DevOps

2020-2021 - B3 Ingésup - Campus Ynov Aix

Me, My Self and I

- Jérome Masson - Architecte IT - DevOps Evangelist
  - M2 Ingésup
  - 10 years of Consulting, Dev & Hosting and Infrastructure
- Consultant DevOps
- Containerization Expert
- Docker official trainer
- Kubernetes, Gitlab & Ansible Trainer

Game Rules - Beginning

- Start at (see your hyperplanning)
- No grace period, be ready before the beginning of the training
- when the door is closed, go to "administration to pick up a late ticket"

Game Rules - Absence

- Notify :
  - Administration
  - Me by mail
- Ask homeworks to the other students

Game Rules - Late or Missing the evaluation

- Notify :
  - Me by mail
- I notify the administration
- Administration decide the sanction to apply

Game Rules - Behavior

- No cell phone on the table :
  - only can decide an express derogation
- Laptop :
  - Update to date and ready
  - No personnal use during training
- Noise :
  - Be quiet during theorical part
  - Be aware your not alone during pratical part
- Outgoing :
  - only with my approval

> /!\ Be aware /!
> 1st : verbal advice 2nd : possible exclusion

Flipped Classroom

- Learn by doing, pratice to learn
- Pratice :
    - Projects :
        - Le premier dès aujourd'hui
    - Technical adds during training
        - I give you the key, you open the door !
    - Personal researchs

> You are the key

Training

Date	All
06/10	8:30 -> 12:30
13/10	8:30 -> 12:30
20/10	8:30 -> 12:30
27/10	8:30 -> 12:30
02/11	8:30 -> 12:30
03/11	8:30 -> 12:30
10/11	8:30 -> 12:30
17/11	8:30 -> 12:30
24/11	8:30 -> 12:30
01/12	8:30 -> 12:30

Participation

- Albert Einstein

> "There are no stupid questions, only stupid answers"

... well, unless you haven't read or listened

Training Objctives

Officials :

- Comprendre les enjeux de l'organisation DevOps
- Être capable d'identifier les blocages issus d'une mauvaise communication entre dévs et ops
- Savoir apporter des solutions à ces blocages, qui satisfassent les deux équipes
- Apprendre à connaître les besoins et les contraintes qui font le quotidien de « l'autre équipe »
- S'initier aux outils et aux technologies orientées DevOps

My Objectives :

- Technical and soft skills of DevOps pipeline

Evaluations

Date	All
06/10	x
13/10	x
20/10	x
26/10	x
27/10	x
10/11	Personnal Eval
17/11	x
24/11	x
01/12	Middle Project
08/12	x
15/12	Final Project
05/01	Final Project

> refer to Hyperplanning to keep in touch

Evaluations

- Group + Individual
  - each evaluation is divided into 2 parts (/20)
- each evaluation is detailed at the begging of training day
- Cross team "Final Project"

Evaluations

- Participation :
    - Do not disturb others
- Project :
    - Individual :
        - every commit and production
    - Group :
        - teamwork and result
- Final project :
    - tools & quality

Survey

- password: ynovdevops2020

DASA Scan

Additionnal videos

Linkedin Learning

DevOps :

- DevOps Introduction :
  - https://www.linkedin.com/learning/devops-foundations/development-and-operations?u=56745737
- DevSecOps :
  - https://www.linkedin.com/learning/devops-foundations-devsecops/welcome?u=56745737

IAC :

- Infrastructure as code :
  - https://www.linkedin.com/learning/paths/improve-your-infrastructure-automation-with-hashicorp-tools?u=56745737

Versionning :

- Git :
  - https://www.linkedin.com/learning/git-essential-training-the-basics/use-git-version-control-software-to-manage-project-code?u=56745737
- Git Flow :
  - https://www.grafikart.fr/tutoriels/git-flow-742
  - https://www.linkedin.com/learning/git-for-teams/using-git-for-team-collaboration?u=56745737
- GitOps :
  - https://www.weave.works/technologies/gitops/
  -

Language for Dev :

- Node.Js :
  - https://www.lynda.com/Node-js-tutorials/Learning-Node-js/612195-2.html?srchtrk=index%3a5%0alinktypeid%3a2%0aq%3anodejs%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2

Language for Ops :

- Ansible :
  - https://www.linkedin.com/learning/learning-ansible-2017/welcome?u=56745737

Ultimate DevOps

https://www.linkedin.com/learning/paths/become-a-devops-engineer?u=56745737

General notions

What is DevOps ?

DevOps is :

- Culture rather than a technique
- Team collaboration Dev & Ops & ...
- Based on shared & Respects
- No failure but learning
- Based on Agile & Lean

What is your own experience ?

Opposites

Dev	**Ops**
- Dev environment	- Owned infrastructure
- Tools	- Receive artifacts
- CI	- Put in production
`Deliver on time and respect spec`	`Production up and running`

Business
use IT as a Service

Dev & Ops against the world

- Loose time
- Bad quality

> Increase failure Increase time to cost Lost customer

Faith is the key

- Env + Time => Dev to Ops
- Quality + Test => Ops to Dev

> Increase SI quality Increase time to cost

DevOps

1. Méthodologie / Mouvance + qu'une pratique (== agilité)
2. Principes Généraux :
    1. Amener + rapidement le code en production : Livrer plus rapidement, plus souvent
    2. "Sécuriser" les livraisons : limiter bug, boucle feedback
    3. Automatisation
    4. Partage : Amélioration continue

Fundamentals

- CALMS
    - Culture
    - Automation

- Lean
- Mesure
- Share

How to

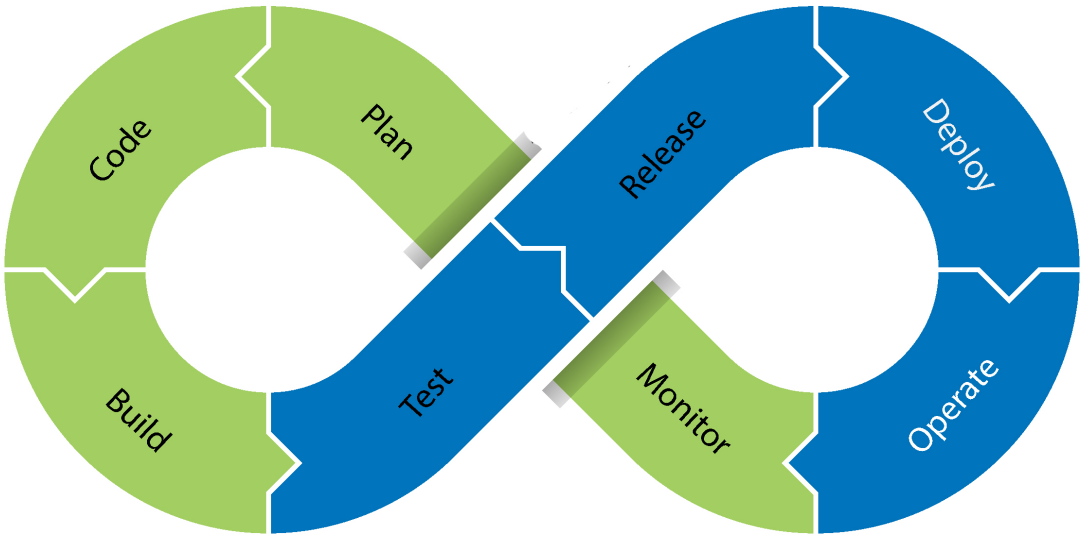
C	A	L	M	S
Collaboration	Industrialise the supply chain	Quality	Monitore	Share knowledge

- Reduce & Optimise all the supply chain
- Optimise the resources
- Increase quality & SLA
- Culture

Supply Chain

- standardize deliveries
- Contiunous Integration
- Conitnuous Delivery
- Continuous Deployment

Pipeline DevOps



Supply Chain

> Think the future

- Why ? Maintenability ?

> Keep it simple

- Do what is design for and keep it simple

> Microservices Architecture

- Decrease dependancies between parts of development increase maintenability

> Patterns

- Software design (singleton, abstract...)

> Versioning

- more easier to evolve
- try is the key

Continuous Supply Chain

Type	Code	Unit tests	Intégration tests	Preproduction tests	Production
CI	> Manual	> Auto	> Auto	Manual	x
CD	> Auto	> Auto	> Auto	> Manual	x
CD	> Auto	> Auto	> Auto	> Auto	x

Let's Pratice

Techno :

- git:
  - repo : gitlab
  - methodo :
    - dev : gitflow
    - prod : gitops
- docker:
  - repo : quay.io / dockerhub
  - docker-compose
- terraform:
  - nativement: go
  - conf: hcl (hashicorp core language)
- ansible :
  - nativement : python
  - conf : yaml
- dev:
  - lang:
    - node.js
    - python

Todo :

- Création d'un compte gitlab
- Création d'un compte quay.io / dockerhub
- Création d'un compte Terraform cloud
- Création d'un compte heroku

Todo Automation :

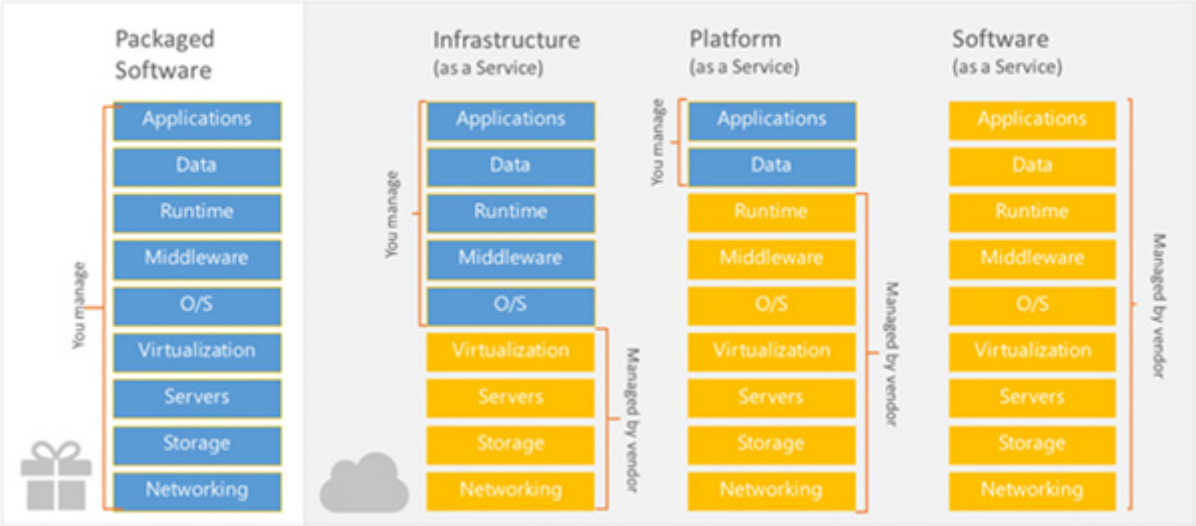
- Accéder : https://labs.play-with-docker.com/
  - web tty container
- Ajouter une instance
- Commands :

```
mkdir docker
cd docker
cat <<EOF > Dockefile
FROM centos
EOF

docker image build -t test .

# instanciation du conteneur et accés bash
docker container run -it test bash

cat /etc/os-release
```

Cloud : Public / Private

- Cloud
- Virtual
- Autoscaling
- Automation & Orchestration

Cloud Computing



author link

Onprem IAAS PAAS CAAS : Container as a service SAAS FAAS : Function as a Service

---

Cloud types

> Private

- Intern infrastructure

> Public

- Azure / AWS / GCP / ...

> Hybrid

- Intern & External

Why virtualize ?

Economies à l'achat et à la maintenance Un « gros » serveur coûte moins cher que de nombreux « petits » Maintenance matérielle simplifiée

Optimisation de l'utilisation des ressources Mieux vaut un « gros » serveur utilisé en moyenne à 90% que 5 ou 6 « petits » utilisés à 15%

Simplification des évolutions d'infrastructure Déploiement de nouvelles VM plutôt qu'achat de nouvelles machines (test, pré-prod ou prod)

Facilité de gestion des environnements de développement Mise à disposition de VM de dév iso-production qu'installation sur les postes des développeurs ou sur un serveur central

---

How to ?

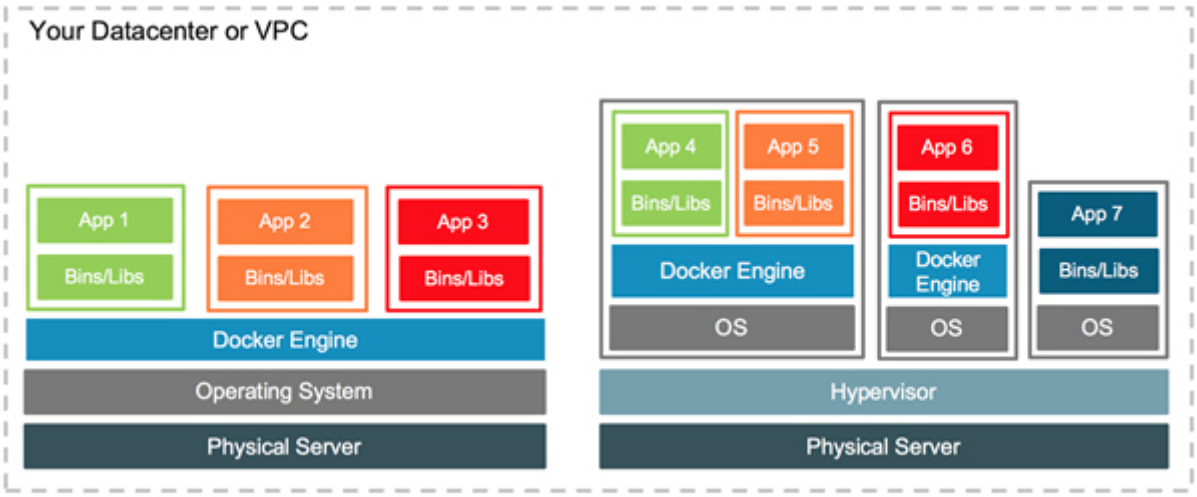
Virtualisation matérielle Support de la virtualisation intégrée ou assistée par le processeur ; ex : IBM Power, AIX, Intel VT…

Hyperviseur de type 1 (nati ou baremetal) Noyau système hyper-léger gérant les accès des OS « virtuels » au matériel ; meilleures performances avec des OS optimisés ; ex : Citrix Xen Server, VMWare ESXI, MS Hyper-V Server…

Hyperviseur de type 2 (hosted) Exécuté sur l'OS hôte et gérant le matériel pour les OS « virtuels » de façon transparente ; permet l'exécution de tous types d'OS ; ex : MS VirtualPC & Virtual Server, Oracle VirtualBox, VMWare Workstation…

Virtualisation par container Isolement des processus dans un espace de nom virtualisé, leur donnant l'illusion d'être seuls sur la machine (ex : Docker…)

How to ?



Le container docker n'est pas un environnement OS complet. Container :

- Isolation process : RAM / CPU / ...

Infrastructure as Code

Provisioning

- Physical allocation
- Ansible
- Chef
- Puppet
- Powershell ...

Bare metal

- Install

- Configure

> Provisionning VM

- Terraform
- Shell
- CLI

> Container

- Distribute and deploy
- Lighter

Autoscaling

> VM

- Dynamic attribution of RAM, Disk, ...

> Cloud

- More instance

> Container

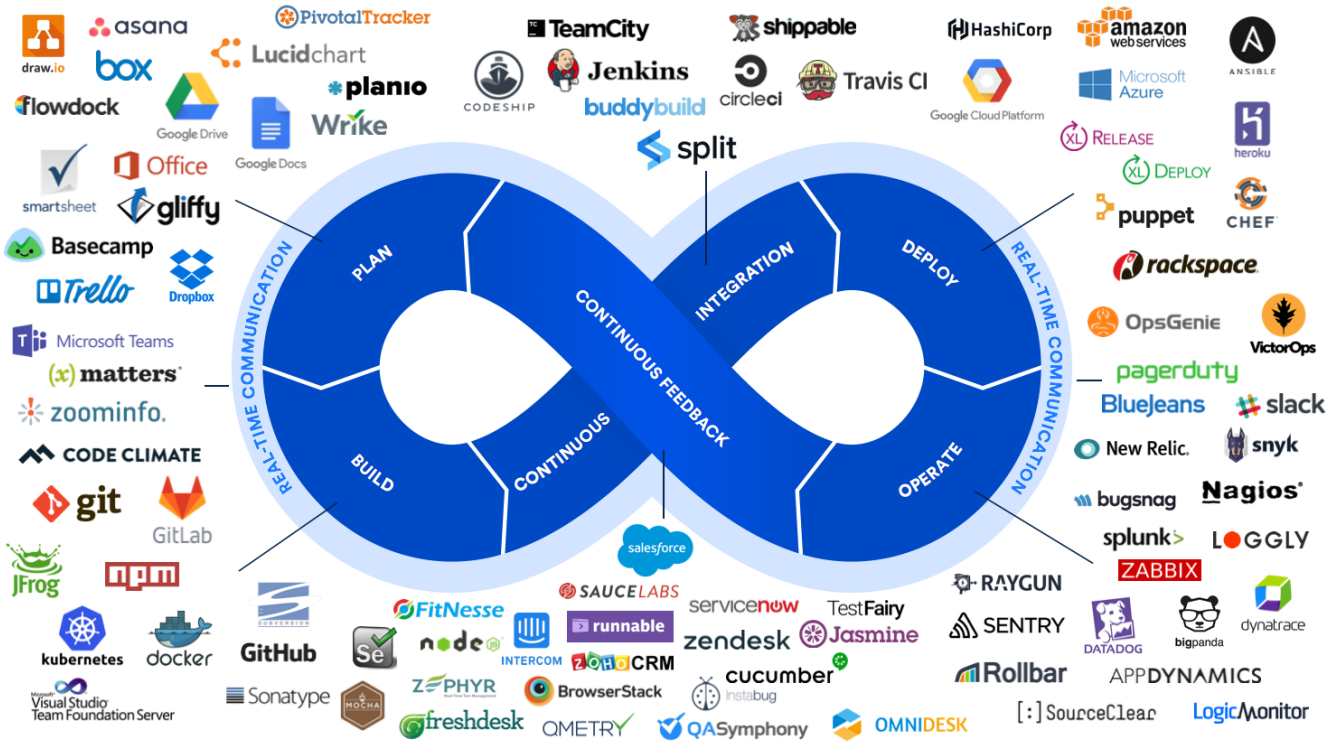
- Horizontal scalability

Let's Pratice

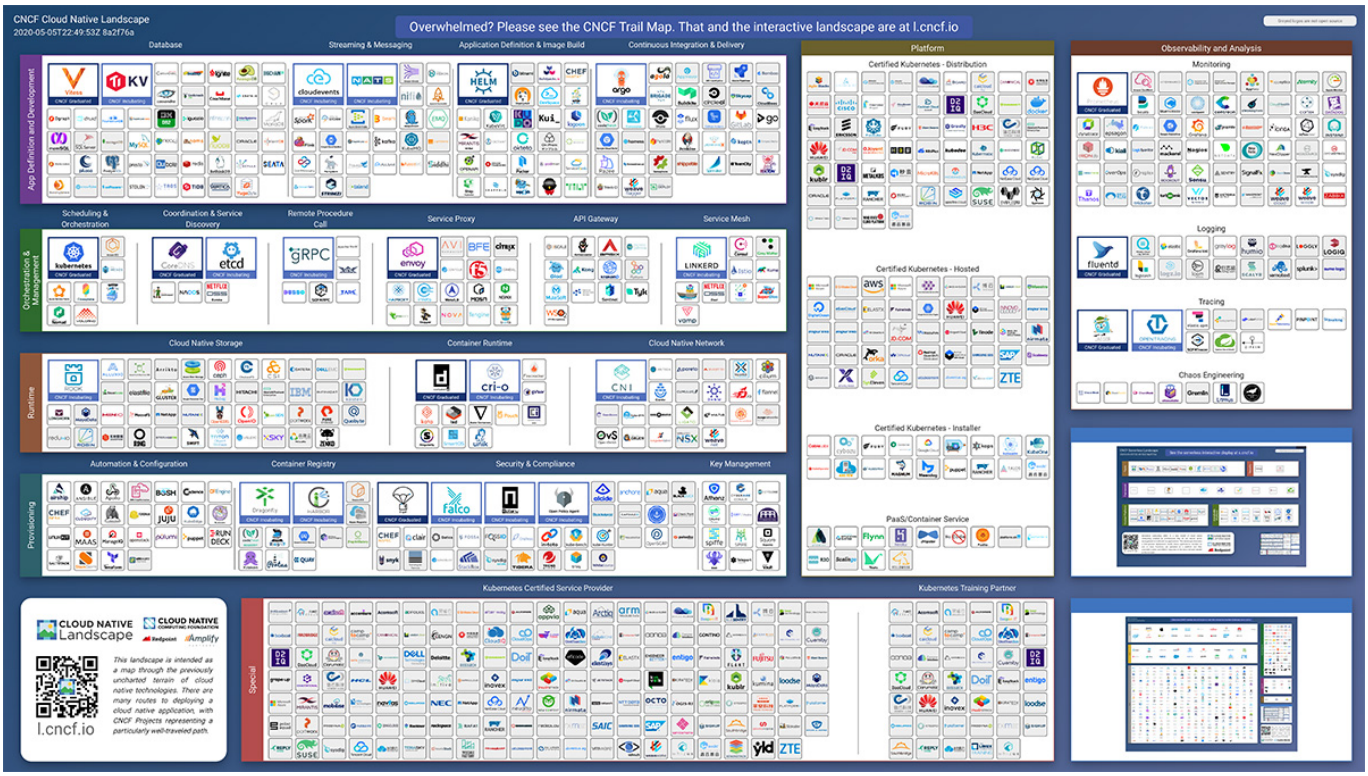
Todo Docker :

- go to play with docker
- Créer un cluster :
    - Master : Tête pensante
        - 3 Bonnes pratiques
    - Worker : Machine de travail
        - min 2

DevOps Tools



CNCF Tools



CNCF Landscape

End

> Next training :

- 20/10/2020

> Next time :

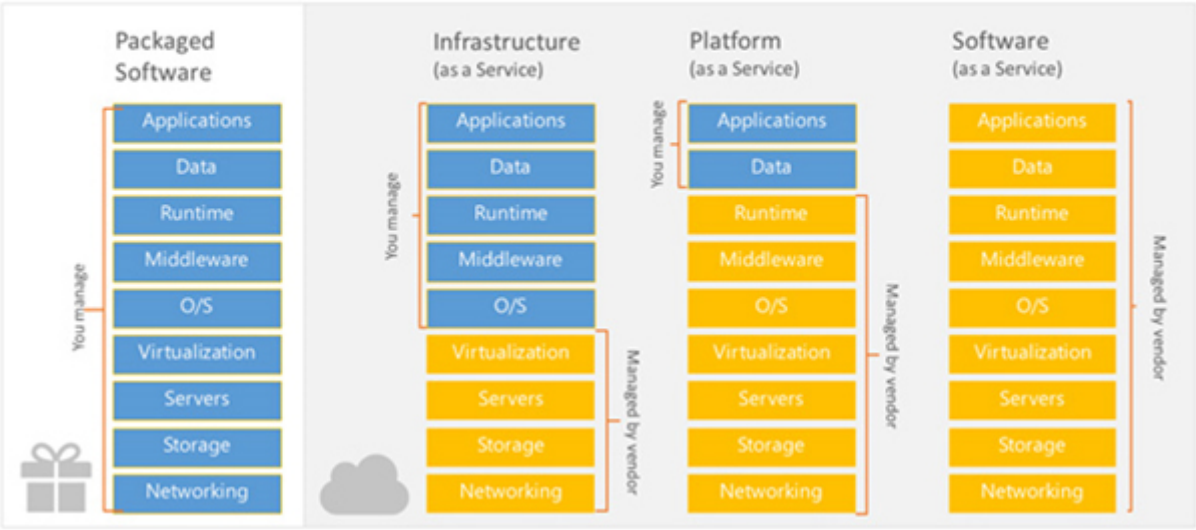
- 2 VM : 1 VM : - CPU : 1-2 - 2 Go 1 VM : - CPU : 2 - 4 Go
- OS libre
- Packages :
  - Docker CE
  - GIT
  - Ansible
  - Python 3.x

Training DevOps

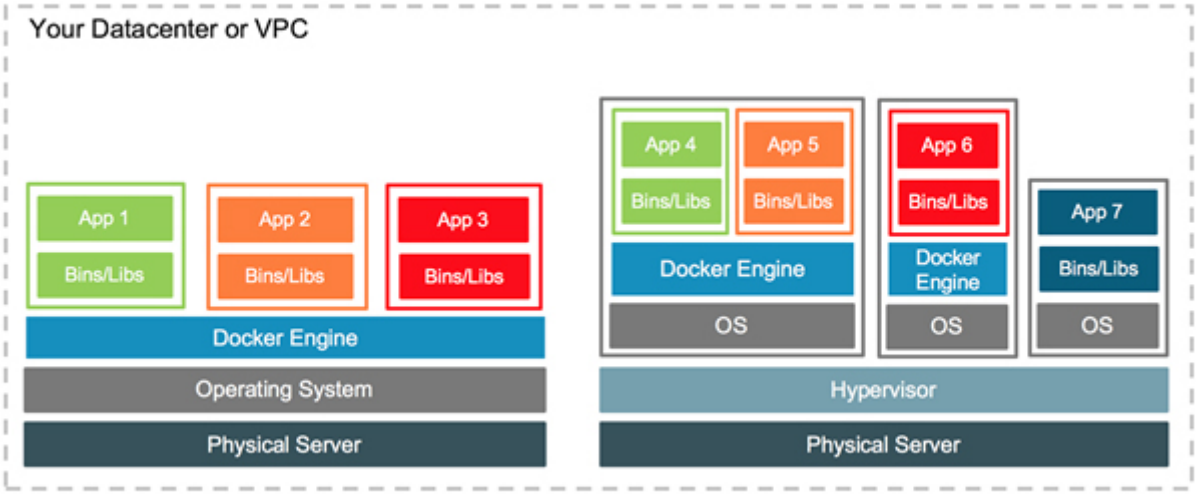
Summary

- Automation
- Virtualization
- Orchestration

Automation



author link



IAC : Infrastructure as Code

- Infrastructure defined as dev

- Files description
- Reproductible, Scalable, ...

Prérequis IAC

Techniques :

- Git
- SHELL - POWERSHELL
- Cours :
  - Ansible : Gestion de configuration
  - Terraform : Gestion infra (statique)
  - Docker : Pseudo Virtual - Dev (OPS)

Prérequis DEV

Techniques :

- Git
- Cours :
  - Docker : Pseudo Virtual - Dev (OPS)
  - API
  - Code cloud native :
    - tolère la panne
    - scalable
    - minimaliste

Ansible

Ansible

What is Ansible ?

> **Simple, agentless IT automation that anyone can use**
>
> > Ansible is a universal language, unraveling the mystery of how work gets done. Turn tough tasks into repeatable playbooks. Roll out enterprise-wide protocols with the push of a button. Give your team the tools to automate, solve, and share.

Ansible

What do Ansible ?

- Standardize command
- Agentless
- SSH connection

- Apply task to defined hosts

Ansible

Ansible command line

- Could be use in command line tool

PRO	CON
easy to use	not repoducible

Ansible

Ansible command line

```
ansible localhost -m ping
ansible all --key-file ~/.ssh/id_rsa -m ping
```

Ansible

Ansible playbook

- Define a reusable task
- Structured

Ansible

Ansible playbook (2/2)

- Mono playbook
- Roles
  - Task
  - Default
  - Handler
  - Vars
  - Template

Let's Pratice

TP :

Installer Ansible :

- CentOS :
  - yum install epel-release
  - yum install pip
  - pip install --user ansible

Créer un dossier `ansible` dans votre home directory

- /home/monuser
- si root : /root

Créer inventaire : `hosts.cfg`

- commander :
  - ip 127.0.0.1
  - ou ip VM

```
ansible -m ping -i hosts.cfg commander
```

Créer clé ssh RSA 2048 :

- privée et publique

```
ssh-keygen
```

Autoriser la clé ssh à se connecter à votre machine :

```
cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
```

```
cd
mkdir ansible
cd ansible
echo "commander ansible_host=127.0.0.1" > hosts.cfg
# aka : vi hosts.cfg
# i pour insert
# ESC
# :wq ou :x

ansible -m ping -i hosts.cfg commander
ssh-keygen
cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys

ansible -m ping -i hosts.cfg commander
```

Troubleshooting

ssh localhost -vv

Corriger les droits chmod 0600 ~/.ssh/id_rsa

Renommer : mv ~/.ssh/2048 ~/.ssh/id_rsa

Let's Pratice 2

TP 2 :

```
-  groupes de machine
    -  Front
    -  Backend
    -  Bdd
-  Spécifier
    -  2 machines
```

```
      -  1 machine : doit être dans 2 groupe
  - Groupe global frontend + backend
```

IAC : Infrastructure as Code

VM Provisioning

Vagrant

Objective

- Provision VM for DEV environment

vagrant tutorial - linkedin learning

Terraform

Terraform

Objective

Projet Intermédiaire

- Dev : Gestionnaire d'étudiants
    - env :
        - VM

    - Créer mini site :
        - front
            - Techno autorisées :
                - node.js
                - python
                - html
                - css
                - js
        - back :
            - BDD libre
            - Techno autorisées :
                - node.js
                - python
                - php -> simplyapi
    - API REST :
        - ETUDIANT :
            - NOM : string
            - PRENOM : string
            - USER : string
            - PASS : string
            - SPECIALITE : id
            - PROFIL : id
        - PROFIL :
            - TYPE : string (admin / non)
        - SPECIALITE :
            - TYPE : string (dev / infra / reseau / secu)
    - IHM :
        - LOGIN
        - AFFICHAGE
            - ETUDIANTS
            - ETUDIANT
        - GESTION
            - AJOUT
            - SUPPRESSION

- infra :

    - env :
        - via script
            - shell
            - ansible
        - system
            - centos
            - autant de ressources
                - spec :
                    - fichier normalisé
                - archi :

        - schéma
        - fichier normalisé
    - BDD dev :
        - Installer + hardening
    - Install toutes les briques applicatives
    - Install du code
    - nginx (min 1 - max 2)
        - ssl :
            - group01.ynovdevops.fr
            - group10.ynovdevops.fr
        - site + api en https

Doc :

- README.md