Ansible Automation

What is Ansible?

- Ansible is a tool that generates written instructions for automating IT professionals' work throughout the entire system infrastructure.
- It's designed particularly for IT professionals who use it for application deployment, configuration management, and practically anything else a systems administrator does on a weekly or daily basis.
- Ansible is a command line tool.
- Here, we'll continue with the open version of Ansible.

Ansible over other similar tools

- Agentless
- Minimal
- Descriptive
- Time saving
- Secure

Ansible Nodes

- Two types of nodes
  - Control node
  - Managed node
- Control node
  - The machine from which you install run the Ansible
  - You can use any computer that meets the software requirements as a control node - laptops, shared desktops, and servers can all run Ansible.
- Managed node
  - Also referred to as ‘hosts’, these are the target devices you aim to manage with Ansible.
  - Ansible is not normally installed on managed nodes.

Ansible Playbook

- Playbooks are one of the core features of Ansible and tell Ansible what to execute.
- They are like a to-do list for Ansible that contains a list of tasks.
- Written in YAML (.yaml / .yml)
- A YAML file cannot contain tabs as indentation
- Correct placing of indentation is very important
- Different YAML tags are there.

Playbook tags

- name
  - This tag specifies the name of the playbook/ play / task, etc.
  - As in what this playbook will be doing.

```
- name: copy a file
  hosts: localhost
```

- hosts
  - This tag specifies the lists of hosts or host group against which we want to run the task.
  - The **hosts** tag is **mandatory**.
  - It tells Ansible on which hosts to run the listed tasks.
  - The tasks can be run on the same machine or on a remote machine.

Cont'd

- become
  - Set privilege escalation
  - Set to **yes** to activate privilege escalation.
  - **Sudo** passwords might be required

```yaml
- name: copy a file
  hosts: localhost
  become: yes
  gather_facts: no
```

- gather_facts
  - This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks.
  - This has a Boolean value.
  - By default this is **true**.
  - Values: yes, no

Cont'd

- vars
  - **vars** tag lets you define the variables which you can use in your playbook.
  - Usage is similar to variables in any programming language.

```yaml
gather_facts: no
vars:
  - file_name: test
tasks:
  - name: copy file
    command: cp $PWD/moodle.yaml $PWD/{{ file_name }}.yaml
```

- tasks
  - All playbooks should contain tasks or a list of tasks to be executed.
  - Tasks are a list of actions one needs to perform.
  - A tasks field contains the name of the task.
  - The name is not mandatory but proves useful in debugging the playbook.

Simple Playbook Examples

- Copy a file in same directory

```yaml
hosts: localhost
gather_facts: no
vars:
  - file_name: test
tasks:
  - name: copy file
    command: cp $PWD/moodle.yaml $PWD/{{ file_name }}.yaml
```

Cont'd

- Install Apache2

```yaml
- name: install Apache in Ubuntu
  hosts: all
  become: yes
  gather_facts: false
  tasks:
    - name: Update
      shell:  apt update

    - name: install Apache
      apt:
        name: apache2
        state: latest
```

Cont'd

- Install MySQL

```yaml
- name: setup mysql
  hosts: all
  become: yes
  gather_facts: false
  vars:
    root_password: Redact#12
    db_name: moodle
    user_name: moodle
    user_password: Redact#13
  tasks:
    - name: Update
      shell:  apt update

    - name: install python, pip etc
      shell: apt-get -y install "{{ item }}"
      with_items:
        - pip
        - python3-dev
        - default-libmysqlclient-dev
        - build-essential
```

Cont'd

```yaml
- name: Install MySQL server
  shell: apt-get -y install mysql-server

- name: Install MySQL client
  shell: apt-get -y install mysql-client

- name: pip install mysqlclient
  shell: pip install mysqlclient

- name: Start the MySQL service
  action: service name=mysql state=started

- name: copy .my.cnf file with root password credentials
  template: src=/home/docker/my.cnf.j2 dest=/root/.my.cnf owner=root mode=0600

- name: update mysql root password for all root accounts
  mysql_user:
    name: root
    host: localhost
    password: "{{ root_password }}"
```

Cont'd

```
- name: Create database
  shell: mysql -u root -p{{ root_password }} -e 'CREATE DATABASE {{ db_name }} DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

- name: Create user
  shell: mysql -u root -p{{ root_password }} -e "CREATE USER '{{ user_name }}'@'%' IDENTIFIED BY '{{ user_password }}';"
```

```
- name: Grant permissions
  shell: mysql -u root -p{{ root_password }}
  -e "GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY TABLES,DROP,INDEX,ALTER ON {{ db_name }}.*
  TO '{{ user_name }}'@'%';"

- name: Reload privileges
  shell: mysql -u root -p{{ root_password }} -e "FLUSH PRIVILEGES;"
```

Cont'd

- Install Moodle (after installing MySQL)

```yaml
- name: Install Moodle on Ubuntu
  hosts: all
  become: yes
  gather_facts: false
  vars:
    site_fullname: moodle
    site_shortname: m1
    email: email@institute.ac.lk
    password: Password123#
  tasks:
    - name: Update
      shell:  apt update

    - name: Install Apache
      apt:
        name: apache2
        state: latest

    - name: Add repository
      shell: add-apt-repository -y ppa:ondrej/php

    - name: Install php7.4
      shell: apt -y install php7.4
```

Cont'd

```
- name: Install packages
  shell: apt-get -y install "{{ item }}"
  with_items:
    - graphviz
    - aspell
    - ghostscript
    - clamav
    - php7.4-pspell
    - php7.4-curl
    - php7.4-gd
    - php7.4-intl
    - php7.4-mysql
    - php7.4-xml
    - php7.4-xmlrpc
    - php7.4-ldap
    - php7.4-zip
    - php7.4-soap
    - php7.4-mbstring

- name: Download Moodle code
  shell: cd /opt && git clone -b MOODLE_400_STABLE git://git.moodle.org/moodle.git

- name: Move Moodle code to webroot
  shell: mv /opt/moodle/* /var/www/html/

- name: Make Moodledata directory
  shell: mkdir /var/moodledata
```

Cont'd

- config.php

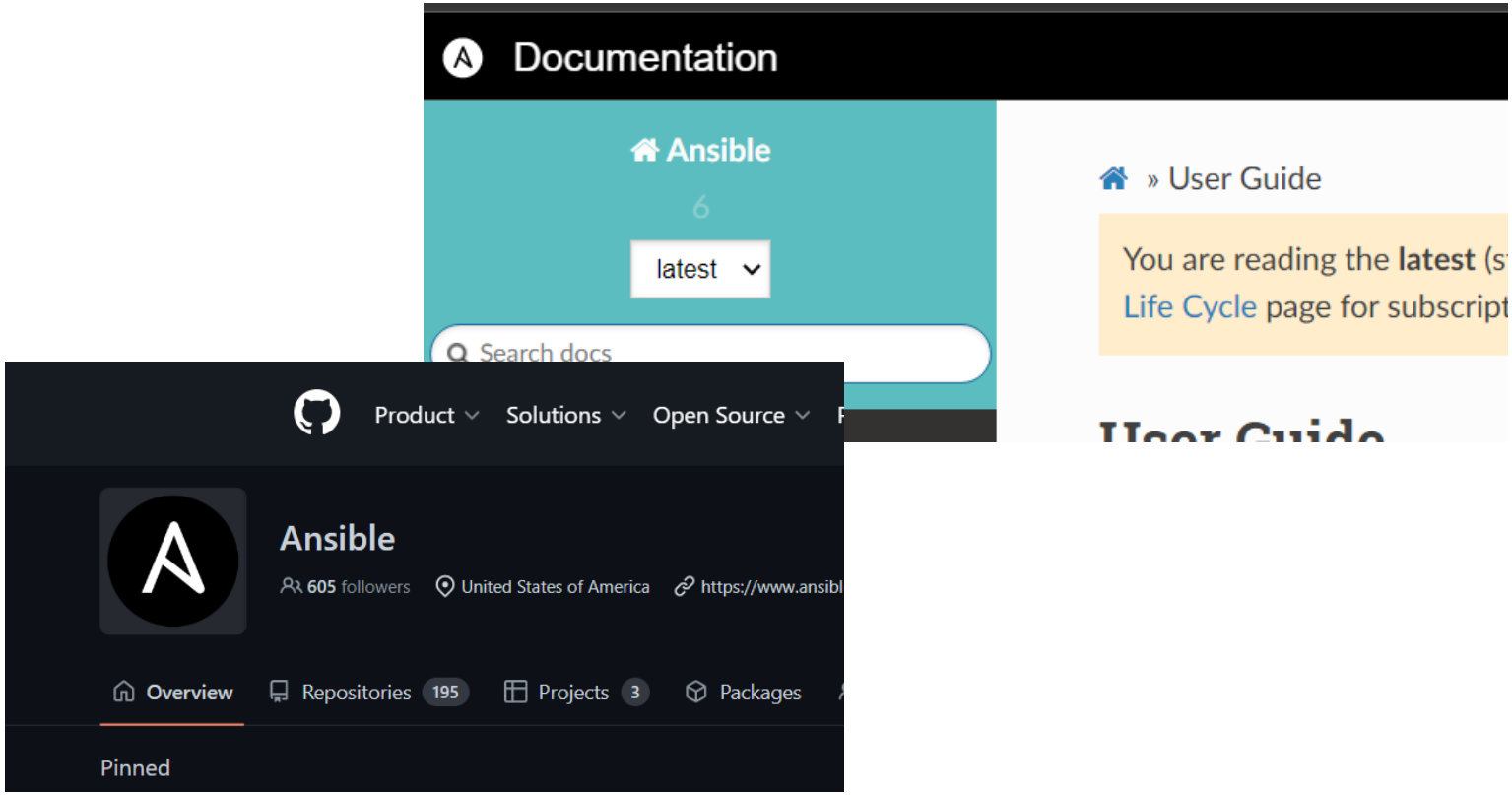
```php
<?php  // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype    = 'mysqli';
$CFG->dblibrary = 'native';
$CFG->dbhost    = 'localhost';
$CFG->dbname    = 'new';
$CFG->dbuser    = 'newuser';
$CFG->dbpass    = 'Redact#13';
$CFG->prefix    = 'mdl_';
$CFG->dboptions = array (
  'dbpersist' => 0,
  'dbport' => 3306,
  'dbsocket' => '',
  'dbcollation' => 'utf8mb4_unicode_ci',
);

$CFG->wwwroot   = 'http://'.$_SERVER['HTTP_HOST'];
//$CFG->reverseproxy = true;
//$CFG->sslproxy  = true;
$CFG->dataroot  = '/var/moodledata';
$CFG->admin     = 'admin';

$CFG->directorypermissions = 02777;

require_once(__DIR__ . '/lib/setup.php');
```

Cont'd

```
- name: Permissions
  shell: chown -R www-data /var/moodledata && chmod -R 777 /var/moodledata && chmod -R 0755 /var/www/html

- name: Remove index.html
  shell: rm /var/www/html/index.html

- name: Copy config.php to webroot
  ansible.builtin.copy:
    src: /home/docker/config.php
    dest: /var/www/html/config.php
    owner: root
    mode: '0644'

- name: Install moodle
  shell: cd /var/www/html/ && php admin/cli/install_database.php --fullname={{ site_fullname }}
  --shortname={{ site_shortname }} --adminemail={{ email }} --adminpass={{ password }} --agree-license

- name: Restart Apache2
  shell: systemctl restart apache2
```

Need Help ?

- Support channels
  - Ansible Community Docs
  - GitHub



Thank You