**Ansible Cheat Sheet**

### YAML

**Key-value pair**

```
Dictionary is represented in key value pair
Ex: names:
    name: james john
    rolNo: 34
    div: B
    sex: male
```

**Representing lists**

- Each element has to be written in a new line with "-" as the prefix
- **Countries**:
  - America
  - Iceland

**Lists inside the dictionary**

- name: james john
- rolNo: 34
- div: B
- sex: male
- likes: English

- Boolean terms are also used in YAML

### ANSIBLE

**Types of machines**

- **Control machine**: manages other machines
- **Remote machine**: controlled by other machines

### Environment Setup

- Multiple remote systems can be handled by one machine.
- Remote machine managing is done by ansible by default.
- Ansible doesn’t leave any software running on them. Therefore there is no need of an upgrade when moving to a newer version.
- It installs through apt, yum, ppa, OpenCSW
- Installing it through apt
- $ sudo apt-get update 
- $ sudo apt-get install software-properties-common
- $ sudo apt-add-repository ppa:ansible/ansible
- $ sudo apt-get install ansible
- Run ansible version to make sure it was installed properly.

### Ad-hoc Commands

- General syntax of ad-hoc command:
  - `ansible hostgroup module/option[arguments]`

### Inventory

- YAML syntax is used to express ansible playbooks
- The `hosts` in any playbooks need to be present in the `inventory` file.

### How Does It Work?

- Connects nodes and pushes small programs called modules to them and removes when they are done.
- The management node controls whole execution of the playbook.
- The `inventory` file provides the list of hosts where the modules need to be run.
- The management node does an `ssh` connection and executes the modules and installs the software.

### Troubleshooting

- Common strategies to debug playbooks are:
  - Debug and register
  - Use verbosity level

### Advantages of Ansible

- It is free and open source.
- Agentless. No master client model.
- System requirements.
- Developed in python.
- Lightweight and quick deployment.
- Ansible uses YAML syntax in config files.
- Large community base.

### Playbooks

- It is the place where all YAML files are stored and executed.
- Acts like a to-do list
- YAML - yet another markup language
- A playbook can have more than one plays. Plays map the instructions defined against a particular host.
- Typically written in a text editor like notepad or notepad++.

### Variables

- Same as using variables in programming languages
- Ex: `hosts: your hosts`
- `tomcat_port: 8080`
- Here `tomcat_port` is assigned to 8080
- Keywords used:
  - `block`: ansible syntax to execute a block
  - `become`: yes
  - `try`: the output
  - `except`: code that is to be executed
  - `register`: registers the output
  - `always`: states that below word will be run
  - `msg`: displays the message
- Exception handling:
  - Similar to any other programming language
- Keywords: rescue and always
- The code is written in block
- It goes to the rescue phase and gets executed if the command in the block fails.
- Thereby block is the same as “try block”, catch block is like “ rescue” and always performs the same function as we know.

### Terms

- `Service/server`: a process that provides service
- `Machine`: physical machine, VM or a container
- `Target machine`: end machine to be configured by ansible
- `Task`: an action
- `_playbook`: location where YAML files are written and executed.

---

**Ansible Management Tools**

![Ansible Management Tools Diagram]

**Playbook**

```
hosts: [jane1, jane2]
name: install and configure DB
```

**Inventory**

```
[hosts]
jane1
jane2
```

---

**Ad-hoc Commands**

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>COMMANDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check connectivity of hosts</td>
<td>ansible -g group ping</td>
</tr>
<tr>
<td>Rebooting hosts</td>
<td>ansible -g group -a &quot;service reboot&quot;</td>
</tr>
<tr>
<td>Check host system’s info</td>
<td>ansible -g group -m info less</td>
</tr>
<tr>
<td>Transfering files</td>
<td>ansible -g group -m copy -a &quot;src=home/ansible dest=/tmp/&quot;</td>
</tr>
<tr>
<td>Create new user</td>
<td>ansible -g group -m user -a &quot;name=ansible password=encrypted password=&quot;</td>
</tr>
<tr>
<td>Deleting user</td>
<td>ansible -g group -m user -a &quot;name=ansible state=absent&quot;</td>
</tr>
<tr>
<td>Check file is installed and update it</td>
<td>ansible -g group -m yum -a &quot;name=httpd state=latest&quot;</td>
</tr>
<tr>
<td>Check file is installed and dont update it</td>
<td>ansible -g group -m yum -a &quot;name=httpd state=present&quot;</td>
</tr>
<tr>
<td>Check file is a specific version</td>
<td>ansible -g group -m yum -a &quot;name=httpd 1.8 state=latest&quot;</td>
</tr>
<tr>
<td>Check file is not installed</td>
<td>ansible -g group -m yum -a &quot;name=httpd state=absent&quot;</td>
</tr>
<tr>
<td>Starting a service</td>
<td>ansible -g group -m service -a &quot;name=httpd httpd/started&quot;</td>
</tr>
<tr>
<td>Stopping a service</td>
<td>ansible -g group -m service -a &quot;name=httpd httpd/stopped&quot;</td>
</tr>
<tr>
<td>Restarting a service</td>
<td>ansible -g group -m service -a &quot;name=httpd httpd/restart&quot;</td>
</tr>
</tbody>
</table>

---

**FURTHERMORE:**

- Developed in python.
- Lightweight and quick deployment.
- Ansible uses YAML syntax in config files.
- Large community base.
- Ansible is an open source engine that automates deployment, orchestration, cloud provisioning and other tools.
- It uses a playbook to describe jobs and uses YAML which is human readable.
- It is designed for multi-tier deployment. It is agentless and works by connecting nodes through ssh.
- Always
- Thereby
- A
- Action
- apt
- It
- YAML
- Control
- name
- sex
- Machine
- It
- YAML
- yet another markup language
- A
- playbook can have more than one plays. Plays map the instructions defined against a particular host.
- Typically written in a text editor like notepad or notepad++.
- Sample playbook YAML file:
  - name: install and configure DB hosts: testServer
  - become: yes
  - vars: oracle_db_port_value: 1521
  - tasks:
    - name: Install the Oracle DB
      yum: name=OracleDB src=/home/ansible dest=/tmp/home 
    - name: Ensure the installed service is enabled
      service: name=your_service_name state=enabled
  - tags: YAML
  - vars:
    - name: your playbook
    - hosts: list of hosts
  - tasks:
    - name: Do something useful
      shell: command
  - vars:
    - name: my_var
  - tasks:
    - name: Do something else
      shell: command

---