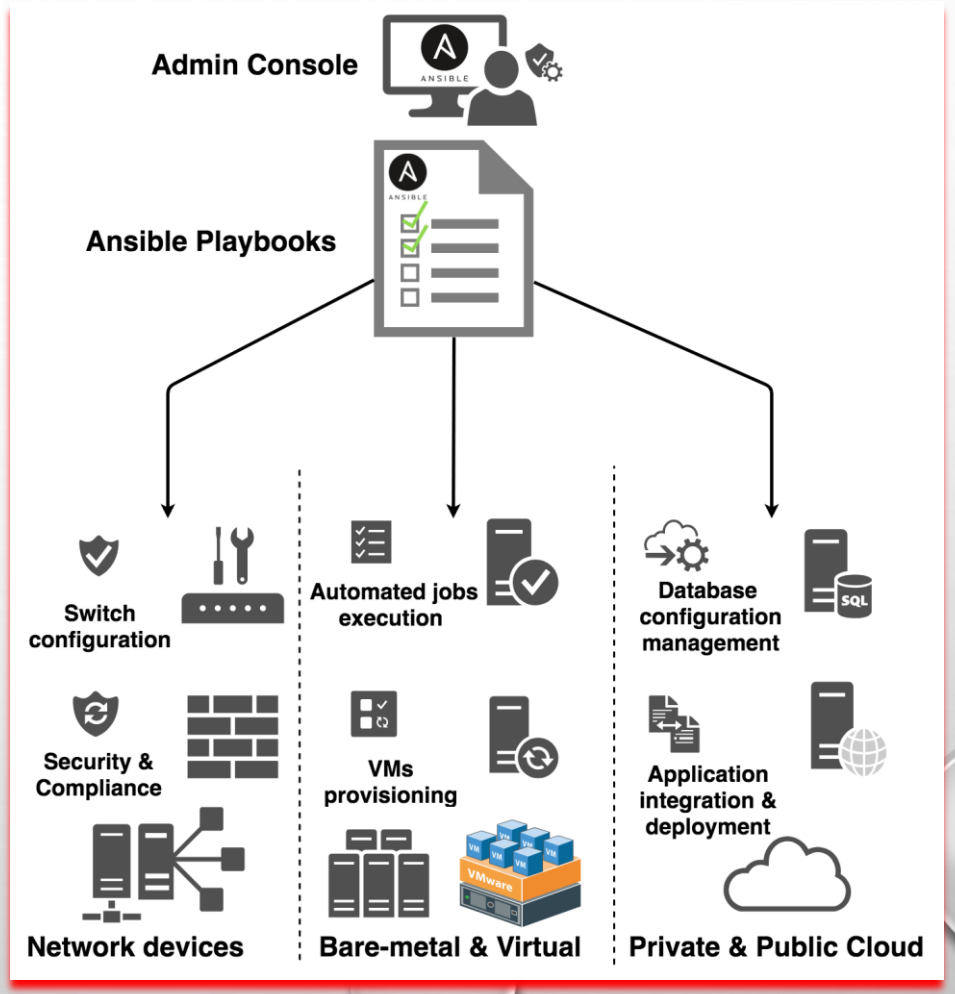
Ansible

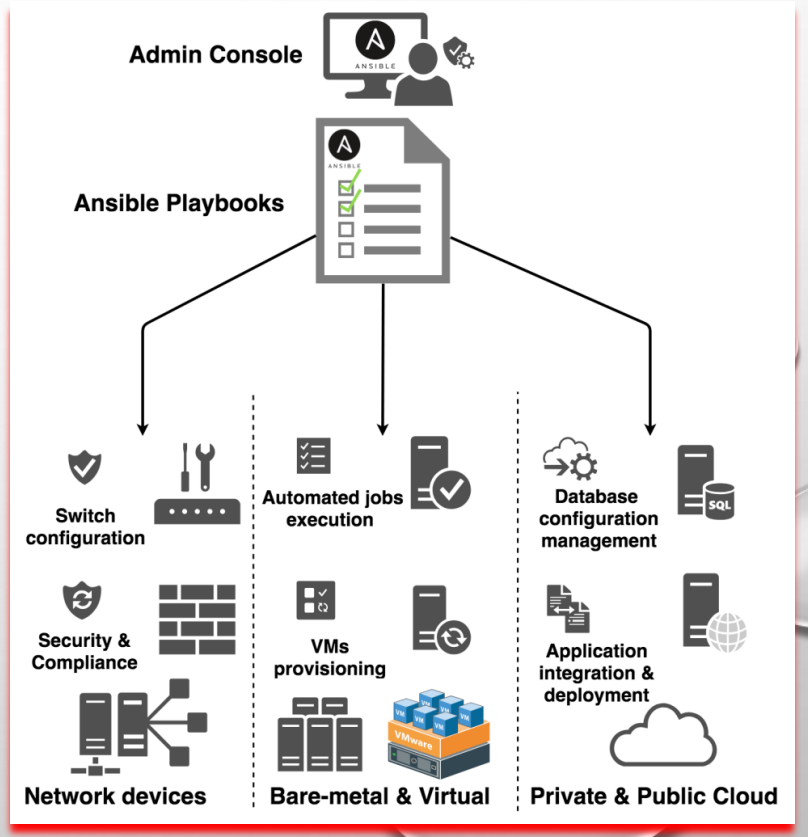
- Ansible Introduction.
- Why do we require Ansible?
- Advantages of Using Ansible.
- Pull Based Vs Push Based.
- Ansible Architecture.
- Ansible Terms.
- Ansible Working.
- Role of Ansible in DevOps.
- Core components of Ansible.
- Ansible Installation.
- Ansible Ad-Hoc Commands.
- Ansible Modules.
- Introduction To YAML.
- Ansible Plays.
- Ansible Playbooks.
- Ansible Configuration Files Writing.
- Automate Standard RHCSA Tasks Using Ansible.
- Ansible Vault.
- Ansible Roles.
- Ansible Galaxy.



Ansible

➢ **Ansible Introduction:**

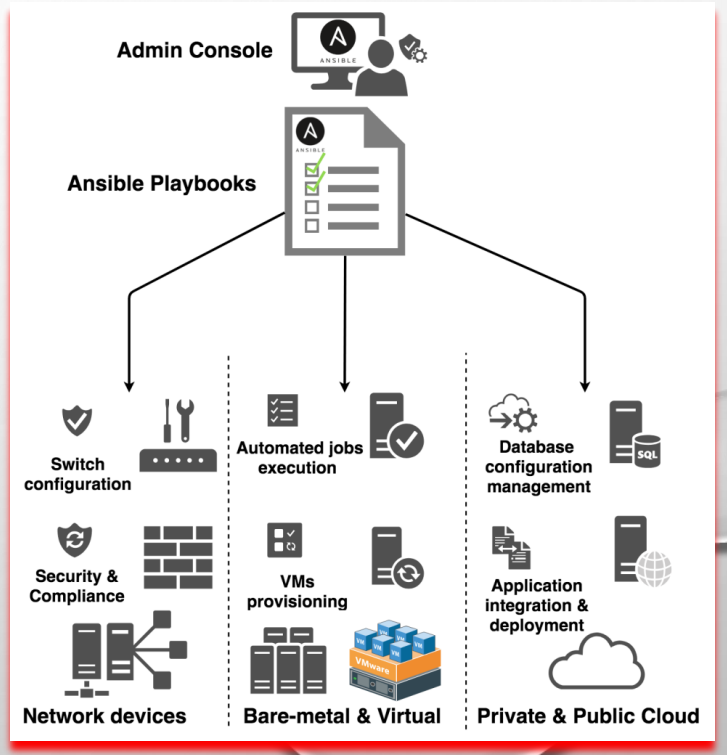
- ✓ Ansible is simple open source IT engine which automates: application deployment, intra service orchestration, cloud provisioning & many other IT tools.
- ✓ Ansible is easy to deploy because it does not use any agents or custom security infrastructure.
- ✓ Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. YAML
- ✓ Ansible is designed for multi-tier deployment.
- ✓ Ansible uses the hosts file where one can group the hosts and can control the actions on a specific group in the playbooks.



Ansible

➢ **Why Do We Require Ansible:**

- ✓ Ansible automates and simplifies repetitive, complex, and tedious operations.
- ✓ Ansible is open source, saves time as well as human efforts and is very easy to implement.
- ✓ Ansible architecture is simple and effective, It works by connecting to your nodes & pushing small programs to them.
- ✓ Ansible is push-based architecture & doesn't need any agents running on the client nodes.
- ✓ Ansible **works over SSH** and doesn't require any daemons, special servers, or libraries to work. A text editor and a command line tool are usually enough to get your work done.
- ✓ Ansible infrastructure is described in a text file (INI) & all the information abo ut the desired state of these machines are organized in playbooks.

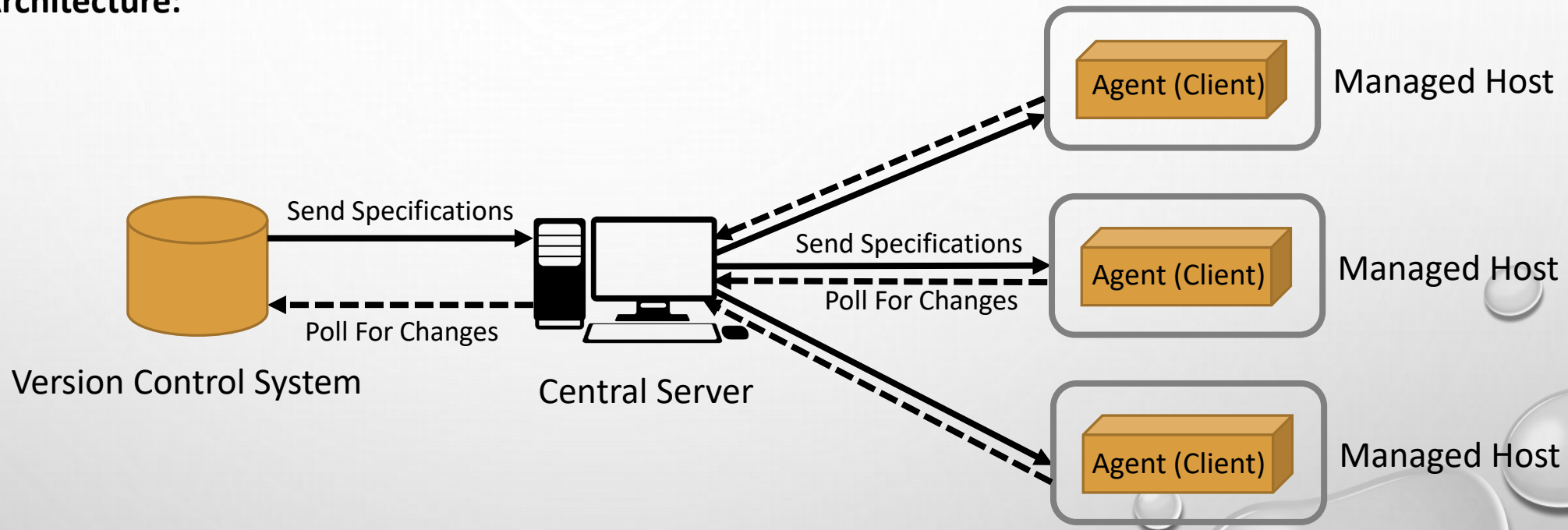


- **Advantages of Using Ansible:**

- ✓ **Agentless:** It doesn't require any additional software on your server nodes. This helps keep the installation clean while ensuring that there are no conflicts with our software.
- ✓ **Simple:** Ansible uses a simple syntax written in YAML called ***playbooks***. YAML is a human readable data seri alization language, doesn't require coding skills.
- ✓ **Powerful & Flexible:** Ansible has powerful features that can enable you to model even the most complex IT workflows.
- ✓ **Efficient:** No extra software on your servers means more resources for your applications.
- ✓ **Open-Source**: Ansible is one of the powerful DevOps tools which is open-source.
- ✓ **Secure:** Ansible uses SSH connection which is encrypted and secure.
- ✓ **Ease of Use:** One can configure & manage complex infrastructure solutions very easily using Ansible.

Ansible

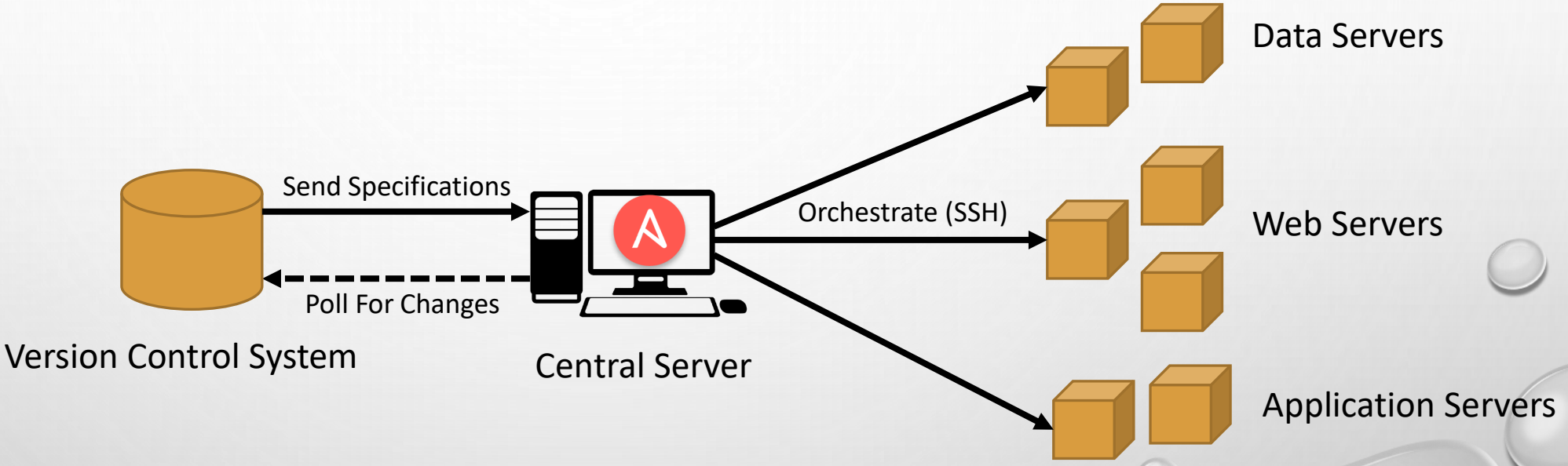
- **Pull Based Architecture:**



- ✓ Tools like chef & puppet are pull architecture based.
- ✓ Agents on the target servers (nodes) periodically checks for the configuration information from the central server.

Ansible

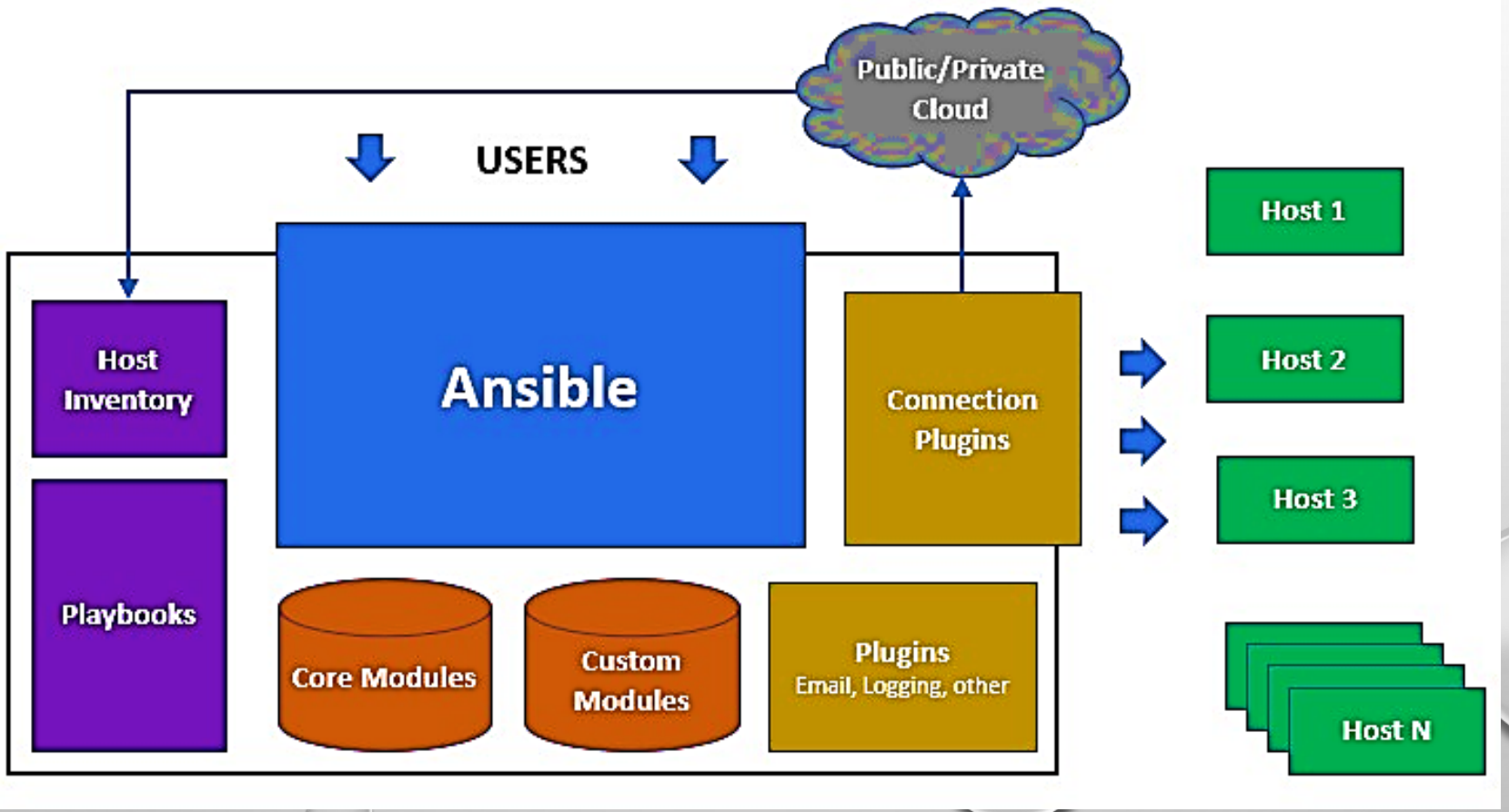
- **Push Based Architecture:**



- ✓ Ansible has push-based architecture & doesn't need any agents running on the client nodes.
- ✓ Central server pushes the configuration information on the target servers (nodes).
- ✓ You can control when the changes are implemented on the target servers.

Ansible

- **Ansible Architecture: (Push Based)**

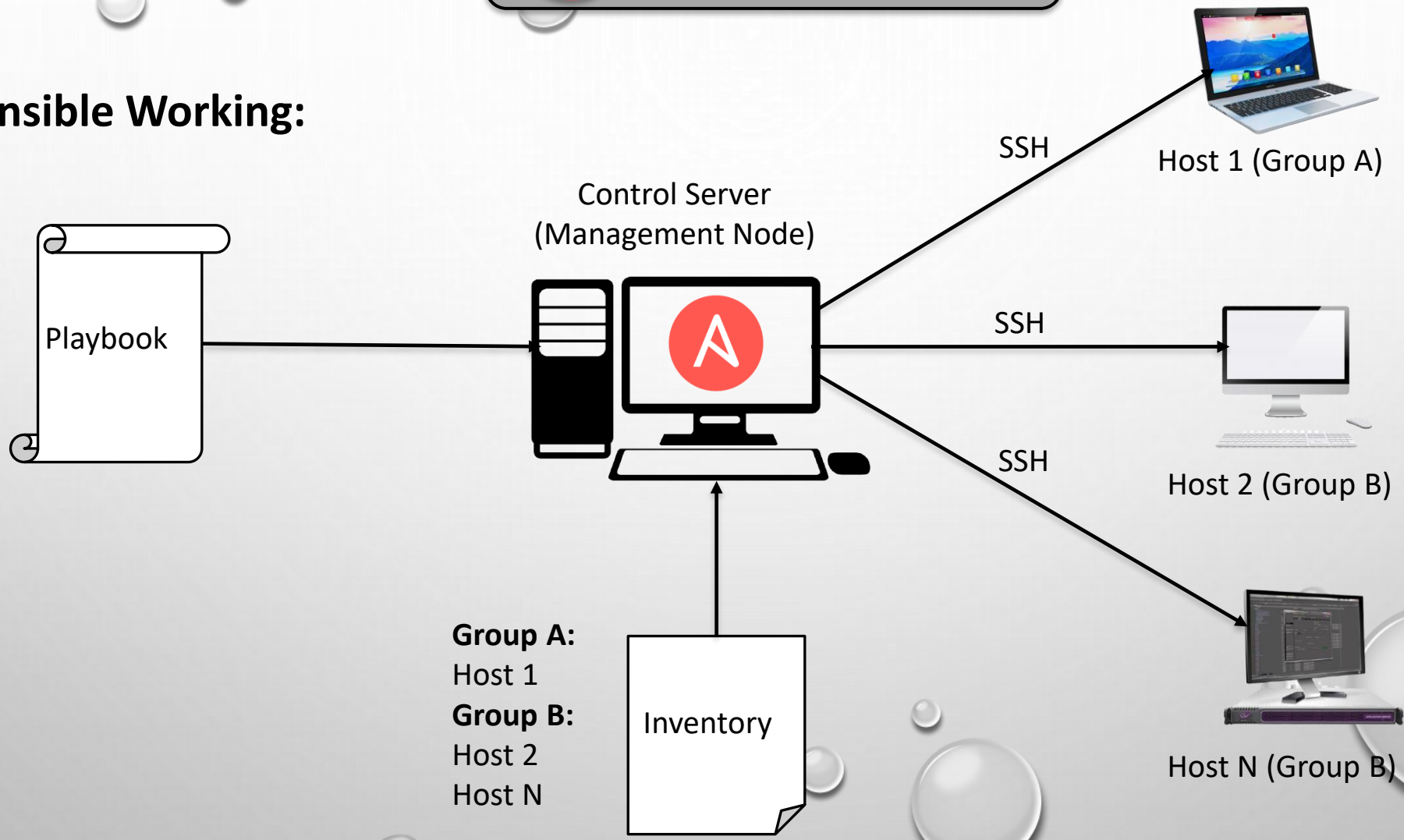


➢ Ansible Terms:

- ✓ **Controller Machine**: The machine where Ansible is installed, responsible for running the provisioning on the servers you manage.
- ✓ **Inventory**: An initialization file that contains information about the servers you are managing.
- ✓ **Playbook**: The entry point for Ansible provisioning, where the automation is defined through tasks using YAML format.
- ✓ **Task**: A block that defines a single procedure to be executed, e.g. Install a package.
- ✓ **Module**: A module typically abstracts a system task, like dealing with packages or creating and changing files. Ansible has a multitude of built-in modules, but you can also create custom ones.
- ✓ **Role**: A pre-defined way for organizing playbooks and other files in order to facilitate sharing and re-using portions of a provisioning.
- ✓ **Play**: A provisioning executed from start to finish is called a play. In simple words, execution of a playbook is called a play.
- ✓ **Facts**: Global variables containing information about the system, like network interfaces or the OS.
- ✓ **Handlers**: Used to trigger service status changes, like restarting or stopping a service.

Ansible

➢ **Ansible Working:**



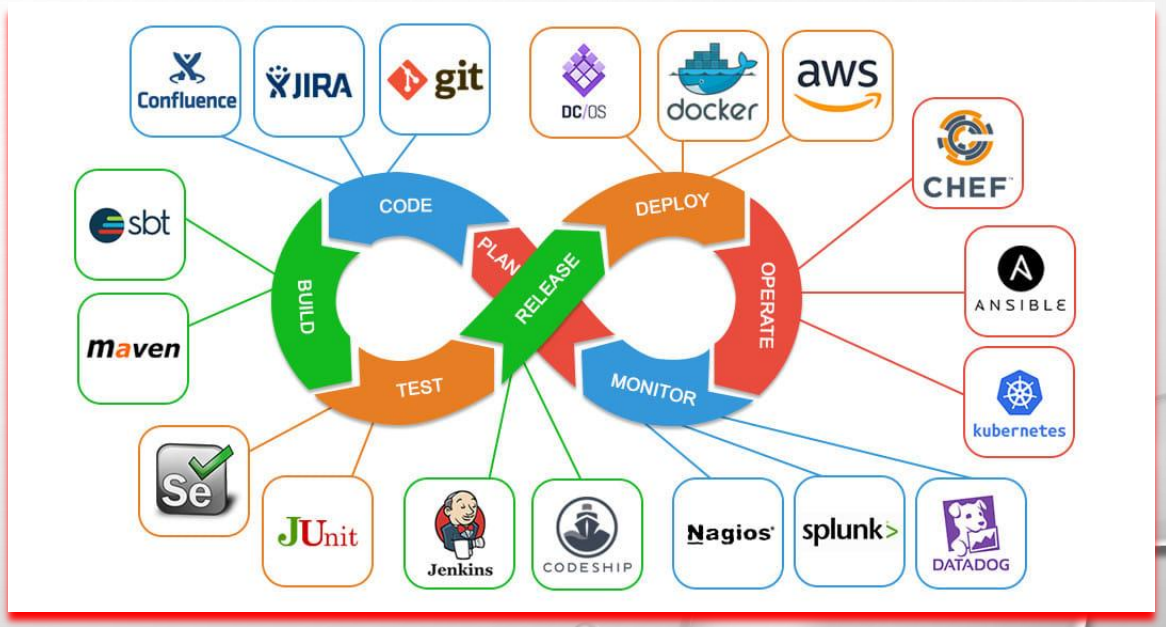
Ansible

➢ **Role of Ansible In DevOps:**

✓ Ansible is a popular configuration management tool that plays a key role in DevOps. It is used in IT Infra to automate the deployment, configuration and management of software applications and systems.

✓ **IaaC:** Ansible enables you to define your infrastructure as code (IaaC), which means you can use playbooks to define how your infrastructure should be configured & managed. This enables you to version control your infrastructure, collaborate on chan ges and automate the deployment of infrastructure changes.

✓ **Configuration Management:** Ansible helps automate the configuration of servers, networks, and other devices. This includes tasks such as installing and configuring software, updating system configurations, and managing users and permissions.



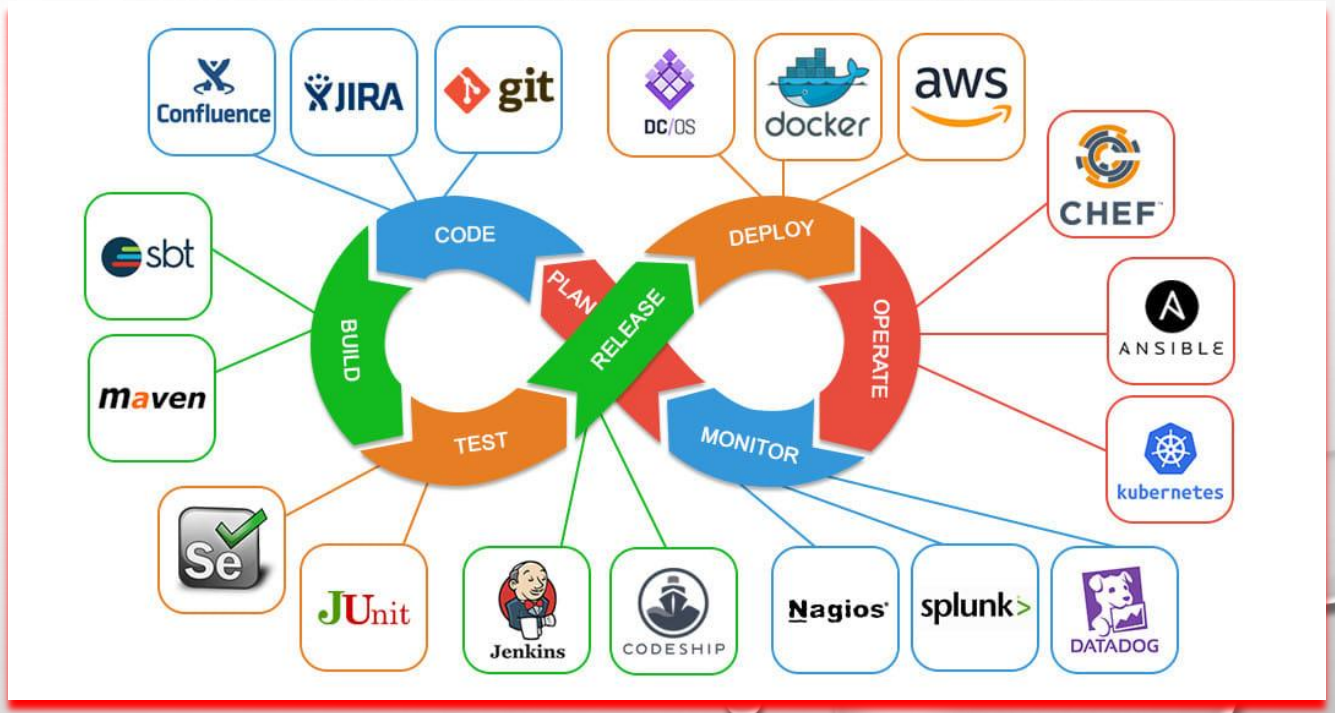
Ansible

➢ **Role of Ansible In DevOps:**

✓ **Continuous Deployment**: Ansible can help automate the deployment of new releases or changes to production environments. This can help reduce the risk of errors & increase the speed at which changes can be deployed.

✓ **Orchestration:** Ansible can help orchestrate the deployment & configuration of multiple systems, making it easier to manage complex systems and infrastructure.

✓ **Monitoring:** Ansible can help with monitoring, by automating the setup & configuration of monitoring tools, and triggering alerts based on predefined conditions.



Ansible

➢ **Role of Ansible In DevOps:**

Overall, Ansible helps DevOps teams to automate repetitive tasks, improve efficiency, and reduce the risk of errors in deployments. It also helps to ensure that infrastructure and applications are consistent and can be managed more effectively.