2-Ansible

[Setup Ansible](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.6wkx9s42dniq)

[Install Ansible on Lab “master” machine](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.ws46c4g7nruo)

[Validate Installation](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.rawxpe3weduj)

[Configure Ansible defaults](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.q3f18rvlpz8a)

[Verify the configuration](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.n8trfld6gyl4)

[Setup default Inventory](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.3uq4fgb34ra5)

[Verify the added hosts](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.hzf0ty6vtef)

[Setup SSH connection between master and host machines](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.kcny2g8f52q2)

[Verify the SSH connection to hosts](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.b0hiyjxw3bhj)

[Ansible Connection Type](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.go3w1kj7q7n8)

[ansible-controller host](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.1mxcwh7gx0w6)

[Verify Ansible server is connecting properly](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.nkahyc8hgmx5)

[Interacting with Hosts - Ad Hoc Tasks](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.ytcwipmxy7t3)

[Ad-Hoc Command Format](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.7a1lcffwruzv)

[Choose Hosts](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.q6svucjcljgf)

[Manage Users](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.v2sc8i6oh4j2)

[Manage Files](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.6038zx21tw89)

[Manage Packages](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.anw8et4xz66s)

[Some More Modules](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.h5uwb5qk0xjy)

[Playbook](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.q9k7jy15nlfy)

[Execute Playbook](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.9p70fbtk7gjv)

[Introduction to playbook](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.m5kxq0a0wklo)

[Variables](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.f7iy1mzfrzst)

[Facts](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.spa094dl981q)

[Custom Facts](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.76xx2lh5ykwf)

[Loop](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.nscixtaw8dx2)

[Conditionals](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.9al3pr17a2lb)

[Handlers](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.uzvsteoj02dd)

[Tags](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.qo885umbircj)

[Disable Implicit Facts Gathering](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.xd3r0iolgbbz)

[Register](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.3tbrlx6t44t8)

[Templates](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.bct1tovfgwxd)

[Roles](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.8g7c6n9p1dys)

[Install Role](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.9x3argitvy9j)

[Use Role](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.4i59ct67gjnr)

[Create Role](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.lgu4bn1u7dhj)

[Initialize a role](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.pddjp7xxudka)

[Check out the Directory structure created for the role](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.cza5hbmtd1nx)

[Add Variables](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.pzagmzhkrpeb)

[Add tasks](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.icac5yo8ctfa)

[Use the role in a playbook](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.t9qjru3arlkb)

[Inventory in detail](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.epn9o89c12p3)

[Implicit Localhost](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.2mj2plfhy6ub)

[Custom Inventory](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.kvzdz0ag2lx7)

[Dynamic Inventory](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.fwlt7uuimclz)

[Install required tools](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.glczidhec8g4)

[Enable AWS plugin](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.l0np66q820k7)

[Create inventory file](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.cc8mie6umssl)

[List all hosts from dynamic inventory](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.214gbt1equkz)

[Ansible Vault](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.ln1fgo6ya5ck)

[Vault commands](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.wst3xxks4jth)

[Vault in a Playbook - Example](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.1b8hh7x0cai1)

[Create an encrypted string](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.4q9008adjnxh)

[Create a playbook which used the encrypted string](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.oxn4r8bcbksr)

[Apply the playbook](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.gq0kcekh1vn9)

[Verify the results](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.bfy0bf10gh2q)

[Ansible AWX](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.pjdeafc1vjqv)

[Installation](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.s4a1o6a0b7bd)

[Ensure that Ansible is installed : https://docs.ansible.com/ansible/latest/installation\_guide/installation\_distros.html#installing-ansible-on-ubuntu](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.58jxn927atet)

[Install required packages](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.9pxvrqgmk3lx)

[Install Docker](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.wceq9x148ytk)

[Download AWX source code](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.4ltqlcwe07y)

[Uninstall existing pip modules and Install new ones](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.4fjicw47wflh)

[Compile the AWX code into Docker-image](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.hys7961v3znd)

[Verify that docker image is created](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.f8vwhzluiwvg)

[Run AWX server and all supporting components](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.fuke9alxgubg)

[Verify AWX server is running](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.ddcix5xew1g)

[Open New Terminal for master](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.c7qe5fkyj0hs)

[Build AWS UI](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.k9plf1x81vyk)

[Verify the build](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.v5nb0alyrw16)

[Create Admin account](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.17xa9c98n3ec)

[Access the AWX UI](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.njumqz50whtf)

[Exercises](https://docs.google.com/document/d/1sLwjFdyuuGpn5qXc6sA3YvTAMZe-2Zz7ThWFVzdCdEw/edit#heading=h.bs0921hi1qkg)

ALL the commands in this document shall be run as ROOT

**sudo su -**

Setup Ansible

Install Ansible on Lab “master” machine

<https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu>

Validate Installation

Execute command after completing installation

|  |
| --- |
| ansible --version |

Expected Output

|  |
| --- |
| ansible [core 2.12.10]    config file = /etc/ansible/ansible.cfg    configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']    ansible python module location = /usr/lib/python3/dist-packages/ansible    ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections    executable location = /usr/bin/ansible    python version = 3.8.10 (default, May 26 2023, 14:05:08) [GCC 9.4.0]    jinja version = 2.10.1    libyaml = True |

Configure Ansible defaults

Execute below command to add some configurations in /etc/ansible/ansible.cfg.

**THIS IS A SINGLE COMMAND. COPY BOTH LINES AT ONCE**

|  |
| --- |
| echo "[defaults]  host\_key\_checking = False" > /etc/ansible/ansible.cfg |

Verify the configuration

Execute below command

|  |
| --- |
| tail -n2 /etc/ansible/ansible.cfg |

Expected output

|  |
| --- |
| [defaults]  host\_key\_checking = False |

Setup default Inventory

[On Master lab machine] Add the hostname of the ansible-hosts to the inventory file

sudo vim /etc/ansible/hosts

|  |
| --- |
| db ansible\_host=hostname for worker1 e.g. ip-172-31-31-225  web ansible\_host=hostname for worker2 |

Verify the added hosts

On master machine

|  |
| --- |
| ansible-inventory --list |

Expected Output

|  |
| --- |
| {      "\_meta": {          "hostvars": {              "db": {                  "ansible\_host": "ip-172-31-51-219"              },              "web": {                  "ansible\_host": "ip-172-31-57-43"              }          }      },      "all": {          "children": [              "ungrouped"          ]      },      "ungrouped": {          "hosts": [              "db",              "web"          ]      }  } |

Setup SSH connection between master and host machines

Execute All commands as root on all machines

1. On the master machine
   1. Check if a SSH keypair exists already
      1. ll ~/.ssh
      2. if SSH Key exists, skip step iii
      3. If SSH keys don't exist, generate SSH keys
         1. ssh-keygen
         2. (just press enter for any prompts)
   2. Print SSH public Key
      1. cat ~/.ssh/id\_rsa.pub
   3. Copy the above printed key
2. On each Host machine as root
   1. Please DO NOT delete any existing data from the below file
   2. Paste public key (copied from the master machine ) to the end of below file
      1. vim ~/.ssh/authorized\_keys

Verify the SSH connection to hosts

On master machine

|  |
| --- |
| ansible all -m ping |

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"  } |

Ansible Connection Type

List all the connection type: ansible-doc -t connection -l

Most popular ones: SSH (default), win\_rm, local

ansible-controller host

Add below line to the inventory to add ansible-controller as a host:

sudo vim /etc/ansible/hosts

|  |
| --- |
| ansible-server ansible\_connection=local |

Verify Ansible server is connecting properly

ansible all -m ping

Output:

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

Interacting with Hosts - Ad Hoc Tasks

Ad-Hoc Command Format

*ansible <hosts> -m <modulename> -a <attributes>*

Choose Hosts

|  |
| --- |
| ansible all -m ping # run a module on all hosts ansible web -m ping # run a module on a single host ansible web,db -m ping # run a module on some specific hosts ansible all:'!db' -m ping # run a module on all hosts, except some |

Manage Users

|  |
| --- |
| ansible all -m user "name=alice state=present" |

Manage Files

|  |
| --- |
| # Ensure that a directory exists ansible all -m file -a "path=/tmp/mydir state=directory" # Ensure that a file with a content exists ansible all -m copy -a "content=HelloWorld dest=/tmp/file-with-a-content" # Ensure a file does not exist ansible all -m file -a "path=/tmp/file-with-a-content state=absent" |

Manage Packages

|  |
| --- |
| # Ensure a package is installed on all hosts ansible all -m apt -a "name=finger" # Ensure a package is absent from all hosts ansible all -m apt -a "name=finger state=absent" |

Some More Modules

|  |
| --- |
| # Print a message on stdout ansible web -m debug -a "msg=Hello" # Execute shell commands on hosts ansible db -m shell -a "date" # print ansible facts for a host ansible web -m setup |

Playbook

Execute Playbook

|  |
| --- |
| ansible-playbook your-playbook-file.yaml  #Increase the verbosity of playbook execution ansible-playbook playbook.yaml -vvv |

Introduction to playbook

playbook-1-introduction.yml

|  |
| --- |
| - name: Manage database on DB server   hosts: db   tasks:     - name: Install Mysql Server       apt:         name: mysql-server         state: present         update\_cache: yes     - name: Manage DB user       user:         name: db\_user - name: manage web host   hosts: web   tasks:     - name: apache       apt:         name: apache2         update\_cache: yes     - name: user bob       user:         name: bob         uid: 2000 |

Variables

|  |
| --- |
| - name: "Variable Demonstration"   hosts: web   vars:           filecontent: "Hello Everyone"   tasks:           - name: "manage file 1"             copy:                     content: "{{ filecontent }}"                     dest: /tmp/file1           - name: "manage file 2"             copy:                     content: "{{ filecontent }}"                     dest: /tmp/file2           - name: "manage file 3"             copy:                     content: "{{ filecontent }}"                     dest: /tmp/file3 |

ansible-playbook *your-playbook.yaml* --extra-vars "filecontent=Haiyaa"

Facts

Run the below command to display facts for a host  
ansible web -m setup

You can refer any of the displayed facts in your playbook

|  |
| --- |
| - name: facts demo   hosts: all   tasks:           - name: print fqdn              debug:                     msg: "{{ ansible\_fqdn }}"           - name: print IP address             debug:                     msg: "{{ ansible\_default\_ipv4.address }}"           - name: create a file containing default IP address of the host             copy:                     content: "{{ ansible\_default\_ipv4.address }}"                     dest: /tmp/file-fact |

Custom Facts

on db host, create a custom fact

|  |
| --- |
| mkdir -p /etc/ansible/facts.d echo {\"zzzclassname\" :  \"simplilearn\"} > /etc/ansible/facts.d/classname.fact |

On ansible master

|  |
| --- |
| ansible db -m setup -a "filter=ansible\_local" |

Expected output

|  |
| --- |
| db | SUCCESS => {     "ansible\_facts": {         "ansible\_local": {             "classname": {                 "zzzclassname": "simplilearn"             }         },         "discovered\_interpreter\_python": "/usr/bin/python3"     },     "changed": false } |

Loop

|  |
| --- |
| - name: "Variable demo"   hosts: web   vars:           filecontent: "Hello World"   tasks:           - name: "manage all files"             copy:                     content: "{{ filecontent }}"                     dest: "{{ item }}"             loop:                     - /tmp/ansible\_file1                     - /tmp/ansible\_file2                     - /tmp/ansible\_file3 |

multivalue list elements

|  |
| --- |
| - name: "Variable Demonstration"   hosts: web   vars:   tasks:           - name: "create all files"             copy:                     content: "{{ item.text }}"                     dest: "{{ item.filename }}"             loop:                     - filename: /tmp/file1                       text: "Hello there"                     - filename: /tmp/file2                       text: "How do you do"                     - filename: /tmp/file3                       text: "Ciao" |

Conditionals

|  |
| --- |
| - name: "conditional demo"   hosts: all   vars:           allowed: "no"   tasks:           - name: print FQDN             debug:                    msg: "The FQDN is {{ ansible\_fqdn }}"             when: ansible\_hostname != "ip-172-31-31-225" # this task will be executed only if this condition is true           - name: install finger package             apt:                     name: finger                      when: allowed == "yes" # this task will be executed only if this condition is true |

Handlers

|  |
| --- |
| - name: "handlers"   hosts: db   tasks:           - name: "manage user called fred"             user:                     name: "fred"             notify:              - "create directory for fred"   handlers:           - name: "create directory for fred"             file:                     name: "/tmp/fred\_dir"                     state: directory |

Tags

|  |
| --- |
| - name: "manage db server"   hosts: db   tasks:           - name: "mysql server"             apt:                     name: mysql-server                     update\_cache: yes             tags:                     - package           - name: "user alice"             user:                     name: alice             tags:                     - user - name: "manage web host"   hosts: web   tasks:           - name: "apache"             apt:                     name: apache2                     update\_cache: yes             tags:                     - package           - name: "user bob"             user:                     name: bob                     uid: 2000             tags:                     - user |

ansible-playbook playbook-1.yaml --tags package

Disable Implicit Facts Gathering

|  |
| --- |
| - name: "handlers"   hosts: db   gather\_facts: false   tasks:           - name: "manage user called fred"             user:                     name: "george"             notify:              - "create directory for fred"   handlers:           - name: "create directory for fred"             file:                     name: "/tmp/fred\_dir"                     state: directory |

Register

|  |
| --- |
| - name: "register demo"   hosts: all   tasks:           - name: "execute shell"             shell: "hostname123"             register: hostname\_out             ignore\_errors: yes           - name: "print shell output details"             debug:                     msg: "Shell Execution was successful"             when: hostname\_out.rc == 0 |

Templates

Create a template file /tmp/local\_index\_template.j2

|  |
| --- |
| ip address =  {{ ansible\_default\_ipv4.address }}  FQDN = {{ ansible\_fqdn }} OS version = {{ ansible\_distribution }} |

Deploy the template as files on hosts

|  |
| --- |
| - name: "manage web server with custom index.html"   hosts: all   tasks:           - name: "apache2 package"             apt:                     update\_cache: yes                     name: apache2           - name: "Create default index file"             template:                     dest: /var/www/html/index.html                     src: local\_index\_template.j2 |

Roles

Install Role

Roles need to be installed on the Ansible master machine

|  |
| --- |
| ansible-galaxy install geerlingguy.java ansible-galaxy install geerlingguy.go |

Use Role

|  |
| --- |
| - name: "demonstrate Roles"   hosts: web   roles:           - role: geerlingguy.java |

Create Role

Initialize a role

|  |
| --- |
| cd /etc/ansible/roles/ sudo ansible-galaxy init helloworld |

Check out the Directory structure created for the role

|  |
| --- |
| ls -la /etc/ansible/roles/helloworld |

Add Variables

|  |
| --- |
| sudo vim /etc/ansible/roles/helloworld/vars/main.yml |

|  |
| --- |
| --- # vars file for helloworld guestname: "John Doe" |

Add tasks

|  |
| --- |
| sudo vim /etc/ansible/roles/helloworld/tasks/main.yml |

|  |
| --- |
| --- # tasks file for helloworld - name: "Print name"   debug:           msg: "Hello {{ guestname }}" - name: "Welcome to Devops"   debug:           msg: "Welcome to the Devops club!" |

Use the role in a playbook

|  |
| --- |
| - name: "use a custom role"   hosts: all   roles:           - role: helloworld             guestname: Amit # override the value of variable (optional) |

Apply the playbook and watch the magic!!

Inventory in detail

Implicit Localhost

Manage localhost without adding it to inventory

|  |
| --- |
| ansible localhost -m ping |

Custom Inventory

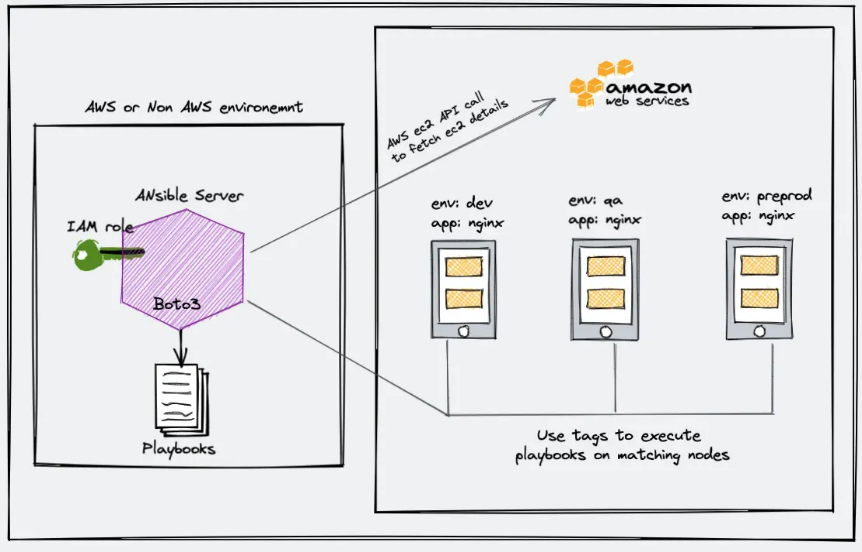
You can create your own inventories and pass them to ***any*** ansible command

|  |
| --- |
| ansible all -m ping -i custom-ansible-inventory |

e.g.

ansible-inventory --list -i ~/ansible\_code\_2023-08-19/my\_inventory

Dynamic Inventory



Install required tools

|  |
| --- |
| sudo apt install python3-pip pip3 install boto3 |

Enable AWS plugin

|  |
| --- |
| echo "enable\_plugins=aws\_ec2" >> /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 ~/my\_aws\_ec2.yaml

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

List all hosts from dynamic inventory

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

Ansible Vault

Vault commands

Create an encrypted file

ansible-vault create sensitive-data-vault

View encrypted data

cat sensitive-data-vault

View decrypted data

ansible-vault view sensitive-data-vault

Edit encrypted file

ansible-vault edit sensitive-data-vault

Change the password for an encrypted file

ansible-vault rekey sensitive-data-vault

Encrypt a string

ansible-vault encrypt\_string thisisuserpassword

Vault in a Playbook - Example

Create an encrypted string

|  |
| --- |
| ansible-vault encrypt\_string dF4sOMU266WMy1rBt2JFeKakjIGFWIs0Oz5qUXPG |

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" |

Ansible AWX

<https://www.ansible.com/products/controller>

<https://www.ansible.com/community/awx-project>

Perform below activities on Master , as ROOT user

Installation

Ensure that Ansible is installed : <https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu>

Install required packages

|  |
| --- |
| sudo apt install python3-pip openssl |

Install Docker

Reference : <https://docs.docker.com/engine/install/ubuntu/>

|  |
| --- |
| * sudo apt-get remove docker docker-engine docker.io containerd runc * sudo apt-get update * sudo apt-get install ca-certificates curl gnupg * sudo install -m 0755 -d /etc/apt/keyrings * curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg * sudo chmod a+r /etc/apt/keyrings/docker.gpg * echo "deb [arch="$(dpkg --print-architecture)"  signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu  "$(. /etc/os-release && echo "$VERSION\_CODENAME")" stable" |  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null * sudo apt-get update * sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-compose |

Download AWX source code

|  |
| --- |
| mkdir ~/awx cd ~/awx git clone <https://github.com/ansible/awx>  cd awx git checkout 22.7.0 |

Uninstall existing pip modules and Install new ones

|  |
| --- |
| pip3 uninstall -y boto3  pip3 install docker requests==2.20.0 |

Compile the AWX code into Docker-image

|  |
| --- |
| cd ~/awx/awx sudo make docker-compose-build |

*Please wait for this to complete. It may take many minutes*

Verify that docker image is created

|  |
| --- |
| docker images |

you should see similar output

|  |
| --- |
| REPOSITORY                  TAG       IMAGE ID       CREATED         SIZE ghcr.io/ansible/awx\_devel   HEAD      2bedcff5b4fa   3 minutes ago   1.93GB |

Run AWX server and all supporting components

|  |
| --- |
| cd ~/awx/awx sudo make docker-compose |

*NOTE: Terminal will now be occupied.*

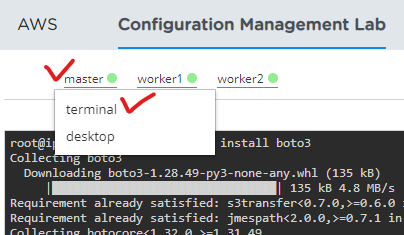
*Please open a new terminal window for Master machine to proceed further*

Verify AWX server is running

if you see output similar to below, then AWX server is running fine

|  |
| --- |
| redis\_1\_1   | 1:M 16 Sep 2023 14:47:19.059 \* 100 changes in 300 seconds. Saving... redis\_1\_1   | 1:M 16 Sep 2023 14:47:19.059 \* Background saving started by pid 11 redis\_1\_1   | 11:C 16 Sep 2023 14:47:19.065 \* DB saved on disk redis\_1\_1   | 11:C 16 Sep 2023 14:47:19.065 \* Fork CoW for RDB: current 0 MB, peak 0 MB, average 0 MB redis\_1\_1   | 1:M 16 Sep 2023 14:47:19.160 \* Background saving terminated with success |

Open New Terminal for master



Build AWS UI

In the new terminal, run below command, as ROOT

|  |
| --- |
| sudo docker exec tools\_awx\_1 make clean-ui  sudo docker exec tools\_awx\_1 chmod -R 777 /awx\_devel/awx/ui/build/ sudo docker exec tools\_awx\_1 make ui-devel |

*Please wait for this to complete. It may take many minutes*

Verify the build

if you see output similar to below, then AWX UI is built successfully

|  |
| --- |
| The bundle size is significantly larger than recommended. Consider reducing it with code splitting: https://goo.gl/9VhYWB You can also analyze the project dependencies: https://goo.gl/LeUzfb  The project was built assuming it is hosted at ./. You can control this with the homepage field in your package.json.  The build folder is ready to be deployed.  Find out more about deployment here:    https://cra.link/deployment  touch awx/ui/.ui-built make[1]: Leaving directory '/awx\_devel' |

Create Admin account

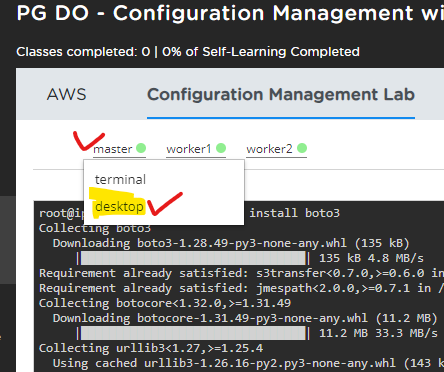
In the new terminal (already launched previously), run below command, as ROOT

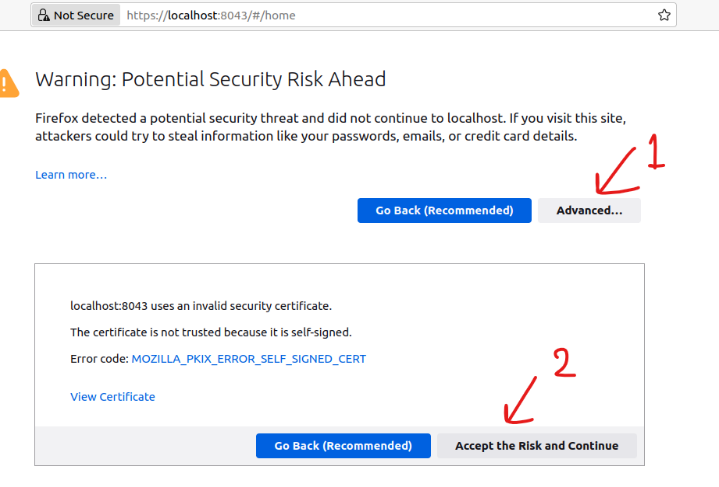
|  |
| --- |
| sudo docker exec -ti tools\_awx\_1 awx-manage createsuperuser |

|  |
| --- |
| Username: devops       Email address: amit@awx.com Password: <type your desired password> Password (again): <type your desired password> Superuser created successfully. |

Access the AWX UI

On the browser in the lab machine, access the UI: <https://localhost:8043/#/home>





Integrate with Jenkins

On Master Machine

Install Dependencies:

|  |
| --- |
| sudo apt update && apt-get install openjdk-11-jdk ca-certificates -y |

Install Jenkins:

If Jenkins is not installed already on the master machine, please run the commands under ***“Debian/Ubuntu -> Long Term Support***” section on the below web page:

|  |
| --- |
| <https://www.jenkins.io/doc/book/installing/linux/#long-term-support-release> |

Jenkins Post Installation Step:

Once done, follow the “***Post-installation setup wizard***” steps in this guide: <https://www.jenkins.io/doc/book/installing/linux/#setup-wizard>

Exercises

1. Write and execute ad hoc tasks to:
   1. Ensure bob user exists on web host with user-id 2000
   2. Ensure that the below file does not exist on all hosts "/etc/apt/sources.list.d/kubernetes.list"
   3. Ensure that apache2 package is installed on web host

Solution:

|  |
| --- |
| ansible web -m user -a "name=bob uid=2000" ansible all -m file -a "name=/etc/apt/sources.list.d/kubernetes.list state=absent" ansible web -m apt -a "name=apache2 update\_cache=yes" |

1. Write a play in a playbook to
   1. Ensure that the file does not exist on **all** hosts "/etc/apt/sources.list.d/kubernetes.list"

Solution:

|  |
| --- |
| - name: "remove kubernetes source file"   hosts: all   tasks:           - name: "delete the k8 source file"             file:                     name: "/etc/apt/sources.list.d/kubernetes.list"                     state: absent |

1. Variables
   1. create a user called “charlie” on “db” host.
   2. ensure that the username is taken from a variable in the playbook
2. Facts
   1. Create below custom fact on web host, and print it on ansible master

fact name: batch\_start

fact value: 2023-08-19

Solution

On web host, create a custom fact

|  |
| --- |
| mkdir -p /etc/ansible/facts.d echo {\"batch\_start\" :  \"2023-08-19\"} > /etc/ansible/facts.d/classname.fact |

On ansible master

|  |
| --- |
| ansible web -m setup -a "filter=ansible\_local" |

1. Loop
   1. Create below users as part of a single task (use loops)
      1. alice
      2. bob
      3. charlie
      4. dave
      5. edith

Solution

|  |
| --- |
| - name: "loop"   hosts: db   tasks:           - name: "Create User"             user:                     name: "{{ item }}"             loop:                     - "alice"                     - "bob"                     - "charlie"                     - "dave"                     - "edith" |

1. 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 "us-west-2.compute.internal"
   4. solution

|  |
| --- |
| - name: "Install package with conditional"   hosts: all   tasks:          - name: "install package finger"            apt:                   name: "finger"                   state: present            when: ansible\_domain != "us-west-2.compute.internal" |

1. Handlers
   1. install a package called finger on all hosts
   2. if the package is installed, create a file called “/tmp/finger.conf” with content “id=1”
   3. file should not be created if package is not installed
   4. Solution

|  |
| --- |
| - name: Install APT Package named "finger"   hosts: all   become: yes   tasks:     - name: Install Finger Package       apt:         update\_cache: yes         name: finger       notify: "Create file"   handlers:     - name: "Create file"       copy:         content: "id=1"         dest: /tmp/finger.conf |

1. Install nodejs on all the hosts using a role