# Create Ansible Playbook

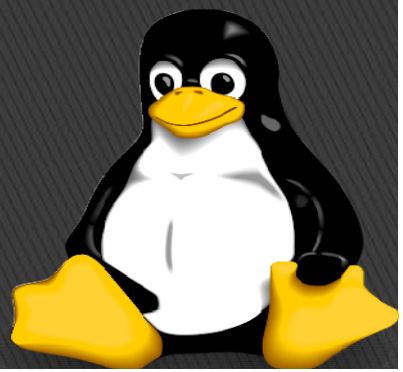
# What Is playbook?

- **Ansible Playbooks** are the way of sending commands to remote systems through scripts. Ansible playbooks are used to configure complex system environments to increase flexibility by executing a script to one or more systems.
- Playbook push configuration Ansible what to execute.
- Playbook contains a list of task
- Playbooks contain the steps which the user want to execute on a particular machine.
- Playbooks are run sequentially.
- All playbooks written in yaml format
- Playbook define set of play with activates to run on hos
- A task is a single action perform on a host

# How to create a Ansible Playbooks?

- Create a playbook with a file name, not necessary to have the filename with any extension. but create with extension with ".yml" else you wont find the difference of normal text file anda playbook.

- The file starts with: –



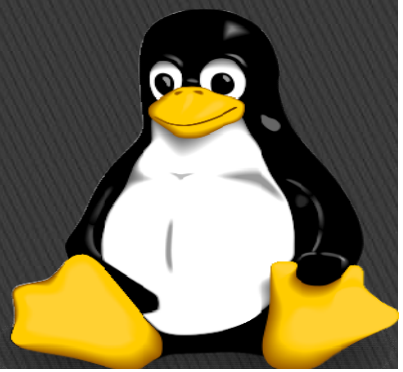
# What is Yaml?

YAML is a human-readable data serialization standard that can be commonly used for configuration files and in applications where data is being stored or transmitted.

YAML was said to mean *Yet Another Markup Language*, referencing its purpose as a markup language with the yet another construct, but it was then repurposed as *YAML Ain't Markup Language*, a recursive acronym, to distinguish its purpose as data-oriented, rather than document markup

Extension

- .yml
- .yaml



# YAML is very sensitive

- We should be more cautious with the spaces while writing a ansible playbook. Tabs are notallowed here.
- Character should start exactly or after the previous parameter starts and itshould looks like a statement as shown in the above examples.

Basically Playbooks consists of three sections

1. Host declaration
2. Variable declaration (optional)
3. Action / Tasks Declaration

1. Host declaration - It defines that on which server groups playbook should run based on the ansible inventory file.
2. Variable declaration (optional) -is optional, we see some examples how to declare a variable below.
3. Action / Tasks Declaration -You can also use other par like, name, actions, notify, remote users depend on th requirement.

# Write yaml sample file

```
#vim        yaml-example.yml
Person:
    - name: “Vijay”
      age: 30
       gender: “male”
    - name: “ajay”
      age: 32 #1/10/1990
       gender: “male”
     -hobbies:
       -cricket
       -hockey
       -reading
     -hobbies: {“cricket”,”hockey”,”readng”]
     -friends
       - name: “sara”
         age: 25
       -{name:  “harry”, age:27}
```

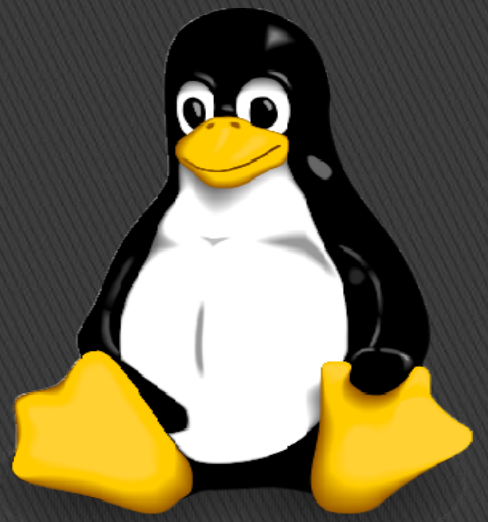
# Yaml program(playbook hosts)

```
#vim    test.yml
-
    name: Play1
    hosts: web
    tasks:
            - name: “checking date”
              command: date
```

# Verify the playbook for syntax errors:

- check the syntax error, run the following command. If it finds no error, it only shows the given file name. If it detects any error, you will get an error as follows,

```
$ansible-playbook test.yml --syntax
```



# Playbook Execute

Syntax:

$ansible-playbook <playbook name> -i <inventory file>

Eg

$ansible-playbook test.yml -i /etc/ansible/hosts

# Understanding the Ansible Playbooks Terminology

- **Control Node:** The machine where Ansible is installed. It is responsible for managing client nodes.
- **Managed Nodes:** List of hosts managed by the control node
- **Playbook:** A Playbook file contains a set of procedures used to automate a task.
- **Inventory:** The inventory file contains information about the servers you manage.
- **Task:** Each play has multiple tasks, tasks that are executed one by one against a given machine (it a host or multiple host or a group of host).
- **Module:** Modules are a unit of code that is used to gather information from the client node.
- **Role:** Roles are ways to automatically load some vars_files, tasks, and handlers based on known file structure.
- **Play:** Each playbook has multiple plays, and a play is the implementation particular automation from beginning to end.

# How to Understand Ansible Output

The Ansible Playbook output comes with 4 colors, see below for color definitions.

- Green: ok – If that is correct.
- Yellow: changed – Specific data has updated or modified according to the needs of the tasks.
- Red: FAILED – If there is any problem while doing a task, it returns a failure message.
- White: It comes with multiple parameters

# Yaml program(playbook hosts)

```
#vim    test1.yml
-
   name: Play1
   hosts: web
   tasks:
              -  name: “checking date”
                 command: date

-
  name: Play2
  hosts: web
  tasks:
            -  name: “installing httpd”
               yum:
                          name: httpd
                          state: present
            -name :”start httpd  server”
             service:
                          name: httpd
                          state : started
```

# Yaml program(as a root)

```
#vim    test2.yml
-
   name: Play1
   hosts: web
   remote_user: root
   tasks:
              -  name: “checking date”
                 command: date

-
  name: Play2
  hosts: web
  tasks:
            -  name: “installing httpd”
               yum:
                           name: httpd
                           state: present
           -name :”start httpd  server”
            service:
                           name: httpd
                           state : started
```

# Copying Files from Local to Remote Linux

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:

    - name :  copy file local to remote
      copy:
        src : /home/info
        dest : /mnt/info
        owner : sachin
        group : imbgrp
        mode : '0644'
```

# Perform file permission

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:

    - name:  File permission
      file:
          path: /home/pune
          owner : sachin
          group: imbgrp
          mode: '0644'
```

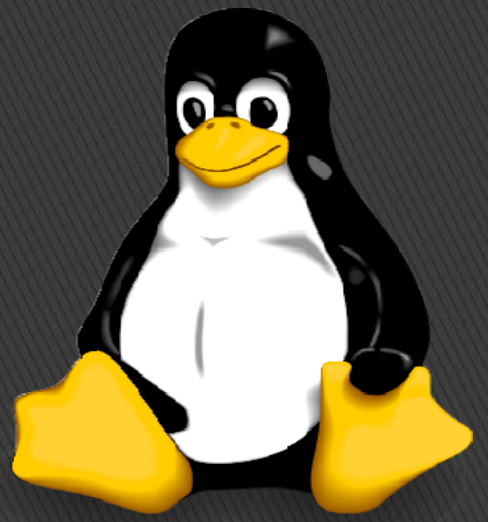
# Create directory

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:

    - name:  create directory with perm
      file:
          path: /home/database
          state: directory
          mode: '0777'
```

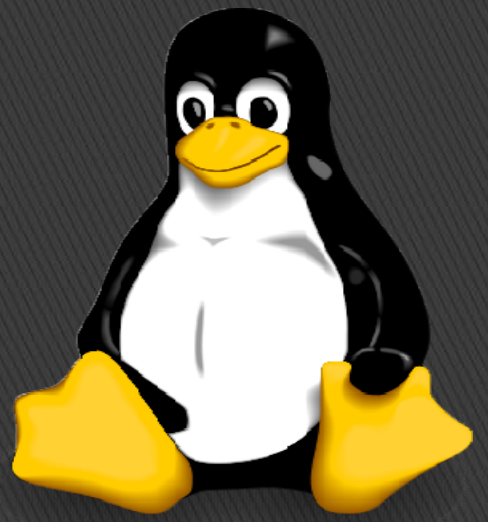
# Delete File and directory

- name:  remove file or directory
  file:
    path: /home/doc.txt
    state: absent
- name:  remove directory
  file:
    path: /etc/india
    State: absent



# Manipulate Files in Linux

- name:  change selinux mode from file
  lineinfile:
    path: /etc/selinux/config
    regexp: ‘^SELINUX=‘
    line: SELINUX=disabled



# Add line into file

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:
    - name:  add line into file
      lineinfile:
      path: /etc/hosts
      line: 192.168.43.49   client1.example.com
      create: yes
```

# Install & Enable Web Server

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:
    - name: install Apache server
      yum:
            name: httpd
            state: latest

    - name: enable and start Apache server
      service:
            name: httpd
            enabled: yes
            state: started
```

# Start and enable firewall service

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:
        - name: firewalld enabled and running
          service:
              name: firewalld
              enabled: true
              state: started
```

# Open Firewall Port

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:

    - name: open firewall port
      firewalld:
        service: http
        immediate: true
        permanent: true
        state: enabled
```

# Directory Permission

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:

      - name: set content directory group/permissions
        file:
            path: /var/www/html
            owner: root
            group: web
            state: directory
            mode: u=rwx,g=rwx,o=rx,g+s
```

# Create tar Archive

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:
      - name: Compress Directory contents
        archive:
              path: /usr/sbin
              dest: /mnt/backup.tar
              format: tar
```

# Add user account

```
-
  name: Play1
  hosts: web
  remote_user: root
  tasks:

    - name: Add a simple user called harsh
      user:
        name: harsh
        comment: harshad
```

# Add group account

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:

        - name: Add a group called developer
          group:
                name: developer
                state: present
```

# Add user into group

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:

        - name: Add a user john and add them to
    a group developer
            user:
                    name: john
                    groups: developer
                    append: yes
```

# Delete user account

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:
            - name: Remove janedoe
              user:
                  name: janedoe
                  state: absent
                  remove: yes
```

# Remove group Account

```
-
    name: Play1
    hosts: web
    remote_user: root
    tasks:
        - name: Remove developer group
          group:
                name: developer
                state: absent
```