Ansible Cheatsheet

Ansible is an open-source automation tool used for configuration management, application deployment, and task automation. It allows IT teams to automate repetitive and complex tasks, such as server configuration, application deployment, and system updates, in a fast, reliable, and scalable way.

Ansible's importance lies in its ability to reduce the time and effort required to manage large-scale infrastructure, alowing IT teams to focus on higher-value tasks. It provides a standardized way of automating IT infrastructure that is easy to learn and implement, making it accessible to both experienced and novice users. Ansible's agentless architecture also simplifies deployment, making it easy to integrate with existing infrastructure and tools.

Ansible's features include:

- Simple, agentless architecture
- Easy-to-learn language based on YAML and Jinja2 templates
- Automation of IT tasks, such as configuration management and application deployment
- Modular design and extensive library of modules for managing different systems and technologies
- Reusability of playbooks and roles, allowing for efficient automation and standardization
- Support for cloud environments, including Amazon Web Services, Microsoft Azure, and Google Cloud Platform
- Integration with other tools, such as Jenkins, Git, and Docker

Installation

Install Ansible on 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
```

Getting Started

Create a new Ansible playbook:

```
nano playbook.yml
```

Run the Ansible playbook on localhost:

```
ansible-playbook playbook.yml --connection=local
```

Playbook Structure

A minimal Ansible playbook:

```
- hosts: all
  tasks:
  - name: Example Task
    shell: echo "Hello World!"
```

Use a role in a playbook:

```
- hosts: all
  roles:
    - my-role
```

Roles

Import an existing role from Ansible Galaxy:

```
cd roles
ansible-galaxy import {name of the project/role}
```

Import a task from a role:

```
- hosts: localhost
  tasks:
    - name: Example Task
      include_role:
        name: my-role
        tasks_from: task-name
```

Variables

Include external variables in a playbook:

```
- name: Include Variables
  include_vars: path/to/vars.yaml
```

Define variables in a playbook:

```
- hosts: all
  vars:
    my_var: my_value
  tasks:
    - name: Example Task
      shell: echo "{{ my_var }}"
```

Modules

Install packages with the apt module:

```
- name: Install Package
  apt:
    name: package-name
    state: present
```

Start a service with the service module:

```
- name: Start Service
  service:
    name: service-name
    state: started
    enabled: yes
```

Manage Python packages with the pip module:

```
- name: Manage Python Packages
  pip:
    name: package-name
    state: present
```

Vault

Encrypt a file with Ansible Vault:

```
ansible-vault encrypt file.txt
```

Decrypt a file with Ansible Vault:

```
ansible-vault decrypt file.txt
```

Create a new file with Ansible Vault:

```
ansible-vault create file.txt
```

Run a playbook with Ansible Vault:

```
ansible-playbook playbook.yml --vault-password-file password-file.txt
```

Debugging

Increase verbosity with the -v flag:

```
ansible-playbook playbook.yml -v
```

Use the -vv or -vvv flag for more verbose output.

Print the value of a variable with the debug module:

```
- name: Print Variable
  debug:
    var: my_var
```

Views 56

Join our Discord Server