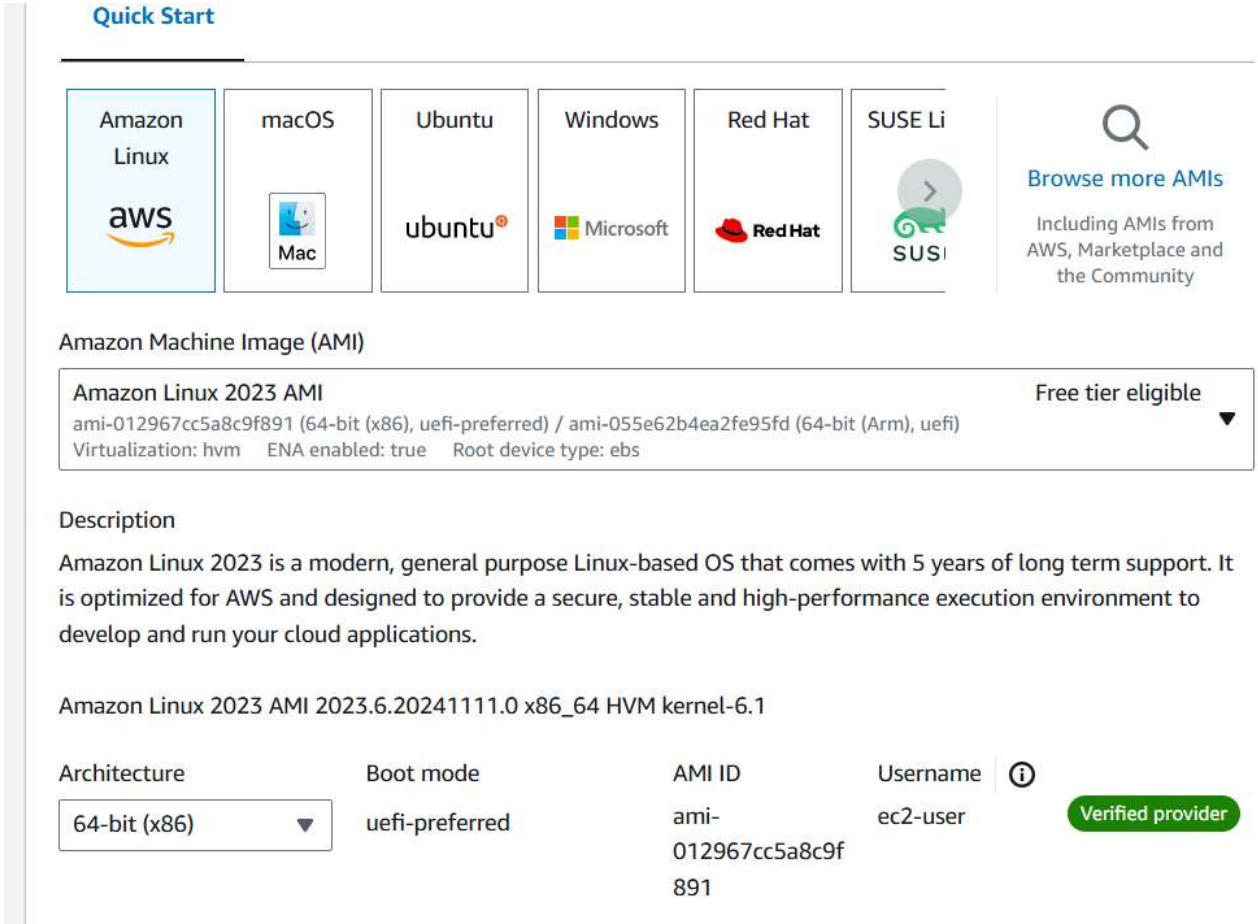
INSTALLING CLOUDWATCH AGENT THROUGH ANSIBLE

- Launch the ec2s





Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

vsv

Create new key pair

Network settings Info

Edit

Network Info

vpc-0b02e253d16efa60a

Network Info

vpc-0b02e253d16efa60a

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group | Select existing security group

Common security groups Info

Select security groups

default sg-02c11ee51ac59d0ea
VPC: vpc-0b02e253d16efa60a

Compare security group rules

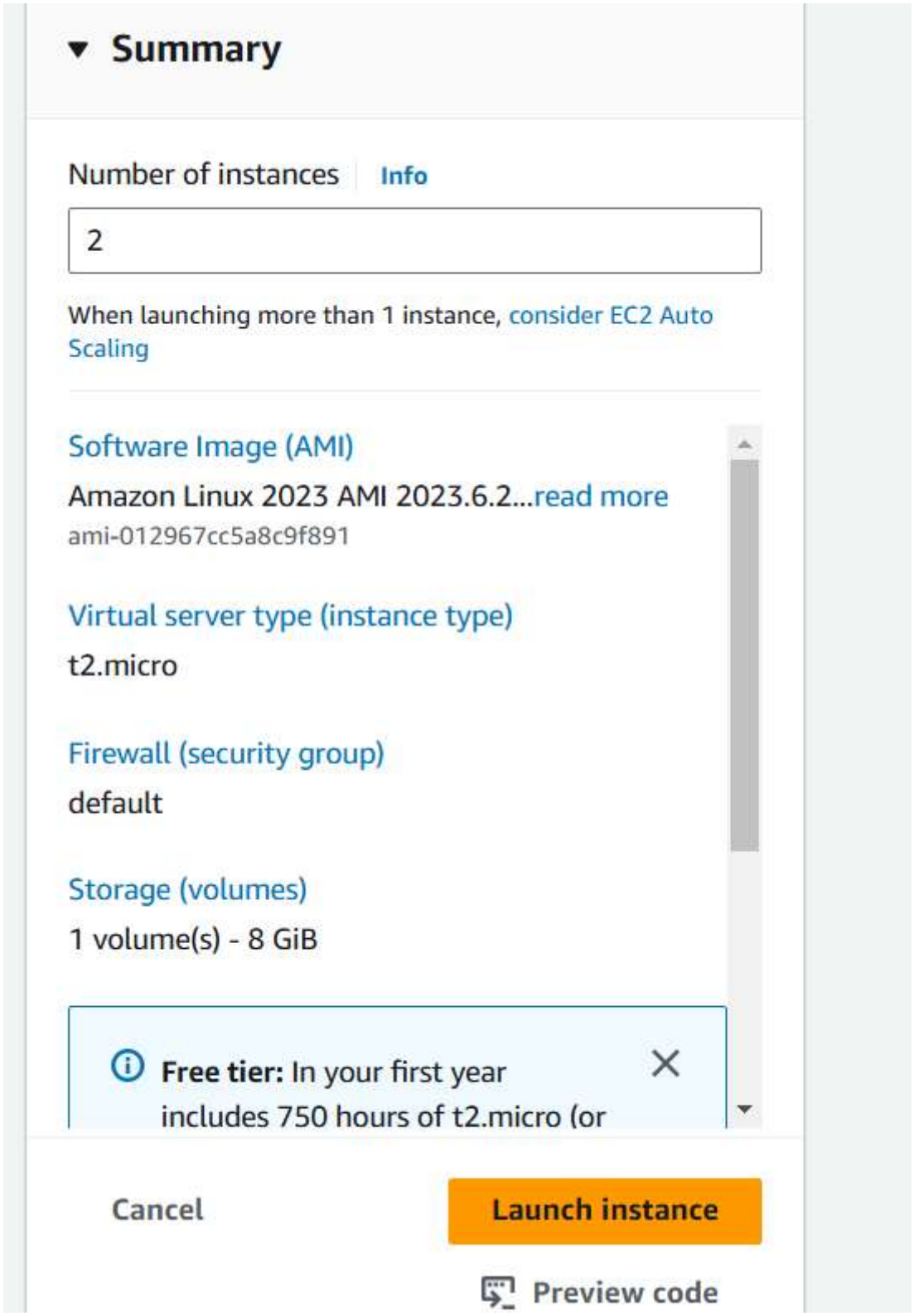
Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage Info

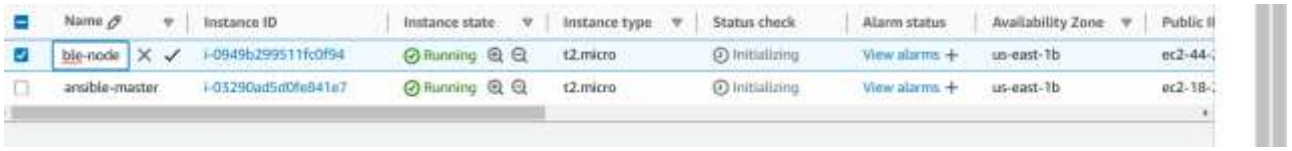
Advanced

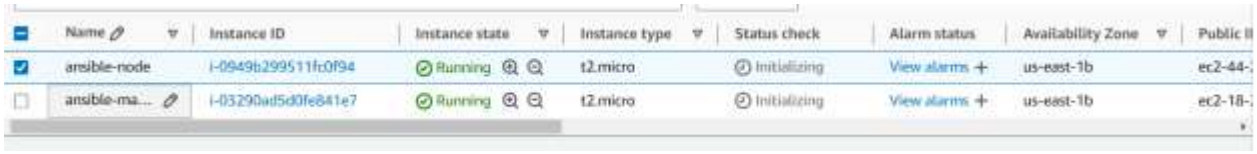
1x 8 GiB gp3 Root volume (Not encrypted)

- In number of instances select 2 one for node another is master

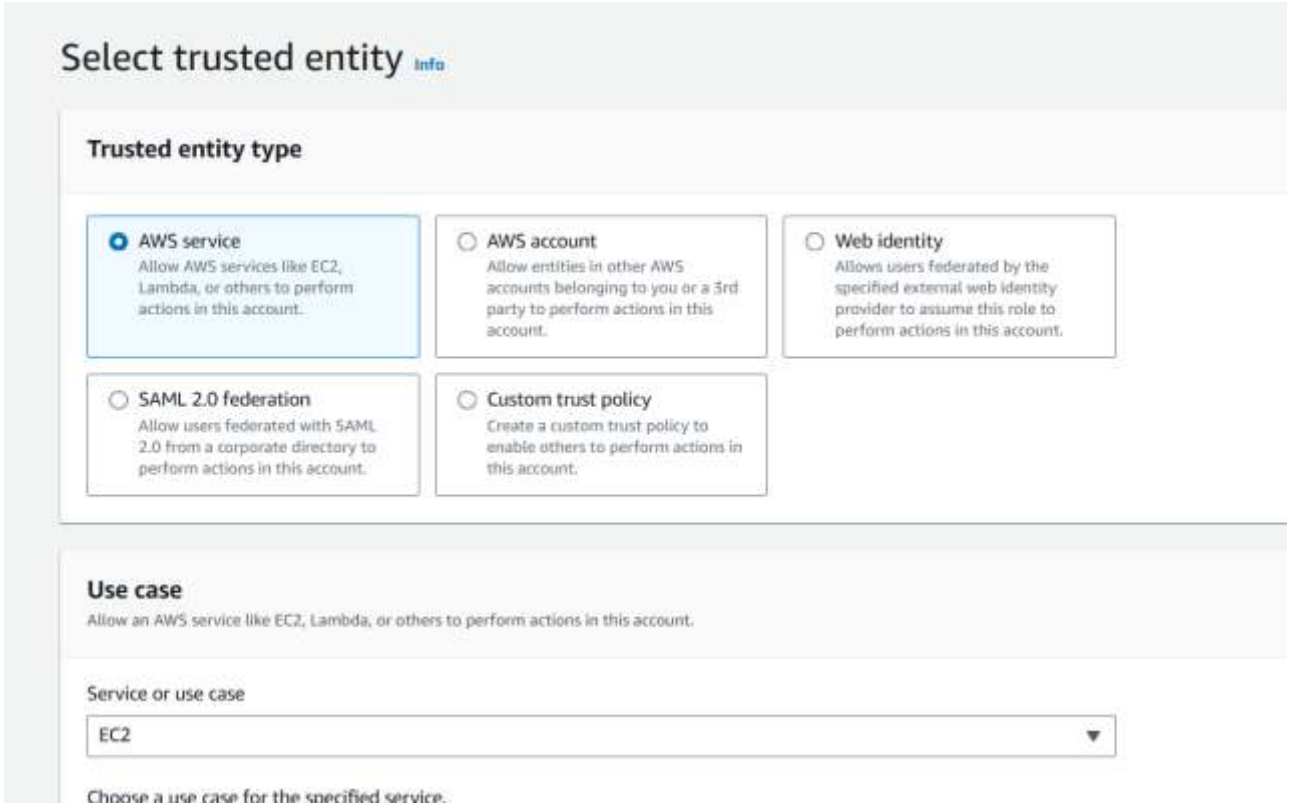


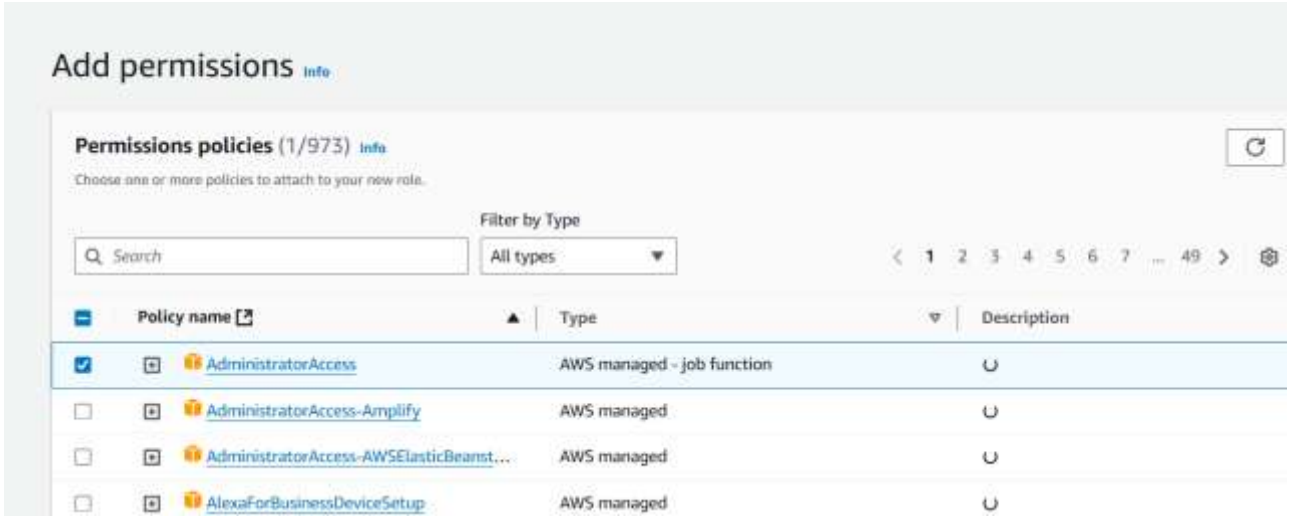
- Change the one instance name to ansible node

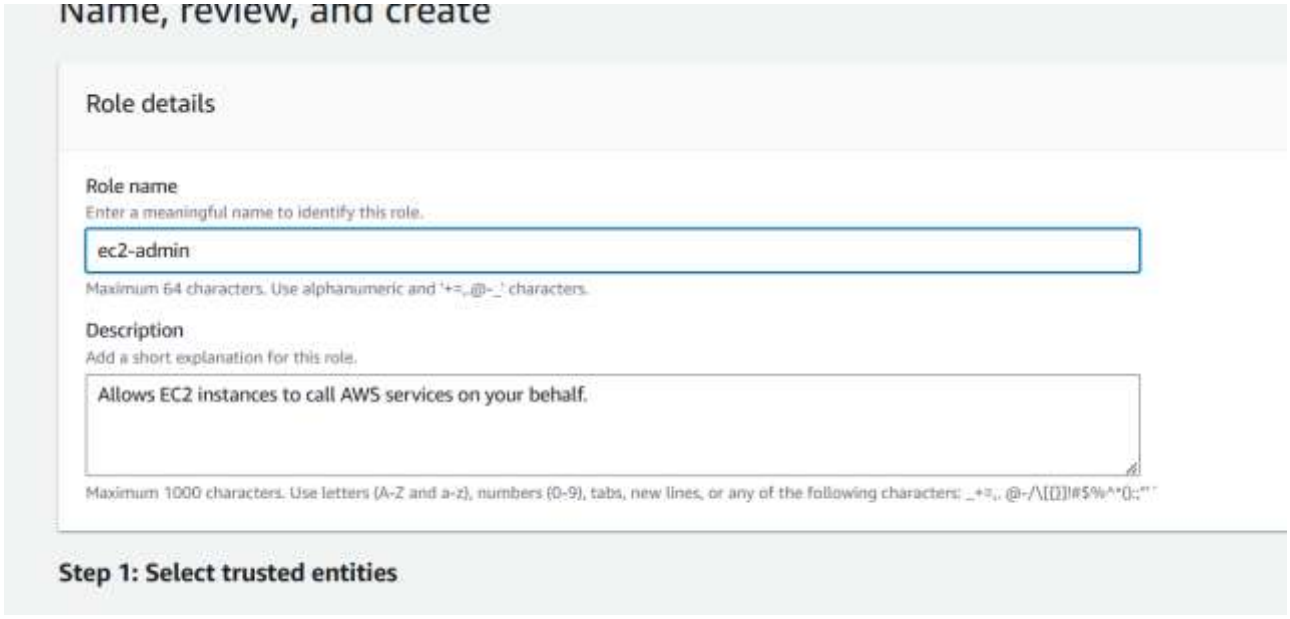




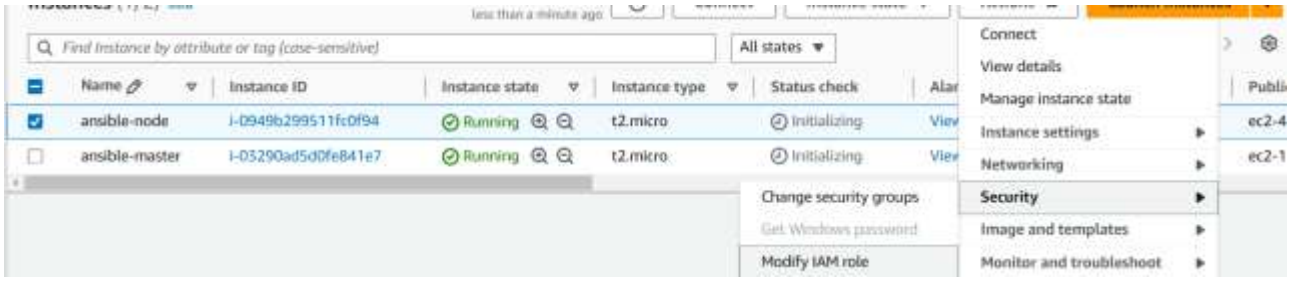
- Crete iam role for ansible node



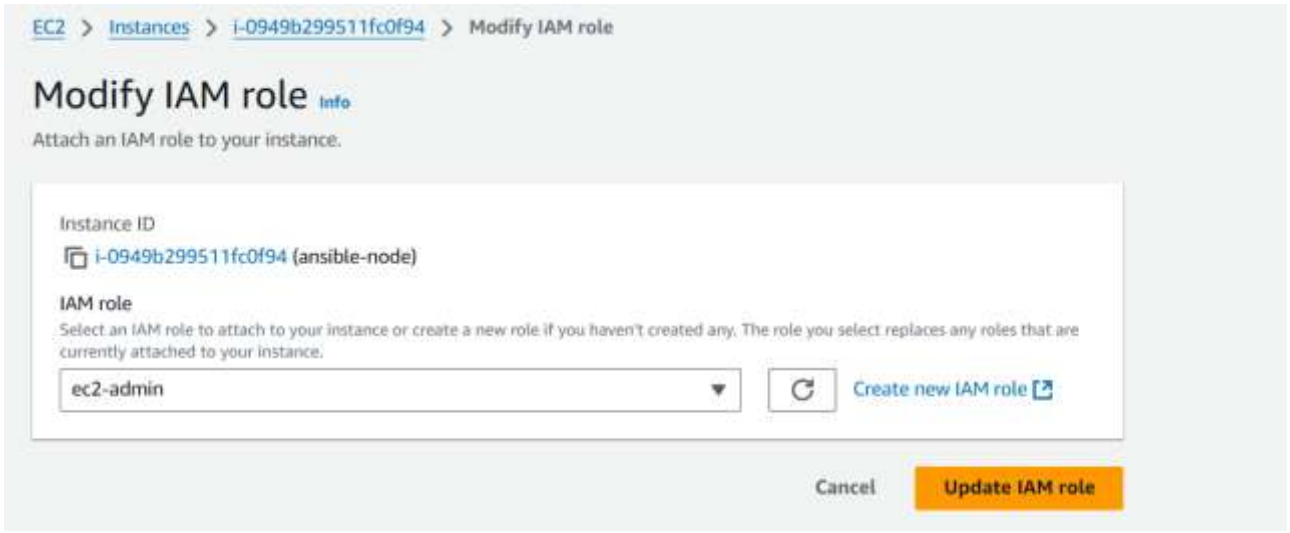




- Attach role to ansible node

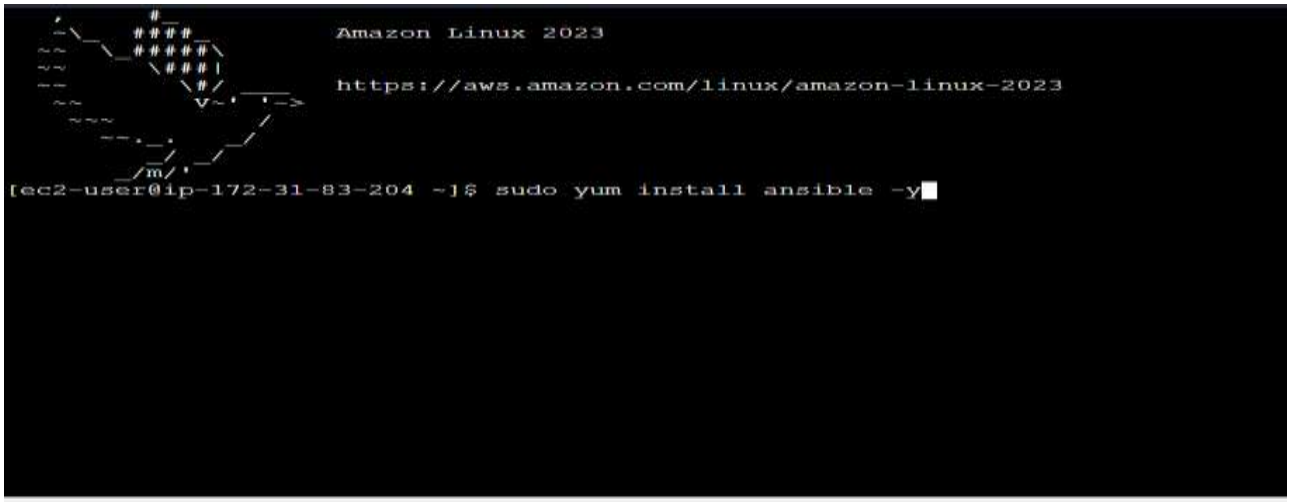


-- ansible ec2 want to communicate with to cloudwatch



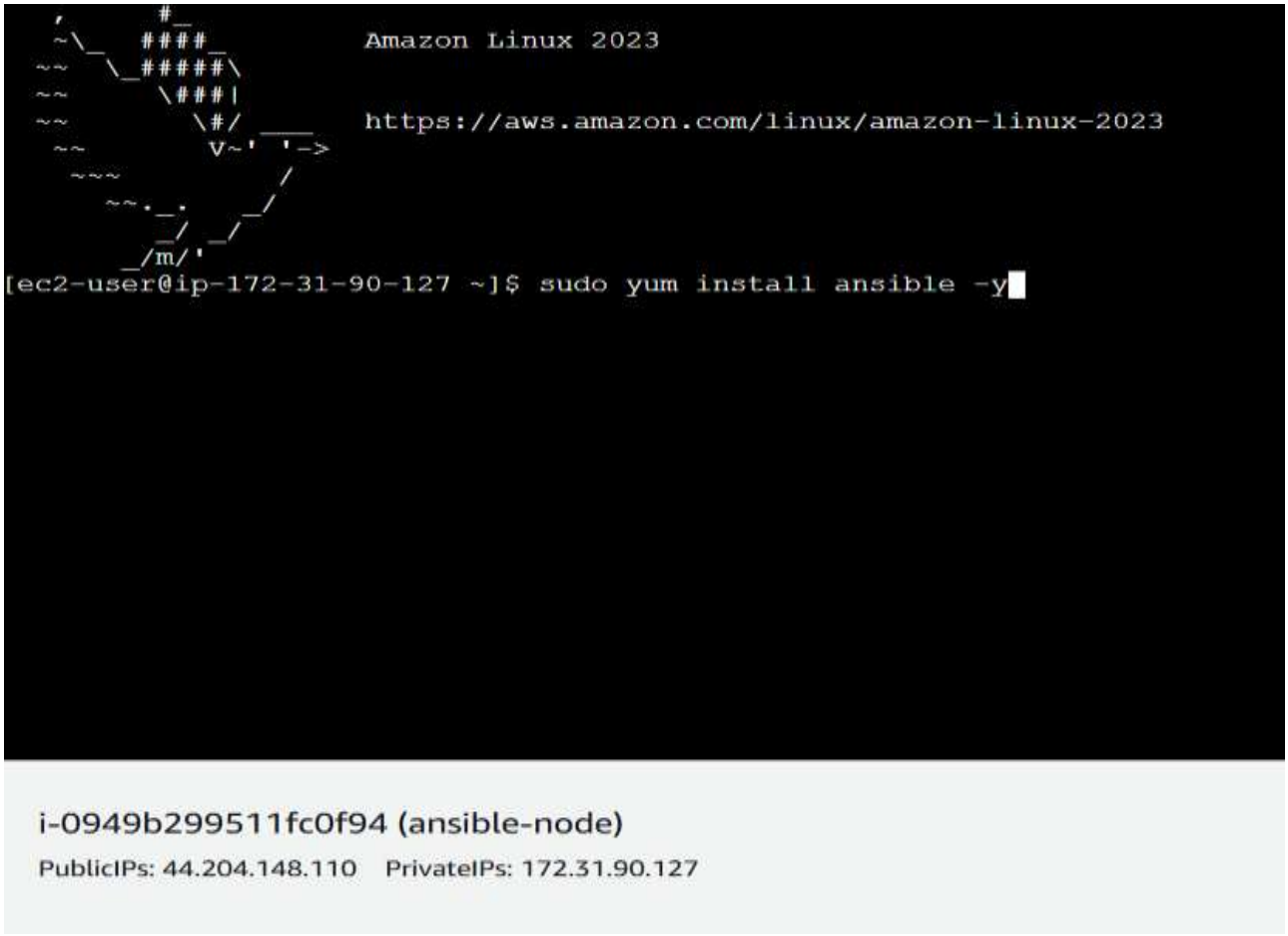
- Connect to ansible master install ansible

sudo yum install ansible -y



- Connect to ansible node install ansible

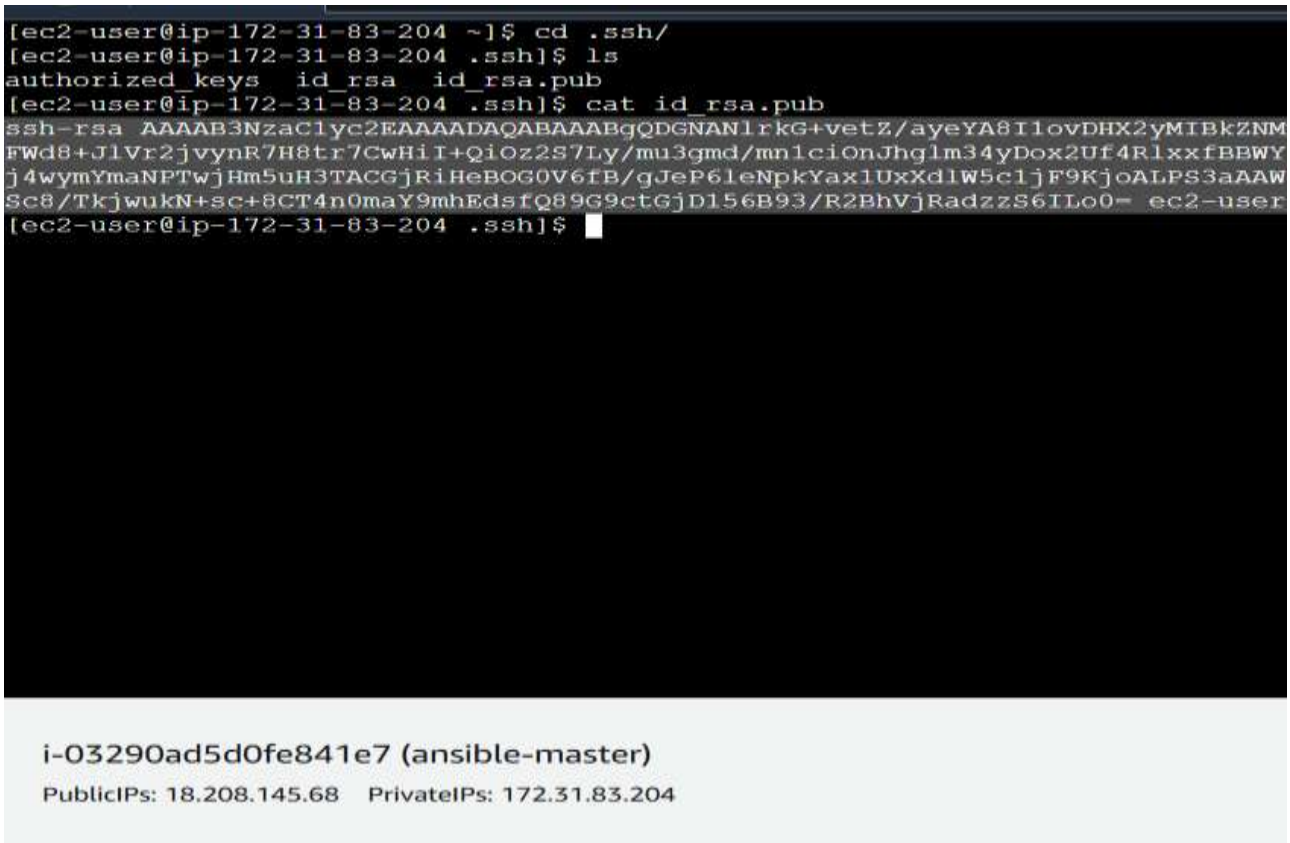
sudo yum install ansible -y



- Generate a keys in ansible master

ssh-keygen

- Switch to .ssh folder copy the id_rsa.pub file key



- Paste the master public key into authorized keys

- Like this paste ansible master public key into ansible node authorized keys

################ same like launch another node add the ansible master public key new node ##################################

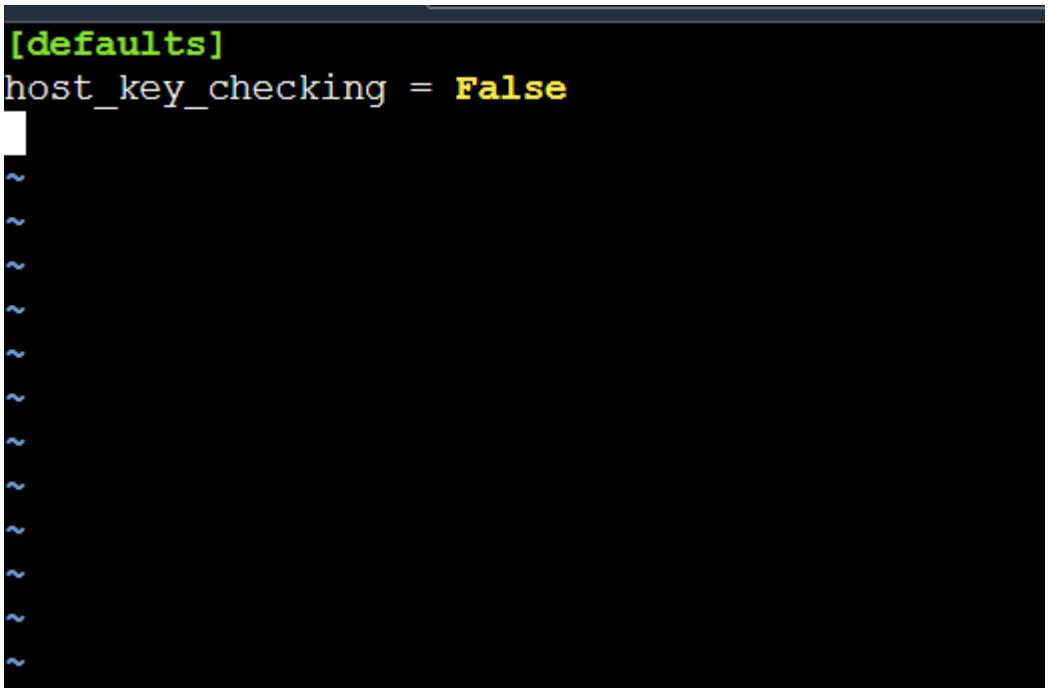
- Crete anible.cfg file and paste below mater

sudo vi /etc/ansible/ansible.cfg #create config file

- Paste this one into ansible.cfg file

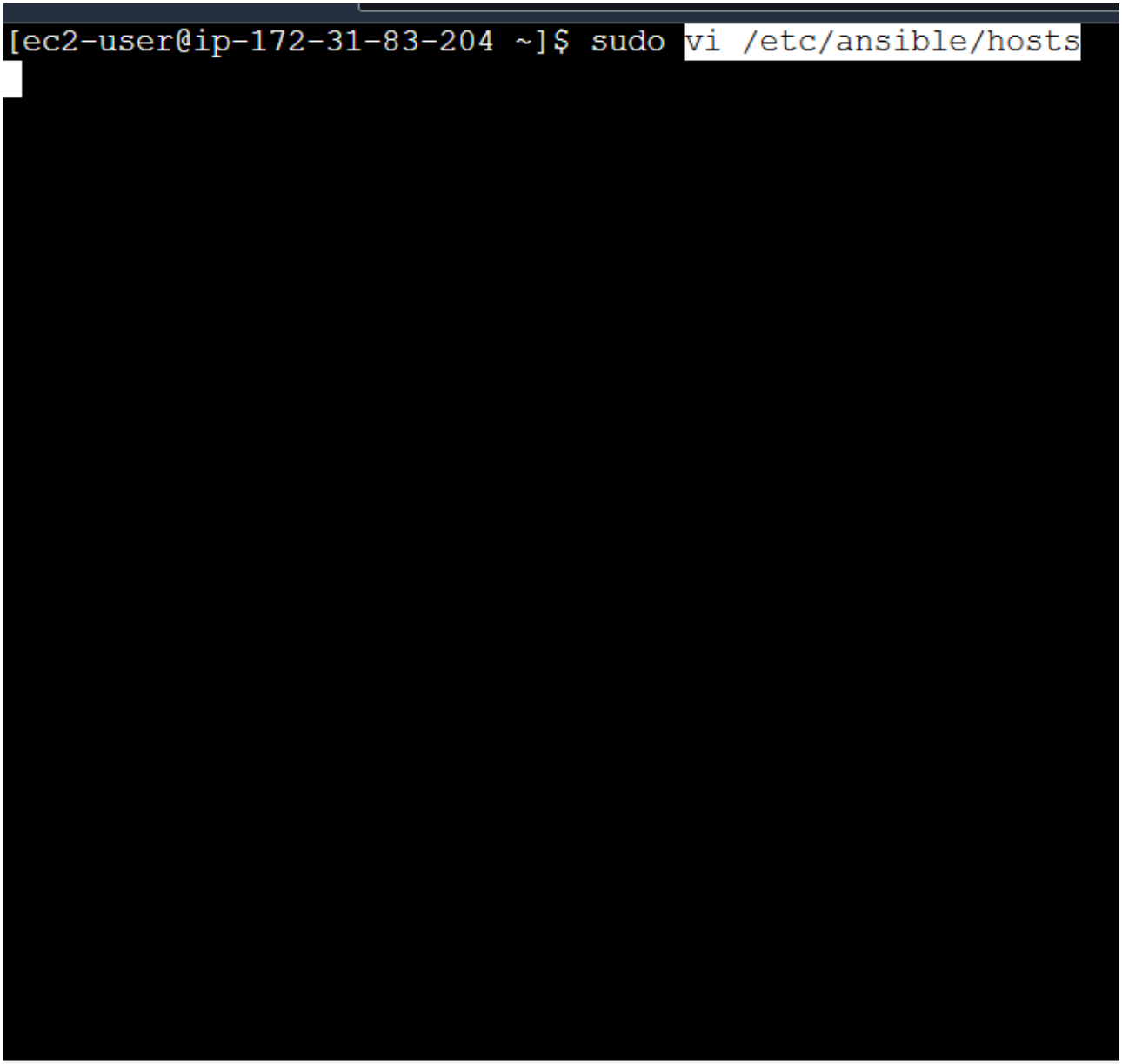
[defaults]

host_key_checking = False



- Creet hosts file for nodes connection and paste below mater

sudo vi /etc/ansible/hosts



- Paste your node private ip into hosts file

172.31.29..82

172.31.30.27

```
172.31.29.82
172.31.30.27
~
~
~
~
~
~
~
~
~
~
~
```

- After done above process just give ping command to check the node master connection
- If it is successfully connected mens you will get below output

ansible all -m ping

```
[root@ip-172-31-21-75 ~]# ansible all -m ping
[WARNING]: Platform linux on host 172.31.30.27 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter
could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.30.27 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.29.82 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter
could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.29.82 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
```

- Create a ansible playbook and paste bellow code

vi ansible-cloudwatch-playbook.yaml

```
[ec2-user@ip-172-31-83-204 ~]$ vi ansible-cloudwatc-playbook.yaml
```

- Pate below code

```
---
- name: Setup Apache and CloudWatch on EC2
  hosts: all
  become: yes
  tasks:

    - name: Install CloudWatch Agent
      yum:
        name: amazon-cloudwatch-agent
        state: present

    - name: Create CloudWatch agent config
      copy:
        dest: /opt/aws/amazon-cloudwatch-agent/bin/config.json
        content: |
          {
            "logs": {
              "logs_collected": {
                "files": {
                  "collect_list": [
                    {
                      "file_path": "/var/log/*",
                      "log_group_name": "LOG-FROM-EC2",
                      "log_stream_name": " {instance_id}",
                      "retention_in_days": 1
                    }
                  ]
                }
              }
            }
          }
```

- name: Start the CloudWatch agent

command: /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s

- name: Check CloudWatch agent status

command: /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status

register: cw_agent_status

- debug:

var: cw_agent_status.stdout_lines

```
---
- name: Setup Apache and CloudWatch on EC2
  hosts: all
  become: yes
  tasks:

    - name: Install CloudWatch Agent
      yum:
        name: amazon-cloudwatch-agent
        state: present

    - name: Create CloudWatch agent config
      copy:
        dest: /opt/aws/amazon-cloudwatch-agent/bin/config.json
        content: |
          {
            "logs": {
              "logs_collected": {
                "files": {
                  "collect_list": [
                    {
                      "file_path": "/var/log/*",
                      "log_group_name": "LOG-FROM-EC2",
                      "log_stream_name": "{instance_id}",
```

- this is yaml file

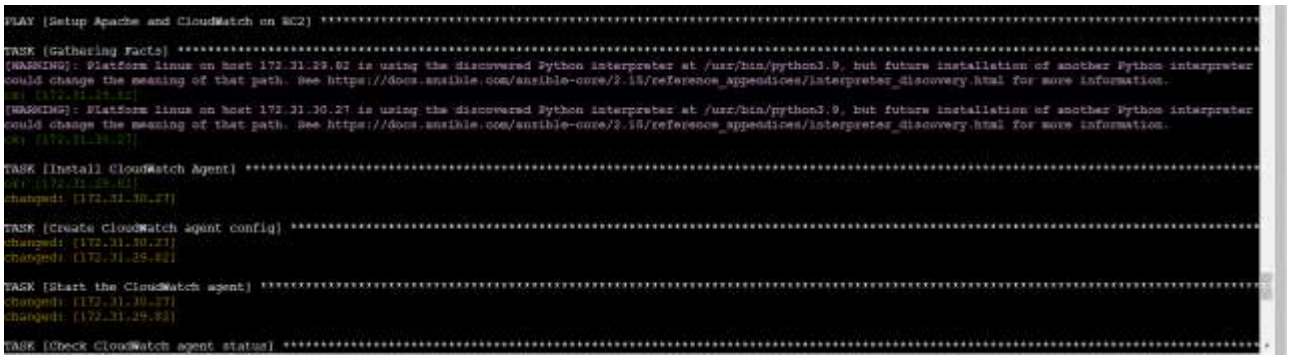
```
[ec2-user@ip-172-31-83-204 ~]$ ls
ansible-cloudwatc-playbook.yaml
[ec2-user@ip-172-31-83-204 ~]$
```

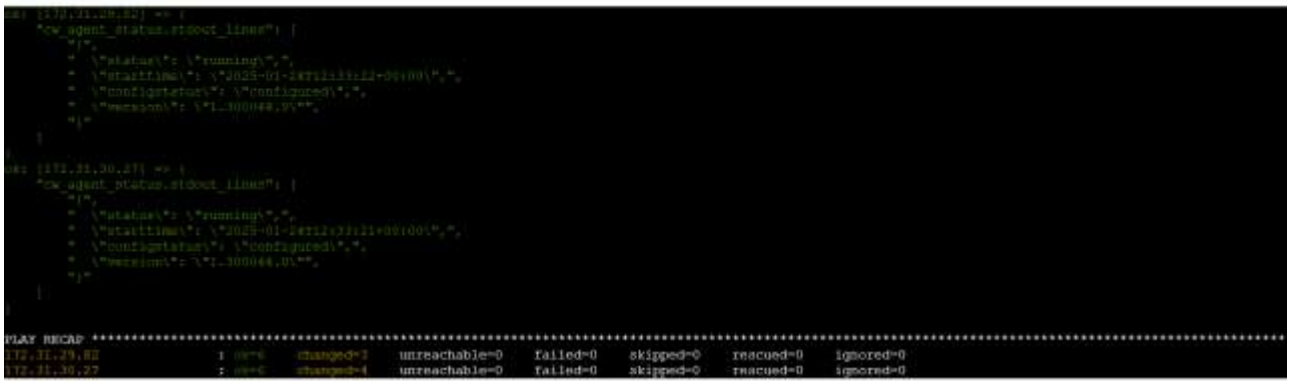
- execute the ansible-playbook yaml file

ansible-playbook ansible-cloudwatch-playbook.yaml

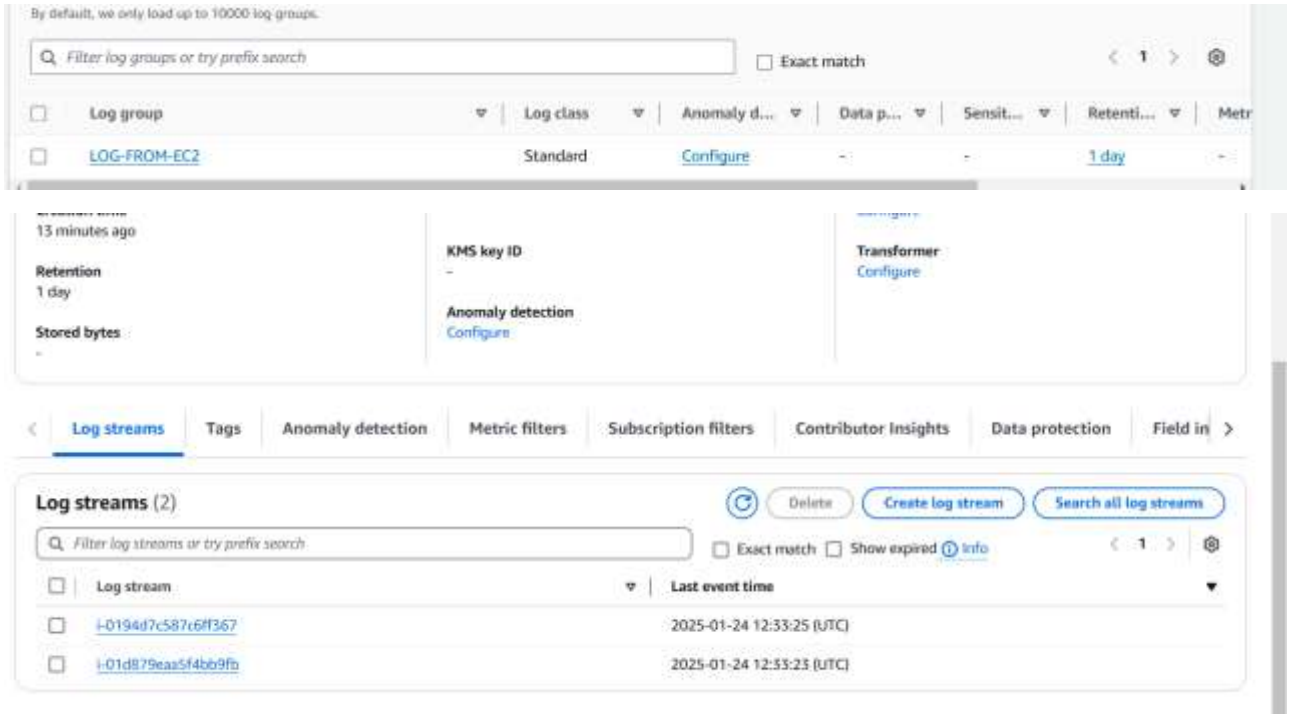
```
[ec2-user@ip-172-31-83-204 ~]$ ls
ansible-cloudwatc-playbook.yaml
[ec2-user@ip-172-31-83-204 ~]$ ansible-playbook ansible-cloudwatc-playbook.yaml
```

- successfully completed ansible playbook





- log grop is creted



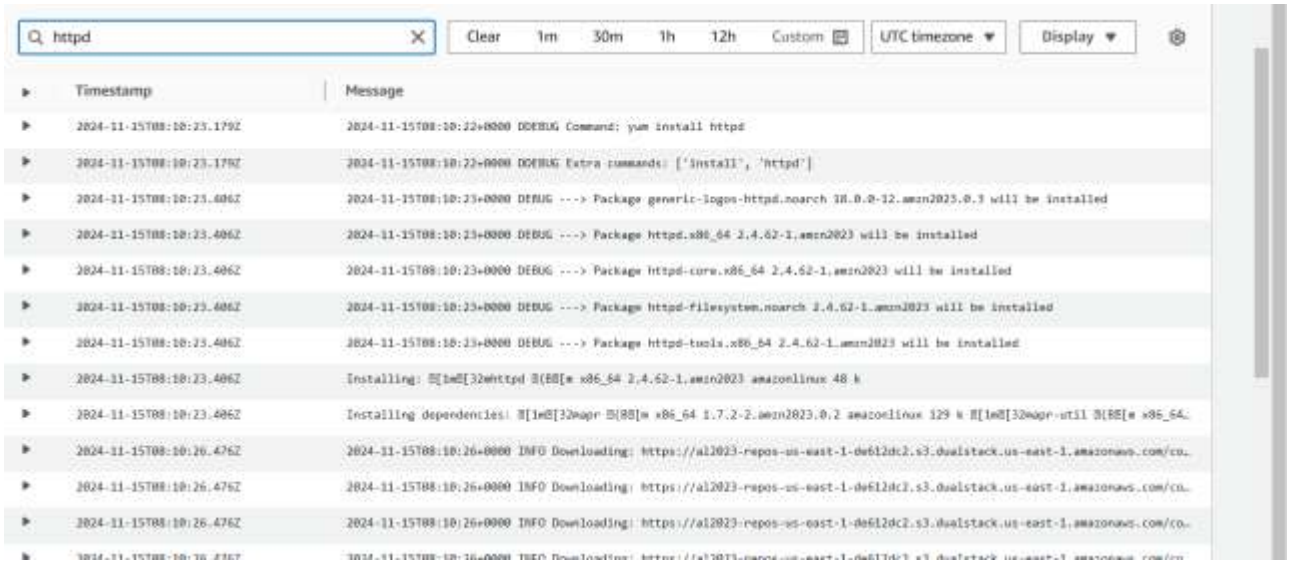
- iam installed httpd in ansible node

```
[ec2-user@ip-172-31-90-127 ~]$ sudo yum install httpd
Last metadata expiration check: 0:24:01 ago on Fri Nov 15 07:46:22 2024.
Dependencies resolved.
================================================================================
 Package                         Architecture                Version
================================================================================
Installing:
 httpd                           x86_64                      2.4.62-1.amzn2
Installing dependencies:
 apr                             x86_64                      1.7.2-2.amzn2
 apr-util                        x86_64                      1.6.3-1.amzn2
 generic-logos-httpd             noarch                      18.0.0-12.amzn
 httpd-core                      x86_64                      2.4.62-1.amzn2
 httpd-filesystem                noarch                      2.4.62-1.amzn2
 httpd-tools                     x86_64                      2.4.62-1.amzn2
 libbrotli                       x86_64                      1.0.9-4.amzn2
 mailcap                         noarch                      2.1.49-3.amzn2
Installing weak dependencies:
 apr-util-openssl                x86_64                      1.6.3-1.amzn2
 mod_http2                       x86_64                      2.0.27-1.amzn2
 mod_lua                         x86_64                      2.4.62-1.amzn2

Transaction Summary
================================================================================
Install  12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
```

i-0949b299511fc0f94 (ansible-node)

PublicIPs: 44.204.148.110 PrivateIPs: 172.31.90.127

- In log group I am able to see httpd logs


httpd
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
2024-11-15T08:10:23.179Z
2024-11-15T08:10:22+0000 DDEBUG Command: yum install httpd
2024-11-15T08:10:23.179Z
2024-11-15T08:10:22+0000 DDEBUG Extra commands: ['install', 'httpd']
2024-11-15T08:10:23.406Z
2024-11-15T08:10:23+0000 DEBUG ---> Package generic-logos-httpd.noarch 18.0.0-12.amzn2023.0.3 will be installed
2024-11-15T08:10:23.406Z
2024-11-15T08:10:23+0000 DEBUG ---> Package httpd.x86_64 2.4.62-1.amzn2023 will be installed
2024-11-15T08:10:23.406Z
2024-11-15T08:10:23+0000 DEBUG ---> Package httpd-core.x86_64 2.4.62-1.amzn2023 will be installed
2024-11-15T08:10:23.406Z
2024-11-15T08:10:23+0000 DEBUG ---> Package httpd-filesystem.noarch 2.4.62-1.amzn2023 will be installed
2024-11-15T08:10:23.406Z
2024-11-15T08:10:23+0000 DEBUG ---> Package httpd-tools.x86_64 2.4.62-1.amzn2023 will be installed
2024-11-15T08:10:23.406Z
Installing: [1m[32mhttpd [0m[x86_64 2.4.62-1.amzn2023 amazonlinux 48 k
2024-11-15T08:10:23.406Z
Installing dependencies: [1m[32mapr [0m[x86_64 1.7.2-2.amzn2023.0.2 amazonlinux 129 k [1m[32mapr-util [0m[x86_64
2024-11-15T08:10:26.476Z
2024-11-15T08:10:26+0000 INFO Downloading: https://al2023-repos-us-east-1-de612dc2.s3.dualstack.us-east-1.amazonaws.com/co
2024-11-15T08:10:26.476Z
2024-11-15T08:10:26+0000 INFO Downloading: https://al2023-repos-us-east-1-de612dc2.s3.dualstack.us-east-1.amazonaws.com/co
2024-11-15T08:10:26.476Z
2024-11-15T08:10:26+0000 INFO Downloading: https://al2023-repos-us-east-1-de612dc2.s3.dualstack.us-east-1.amazonaws.com/co