Ansible Theory:

1. **What is Ansible?** Ansible is a computer program that helps you automate repetitive tasks on your computers, like setting up servers, installing software, and configuring them.
2. **How does it work?** Ansible uses a simple language to describe what you want to happen on your computers. You write "playbooks" to tell Ansible what tasks to perform, and it takes care of making them happen.
3. **No Agents:** Unlike some other automation tools, Ansible doesn't need extra software installed on your computers. It talks to them over the network, which makes it easy to use.
4. **Idempotence:** Ansible is smart. You can tell it to do something over and over, and it will only change what's necessary to make your computers match your instructions.
5. **Organized Work:** Ansible organizes tasks into "playbooks" and "roles," making it easier to manage and reuse your automation instructions.
6. **Inventory:** You list your computers in an "inventory" file, so Ansible knows where to work. This list can be dynamic, so you can add or remove computers easily.
7. **Modules:** Ansible comes with lots of pre-built tools(Commands) called "modules" that help you do common tasks like installing software, managing files, and more.

History of Ansible:

- Ansible was created in 2012 by Michael DeHaan.
- It became popular because of its simplicity and not needing extra software on the computers it manages.
- Red Hat, a big software company, acquired Ansible, Inc. in 2015, which helped it grow even more.
- IBM later acquired Red Hat in 2019, so Ansible is now under IBM's umbrella.
- As of my last update in September 2021, Ansible remained a widely used and loved tool for automation tasks in IT.

In simple terms, Ansible is like a helpful robot that follows your instructions to set up and manage your computers, and it's known for being easy to use and flexible.

The term "YAML" stands for "yet another markup language."

In simpler language, YAML is a way to write information in a structured and easy-to-read format. It's often used in computer programs to store configuration settings, data, or instructions. It's like a language that both computers and people can understand easily. YAML files use plain text and simple symbols, making them user-friendly for both programmers and non-programmers.

Advantages of Ansible:

1. **Simplicity:** Easy to learn and use.
2. **Agentless:** No need for extra software on servers.
3. **Idempotence:** Safe and predictable automation.
4. **Readability:** Playbooks are easy to understand.
5. **Community:** Large and supportive user community.

Disadvantages of Ansible:

1. **Performance:** Slower for very complex tasks.
2. **Full Automation:** Limited Automation.
3. **Documentation:** As its new in market, therefore limited support and documents are available.

Explanations of Ansible terms:

1. **Playbook:** A set of instructions written in a file to tell Ansible what to do, like a recipe for automation.
2. **Role:** A reusable collection of tasks and configurations to perform common jobs, like installing software.
3. **Inventory:** A list of the computers or devices Ansible will work with.
4. **Task:** A single action or job in Ansible, like copying a file or restarting a service.

5. **Module:** A tiny program(commands) inside Ansible that does a specific task, such as creating a user or installing a package.
6. **Handler:** A special task that only runs when needed, like restarting a service after a change.
7. **Facts:** Information about the managed computers collected by Ansible.
8. **Ad-hoc Command:** A quick, one-time command to do something without writing a playbook.
9. **Vault:** A way to keep secrets, like passwords, safe in Ansible.
10. **Play:** A list of tasks in a playbook that work together.
11. **Gathering Facts:** Ansible's way of learning about the computers it's working with.
12. **Inventory File:** A file listing computers and their details for Ansible to use.
13. **Ad-Hoc Mode:** A quick way to run a single task without using a playbook.
14. **Idempotent:** Ansible tasks that don't change things if run multiple times.
15. **SSH:** A secure way Ansible connects to and controls computers.
16. **Variables:** Data used to make playbooks and roles flexible and reusable.
17. **cModule Parameters:** Settings that customize how Ansible modules work.
18. **Tags:** Labels on tasks, letting you choose which ones to run in a playbook.

Setting Up Ansible

```
yum update -y
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum install epel-release-latest-7.noarch.rpm
yum update -y
yum install git python python-level python-pip openssl ansible -y
ansible --version

vi /etc/ansible/hosts (make a group and mention private ip of nodes)
vi /etc/ansible/ansible.cfg   (uncomment inventory and sudo_user)
```

Below steps do be done on all nodes

```
adduser ansible
```

passwd ansible

root user on all nodes

visudo (ansible ALL=(ALL) NOPASSWD: ALL)

su ansible

sudo yum install httpd -y

root user(on all nodes)

vi /etc/ssh/sshd_config (uncomment PermitRootLogin yes uncomment PasswordAuthentication yes and comment #PasswordAuthentication no)

service sshd restart

now execute below on ansible server

su ansible

ssh-keygen

cd .ssh

ssh-copy-id ansible@private_ip

ansible all --list-hosts

ansible test_group --list-hosts

ansible test_group[0] --list-hosts

ansible test_group[1] --list-hosts

ansible test_group[-1] --list-hosts

ansible test_group[0:1] --list-hosts

Adhoc commands:

```
ansible test_group -a "ls"
ansible all -a "ls"
ansible all -a "touch cloudnation"
ansible test_group -a "ls -al"
ansible test_group -a "sudo yum install httpd -y"
which httpd
ansible test_group -ba "yum remove httpd -y"
ansible all -a "touch cloudnation"(it will make the file once again)
```

Modules:

```
ansible test_group -b -m yum -a "pkg=httpd state=present"
```

now lets rerun this and see what happens

```
ansible test_group -b -m yum -a "pkg=httpd state=present"
```

for updating

```
ansible test_group -b -m yum -a "pkg=httpd state=latest"
```

to remove

```
ansible test_group -b -m yum -a "pkg=httpd state=absent"
```

install

```
ansible test_group -b -m yum -a "pkg=httpd state=present"
```

```
sudo service httpd status
```

```
ansible test_group -b -m service -a "name=httpd state=started"
```

```
sudo service httpd status

ansible <target_server> -m command -a "sudo service httpd status"
```

to create a user

```
ansible test_group -b -m user -a "name=mubashir"
cat /etc/passwd
```

to copy a file on a particular server

```
ansible test_group [-1] -b -m copy -a "src=filename dest=/tmp"
```

to copy a file on all servers

```
ansible test_group -b -m copy -a "src=filename2 dest=/tmp"
```

Target

```
--- #i am new to devops
- hosts: group_test
  user: ansible
  become: yes
  connection: ssh
  gather_facts: yes
```

```
ansible-playbook myplaybook.yml
```

Target and Tasks

```
--- #i am new to devops
- hosts: group_test
  user: ansible
  become: yes
  connection: ssh
  tasks:
    - name: Install httpd on centos 7
      action: yum name=httpd state=installed
```

Variable

```
--- #i am new to devops
- hosts: group_test
  user: ansible
  become: yes
  connection: ssh
  vars:
    - pkgname: httpd
  tasks:
    - name: install httpd
      action: yum name='{{pkgname}}' state=installed
```