Ansible

Contents

[Lab Setup on AWS 2](#_Toc188691079)

[Create a Keypair 2](#_Toc188691080)

[Setup lab servers 2](#_Toc188691081)

[Setup SSH connectivity 3](#_Toc188691082)

[On the **Ansible control** machine 4](#_Toc188691083)

[On **each Host** machine 4](#_Toc188691084)

[Setup Ansible 4](#_Toc188691085)

[Install Ansible on “Ansible Control machine” 4](#_Toc188691086)

[Verify Installation 4](#_Toc188691087)

[Configure Ansible 4](#_Toc188691088)

[Verify the changes 5](#_Toc188691089)

[Setup Host and inventories 5](#_Toc188691090)

[Create an inventory 5](#_Toc188691091)

[Verify lab setup 5](#_Toc188691092)

[Dynamic Inventory 7](#_Toc188691093)

[Install required tools 7](#_Toc188691094)

[Enable AWS plugin 7](#_Toc188691095)

[Create inventory file 7](#_Toc188691096)

[List all hosts from dynamic inventory 7](#_Toc188691097)

[Ad Hoc Command - Interacting with Hosts 8](#_Toc188691098)

[Ad-Hoc Command Format 8](#_Toc188691099)

[Choose target hosts 8](#_Toc188691100)

[Modules 8](#_Toc188691101)

[Ping 8](#_Toc188691102)

[Apt 8](#_Toc188691103)

[User 8](#_Toc188691104)

[File 8](#_Toc188691105)

[Copy 8](#_Toc188691106)

[Shell 8](#_Toc188691107)

[Debug 8](#_Toc188691108)

[Setup 8](#_Toc188691109)

[Playbooks 10](#_Toc188691110)

[YAML Reference 10](#_Toc188691111)

[Playbook Keywords 10](#_Toc188691112)

[Variables 10](#_Toc188691113)

[Override the default value of the variables: 11](#_Toc188691114)

[Loop 11](#_Toc188691115)

[Loop with list of maps 11](#_Toc188691116)

[Facts 12](#_Toc188691117)

[Conditional 12](#_Toc188691118)

[Tags (Filter tasks) 13](#_Toc188691119)

[Handlers 13](#_Toc188691120)

[Roles 14](#_Toc188691121)

[Install an appropriate role 14](#_Toc188691122)

[Use role in a playbook 14](#_Toc188691123)

[Vault 15](#_Toc188691124)

[Create an encrypted string 15](#_Toc188691125)

[Create a playbook which used the encrypted string 15](#_Toc188691126)

[Apply the playbook 15](#_Toc188691127)

[Verify the results 15](#_Toc188691128)

[Exercises 16](#_Toc188691129)

[Write and execute ad hoc command to: 16](#_Toc188691130)

[Ensure below configuration exists on s1 and s2 using ad-hoc cmds: 16](#_Toc188691131)

[Achieve the above using playbook 16](#_Toc188691132)

[Using playbooks, ensure that s1 and s2 have: 17](#_Toc188691133)

[Rewrite the below playbook to use loop: 17](#_Toc188691134)

[Conditional 17](#_Toc188691135)

# Lab Setup on AWS

## Create a Keypair

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html>

## Setup lab servers

Use the below terraform code to setup lab servers

# Note Please use the keypair name that you created in the previous step

terraform {

  required\_providers {

    aws = {

      source  = "hashicorp/aws"

      version = "5.83.1"

    }

  }

}

provider "aws" {

  # Configuration options

  region     = "us-east-1"

  # Replace below with your access credentials

  access\_key = "AKIA6ODU3KNLZYJA5POP"

  secret\_key = "0r7ou1EM7BzwQv6sZD5HMH94NrsIlSNorzIm8RYn"

}

locals {

  servers = ["A", "s1", "s2", "s3", "s4", "s5"]

}

resource "aws\_instance" "ansible-lab" {

  ami           = "ami-04b4f1a9cf54c11d0"

  instance\_type = "t2.micro"

  tags = {

    Name = each.value

  }

  for\_each = toset(local.servers)

  # Replace below with the created Keypair name

  key\_name = "iitk-cmat-2025-01-18"

  security\_groups = [aws\_security\_group.sg-1.name]

}

resource "aws\_security\_group" "sg-1" {

  egress {

    from\_port        = 0

    to\_port          = 0

    protocol         = "-1"

    cidr\_blocks      = ["0.0.0.0/0"]

    ipv6\_cidr\_blocks = ["::/0"]

  }

  ingress {

    from\_port        = 0

    to\_port          = 0

    protocol         = "-1"

    cidr\_blocks      = ["0.0.0.0/0"]

    ipv6\_cidr\_blocks = ["::/0"]

  }

  name = "my-sg"

}

## Setup SSH connectivity

Setup SSH connection between Ansible and host machines

### On the **Ansible control** machine

#### generate SSH keys

ssh-keygen -t rsa

(just press enter for any input prompts)

#### Print SSH public Key

cat ~/.ssh/id\_rsa.pub

IMPORTANT: Copy the above printed key

### On **each Host** machine

#### Add public key

Add public key (copied from the control machine) to the end of below file as a new line

Please DO NOT delete any existing data from the below file

nano ~/.ssh/authorized\_keys

# Setup Ansible

## Install Ansible on “Ansible Control machine”

|  |
| --- |
| sudo apt update sudo apt install -y software-properties-common sudo add-apt-repository --yes --update ppa:ansible/ansible sudo apt install -y ansible |

*Reference :* [*Official Ansible installation guide*](https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu)

### Verify Installation

|  |
| --- |
| ubuntu@ip-172-31-40-173:~$ ansible --version ansible [core 2.17.5]   config file = /etc/ansible/ansible.cfg   configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']   ansible python module location = /usr/lib/python3/dist-packages/ansible   ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections   executable location = /usr/bin/ansible   python version = 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] (/usr/bin/python3)   jinja version = 3.1.2   libyaml = True |

## Configure Ansible

|  |
| --- |
| echo [defaults] | sudo tee -a /etc/ansible/ansible.cfg echo host\_key\_checking = False | sudo tee -a /etc/ansible/ansible.cfg  echo interpreter\_python = auto\_silent | sudo tee -a /etc/ansible/ansible.cfg |

### Verify the changes

|  |
| --- |
| ubuntu@ip-172-31-40-173:~$ cat /etc/ansible/ansible.cfg # Since Ansible 2.12 (core):  # To generate an example config file (a "disabled" one with all default settings, commented out):  #               $ ansible-config init --disabled > ansible.cfg  #  # Also you can now have a more complete file by including existing plugins:  # ansible-config init --disabled -t all > ansible.cfg  # For previous versions of Ansible you can check for examples in the 'stable' branches of each version  # Note that this file was always incomplete  and lagging changes to configuration settings  # for example, for 2.9: https://github.com/ansible/ansible/blob/stable-2.9/examples/ansible.cfg  **[defaults]**  **host\_key\_checking = False**  **interpreter\_python = auto\_silent** |

## Setup Host and inventories

### Create an inventory

Create a file ~/inventory

|  |
| --- |
| s1 ansible\_host=ip-172-31-23-180  s2 ansible\_host=ip-172-31-27-10  s3 ansible\_host=ip-172-31-30-82  s4 ansible\_host=ip-172-31-24-190  s5 ansible\_host=ip-172-31-24-216 |

# Verify lab setup

On Ansible machine

|  |
| --- |
| ansible all -m ping -i ~/inventory |

Expected output

|  |
| --- |
| web | SUCCESS => {     "ansible\_facts": {         "discovered\_interpreter\_python": "/usr/bin/python3"     },     "changed": false,     "ping": "pong" } db | SUCCESS => {     "ansible\_facts": {         "discovered\_interpreter\_python": "/usr/bin/python3"     },     "changed": false,     "ping": "pong" } |

# Dynamic Inventory

A diagram of a server

Description automatically generated

## Install required tools

|  |
| --- |
| sudo apt install python3-boto3 |

## Enable AWS plugin

|  |
| --- |
| echo enable\_plugins=aws\_ec2 | sudo tee -a /etc/ansible/ansible.cfg |

## Create inventory file

*Note: The inventory file is a YAML configuration file and must end with aws\_ec2.{yml|yaml}.*

Add below lines to ~/inventory\_aws\_ec2.yml

|  |
| --- |
| plugin: amazon.aws.aws\_ec2 regions:   - us-east-1 aws\_access\_key: get-from-lab aws\_secret\_key: get-from-lab |

## List all hosts from dynamic inventory

|  |
| --- |
| ansible-inventory --list -i ~/inventory\_aws\_ec2.yaml |

# Ad Hoc Command - Interacting with Hosts

## Ad-Hoc Command Format

*ansible <hosts> -i <inventory> -m <modulename> -a <attributes> [-b]*

## Choose target hosts

|  |
| --- |
| ansible all -m ping # all hosts ansible web1 -m ping # single host ansible web\* -m ping # all hosts starting with web ansible web1,db2 -m ping # some specific hosts ansible all:'!db' -m ping # all hosts, except specific  ansible localhost -i ~/inventory -a "hostname" # explicit localhost |

## [Modules](https://docs.ansible.com/ansible/2.9/modules/modules_by_category.html)

### [Ping](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/ping_module.html)

|  |
| --- |
| ansible all -m ping -i ~/inventory |

### [Apt](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html)

|  |
| --- |
| ansible all -i ~/inventory -m apt -a "name=finger state=present update\_cache=yes" -b |

### [User](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/user_module.html)

|  |
| --- |
| ansible web\* -i ~/inventory -m user -a "name=alice state=present" -b |

### [File](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html)

|  |
| --- |
| ansible web\* -m file -i ~/inventory -a "path=/tmp/mydir state=directory" |

### [Copy](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html)

|  |
| --- |
| ansible all -i ~/inventory -m copy -a "dest=/tmp/hello-world content=HelloFromAnsible" |

### [Shell](https://docs.ansible.com/ansible/2.9/modules/shell_module.html#shell-module)

|  |
| --- |
| ansible all -i ~/inventory -m shell -a "hostname -f" |

### [Debug](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/debug_module.html)

|  |
| --- |
| ansible web1 -m debug -i ~/inventory -a "msg=HelloThere" |

### [Setup](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/setup_module.html)

|  |
| --- |
| ansible s1 -m setup -i ~/inventory |

# Playbooks

## YAML Reference

<https://www.youtube.com/watch?v=1uFVr15xDGg>

## [Playbook Keywords](https://docs.ansible.com/ansible/latest/reference_appendices/playbooks_keywords.html#playbook-keywords)

ansible-playbook -i <your-inventory> <path-to-playbook-file>

e.g. ansible-playbook -i ~/inventory playbook1.yml

- name: "manage server 1"

  hosts: s1

  become: true

  tasks:

    - name: "nginx"

      apt:

        name: "nginx"

    - name: "alice"

      user:

        name: "alice"

        uid: 1005

- name: "manage server 2"

  hosts: s2

  become: true

  tasks:

    - name: "java"

      apt:

        name: "openjdk-21-jdk"

    - name: "alice"

      user:

        name: "alice"

        uid: 1005

    - name: "dave"

      user:

        name: "dave"

        uid: 1006

## Variables

- name: "configure hosts with users and files"

  hosts: "all"

  vars:

    filecontent: "Hello\_World"

  tasks:

    - name: "ensure file1 is present"

      copy:

        dest: "/tmp/file1"

        content: "{{ filecontent }}"

    - name: "ensure file2 is present"

      copy:

        dest: "/tmp/file2"

        content: "{{ filecontent }}"

    - name: "ensure file3 is present"

      copy:

        dest: "/tmp/file3"

        content: "{{ filecontent }}"

### Override the default value of the variables:

ansible-playbook -i ~/inventory playbook1.yml --extra-vars "filecontent=Hello\_From\_Ansible"

## Loop

- name: Create users and files with content

  hosts: all

  become: true

  vars:

    filecontent: "hello world"

  tasks:

    - name: manage all users

      loop:

        - "alice"

        - "dave"

        - "charlie"

      user:

        name: "{{ item }}"

    - name: manage all files

      loop:

        - "/tmp/file1.txt"

        - "/tmp/file2.txt"

      copy:

        dest: "{{ item }}"

        content: "{{ filecontent }}"

### Loop with list of maps

- name: Create users and files with content

  hosts: all

  become: true

  vars:

    filecontent1: "hello 1 world"

    filecontent2: "hello 2 world"

  tasks:

    - name: manage all users

      loop:

        - "alice"

        - "dave"

        - "charlie"

      user:

        name: "{{ item }}"

    - name: manage all files

      loop:

        - filename: "/tmp/file1.txt"

          content: "{{ filecontent1 }}"

        - filename: "/tmp/file2.txt"

          content: "{{ filecontent2 }}"

      copy:

        dest: "{{ item.filename }}"

        content: "{{ item.content }}"

## Facts

Run the below command to display facts for a host

ansible web1 -m setup -i ~/inventory

You can refer any of the displayed facts in your playbook

- name: facts demo

  hosts: all

  tasks:

    - name: print facts

      debug:

        msg: "{{ ansible\_distribution }}"

    - name: print ipv4 address

      debug:

        msg: "{{ ansible\_default\_ipv4.address }}"

    - name: store fact in a file

      copy:

        dest: /tmp/from\_fact

        content: "{{ ansible\_default\_ipv4.address }}"

## Conditional

- name: "configure hosts with users and files"

  hosts: all

  vars:

    enable\_files: true

  tasks:

    - name: "ensure file1 is present"

      copy:

        dest: "{{ item }}"

        content: "some filecontent"

      loop:

        - "/tmp/file1"

        - "/tmp/file2"

        - "/tmp/file3"

      when: enable\_files == true

    - name: "print statement"

      debug:

        msg: "Hello I am a debug statement"

      when: ansible\_hostname == "ip-172-31-89-146"

ansible-playbook -i ~/inventory pb1.yml --extra-vars '{ enable\_files: False }'

## Tags (Filter tasks)

- name: "manage server 1"

  hosts: s1

  become: true

  tasks:

    - name: "nginx"

      apt:

        name: "nginx"

        state: "present"

      tags:

        - "pkg"

    - name: "alice"

      user:

        name: "alice"

        uid: 1005

      tags:

        - "user"

- name: "manage server 2"

  hosts: s2

  become: true

  tasks:

    - name: "java"

      apt:

        name: "openjdk-21-jdk"

        state: "present"

      tags:

        - "pkg"

    - name: "alice"

      user:

        name: "alice"

        uid: 1005

      tags:

        - "user"

    - name: "dave"

      user:

        name: "dave"

        uid: 1006

      tags:

        - "user"

ansible-playbook -i ~/inventory playbook-1.yml --tags user

## Handlers

- name: Handlers demo

  hosts: all

  tasks:

    - name: manage finger pacakge

      become: true

      apt:

        name: finger

        update\_cache: true

      notify:

        - "print success message"

  handlers:

    - name: "print success message"

      debug:

        msg: "pacakge installed successfully"

# Roles

Let’s Install Docker using a role

## Install an appropriate role

ansible-playbook -i ~/inventory role-docker.yml

## Use role in a playbook

- name: install docker using role

  hosts: s1

  become: true

  roles:

    - role: geerlingguy.docker

# Vault

Vault in a Playbook - Example

## Create an encrypted string

|  |
| --- |
| ansible-vault encrypt\_string mysecretpassword |

Input the desired password, and REMEMBER IT

## Create a playbook which used the encrypted string

vim playbook-to-demo-vault.yaml

- name: "encrypted string"

  hosts: db

  tasks:

          - name: "create a file containing access\_key"

            copy:

                    dest: /tmp/sensitive-file

                    content: !vault |

                             $ANSIBLE\_VAULT;1.1;AES256

                             31346266376233653634306662306264643238356432383962303336626637383266353337613466

                             3438316532353062666366333232653433623937306462660a326161373765663961316530313666

                             35646132343733633865333538663334303764333236366331626238323430396464363466336633

                             6266613038643735390a396133363665376563613566353865623864306363326430316335383036

                             61383038643439366231346662646465323162636638643131386230643339373837623265396534

                             3732333361343633666632306633623261323336633836363161

## Apply the playbook

|  |
| --- |
| ansible-playbook playbook-to-demo-vault.yaml --ask-vault-pass |

## Verify the results

On Master

|  |
| --- |
| ansible db -m shell -a "cat /tmp/sensitive-file" |

# Exercises

## Write and execute ad hoc command to:

1. Ensure that apache2 package is installed on s1 host
2. Ensure that bob user exists on all hosts
3. Use File module to ensure that a directory called /tmp/mydir exists on all hosts
4. Solution

ansible all -i ~/inventory -m file -a "name=/tmp/mydir state=directory"  
ansible s1 -i ~/inventory -m apt -a "name=apache2 state=present" –become

ansible all -i ~/inventory -m user -a "name=bob" –become

## Ensure below configuration exists on s1 and s2 using ad-hoc cmds:

1. user “alice”, “bob” and “charlie”
2. files “/tmp/file1”, “/tmp/file2” and “/tmp/file3”
   1. with content = “helloworld”
3. solution:

ansible all -i ~/inventory -m user -a "name=alice" --become  
ansible all -i ~/inventory -m user -a "name=bob" -b  
ansible all -i ~/inventory -m user -a "name=charlie " -b  
ansible all -i ~/inventory -m copy -a "dest=/tmp/file1 content=helloworld"  
ansible all -i ~/inventory -m copy -a "dest=/tmp/file2 content=helloworld"  
ansible all -i ~/inventory -m copy -a "dest=/tmp/file3 content=helloworld"

## Achieve the above using playbook

Solution:

- name: Create users and files with content

  hosts: all

  become: true

  tasks:

    - name: Create user alice with UID 1001

      user:

        name: alice

    - name: Create user dave with UID 1002

      user:

        name: Dave

    - name: Create user charlie with UID 1003

      user:

        name: charlie

    - name: Create file1.txt with hello world

      copy:

        dest: /tmp/file1.txt

        content: "hello world"

    - name: Create file2.txt with hello

      copy:

        dest: /tmp/file2.txt

        content: "hello world"

## Using playbooks, ensure that s1 and s2 have:

1. user ajay present
2. package finger installed

## Rewrite the below playbook to use loop:

|  |
| --- |
| - name: "configure hosts with users and files"   hosts: "web\*"   vars:     filecontent: "Hello\_World"   tasks:     - name: "ensure file1 is present"       copy:         dest: "/tmp/file1"         content: "{{ filecontent }}"     - name: "ensure file2 is present"       copy:         dest: "/tmp/file2"         content: "{{ filecontent }}"     - name: "ensure file3 is present"       copy:         dest: "/tmp/file3"         content: "{{ filecontent }}" |

## Conditional

1. create a play which runs on all hosts
2. add a task to install a package finger
3. package should not be installed if fact "ansible\_domain" is set to "ec2.internal"