

Automation with Ansible





- Deploying complex cloud systems requires a lot of moving parts
 - Easy to forget what software you installed, and what steps you took to configure the system
 - Manual process is error-prone, can be non-repeatable
 - Snapshots are monolithic - provide no record of what has changed
- 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

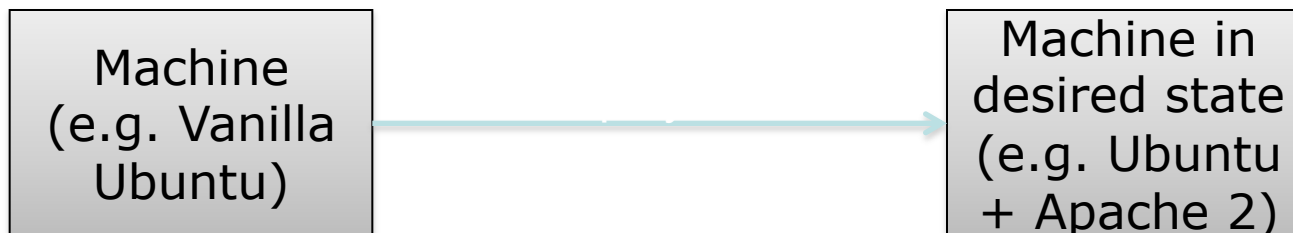
Configuration management refers to the process of ***systematically*** handling ***changes*** to a system in a way that it ***maintains integrity*** over time.

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, ...



- 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!?





- 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
```



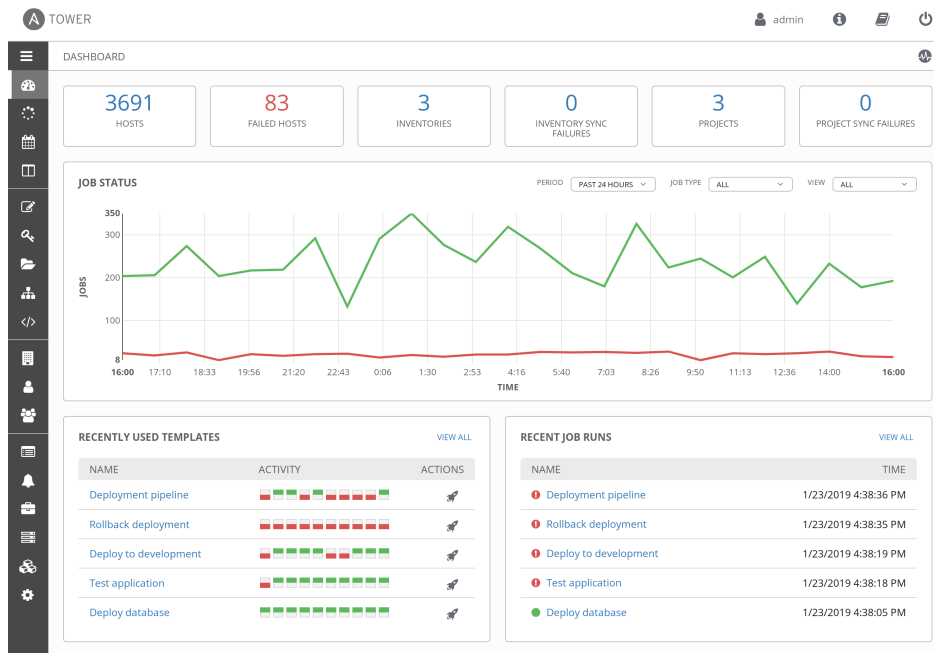
Ansible: More Features

- 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.





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
- 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



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 (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=9g0IGoRJtzM>
- Ansible documentation:
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.yml
| - inventory
|   | _ inventory.ini
| - roles
|   | - defaults
|   | - tasks
|   |   | - task1.yml
|   |   | _ task2.yml
|   | _ templates / files
| _ playbook.yml
```

```
[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*

apt:

name: "{{ package }}"

state: latest

- *name: Write the Apache config file*

file:

src: /srv/httpd.conf

dest: /etc/httpd.conf

- *name: Ensure Apache is restarted*

service:

name: httpd

state: restarted

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



Ansible Demo 1:

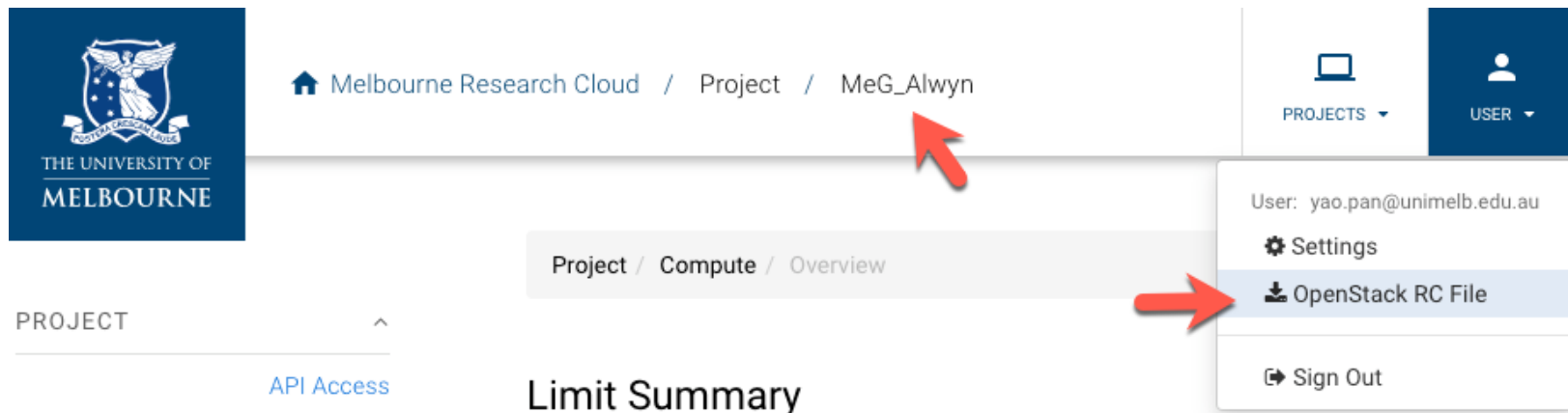
- Connect to Melbourne Research Cloud
- List all images
- Create volumes
- Create security groups with security rules
- Launch an instance and attach the volume and security groups
- Create snapshots of the volumes



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 user-related links: "PROJECTS", "USER", and "SUPPORT". The "USER" link is highlighted with a red arrow and the number 1. A dropdown menu is open under "USER", showing the user email "yao.pan@unimelb.edu.au" and options: "Settings" (highlighted with a red arrow and the number 2), "OpenStack RC File", and "Sign Out". On the left sidebar, the "SETTINGS" section is expanded, and the "Reset Password" link is highlighted with a red arrow and the number 3. The main content area displays the "Password Reset Form" with a "Description:" section explaining that the password will be forgotten and a "Reset Password" button is available. The button is highlighted with a red arrow and the number 4.

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:

4. Instance Flavor: [uom.general.2c8g](#)

5. Availability Zone: [melbourne-qh2-uom](#)

6. Image Id: [45225edb-66d8-4fd0-bf41-132a31a18166](#)

Dashboard -> Compute -> Images

PROJECT ^

API Access

Compute ^

Overview

Instances

Images

Key Pairs

Volumes v

Network v

Orchestration v

Database v

Data Processing v

Object Store v

Share v

APP CATALOG v

ALLOCATIONS v

Search: NeCTAR Ubuntu 18.04 LTS x + Create Image Delete Images

Displaying 2 items

<input type="checkbox"/>	Name ^	Type	Status	Visibility	Protected	Disk Format	Size	
<input type="checkbox"/>	NeCTAR Ubuntu 18.04 LTS (Bionic) amd64	Image	Active	Public	No	QCOW2	365.02 MB	Launch v

Name
NeCTAR Ubuntu 18.04 LTS (Bionic) amd64

Visibility
Public

Protected
No

Min. Disk
0

Min. RAM
0

ID
45225edb-66d8-4fd0-bf41-132a31a18166

libvirt Driver Options for Images

QEMU Guest Agent True

Detail Information
The libvirt Compute Driver Options for Glance Images. These are properties specific to compute drivers. For a list of all hypervisors, see here:
<https://wiki.openstack.org/wiki/HypervisorSupportMatrix>.



Ansible Demo: Melbourne Research Cloud

- *playbook* to interact with Melbourne Research Cloud
 - List all images
[os_image_info](#)
 - Create a volume
[os_volume](#)
 - Create a security group with rules
[os_security_group](#), [os_security_group_rule](#)
 - Launch an instance and attach the volume and security group
[os_server](#)
 - Create a snapshot of the volume
[os_volume_snapshot](#)
 - Other Ansible modules used
[apt](#), [pip](#), [become](#), [register](#), [set_fact](#), [debug](#), [loop](#),
[when](#), [wait_for](#), [add_host](#), [gather_facts](#)
- Run playbook:

```
$ ./openrc.sh; ansible-playbook [--ask-become-pass] nectar.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 “os_”*
- *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: os_server



Ansible Demo: Wordpress

- Simple *playbook* to deploy a *WordPress* instance with *Docker**
 - Install dependencies
[apt](#), [pip](#)
 - Mount volumes and make filesystems
[filesystem](#), [stat](#), [file](#), [mount](#)
 - Install *Docker* on one of the volumes
[apt_key](#), [apt_repository](#)
 - Deploy a *WordPress* instance with *Docker*
[template](#), [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
- 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/>
- A simple git guide, <http://rogerdudler.github.io/git-guide/>
- Git terminology, <https://git-scm.com/docs/gitglossary>