Introduction

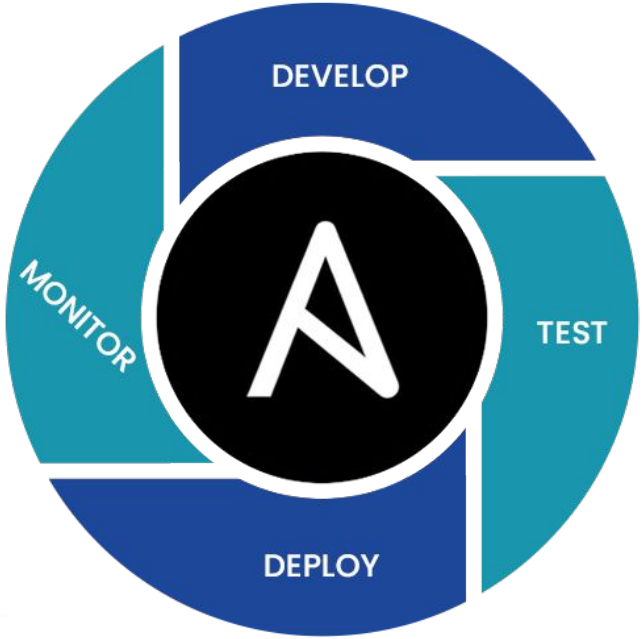
Table of Contents

1

About Ansible

About Ansible

Ansible is **an open-source IT automation tool**. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments.



About Ansible

Scripts

- Time
- Coding Skills
- Maintenance

- **Simple**
- **Powerfull**
- **Agentless**

About Ansible

Scripts

```
#!/bin/bash
# Script to add a user to Linux system
if [ $(id -u) -eq 0 ]; then
   $username=johndoe
   read -s -p "Enter password : " password
   egrep "^$username" /etc/passwd >/dev/null
   if [ $? -eq 0 ]; then
      echo "$username exists!"
      exit 1
   else
      useradd -m -p $password $username
      [ $? -eq 0 ] && echo "User has been added
to system!" || echo "Failed to add a user!"
   fi
fi
```

Playbook

```
- hosts: all_my_web_servers_in_DR
  tasks:
    - user:
        name: johndoe
```

About Ansible

SIMPLE

- Human readable automation
- No special codding skills needed
- Tasks executed in order

Get productive quickly

POWERFULL

- App deployment
- Configuration management
- Workflow orchestration

Orchestrate the app lifecycle

AGENTLESS

- Agentless architecture
- Uses Open SSH
- No agents to exploit or update

More efficient & more secure

2 Installation

Installation

We can install Ansible using **yum** and **apt** package managers.

For install with **yum**:
sudo yum -y install ansible

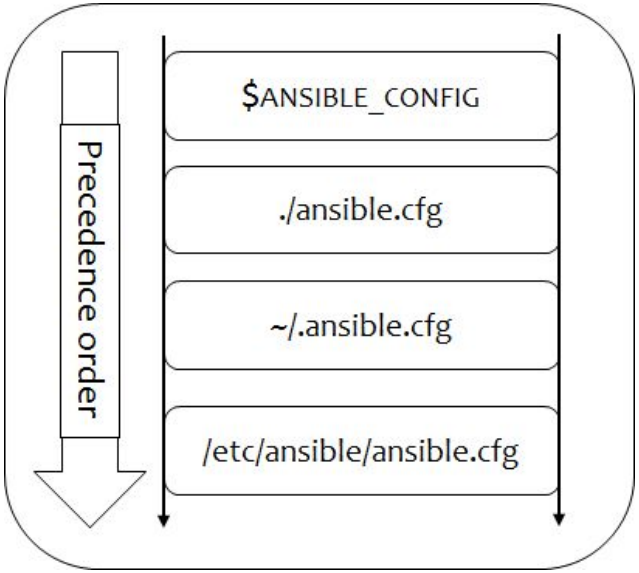
For install with **apt**:
sudo apt-get -y install ansible

3

Configuring Ansible

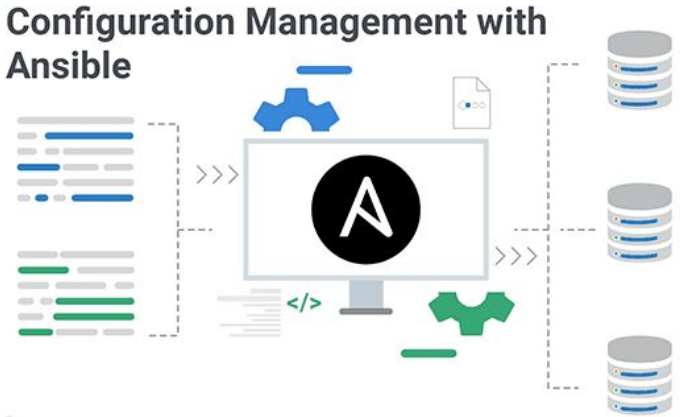
Configuring Ansible

- Ansible supports several sources for configuring its behavior, including an file named ansible.cfg, environment variables, command-line options, playbook keywords, and variables.
- Certain settings in Ansible are adjustable via a configuration file (ansible.cfg).
- Changes can be made and used in a configuration file which will be searched for in the following order:



4

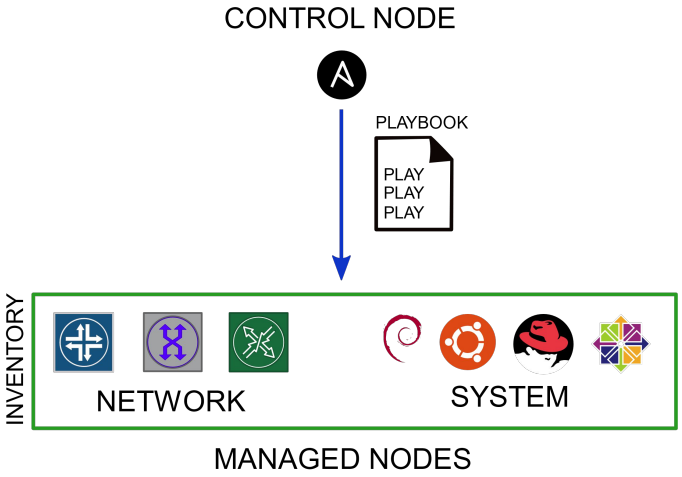
Ansible Concepts



Ansible Concepts

Control node:

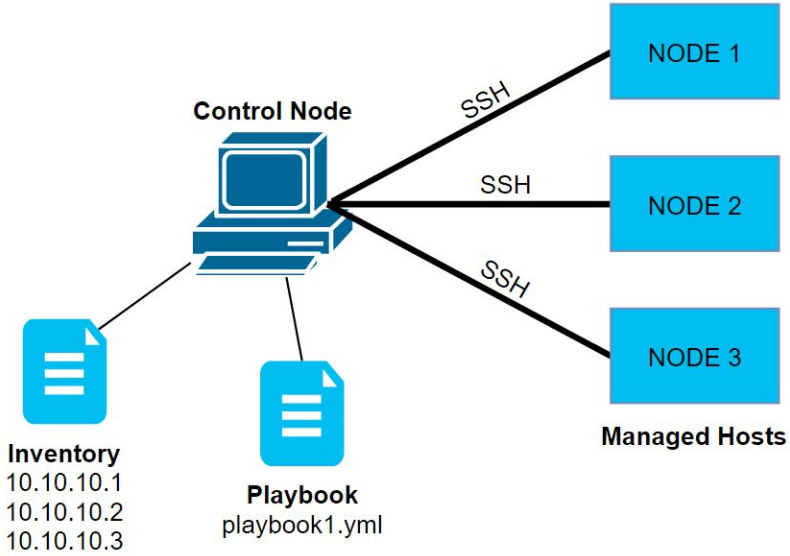
Any machine with Ansible installed. You can run commands and playbooks, invoking /usr/bin/ansible or /usr/bin/ansible-playbook, from any control node. You can use any computer that has Python installed on it as a control node - laptops, shared desktops, and servers can all run Ansible. However, you cannot use a Windows machine as a control node.



Ansible Concepts

Managed Nodes:

The network devices (and/or servers) you manage with Ansible. Managed nodes are also sometimes called **hosts**. Ansible is not installed on managed nodes.



Ansible Concepts

Inventory:

A list of managed nodes. An inventory file is also sometimes called a **hostfile**. Your inventory can specify information like IP address for each managed node. An inventory can also organize managed nodes, creating and nesting groups for easier scaling.

The inventory file

Where it is located

/etc/ansible/hosts

What is the format

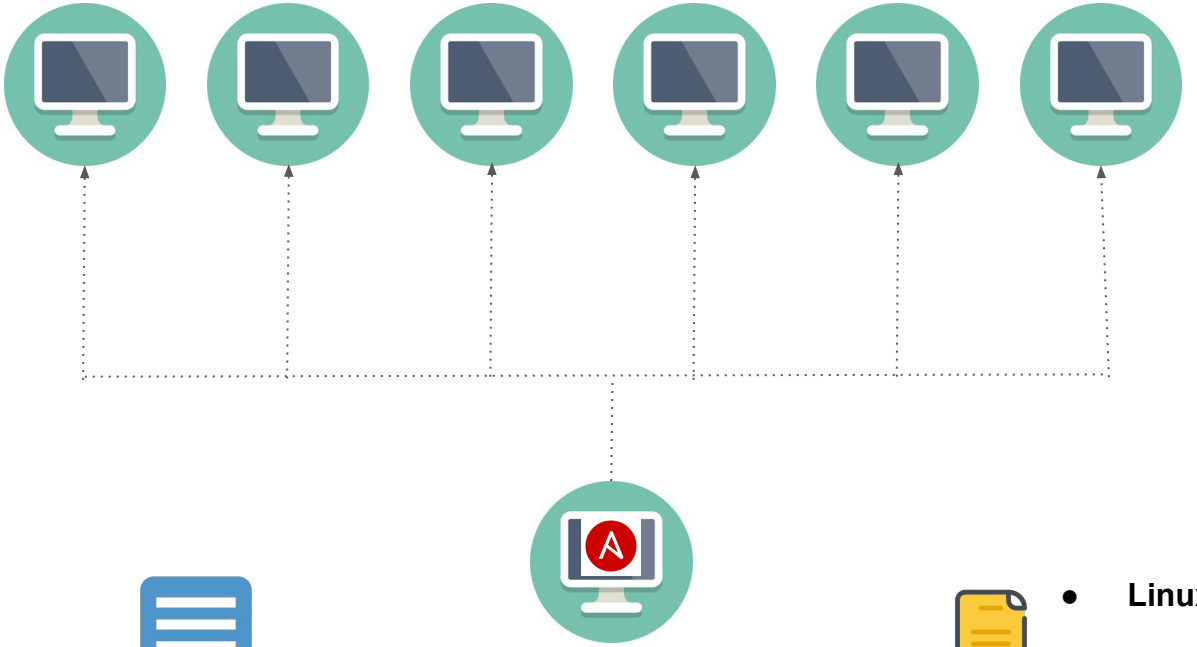
[mailservers]
mail.example.com

[webservers]
foo.example.com ansible_ssh_user = user001
bar.example.com ansible_ssh_private_key_file = /.ssh/ansible_key001

[dbservers]
one.example.com
two.example.com
db-[a:f].example.com

Ansible Concepts

Inventory



```
server1.company.com
server2.company.com

[mail]
server3.company.com
server4.company.com

[db]
server5.company.com
server6.company.com

[web]
server7.company.com
server8.company.com
```

inventory

/etc/ansible/hosts

- **Linux - SSH**

  Windows - Powershell Remoting

- **Agentless**

Ansible Concepts

Group hosts for easier inventory selection and less conditional tasks -- the more groups the better.

WHAT

```
[db]
db[1:4]

[web]
web[1:4]
```

```
db1 = db, east, dev
```

WHERE

```
[east]
db1
web1
db3
web3

[west]
db2
web2
db4
web4
```

WHEN

```
[dev]
db1
web1

[test]
db3
web3

[prod]
db2
web2
db4
web4
```

Ansible Concepts

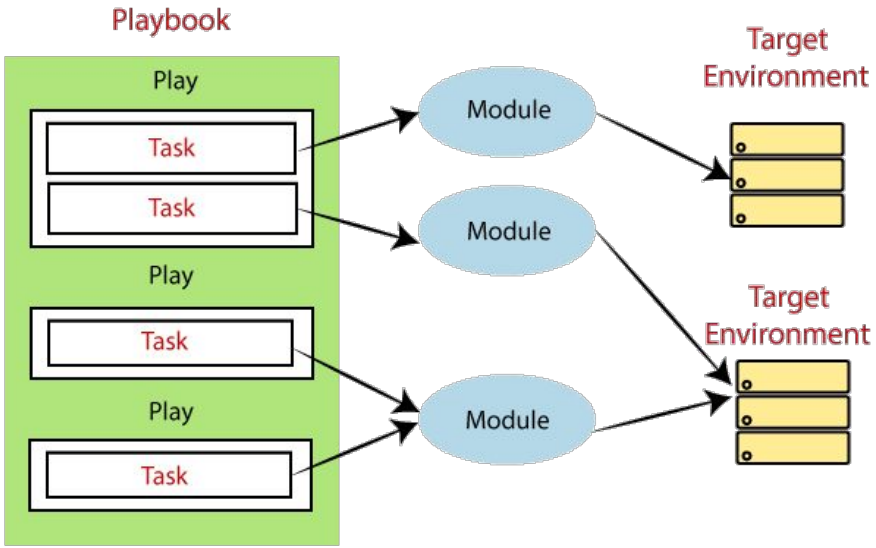
Playbooks:

Ordered lists of tasks, saved so you can run those tasks in that order repeatedly. Playbooks can include variables as well as tasks. Playbooks are written in YAML and are easy to read, write, share and understand.

Ansible Concepts

Tasks:

The units of action in Ansible. You can execute a single task once with an ad-hoc command.



Ansible Concepts

Modules:

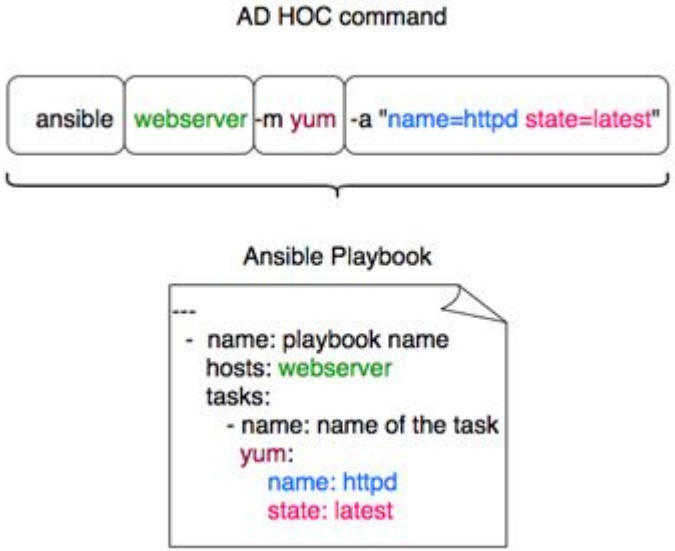
The units of code Ansible executes. Each module has a particular use, from administering users on a specific type of database to managing VLAN interfaces on a specific type of network device.

Modules	Module Categories							
System	User	Group	Iptables	Mount	Ping	Systemd	Service	Hostname
Commands	Command	Expect	Raw	Script	Shell			
Files	Acl	Archive	FInd	Copy	Replace	Stat	File	Unarchive
Database	MySQL	MongoDB	MSSQL	PostgreSQL	ProxySQL	Vertica		
Cloud	Amazon	Azure	Google	Linode	Openstack	VMware	Docker	Atomic
Windows	Win_copy	Win_command	Win_msi	Win_ping	Win_msq	Win_shell	Win_path	Win_service

5 ad-hoc Commands

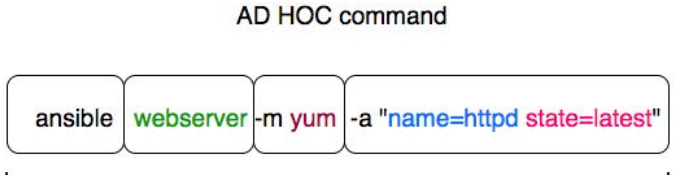
ad-hoc Commands

- An Ansible ad-hoc command uses the /usr/bin/ansible command-line tool to automate a single task on one or more managed nodes.
- Ad-hoc commands are quick and easy, but they are not reusable.
- Ad-hoc commands demonstrate the simplicity and power of Ansible.
- Ad-hoc commands are great for tasks you repeat rarely.



ad-hoc commands

- **ansible** <inventory> -m



Runs a command or **calls a module** directly from the **command line**, no Playbook required

```
ansible <inventory> <options>
ansible web -a /bin/date
ansible web -m ping
ansible web -m yum -a “name=openssl state=latest”
```

THANKS!

Any questions?

You can find me at:

- tyler@clarusway.com
- alex.duncan@clarusway.com