

# MODULE 2: GETTING STARTED WITH ANSIBLE

# Poll Question

---

What is your experience with Ansible?

- A. No experience
- B. Just a little
- C. Very experienced



# Module Overview

---

- Ansible Installation and Inventory
- Ansible Command Line Tools
- Ansible First Steps

# ANSIBLE INSTALLATION & INVENTORY



# Ansible Installation

---

- The control node must be running linux
  - On Windows this means WSL!
- The only other dependency is Python
- Installation is usually one line

```
> pip install ansible
```

- Some platforms might need `sudo apt-get install ansible` or similar
- You can check your Ansible install with

```
> ansible --version
```

# Inventory File

---

- The list of servers you have to manage is called your **inventory**.
- You specify your inventory in an **inventory file**
  - This matches servers (IP addresses or domain names) to groups.

```
[web]  
Websrv1.mydomain.com  
Websrv2.mydomain.com
```

```
[db]  
11.22.33.44  
44.55.66.77
```

# Inventory Conditions

---

- By default servers are connected by SSH
- For everything to work seamlessly you need you SSH keys configured properly
  - SSH/Ansible does allow other access modes, but this is simple and secure.
- Our servers are setup as follows:
  - ✓ Port 22 access enabled (SSH)
  - ✓ A login (ec2-user) that is in the sudoers group
  - ✓ SSH Key-based authentication based on a key in the .ssh directory of your command node home
  - ✓ Added to the known\_hosts file in .ssh directory so access is without confirmation
  - ✓ This means `ssh -A ec2-user@managed-node-ip` will log you in immediately

# Inventory Management

---

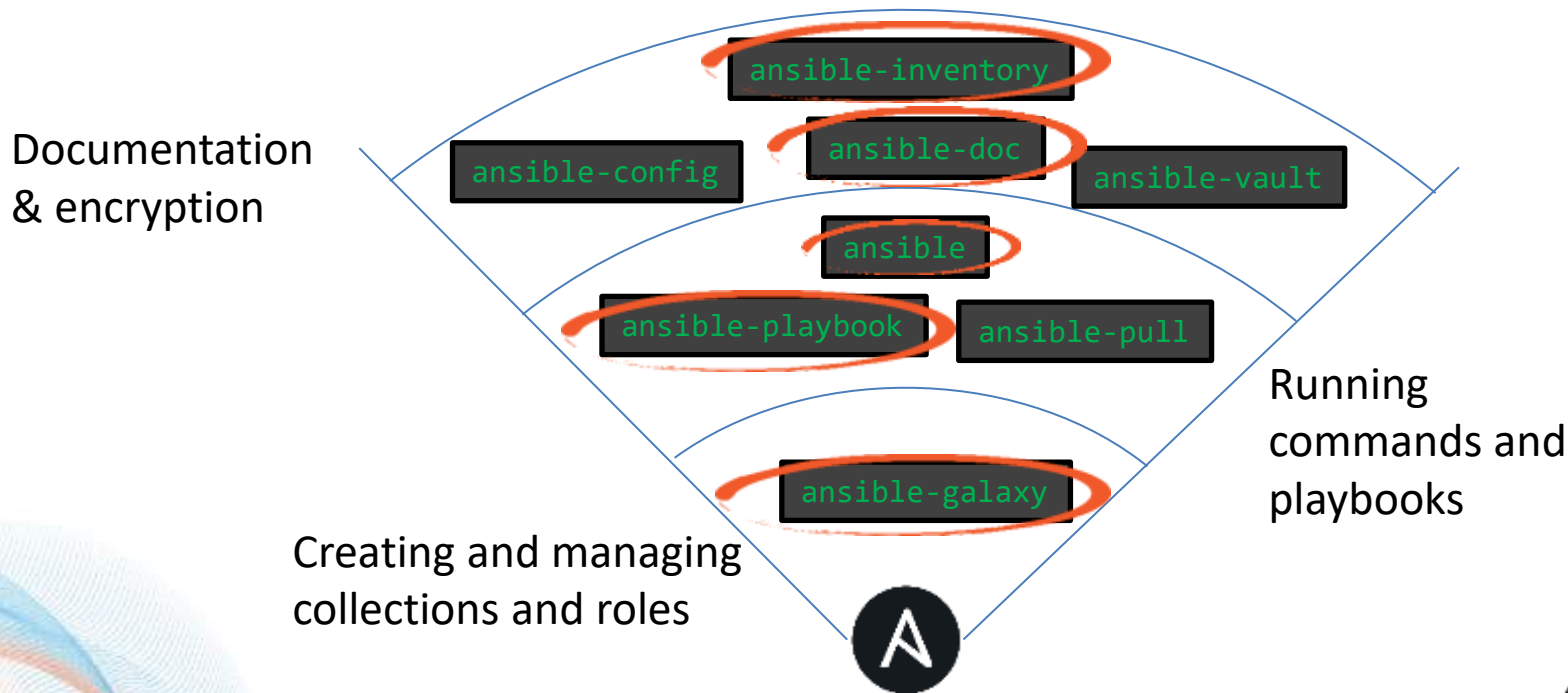
- Larger installations can have quite complex inventories
  - Dev, qa, prod servers
  - Many server types
  - Networking devices
- Inventory in modern applications is often dynamic
  - Auto-scaling, load balancing
  - Containerized applications
- The large cloud providers all have Ansible plugins to assist with these issues
  - Dynamic inventory provisioning and management
- Some can even run Ansible playbooks natively
  - [AWS Systems Manager](#) can apply an ansible playbook to a specified server grouping



# ANSIBLE COMMAND LINE TOOLS



# Command-line tools



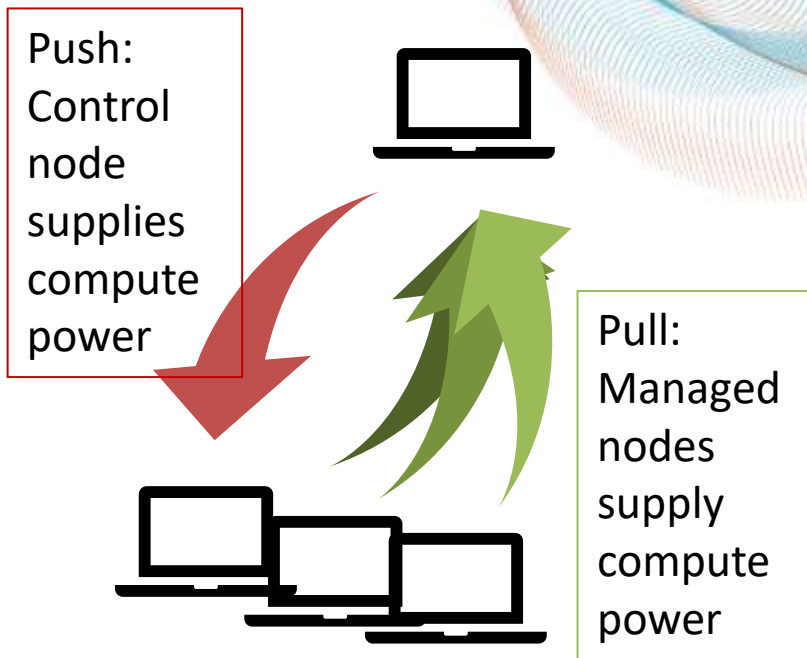
# Documentation and Encryption

---

- These are in your first lab
  - `ansible-inventory`: Inventory information
  - `ansible-doc`: Documentation for available modules
- `ansible-config`: View configuration information
- `ansible-vault`: Encryption for playbooks, inventory files etc

# Running commands and playbooks

- `ansible`: ad-hoc commands.
- `ansible-playbook`: for running playbooks.
- `ansible-pull`:
  - Ansible uses a “push” mode by default
  - Data is packaged on the control node and pushed to the managed nodes
  - `ansible-pull` reverses this by
    - pulling a remote copy of ansible on each managed node
    - set the managed nodes to run playbooks from a source repository



# ansible-galaxy

---

- We will encounter collections and roles later on today.
- `ansible-galaxy` has subcommands to create and manage both roles and collections.

# ANSIBLE FIRST STEPS



# LAB-1

---

- Follow the instructions in the first lab

# Services

---

- In the lab you installed some software, but what about a service?

```
$ ansible all -b -m yum -a "name=chrony state=present"
```

- This command installs the software, but the service is not yet running

```
$ ansible all -b -m service -a "name=chronyd state=started enabled=yes"
```

- Now we can run the chronyc command to see how synchronized our servers are

```
$ ansible all -b -a "chronyc tracking"
```

- The service can be stopped the same way:

```
$ ansible all -b -m service -a "name=chronyd state=stopped"
```



# Configuring a database

```
$ansible db -b -m yum -a "name=mariadb-server state=present"
$ansible db -b -m service -a "name=mariadb state=started
enabled=yes"
$ansible db -b -m yum -a "name=firewalld state=present"
$ansible db -b -m service -a "name=firewalld state=started
enabled=true"
$ansible db -b -m firewalld -a "zone=database state=present
permanent=yes"
$ansible db -b -m firewalld -a "source=18.130.36.125
zone=database state=enabled permanent=yes"
$ansible db -b -m firewalld -a "port=3306/tcp zone=database
state=enabled permanent=yes"
```

# User Management

---

- Create a user group on your inventory:

```
$ ansible app -b -m group -a "name=admin state=present"
```

- Add a user to the inventory group:

```
$ ansible app -b -m user -a "name=eccles group=admin state=present"
```

- And you can delete a user just as easily:

```
$ ansible app -b -m user -a "name=eccles group=admin state=absent"
```

# More about Files and Directories

---

- You can get information about a file using the stat module

```
$ ansible all -m stat -a "path=/home/ec2-user/vitalstatistix"
```

- Remember how copy let you place files? Well, fetch will retrieve them!

```
$ ansible app -m fetch -a "src=/etc/hosts dest=./app-hosts.txt"
```

- The file module lets you create and delete files, directories and symlinks

```
$ ansible app -m file -a "dest=/tmp/test mode=664 state=directory"  
$ ansible app -m file -a "dest=/tmp/test state=absent"
```

# Cron jobs

---

- The cron package allows you to run cron jobs on your inventory:

```
cat <<-EOF > cron-script.sh
cowsay "are we there yet?"
EOF
```

```
ansible web -m yum -a "name=cowsay state=latest" -b
ansible web -b -m cron -a "name='annoying-cow' minute=*
job='./cron-script.sh'"
ansible web -b -m cron -a "name='annoying-cow' state=absent"
```

# REVIEW



# Module Review

---

In this module you learned about:

- ✓ Ansible Installation and Inventory
- ✓ Ansible Command Line Tools
- ✓ Ansible First Steps

Next we will do a short quiz

- Knowledge Check

# KNOWLEDGE CHECK



# Knowledge check question 1

Which ports needs to be open for a Server to be managed by Ansible using SSH?

Choice	Response
A	80, 443, 22
B	80, 443
C	22 only



# Knowledge check question 1

Which ports needs to be open for a Server to be managed by Ansible using SSH?

Choice	Response
A	80, 443, 22
B	80, 443
C	22 only

## Knowledge check question 2

You want to know how to use a new module which tool will help you the most?

Choice	Response
A	<code>ansible-galaxy</code>
B	<code>ansible-doc</code>
C	<code>ansible</code>
D	<code>ansible-help</code>

## Knowledge check question 2

You want to know how to use a new module which tool will help you the most?

Choice	Response
A	<code>ansible-galaxy</code>
B	<code>ansible-doc</code>
C	<code>ansible</code>
D	<code>ansible-help</code>

## Knowledge check question 3

What arguments allow ad-hoc commands to execute with elevated privileges?

Choice	Response
A	--sudo
B	--become
C	--root
D	-b

## Knowledge check question 3

What arguments allow ad-hoc commands to execute with elevated privileges?

Choice	Response
A	--sudo
B	--become
C	--root
D	-b