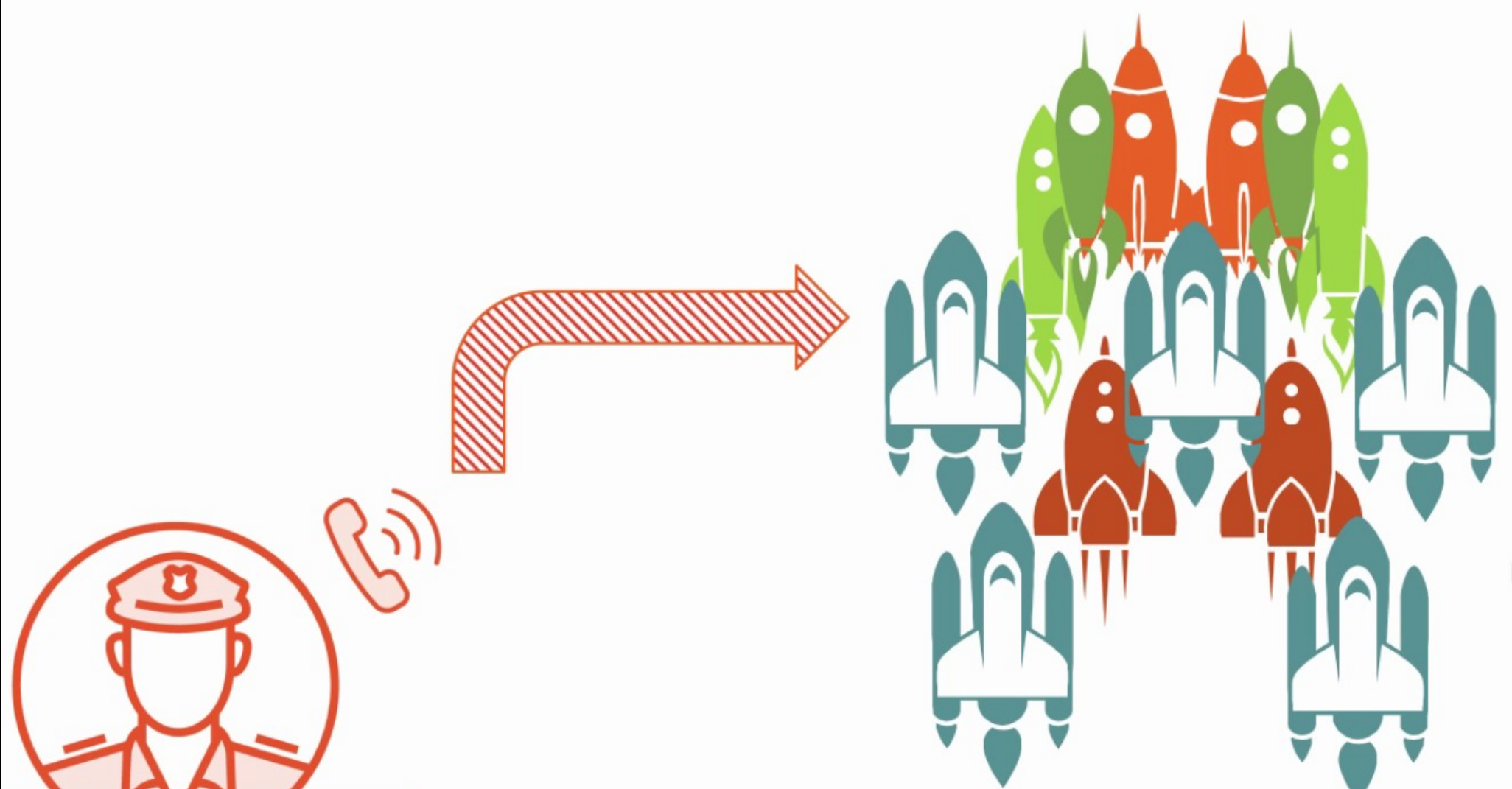
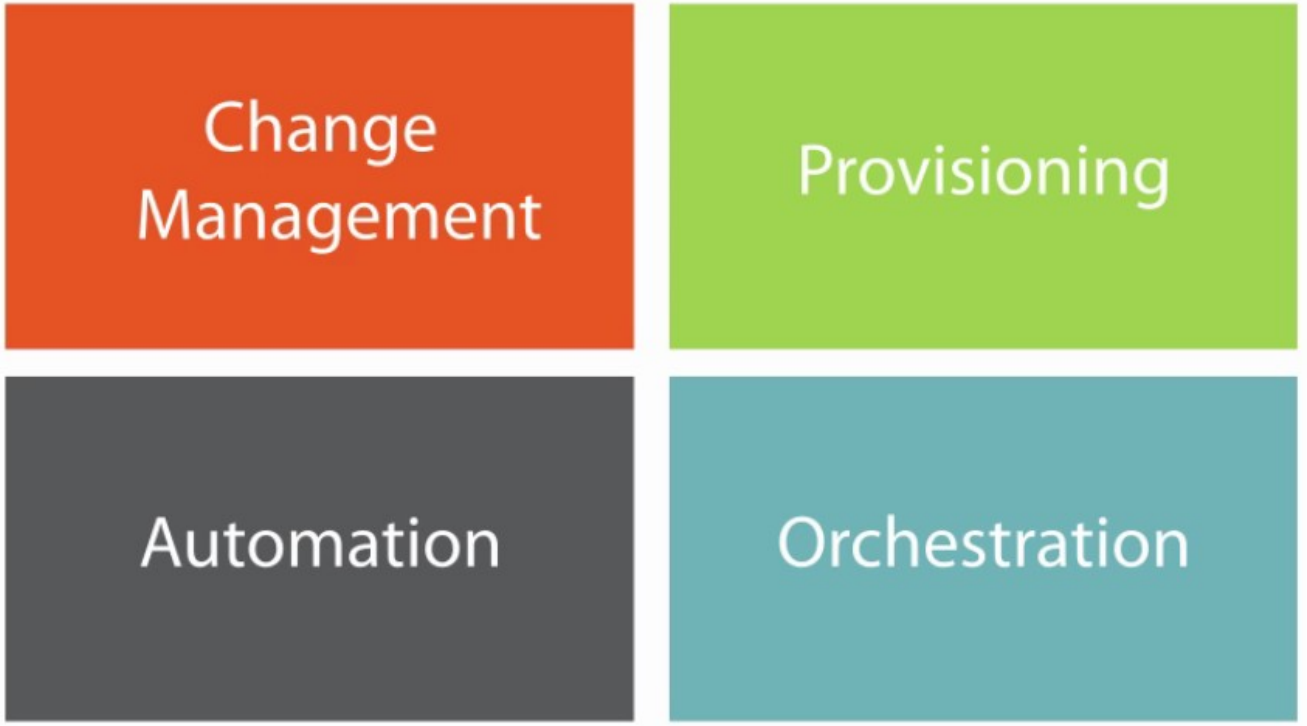
What Is Ansible?



What Is Ansible?



Change Management

Define a "System State"

Enforce the System State

System State

Apache Web Installed

Apache Web at version x.xx.x

Apache Web Started

A function is idempotent if repeated applications has the same affect as a single application

IDEMPOTENCE

Provisioning

Prepare a system to make it ready

Transition from one state to a different state

Examples

- Make an FTP Server
- Make an Email Server
- Make a DB Server

1. Install web software
2. Copy configurations
3. Copy web files
4. Install security updates
5. Start web service

Automation

Define tasks to be executed automatically

- Ordered Tasks
- Make decisions
- Ad-hoc tasks

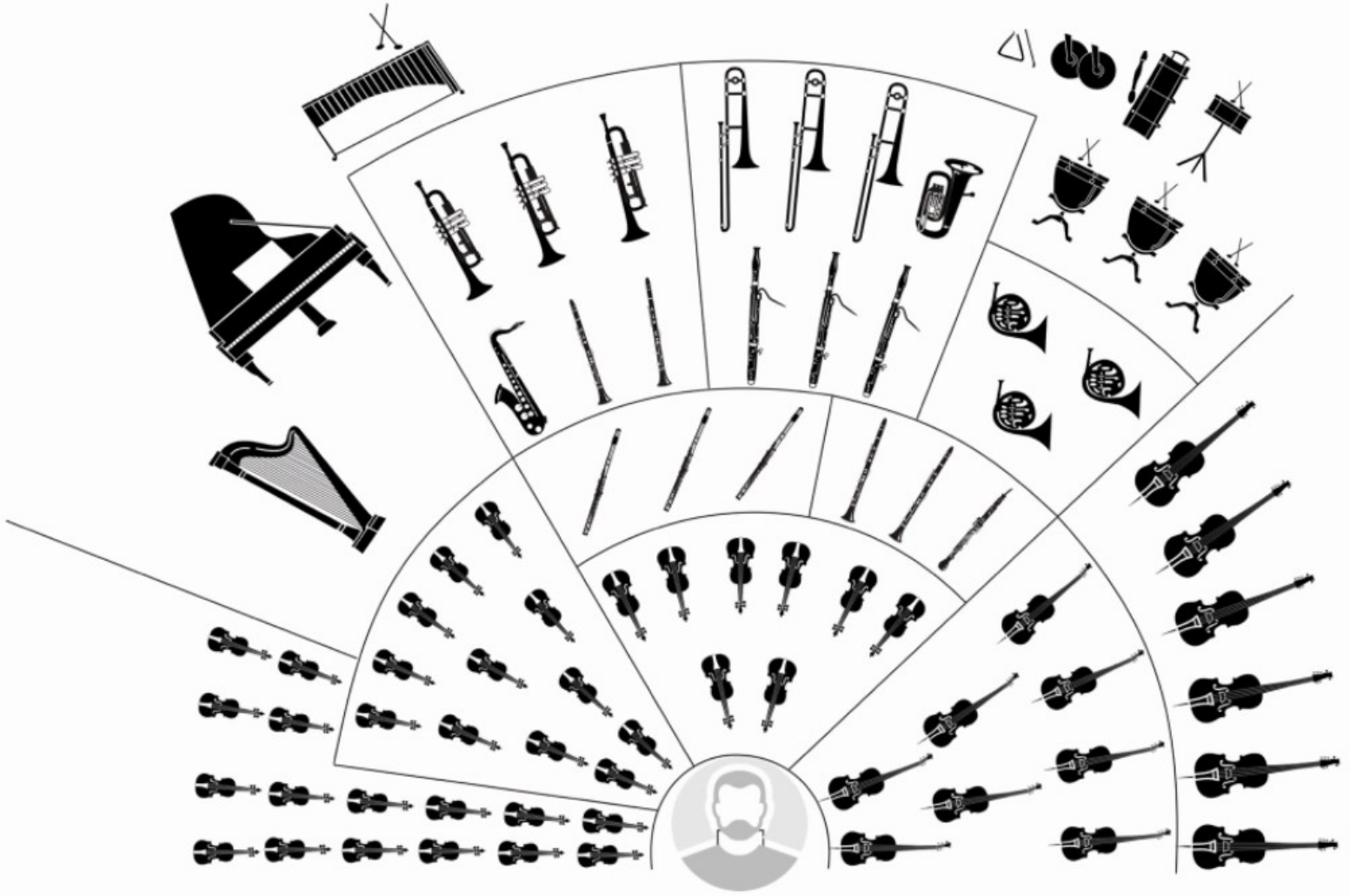
Set it and Forget it

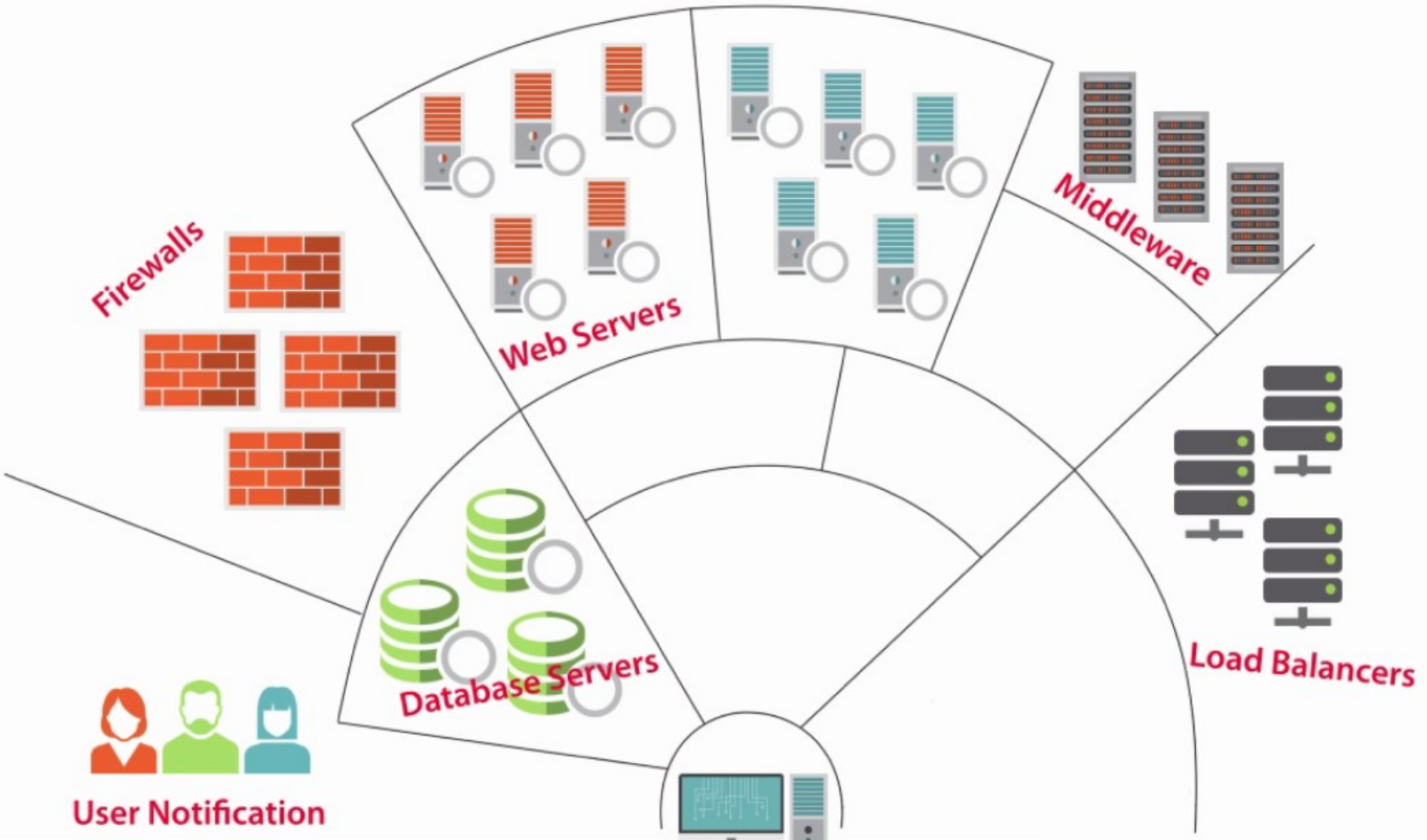
- Run the task
- Get a cup of coffee
- Walk back to desk seeing tasks finished
- Sip your coffee and feel productive

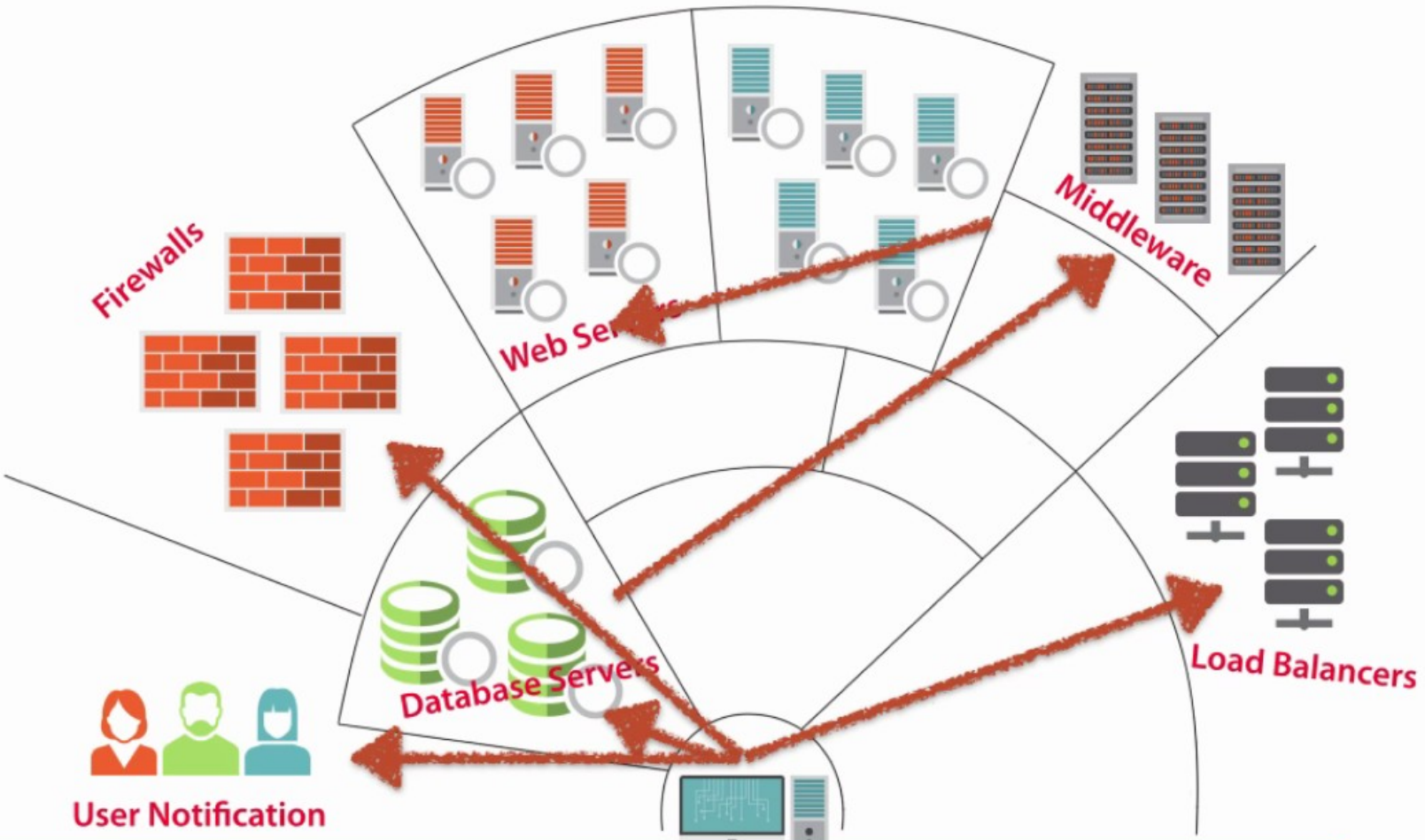
Orchestration

Coordinates automation BETWEEN systems

- Task 1 - System 1
- Task 2 - System 2
- Task 3 - System 3
- Task 4 - System 1




Firewalls
Web Servers
Middleware
Load Balancers
Database Servers
User Notification


Firewalls
Web Servers
Middleware
Database Servers
Load Balancers
User Notification

Why Ansible?

What makes it so different?

It's clean!

- No agents
- No database
- No residual software
- No complex upgrades

YAML

Ansible Execution

No programming required

NOT a markup language

Structured

Easy to read and write

Easy to extend

- URL / RESTful calls
- Shell Commands
- Scripts
- Ansible-Galaxy


Remote Server
Ansible Control Server

Ansible Requirements
(Remote Server)

Ansible Requirements (Remote Server)

*NIX:

Python 2.4 (simplejson)

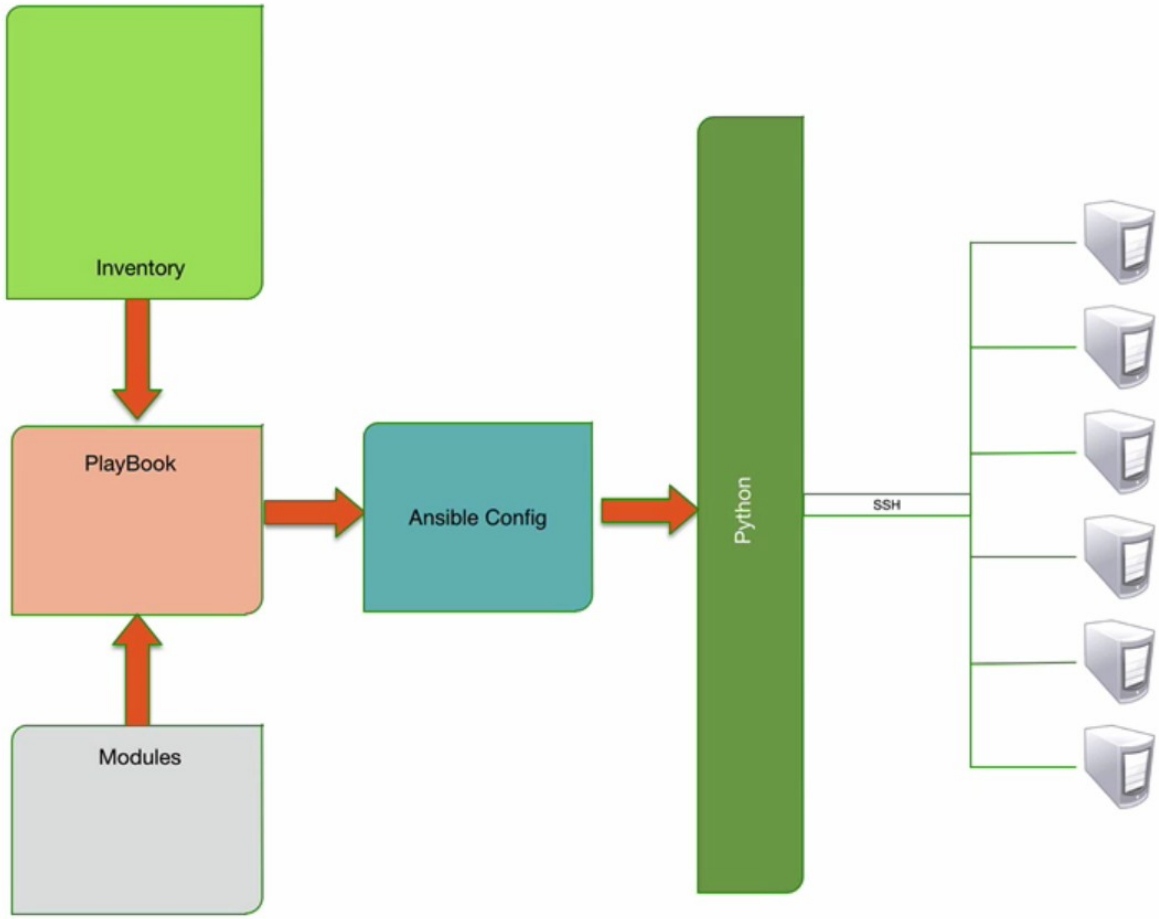
Python 2.5+

SSH

Windows:

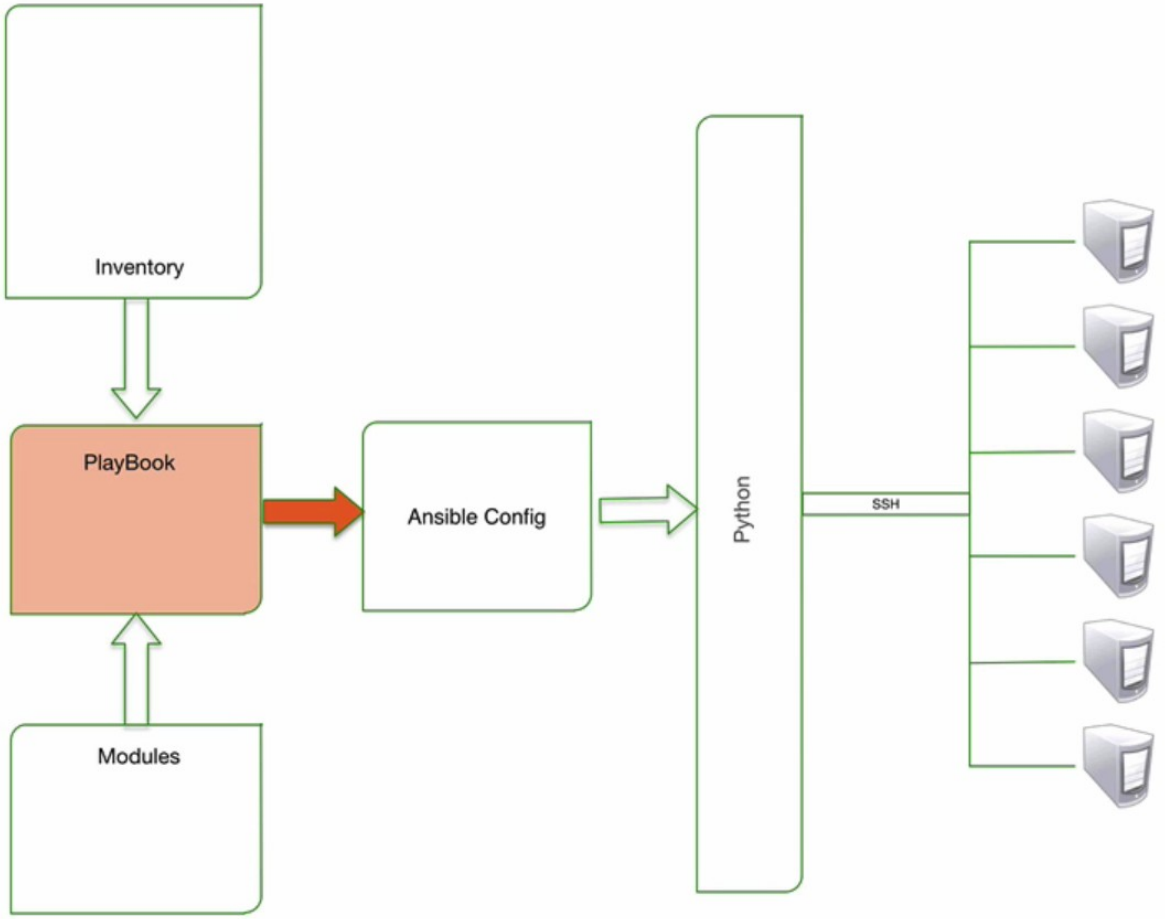
Remote Powershell Enabled

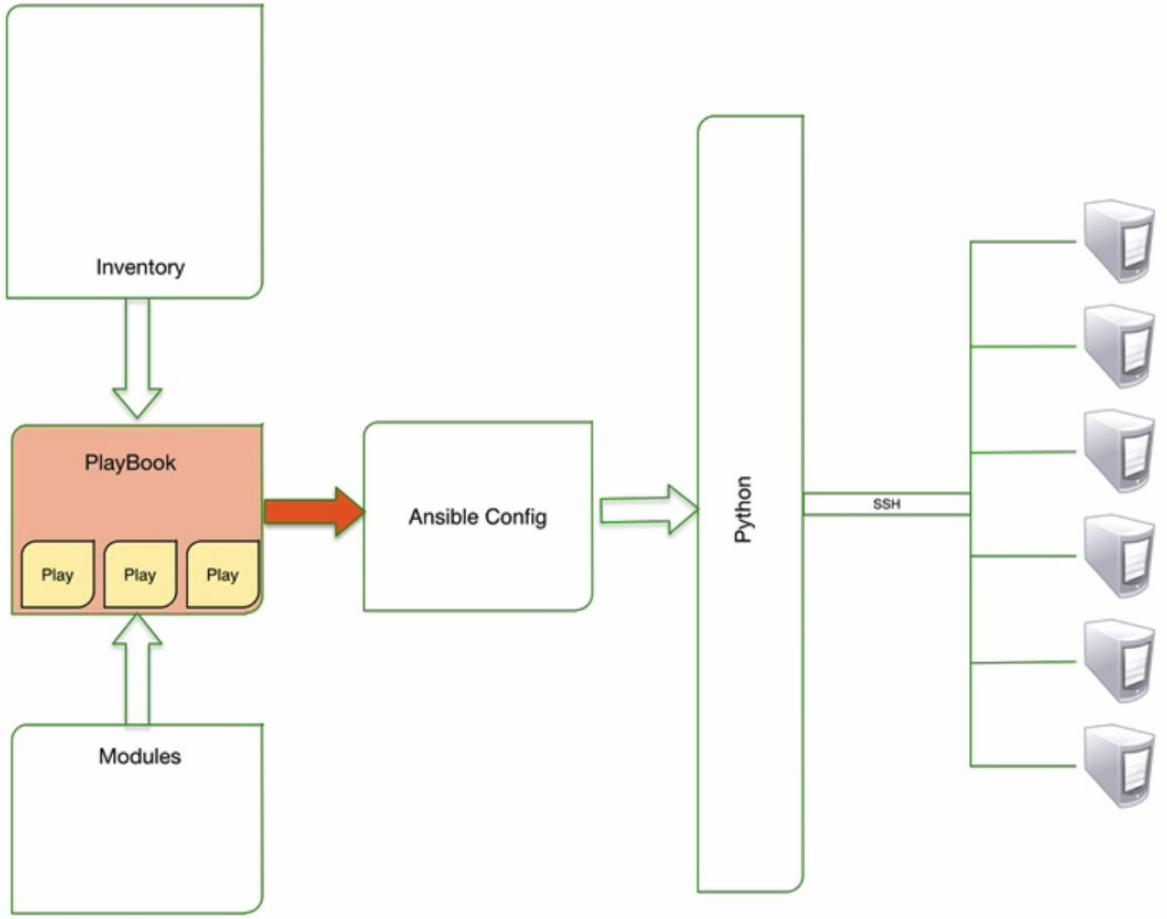
Python 3.x is not an upgrade to Python 2.x


Inventory
PlayBook
Modules
Ansible Config
Python
SSH

Module

A programmed unit of work to be done.


Inventory
PlayBook
Modules
Ansible Config
Python
SSH

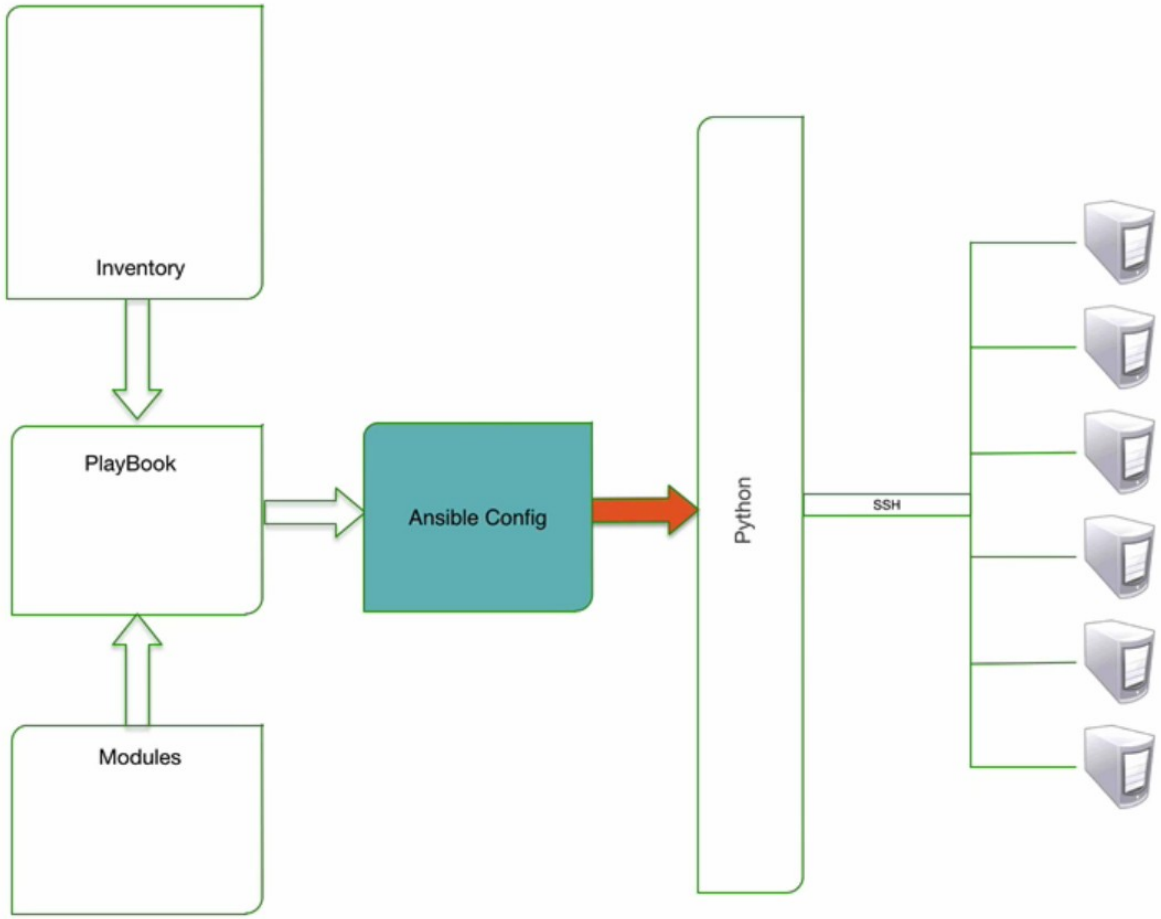

Inventory
PlayBook
Play
Play
Play
Modules
Ansible Config
Python
SSH

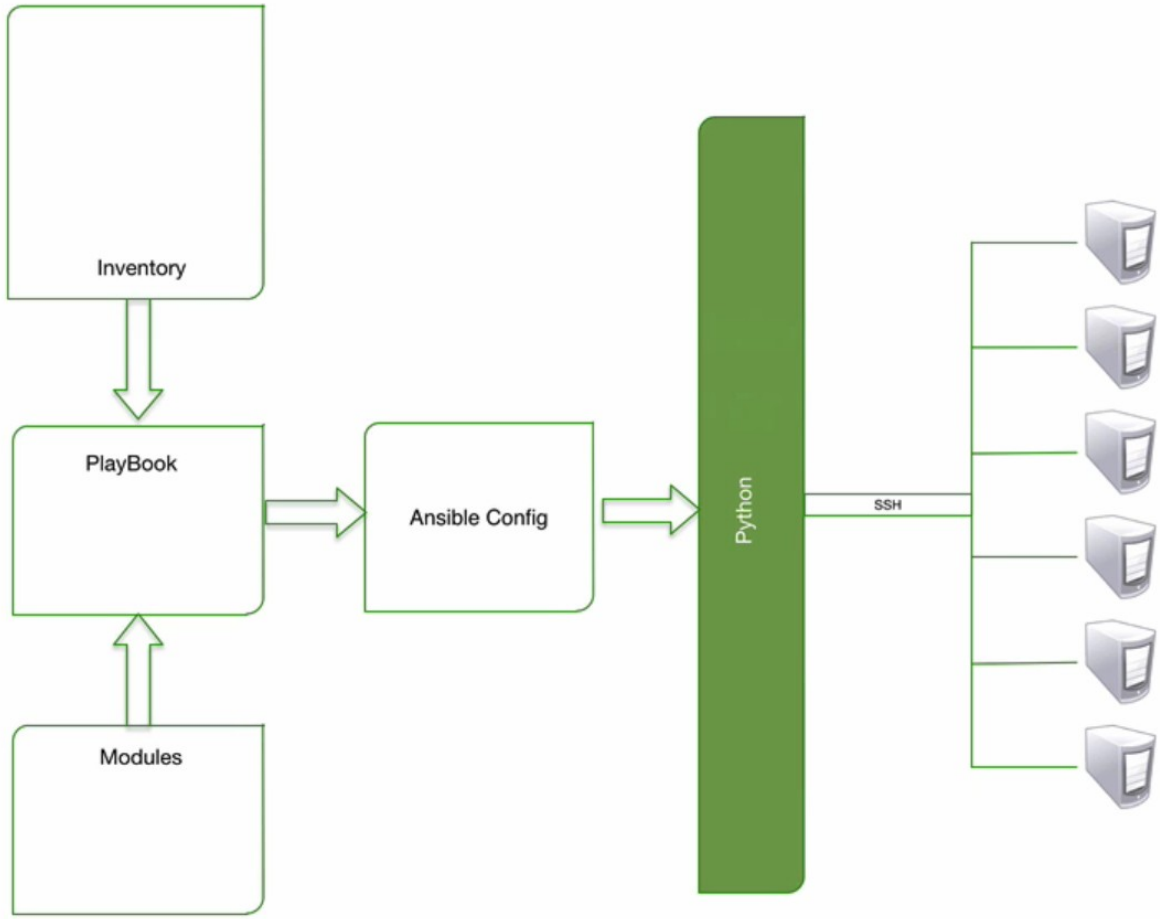
Play

A single or set of tasks using modules, executed on a defined set of hosts.

Playbook

A set of plays built in specific order sequence to produce an expected outcome or outcomes across many different sets of hosts.


Inventory
PlayBook
Modules
Ansible Config
Python
SSH


Inventory
PlayBook
Modules
Ansible Config
Python
SSH

Variables

Host Variables

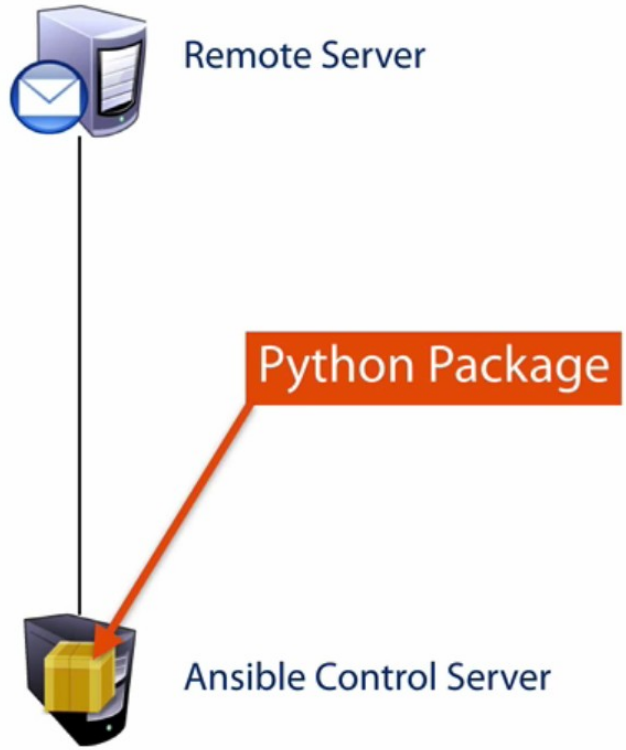
Use variables defined in Inventory per host or group

Facts

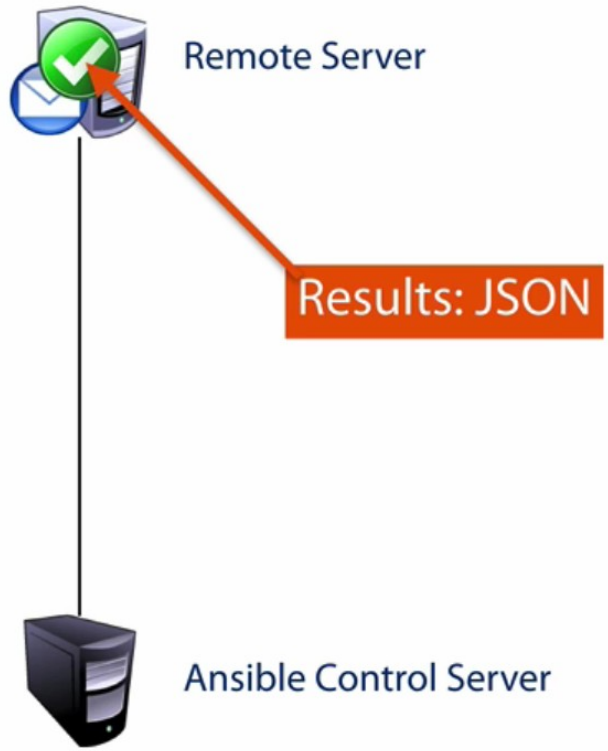
Use data gathered from the remote managed host

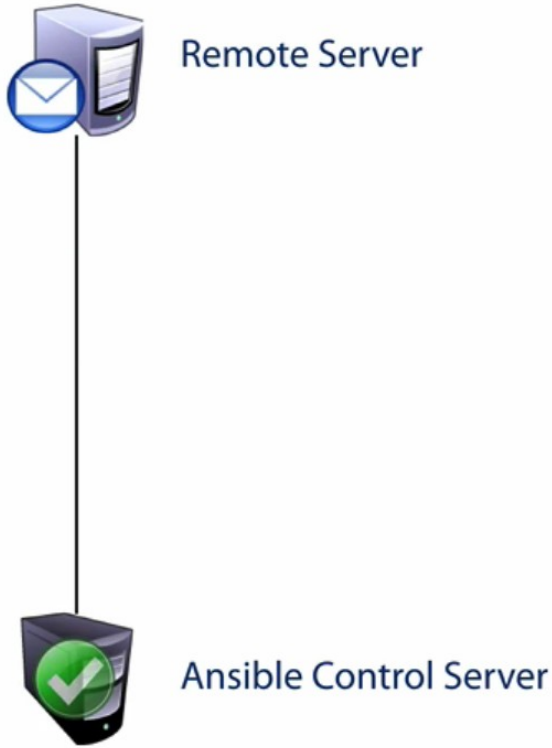
Dynamic Variables

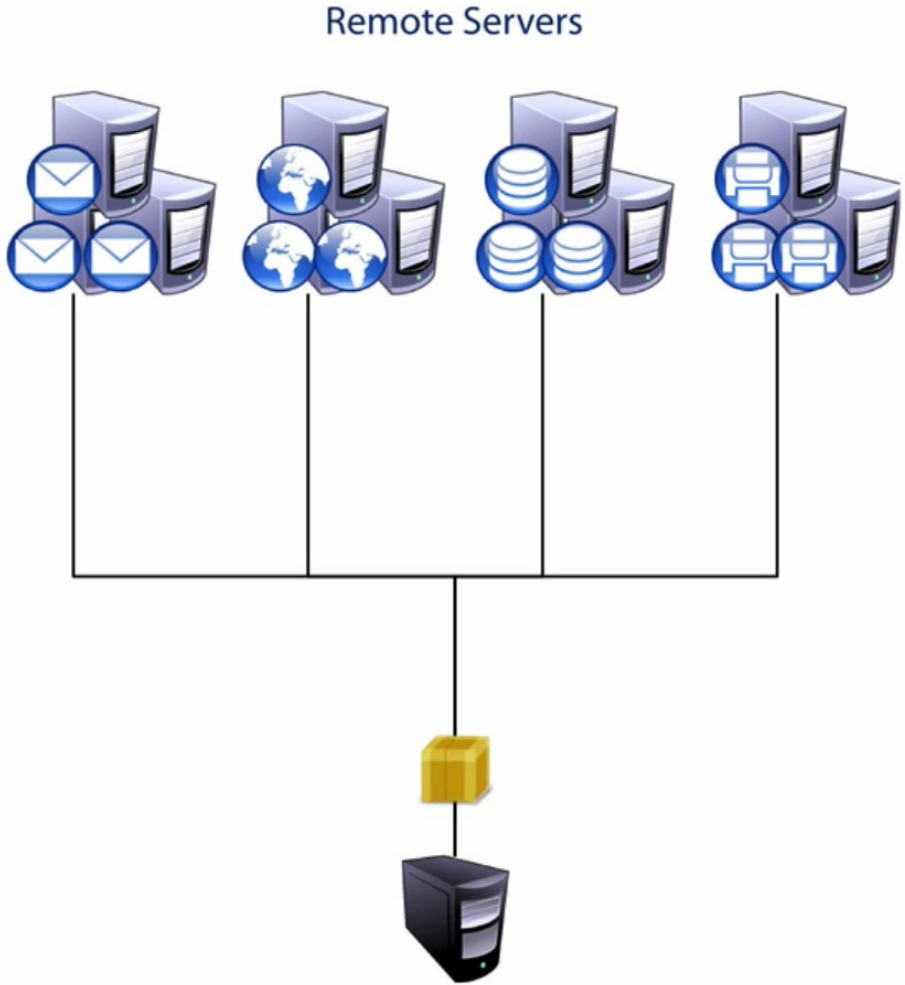
Use data gathered by tasks or created at runtime


Remote Server
Python Package
Ansible Control Server

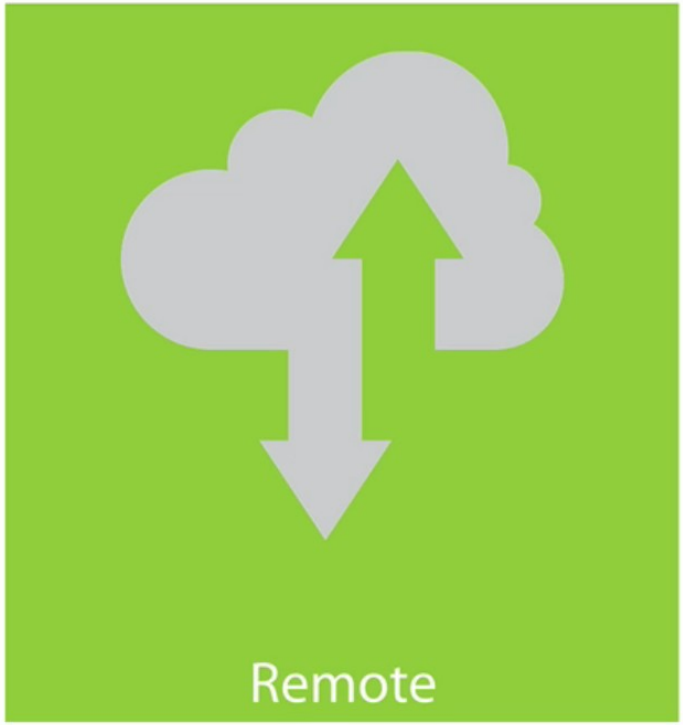

Remote Server
Ansible Control Server

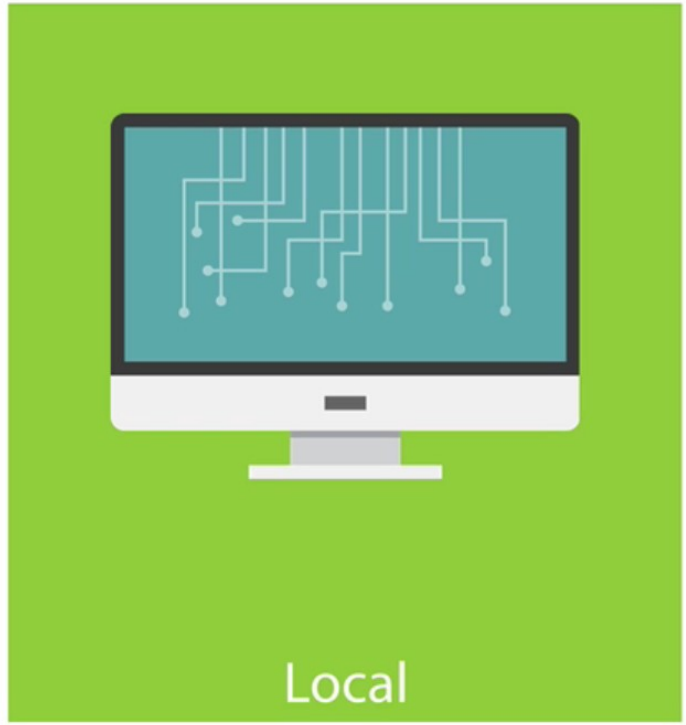

Remote Server
Results: JSON
Ansible Control Server


Remote Server
Ansible Control Server


Remote Servers

Execution Types





Remote Server

Ansible Control Server

Packaged Tasks Executed on Ansible Server

Mostly used for webservice/API calls

Remote Server

Ansible Control Server

Packaged Tasks Executed on Ansible Server

Mostly used for webservice/API calls

Ansible Architecture

Inventory maps hosts

Configuration sets Ansible parameters

Modules define actions

Ansible Architecture

Playbooks to coordinate multiple tasks

Python to build the execution

SSH to deliver the tasks

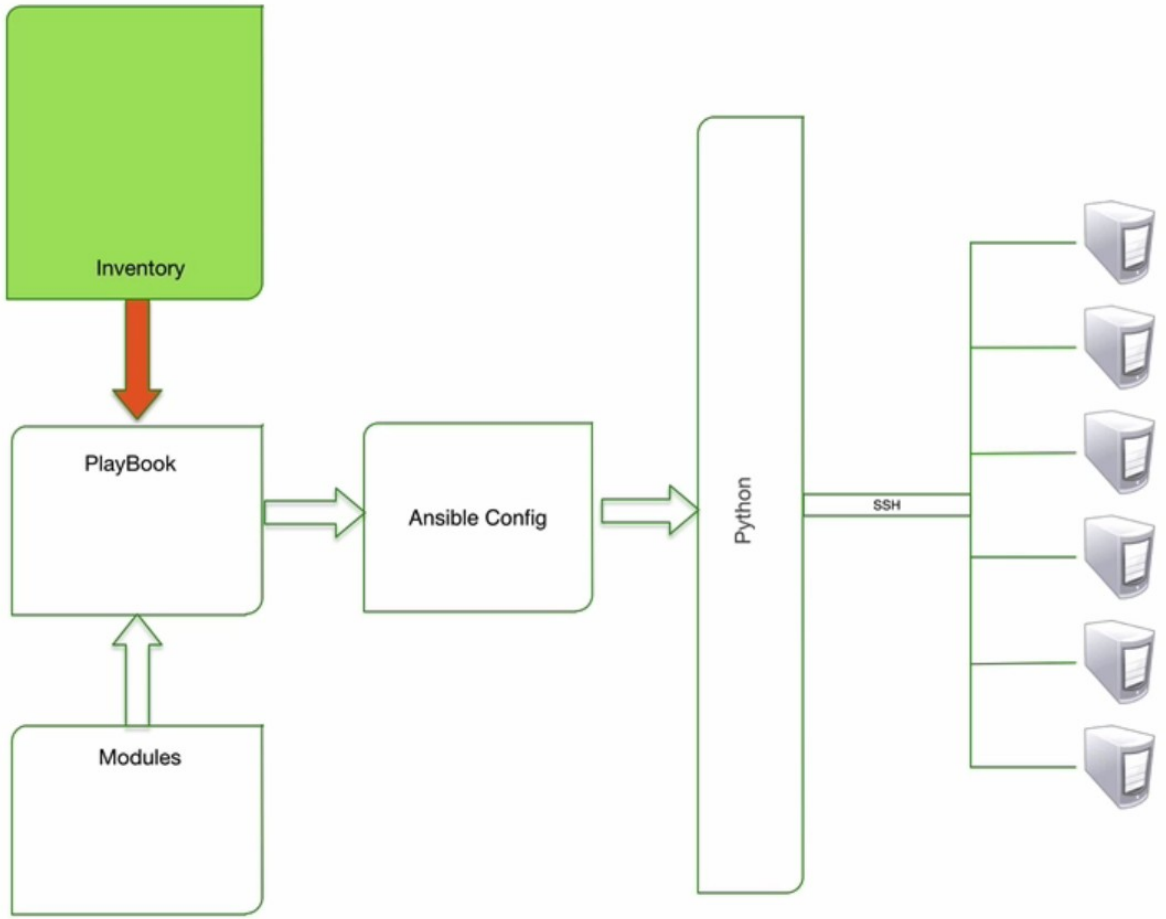
Execution Types

Remote

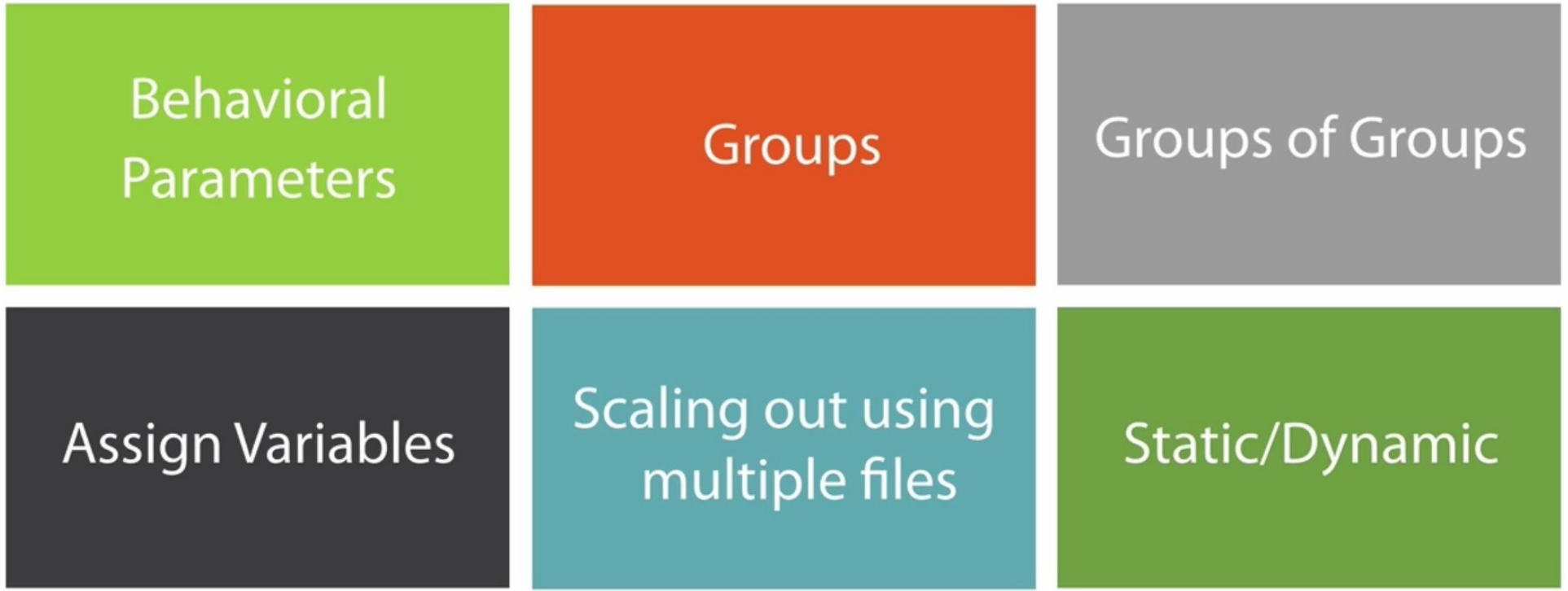
Remote execution of plays

Local

When remote box is not executing plays


Inventory
PlayBook
Modules
Ansible Config
Python
SSH

Inventory Features



Inventory File

```
[db]
db1.company.com ansible_ssh_user=aaron ansible_ssh_pass=123
db2.company.com ansible_python_interpreter=/usr/bin/python

[datacenter-west:children]
db
[datacenter-west:vars]
ansible_ssh_user=ansible_user
ansible_ssh_pass=#45e!@Gh
```

Creating our Inventory File

- Add Behavioral Parameters
- Create host-based variables
- Create a Group
- Create group-based variables

```
web1 ansible_ssh_host=192.168.33.20
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    1,36          All
```

```
web1 ansible_ssh_host=192.168.33.20 ansible_ssh_user=vagrant ansible_ssh_pass=vagrant
~
~
~
```

```
vagrant@acs:~$ cd exercise_4.1
vagrant@acs:~/exercise_4.1$ ls
vagrant@acs:~/exercise_4.1$ vim inventory
vagrant@acs:~/exercise_4.1$ ansible web1 -i inventory -m ping
web1 | FAILED => SSH encountered an unknown error during the connection. We recommend you re-run the command using -vvvv, which will en
able SSH debugging output to help diagnose the issue
vagrant@acs:~/exercise_4.1$ vim inventory
vagrant@acs:~/exercise_4.1$ ansible web1 -i inventory -m ping
web1 | success >> {
    "changed": false,
    "ping": "pong"
}

vagrant@acs:~/exercise_4.1$
```

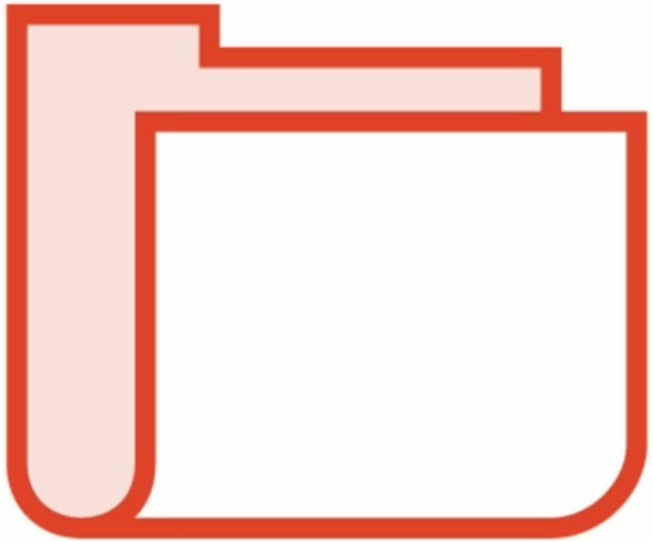
```
web1 ansible_ssh_host=192.168.33.20 ansible_ssh_user=vagrant ansible_ssh_pass=vagrant

[webservers]
web1
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                        4,5          All
```

```
vagrant@acs:~$ cd exercise_4.1
vagrant@acs:~/exercise_4.1$ ls
vagrant@acs:~/exercise_4.1$ vim inventory
vagrant@acs:~/exercise_4.1$ ansible web1 -i inventory -m ping
web1 | FAILED => SSH encountered an unknown error during the connection. We recommend you re-run the command using -vvvv, which will en
able SSH debugging output to help diagnose the issue
vagrant@acs:~/exercise_4.1$ vim inventory
vagrant@acs:~/exercise_4.1$ ansible web1 -i inventory -m ping
web1 | success >> {
    "changed": false,
    "ping": "pong"
}

vagrant@acs:~/exercise_4.1$ vim inventory
vagrant@acs:~/exercise_4.1$ ansible webservers -i inventory -m ping
web1 | success >> {
    "changed": false,
    "ping": "pong"
}

vagrant@acs:~/exercise_4.1$
```

```
web1 ansible_ssh_host=192.168.33.20 ansible_ssh_user=vagrant ansible_ssh_pass=vagrant
db1 ansible_ssh_host=192.168.33.30 ansible_user_user=vagrant ansible_ssh_pass=vagrant
[webservers]
web1

[dbservers]
db1

[datacenter:children]
webservers
dbservers


~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:
```

```
web1 ansible_ssh_host=192.168.33.20
db1 ansible_ssh_host=192.168.33.30
[webservers]
web1

[dbservers]
db1

[datacenter:children]
webservers
dbservers

[datacenter:vars]
ansible_ssh_user=vagrant
ansible_ssh_pass=vagrant


~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

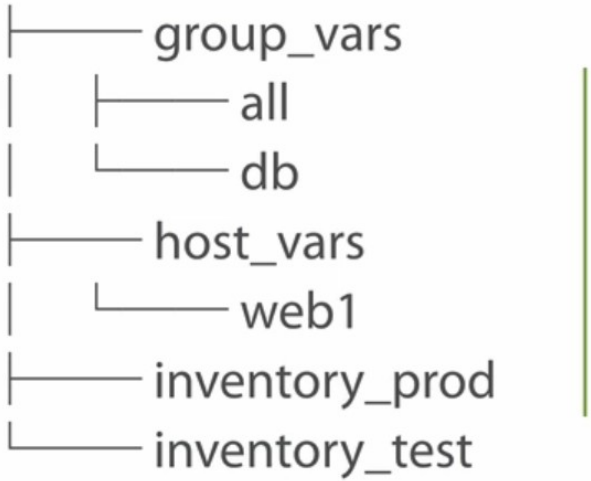
Scaling-out Inventory Files



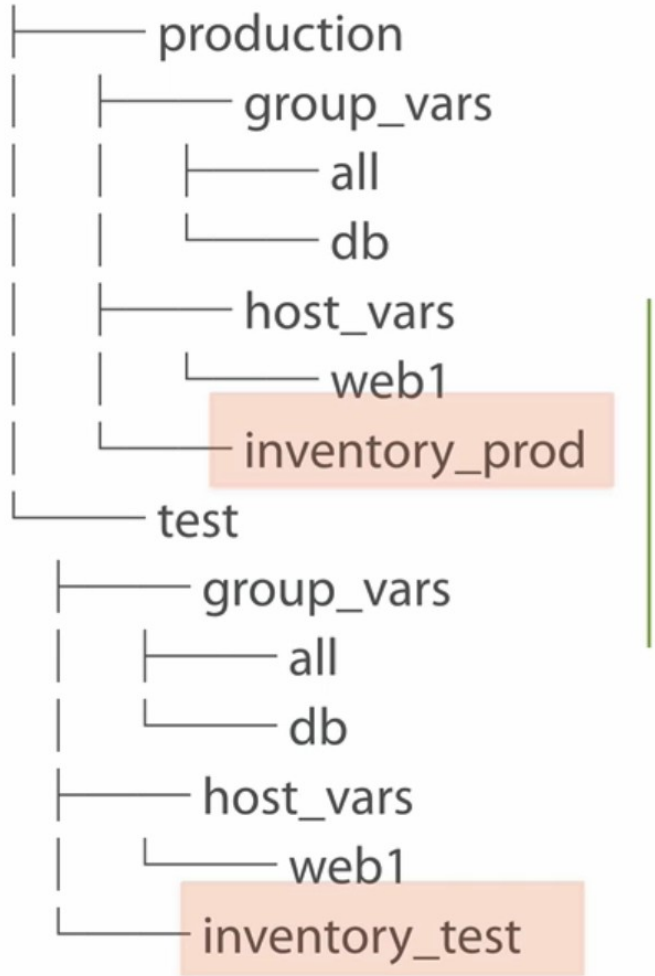
Using Directories

Can use to break-out long-running inventory files.

Very useful when dealing with large environments.



Basic Directory Structure



Multi-Environment Directory Structure

Order-of-Operations (Precedence)

1). (Group_Vars) All

2). (Group_Vars) GroupName

3). (Host_Vars) HostName

Variable File Example

```
---
# file: group_vars/dc1-west
ntp: ntp-west.company.com
syslog: logger-west.company.com
```

Variable files are written in YAML

Comments use #

Key : Value pairs

Scaling Variable Files

- Create Group Variables in separate file
- Show Order-of-Precedence

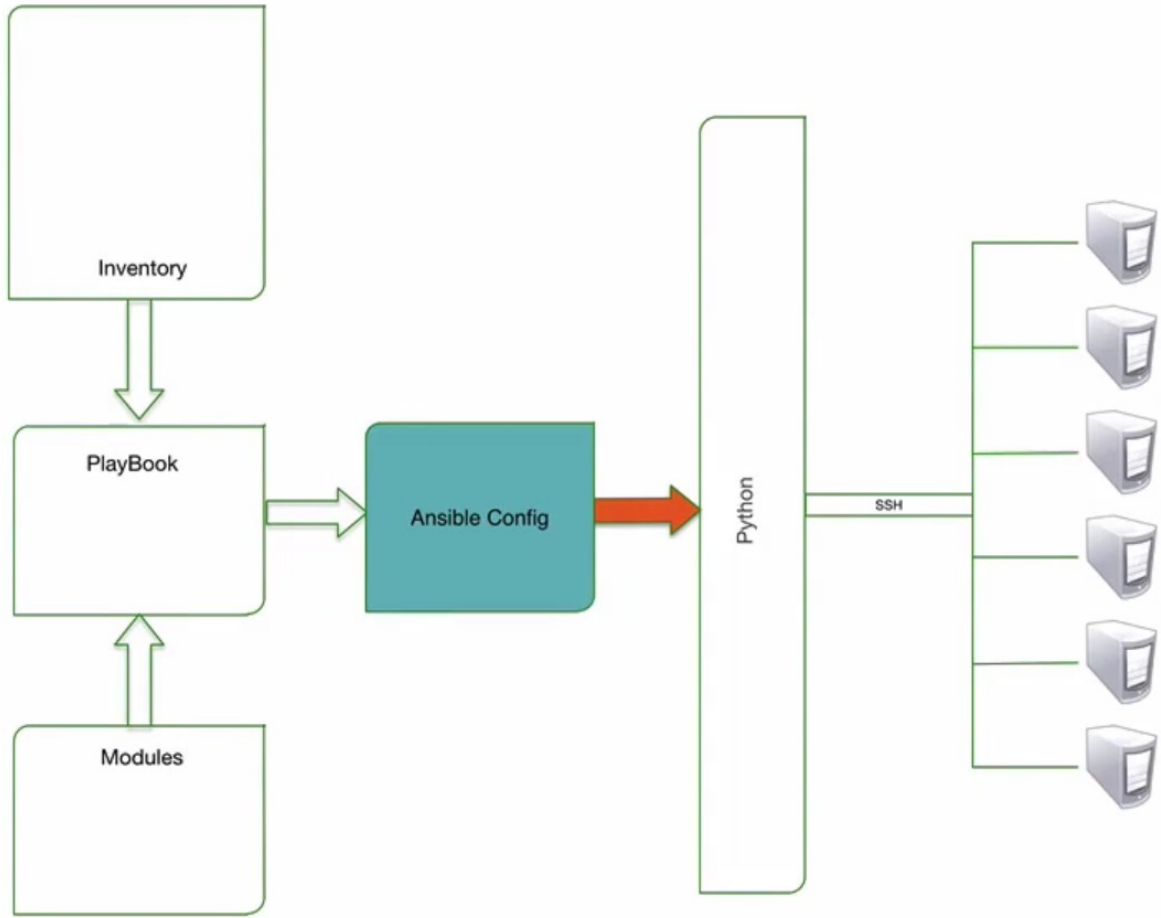
```
vagrant@acs:~/exercise_4.2$ tree
.
├── production
|   ├── group_vars
|   |   ├── all
|   |   └── db
|   ├── host_vars
|   |   └── web1
|   └── inventory_prod
└── test
    ├── group_vars
    |   ├── all
    |   └── db
    ├── host_vars
    |   └── web1
    └── inventory_test

6 directories, 8 files
vagrant@acs:~/exercise_4.2$
```

```
---

# This is our user
username:  all_username
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
vagrant@acs:~/exercise_4.2/production$ rm host_vars/web1
vagrant@acs:~/exercise_4.2/production$ ansible webservers -i inventory_prod -m user -a "name={{username}} password=12345"
web1 | FAILED >> {
    "cmd": [
        "/usr/sbin/useradd",
        "-p",
        "12345",
        "-m",
        "all_username"
    ],
    "failed": true,
    "msg": "[Errno 13] Permission denied",
    "rc": 13
}

vagrant@acs:~/exercise_4.2/production$ ansible webservers -i inventory_prod -m user -a "name={{username}} password=12345" --sudo
web1 | success >> {
    "changed": true,
    "comment": "",
    "createhome": true,
    "group": 502,
    "home": "/home/all_username",
    "name": "all_username",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 502
}

vagrant@acs:~/exercise_4.2/production$
```

```
---

# This is a good habit to have
username: group_user
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
    "state": "present",
    "system": false,
    "uid": 502
}

vagrant@acs:~/exercise_4.2/production$ ls
group_vars  host_vars  inventory_prod
vagrant@acs:~/exercise_4.2/production$ cd group_vars
vagrant@acs:~/exercise_4.2/production/group_vars$ ls
all  db
vagrant@acs:~/exercise_4.2/production/group_vars$ vim webservers
vagrant@acs:~/exercise_4.2/production/group_vars$ ansible webservers -i inventory_prod -m user -a "name={{username}} password=12345" --
sudo
ERROR: Unable to find an inventory file, specify one with -i ?
vagrant@acs:~/exercise_4.2/production/group_vars$ ansible webservers -i ../inventory_prod -m user -a "name={{username}} password=12345"
 --sudo
web1 | success >> {
    "changed": true,
    "comment": "",
    "createhome": true,
    "group": 503,
    "home": "/home/group_user",
    "name": "group_user",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 503
}

vagrant@acs:~/exercise_4.2/production/group_vars$
```


Inventory
PlayBook
Modules
Ansible Config
Python
SSH

Configuration Settings Order-of-Operations

1. $ANSIBLE_CONFIG
2. ./ansible.cfg
3. ~/.ansible.cfg
4. /etc/ansible/ansible.cfg

[defaults]
host_key_checking

Default set to True

For Production environments, do not change

Development Environment: set to False

Due to the dynamic environment of Dev, keeps it easy

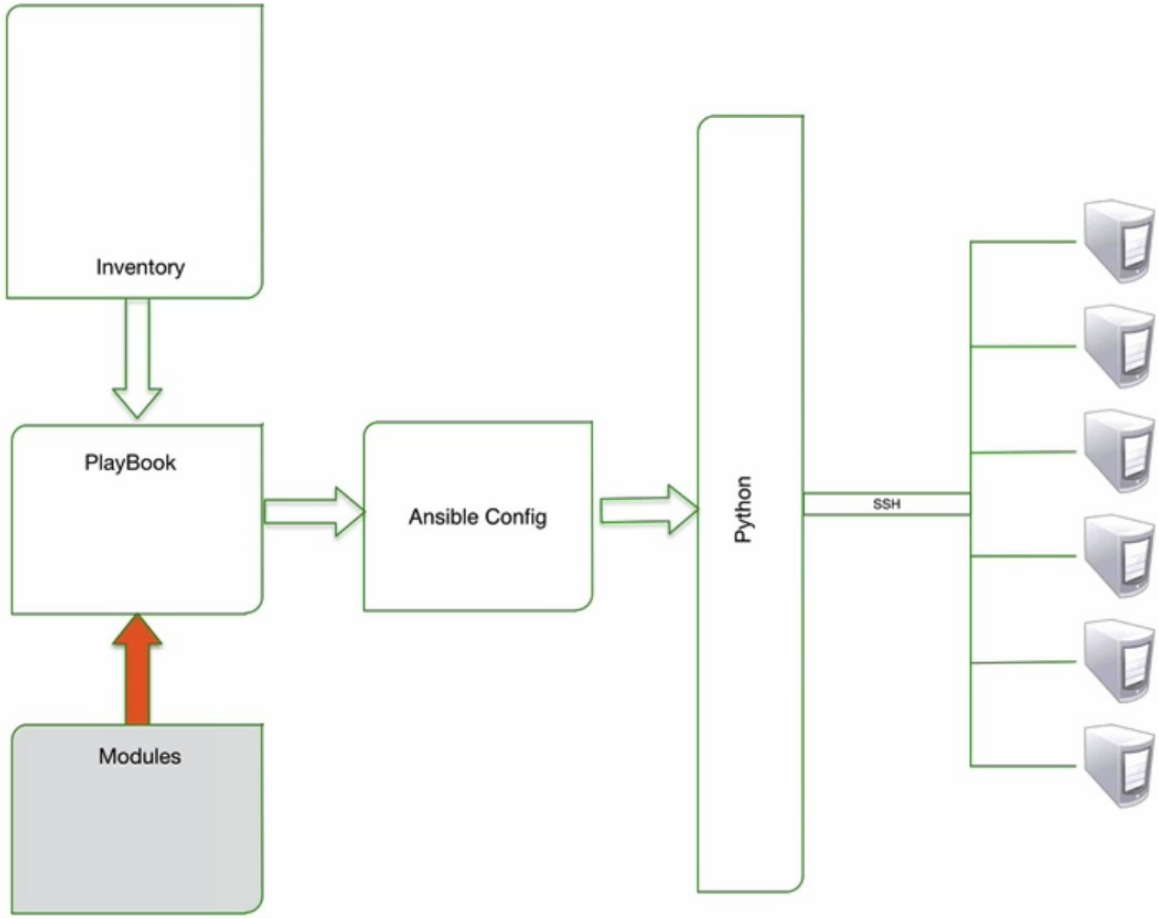
Editing Configuration

- Define settings in configuration file
- Override setting in environment variable

Configuration file

Topics

```
vagrant@acs:~/exercise_4.3/production$ tree
.
├── group_vars
|   ├── all
|   ├── db
|   └── webservers
├── host_vars
|   └── web1
└── inventory_prod

2 directories, 5 files
vagrant@acs:~/exercise_4.3/production$ vim ~/.ssh/known_hosts
vagrant@acs:~/exercise_4.3/production$ ansible web1 -i inventory_prod -m ping
web1 | FAILED => Using a SSH password instead of a key is not possible because Host Key checking is enabled and sshpass does not suppor
t this.  Please add this host's fingerprint to your known_hosts file to manage this host.
vagrant@acs:~/exercise_4.3/production$ vim ansible.cfg
vagrant@acs:~/exercise_4.3/production$ cat ~/.ssh/known_hosts
vagrant@acs:~/exercise_4.3/production$ ansible web1 -i inventory_prod -m ping
web1 | success >> {
    "changed": false,
    "ping": "pong"
}

vagrant@acs:~/exercise_4.3/production$ cat ~/.ssh/known_hosts
|1|O6SMTksdnDqur5rOlr8VQ5W/oKo=|1RA1vQL/JfScoN6CCBNGLpYv/jk= ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAtB6D1NPOMETyrlac8fVE3GTJAsd4b2zOzcQmeY
1O1PfV4TmvEeHTlC38fU1V3edj++WdMN7GoZ5cW4CB7Z5EyalrqyDGuJ63i3044x6qnz4qV/us/gpTNewOrQPKzJh7lSMP5An25f+TUIPegw/sHvZpgDmUE+8GZ+YckNp4WcToS
7P7EGR23Mux8oYOcvrTX71JRPjclMYekedc4NP1dHXw8HNGFB1ymRpWY+uhrSmQLFlM0AN2D82tu8zvk3FdHk9AKQR/Bycv54HANUBd4d0ARBHt/wmsAMQMEggzRYcBQfzJFOte
rGHma9ND46FaTTzSflvxDDDYhwz1obdq9Q==
vagrant@acs:~/exercise_4.3/production$
```


Inventory
PlayBook
Modules
Ansible Config
Python
SSH

3 Types of Modules

Core

Extras

Deprecated

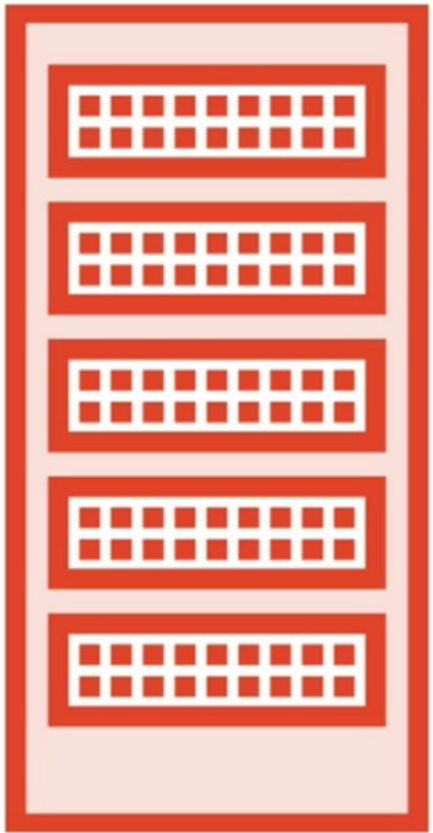
Module Docs

```
$ ansible-doc -l
```

```
$ ansible-doc -s
    <name>
```

```
$ ansible-doc <name>
```

Module Categories

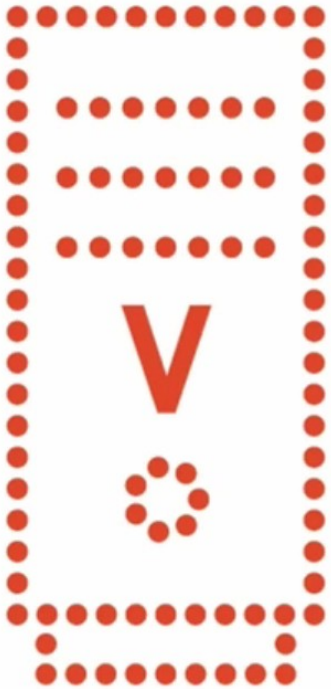


- Manage Servers
- Deploy Configurations

Module Categories

- Configure network equipment

Module Categories

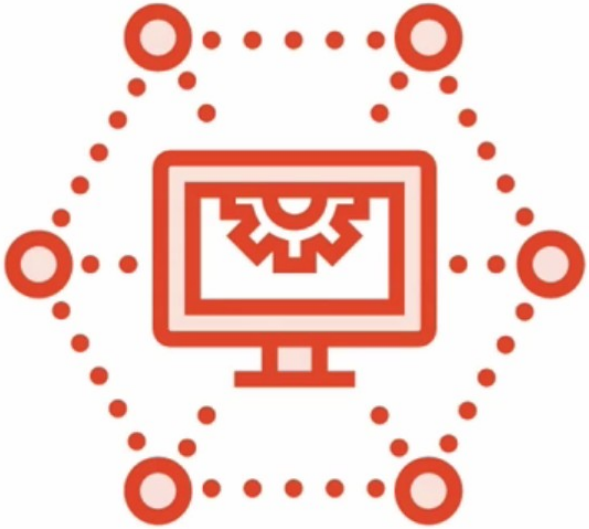


- Maintain virtual servers

Module Categories

- Manage databases and tables

Module Categories

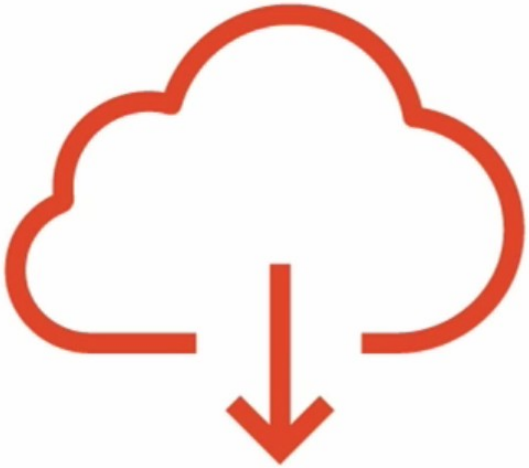


- Deploy load-balancer configurations

Copy Module

- Copies a file from local box to remote system
- Has "backup" capability
- Can do validation remotely

Fetch Module



- Pulls a file from remote host to local system
- Can use md5 checksums to validate

Apt Module

- Manages installed applications on Debian-based systems
- Can install, update, or delete packages
- Can update entire system

Yum Module

- Manages installed applications on Redhat-based systems
- Can install, update, or delete packages
- Can update entire system

Demo: Using Modules to Install/Start

- Browse module documentation
- Install Web Server (Yum module)
- Start Web Server (Service module)
- Install DB Server (Yum module)
- Start DB Server (Service module)
- Stop Firewalls (Service module)

```
vagrant@acs:~/exercise_5.1$ ansible webservers -i inventory -m yum -a "name=httpd state=present"
web1 | FAILED >> {
    "changed": true,
    "msg": "You need to be root to perform this command.\n",
    "rc": 1,
    "results": [
        "Loaded plugins: fastestmirror\n"
    ]
}

vagrant@acs:~/exercise_5.1$ ansible webservers -i inventory -m yum -a "name=httpd state=present" --sudo
web1 | success >> {
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Loaded plugins: fastestmirror\nSetting up Install Process\nLoading mirror speeds from cached hostfile\n * base: mirror
.cisp.com\n * epel: mirror.steadfast.net\n * extras: mirror.cogentco.com\n * updates: repos.mia.quadranet.com\nResolving Depend
encies\n--> Running transaction check\n---> Package httpd.x86_64 0:2.2.15-47.el6.centos.1 will be installed\n--> Finished Depen
dency Resolution\n\nDependencies Resolved\n\n================================================================================\n
 Package      Arch          Version                       Repository      Size\n================================================
================================\nInstalling:\n httpd        x86_64        2.2.15-47.el6.centos.1         updates        830
k\n\nTransaction Summary\n================================================================================\nInstall       1 Pac
kage(s)\n\nTotal download size: 830 k\nInstalled size: 2.9 M\nDownloading Packages:\nRunning rpm_check_debug\nRunning Transacti
on Test\nTransaction Test Succeeded\nRunning Transaction\n\r  Installing : httpd-2.2.15-47.el6.centos.1.x86_64
        1/1 \n\r  Verifying  : httpd-2.2.15-47.el6.centos.1.x86_64                                1/1 \n\nInstalled:\n  httpd.x86_64
 0:2.2.15-47.el6.centos.1                                      \n\nComplete!\n"
    ]
}
```

```
> SERVICE

  Controls services on remote hosts.

Options (= is mandatory):

- arguments
        Additional arguments provided on the command line

- enabled
        Whether the service should start on boot. At least one of
        state and enabled are required. (Choices: yes, no)

= name
        Name of the service.

- pattern
        If the service does not respond to the status command, name a
        substring to look for as would be found in the output of the
        `ps' command as a stand-in for a status result.  If the string
        is found, the service will be assumed to be running.

- runlevel
        For OpenRC init scripts (ex: Gentoo) only.  The runlevel that
        this service belongs to.

- sleep
        If the service is being `restarted' then sleep this many
        seconds between the stop and start command. This helps to
        workaround badly behaving init scripts that exit immediately
        after signaling a process to stop.
```

```
vagrant@acs:~/exercise_5.1$ ansible webservers -i inventory -m service -a "name=httpd enabled=yes state=started" --sudo
web1 | success >> {
    "changed": true,
    "enabled": true,
    "name": "httpd",
    "state": "started"
}

vagrant@acs:~/exercise_5.1$
```

```
vagrant@acs:~/exercise_5.1$ ansible dbservers -i inventory -m yum -a "name=mysql-server state=present" --sudo
db1 | success >> {
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Loaded plugins: fastestmirror\nSetting up Install Process\nLoading mirror speeds from cached hostfile\n * base: centos
.mia.host-engine.com\n * extras: mirror.sanctuaryhost.com\n * updates: centos.chi.host-engine.com\nResolving Dependencies\n-->
Running transaction check\n---> Package mysql-server.x86_64 0:5.1.73-5.el6_6 will be installed\n--> Finished Dependency Resolut
ion\n\nDependencies Resolved\n\n================================================================================\n Package
        Arch          Version                 Repository    Size\n================================================================
====================\nInstalling:\n mysql-server        x86_64        5.1.73-5.el6_6         base         8.6 M\n\nTransact
ion Summary\n================================================================================\nInstall       1 Package(s)\n\nTo
tal download size: 8.6 M\nInstalled size: 25 M\nDownloading Packages:\nRunning rpm_check_debug\nRunning Transaction Test\nTrans
action Test Succeeded\nRunning Transaction\n\r  Installing : mysql-server-5.1.73-5.el6_6.x86_64                          1/1 \
n\r  Verifying  : mysql-server-5.1.73-5.el6_6.x86_64                          1/1 \n\nInstalled:\n  mysql-server.x86_64 0:5.1.
73-5.el6_6                                  \n\nComplete!\n"
    ]
}

vagrant@acs:~/exercise_5.1$
```

```
vagrant@acs:~/exercise_5.1$ ansible dbservers -i inventory -m yum -a "name=mysql-server state=present" --sudo
db1 | success >> {
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Loaded plugins: fastestmirror\nSetting up Install Process\nLoading mirror speeds from cached hostfile\n * base: centos
.mia.host-engine.com\n * extras: mirror.sanctuaryhost.com\n * updates: centos.chi.host-engine.com\nResolving Dependencies\n--> 
Running transaction check\n---> Package mysql-server.x86_64 0:5.1.73-5.el6_6 will be installed\n--> Finished Dependency Resolut
ion\n\nDependencies Resolved\n\n================================================================================\n Package
         Arch          Version                 Repository    Size\n================================================================
====================\nInstalling:\n mysql-server        x86_64        5.1.73-5.el6_6         base         8.6 M\n\nTransact
ion Summary\n================================================================================\nInstall       1 Package(s)\n\nTo
tal download size: 8.6 M\nInstalled size: 25 M\nDownloading Packages:\nRunning rpm_check_debug\nRunning Transaction Test\nTrans
action Test Succeeded\nRunning Transaction\n\r  Installing : mysql-server-5.1.73-5.el6_6.x86_64                       1/1 \
n\r  Verifying  : mysql-server-5.1.73-5.el6_6.x86_64                       1/1 \n\nInstalled:\n  mysql-server.x86_64 0:5.1.
73-5.el6_6                                  \n\nComplete!\n"
    ]
}

vagrant@acs:~/exercise_5.1$ ansible dbservers -i inventory -m service -a "name=mysqld state=started" --sudo
```

```
vagrant@acs:~/exercise_5.1$ ansible webservers:dbservers -i inventory -m service -a "name=iptables state=stopped" --sudo
```

Host/Group Target Patterns

OR
(group1:group2)

NOT
(!group2)

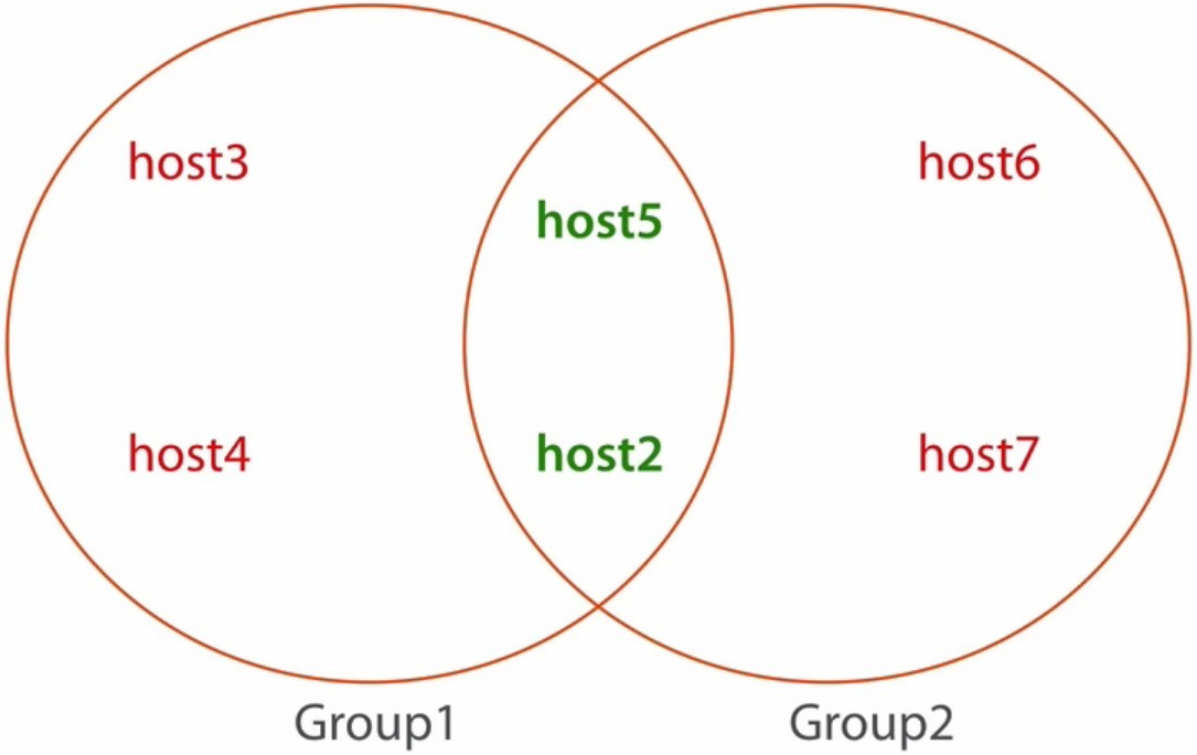
Wildcard
(web*.ex.com)

Regex
(~web[0-9]+)

Complex Patterns

AND

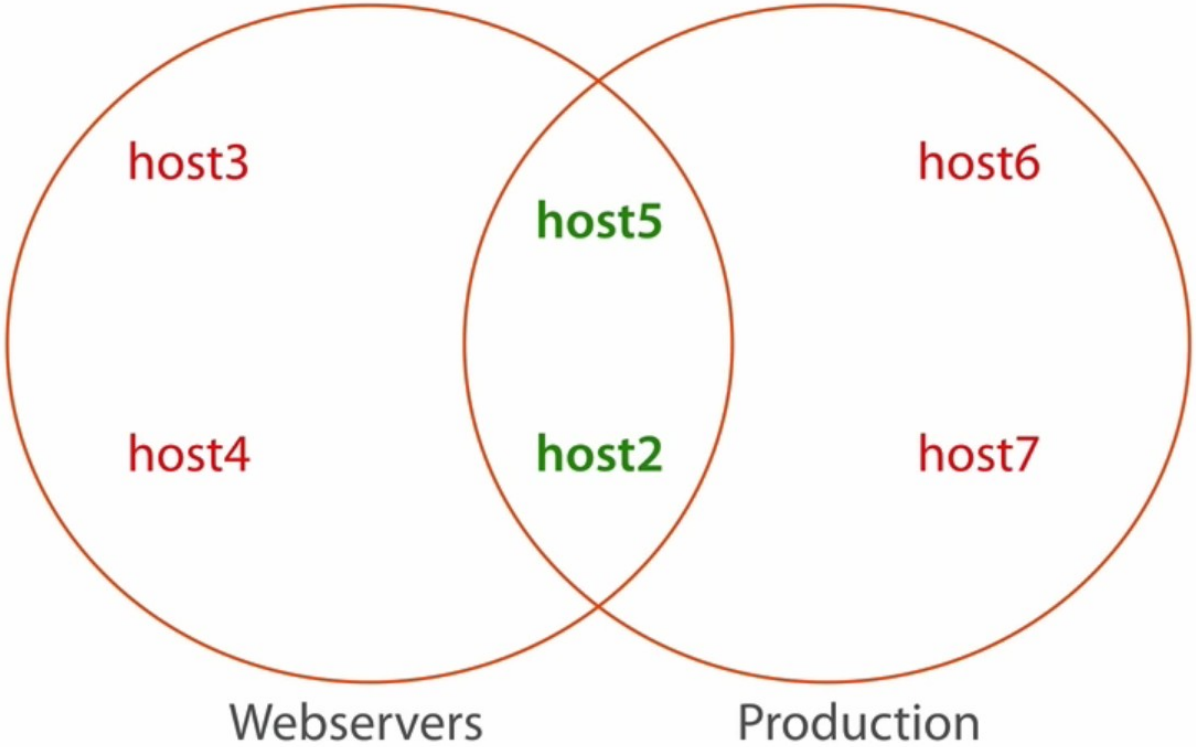
(group1:&group2)



Complex Patterns

AND

(Webservers:&Production)



Demo: Using Setup Module

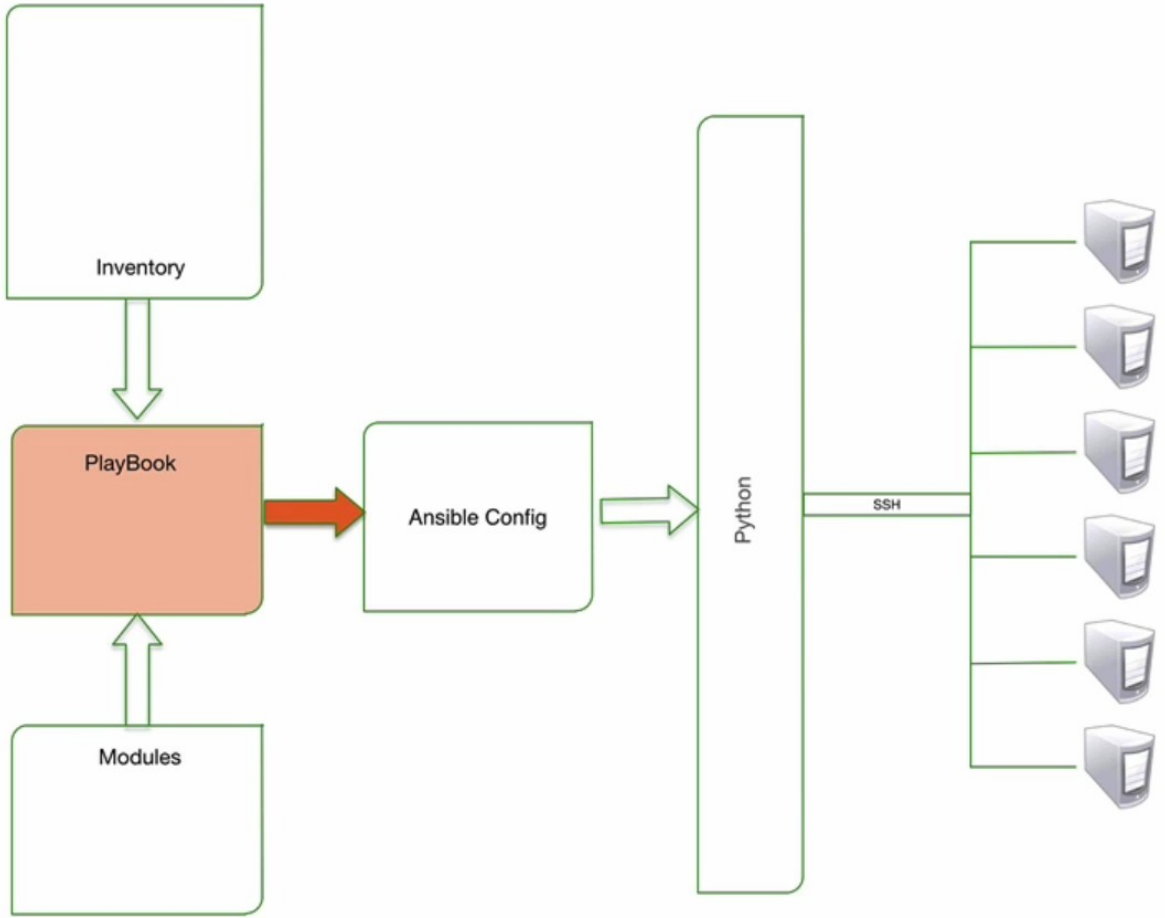
- Gather facts on remote systems
- Used in Playbooks

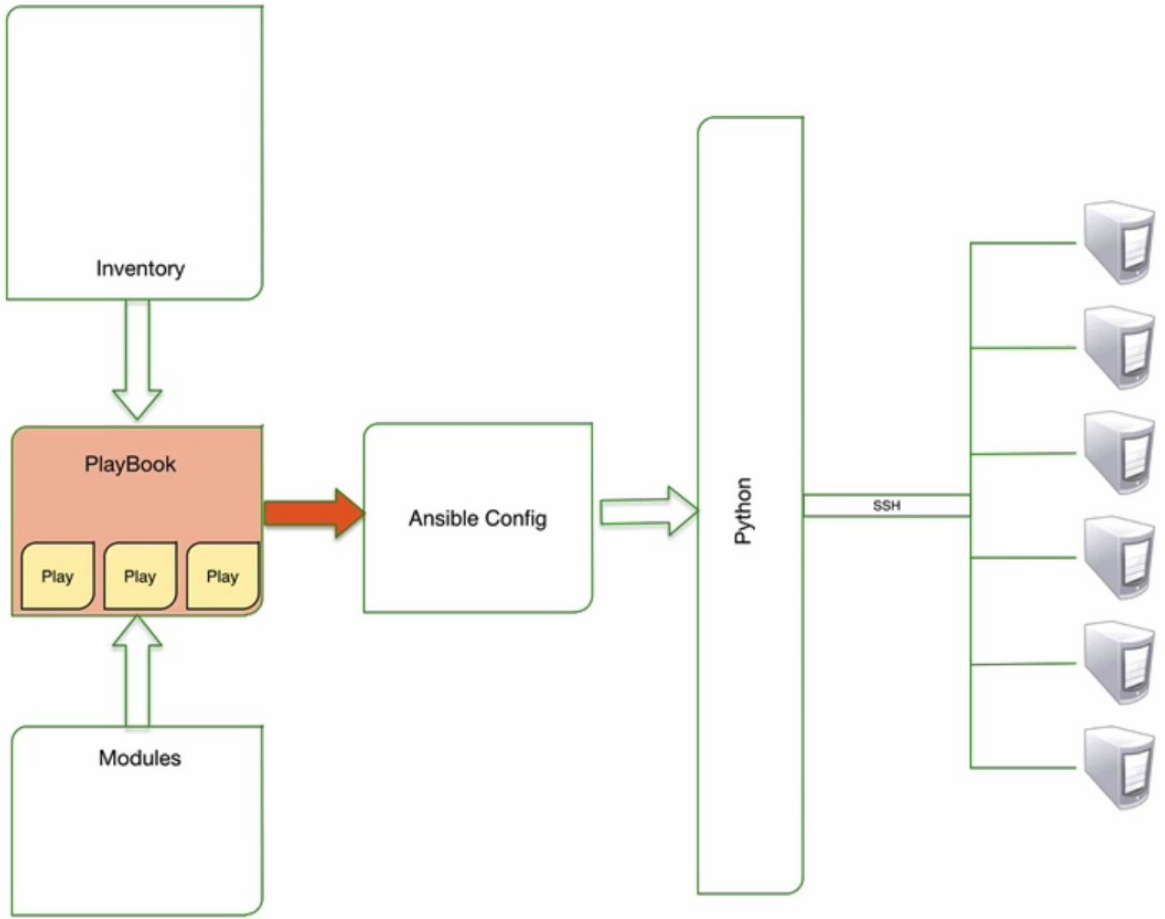
```
vagrant@acs:~/exercise_5.1$ ansible web1 -i inventory -m setup
```

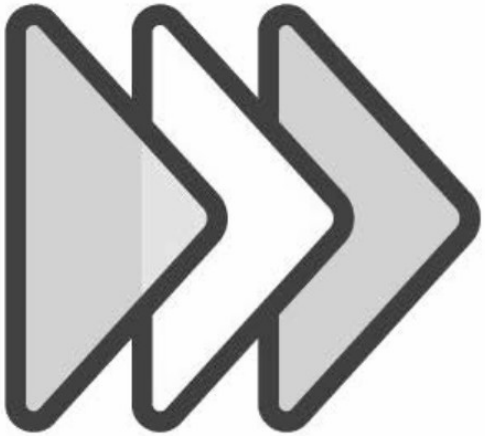
```
vagrant@acs:~/exercise_5.1$ ansible web1 -i inventory -m setup -a "filter=ansible_eth*"
```

```
    },
    "changed": false
}

vagrant@acs:~/exercise_5.1$
vagrant@acs:~/exercise_5.1$
vagrant@acs:~/exercise_5.1$ ansible web1 -i inventory -m setup -a "filter=ansible_mounts"
web1 | success >> {
    "ansible_facts": {
        "ansible_mounts": [
            {
                "device": "/dev/mapper/VolGroup-lv_root",
                "fstype": "ext4",
                "mount": "/",
                "options": "rw",
                "size_available": 194492116992,
                "size_total": 206494973952
            },
            {
                "device": "/dev/sda1",
                "fstype": "ext4",
                "mount": "/boot",
                "options": "rw",
                "size_available": 424284160,
                "size_total": 499355648
            }
        ]
    },
    "changed": false
}
```

```
vagrant@acs:~/exercise_5.1$ ansible all -m setup --tree ./setup
No hosts matched
vagrant@acs:~/exercise_5.1$ ansible all -i inventory -m setup --tree ./setup
```


Inventory
PlayBook
Modules
Ansible Config
Python
SSH


Inventory
PlayBook
Play
Play
Play
Modules
Ansible Config
Python
SSH



Plays map hosts to tasks

A play can have multiple tasks

A playbook can have multiple plays

Playbook Breakdown

```
---
- hosts: webservers
  remote_user: root
  tasks:
  - name: Install Apache
    yum: name=httpd state=present
  - name: Start Apache
    service: name=httpd state=started

- hosts: dbservers
  remote_user: root
  tasks:
  - name: Install MySQL
    yum: name=mysql-server state=present
  - name: Start MySQL
    service: name=mysqld state=started
```

← Playbook

YAML Whitespace

```
---
- hosts: webservers
  remote_user: root
  tasks:
  - name: Install Apache
    yum: name=httpd state=present
  - name: Start Apache
    service: name=httpd state=started

- hosts: dbservers
  remote_user: root
  tasks:
  - name: Install MySQL
    yum: name=mysql-server state=present
  - name: Start MySQL
    service: name=mysqld state=started
```

Whitespace is crucial!

Play Breakdown

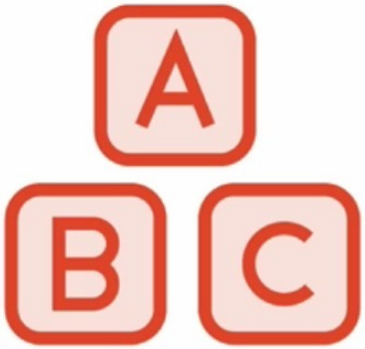
```
---
- hosts: webservers
  remote_user: root
  tasks:
    - name: Install Apache
      yum: name=httpd state=present
    - name: Start Apache
      service: name=httpd state=started
```

Global Play Declaration

Play Declarations

```
- hosts: webservers
  vars:
    git_repo: https://github.com/repo.git
    http_port: 8080
    db_name: wordpress
  sudo: yes
  sudo_user: wordpress_user
  gather_facts: no
```

Declare user to run tasks

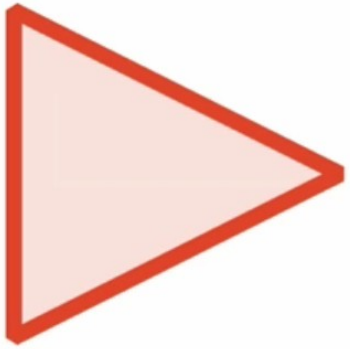


Tasks are executed in order - top down

Tasks use modules

Tasks

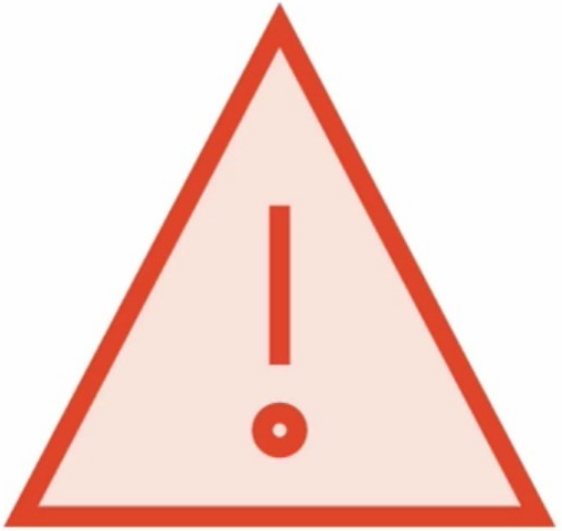
```
tasks:
  - name: Name this task for readability
    module: parameters=go_here

  - name: Deploy Apache Configuration File
    copy: src=/ansible/files/conf/httpd.conf
          dest=/etc/httpd/conf/
```



Execution of playbooks:

```
$ ansible-playbook playbook.yml
```

If a host fails a task, that host is removed from the rest of the playbook execution



Retrying Failed Host Executions

```
PLAY RECAP *********************************************************************
        to retry, use: --limit @/home/vagrant/ping.retry

db1                        : ok=0    changed=0    unreachable=1    failed=0
web1                       : ok=2    changed=0    unreachable=0    failed=0
```

Demo: Our First Playbook

Write a playbook

Add play to install web server

Add play to install db server

Add play to start services

Fail a play

Retry a failed play

```
vagrant@acs:~/exercise_6.1$ ls
ansible.cfg  inventory
vagrant@acs:~/exercise_6.1$ cat ansible.cfg
[defaults]
hostfile = inventory
vagrant@acs:~/exercise_6.1$ cat inventory
acs ansible_ssh_host=192.168.33.10 ansible_ssh_user=vagrant ansible_ssh_pass=vagrant
web1 ansible_ssh_host=192.168.33.20 ansible_ssh_user=vagrant ansible_ssh_pass=vagrant username=ansible_user
db1 ansible_ssh_host=192.168.33.30
[webservers]
web1

[dbservers]
db1

[datacenter:children]
webservers
dbservers

[datacenter:vars]
ansible_ssh_user=vagrant
ansible_ssh_pass=vagrant

vagrant@acs:~/exercise_6.1$
```

```
vagrant@acs:~/exercise_6.1$ vim web_db.playbook
```

```
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
  - name: Start MySQL
    service: name=mysqld state=started

- hosts: webservers:dbservers
  sudo: yes

  tasks:
  - name: Stop IPTABLES NOW!!!
    service: name=iptables state=stopped
~
~
~
~
```

```
vagrant@acs:~/exercise_6.1$ ansible-playbook web_db.yaml

PLAY [webservers] ***************************************************************

GATHERING FACTS *****************************************************************
ok: [web1]

TASK: [Ensure that Apache is installed] *****************************************
ok: [web1]

TASK: [Start Apache Services] ***************************************************
ok: [web1]

PLAY [dbservers] ****************************************************************

GATHERING FACTS *****************************************************************
fatal: [db1] => SSH encountered an unknown error during the connection. We recommend you re-run the command using -vvvv, which
will enable SSH debugging output to help diagnose the issue

TASK: [Ensure MySQL is installed] ***********************************************
FATAL: no hosts matched or all hosts have already failed -- aborting


PLAY RECAP **********************************************************************
          to retry, use: --limit @/home/vagrant/web_db.yaml.retry

db1                        : ok=0    changed=0    unreachable=1    failed=0
web1                       : ok=3    changed=0    unreachable=0    failed=0

vagrant@acs:~/exercise_6.1$ cat /home/vagrant/web_db.yaml.retry
db1
```

```
db1                        : ok=0    changed=0    unreachable=1    failed=0
web1                       : ok=3    changed=0    unreachable=0    failed=0

vagrant@acs:~/exercise_6.1$ cat /home/vagrant/web_db.yaml.retry
db1
vagrant@acs:~/exercise_6.1$ vim inventory
vagrant@acs:~/exercise_6.1$ ansible-playbook web_db.yaml --limit /home/vagrant/web_db.yaml.retry
ERROR: provided hosts list is empty
vagrant@acs:~/exercise_6.1$ ansible-playbook web_db.yaml --list @/home/vagrant/web_db.yaml.retry
Usage: ansible-playbook playbook.yml

ansible-playbook: error: ambiguous option: --list (--list-hosts, --list-tasks?)
vagrant@acs:~/exercise_6.1$ ansible-playbook web_db.yaml --limit @/home/vagrant/web_db.yaml.retry

PLAY [webservers] **************************************************************
skipping: no hosts matched

PLAY [dbservers] ***************************************************************

GATHERING FACTS ****************************************************************
ok: [db1]

TASK: [Ensure MySQL is installed] **********************************************
ok: [db1]

TASK: [Start MySQL] ************************************************************
ok: [db1]

PLAY [webservers:dbservers] ****************************************************

GATHERING FACTS ****************************************************************
```

Including Files

Include Files to Extend Playbook

```
tasks:
  - include: wordpress.yml
    vars:
      sitename: My Awesome Site
  - include: loadbalancer.yml
  - include_vars: variables.yml
```

- Breaks up long playbooks
- Use to add external variable files
- Reuse other playbooks

Register Task Output

Grab output of task for another task

```
tasks:
  - shell: /usr/bin/whoami
    register: username
  - file: path=/home/myfile.txt
          owner={{ username }}
```

- Useful to use tasks to feed data into other tasks
- Useful to create custom error trapping

Debug Module

Add debug to tasks

```
tasks:
  - debug: msg="This host is
{{ inventory_hostname }} during
execution"

  - shell: /usr/bin/whoami
    register: username
  - debug: var=username
```

- Useful to send output to screen during execution
- Helps find problems

Prompting for Input

Prompt user during execution

```
- hosts: web1

  vars_prompt:
   - name: "sitename"
     prompt: "What is new site name?"

  tasks:
   - debug: msg="The name is {{ sitename }}"
```

- Creates Dynamic Playbooks

Playbook Handlers

- Tasks with asynchronous execution
- Only runs tasks when notified
- Tasks only notify when state=changed
- Does not run until all playbook tasks have executed
- Most common for restarting services to load changes (if changes are made)

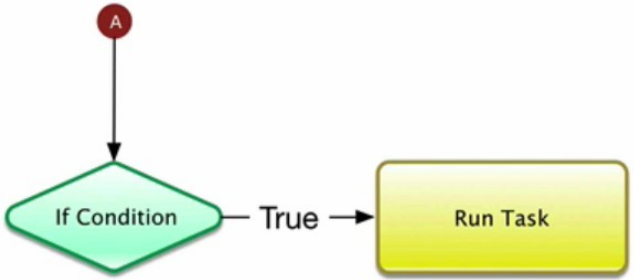
Handlers

Notify handlers from your tasks

```
tasks:
   - copy: src=files/httpd.conf
           dest=/etc/httpd/conf/
     notify:
        - Apache Restart
handlers:
   - name: Apache Restart
     service: name=httpd state=restarted
```

- Copies config file to host
- If state=change on "COPY", tell "Apache Restart"
- Run "Service" module.

Conditional Execution



Use the clause "when" to choose if task should run.

Conditional Clause

Choose when to execute tasks

```
tasks:
   - yum: name=httpd state=present
     when: ansible_os_family == “RedHat”

   - apt: name=apache2 state=present
     when: ansible_os_family == “Debian”
```

- Uses YUM if OS is RedHat
- Uses APT if OS is Debian

Conditional Clause Based on Output

Choose when to execute tasks

```
tasks:
  - command: ls /path/doesnt/exist
    register: result
    ignore_errors: yes

  - debug: msg="Failure!"
    when: result|failed
```

- Track whether previous task ran
- Searches JSON result for status
- Status Options:
  - success
  - failed
  - skipped

Templates

Uses Jinja2 Engine

Insert variables into static files

Creates and copies dynamic files

Deploy custom configurations

Template Module

Modify Template and Copy

```
tasks:
   - template:
       src=templates/httpd.j2
       dest=/etc/httpd/conf/httpd.conf
       owner=httpd
```

- Takes a file with pre-defined variable names
- Inserts variable values in file
- Copies file to destination

httpd.j2

```
.....

<VirtualHost *:80>
  ServerAdmin {{ server_admin }}
  DocumentRoot {{ site_root }}
  ServerName {{ inventory_hostname }}
</VirtualHost>

.....
```

Demo: Playbook Controls

Add install decisions based on OS

Create template for Apache Config

Deploy configuration

Restart service if needed

```
vagrant@acs:~/exercise_6.2$ ls
ansible.cfg  inventory  templates  web_db.yaml
vagrant@acs:~/exercise_6.2$ vim
```

```yaml
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
  - name: Start MySQL
    service: name=mysqld state=started

- hosts: webservers:dbservers
  sudo: yes

  tasks:
  - name: Stop IPTABLES NOW!!!
    service: name=iptables state=stopped
~
~
~
~
```

```yaml
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
    when: ansible_os_family == "RedHat"

  - name: Start MySQL
    service: name=mysqld state=started

- hosts: webservers:dbservers
  sudo: yes

  tasks:
  - name: Stop IPTABLES NOW!!!
    service: name=iptables state=stopped
~
```

```yaml
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

  - name: Deploy configuration File
    template: src=templates/httpd.j2 d
- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
    when: ansible_os_family == "RedHat"

  - name: Start MySQL
    service: name=mysqld state=started

- hosts: webservers:dbservers
  sudo: yes

  tasks:
  - name: Stop IPTABLES NOW!!!
    service: name=iptables state=stopped
```

```yaml
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

  - name: Deploy configuration File
    template: src=templates/httpd.j2 dest=/etc/httpd/conf/httpd.conf

- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
    when: ansible_os_family == "RedHat"

  - name: Start MySQL
    service: name=mysqld state=started

- hosts: webservers:dbservers
  sudo: yes

  tasks:
  - name: Stop IPTABLES NOW!!!
```

```yaml
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

  - name: Deploy configuration File
    template: src=templates/httpd.j2 dest=/etc/httpd/conf/httpd.conf
    notify:
      - Restart Apache

   handlers:
     - name: Restart Apache
       service: name=httpd state=restarted

- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
    when: ansible_os_family == "RedHat"

  - name: Start MySQL
    service: name=mysqld state=started
```

```yaml
---
- hosts: webservers
  sudo: yes

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

  - name: Deploy configuration File
    template: src=templates/httpd.j2 dest=/etc/httpd/conf/httpd.conf
    notify:
      - Restart Apache

   - name: Copy Site Files
     template: src=templates/index.j2 dest={{ doc_root }}/index.html

   handlers:
     - name: Restart Apache
       service: name=httpd state=restarted

- hosts: dbservers
  sudo: yes

  tasks:
  - name: Ensure MySQL is installed
    yum: name=mysql-server state=present
    when: ansible_os_family == "RedHat"
```

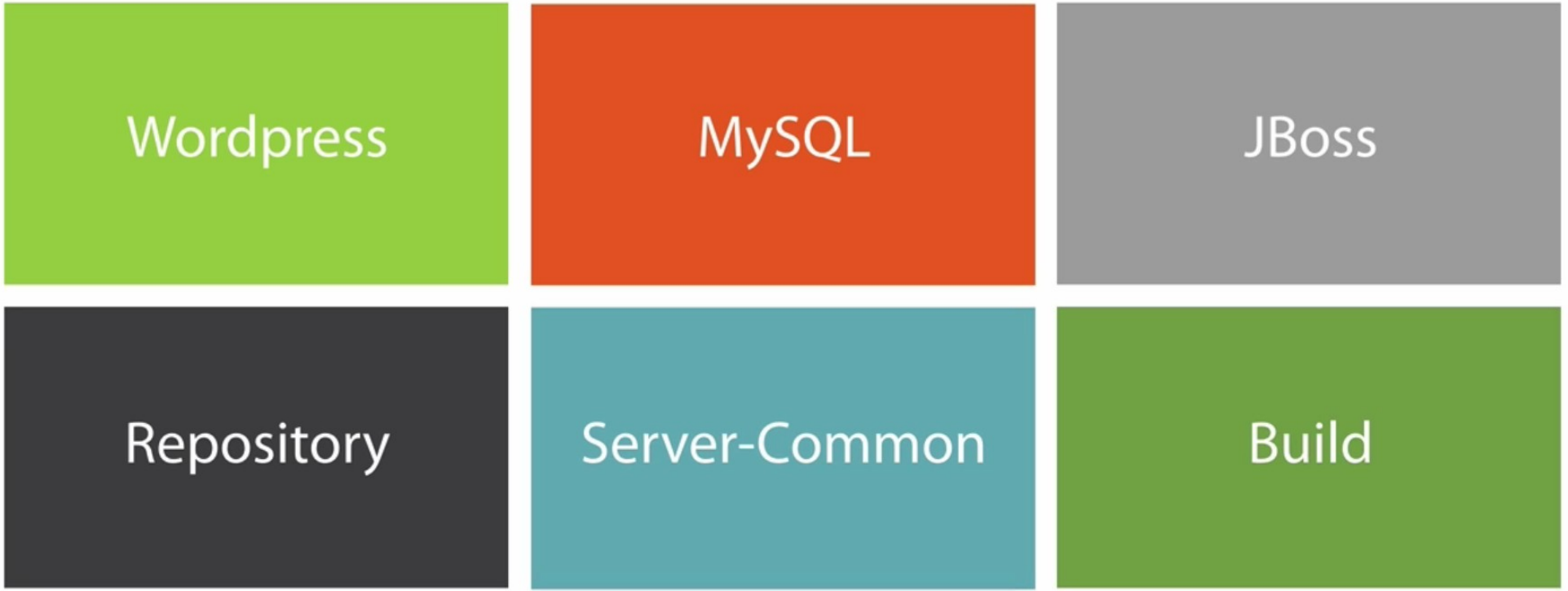
```yaml
---
- hosts: webservers
  sudo: yes
  vars:
    http_port: 80
    doc_dir: /ansible/
    doc_root: /var/www/html/ansible/
    max_clients: 5

  vars_prompt:
    - name: username
      prompt:  What is your name?  What is your Quest?  What is your fav color?
  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

  - name: Deploy configuration File
    template: src=templates/httpd.j2 dest=/etc/httpd/conf/httpd.conf
    notify:
      - Restart Apache

   - name: Copy Site Files
     template: src=templates/index.j2 dest={{ doc_root }}/index.html

   handlers:
     - name: Restart Apache
       service: name=httpd state=restarted
```

```yaml
---
- hosts: webservers
  sudo: yes
  vars:
    http_port: 80
    doc_dir: /ansible/
    doc_root: /var/www/html/ansible/
    max_clients: 5

  vars_prompt:
    - name: username
      prompt:  What is your name?  What is your Quest?  What is your fav color?

  tasks:
  - name: Ensure that Apache is installed
    yum: name=httpd state=present
    when: ansible_os_family == "RedHat"

  - name: Start Apache Services
    service: name=httpd enabled=yes state=started

  - name: Deploy configuration File
    template: src=templates/httpd.j2 dest=/etc/httpd/conf/httpd.conf
    notify:
      - Restart Apache

  - name: Copy Site Files
     template: src=templates/index.j2 dest={{ doc_root }}/index.html

   handlers:
     - name: Restart Apache
```

```
<html>
<head><title>Hello from Ansible!!</title></head>
<body>
<H1>Congratulations!</h1>
<p>Nice job {{ username }}!!  You have successfully ran your first playbook!  Woohooo!  Now get yourself a drink to celebrate!<
/p>
</body>
</html>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
ServerTokens OS
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
Timeout 60
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 15
<IfModule prefork.c>
StartServers       8
MinSpareServers    5
MaxSpareServers   20
ServerLimit      256
MaxClients       256
MaxRequestsPerChild  4000
</IfModule>
<IfModule worker.c>
StartServers         4
# Adding --max_clients-- template variable
MaxClients         {{ max_clients }}
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>
# Adding Ansible --doc_dir-- and --doc_root-- variables
Alias {{ doc_dir }} {{ doc_root }}
# Adding port number --http_port--
Listen {{ http_port }}
LoadModule auth_basic_module modules/mod_auth_basic.so
```

Role Examples



Current Playbook

Current Playbook

ROLE: Builder

- tasks:
  - name: install gcc
  - name: install jdk
  - name: setup git

ROLE: Server-Common

- tasks:
  - name: configure SNMP
  - name: configure SYSLOG
  - name: configure NTP

ROLE: Repo

- tasks:
  - name: git
  - name: configure
  - name: pull latest

Efficient Role Design

BUILD: Compiler/Unit Test Role

Install GCC

Install JDK

Install Unit Testing

REPO: Code Repository Role

Install Git

Configure Git

Schedule hourly pulls

Efficient Role Design

BUILD: Compiler/Unit Test Role

Install GCC

Install Git

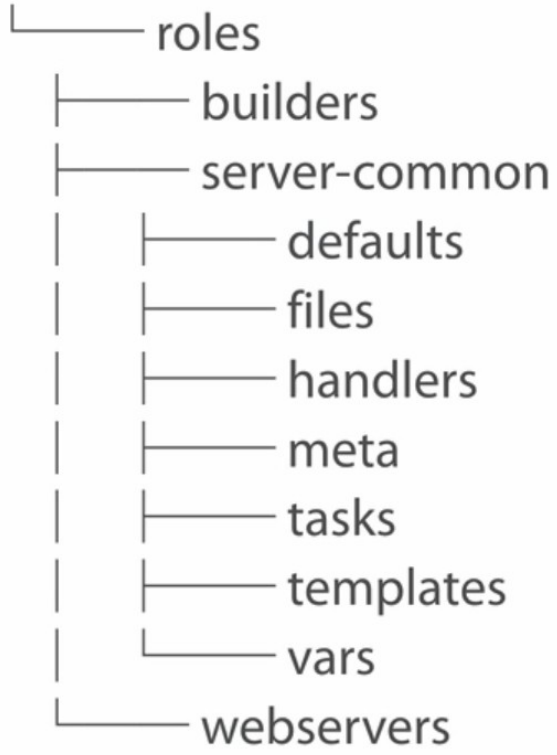
Install JDK

Configure Git

Install Unit Testing

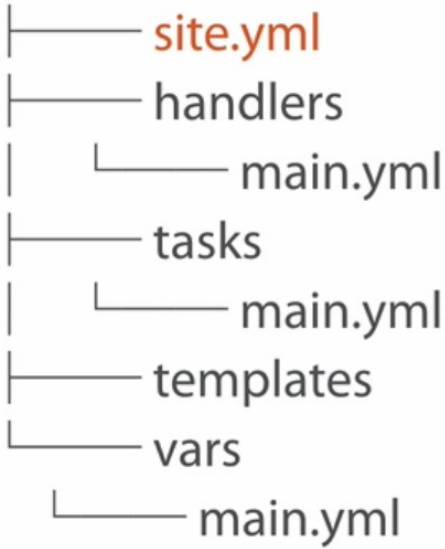
Schedule hourly pulls

Directory Structure



main.yml

Primary file that Ansible looks for



site.yml

Primary file to include entire infrastructure

site.yml

```
---
- include: webservers.yml
- include: dbservers.yml
```

Can add includes to break-up long files

site.yml

```
---
- include: webservers.yml tags=web
- include: dbservers.yml tags=db
```

Use tags to define categories within your playbooks

Tagging Tasks

```
tasks:
  debug: msg="This will only run on tag 'debug'"
  tags:
    - debug

  debug: msg="You can also use multiple tags"
  tags:
    - debug
    - ubercool
```

Adding Roles to Playbook

```
---
- hosts: code-dev
  roles:
    - server-common
    - builders
  gather_facts: no

  tasks:
    # Build your extra tasks here like
    # creating users, or deploying a specific config
```

Pre-tasks and Post-tasks

pre_tasks:

- Executes plays BEFORE roles
  - Use-Cases
    - Setup of maintenance windows
    - Removing servers from Load-balancers
    - Silencing alarms

post_tasks:

- Executes plays AFTER roles
  - Use-Cases
    - Clearing of maintenance windows
    - Adding servers to Load-balancers
    - Enabling Alarms

Adding Pre and Post Tasks

```
---
- hosts: webservers
  pre_tasks:
    - # Remove from load-balancer
  roles:
    - server-common
    - jboss
  post_tasks:
    - # Add to load-balancer
  gather_facts: no
```

Executing Roles - Basic

Basic execution of roles:

```
$ ansible-playbook site.yml
```

Executing Roles - Tags

Tagged execution of roles:

```
$ ansible-playbook site.yml
   —tags “web”
```

Executing Roles - Tags with Limits

Limited tagged execution of roles:

```
$ ansible-playbook site.yml
    –tags “web”
    –limit atlanta
```

Demo: Build Webserver Role

Define “webserver” role

Define “dbserver” role

Define “common-server” role

Apply roles

```
vagrant@acs:~/exercise_7.1$ ls
ansible.cfg  inventory
vagrant@acs:~/exercise_7.1$ mkdir roles
vagrant@acs:~/exercise_7.1$ cd roles
vagrant@acs:~/exercise_7.1/roles$ ls
vagrant@acs:~/exercise_7.1/roles$ mkdir webserver
vagrant@acs:~/exercise_7.1/roles$ cd webserver
vagrant@acs:~/exercise_7.1/roles/webserver$ mkdir vars
vagrant@acs:~/exercise_7.1/roles/webserver$ mkdir tasks
vagrant@acs:~/exercise_7.1/roles/webserver$ mkdir handlers
vagrant@acs:~/exercise_7.1/roles/webserver$ mkdir templates
vagrant@acs:~/exercise_7.1/roles/webserver$ cd ..
vagrant@acs:~/exercise_7.1/roles$ tree
.
└── webserver
    ├── handlers
    ├── tasks
    ├── templates
    └── vars

5 directories, 0 files
vagrant@acs:~/exercise_7.1/roles$
```

```
---
- name: Ensure Apache is installed
  yum: name=httpd state=present

- name: Start Apache
  service: name=httpd state=started enabled=yes

- name: Deploy configuration File
  template: src=httpd.j2 dest=/etc/httpd/conf/httpd.conf
  notify:
  - Restart Apache

- name: Copy Site Files
  template: src=index.j2 dest={{ doc_root }}/index.html
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```yaml
---
http_port: 80
doc_dir: /ansible/
doc_root: /var/www/html/ansible/
max_clients: 5
username: My name
```

```yaml
---
- name: Restart Apache
  service: name=httpd state=restarted
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
|   |   └── snmpd.conf.j2
|   └── vars
|       └── main.yml
└── webserver
    ├── handlers
    |   └── main.yml
    ├── tasks
    |   └── main.yml
    ├── templates
    |   ├── httpd.j2
    |   └── index.j2
    └── vars
        └── main.yml

10 directories, 12 files
vagrant@acs:~/exercise_7.1/roles$ cd webserver
vagrant@acs:~/exercise_7.1/roles/webserver$ tree
.
├── handlers
|   └── main.yml
├── tasks
|   └── main.yml
├── templates
|   ├── httpd.j2
|   └── index.j2
└── vars
    └── main.yml

4 directories, 5 files
vagrant@acs:~/exercise_7.1/roles/webserver$ cd ..
vagrant@acs:~/exercise_7.1/roles$ cd ..
vagrant@acs:~/exercise_7.1$ vim webserver.yml
```

```yaml
---
- hosts: webservers
  sudo: yes
  gather_facts: no
  roles:
  - webserver
```

```
        └── main.yml

10 directories, 12 files
vagrant@acs:~/exercise_7.1/roles$ cd webserver
vagrant@acs:~/exercise_7.1/roles/webserver$ tree
.
├── handlers
|   └── main.yml
├── tasks
|   └── main.yml
├── templates
|   ├── httpd.j2
|   └── index.j2
└── vars
    └── main.yml

4 directories, 5 files
vagrant@acs:~/exercise_7.1/roles/webserver$ cd ..
vagrant@acs:~/exercise_7.1/roles$ cd ..
vagrant@acs:~/exercise_7.1$ vim webserver.yml
vagrant@acs:~/exercise_7.1$ ansible-playbook webserver.yml

PLAY [webservers] ***************************************************************

TASK: [webserver | Ensure Apache is installed] ********************************
ok: [web1]

TASK: [webserver | Start Apache] ***********************************************
ok: [web1]
```

```
web1                       : ok=3    changed=1    unreachable=0    failed=1

vagrant@acs:~/exercise_7.1$ ansible-playbook webserver.yml

PLAY [webservers] ************************************************************

TASK: [webserver | Ensure Apache is installed] *******************************
ok: [web1]

TASK: [webserver | Start Apache] *********************************************
ok: [web1]

TASK: [webserver | Deploy configuration File] ********************************
ok: [web1]

TASK: [webserver | Copy Site Files] ******************************************
changed: [web1]

PLAY RECAP *******************************************************************
web1                       : ok=4    changed=1    unreachable=0    failed=0

vagrant@acs:~/exercise_7.1$ ls
ansible.cfg  inventory  roles  webserver.yml
vagrant@acs:~/exercise_7.1$ vim webserver.yml
vagrant@acs:~/exercise_7.1$ ls
ansible.cfg  inventory  roles  webserver.yml
vagrant@acs:~/exercise_7.1$ cd roles
vagrant@acs:~/exercise_7.1/roles$ mkdir dbserver
vagrant@acs:~/exercise_7.1/roles$ cd dbserver
vagrant@acs:~/exercise_7.1/roles/dbserver$ mkdir tasks
vagrant@acs:~/exercise_7.1/roles/dbserver$ cd tasks
vagrant@acs:~/exercise_7.1/roles/dbserver/tasks$ vim main.
```

```
---
- name: Ensure MySQL installed
  yum: name=mysql-server state=present

- name: Start MySQL
  service: name=mysqld state=started
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
TASK: [webserver | Start Apache] ******************************************
ok: [web1]

TASK: [webserver | Deploy configuration File] *********************************
ok: [web1]

TASK: [webserver | Copy Site Files] ***************************************
changed: [web1]

PLAY RECAP ********************************************************************
web1                       : ok=4    changed=1    unreachable=0    failed=0

vagrant@acs:~/exercise_7.1$ ls
ansible.cfg  inventory  roles  webserver.yml
vagrant@acs:~/exercise_7.1$ vim webserver.yml
vagrant@acs:~/exercise_7.1$ ls
ansible.cfg  inventory  roles  webserver.yml
vagrant@acs:~/exercise_7.1$ cd roles
vagrant@acs:~/exercise_7.1/roles$ mkdir dbserver
vagrant@acs:~/exercise_7.1/roles$ cd dbserver
vagrant@acs:~/exercise_7.1/roles/dbserver$ mkdir tasks
vagrant@acs:~/exercise_7.1/roles/dbserver$ cd tasks
vagrant@acs:~/exercise_7.1/roles/dbserver/tasks$ vim main.yml
vagrant@acs:~/exercise_7.1/roles/dbserver/tasks$ cd ..
vagrant@acs:~/exercise_7.1/roles/dbserver$ cd ..
vagrant@acs:~/exercise_7.1/roles$ ls
dbserver  server-common  webserver
vagrant@acs:~/exercise_7.1/roles$ cd ..
vagrant@acs:~/exercise_7.1$ ls
ansible.cfg  inventory  roles  webserver.yml
vagrant@acs:~/exercise_7.1$ vim dbserver.ym
```

```
---
- hosts: dbservers
  sudo: yes
  roles:
  - dbserver
  gather_facts: no
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```yaml
---
- hosts: webservers:dbservers
  sudo: yes
  gather_facts: no
  roles:
  - server-common

- include: webserver.yml
- include: dbserver.yml
```

Getting Roles

Create your own roles

Perfect for proprietary applications or workflows

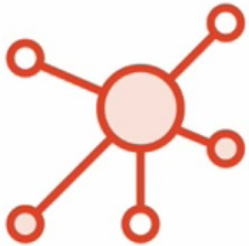
Find roles to download

Look for others that had the same requirement and shared their work

Ansible Galaxy

Download

Share



Review

Installing Galaxy Roles

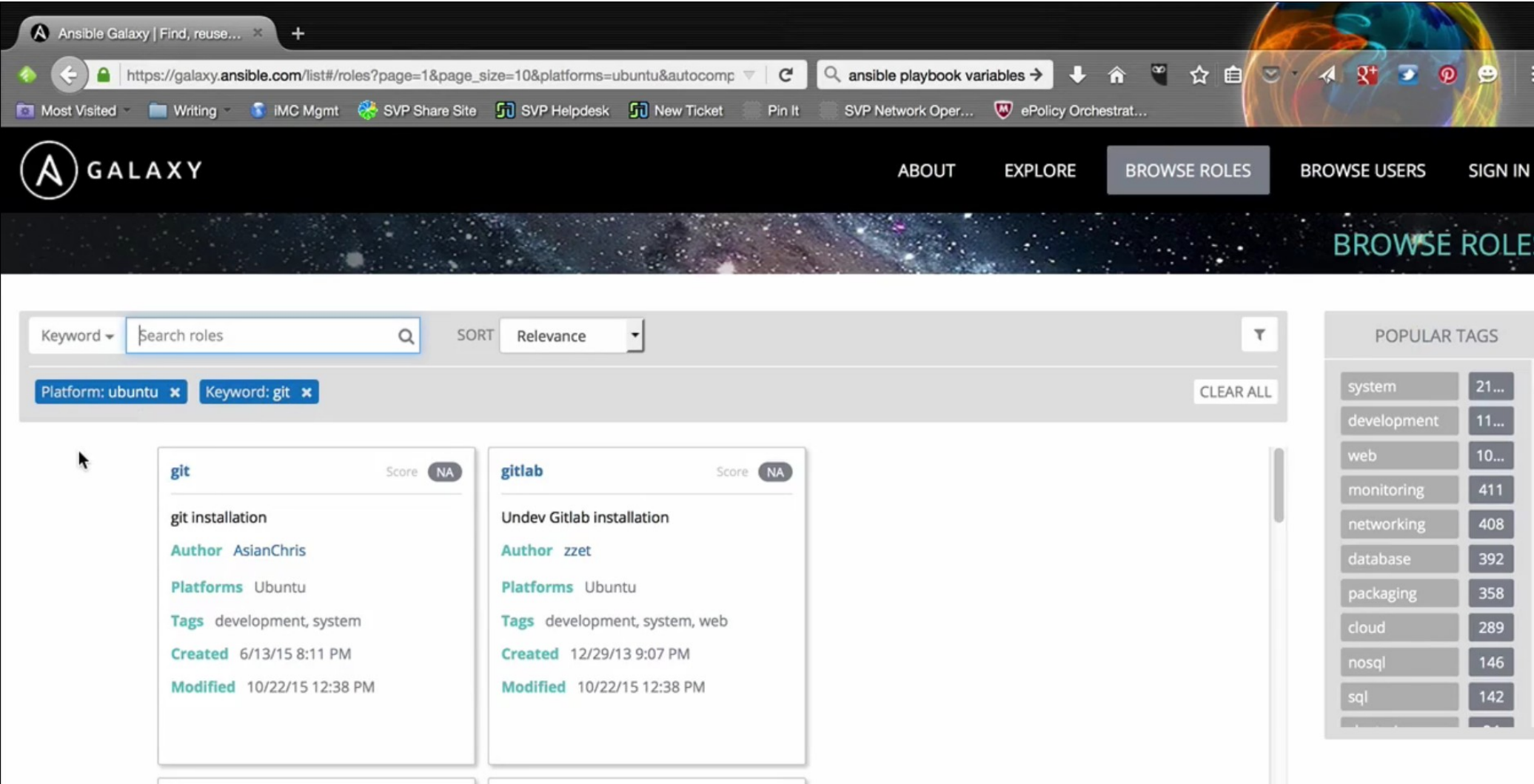
Use username.role

```
$ ansible-galaxy install username.role
```

Demo: Ansible Galaxy

Browse/Search Ansible Galaxy

Find a Role and install


Ansible Galaxy | Find, reuse...
https://galaxy.ansible.com/list#/roles?page=1&page_size=10&platforms=ubuntu&autocomp
ansible playbook variables
Most Visited
Writing
iMC Mgmt
SVP Share Site
SVP Helpdesk
New Ticket
Pin It
SVP Network Oper...
ePolicy Orchestrat...
GALAXY
ABOUT
EXPLORE
BROWSE ROLES
BROWSE USERS
SIGN IN
BROWSE ROLE
Keyword
Search roles
SORT
Relevance
Platform: ubuntu
Keyword: git
CLEAR ALL
POPULAR TAGS
system 21...
development 11...
web 10...
monitoring 411
networking 408
database 392
packaging 358
cloud 289
nosql 146
sql 142
git
Score NA
git installation
Author AsianChris
Platforms Ubuntu
Tags development, system
Created 6/13/15 8:11 PM
Modified 10/22/15 12:38 PM
gitlab
Score NA
Undev Gitlab installation
Author zzet
Platforms Ubuntu
Tags development, system, web
Created 12/29/13 9:07 PM
Modified 10/22/15 12:38 PM

```
vagrant@acs:~/exercise_7.2$ sudo ansible-galaxy install geerlingguy.git
 downloading role 'git', owned by geerlingguy
 no version specified, installing 1.0.5
 - downloading role from https://github.com/geerlingguy/ansible-role-git/archive/1.0.5.tar.gz
 - extracting geerlingguy.git to /etc/ansible/roles/geerlingguy.git
geerlingguy.git was installed successfully
vagrant@acs:~/exercise_7.2$ ls /etc/ansible/roles/geerlingguy.git
defaults  meta  README.md  tasks  tests  vars
vagrant@acs:~/exercise_7.2$ ls /etc/ansible/roles/geerlingguy.git/defaults
main.yml
vagrant@acs:~/exercise_7.2$ vim /etc/ansible/roles/geerlingguy.git/defaults/main.ym
```

```
---
workspace: /root

# If git_install_from_source is set to false, these two variables define whether
# to use an additional repo for the package installation, and which git packages
# will be installed.
git_enablerepo: ""
git_packages:
  - git
  - git-svn

# If set to TRUE, git will be installed from source, using the version set with
# the 'git_version' variable instead of using a package.
git_install_from_source: false
git_install_path: "/usr"
git_version: "2.2.2"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
vagrant@acs:~/exercise_7.2$ sudo ansible-galaxy install geerlingguy.git
 downloading role 'git', owned by geerlingguy
 no version specified, installing 1.0.5
 - downloading role from https://github.com/geerlingguy/ansible-role-git/archive/1.0.5.tar.gz
 - extracting geerlingguy.git to /etc/ansible/roles/geerlingguy.git
geerlingguy.git was installed successfully
vagrant@acs:~/exercise_7.2$ ls /etc/ansible/roles/geerlingguy.git
defaults  meta  README.md  tasks  tests  vars
vagrant@acs:~/exercise_7.2$ ls /etc/ansible/roles/geerlingguy.git/defaults
main.yml
vagrant@acs:~/exercise_7.2$ vim /etc/ansible/roles/geerlingguy.git/defaults/main.yml
vagrant@acs:~/exercise_7.2$ vim git.yml
```

```
---
- hosts: webservers
  sudo: yes
  roles:
   - geerlingguy.git
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
vagrant@acs:~/exercise_7.2$ ansible-playbook git.yml

PLAY [webservers] ***************************************************************

GATHERING FACTS *****************************************************************
ok: [web1]

TASK: [geerlingguy.git | Ensure git is installed (RedHat).] ********************
changed: [web1] => (item=git,git-svn)

TASK: [geerlingguy.git | Ensure git is installed (Debian).] ********************
skipping: [web1]

TASK: [geerlingguy.git | Ensure git's dependencies are installed (RedHat).] ***
skipping: [web1]

TASK: [geerlingguy.git | Ensure git's dependencies are installed (Debian).] ***
skipping: [web1]

TASK: [geerlingguy.git | Download git.] *****************************************
skipping: [web1]

TASK: [geerlingguy.git | Expand git archive.] ***********************************
skipping: [web1]

TASK: [geerlingguy.git | Check if git is installed.] ***************************
skipping: [web1]

TASK: [geerlingguy.git | Build git.] ********************************************
skipping: [web1] => (item=all)
skipping: [web1] => (item=install)
```