**Day 38: Ansible**

Configuration management:

* CM is the practice of handling changes systematically, so that a system maintains it integrity over time
* CM ensures that the current design and build state of the system is known, good and trusted.

Tools of CM

1. Puppet
2. CHEF
3. ANSIBLE
4. SALT STACK

1 and 2 are Pull configuration. 3 and 4 are Push configuration

Ansible is built on Python

A screenshot of a computer program

Description automatically generated

Steps to setup Ansible

1. Create 2 instances
2. Ansible host and Ansible Target
3. Establish the password less connect from Target to host  
   -- copy the ssh public key from host to target (Authorized keys)  
   -- Do it as a root user
4. apt update – to update all the current service
5. apt install ansible -y – to install the ansible
6. ansible –version – to check the version
7. We need to create a directory ansible under etc folder  
   mkdir /etc/ansible
8. create a file with the name hosts and add the private IPs of the targets  
   vi hosts  
   172.31.5.169
9. To check the connection, run the below command  
   ansible all -m ping

**hosts**

We can mention different Ips as shown below

|  |
| --- |
| 172.31.5.169  172.31.6.169 |

ansible all -m ping

* This will try to ping all the servers under hosts

ansible 172.31.5.169 -m ping

* This will try to ping the mentioned server under hosts

Note: Here m stands for module and ping is the module  
we have different modules like ping, shell, copy, file,apt etc..,

We can group them as well

|  |
| --- |
| [Java]  172.31.5.169  [Python]  172.31.6.169 |

ansible all -m ping

* This will try to ping all the servers under hosts

ansible Java -m ping

* This will try to ping the mentioned servers under Java group in hosts file

**shell:**

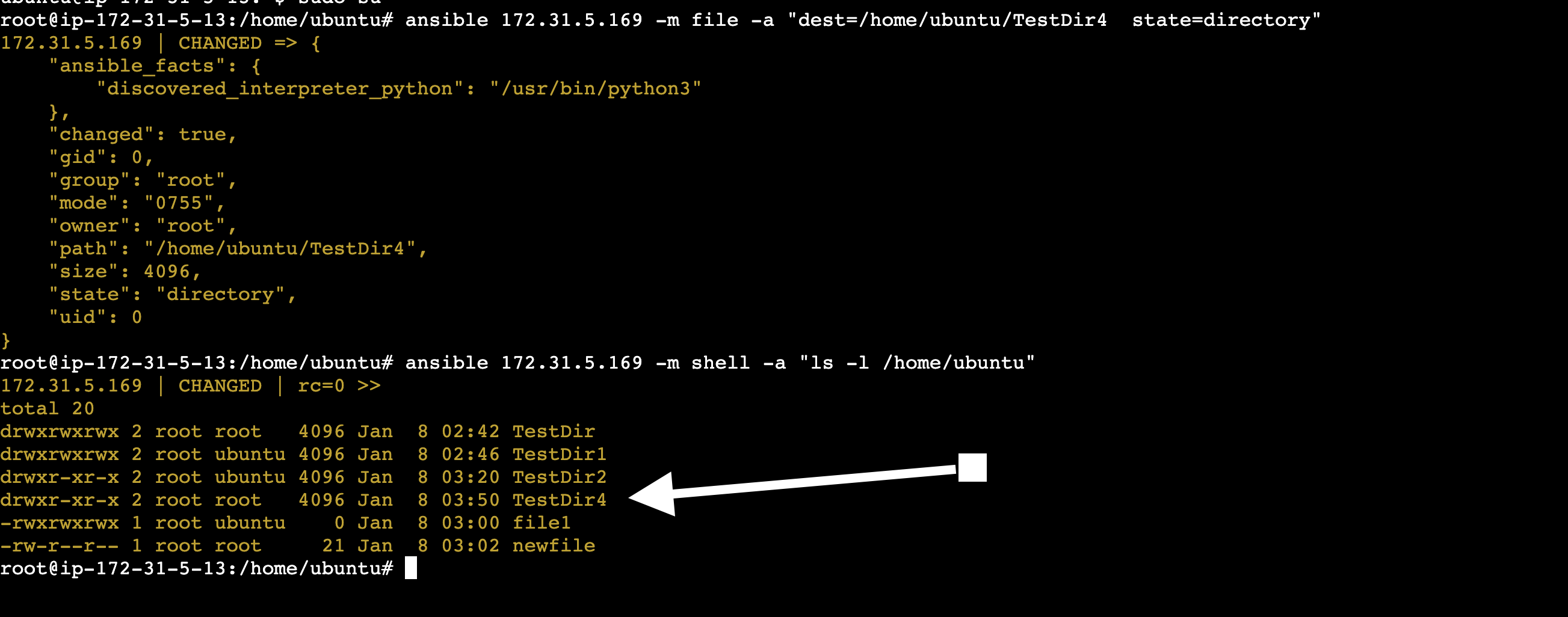
* using shell module we can run the unix commands on the target
  + ansible Java -m shell -a 'ls'
  + ansible Java -m shell -a 'ls -l'
  + ansible Java -m shell -a 'touch file1'
  + ansible Java -m shell -a 'cat file1'

**copy:**

* using copy module, we can move the files from host to targets
  + ansible Java -m copy -a "src=/etc/ansible/hosts dest=/home/ubuntu"
* We can also create a file under target
  + ansible Java -m copy -a 'dest=/home/ubuntu/newfile content="This is a sample file"'

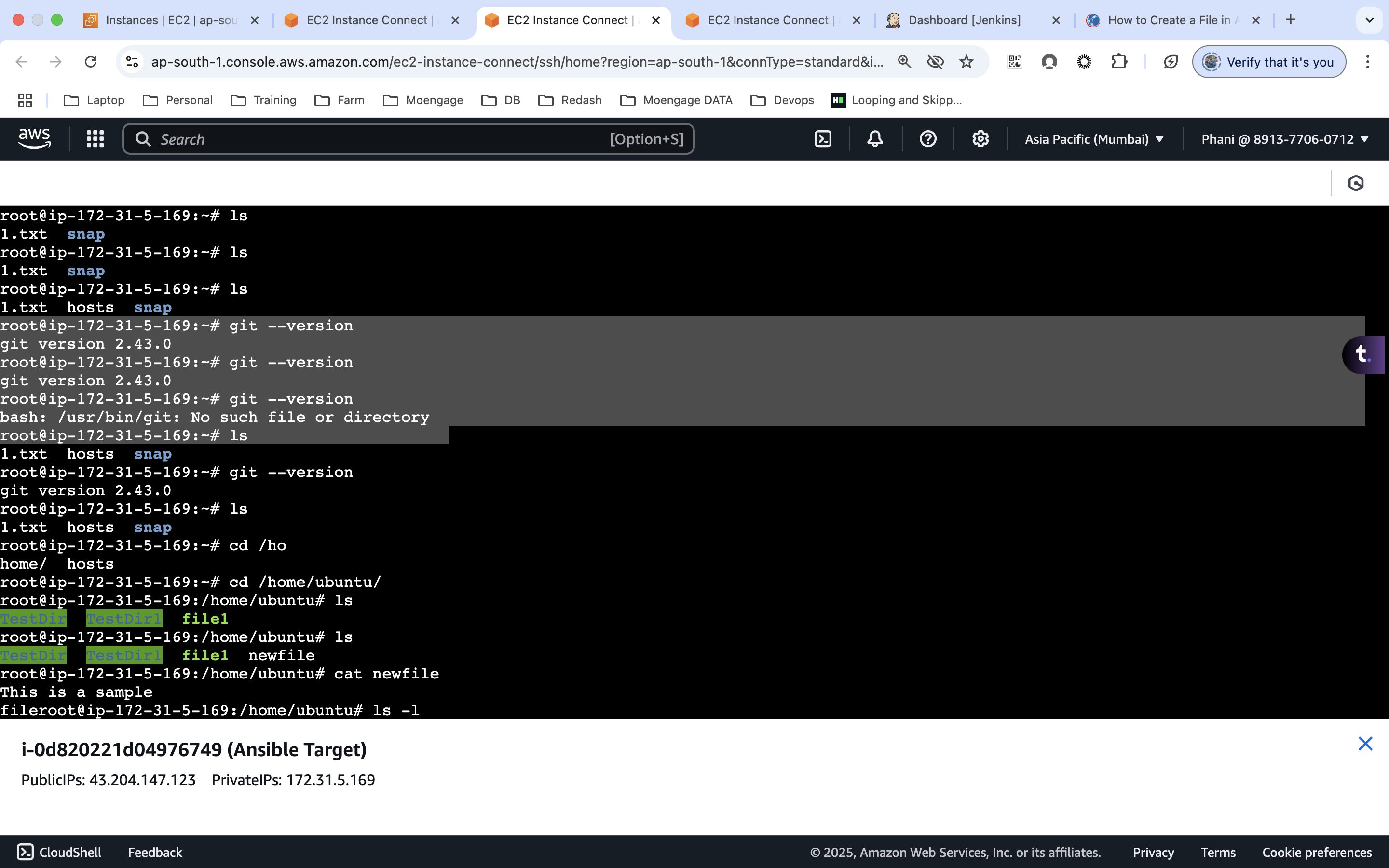
**file:**

* using file module, we can create directories and files under the target
  + ansible Java -m file -a "dest=/home/ubuntu/TestDir state=directory"
* To create a file, we use touch state
  + ansible Java -m file -a "dest=/home/ubuntu/File1 state=touch"
* We can also specify the permissions, ownership etc..
  + ansible 172.31.5.169 -m file -a "dest=/home/ubuntu/TestDir mode=777 state=directory"
  + ansible 172.31.5.169 -m file -a "dest=/home/ubuntu/TestDir mode=777 owner=root group=ubuntu state=directory"
  + Here we don’t user username:groupname for ownership, instead we refer to ownername:groupname
* The default values for permission and ownership will be 755 and root:root



**apt**

* Using apt module, we can install the services
* The below command will try to install the mentioned service, if it’s already present, it will ignore this
  + ansible Java -m apt -a "name=git state=present"
* The below command will try to uninstall the mentioned service
  + ansible Java -m apt -a "name=git state=absent"
* This will update the mentioned service
  + ansible Java -m apt -a "name=git state=latest"



**Day 39: Ansible**

|  |
| --- |
| ---  - hosts: all  tasks:  - name: installing\_the\_packages  apt:  name: git  state: present  become: yes |

test.yml

|  |
| --- |
| ---   * - hosts: all * tasks: * - name: installing\_the\_packages\_for\_ubuntu * when: ansible\_distribution=="Ubuntu" * apt: * name: git * state: present * become: yes * - name: installing\_the\_packages\_for\_centos * when: ansible\_distribution=="centos" * yum: * name: git * state: present * become: yes * - name: copying\_file * copy: * src: /etc/ansible/test.yml * dest: /home/ubuntu * - name: Modifying\_permission * file: * path: /home/ubuntu/test.yml * mode: 777 * owner: root * group: root * - name: create\_and\_add\_content * copy: * dest: /home/ubuntu/file2   content: "Welcome" |

test2.yml

|  |
| --- |
| * --- * - hosts: all * tasks: * - name: Listing\_files * shell: ls /home/ubuntu * register: list\_values * - debug: * var: list\_values.stdout\_lines * - name: date\_display * shell: date * register: date\_value * - debug:   var: date\_value.stdout\_lines |

test3.yml

|  |
| --- |
| * --- * - hosts: all * tasks: * - name: install\_multiple\_packages * apt: * name: ['git', 'maven'] * state: latest   update\_cache: yes |

test4.yml

|  |
| --- |
| * --- * - hosts: all * tasks: * - name: checking\_servers * ping: * - name: checking\_shell\_commands * shell: ls /home/ubuntu * register: ls\_command * - debug: * var: ls\_command.stdout\_lines * - name: moving\_a\_file * copy: * src: /etc/ansible/hosts * dest: /home/ubuntu * mode: 0777 * - name: permison\_f\_file * file: * dest: /home/ubuntu/hosts * state: touch * become: yes * - name: installoing * when: ansible\_distribution=="centos" * apt: * name: git * state: present * register: output * - debug:   var: output |

**Day 40: Ansible**

test2.yml

|  |
| --- |
| * --- * - hosts: all * tasks: * - name: Listing\_files * shell: ls1 /home/ubuntu * register: list\_values * ignore\_errors: yes * - debug: * var: list\_values.stdout\_lines * - name: date\_display * shell: date * register: date\_value * - debug:   var: date\_value.stdout\_lines |

ignore\_errors: yes

* this is used to ignore the errors, in this case ls1 is not a right command, if we don’t use the ignore\_errors, then it won’t execute the rest of the tasks
* We can use this at the global level  
  ---  
  - host: all  
  ignore\_errors

**Day 41: Ansible**

roles

To create a role  
ansible-galaxy init sample

ansible-playbook test100.yml

**Day 42: Ansible**

ansible collection

ansible-galaxy collection list

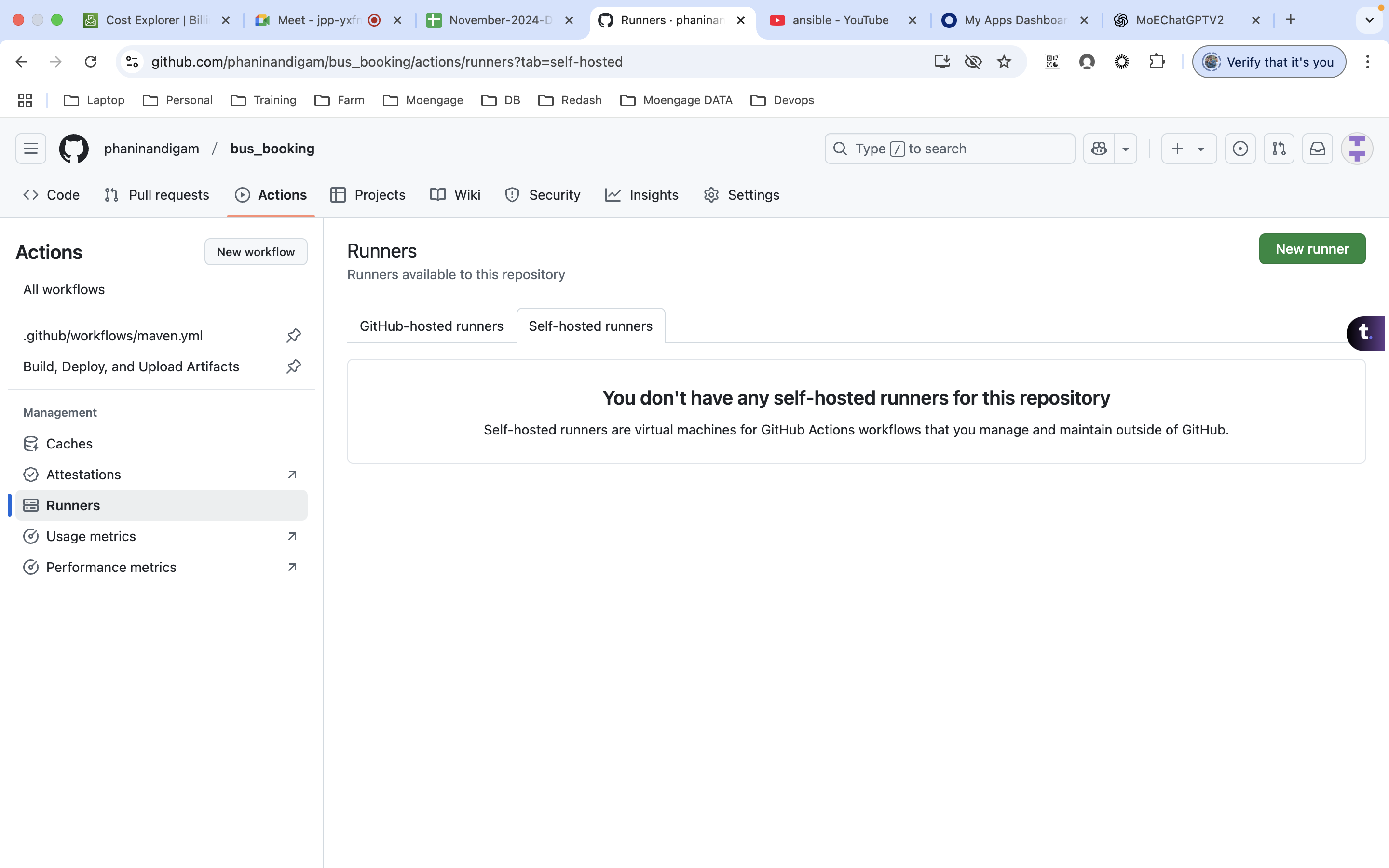
ansible-doc -l | grep ansible.posix (offical)

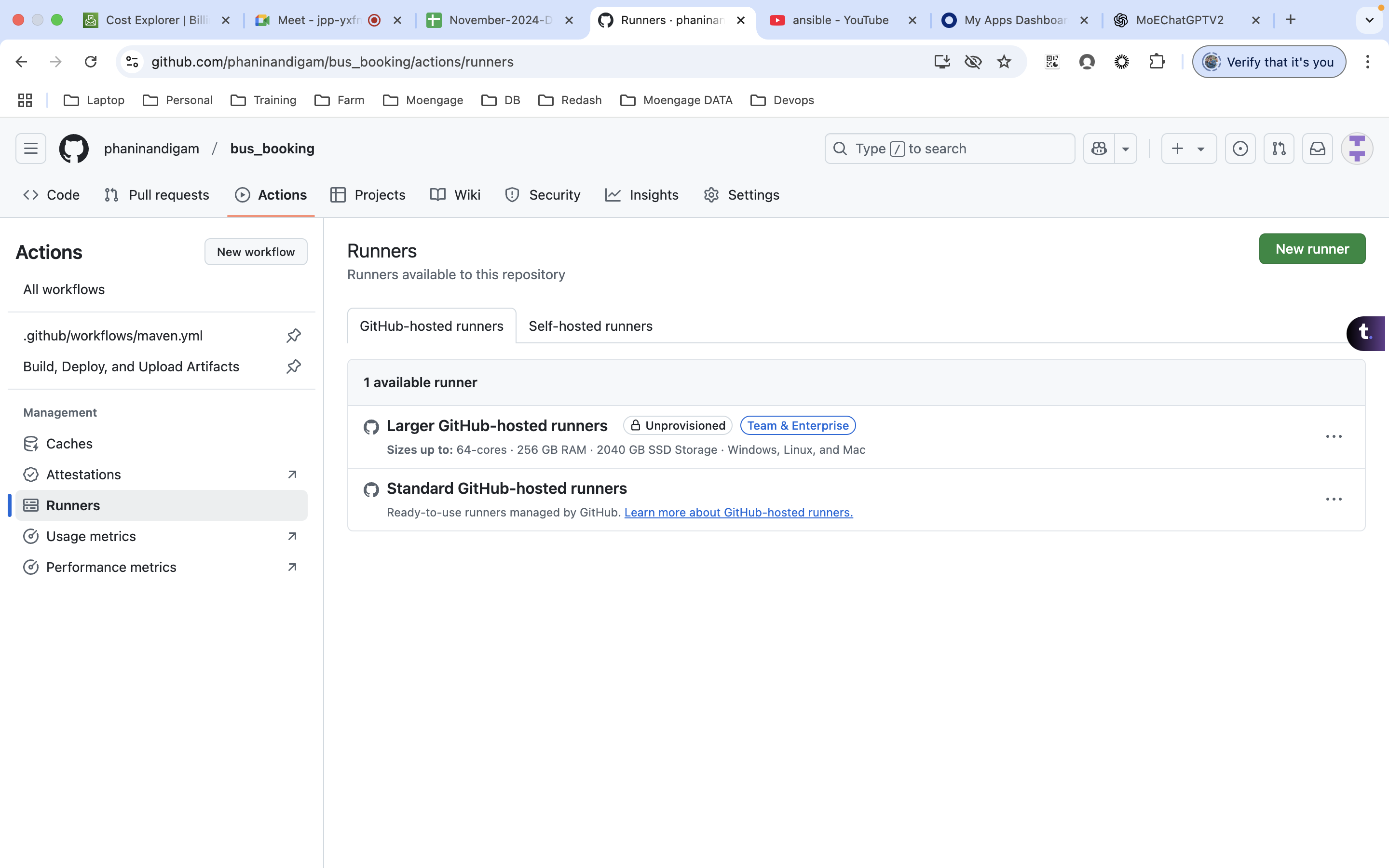
ansible-doc -l | grep community.general (community)

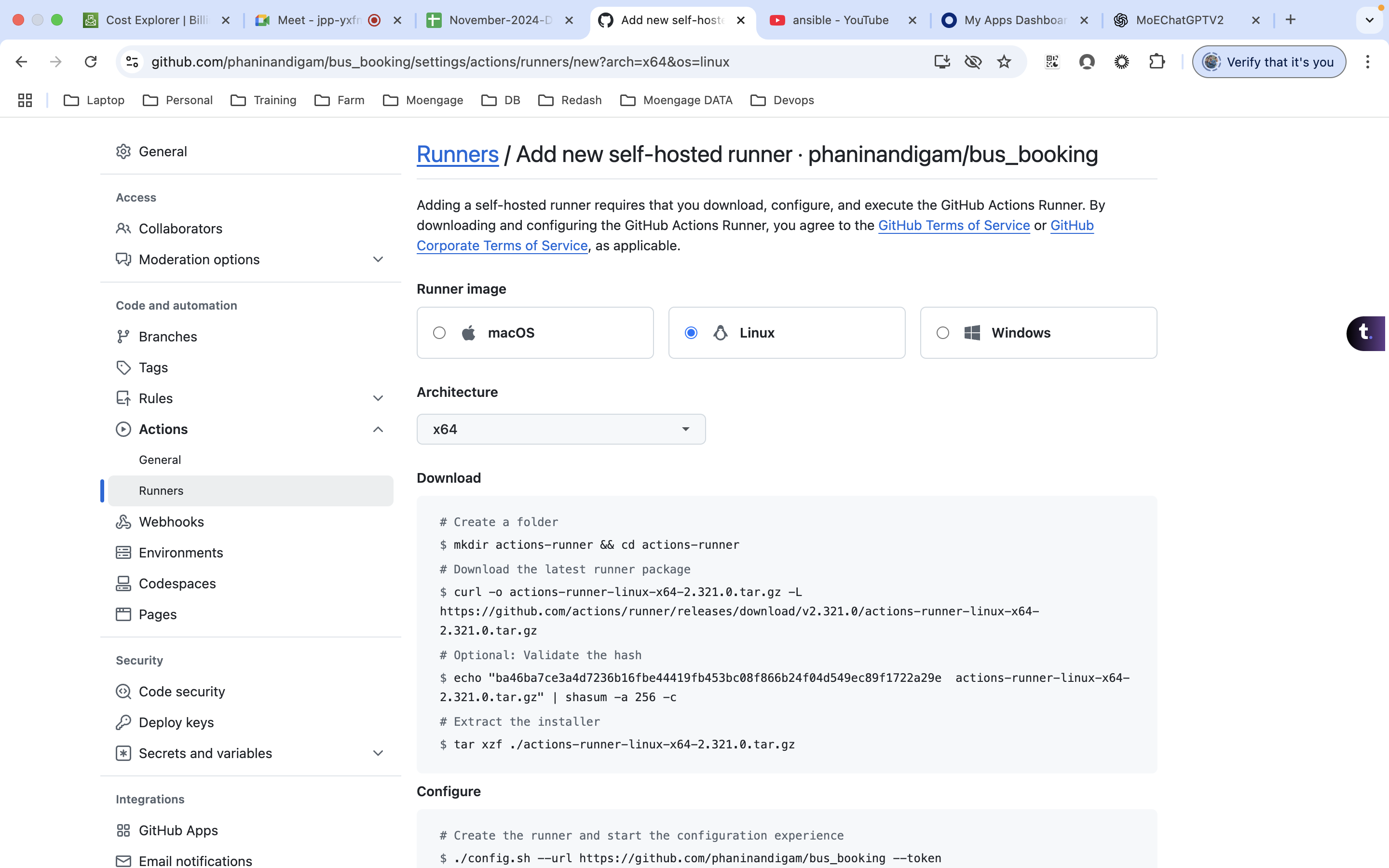
ansible-doc -l | grep amazon.aws (cloud)

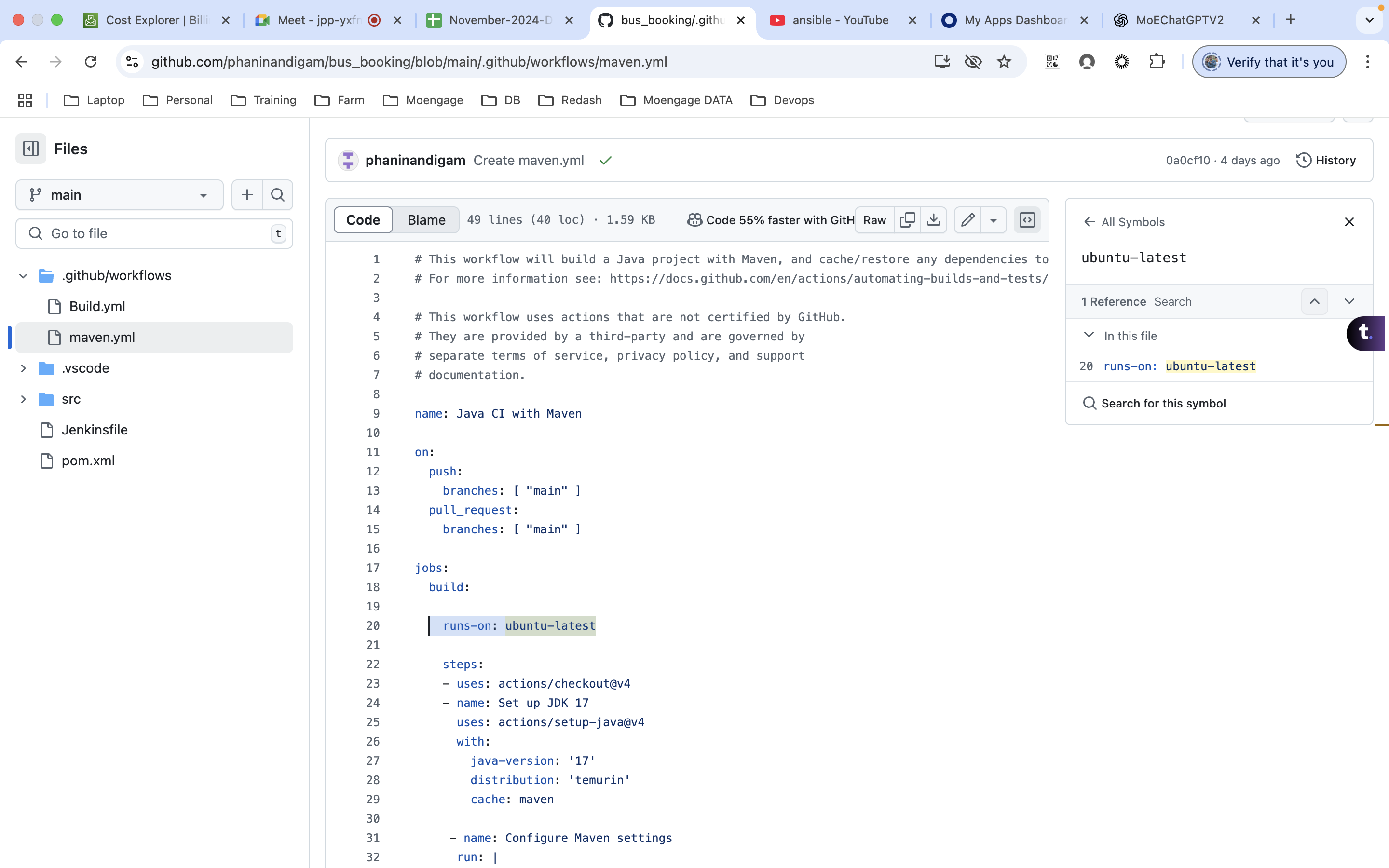
**Day 43: git actions**

Runners









**Day 44: Jfrog**

**Day 45: git actions**