
MPRASHANT ACADEMY

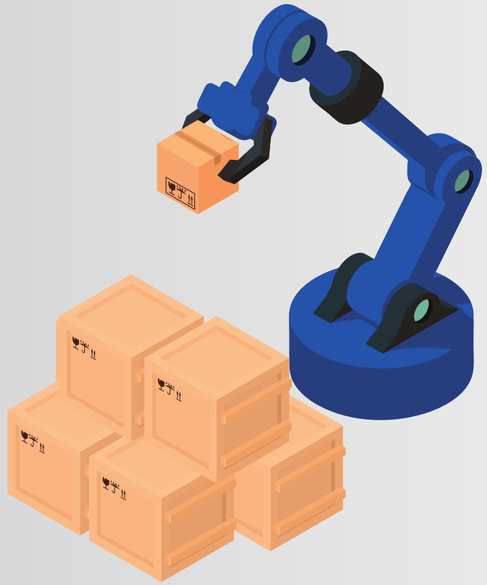

ANSIBLE

- What is ansible?
- Why we use it?
- Use case of ansible
- Features of ansible
- Ansible architecture and it's components
- Example of config files

What is Ansible?

ANSIBLE

An IT Automation Tool



Why do we need to Automate?

ServerA

ServerB

ServerC

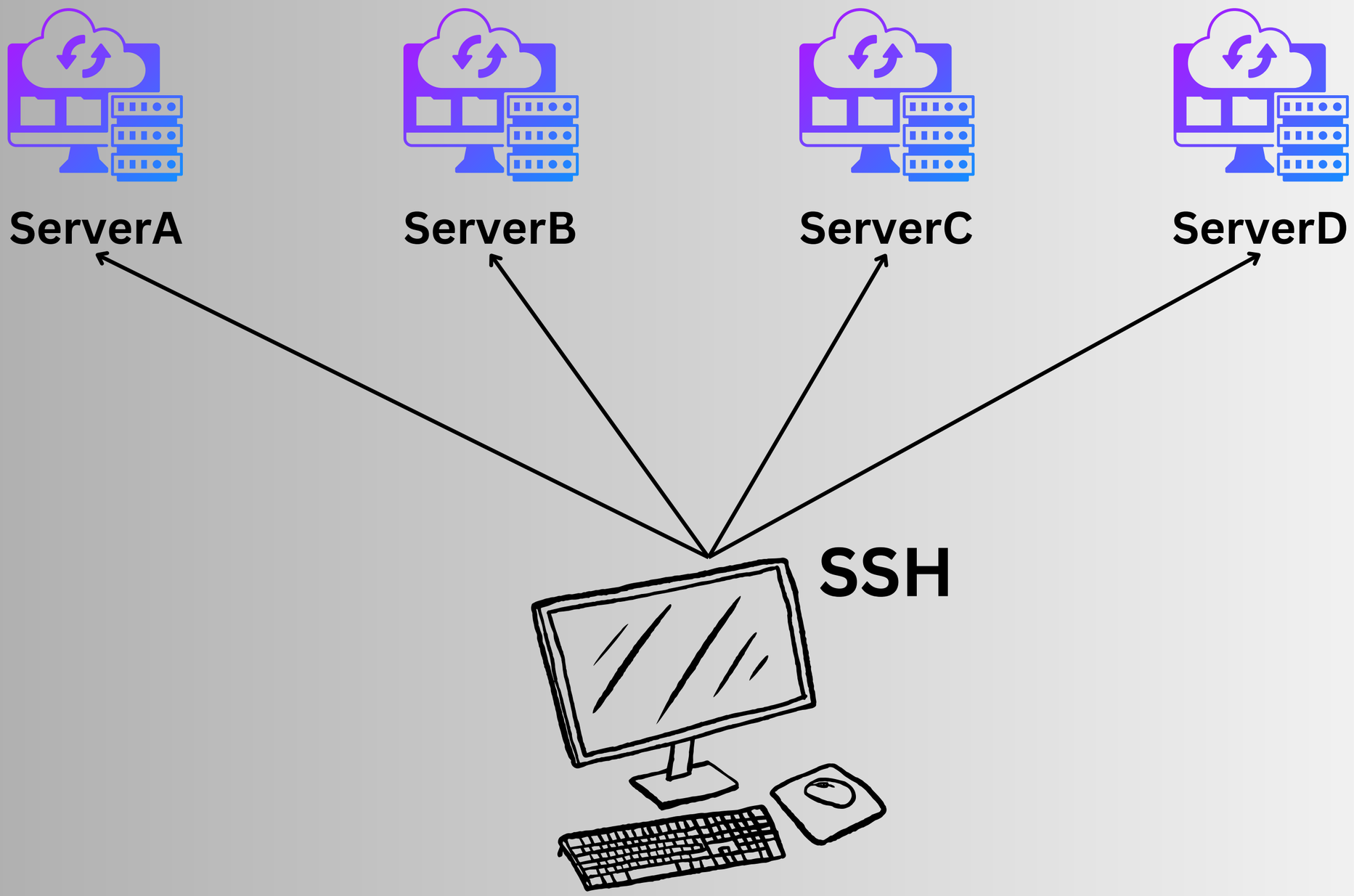
ServerD

ServerA

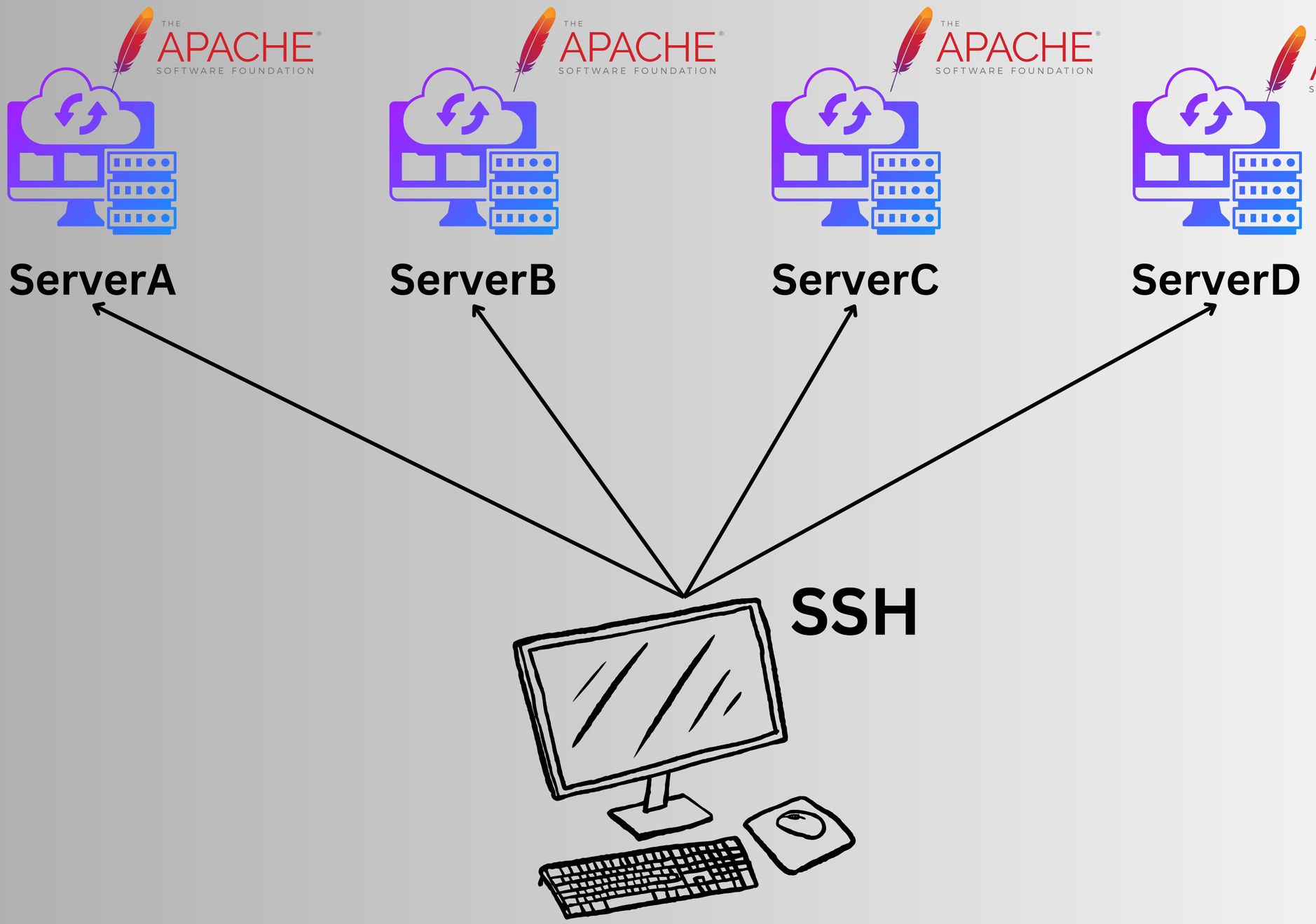
ServerB

ServerC

ServerD


ServerA
ServerB
ServerC
ServerD
SSH


MPRASHANT ACADEMY


THE APACHE SOFTWARE FOUNDATION
THE APACHE SOFTWARE FOUNDATION
THE APACHE SOFTWARE FOUNDATION
THE APAC
SOFTWARE FOU
ServerA
ServerB
ServerC
ServerD
SSH


MPRASHANT ACADEMY

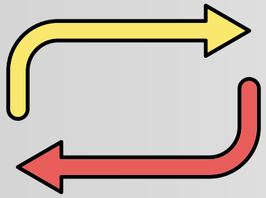
Problem with this approach:

- Time consuming

Problem with this approach:

- Time consuming 
- Chances of error 

Problem with this approach:

- Time consuming 
- Chances of error 
- Repetitive task 

What is the solution now?

ANSIBLE

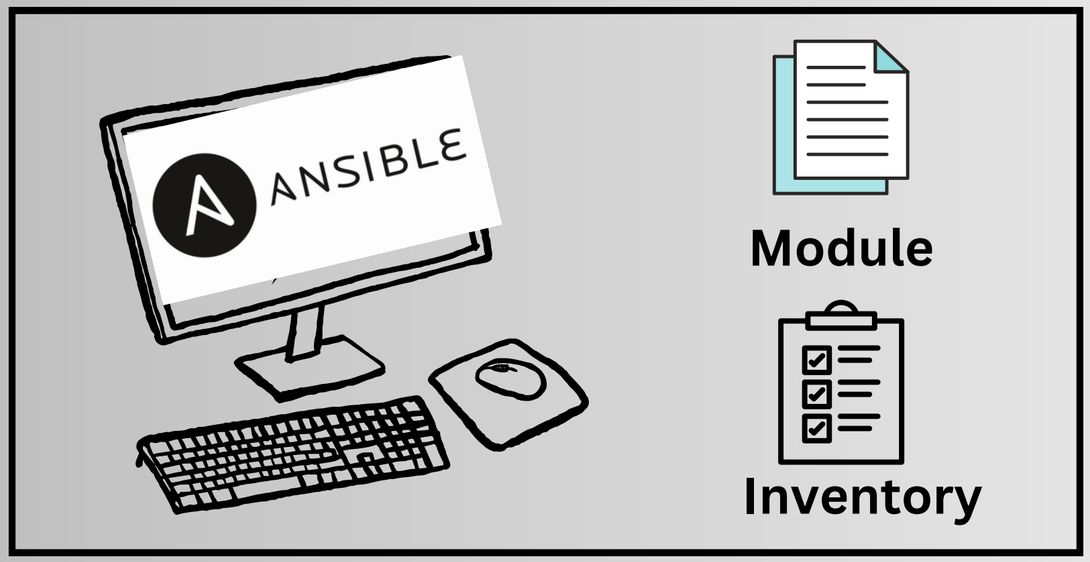
ServerA

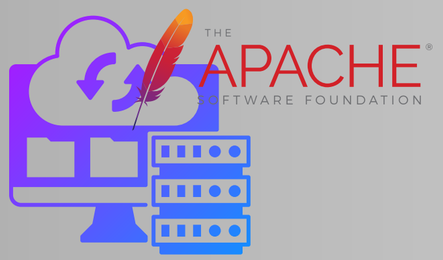
ServerB

ServerC

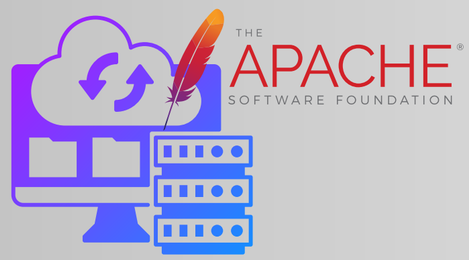
ServerD

Local Computer

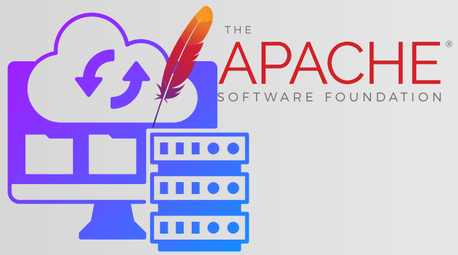




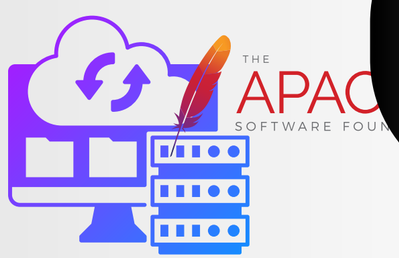
ServerA



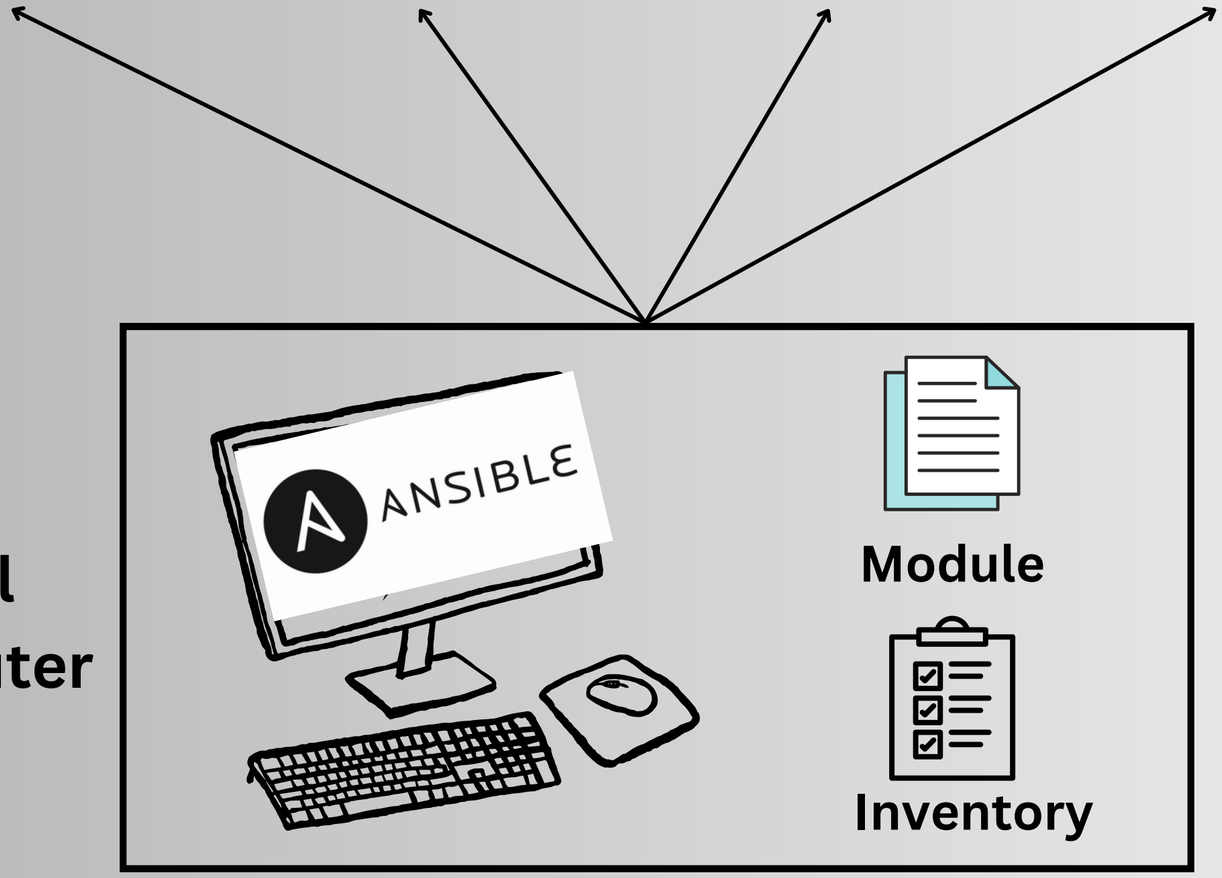
ServerB



ServerC



ServerD



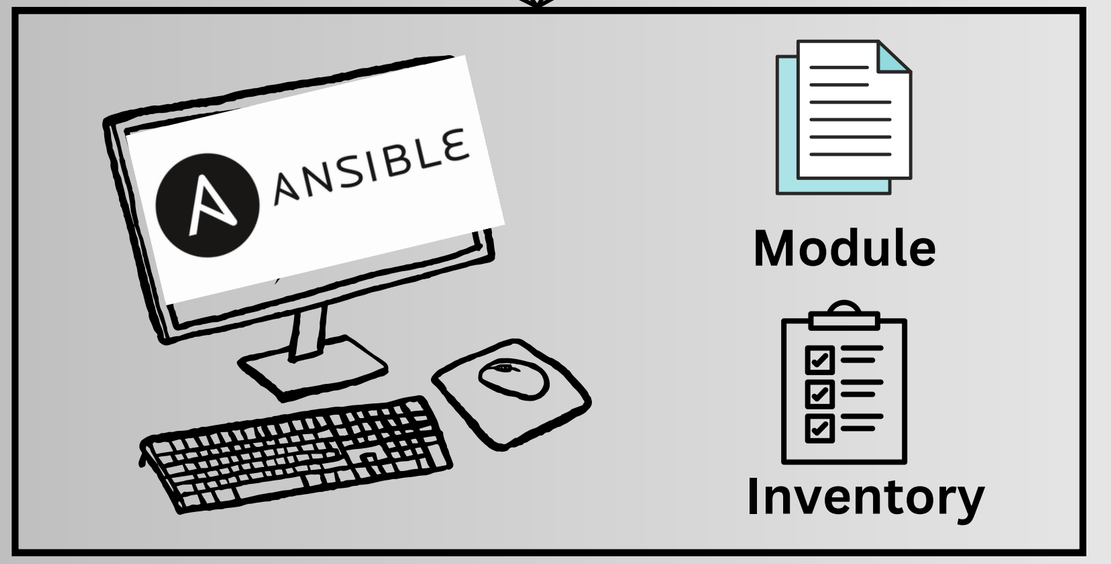
Local
Computer

ServerA

ServerB

ServerC

ServerD



Let's talk about some advantages of using ANSIBLE

Advantages of using Ansible

- **Simple and easy to use:** Ansible uses a simple and easy-to-learn language (YAML) to define playbooks, which makes it easy for anyone to use, even those with little or no programming experience.

- **Agentless architecture:** Ansible does not require any agents to be installed on remote systems, which makes it easy to set up and use.

- **Configuration management:** Ansible can be used to automate configuration management tasks such as provisioning, application deployment, and infrastructure management.

- **Scalability:** Ansible can manage a large number of systems simultaneously, making it ideal for large-scale deployments.

- Ansible playbooks can be run multiple times without changing the system state.
- **Open-source:** Ansible is an open-source tool, which means it is free to use and has a large community of contributors who regularly contribute to its development.
- **Integration with other tools:** Ansible can be integrated with other tools such as Docker, Kubernetes, and AWS, which makes it versatile and easy to use in a variety of environments.

How Ansible Works?

ANSIBLE

Inventory

Playbook

Module

Module

Small Programs to do a task

Start docker

Start a server

Install Nginx

Module

Create a file

Small Programs to do a task

Upgrade

Install Nginx

Uses

For writing configuration files because it's easy to read, write and understand


```yaml
# Dictionary of employees
employees:
  john:
    age: 30
    department: sales
  jane:
    age: 25
    department: marketing
```

Inventory

```
[webservers]
192.168.1.100 ansible_user=ubuntu ansible_password=secretpassword
192.168.1.101 ansible_user=ubuntu ansible_password=secretpassword
```

inventory.yml

Playbook

To install web server on remote server...

webserver.yml

```yaml
---
- name: Install Apache web server
  hosts: webservers
  become: true
  tasks:
    - name: Install Apache2
      apt:
        name: apache2
        state: present
```

Playbook

To install web server on remote server...

webserver.yml

```yaml
---
- name: Install Apache web server
  hosts: webservers
  become: true
  tasks:
    - name: Install Apache2
      apt:
        name: apache2
        state: present
```

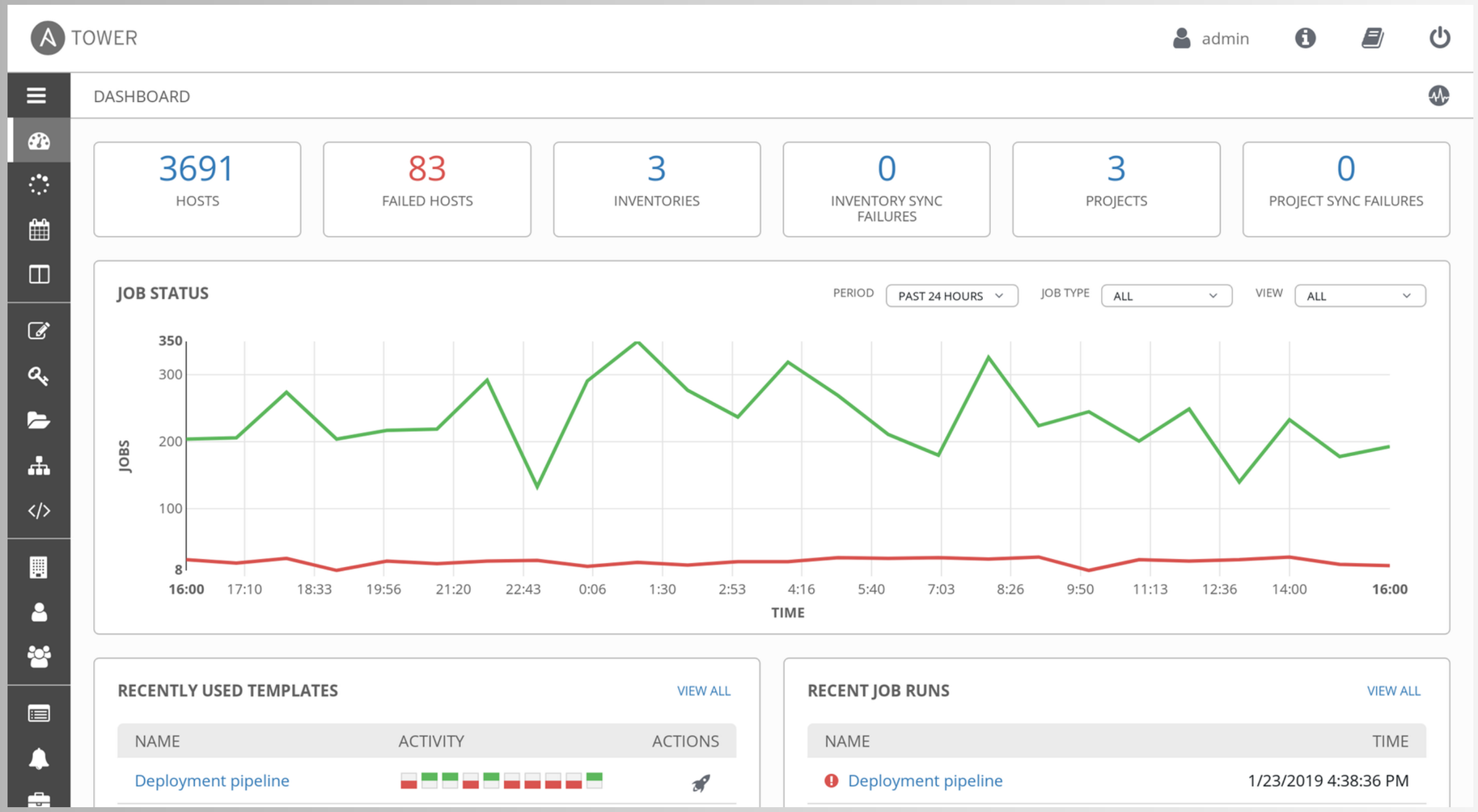
We can add multiple modules in a playbook

- Make directory
- Install apache
- Start webserver


```
ansible-playbook -i inventory.yml webserver.yml
```

RED HAT® ANSIBLE® Tower

- A web-based UI
- Centralized Management
- Job Scheduling
- Reporting and Analytics
- Integrate with ticketing system tool


TOWER
admin
DASHBOARD
3691
HOSTS
83
FAILED HOSTS
3
INVENTORIES
0
INVENTORY SYNC FAILURES
3
PROJECTS
0
PROJECT SYNC FAILURES
JOB STATUS
PERIOD
PAST 24 HOURS
JOB TYPE
ALL
VIEW
ALL
350
300
200
100
8
JOBS
16:00
17:10
18:33
19:56
21:20
22:43
0:06
1:30
2:53
4:16
5:40
7:03
8:26
9:50
11:13
12:36
14:00
16:00
TIME
RECENTLY USED TEMPLATES
VIEW ALL
NAME
ACTIVITY
ACTIONS
Deployment pipeline
RECENT JOB RUNS
VIEW ALL
NAME
TIME
Deployment pipeline
1/23/2019 4:38:36 PM


MPRASHANT ACADEMY

Compare to with other Tools like Puppet and Chef

- Both use DSL based on Ruby
- Use agent-based architecture
- Ansible is faster

Installation of Ansible

https://docs.ansible.com/ansible/latest/installation_guide/index.html

Ansible Config

/etc/ansible/ansible.cfg

/etc/ansible/hosts

Get a Sample Ansible Config

```
sudo ansible-config init --disabled -t
all > ansible.cfg
```

First Ansible Playbook

Playbook for Testing Connectivity on Localhost

Playbook for Print a Message on Terminal

Playbook for Installing and Starting a Package

Overview of Ansible Playbook


```yaml
---
- name: Install and start Nginx
  hosts: webservers  # Assuming 'webservers' is defined in your Ansible inventory
  become: true       # Tasks should be run with sudo privileges

  tasks:
  - name: Install Nginx
    yum:           # Using the yum module for package manageme module for package
      name: nginx    # Name of the package
      state: present # Ensure the package is installed

  - name: Start Nginx
    service:       # Using the service module to manage the service
      name: nginx
      state: started
      enabled: true  # Ensure Nginx is enabled to start on boot
```

Adding Cron JOb on Remote

```
0 * * * * /path/to/script.sh
```

Ansible Ad Hoc Tasks

ansible <host-pattern> -m <module> -a "<module arguments>"
-u <username> -b

ansible myserver -m command -a "df -h"

- ansible all -m ping

- ansible webservers -m copy -a "src=/path/to/local/file dest=/path/on/remote"

- ansible webservers -m service -a "name=httpd state=restarted"

- ansible all -m script -a "/path/to/local/script.sh"

- ansible all -m apt -a "name=vim state=present" # for Debian/Ubuntu
- ansible all -m yum -a "name=vim state=present" # for RHEL/CentOS

Ansible Conditions

Ubuntu

Redhat

ansible localhost -m setup

Ansible Built in Variables

Variables:

- ansible_hostname: The hostname of the target machine.
- ansible_os_family: The OS family (e.g., 'RedHat', 'Debian', 'Windows') of the target machine.
- ansible_distribution: The name of the distribution (e.g., 'Ubuntu', 'Fedora').
- ansible_distribution_version: The version of the distribution.
- ansible_architecture: The architecture of the target machine (e.g., 'x86_64').
- ansible_fqdn: The fully qualified domain name of the target machine.
- ansible_user_dir: The home directory of the user executing the Ansible playbook.
- inventory_hostname: The name of the current node being worked on, as known by Ansible's inventory.

Ansible Loops


```
- name: Install multiple packages
  apt:
    name: "{{ item }}"
    state: present
  with_items:
    - package1
    - package2
    - package3
```

Ansible Roles

An Ansible role is a structured way of grouping together various functionalities and making it easier to reuse and share common setup tasks.

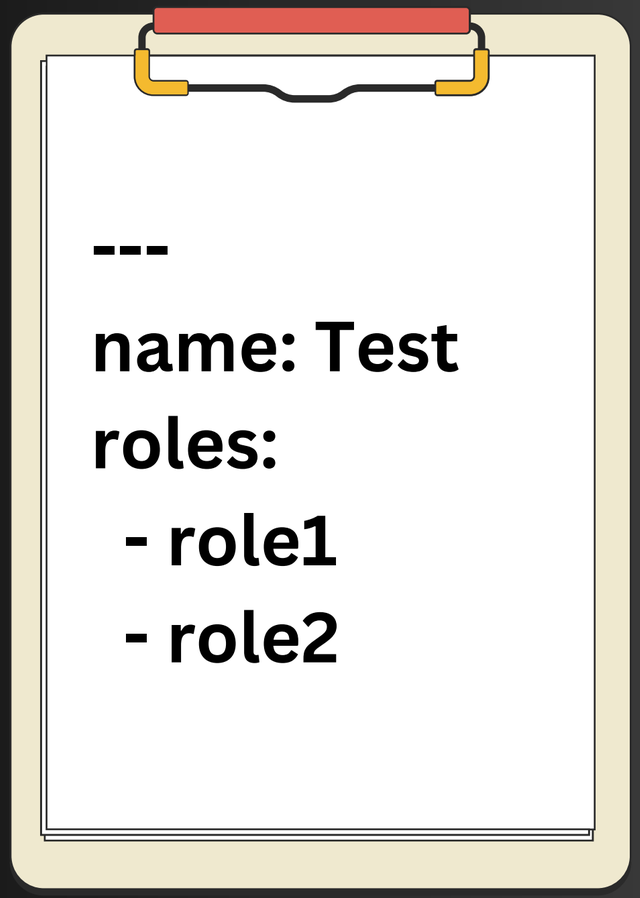
Ansible playbook

role1

role2

role3

playbook.yml



```
---
name: Test
roles:
  - role1
  - role2
```

Use Case

- Install apache httpd webserver on remote server
- Place our custom HTML File to use
- Start the service
- Enable service in firewall
- Reload the Firewall

ansible-galaxy init role_name


```
role_name/
    ├── defaults
    │   └── main.yml   # Default variables for the role
    ├── handlers
    │   └── main.yml   # Handlers, which can be used by this role or even anywhere outside this role
    ├── meta
    │   └── main.yml   # Metadata for the role, including author, support details, and dependencies
    ├── README.md      # Optional: A human-readable description of the role and its requirements
    ├── tasks
    │   └── main.yml   # The main list of tasks that the role executes
    ├── templates
    │   └── ...        # Template files, which use Jinja2 templating language
    ├── tests
    │   ├── inventory  # Inventory file for testing the role
    │   └── test.yml   # Test playbook for the role
    └── vars
        └── main.yml   # Other variables for the role
```