Ansible video : <https://www.youtube.com/watch?v=wgQ3rHFTM4E>

**Ansible** is an effective tool for automating **the creation of the target environment** (collection of predefined options that contain information about the environment for a specific system) and **subsequent application deployment**.

For CI/CD to provide the target environment, **Jenkins** is employed.

Ansible is used to:

1. Automate the IT setup.
2. Setting/Config the setup.
3. Automate the deployment.

Ansible uses the Push based configuration that does not require any client/agent installed in its client servers.

The Master server will have the playbook and inventory (list of slave servers/nodes). It sends small programs known as Ansible Modules to the nodes.

Modules of Ansible are independent scripts inside an Ansible playbook. Ideally, they are units of code that can be used from the command line.

Ansible provides seamless application configuration, while Docker provides a containerized environment for building and deploying applications. Therefore, you should consider Ansible for application configuration and Docker for code shipping & deployment.

Ansible is primarily a configuration tool. It helps configure the application or software inside the container. The software or application can be either inside or outside the container. Ansible provides the server installation, configuration, and deployment.

**In my office setup**:

Ansible, running in your Docker container, will connect to the target hosts specified in your inventory file and execute tasks on those remote systems.

**What is provisioning in ansible? Is it equivalent to pom in Maven: No.**

**Major commands/modules used in ansible:**

Got these from ChatGPT

Ansible provides a wide range of modules that cover various aspects of IT automation and configuration management. These modules enable you to perform specific tasks on target hosts. Here's a list of some commonly used Ansible modules, categorized by functionality:

**System Administration**:

command: Run arbitrary shell commands on target hosts.

shell: Execute shell commands on target hosts, similar to the command module.

copy: Copy files or directories from the control host to target hosts.

file: Perform file and directory operations on target hosts (e.g., create, delete, set permissions).

template: Use Jinja2 templates to create configuration files on target hosts.

cron: Manage cron jobs on target hosts.

**Package Management**:

yum: Manage packages on RPM-based Linux distributions (e.g., CentOS, Red Hat).

apt: Manage packages on Debian-based Linux distributions (e.g., Ubuntu).

**Service Management**:

service: Manage system services (e.g., start, stop, enable, disable).

systemd: Control systemd services on modern Linux distributions.

**User and Group Management**:

user: Manage user accounts on target hosts.

group: Manage groups on target hosts.

passwd: Change user passwords on target hosts.

**File Content Management**:

lineinfile: Manage lines in text files on target hosts.

replace: Search and replace text in files on target hosts.

blockinfile: Insert or update a block of text in a file.

**Network Configuration**:

ios\_command: Execute commands on Cisco IOS devices.

nxos\_command: Execute commands on Cisco NX-OS devices.

bigip\_command: Execute commands on F5 BIG-IP devices.

**Cloud and Virtualization**:

ec2: Manage Amazon Web Services (AWS) resources.

azure\_rm: Manage Microsoft Azure resources.

vmware\_vm\_facts: Retrieve information about VMware virtual machines.

docker\_container: Manage Docker containers.

k8s: Manage Kubernetes resources.

**Database Management**:

mysql\_db: Manage MySQL databases.

postgresql\_db: Manage PostgreSQL databases.

**Security**:

acl: Manage Access Control Lists on files and directories.

selinux: Manage SELinux settings on target hosts.

**Monitoring and Logging**:

syslog: Manage syslog settings.

logrotate: Manage log rotation configuration.