20th International Conference on Network and Service Management
Prague, Czech Republic // 28 - 31 October 2024
AI-Powered Network and Service Management for Tomorrow's Digital World

How to Ansible

Richard Plný
FIT CTU in Prague
plnyrich@fit.cvut.cz

27th October 2024
Ph.D. School, Prague
Czech Republic

Outline

Part 1 - Basics

Problem

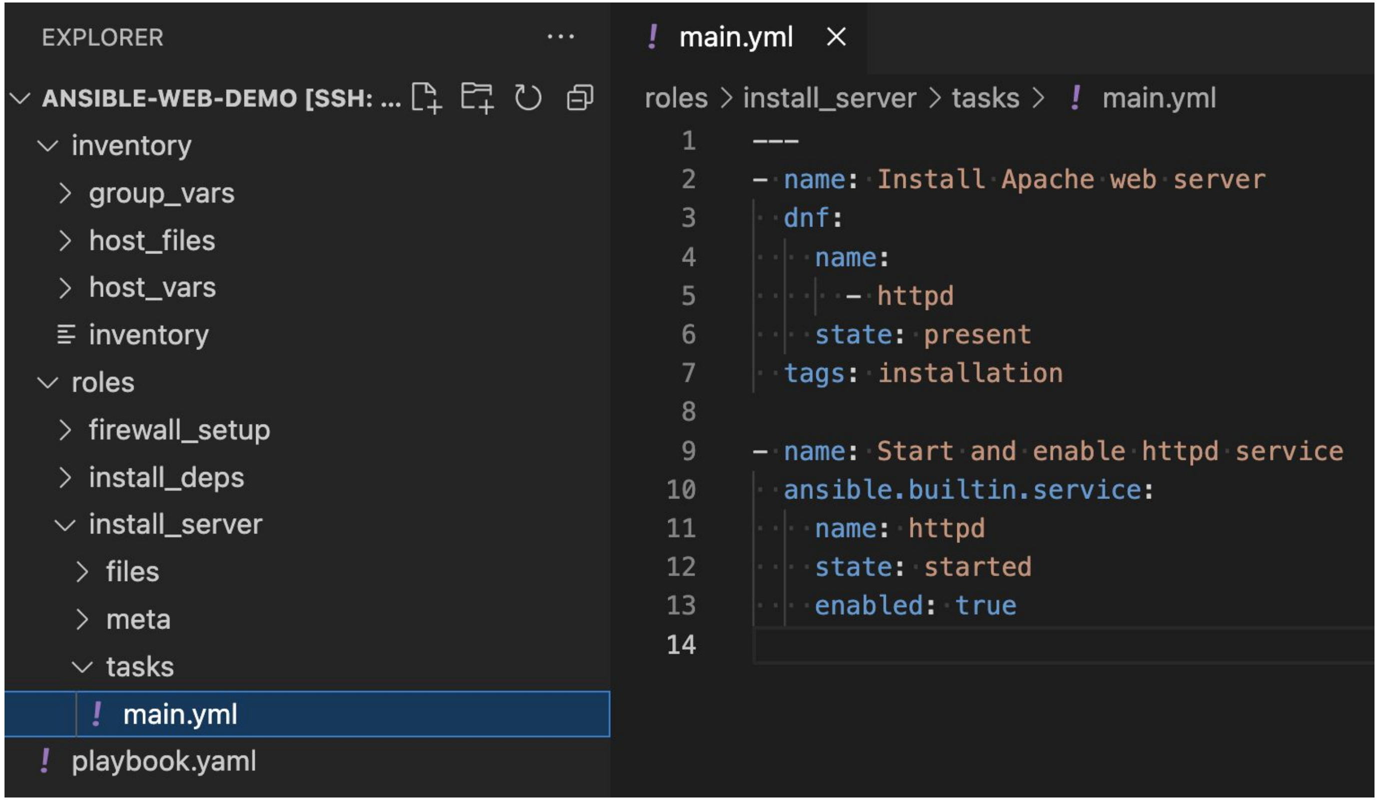
- We often need to configure a new server quickly..
  - Deployment of an application
  - Prepare an exact environment for research
  - Manage cloud machines
- A lot of work manually…

=> **Ansible**

What is Ansible?

- Automation engine for:
    - Provisioning
    - Configuration Management
    - Application Deployment
    - Orchestration
- YAML format
- Main website: https://www.ansible.com
- GitHub: https://github.com/ansible/ansible

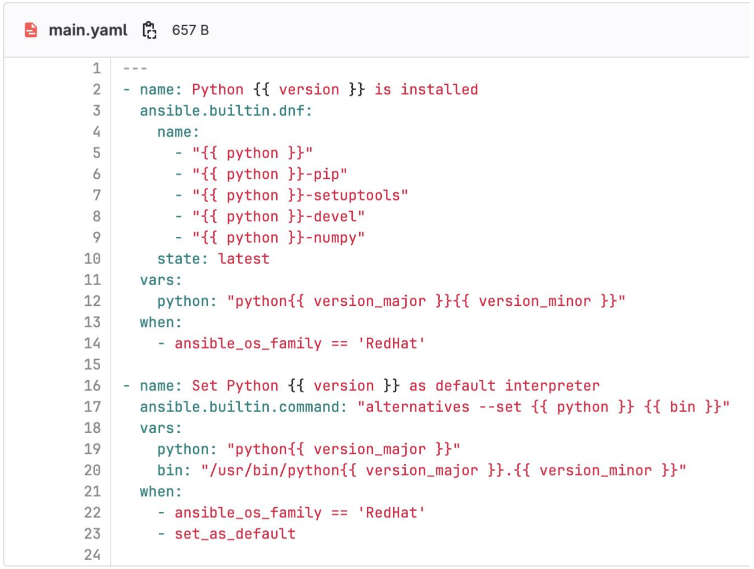
Example



```yaml
---
- name: Install Apache web server
  dnf:
    name:
      - httpd
    state: present
  tags: installation

- name: Start and enable httpd service
  ansible.builtin.service:
    name: httpd
    state: started
    enabled: true

```

Where is it used?

1. GitHub repositories using Docker/Vagrant to prepare the machine
2. Gitlab CI
3. App deployment
4. Cloud provisioning


main.yaml 657 B

```yaml
---
- name: Python {{ version }} is installed
  ansible.builtin.dnf:
    name:
      - "{{ python }}"
      - "{{ python }}-pip"
      - "{{ python }}-setuptools"
      - "{{ python }}-devel"
      - "{{ python }}-numpy"
    state: latest
  vars:
    python: "python{{ version_major }}{{ version_minor }}"
  when:
    - ansible_os_family == 'RedHat'

- name: Set Python {{ version }} as default interpreter
  ansible.builtin.command: "alternatives --set {{ python }} {{ bin }}"
  vars:
    python: "python{{ version_major }}"
    bin: "/usr/bin/python{{ version_major }}.{{ version_minor }}"
  when:
    - ansible_os_family == 'RedHat'
    - set_as_default

```

Alternatives

- Attune
  - https://attuneops.io
- Puppet
  - https://www.puppet.com
- Terraform
  - https://www.terraform.io
- Chef
  - https://www.chef.io

Part 2 - Installation

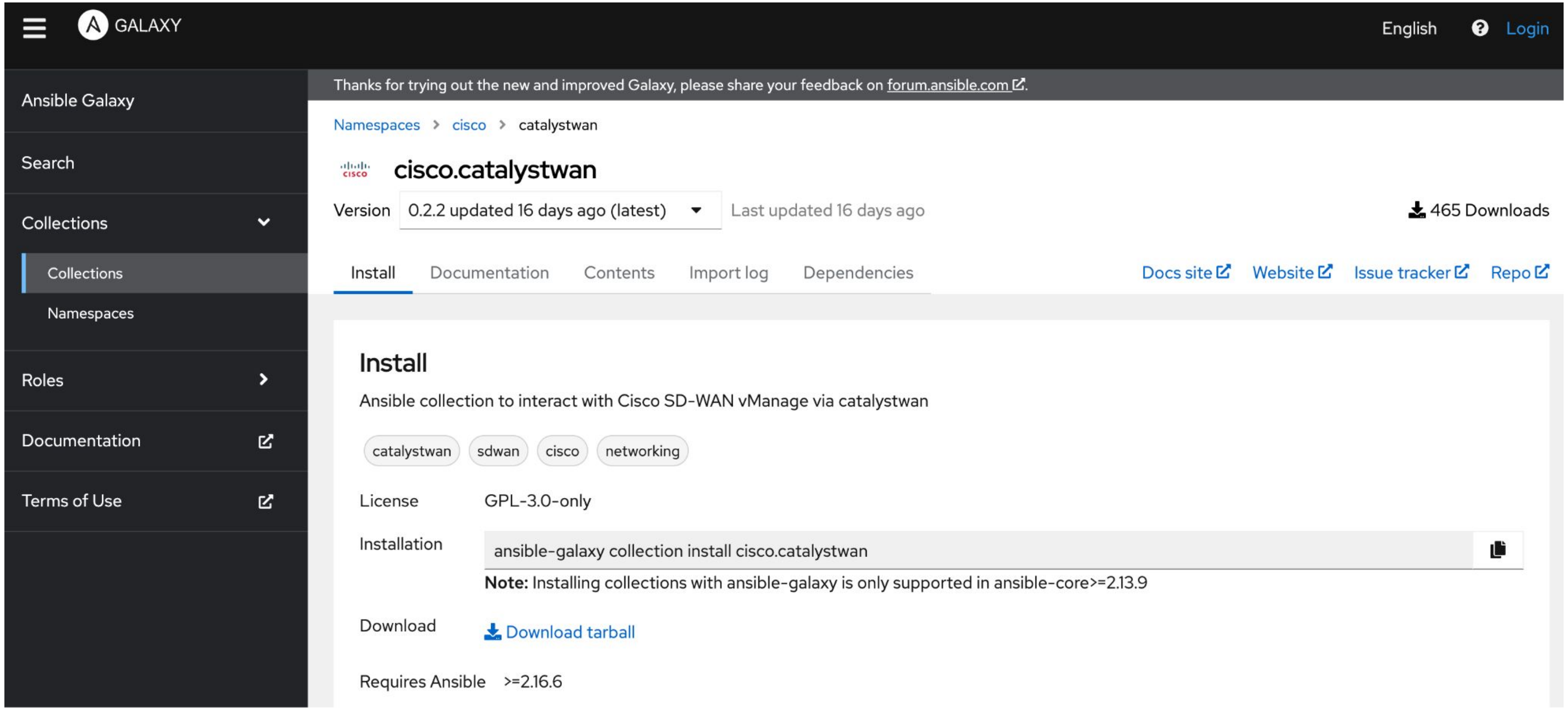
Installation

- Minimal installation
  ```
  pip3 install ansible-core
  ```
- Larger package with additional community-selected roles
  ```
  pip3 install --include-deps ansible
  ```
- Also possible with dnf/yum
  ```
  sudo {dnf,yum} install ansible
  ```
- Python3 on the managed machines
- https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Ansible Galaxy

- Ansible's "pip"
  `ansible-galaxy role install namespace.role_name`
- **`requirements.yml`**
- Database : https://galaxy.ansible.com/
- Docs: https://docs.ansible.com/ansible/latest/galaxy/user_guide.html

Ansible Galaxy



Thanks for trying out the new and improved Galaxy, please share your feedback on forum.ansible.com.

Namespaces > cisco > catalystwan

cisco.catalystwan

Version 0.2.2 updated 16 days ago (latest) Last updated 16 days ago 465 Downloads

Install

Ansible collection to interact with Cisco SD-WAN vManage via catalystwan

catalystwan sdwan cisco networking

License	GPL-3.0-only
Installation	`ansible-galaxy collection install cisco.catalystwan` **Note:** Installing collections with ansible-galaxy is only supported in ansible-core>=2.13.9
Download	Download tarball
Requires Ansible	>=2.16.6

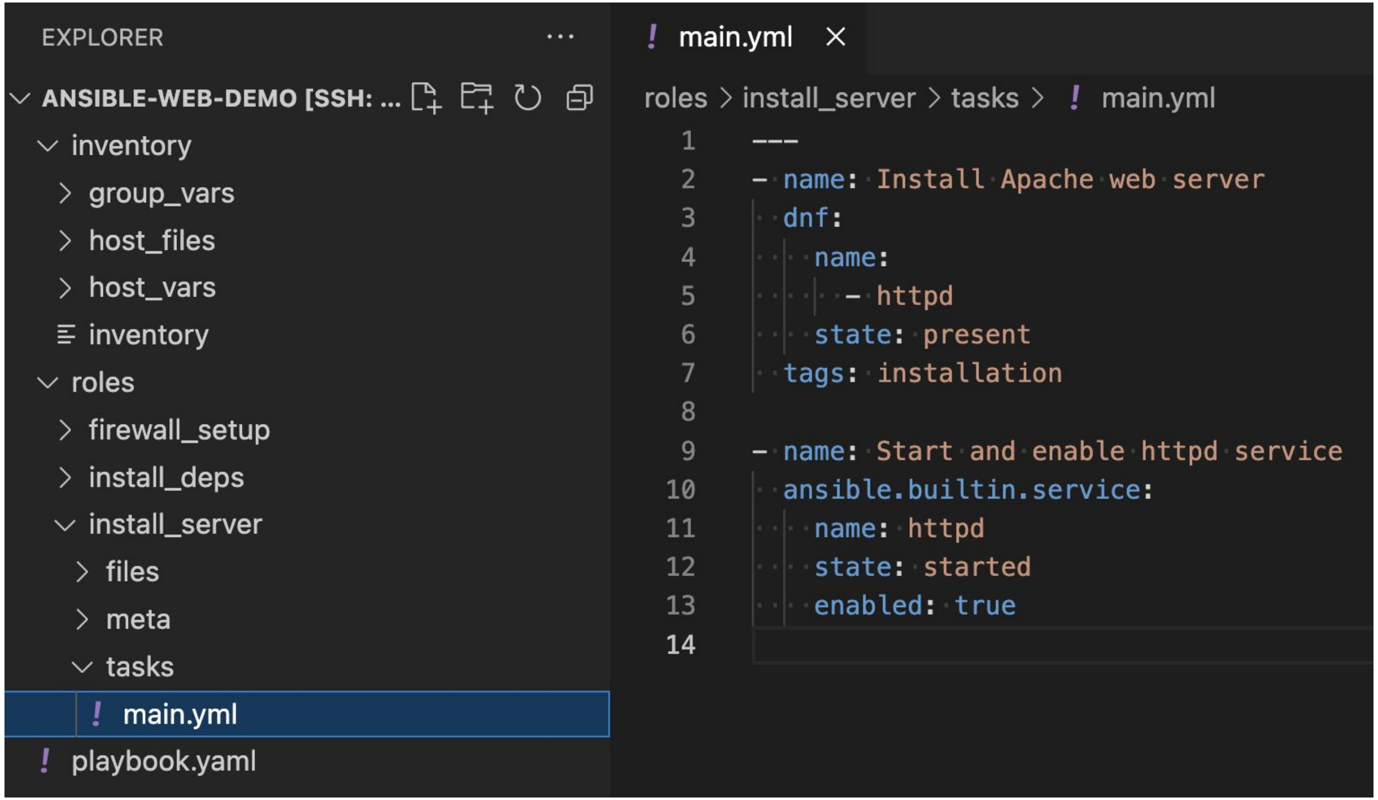
Part 3 - Ansible

Repository with sources

https://github.com/plnyrich/Ansible-Demo

```
vagrant up --provider=virtualbox
```

Structure

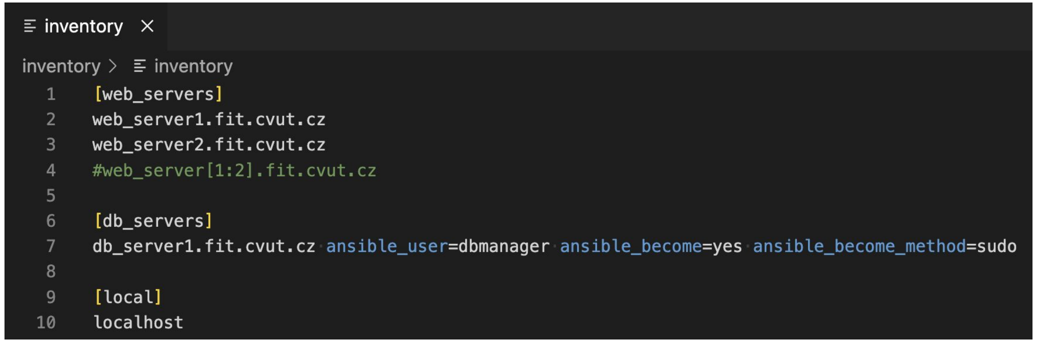

EXPLORER
ANSIBLE-WEB-DEMO [SSH: ...
main.yml
roles > install_server > tasks > main.yml

```
inventory
  group_vars
  host_files
  host_vars
  inventory
roles
  firewall_setup
  install_deps
  install_server
    files
    meta
    tasks
      main.yml
playbook.yaml
```

```yaml
---
- name: Install Apache web server
  dnf:
    name:
      - httpd
    state: present
  tags: installation

- name: Start and enable httpd service
  ansible.builtin.service:
    name: httpd
    state: started
    enabled: true
```

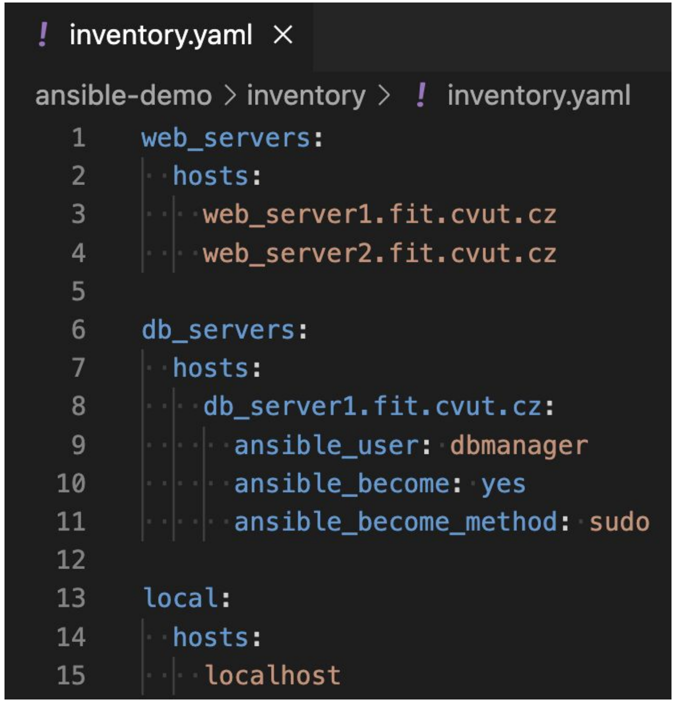
Inventory

- YAML or .ini format
- Define servers and their roles
  - Login credentials can be defined
    - `ansible_ssh_user`
    - `ansible_ssh_pass`
  - Ranges (inclusive)
    - db_server_[0:2].company.com =>
      db_server_0.company.com
      db_server_1.company.com
      db_server_2.company.com
- https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html



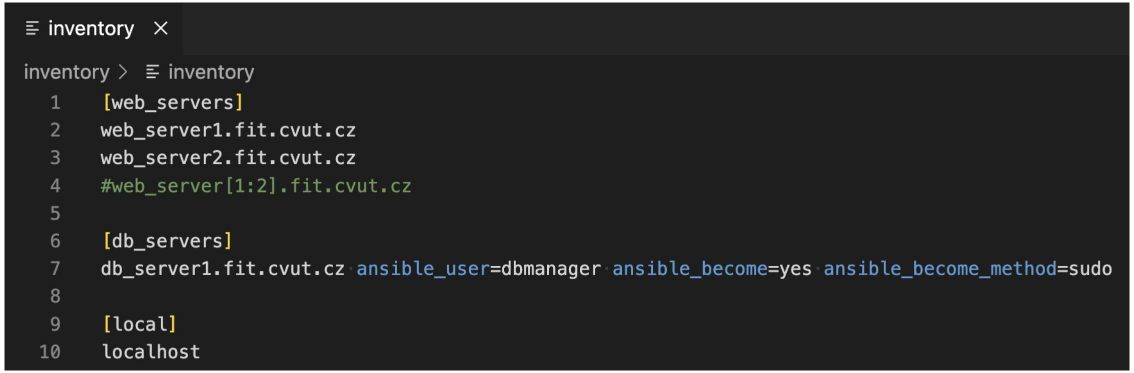
```
[web_servers]
web_server1.fit.cvut.cz
web_server2.fit.cvut.cz
#web_server[1:2].fit.cvut.cz

[db_servers]
db_server1.fit.cvut.cz ansible_user=dbmanager ansible_become=yes ansible_become_method=sudo

[local]
localhost
```

Inventory


inventory.yaml

ansible-demo > inventory > inventory.yaml

```yaml
web_servers:
  hosts:
    web_server1.fit.cvut.cz
    web_server2.fit.cvut.cz

db_servers:
  hosts:
    db_server1.fit.cvut.cz:
      ansible_user: dbmanager
      ansible_become: yes
      ansible_become_method: sudo

local:
  hosts:
    localhost
```


inventory

inventory > inventory

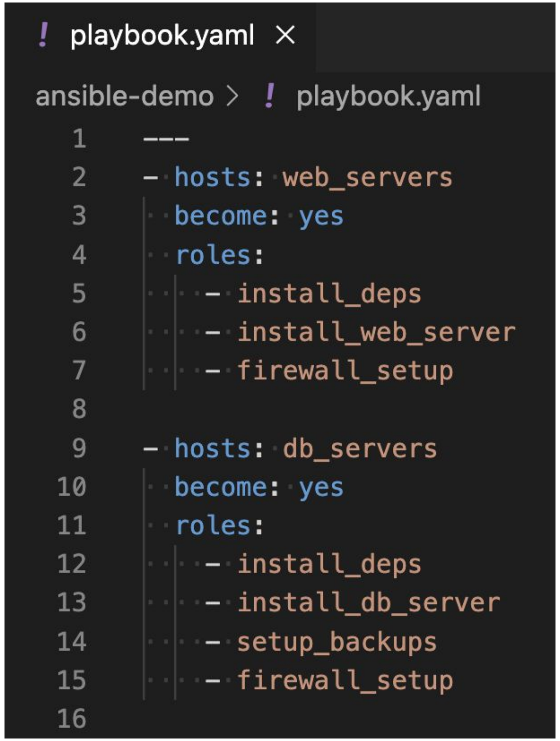
```ini
[web_servers]
web_server1.fit.cvut.cz
web_server2.fit.cvut.cz
#web_server[1:2].fit.cvut.cz

[db_servers]
db_server1.fit.cvut.cz ansible_user=dbmanager ansible_become=yes ansible_become_method=sudo

[local]
localhost
```

Playbooks

- Define what should be done on what machines
- “Roles” are executed in the defined order
- `--check` for syntax check without running the commands

```
ansible-playbook -i inventory playbook.yml
```

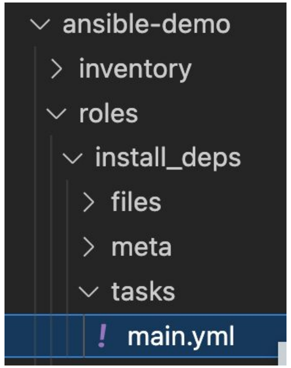
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html



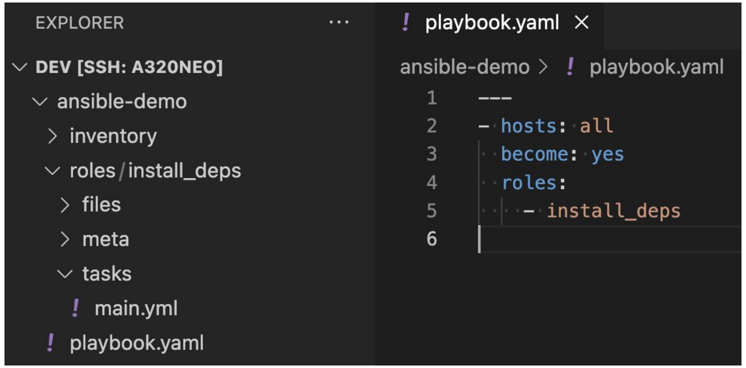
```yaml
---
- hosts: web_servers
  become: yes
  roles:
    - install_deps
    - install_web_server
    - firewall_setup

- hosts: db_servers
  become: yes
  roles:
    - install_deps
    - install_db_server
    - setup_backups
    - firewall_setup

```

Roles

- Playbooks composed from roles
- Must be in **`roles/`** directory
- Name of folder in roles/ can be used in playbook
- Role is set of tasks, task is one action



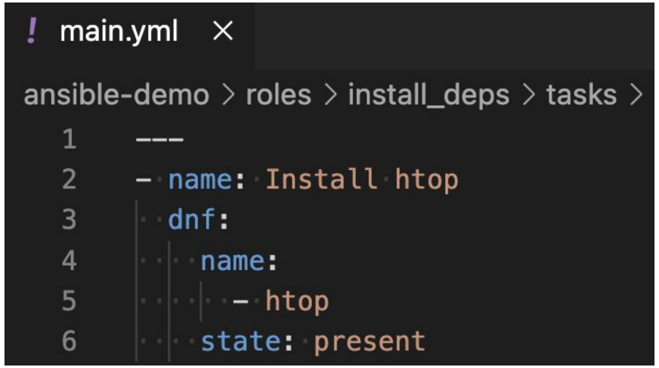
```
ansible-demo
  inventory
  roles
    install_deps
      files
      meta
      tasks
        main.yml
```



```
EXPLORER
DEV [SSH: A320NEO]
  ansible-demo
    inventory
    roles / install_deps
      files
      meta
      tasks
        main.yml
    playbook.yaml
```

playbook.yaml — ansible-demo > playbook.yaml

```yaml
---
- hosts: all
  become: yes
  roles:
    - install_deps
```

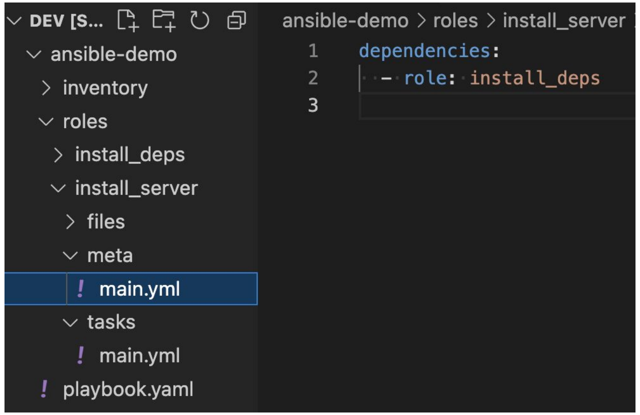


main.yml — ansible-demo > roles > install_deps > tasks >

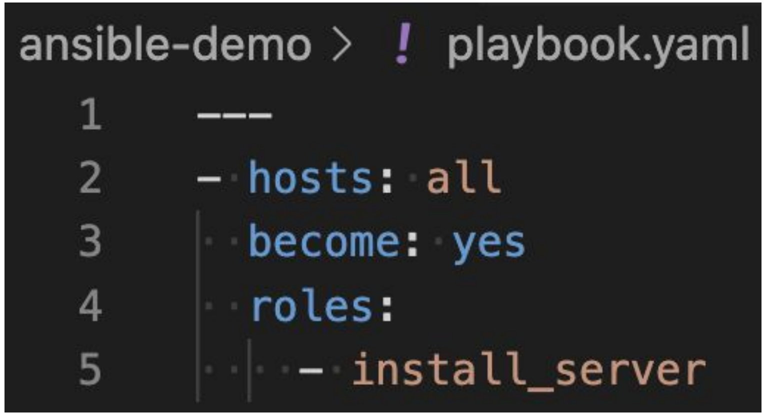
```yaml
---
- name: Install htop
  dnf:
    name:
      - htop
    state: present
```

Role Dependencies

- We do not want to install web server before we install dependencies
  => **meta/main.yml**
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html

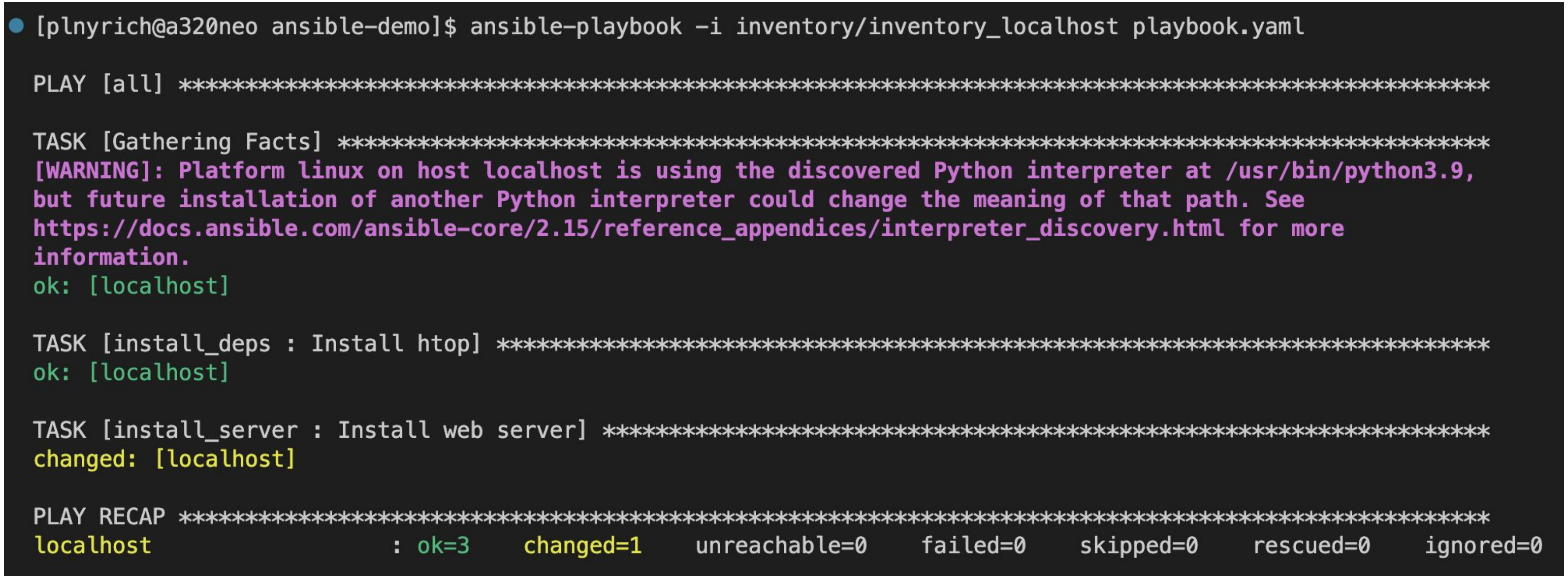


```
dependencies:
  - role: install_deps
```



```
---
- hosts: all
  become: yes
  roles:
    - install_server
```

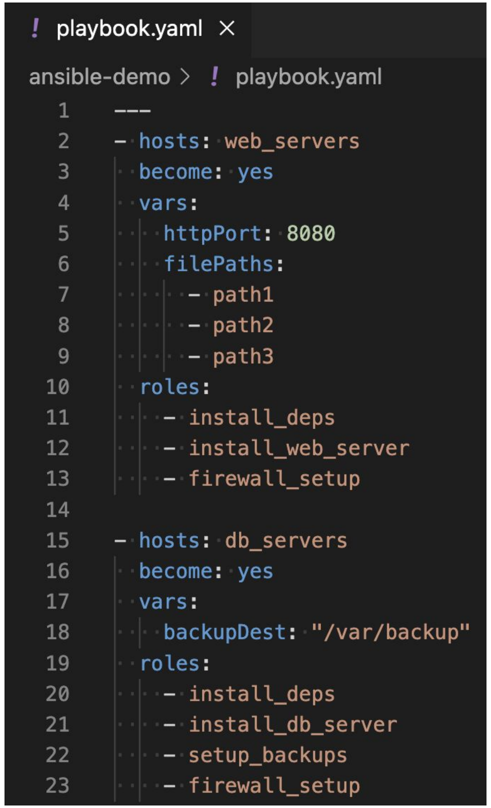
Role Dependencies



```
[plnyrich@a320neo ansible-demo]$ ansible-playbook -i inventory/inventory_localhost playbook.yaml

PLAY [all] ***************************************************************************************************

TASK [Gathering Facts] ***************************************************************************************
[WARNING]: Platform linux on host localhost 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.
ok: [localhost]

TASK [install_deps : Install htop] ***************************************************************************
ok: [localhost]

TASK [install_server : Install web server] *******************************************************************
changed: [localhost]

PLAY RECAP ***************************************************************************************************
localhost                  : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

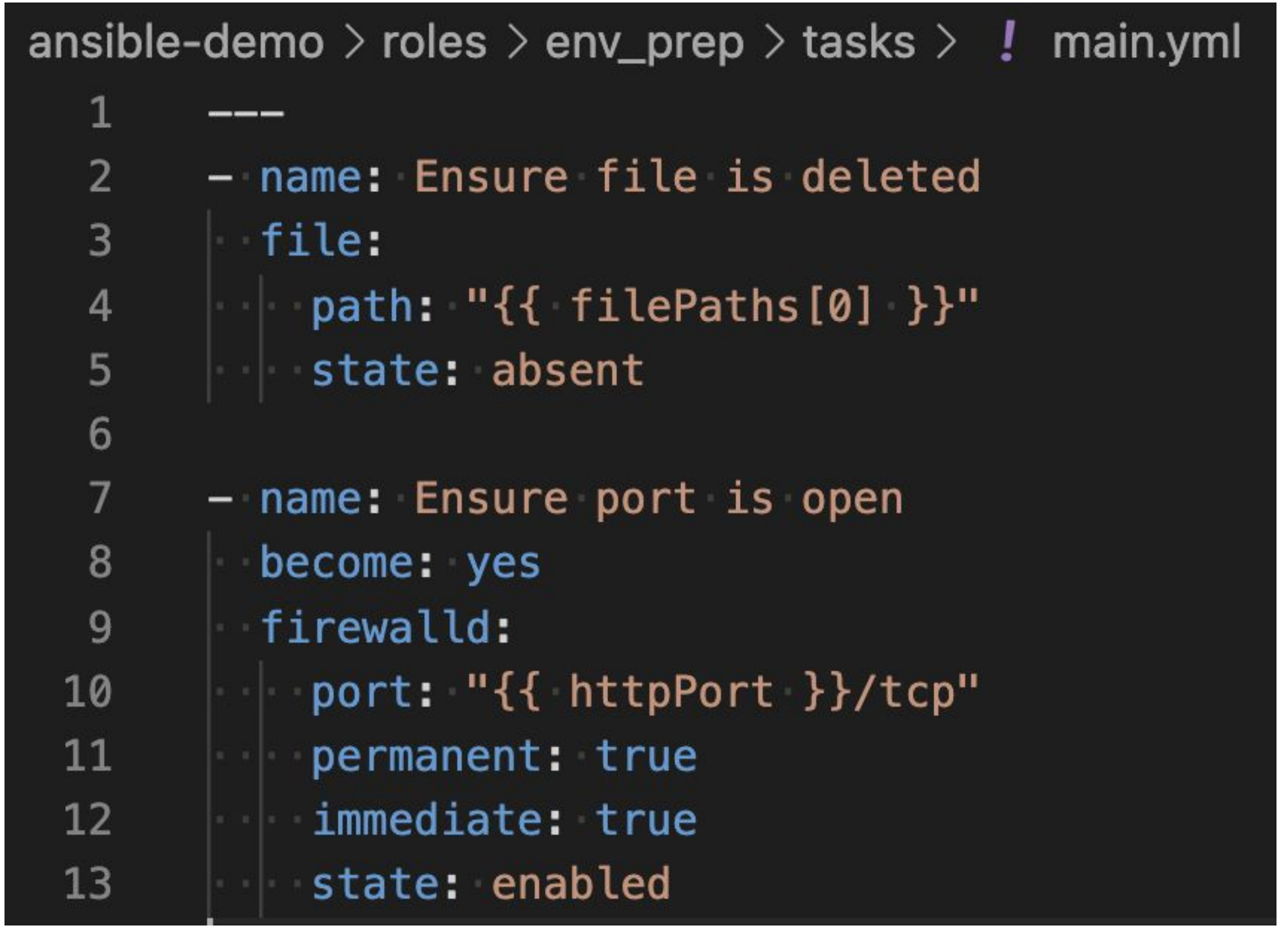
Variables

- Sometimes we need to customize what should be done
  => **variables**
- Use variable in role: **`{{ varName }}`**
- Some vars are prepared by Ansible
  - **`ansible_facts`**
  - **`inventory_dir`**
  - …
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_variables.html
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_vars_facts.html



```yaml
---
- hosts: web_servers
  become: yes
  vars:
    httpPort: 8080
    filePaths:
      - path1
      - path2
      - path3
  roles:
    - install_deps
    - install_web_server
    - firewall_setup

- hosts: db_servers
  become: yes
  vars:
    backupDest: "/var/backup"
  roles:
    - install_deps
    - install_db_server
    - setup_backups
    - firewall_setup
```

Variables

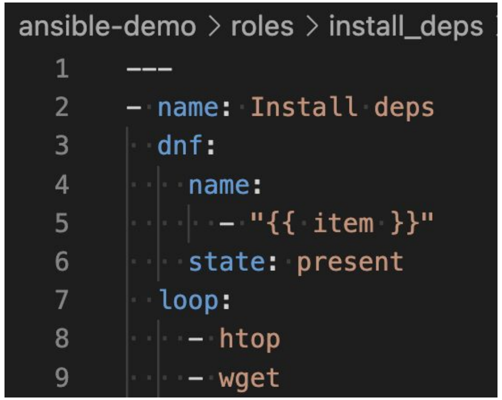


ansible-demo > roles > env_prep > tasks > ! main.yml

```yaml
---
- name: Ensure file is deleted
  file:
    path: "{{ filePaths[0] }}"
    state: absent

- name: Ensure port is open
  become: yes
  firewalld:
    port: "{{ httpPort }}/tcp"
    permanent: true
    immediate: true
    state: enabled
```

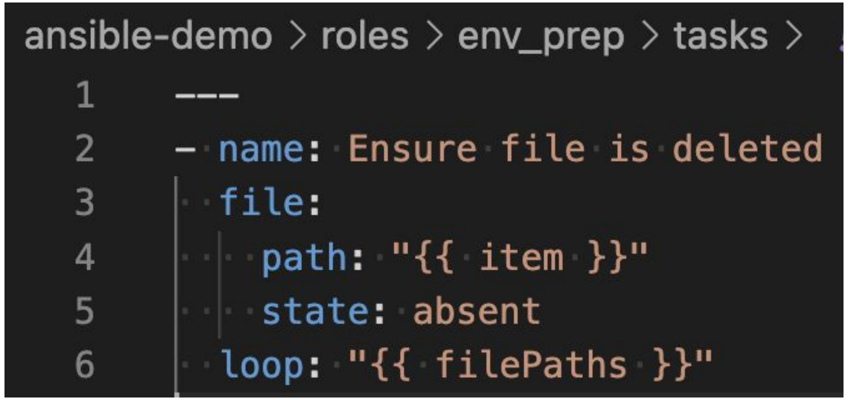
Loops

- We want to install N packages, delete X files, …
- `loop`, `with_items`
  - `with_items`: single-level flattening
  - `loop: "{{ … | flatten(1) }}"` is equivalent to `with_items`
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_loops.html



ansible-demo > roles > install_deps

```yaml
---
- name: Install deps
  dnf:
    name:
      - "{{ item }}"
    state: present
  loop:
    - htop
    - wget
```

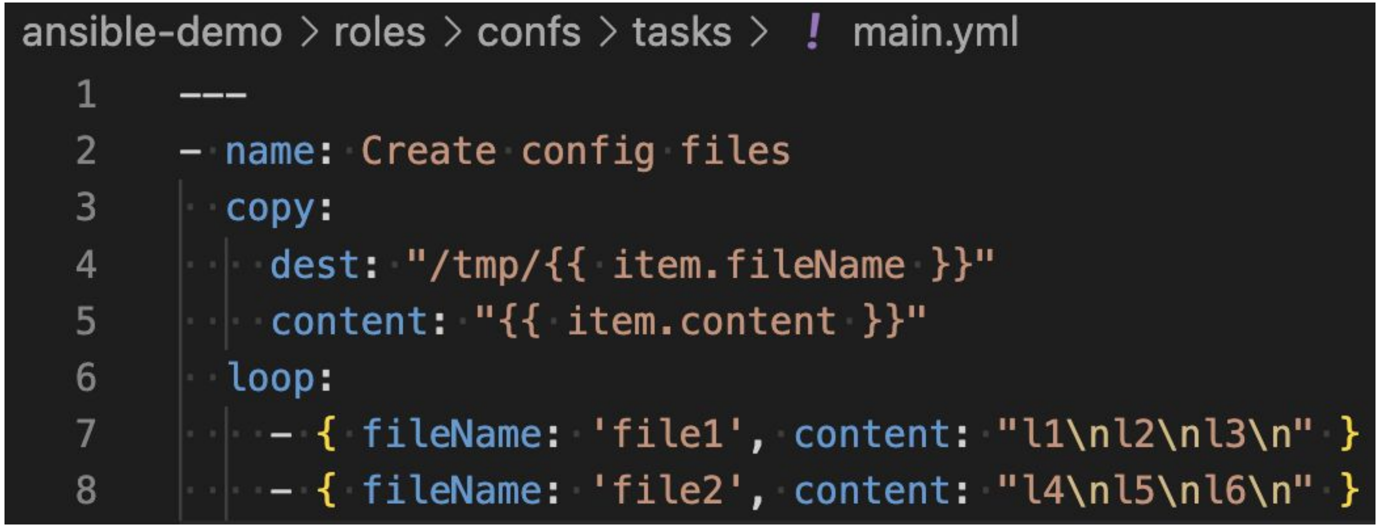


ansible-demo > roles > env_prep > tasks >

```yaml
---
- name: Ensure file is deleted
  file:
    path: "{{ item }}"
    state: absent
  loop: "{{ filePaths }}"
```

Loops

- We can also define more complex items or “objects”

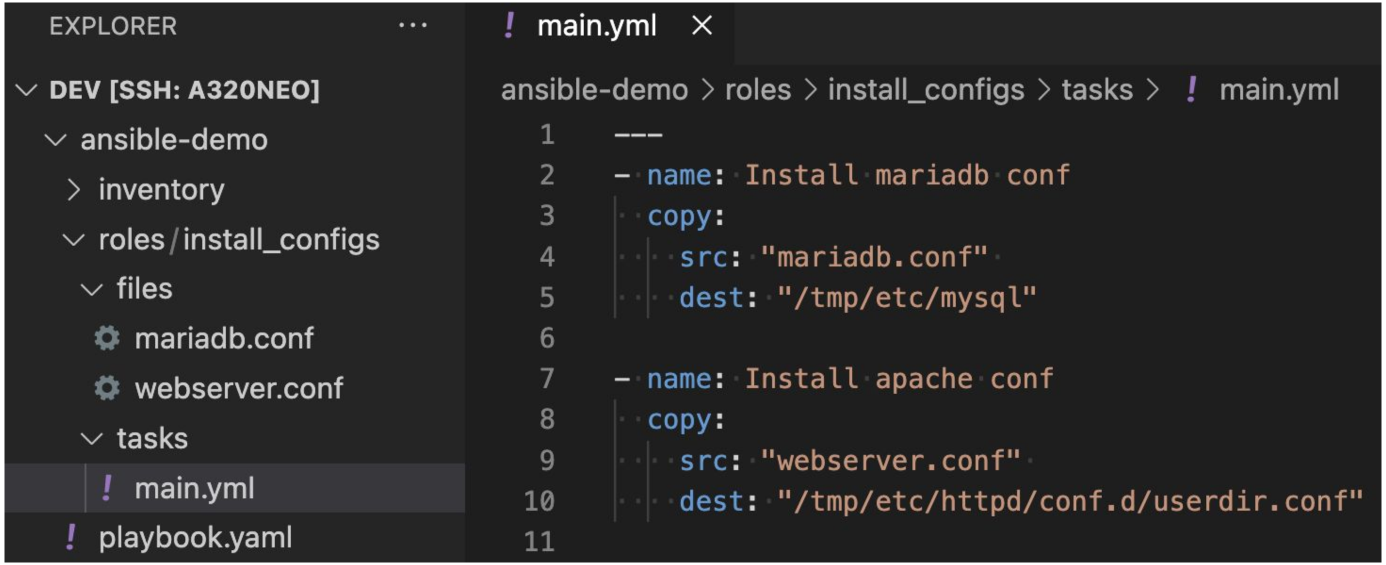


ansible-demo > roles > confs > tasks > ! main.yml

```yaml
---
- name: Create config files
  copy:
    dest: "/tmp/{{ item.fileName }}"
    content: "{{ item.content }}"
  loop:
    - { fileName: 'file1', content: "l1\nl2\nl3\n" }
    - { fileName: 'file2', content: "l4\nl5\nl6\n" }
```

Files

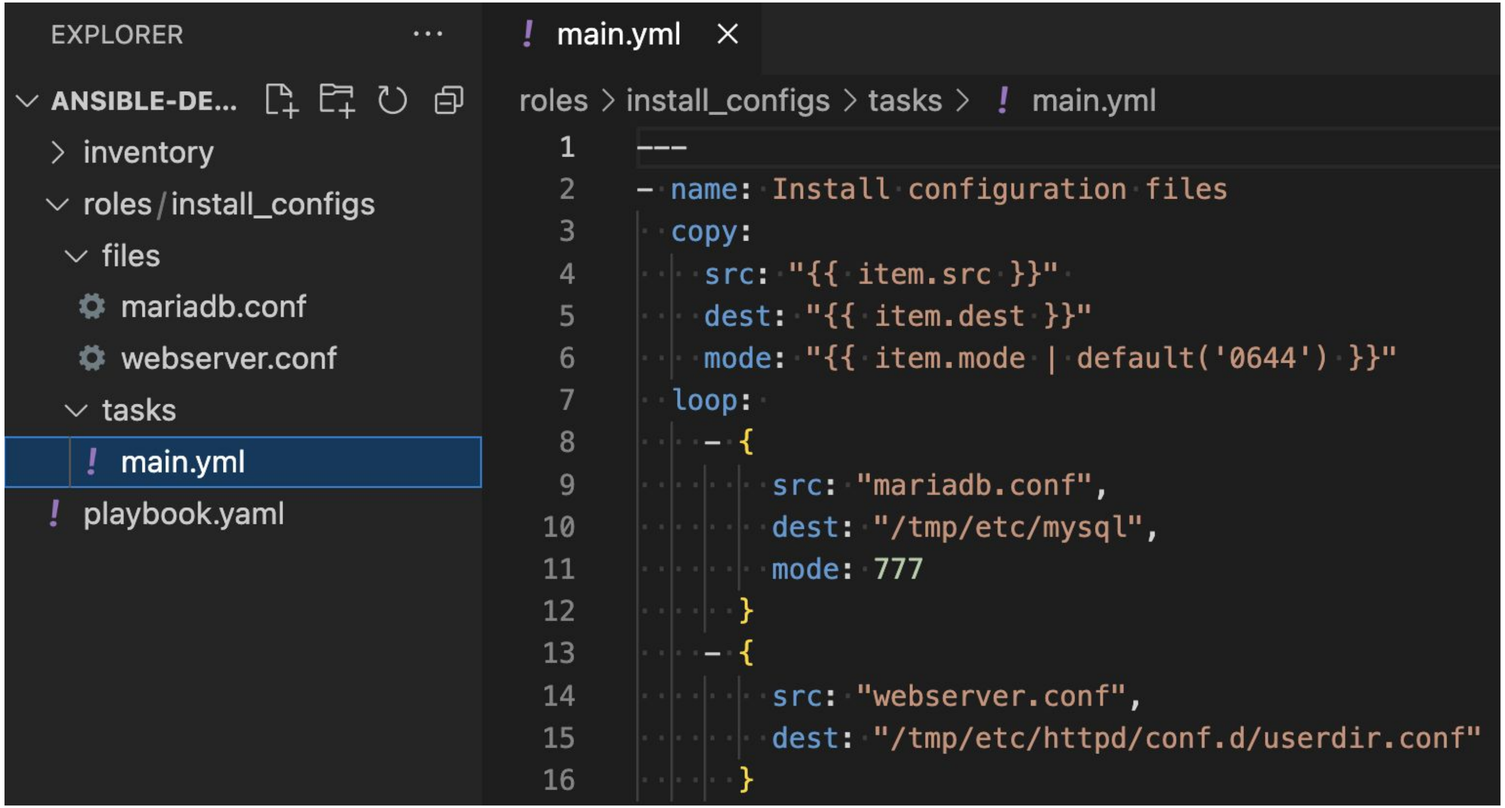
- Prepare configuration files, …
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html



```
ansible-demo
  inventory
  roles/install_configs
    files
      mariadb.conf
      webserver.conf
    tasks
      main.yml
  playbook.yaml
```

```yaml
---
- name: Install mariadb conf
  copy:
    src: "mariadb.conf"
    dest: "/tmp/etc/mysql"

- name: Install apache conf
  copy:
    src: "webserver.conf"
    dest: "/tmp/etc/httpd/conf.d/userdir.conf"
```

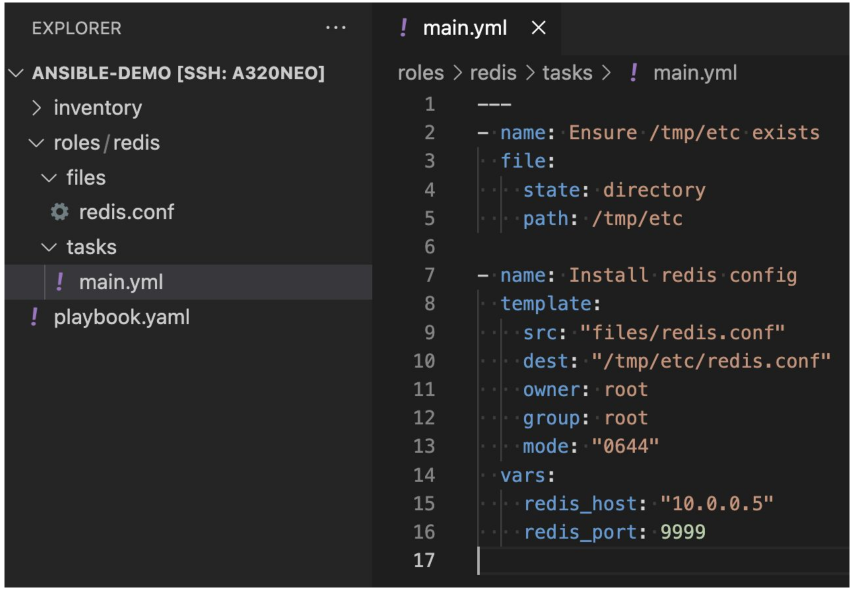
Files



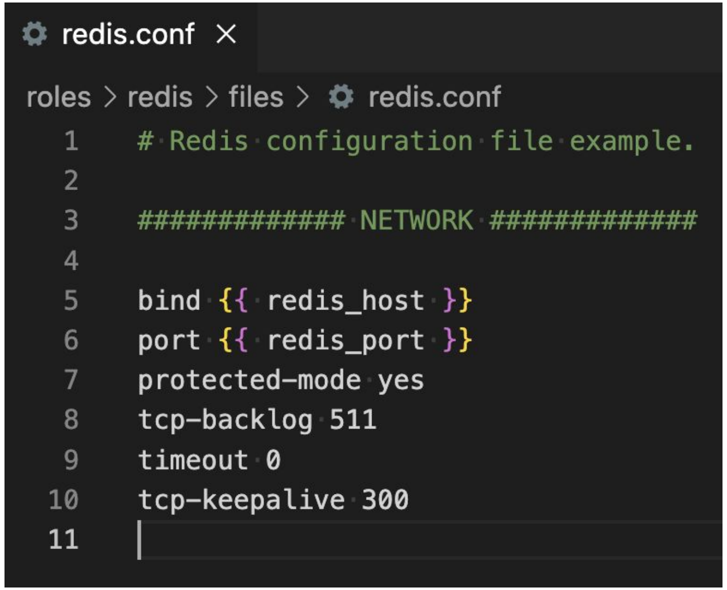
```yaml
---
- name: Install configuration files
  copy:
    src: "{{ item.src }}"
    dest: "{{ item.dest }}"
    mode: "{{ item.mode | default('0644') }}"
  loop:
    - {
        src: "mariadb.conf",
        dest: "/tmp/etc/mysql",
        mode: 777
      }
    - {
        src: "webserver.conf",
        dest: "/tmp/etc/httpd/conf.d/userdir.conf"
      }
```

Templates

- Use vars in the prepared files in **files/**
- https://docs.ansible.com/ansible/latest/collections/ansible/builtin/template_module.html

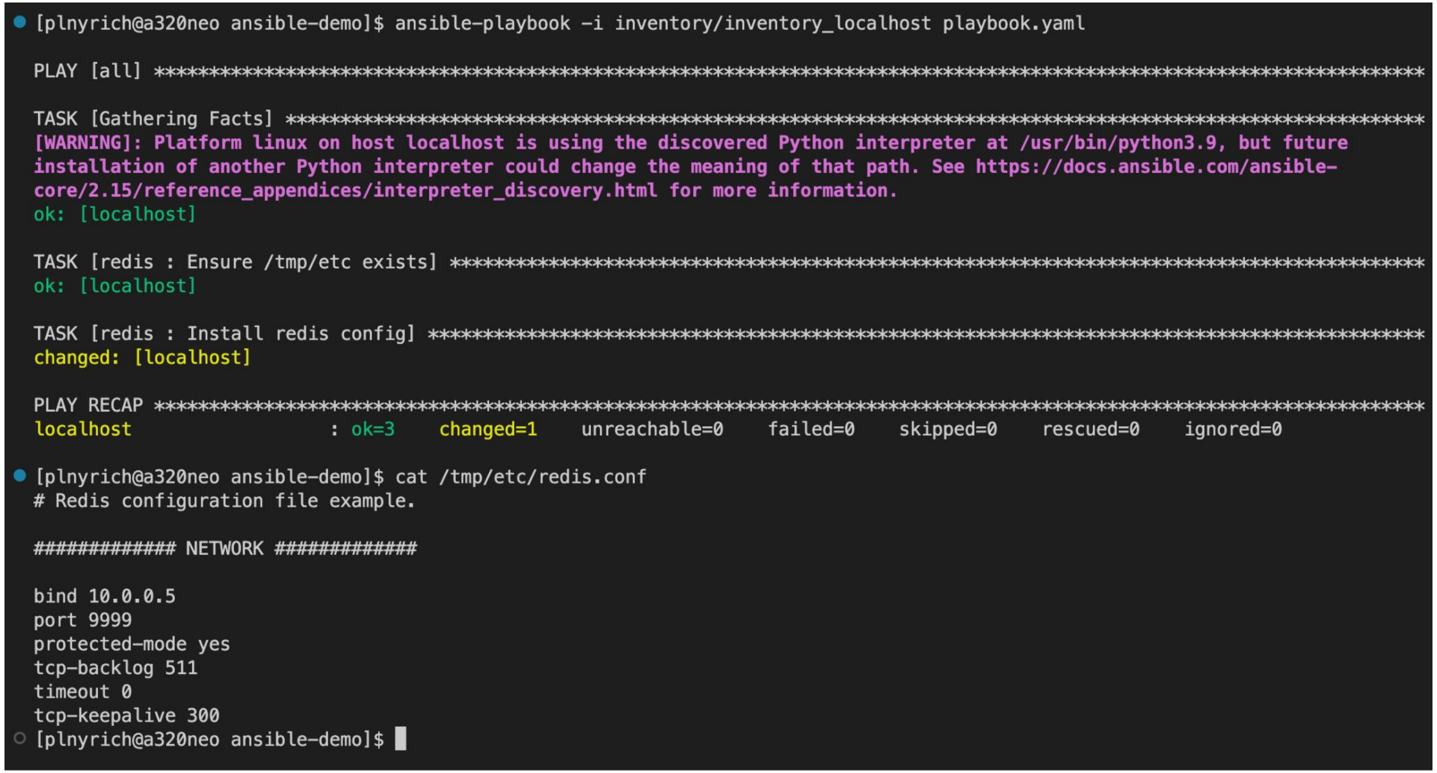


```yaml
---
- name: Ensure /tmp/etc exists
  file:
    state: directory
    path: /tmp/etc

- name: Install redis config
  template:
    src: "files/redis.conf"
    dest: "/tmp/etc/redis.conf"
    owner: root
    group: root
    mode: "0644"
  vars:
    redis_host: "10.0.0.5"
    redis_port: 9999
```



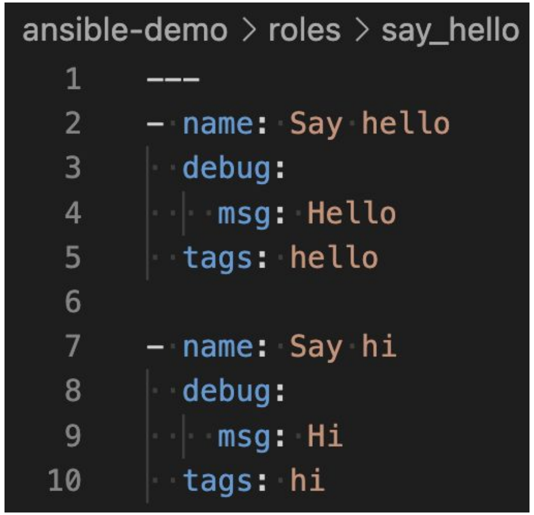
```
# Redis configuration file example.

############# NETWORK ############

bind {{ redis_host }}
port {{ redis_port }}
protected-mode yes
tcp-backlog 511
timeout 0
tcp-keepalive 300
```

Templates



```
● [plnyrich@a320neo ansible-demo]$ ansible-playbook -i inventory/inventory_localhost playbook.yaml

  PLAY [all] ******************************************************************************************************

  TASK [Gathering Facts] ******************************************************************************************
  [WARNING]: Platform linux on host localhost 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.
  ok: [localhost]

  TASK [redis : Ensure /tmp/etc exists] ***************************************************************************
  ok: [localhost]

  TASK [redis : Install redis config] *****************************************************************************
  changed: [localhost]

  PLAY RECAP ******************************************************************************************************
  localhost                  : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

● [plnyrich@a320neo ansible-demo]$ cat /tmp/etc/redis.conf
  # Redis configuration file example.

  ############# NETWORK #############

  bind 10.0.0.5
  port 9999
  protected-mode yes
  tcp-backlog 511
  timeout 0
  tcp-keepalive 300
○ [plnyrich@a320neo ansible-demo]$ █
```

Tags


ansible-demo > roles > say_hello

```
---
- name: Say hello
  debug:
    msg: Hello
  tags: hello

- name: Say hi
  debug:
    msg: Hi
  tags: hi
```

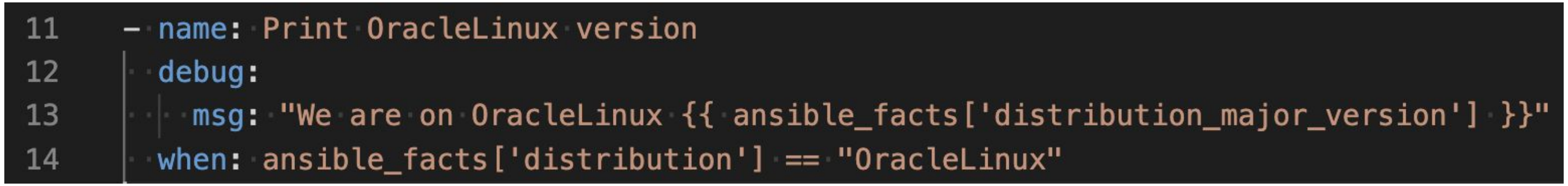
- Mark roles with tags and run only some of them
- Example: installation, configuration, ...
- runs only roles marked with **`hello`** tag:

  `ansible-playbook -i inventory/inventory_localhost playbook.yml --tags ‘hello’`
- Runs everything except **`hi`** tag

  `ansible-playbook -i inventory/inventory_localhost playbook.yml --skip-tags ‘hi’`
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_tags.html

Conditional Execution

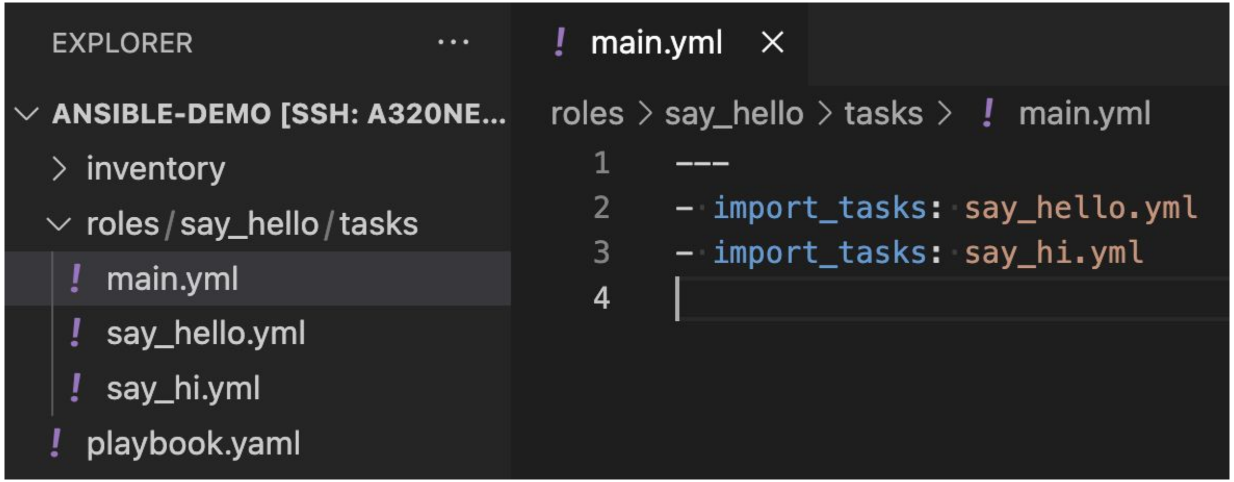
- **`when`** keyword
- https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_conditionals.html
- Run tasks only when a condition is satisfied



```yaml
- name: Print OracleLinux version
  debug:
    msg: "We are on OracleLinux {{ ansible_facts['distribution_major_version'] }}"
  when: ansible_facts['distribution'] == "OracleLinux"
```

Import Tasks

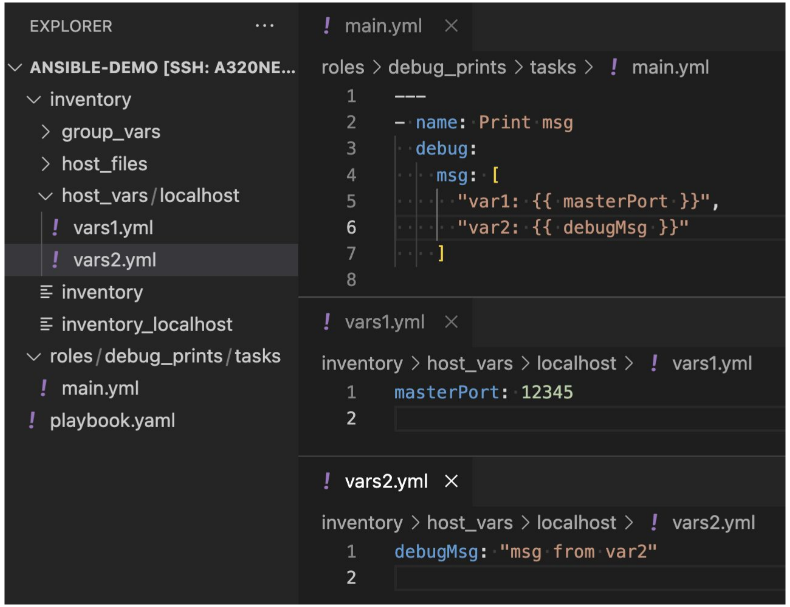
- Split roles into multiple files to keep the codebase clean
- Roles are imported and executed in the defined order
- https://docs.ansible.com/ansible/latest/collections/ansible/builtin/import_tasks_module.html


EXPLORER
ANSIBLE-DEMO [SSH: A320NE...
inventory
roles / say_hello / tasks
main.yml
say_hello.yml
say_hi.yml
playbook.yaml
main.yml
roles > say_hello > tasks > main.yml

```yaml
---
- import_tasks: say_hello.yml
- import_tasks: say_hi.yml
```

Host Vars

- Define variables for each host in a specific file
  `<prevPath>/inventory/host_vars/<host>/*.yml`



main.yml — roles > debug_prints > tasks > main.yml

```
---
- name: Print msg
  debug:
    msg: [
      "var1: {{ masterPort }}",
      "var2: {{ debugMsg }}"
    ]

```

vars1.yml — inventory > host_vars > localhost > vars1.yml

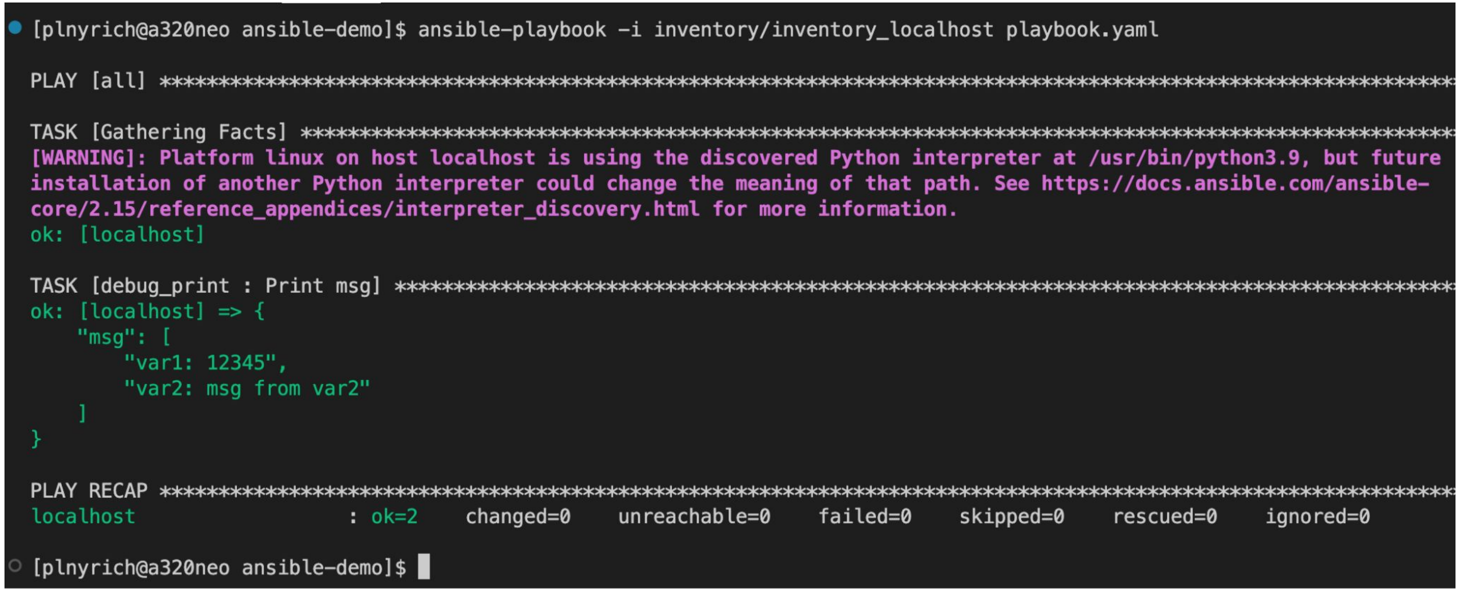
```
masterPort: 12345

```

vars2.yml — inventory > host_vars > localhost > vars2.yml

```
debugMsg: "msg from var2"

```

Host Vars

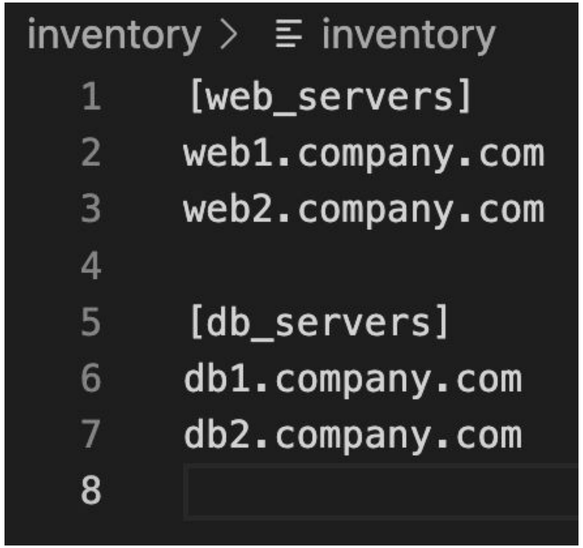
- https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#host-variables



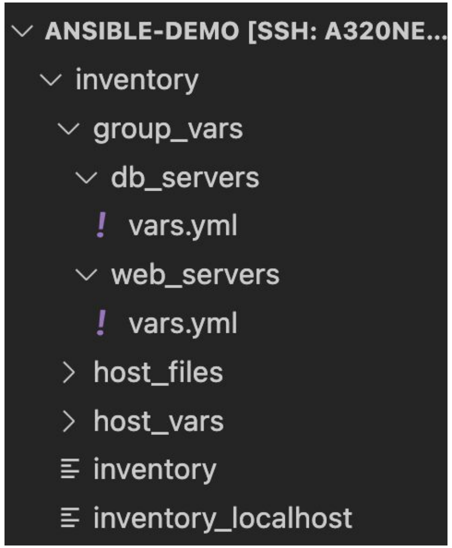
```
[plnyrich@a320neo ansible-demo]$ ansible-playbook -i inventory/inventory_localhost playbook.yaml

PLAY [all] ******************************************************************************************************

TASK [Gathering Facts] ******************************************************************************************
[WARNING]: Platform linux on host localhost 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.
ok: [localhost]

TASK [debug_print : Print msg] **********************************************************************************
ok: [localhost] => {
    "msg": [
        "var1: 12345",
        "var2: msg from var2"
    ]
}

PLAY RECAP ******************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[plnyrich@a320neo ansible-demo]$
```

Group Vars

- Similar as host vars but we can define variables for groups
  `<prevPath>/inventory/group_vars/<group>/*.yml`



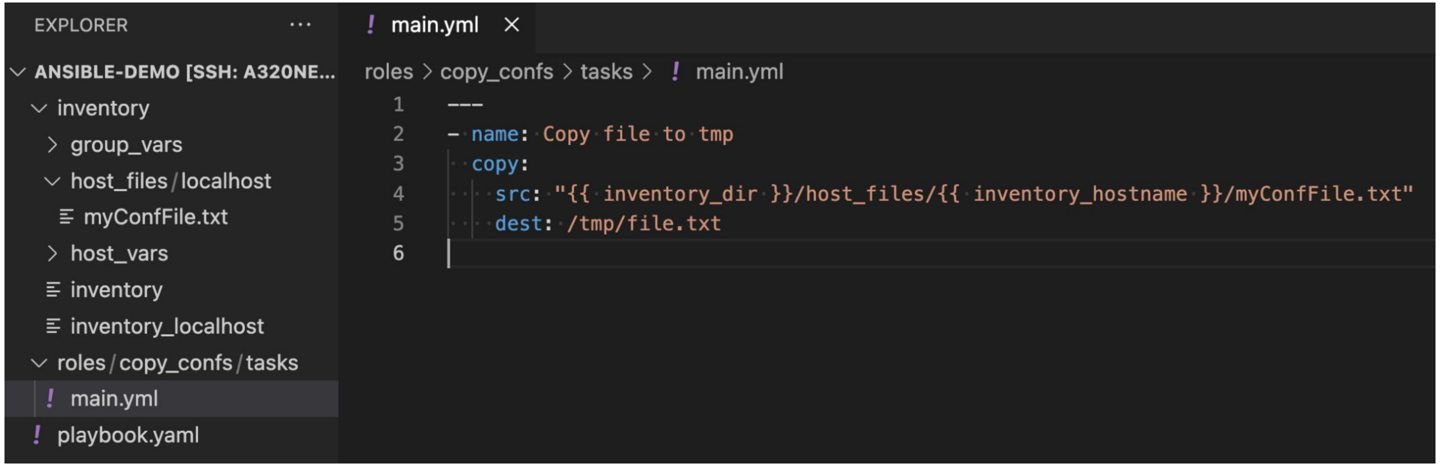
```
[web_servers]
web1.company.com
web2.company.com

[db_servers]
db1.company.com
db2.company.com

```



```
ANSIBLE-DEMO [SSH: A320NE...
  inventory
    group_vars
      db_servers
        vars.yml
      web_servers
        vars.yml
    host_files
    host_vars
    inventory
    inventory_localhost
```

Host Files

- Not directly in Ansible, but handy
  `<prevPath>/inventory/host_files/<host>/*.yml`
- Reference file from role:
  `{{ inventory_dir }}/host_files/{{ inventory_hostname }}/<file>`



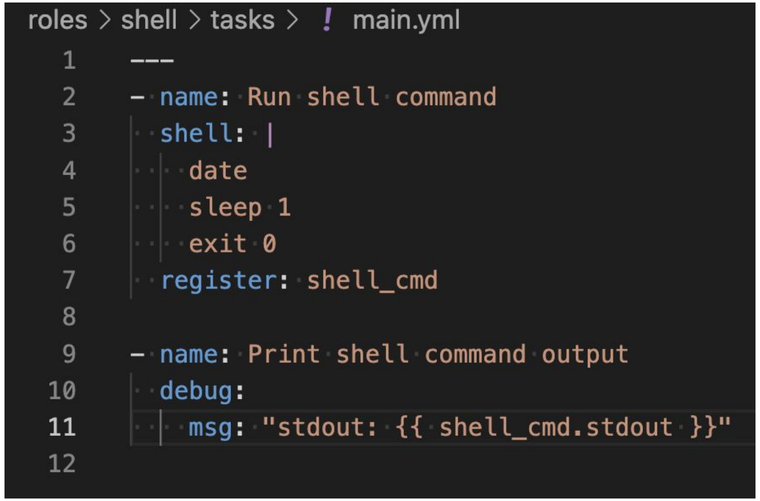
```yaml
---
- name: Copy file to tmp
  copy:
    src: "{{ inventory_dir }}/host_files/{{ inventory_hostname }}/myConfFile.txt"
    dest: /tmp/file.txt
```

Vaults

- Storing sensitive information
- Encryption of variables, roles, whole files, …
- https://docs.ansible.com/ansible/latest/vault_guide/vault.html
- https://www.digitalocean.com/community/tutorials/how-to-use-vault-to-protect-sensitive-ansible-data

Running shell

- Use **`shell`** module
- Use pipe (|) for multiline string input
- https://docs.ansible.com/ansible/latest/collections/ansible/builtin/shell_module.html



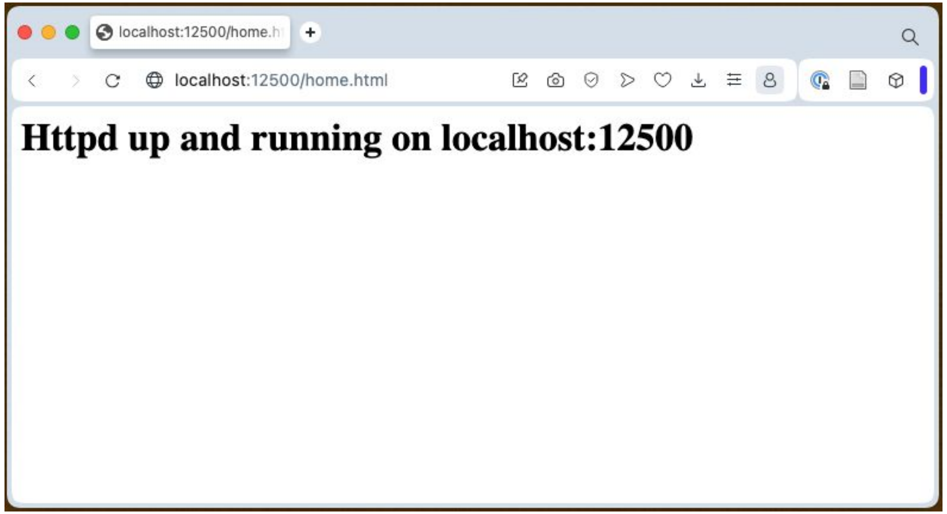
```yaml
---
- name: Run shell command
  shell: |
    date
    sleep 1
    exit 0
  register: shell_cmd

- name: Print shell command output
  debug:
    msg: "stdout: {{ shell_cmd.stdout }}"

```

Part 5 - Demo

What do we want to do?

- Quickly deploy web server with our website
    - Install dependencies
    - Install web server and configure web server
    - Open port in firewall
    - Start the web server
    - Access our website

Result



Httpd up and running on localhost:12500