



# Automation with Ansible



openstack®



nectar



- Deploying complex cloud systems involves many moving parts
  - The capability of managing large number of systems
  - Easy to forget some of the details in configuring the systems
  - Manual process is always error-prone
  - Snapshots are monolithic, provides no record of what has changed
- Benefits of automation
  - Provides a record of what you did
  - Codifies knowledge about the system
  - Makes process repeatable
  - Makes it programmable – “Infrastructure as Code”



# Classification of Scripting tools

---

- Cloud-focused

Used to interact with Cloud services.

- Apache JClouds (Java-based - supports multiple clouds)
- Boto (Python - supports AWS and OpenStack)
- OpenStackClient (Python - supports OpenStack)
- CloudFormation (YAML/JSON - supports AWS, OpenStack Heat)

- Shell scripts

- Bash
- Perl



# Classification of Scripting Tools

---

- Configuration management (CM) tools

Configuration management refers to the process of ***systematically*** handling ***changes*** to a system in a way that it ***maintains integrity*** over time.

Automation is the mechanism used to make servers reach a desirable state, previously defined by provisioning scripts using tool-specific languages and features.

- Chef (uses Ruby for creating cookbooks)
- Puppet (uses its own configuration language)
- **Ansible** (use YAML to express playbooks)
- Fabric (Python library that uses SSH for application deployment and administration tasks)
- Terraform, SaltStack, Docker, ...



[What is configuration Management? Everything you need to know.](#)



- An automation tool for configuring and managing computers. Finer grained set up and configuration of software packages.
- Initial release: Feb. 2012
- Combines multi-node software deployment
- Ad-hoc task execution and configuration management

Configuring thousands of machines manually!?





# Ansible: Features

---

- Easy to learn
  - Playbooks in YAML, templates in Jinja2 etc.
  - Sequential execution
- Minimal requirements
  - No need for centralized management servers/daemons
  - Single command to install (*pip install ansible*)
  - Uses SSH to connect to target machine
- Idempotent (repeatable)
  - Executing N times no different to executing once
  - Prevents side-effects from re-running scripts
- Extensible

Write your own modules



# Ansible: More Features

---

- Supports push or pull
  - Push by default but can use cron job to make it pull
- Rolling updates
  - Useful for continuous deployment / zero downtime deployment
- Inventory management
  - Dynamic inventory from external data sources
  - Execute tasks against host patterns
- *Ansible Vault* for encrypted data

*\$ ansible-vault create demo.yaml*

*\$ ansible-vault decrypt demo.yaml*

*\$ ansible-vault encrypt demo.yaml*

*\$ ansible-vault rekey demo.yaml*



# Ansible: More Features

---

- Ad-hoc commands

Execute a one-off command against your inventory

```
$ ansible -i inventory_file -u ubuntu -m shell -a "reboot"
```

- Ansible Galaxy (<https://galaxy.ansible.com/>)

- Ansible Tower: Enterprise mission control for Ansible  
Dashboard, System Tracker, etc.



[What is Ansible and what can it automate?](#)





# Ansible: YAML & Jinja2

---

- Ansible Playbooks are expressed in YAML.
  - YAML: **Y**AML **A**in't **M**arkup **L**anguage
  - YAML is a human friendly data serialization standard for all programming languages.
  - YAML Syntax:  
[https://docs.ansible.com/ansible/latest/reference\\_appendices/YAMLSyntax.html](https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html)
- Ansible uses Jinja2 templating for dynamic expression
  - Jinja2 is a modern and designer-friendly templating language for Python, modelled after Django's templates.
  - Jinja2 introduction:  
[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_templating.html#templating-jinja2](https://docs.ansible.com/ansible/latest/user_guide/playbooks_templating.html#templating-jinja2)



# Ansible: Installation Guide

---

- Linux (Ubuntu)

```
$ sudo apt-get update && sudo apt-get install software-properties-common  
$ sudo apt-add-repository --yes --update ppa:ansible/ansible  
$ sudo apt-get install ansible
```

- macOS

- Brew (<https://brew.sh/>)

```
$ brew install ansible
```

- Pip

```
$ sudo pip install ansible
```



# Ansible: Installation Guide

---

- Windows 10/11 (WSL)
  - Install Windows Subsystem for Linux  
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>
  - Install Ansible  
[See guide for Linux \(Ubuntu\)](#)
  - Tutorial  
<https://www.youtube.com/watch?v=4sMFybv74Uo>
- Ansible guide:  
[https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)



# Ansible: Structure

- Ansible scripts are called *playbooks*, written as simple YAML files
- Structured in a simple folder hierarchy

*Playbook folder*

```
| - variables  
|   | _vars.yaml  
| - inventory  
|   | _inventory.ini  
| - roles  
|   | - defaults  
|   |   | _main.yaml  
|   | - tasks  
|   |   | - task1.yaml  
|   |   | _task2.yaml  
|   | _templates / files  
| _playbook.yaml
```

```
[webservers]  
foo.example.com  
128.250.0.1
```

```
[dbservers]  
one.example.com  
two.example.com
```



# Ansible: Playbooks

- Executed sequentially from a YAML file

- *hosts*: *webservers*

*vars*:

*package*: ['httpd', 'python-dev']

*tasks*:

- *name*: *Ensure the latest Apache and Python are installed*

*ansible.builtin.apt*:

*name*: "{{ package }}"

*state*: *latest*

- *name*: *Write the Apache config file*

*ansible.builtin.file*:

*src*: */srv/httpd.conf*

*dest*: */etc/httpd.conf*

- *name*: *Ensure Apache is restarted*

*ansible.builtin.service*:

*name*: *httpd*

*state*: *restarted*

{ *[webservers]*  
*www[01:50].example.com*  
*192.168.0.[1:254]* }



# Ansible Demo: Melbourne Research Cloud

---

## Ansible Demo 1:

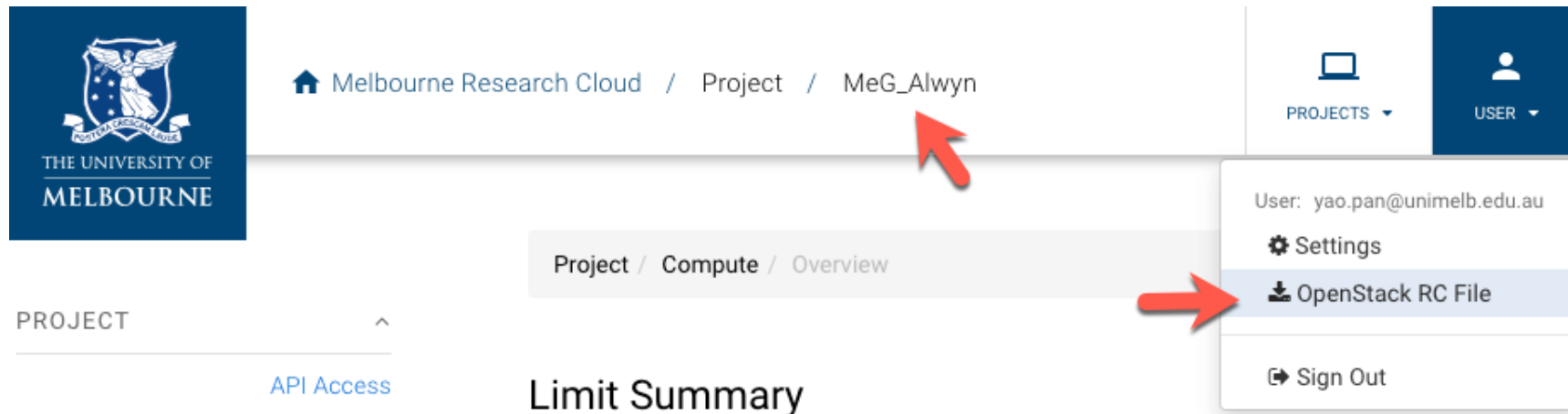
- Connect to Melbourne Research Cloud
- List all images
- Create volumes
- Create security groups with security rules
- Launch an instance and attach the volume and security groups
- Create snapshots of the volumes



# Ansible Demo: Melbourne Research Cloud

- Prerequisites:

1. Login to <https://dashboard.cloud.unimelb.edu.au>
2. Download *openrc.sh* from Dashboard
  - Make sure the correct project is selected
  - Download the OpenStack RC File





# Ansible Demo: Melbourne Research Cloud

- Prerequisites:

- 3. Reset API password

Dashboard -> User -> Settings -> Reset Password

The screenshot shows the Melbourne Research Cloud dashboard. The top navigation bar includes the University of Melbourne logo, the breadcrumb "Melbourne Research Cloud / Project / MeG\_Alwyn", and a user profile dropdown menu. The user profile menu is open, showing the user "yao.pan@unimelb.edu.au" and options for "Settings", "OpenStack RC File", and "Sign Out". A red arrow labeled "1" points to the "USER" dropdown. A red arrow labeled "2" points to the "Settings" option in the dropdown. On the left sidebar, the "SETTINGS" section is expanded, showing "User Settings" and "Reset Password". A red arrow labeled "3" points to the "Reset Password" link. The main content area displays the "Password Reset Form" with a "Description:" section. A red arrow labeled "4" points to the "Reset Password" button at the bottom right of the form.

Melbourne Research Cloud / Project / MeG\_Alwyn

PROJECTS USER SUPPORT

User: yao.pan@unimelb.edu.au

- Settings
- OpenStack RC File
- Sign Out

Settings / None

PROJECT

SETTINGS

User Settings

Reset Password

APP CATALOG

ALLOCATIONS

### Password Reset Form

Description:

To access the Melbourne Research Cloud using the OpenStack API you will need to generate a password. By doing so **any existing password will be forgotten**. To generate a new password click "Reset Password" button.

Reset Password





# Ansible Demo: Melbourne Research Cloud

- Prerequisites:

- Instance Flavor: *uom.general.1c4g*
- Availability Zone: *melbourne-qh2-uom*
- Image Id: *bbaf16f8-2484-48d7-b455-42209cc4b8d2*
- Dashboard -> Compute -> Images

PROJECT

- API Access
- Compute
- Overview
- Instances
- Images**
- Key Pairs
- Server Groups
- Volumes
- Network
- Orchestration
- Database
- DNS
- Object Store

APP CATALOG

ALLOCATIONS

Search: **ubuntu 22.04** [x] [Create Image] [Delete Images]

Displaying 3 items

<input type="checkbox"/>	Name ^	Type	Status	Visibility	Protected	Disk Format	Size	
<input type="checkbox"/>	<b>NeCTAR Ubuntu 22.04 LTS (Jammy) amd64</b>	Image	Active	Public	No	QCOW2	1.30 GB	Launch [v]
<b>Name</b> NeCTAR Ubuntu 22.04 LTS (Jammy) amd64		<b>Visibility</b> Public Protected		<b>Min. Disk</b> 0				
<b>ID</b> <b>bbaf16f8-2484-48d7-b455-42209cc4b8d2</b>				<b>Min. RAM</b> 0				
libvirt Driver Options for Images		QEMU Guest Agent True						
<b>Detail Information</b> The libvirt Compute Driver Options for Glance Images. These are properties specific to compute drivers. For a list of all hypervisors, see here: <a href="https://wiki.openstack.org/wiki/HypervisorSupportMatrix">https://wiki.openstack.org/wiki/HypervisorSupportMatrix</a> .								
<input type="checkbox"/>	<b>&gt; NeCTAR Ubuntu 22.04 LTS (Jammy) amd64 (NVIDIA vGPU)</b>	Image	Active	Public	No	QCOW2	5.92 GB	Launch [v]
<input type="checkbox"/>	<b>&gt; NeCTAR Ubuntu 22.04 LTS (Jammy) amd64 (with Docker)</b>	Image	Active	Public	No	QCOW2	1.44 GB	Launch [v]

**DO NOT USE THIS ONE**



# Ansible Demo: Melbourne Research Cloud

---

- *playbook* to interact with Melbourne Research Cloud
  - List all images  
[openstack.cloud.image\\_info](#)
  - Create a volume  
[openstack.cloud.volume](#)
  - Create a security group with rules  
[openstack.cloud.security\\_group](#), [openstack.cloud.security\\_group\\_rule](#)
  - Launch an instance and attach the volume and security group  
[openstack.cloud.server](#)
  - Create a snapshot of the volume  
[openstack.cloud.volume\\_snapshot](#)
  - Other useful Ansible modules  
[ansible.builtin.apt](#), [ansible.builtin.pip](#), [ansible.builtin.set\\_fact](#)  
[ansible.builtin.debug](#), [ansible.builtin.wait\\_for](#), [ansible.builtin.add\\_host](#)  
[ansible.builtin.gather\\_facts](#)  
[become](#), [register](#), [loop](#), [when](#)

Run *playbook*:

```
$ ./openrc.sh; ansible-playbook [--ask-become-pass] mrc.yaml
```



# Ansible Demo: Quiz

---

- Retrieve facts about the instance created
  - Create an instance and retrieve facts about that instance
- Attach existing volume(s) to an existing instance
  - Create a new volume
  - Attach the new volume to an existing instance

*Hint:*

- *Go to the Ansible Documentation and search docs for “openstack”*
- *Find relevant Ansible module*

- Add / remove existing Security Group to existing instance
  - Create a new Security Group
  - Add Security Group Rules
  - Attach the Security Group to an existing instance

*Hint: openstack.cloud.server*



# Ansible Demo: WordPress

---

- Simple *playbook* to deploy a *WordPress* instance with *Docker*\*
  - Install dependencies  
[\*ansible.builtin.apt\*](#), [\*ansible.builtin.pip\*](#)
  - Mount volumes and make filesystems  
[\*community.general.filesystem\*](#), [\*ansible.builtin.stat\*](#)  
[\*ansible.builtin.file\*](#), [\*ansible.posix.mount\*](#)
  - Install *Docker* on one of the volumes  
[\*ansible.builtin.apt\\_key\*](#), [\*ansible.builtin.apt\\_repository\*](#)
  - Deploy a *WordPress* instance with *Docker*  
[\*ansible.builtin.template\*](#), [\*community.docker.docker\\_compose\*](#)



\* *Docker will be introduced in the coming workshops*



- Hwang, Dongarra & Fox, 2011. Distributed and Cloud Computing, 1st Edition. Elsevier.
- Armbrust et al., 2010. A view of cloud computing. Communications of the ACM 53, 50-58. <http://doi.acm.org/10.1145/1721654.1721672>
- Revolution Not Evolution: How Cloud Computing Differs from Traditional IT and Why it Matters  
[http://www.rackspace.com/knowledge\\_center/whitepaper/revolution-not-evolution-how-cloud-computing-differs-from-traditional-it-and-why-it](http://www.rackspace.com/knowledge_center/whitepaper/revolution-not-evolution-how-cloud-computing-differs-from-traditional-it-and-why-it)
- The 10 Most Important Companies In Cloud Computing  
<http://www.businessinsider.com.au/10-most-important-in-cloud-computing-2013-4?op=1#a-word-about-clouds-1>
- Ansible, <https://www.ansible.com/>

Q & A

# Question 1

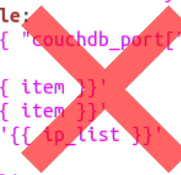
*I am trying to set up a CouchDB cluster, and in order to do that I'd have to modify my security group setting to open up the ports that CouchDB nodes need to use for communication. ... I don't really feel like using a nested loop, ... Just wondering if there is any other way of doing this?*

*Server A:* open 5984, 5986, 4369, 9100 – 9200 to *Server B* and *Server C*

*Server B:* open 5984, 5986, 4369, 9100 – 9200 to *Server A* and *Server C*

*Server C:* open 5984, 5986, 4369, 9100 – 9200 to *Server A* and *Server B*

```
- name: Create rules for each security group
os_security_group_rule:
  security_group: '{{ couchdb_port[" + item|string + "]" }}'
  protocol: "tcp"
  port_range_min: '{{ item }}'
  port_range_max: '{{ item }}'
  remote_ip_prefix: '{{ ip_list }}'
  state: present
  loop: '{{ db_ports }}
```



**Problem:** *remote\_ip\_prefix* only takes a single CIDR notation.

**Solution?** A nested loop with “with\_nested”? Try and tell me if it works ;p

**Solution!** Use the security group as the source of the remote in a security rule.

# Question 1

## Add Rule

Rule \*

Custom TCP Rule

Description ?

Direction

Ingress

Open Port \*

Port Range

From Port \*

9100

To Port \*

9200

Remote ?

Security Group

Security Group

Demo\_Q1 (current)

Ether Type

IPv4

### Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

**Rule:** You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

**Open Port/Port Range:** For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

**Remote:** You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add





## Question 2

**Q:** Random errors, even when run simple “hello world” playbook or the playbooks used in demo.

```
TASK [wordpress : Run docker compose] *****
fatal: [172.26.38.146]: FAILED! => {"changed": false, "msg": "Cannot have both the docker-py and docker python modules installed together as they use the same namespace and cause a corrupt installation. Please uninstall both packages, and re-install only the docker-py or docker python module. It is recommended to install the docker module if no support for Python 2.6 is required. Please note that simply uninstalling one of the modules can leave the other module in a broken state."}
```

**A:** Check Ansible version and make sure the latest Ansible has been installed.

*ansible --version*

## Question 3

**Q:** Cannot SSH to the server just created on NeCTAR / MRC


```
- name: Wait for connection
  wait_for:
    host: "{{ item.openstack.public_v4 }}"
    port: 22
    timeout: 120
    search_regex: OpenSSH
  loop: '{{ os_instance.results }}'
  when: item.openstack is defined
```


**A:** The servers are not ready. Use [wait\\_for](#) module to wait until the servers are ready.


## Question 4

**Q:** I created an instance on MRC and the status showing “Error”

Instance Name	Image Name	IP Address	Flavour	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
					Error	melbourne- qh2-uom	None	Running	1 week, 1 day	Create Snapshot
					Active	melbourne- qh2-uom	None	Running	1 month	Create Snapshot
						melbourne-				

  
Terms and  
Conditions

  
Contact Research  
Computing Services

  
Submit  
Ticket

**A:** Welcome to the FREE cloud. It happens, try to delete the instance if you can. Otherwise Submit a Ticket and the MRC staff will help you out.

# Question 5

**Q:** How to check the host key fingerprint?

[Overview](#) [Interfaces](#) [Log](#) [Console](#) [Action Log](#)

Log Length 35 [Go](#) [View Full Log](#)

### Instance Console Log

```
[ 21.135514] cloud-init[684]: Cloud-init v. 19.4-33-gbb4131a2-0ubuntu1~18.04.1 running 'modules:config' at Tue, 31 Mar 2020 01:44:50.694 - stages.py[WARNING]: Could not find module named cc_snapshots
TASK [Install Python2] *****
The authenticity of host '21.135.514' can't be established.
ECDSA key fingerprint is SHA256:RlCrnUF2thrMw+NYfSAJc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 
```

```
[ 23.063115] cloud-init[760]: 2020-03-31 01:44:50.694 - stages.py[WARNING]: Could not find module named cc_snapshots
ci-info: +-----+
ci-info: | Keytype | Fingerprint (md5) | Options | Comment |
ci-info: +-----+
ci-info: | ssh-rsa | [REDACTED] | [REDACTED] | [REDACTED] |
ci-info: +-----+
<14>Mar 31 01:44:50 ec2:
<14>Mar 31 01:44:50 ec2: #####
<14>Mar 31 01:44:50 ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Mar 31 01:44:50 ec2: 1024 SHA256:QgTUdoMcU1of2 root@cv19 (DSA)
<14>Mar 31 01:44:50 ec2: 256 SHA256:RlCrnUF2thrMw+ root@cv19 (ECDSA)
<14>Mar 31 01:44:50 ec2: 256 SHA256:1ujs9nMr8OWkNV root@cv19 (ED25519)
<14>Mar 31 01:44:50 ec2: 2048 SHA256:d/Aaio4N+al/s root@cv19 (RSA)
<14>Mar 31 01:44:50 ec2: -----END SSH HOST KEY FINGERPRINTS-----
```

## Question 6

**Q:** Ansible keeps prompting host key confirmation, sometimes type “yes” only works for the first host.

```
TASK [Install Python2] *****
The authenticity of host [redacted] can't be established.
ECDSA key fingerprint is [redacted]
Are you sure you want to continue connecting (yes/no/[fingerprint])?
The authenticity of host [redacted] can't be established.
ECDSA key fingerprint is [redacted]
Are you sure you want to continue connecting (yes/no/[fingerprint])?
The authenticity of host [redacted] can't be established.
ECDSA key fingerprint is [redacted]
Are you sure you want to continue connecting (yes/no/[fingerprint])?
The authenticity of host [redacted] can't be established.
ECDSA key fingerprint is [redacted]
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

## Question 6

**A:** SSH with “-o StrictHostKeyChecking=no” option

*[COMP90024]*

*[COMP90024:vars]*

*ansible\_python\_interpreter=/usr/bin/python3*

*ansible\_user=ubuntu*

*ansible\_ssh\_private\_key\_file=<some private key>*

*ansible\_ssh\_common\_args='-o StrictHostKeyChecking=no'*

## Question 7

**Q:** How to safely store the Ansible variables for credentials in a Git repo?

**A:** Don't push it or use [Ansible Vault](#).

To encrypt a file: *ansible-vault encrypt file.yaml*

To decrypt a file: *ansible-vault decrypt file.yaml*

To view an encrypted file: *ansible-vault view file.yaml*

To run an encrypted playbook file:

*ansible-playbook --vault-id @prompt file.yaml*

## Question 8

**Q:** What can we do if we want multiple instances to access data stored in a volume? Can we do this by Ansible?

**A:** Yes and No.

**Yes:** OpenStack supports [multi-attach](#) since Cinder version 3.50.

### Warning

It is the responsibility of the user to ensure that a multiattach or clustered file system is used on the volumes. Otherwise there may be a high probability of data corruption.

**No:** NeCTAR / MRC does not support multi-attach