**Ansible Automation:**

Ansible is a powerful automation tool that can replace many Bash scripts with more maintainable and scalable solutions. Ansible is an automation tool used for configuration management, application deployment, and task automation.

**Key Ansible concepts:**

1. Playbooks: YAML files containing a set of tasks to be executed on remote hosts.

2. Tasks: Individual units of work in a playbook.

3. Modules: Pieces of code Ansible executes to perform specific operations.

4. Inventory: A list of managed nodes that Ansible can work with.

**Ansible Playbook Structure**

---

- name: Playbook Name

hosts: target\_hosts

become: yes/no

vars:

variable1: value1

tasks:

- name: Task 1 Name

module\_name:

param1: value1

param2: value2

- `name`: A description of what the playbook or task does.

- `hosts`: Specifies which hosts from the inventory this play applies to.

- `become`: Whether to escalate privileges (like sudo).

- `vars`: Define variables used in the playbook.

- `tasks`: A list of tasks to be executed.

**Ansible configuration file:**

When we install ansible by default configuration files will get created in the following location:

**/etc/ansible/ansible.cfg**

This configuration file contains several sections, they are:

1. Defaults
2. Inventory
3. Privilege\_escalation
4. SSH\_connection
5. Paramiko\_connection
6. Persist\_connection
7. Colors

Here are some frequently used configuration options:

1. In the [defaults] section:
   * inventory: Specifies the default inventory file
   * remote\_user: Default username for SSH connections
   * host\_key\_checking: Whether to check SSH host keys
   * roles\_path: Where to look for roles
   * forks: Number of parallel processes to use
2. In the [privilege\_escalation] section:
   * become: Whether to use privilege escalation by default
   * become\_method: Default method for privilege escalation (e.g., sudo, su)
   * become\_user: Default user to become when using privilege escalation
3. In the [ssh\_connection] section:
   * ssh\_args: Additional SSH arguments
   * control\_path: Location of ControlPath sockets

**Environment Variable:**

`ANSIBLE\_CONFIG` = `/opt/ansible\_web.cfg`

**Copy of Default Config File in current directory:**

`/opt/web\_playbooks/ansible.cfg`

**Config file in home directory:**

`.ansible.cfg`

**Default Config File:**

`/etc/ansible/ansible.cfg`

If we have all types of configuration files then it follows the priority:

1. Environmental variable
2. Current directory config file
3. Home directory config file
4. Default config file

**Example of ansible.cfg:**

[defaults]

inventory = ./inventory

Log\_path = /var/log/ansible.log

library= /usr/share/my\_modules

roles\_path=atc/ansible/roles

action\_plugins=/usr/share/ansible/plugins/action

remote\_user = ansible

host\_key\_checking = False

gathering= implicit

timeout=10

forks = 5

[privilege\_escalation]

become = True

become\_method = sudo

become\_user = root

[ssh\_connection]

ssh\_args = -o ControlMaster=auto -o ControlPersist=60s

**Ansible configuration variables:**

1. ANSIBLE\_GATHERING= explicit ansible-playbook playbook.yml
2. export ANSIBLE\_GATHERING= explicit

Ansible-playbook playbook.yml

1. /opt/web-playbooks/ansible.cfg

gathering =explicit

**View configuration:**

* ansible -config list ⇒ list all the configurations
* ansible-config view ⇒shows the current active config file details
* ansible-config dump ⇒ shows the current settings
* **Version Control**: Keep your ansible.cfg in version control along with your playbooks.
* **Project-Specific Configurations**: Use project-specific ansible.cfg files in your project directories for settings that should apply only to that project.
* **Comment Your Configurations**: Use comments (lines starting with ';' or '#') to explain non-obvious settings.
* **Security**: Be cautious with settings like host\_key\_checking = False. While convenient for testing, it can be a security risk in production environments.
* **Use Environment Variables**: For sensitive information, use environment variables instead of hardcoding values in ansible.cfg.
* **Regular Review**: Periodically review and update your configuration to ensure it aligns with current best practices and your project needs.