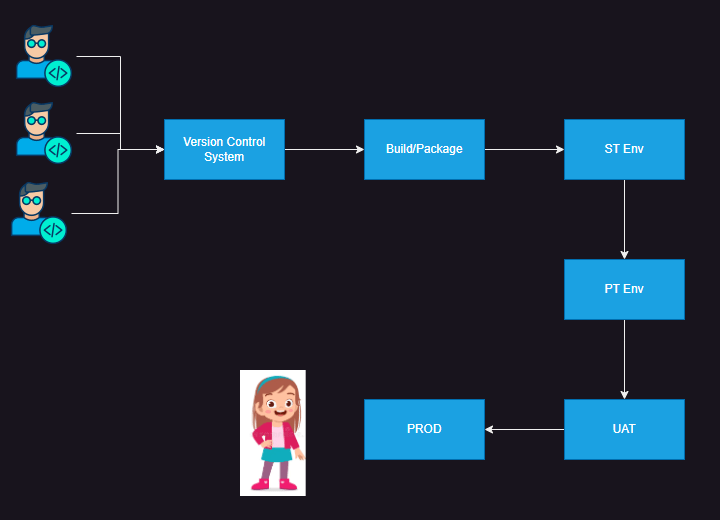
October 29, 2024

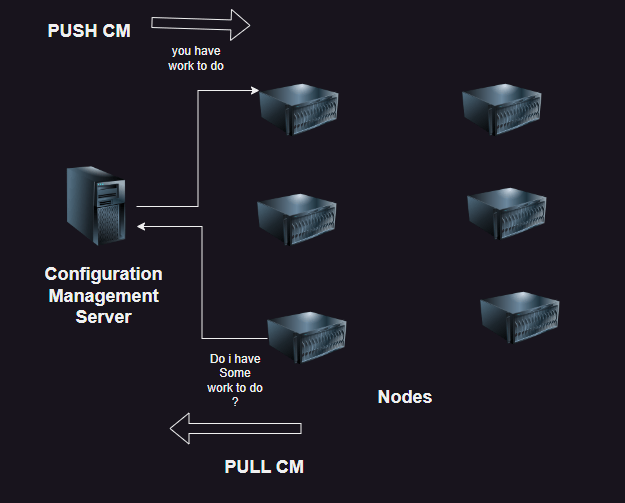
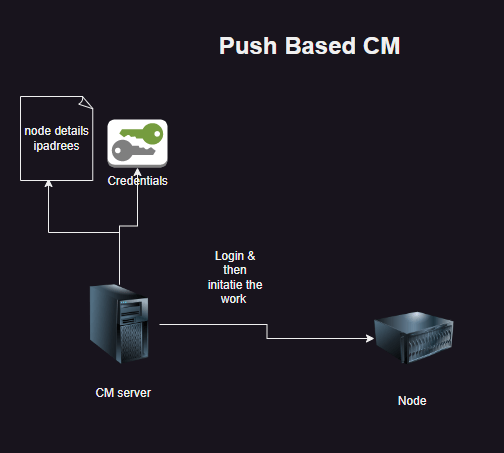
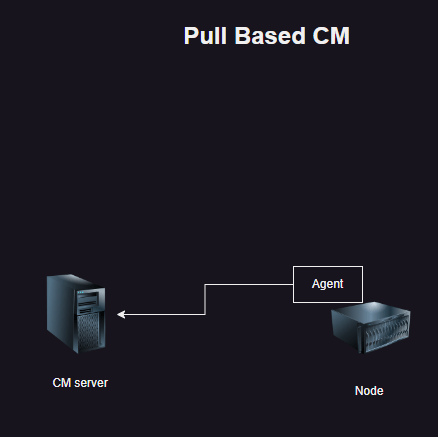
DevOps Classroom notes 29/Oct/2024

**Overview of CI/CD Pipelines**

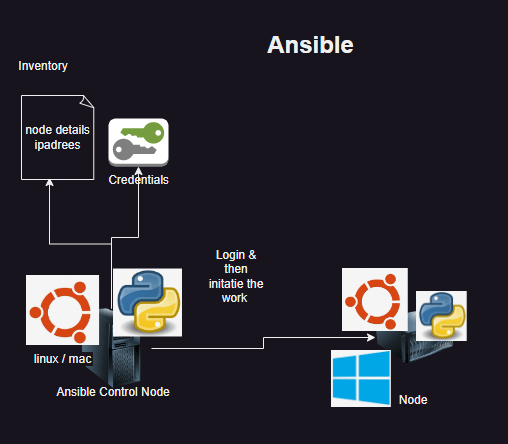
CI/CD services

* Basic Pipeline  
  
* In each Enviroment such as System [Test](https://directdevops.blog/2024/10/29/devops-classroom-notes-29-oct-2024/) or Performance Test or Load Test environment, expectation is
  + Ensure the application is deployed and running
  + We execute the [Automated Tests](https://directdevops.blog/2024/10/29/devops-classroom-notes-29-oct-2024/) and showcase the results
* Relevance of Configuration Management:
  + This comes into play when your application is deployed on Physical or virtual machines

**Configuration Management (CM)**

* This is used to configure applications or setups on servers.
* Where is it used
  + deploying applications on servers
  + network automation
* How does this work ?  
  
* Models in Configuration Managment
  + PUSH BASED CM:
    - Examples:
      * Ansible
      * Salt
  + PULL BASED CM
    - Examples
      * Chef
      * Puppet
  + HYBRID (Support Both)
    - Examples:
      * Powershell DSC
* PUSH BASED CM (Design Considerations):
  + In this case CM Server needs to maintain list of all node details like (name, ipaddress etc)
  + It also needs credentials  
    
* PULL BASED CM
  + In this agent needs to be installed on node
  + No credentials are required  
    

**Ansible**

* Ansible is simple to use,
* Ansible works in two ways
  + directly execute commands
  + write yaml files (playbooks) to define what you want.
* Ansible is an opensource software developed in python [Refer Here](https://github.com/ansible/ansible)
* Ansible is also widely used in Network and Datacenter Automations
* In the world of Ansible
  + CMServer => Ansible control node
  + list of node and ip details => inventory
  + Ansible uses python to work  
    

**Understanding Ansible Workflow**

* Goal: Deploy nginx on a server
* Manual activities:

sudo apt update

sudo apt install nginx -y

* Ansible’s approach (Declarative):
* I will create a playbook

---

- name: install nginx

hosts: all

become: yes

tasks:

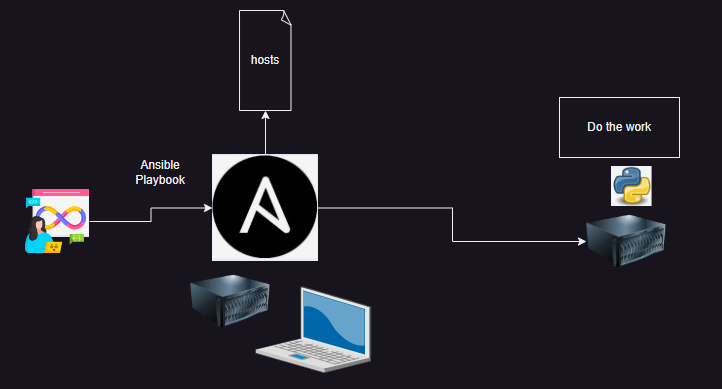
- name: install and update nginx

apt:

name: nginx

update\_cache: yes

state: present



**Goals**

1. Try doing some manual deployments
2. Learn YAML
3. Automate manual deployments by writing playbooks
4. Write Ansible Playbooks effeciently

October 30, 2024

# DevOps Classroom notes 30/Oct/2024

# Deployment of a Simple Website

* We need a linux [server](https://directdevops.blog/2024/10/30/devops-classroom-notes-30-oct-2024/)
  + This can be an [azure](https://directdevops.blog/2024/10/30/devops-classroom-notes-30-oct-2024/) vm or aws ec2 instance
* We need to install webserver software
  + apache
  + nginx
* We need to copy the website [Refer Here](https://www.free-css.com/assets/files/free-css-templates/download/page295/antique-cafe.zip) to particular
  + /var/www/html

### Manual Steps

* Install nginx

sudo apt update

sudo apt install nginx -y

* Now webserver should be installed

sudo systemctl status nginx

* Now we can access the webserver over http => http://publicip
* Now lets download the website into /tmp

cd /tmp

wget https://www.free-css.com/assets/files/free-css-templates/download/page295/antique-cafe.zip

* Install unzip

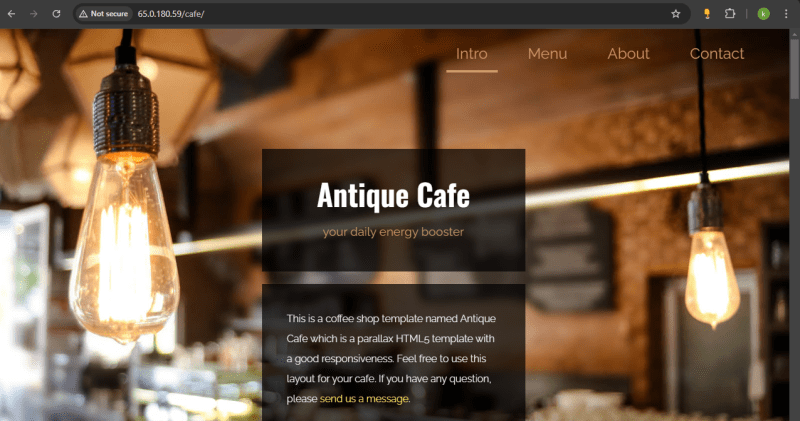
sudo apt install unzip -y

* Now unzip the antique-cafe

unzip antique-cafe.zip

* Now lets copy the folder 2126\_antique\_cafe to /var/www/html/cafe/

sudo mv 2126\_antique\_cafe/ /var/www/html/cafe

* Now access http://<public-ip>/cafe  
  

# Possible Options to automate the website deployment

1. Shell Scripts:
2. Procedural in nature (i.e. define how it has to be done)
3. Are not idempotent by default
4. Solution:

#!/bin/bash

WEB\_DESIGN\_ZIP='https://www.free-css.com/assets/files/free-css-templates/download/page295/antique-cafe.zip'

SITE\_NAME='cafe'

sudo apt update

sudo apt install nginx -y

cd /tmp

wget $WEB\_DESIGN\_ZIP

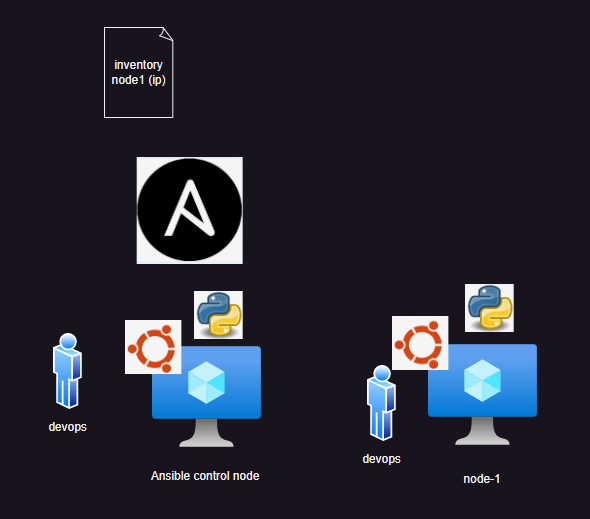
sudo apt install unzip -y

unzip antique-cafe.zip

sudo mv 2126\_antique\_cafe/ /var/www/html/$SITE\_NAME

1. Configuration Management
2. Declarative in nature (what we want )
3. Most of the ansible is idempotent

# Setup Ansible (on Azure VM)

* Installation overview  
  
* For guided creation watch classroom video
* Installation [Refer Here](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html) and [Refer Here](https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html) for distro based installation
* [Refer Here](https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu) for ubuntu based steps

sudo apt update

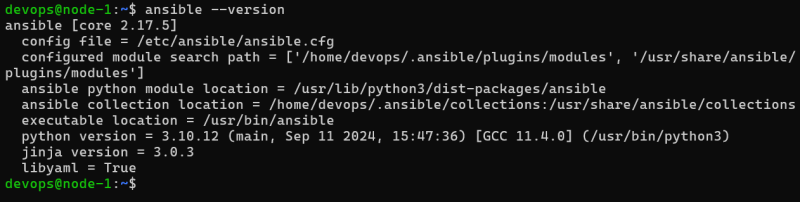
sudo apt install software-properties-common

sudo add-apt-repository --yes --update ppa:ansible/ansible

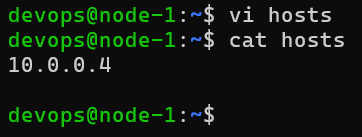
sudo apt install ansible -y

* To verify the installation

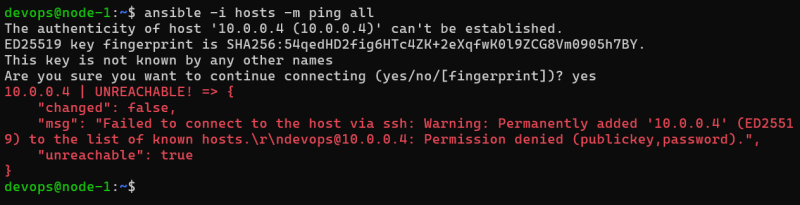
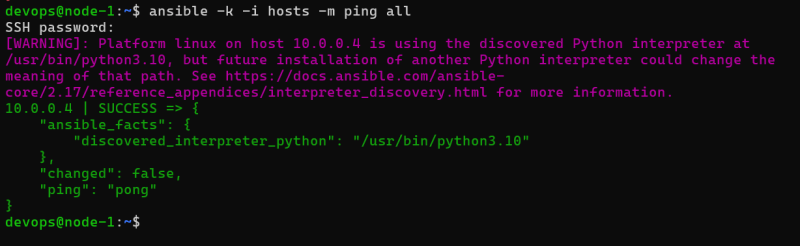
ansible --version



### Ansible ping test

* This is used to check if the ansible control node can communicate with the node or not
* Create an inventory file with node ip in it  
  
* Now execute the following command

ansible -i hosts -m ping all

  
\* In the above case we have not passed credentials  


### Share this:

* [Click to share on Twitter (Opens in new window)](https://directdevops.blog/2024/10/30/devops-classroom-notes-30-oct-2024/?share=twitter&nb=1)
* [Click to share on Facebook (Opens in new window)](https://directdevops.blog/2024/10/30/devops-classroom-notes-30-oct-2024/?share=facebook&nb=1)
* [Click to share on WhatsApp (Opens in new window)](https://directdevops.blog/2024/10/30/devops-classroom-notes-30-oct-2024/?share=jetpack-whatsapp&nb=1)

October 31, 2024

# DevOps Classroom notes 31/Oct/2024

# Ansible configuration (password less)

* Lets create two vms
* create a user in [ansible](https://directdevops.blog/2024/10/31/devops-classroom-notes-31-oct-2024/) control node called as devops and same on node

sudo adduser devops

* give admin permissions password less for devops
* generate a key pair on ansible control node
* copy the public key to the node
* now try connecting using [ssh](https://directdevops.blog/2024/10/31/devops-classroom-notes-31-oct-2024/) without password
* INstall ansible on control node and do what we did yesterday

### AWS

* On both node add user devops

sudo adduser devops

* AWS will not allow password authentication by default, so we need to change the configuration rules
* Change the value of PasswordAuthentication to yes in /etc/ssh/sshd\_config.d/60-cloudimg-settings.conf
* restart sshd service

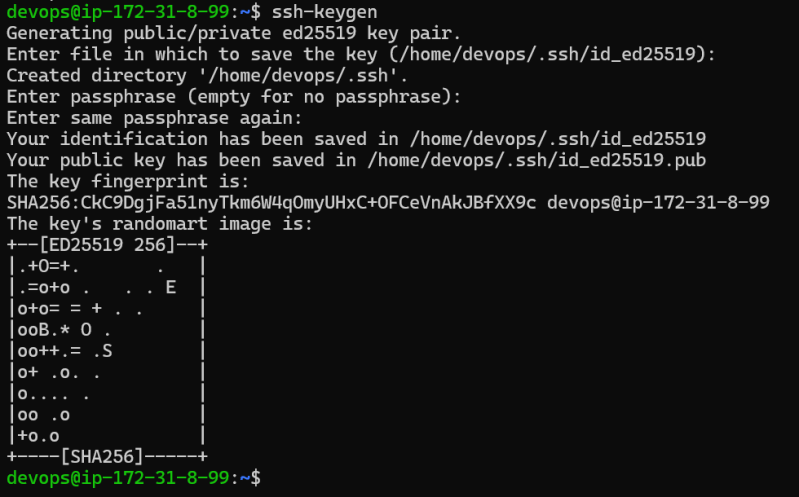
sudo systemctl restart ssh.service

* We need to grant admin permissions to devops user

sudo visudo

* Now add the following line

devops ALL=(ALL:ALL) NOPASSWD:ALL

* Now lets configure key based authentication between ansible control node and node 1
  + Switch to devops user su devops
  + Create a key pair ssh-keygen on ansible control node  
    
  + Now copy the public key into node1 ssh-copy-id devops@<node-1-ip>
  + after this step we are up for passwordless authentication
* Lets install ansible on ansible control node

sudo apt update

sudo apt install software-properties-common

sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible -y

* Now create a hosts file and perform ansible ping test

### Share this:

November 2, 2024

# DevOps Classroom notes 02/Nov/2024

# YAML

* YAML is a file format which is designed over name value pairs
* YAML is used to represent data
* YAML files have extension of .yml or .yaml
* Any thing in yaml

<name>: <value>

* Data is categorized into following types
  + number:  
    yaml  
    experience: 4
  + text  
    yaml  
    designation: DevOps Engineer
  + boolean  
    yaml  
    certifications: yes
  + list/array  
    “`yaml  
    tools:</li>  
    <li>ansible</li>  
    <li>docker</li>  
    <li>k8s  
    “`
  + map/object/dictionary  
    yaml  
    address:  
    flatno: 601  
    building: nilgiri  
    landmark: ameerpet metro  
    city: hyderabad
* Scalar or simple types
  + number
  + text
  + boolean
* Complex
  + list
  + map
* Example 1 : [Ansible](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/) Playbook

---

- name: Install and configure Nginx web server

hosts: webservers

become: true

gather\_facts: yes

tasks:

- name: Display OS information

debug:

var: ansible\_distribution

- name: Update apt cache

apt:

update\_cache: yes

- name: Install Nginx web server

apt:

name: nginx

state: latest

- name: Start Nginx service

service:

name: nginx

state: started

* Structure:
  + list(map/object)
    - name: string
    - become: boolean
    - gather\_facts: boolean
    - tasks: list(map/object)
      * name: string
      * debug:
* Example

trigger: #list(text)

- main

pool: #map/object

vmImage: 'ubuntu-latest' #text

stages: # list(map/object)

- stage: Build

displayName: 'Build Stage'

jobs:

- job: BuildJob

displayName: 'Build Job'

steps:

- script: echo "Building the application..."

displayName: 'Build Step'

- task: DotNetCoreCLI@2

inputs:

command: 'restore'

projects: '\*\*/\*.csproj'

- task: DotNetCoreCLI@2

inputs:

command: 'build'

projects: '\*\*/\*.csproj'

arguments: '--configuration Release'

- stage: Test

displayName: 'Test Stage'

jobs:

- job: TestJob

displayName: 'Test Job'

steps:

- script: echo "Running tests..."

displayName: 'Test Step'

- task: DotNetCoreCLI@2

inputs:

command: 'test'

projects: '\*\*/\*.csproj'

arguments: '--configuration Release --no-build'

- stage: Deploy

displayName: 'Deploy Stage'

jobs:

- job: DeployJob

displayName: 'Deploy Job'

steps:

- script: echo "Deploying the application..."

displayName: 'Deploy Step'

- task: AzureWebApp@1

inputs:

azureSubscription: '<YourAzureSubscription>'

appType: 'webApp'

WebAppName: '<YourWebAppName>'

packageForLinux: '$(System.DefaultWorkingDirectory)/\*\*/\*.zip'

* Example

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

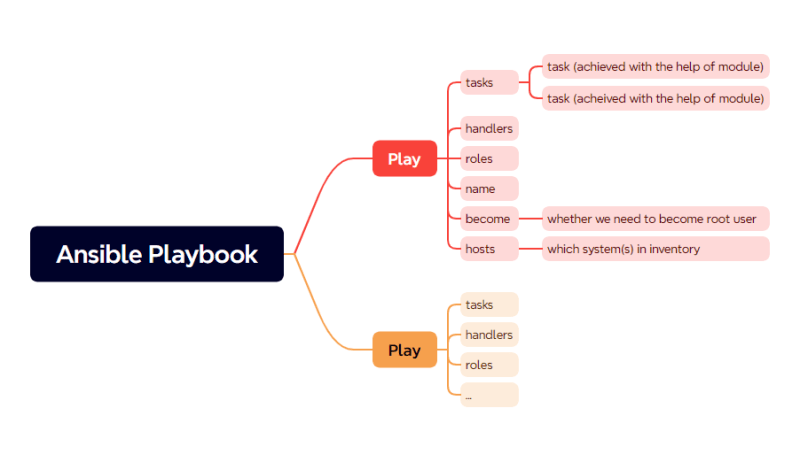
image: nginx:1.14.2

ports:

- containerPort: 80

* [Refer Here](https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html) for yaml syntax
* In devops we use yaml in many tools, each tool has a schema or structure defined and we need to follow.

# Ansible Playbook

* Ansible Playbook is written in YAML format
* Structure  
  
* In each Playbook task is the atomic activity (smallest unit of work) & is achevied with the help of module
* A module is where we can express what we want
* Way of Working
  + List down manual steps and ensure they are working
  + try to find a module for each manual step

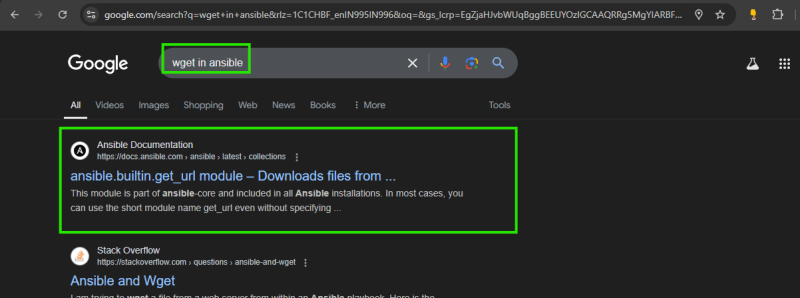
### Trying the above way of working

#### Sample 1

* manual steps

wget https://www.free-css.com/assets/files/free-css-templates/download/page296/oxer.zip

unzip oxer.zip

* Finding [ansible](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/) modules
  + approach 1: [Refer Here](https://docs.ansible.com/ansible/2.8/modules/list_of_all_modules.html) for module list
  + approach 2: google <command> in ansible  
    
* Module expressed in playbook as a task

---

- name: this is sample playbook

hosts: all

become: no

tasks:

- name: download the zip file

<module-name>:

<parameter-1>:

#..

<paramter-n>:

#[state]:

* Lets fill the download file

---

- name: this is sample playbook

hosts: all

become: no

tasks:

- name: download the zip file

ansible.builtin.get\_url:

url: https://www.free-css.com/assets/files/free-css-templates/download/page296/oxer.zip

dest: /tmp/oxer.zip

- name: unzip the file

ansible.builtin.unarchive:

dest: /tmp/oxer

src: /tmp/oxer.zip

#### Sample 2

* Manual steps

touch /tmp/1.txt

mkdir /tmp/test

* Playbook structure

---

- name: this is sample playbook

hosts: all

become: no

tasks:

- name: create a file

ansible.builtin.file:

path: /tmp/1.txt

state: touch

- name: create a directory

ansible.builtin.file:

path: /tmp/test

state: directory

#### Sample 3

* I want to install apache, php in the ubuntu 22.04 [Refer Here](https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu)

sudo apt update

sudo apt install apache2 -y

sudo apt install php libapache2-mod-php php-mysql

* Playbook

- name: install apache and php

become: yes

hosts: all

tasks:

- name: install apache

ansible.builtin.apt:

name: apache2

update\_cache: yes

state: present

- name: install php packages

ansible.builtin.apt:

name:

- php

- libapache2-mod-php

- php-mysql

state: present

### Share this:

* [Click to share on Twitter (Opens in new window)](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/?share=twitter&nb=1)
* [Click to share on Facebook (Opens in new window)](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/?share=facebook&nb=1)
* [Click to share on WhatsApp (Opens in new window)](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/?share=jetpack-whatsapp&nb=1)

### Like this:

November 2, 2024

# DevOps Classroom notes 02/Nov/2024

# YAML

* YAML is a file format which is designed over name value pairs
* YAML is used to represent data
* YAML files have extension of .yml or .yaml
* Any thing in yaml

<name>: <value>

* Data is categorized into following types
  + number:  
    yaml  
    experience: 4
  + text  
    yaml  
    designation: DevOps Engineer
  + boolean  
    yaml  
    certifications: yes
  + list/array  
    “`yaml  
    tools:</li>  
    <li>ansible</li>  
    <li>docker</li>  
    <li>k8s  
    “`
  + map/object/dictionary  
    yaml  
    address:  
    flatno: 601  
    building: nilgiri  
    landmark: ameerpet metro  
    city: hyderabad
* Scalar or simple types
  + number
  + text
  + boolean
* Complex
  + list
  + map
* Example 1 : [Ansible](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/) Playbook

---

- name: Install and configure Nginx web server

hosts: webservers

become: true

gather\_facts: yes

tasks:

- name: Display OS information

debug:

var: ansible\_distribution

- name: Update apt cache

apt:

update\_cache: yes

- name: Install Nginx web server

apt:

name: nginx

state: latest

- name: Start Nginx service

service:

name: nginx

state: started

* Structure:
  + list(map/object)
    - name: string
    - become: boolean
    - gather\_facts: boolean
    - tasks: list(map/object)
      * name: string
      * debug:
* Example

trigger: #list(text)

- main

pool: #map/object

vmImage: 'ubuntu-latest' #text

stages: # list(map/object)

- stage: Build

displayName: 'Build Stage'

jobs:

- job: BuildJob

displayName: 'Build Job'

steps:

- script: echo "Building the application..."

displayName: 'Build Step'

- task: DotNetCoreCLI@2

inputs:

command: 'restore'

projects: '\*\*/\*.csproj'

- task: DotNetCoreCLI@2

inputs:

command: 'build'

projects: '\*\*/\*.csproj'

arguments: '--configuration Release'

- stage: Test

displayName: 'Test Stage'

jobs:

- job: TestJob

displayName: 'Test Job'

steps:

- script: echo "Running tests..."

displayName: 'Test Step'

- task: DotNetCoreCLI@2

inputs:

command: 'test'

projects: '\*\*/\*.csproj'

arguments: '--configuration Release --no-build'

- stage: Deploy

displayName: 'Deploy Stage'

jobs:

- job: DeployJob

displayName: 'Deploy Job'

steps:

- script: echo "Deploying the application..."

displayName: 'Deploy Step'

- task: AzureWebApp@1

inputs:

azureSubscription: '<YourAzureSubscription>'

appType: 'webApp'

WebAppName: '<YourWebAppName>'

packageForLinux: '$(System.DefaultWorkingDirectory)/\*\*/\*.zip'

* Example

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

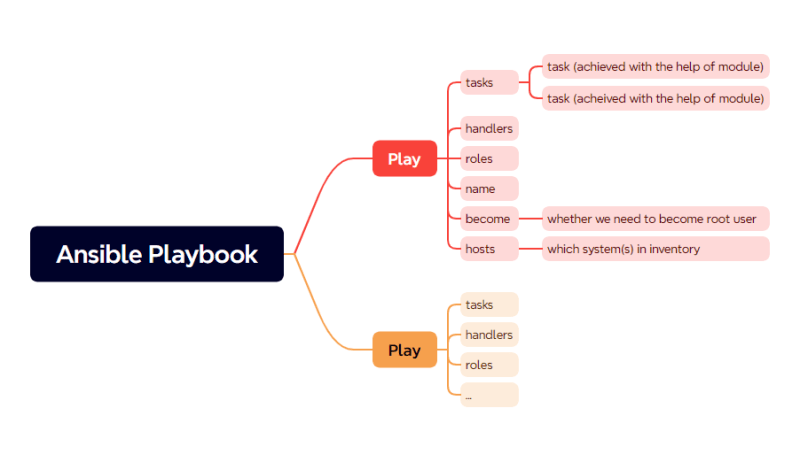
image: nginx:1.14.2

ports:

- containerPort: 80

* [Refer Here](https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html) for yaml syntax
* In devops we use yaml in many tools, each tool has a schema or structure defined and we need to follow.

# Ansible Playbook

* Ansible Playbook is written in YAML format
* Structure  
  
* In each Playbook task is the atomic activity (smallest unit of work) & is achevied with the help of module
* A module is where we can express what we want
* Way of Working
  + List down manual steps and ensure they are working
  + try to find a module for each manual step

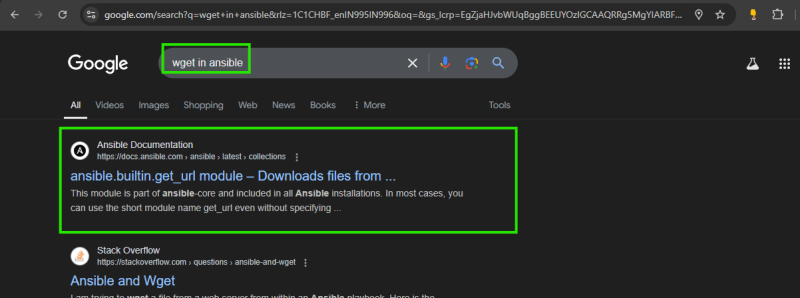
### Trying the above way of working

#### Sample 1

* manual steps

wget https://www.free-css.com/assets/files/free-css-templates/download/page296/oxer.zip

unzip oxer.zip

* Finding [ansible](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/) modules
  + approach 1: [Refer Here](https://docs.ansible.com/ansible/2.8/modules/list_of_all_modules.html) for module list
  + approach 2: google <command> in ansible  
    
* Module expressed in playbook as a task

---

- name: this is sample playbook

hosts: all

become: no

tasks:

- name: download the zip file

<module-name>:

<parameter-1>:

#..

<paramter-n>:

#[state]:

* Lets fill the download file

---

- name: this is sample playbook

hosts: all

become: no

tasks:

- name: download the zip file

ansible.builtin.get\_url:

url: https://www.free-css.com/assets/files/free-css-templates/download/page296/oxer.zip

dest: /tmp/oxer.zip

- name: unzip the file

ansible.builtin.unarchive:

dest: /tmp/oxer

src: /tmp/oxer.zip

#### Sample 2

* Manual steps

touch /tmp/1.txt

mkdir /tmp/test

* Playbook structure

---

- name: this is sample playbook

hosts: all

become: no

tasks:

- name: create a file

ansible.builtin.file:

path: /tmp/1.txt

state: touch

- name: create a directory

ansible.builtin.file:

path: /tmp/test

state: directory

#### Sample 3

* I want to install apache, php in the ubuntu 22.04 [Refer Here](https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu)

sudo apt update

sudo apt install apache2 -y

sudo apt install php libapache2-mod-php php-mysql

* Playbook

- name: install apache and php

become: yes

hosts: all

tasks:

- name: install apache

ansible.builtin.apt:

name: apache2

update\_cache: yes

state: present

- name: install php packages

ansible.builtin.apt:

name:

- php

- libapache2-mod-php

- php-mysql

state: present

### Share this:

* [Click to share on Twitter (Opens in new window)](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/?share=twitter&nb=1)
* [Click to share on Facebook (Opens in new window)](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/?share=facebook&nb=1)
* [Click to share on WhatsApp (Opens in new window)](https://directdevops.blog/2024/11/02/devops-classroom-notes-02-nov-2024/?share=jetpack-whatsapp&nb=1)

### Like this:

November 3, 2024

# DevOps Classroom notes 03/Nov/2024

## Ansible contd

### Activity 1

* Write an [ansible](https://directdevops.blog/2024/11/03/devops-classroom-notes-03-nov-2024/) playbook to deploy a website
* manual steps for ubuntu

sudo apt update

sudo apt install nginx -y

cd /tmp

wget https://www.free-css.com/assets/files/free-css-templates/download/page295/antique-cafe.zip

sudo apt install unzip -y

unzip antique-cafe.zip

sudo mv 2126\_antique\_cafe/ /var/www/html/cafe

* [Ansible](https://directdevops.blog/2024/11/03/devops-classroom-notes-03-nov-2024/) Playbook
* To execute Ansible playbook the command will be ansible-playbook -i hosts <playbook.yaml>
* Ansible output will contain
  + play info
  + an additional task called as gather\_facts
  + task info
* The info will contain
  + ok
  + changed
  + unreachable
  + failed
  + skipped
  + rescued
  + ignored
* We have written the playbook where we got the website deployed at url http://<publicip>/cafe/2126\_antique\_cafe
  + sequence changed [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/0325eeae2b188d3bb42056a9dd03b6b55c50d6b0)
  + sequence followed [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/73cd84b44230694b39fbb71c2afdc3fdd1e2e4e6)
* Ansible verbosity

### Exercise:

* Try installing apache and php

sudo apt update

sudo apt install apache2 -y

sudo apt install php libapache2-mod-php php-mysql

# WOW (Ways of Working) – Playbook writing

* Ensure you have working manual steps
* Find a module for each step an represent as task

November 5, 2024

# DevOps Classroom notes 05/Nov/2024

## Ansible

## inventories in ansible

* Ansible inventory is collection of nodes and is of two types
  + static inventory: we specify ip’s or names
  + dynamic inventory: we execute the script to get ips or names dynamically
* Static inventory can be specified in two formats [Refer Here](https://docs.ansible.com/ansible/latest/getting_started/get_started_inventory.html)
  + ini
  + yaml
* ini format
* yaml format

## Ansible facts

* Ansible collects the information about the node by executing a module setup [Refer Here](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_vars_facts.html)
* setup module [Refer Here](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/setup_module.html)

## Ansible conditionals

* [Refer Here](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_conditionals.html) for ansible conditions

## Ansible adhoc commands

* We can execute adhoc commands [Refer Here](https://docs.ansible.com/ansible/latest/command_guide/intro_adhoc.html)

ansible -i hosts -m "<module-name>" -a "<argumets/parameters>" all

* Lets install unzip

ansible.builts.file:

path: /tmp/1.txt

state: touch

* Adhoc command

ansible -i hosts -m "file" -a "path=/tmp/1.txt state=touch"

## Activity 2: Work with different distributions

* Install apache server on redhat and ubuntu
* manual steps for ubuntu

sudo apt update

sudo apt install apache2 -y

* manual steps for redhat

sudo dnf install httpd -y

sudo systemctl enable httpd

sudo systemctl start httpd

* To install apache on both redhat and ubuntu
  + write two different playbooks [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/a1673f52b988549a99ee6271591e6eabd19c0441)
* Note:
  + on redhat ssh conf /etc/ssh/sshd\_config.d/50-cloud-init.conf
  + service name on redhat sshd
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/2da6f6c545ea13e8945a7113eff9aa189a85f461) for the changes done.

### Share this:

November 6, 2024

# DevOps Classroom notes 06/Nov/2024

# Deploy a Spring boot application

* Spring boot [application](https://directdevops.blog/2024/11/06/devops-classroom-notes-06-nov-2024/) can be deployed in two ways
  + as an application
  + as a service
* Download Jar file [Refer Here](https://referenceappslt.s3.ap-south-1.amazonaws.com/spring-petclinic-3.3.0-SNAPSHOT.jar)
* As an applicaton
  + Install openjdk 17 sudo apt install openjdk-17-jdk
  + run the application java -jar spring-petclinic-3.3.0-SNAPSHOT.jar
  + This application will be hosted on port 8080
* As a service:
  + [Refer Here](https://www.baeldung.com/spring-boot-app-as-a-service) for an article to run spring boot as a service
  + Lets create a user spc with homedirectory as /var/lib/spc => sudo useradd -d '/var/lib/spc' -m -s '/bin/sh' spc
  + Download the jar file into /var/lib/spc => sudo wget -P /var/lib/spc/ https://referenceappslt.s3.ap-south-1.amazonaws.com/spring-petclinic-3.3.0-SNAPSHOT.jar
  + change ownership sudo chown spc:spc spring-petclinic-3.3.0-SNAPSHOT.jar
  + Now create a service file in the following location /usr/lib/systemd/system/spc.service  
    “`  
    [Unit]  
    Description=A Spring Boot application  
    After=syslog.target

[Service]  
User=spc  
ExecStart=/usr/bin/java -jar /var/lib/spc/spring-petclinic-3.3.0-SNAPSHOT.jar SuccessExitStatus=143

[Install]  
WantedBy=multi-user.target  
``  
\* Relaod the daemonsudo systemctl daemon-reload`  
\* Enable and start the spc service

### Ansible Playbook for deploying spring petclinic

* Ubuntu: Initially lets focus on running this playbook on ubuntu
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/dd6788f87ce33b8a4c1ec72a6ef8f1c830e72e06) for the playbook written in the class to automate the deployment of spring petclinic

### Share this:

November 7, 2024

# DevOps Classroom notes 07/Nov/2024

# Ansible contd

## Installing spc on redhat 9

* Manual steps

sudo dnf install java-17-openjdk -y

# for rest of steps refer previous classroom notes

* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/225db453400963a66e95cfe52ef9fa632c080172) for the changes done to accomodate the redhat support

### Accomodating changes in values

* For instance, few months down the line i might need to install java 21 or the username should be something else or url etc
* In [ansible](https://directdevops.blog/2024/11/07/devops-classroom-notes-07-nov-2024/) we can create variables.
* Variables can be passed at runtime and also can have default values if not passed
* [Ansible](https://directdevops.blog/2024/11/07/devops-classroom-notes-07-nov-2024/) allows to create variables at multiple locations (TBD)
* [Refer Here](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_variables.html) for offical docs
* debug module [Refer Here](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/debug_module.html)

#### Variables in inventory

* Generally when we define variables at inventory we have two types of variables
  + host variables:
  + group variables:
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/fdb69be77edf1186e69b4f23f0ec017b954fbd3d) for a host level varaible defined in inventory
* Updated the playbook to use generic package manager [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/338802df0760dfc5d4d38ccb5636bd4bdba17785)
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/27b3a226c89ce574f3584c1038f91ca897c8db38) for the changes to use group variables
* Ansible provides a better way for creating host and group variables by using host\_vars directory and group\_vars directory,
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/df879d1d1089eb8fe0dd97ea7d24ea3e8790aefc) for the changes

### Good Practice

* Ensure you add necessary debug statements to inform users about activities

### Templating

* Ansible uses jinja templates to handle dynamic content in files [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/31f0c888de2ac70468f7b49823f0e15ad630b149)

November 9, 2024

# DevOps Classroom notes 09/Nov/2024

## Ansible Roles

* [Ansible](https://directdevops.blog/2024/11/09/devops-classroom-notes-09-nov-2024/) Roles can make playbooks reusable
* From a playbook we call role
* Roles can be
  + from [ansible](https://directdevops.blog/2024/11/09/devops-classroom-notes-09-nov-2024/) galaxy
  + from your local development
* Lets use a role to install mysql. Navigate to [roles](https://galaxy.ansible.com/ui/standalone/roles/) in Ansible Galaxy
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/26b1538c883938b8cc5e3a4a29cf97a10d46ca73) for the changes done to execute the database installation
* [Refer Here](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html) for Role docs
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/9706af82784e0ee272510bb82baf0cf44eb37d2a) for the changes done to create a nop role

### Ansible Collections

* Ansbile custom modules can be developed in python
* Many communities develop custom modules and have to wait till next release to get the modules
* Ansible collections make it easier to distribute custom modules independent of ansible releases
* Collections can be used to distribute
  + roles
  + custom modules

### Share this:

November 8, 2024

# DevOps Classroom notes 08/Nov/2024

# Ansible

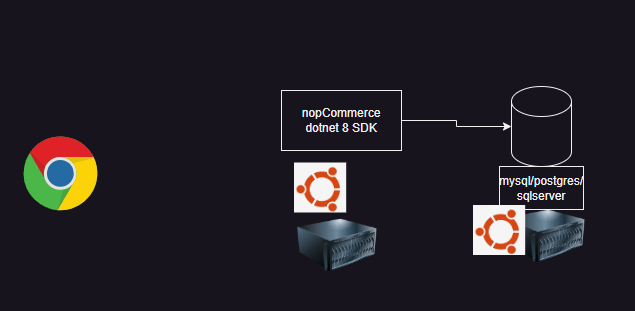
# loops in ansible

* [Loops](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_loops.html) can be used to iterate a task with multiple values.

# handlers

* [Handlers](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_handlers.html) are like tasks but they react to events

# Deploying a .net core applicaton

* Lets try deploying a .net core application nopCommerce
* [Refer Here](https://docs.nopcommerce.com/en/installation-and-upgrading/installing-nopcommerce/installing-on-linux.html) for the steps
* Overview  
  
* Note: we will be skipping the db installation as of now
* Manual steps
  + install dotnet 8 SDK/Runtime
  + install unzip
  + Create a user to run nopCommerce service
  + Download nopCommerce
  + Create a servicefile
  + enable and start the service

sudo apt-get update && \

sudo apt-get install -y dotnet-sdk-8.0

sudo useradd -m -d /var/lib/nopCommerce nop

mkdir /tmp/nopCommerce

cd /tmp/nopCommerce

wget https://github.com/nopSolutions/nopCommerce/releases/download/release-4.70.5/nopCommerce\_4.70.5\_NoSource\_linux\_x64.zip

unzip nopCommerce\_4.70.5\_NoSource\_linux\_x64.zip

mkdir bin logs

sudo mv /tmp/nopCommerce /var/lib/nopCommerce

sudo chown -R nop:nop /var/lib/nopCommerce

# create a service file

sudo systemctl daemon-reload

sudo systemctl enable nopCommerce.service

sudo systemctl start nopCommerce.service

* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/5a992b53a788e67290c9ece2080b360dcb463dac) for the changes done to
  + install dotnet and unzip
  + create a nop User
  + download and extract zip
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/b7f9bea0b570dca04971b85ea62163dd35da9959) for the creating directories
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/fa2fa10dece5e3bae31b44b5a72f2db4b5d01f95) for changes done to copy the service file and enable, start service
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/78c72288e8dfd7ceadfed8dba40d1ff63febe1b5) for the addition of hanlder to reload daemon, enable and restart the service on change, but normally ensure the service is running

# Reusability in Ansible

* The following options provide reusability in ansible
  + include or import playbooks
  + roles:
    - This is a way to make a playbook reusable
  + collections
    - This is a way to make a playbook as well as custom module reusable
* [Ansible Galaxy](https://galaxy.ansible.com/ui/): This is where we find community roles and modules

### Share this:

* [Click to share on Twitter (Opens in new window)](https://directdevops.blog/2024/11/08/devops-classroom-notes-08-nov-2024/?share=twitter&nb=1)
* [Click to share on Facebook (Opens in new window)](https://directdevops.blog/2024/11/08/devops-classroom-notes-08-nov-2024/?share=facebook&nb=1)
* [Click to share on WhatsApp (Opens in new window)](https://directdevops.blog/2024/11/08/devops-classroom-notes-08-nov-2024/?share=jetpack-whatsapp&nb=1)

### Like this:

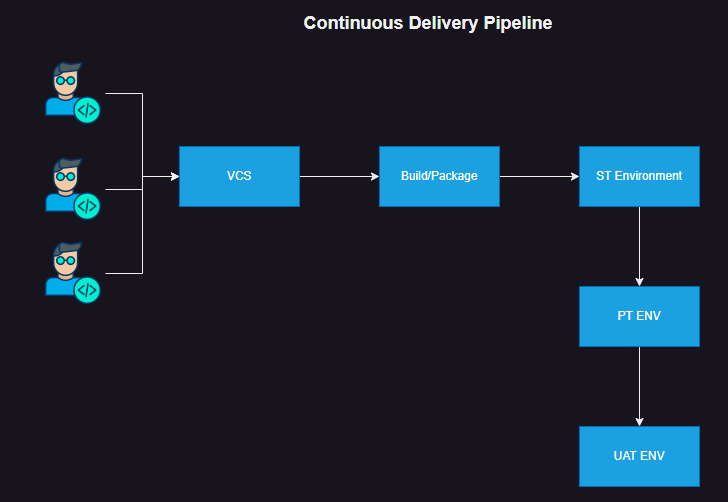
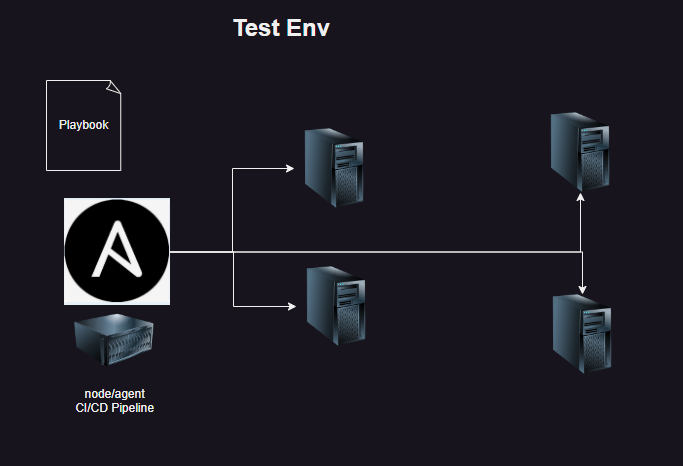
November 10, 2024

# DevOps Classroom notes 10/Nov/2024

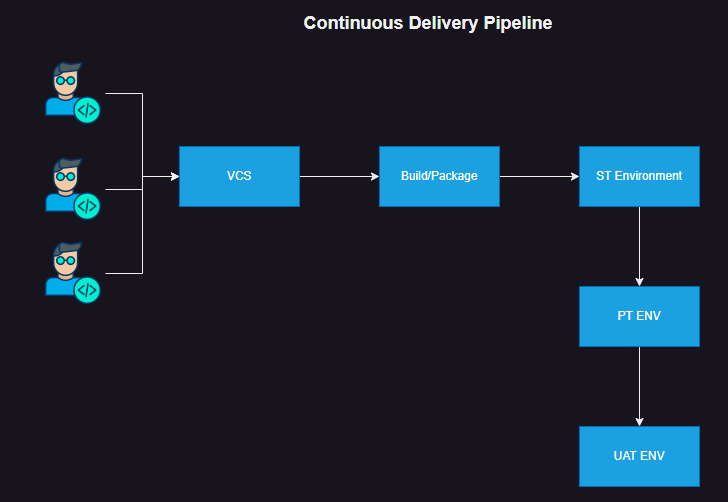
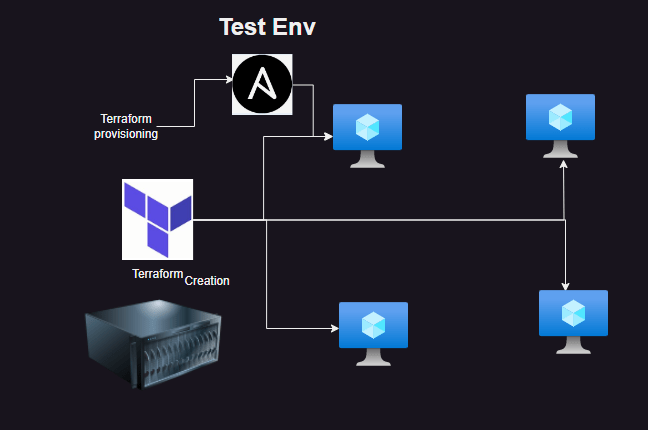
# Usage of Ansible in CI/CD pipelines

CI/CD services

## Earlier

* [Continuous Delivery](https://directdevops.blog/2024/11/10/devops-classroom-notes-10-nov-2024/) Pipeline  
  
* An individual environment was handled as shown below  
  

## Latest Usage

* Continuous Delivery Pipeline  
  
* Infra provisioning changed the [Configuration Managemnt](https://directdevops.blog/2024/11/10/devops-classroom-notes-10-nov-2024/) Landscape  
  
* Watch classroom recording to view the latest usage practically

### Integrating Terraform with ansible

* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/6a0d5ba02f103981b7142519facf4fe7792623ff) for the sample terraform template invoking ansible
* [complete solution](https://github.com/asquarezone/AnsibleZone/tree/master/Nov24/Terraform)

# Ansible Magic Variables

* [Refer Here](https://docs.ansible.com/ansible/latest/reference_appendices/special_variables.html)

### Share this:

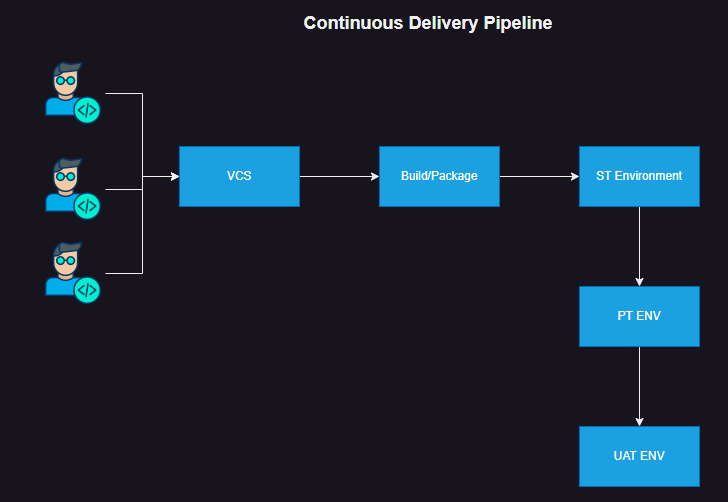
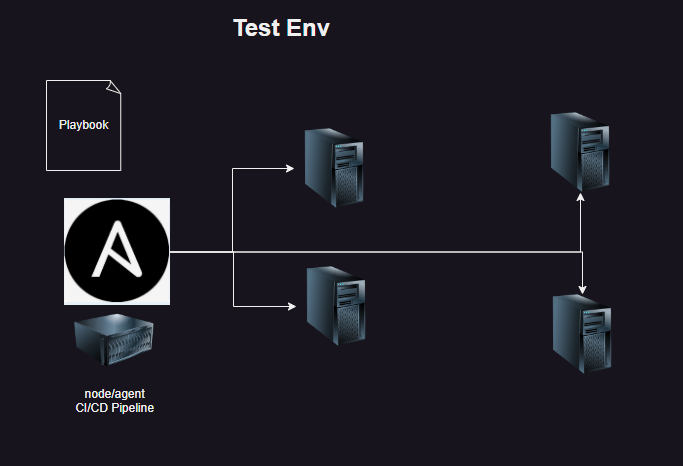
* [Click to share on Twitter (Opens in new window)](https://directdevops.blog/2024/11/10/devops-classroom-notes-10-nov-2024/?share=twitter&nb=1)
* [Click to share on Facebook (Opens in new window)](https://directdevops.blog/2024/11/10/devops-classroom-notes-10-nov-2024/?share=facebook&nb=1)
* [Click to share on WhatsApp (Opens in](https://directdevops.blog/2024/11/10/devops-classroom-notes-10-nov-2024/?share=jetpack-whatsapp&nb=1)

November 10, 2024

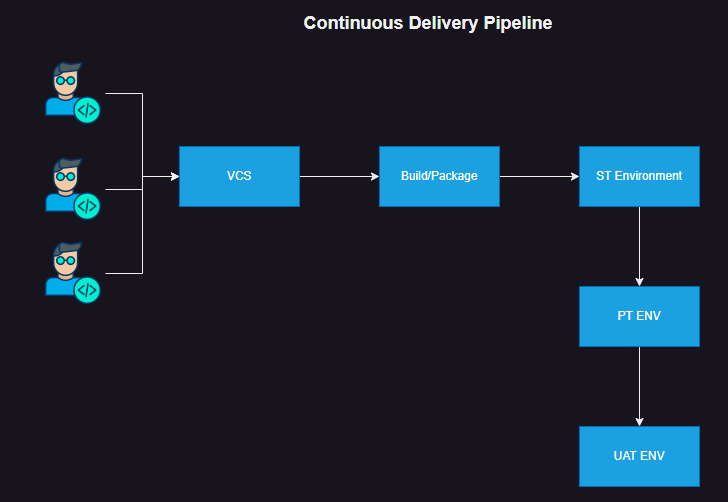
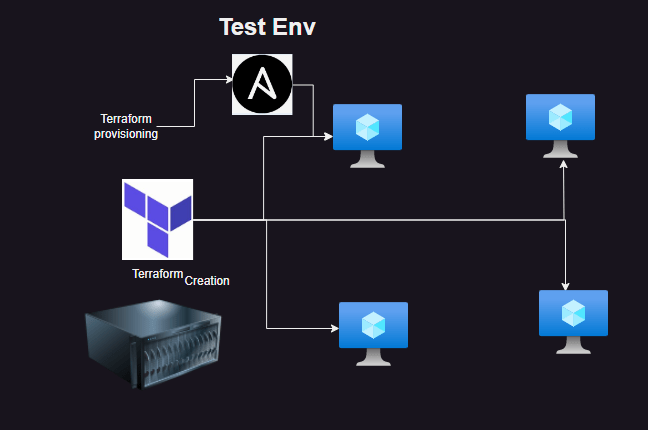
# DevOps Classroom notes 10/Nov/2024

# Usage of Ansible in CI/CD pipelines

## Earlier

* Continuous Delivery Pipeline  
  
* An individual environment was handled as shown below  
  

## Latest Usage

* Continuous Delivery Pipeline  
  
* Infra provisioning changed the [Configuration](https://directdevops.blog/2024/11/10/devops-classroom-notes-10-nov-2024/) Managemnt Landscape  
  
* Watch classroom recording to view the latest usage practically

### Integrating Terraform with ansible

* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/6a0d5ba02f103981b7142519facf4fe7792623ff) for the sample terraform template invoking ansible
* [complete solution](https://github.com/asquarezone/AnsibleZone/tree/master/Nov24/Terraform)

# Ansible Magic Variables

* [Refer Here](https://docs.ansible.com/ansible/latest/reference_appendices/special_variables.html)

### Share this: