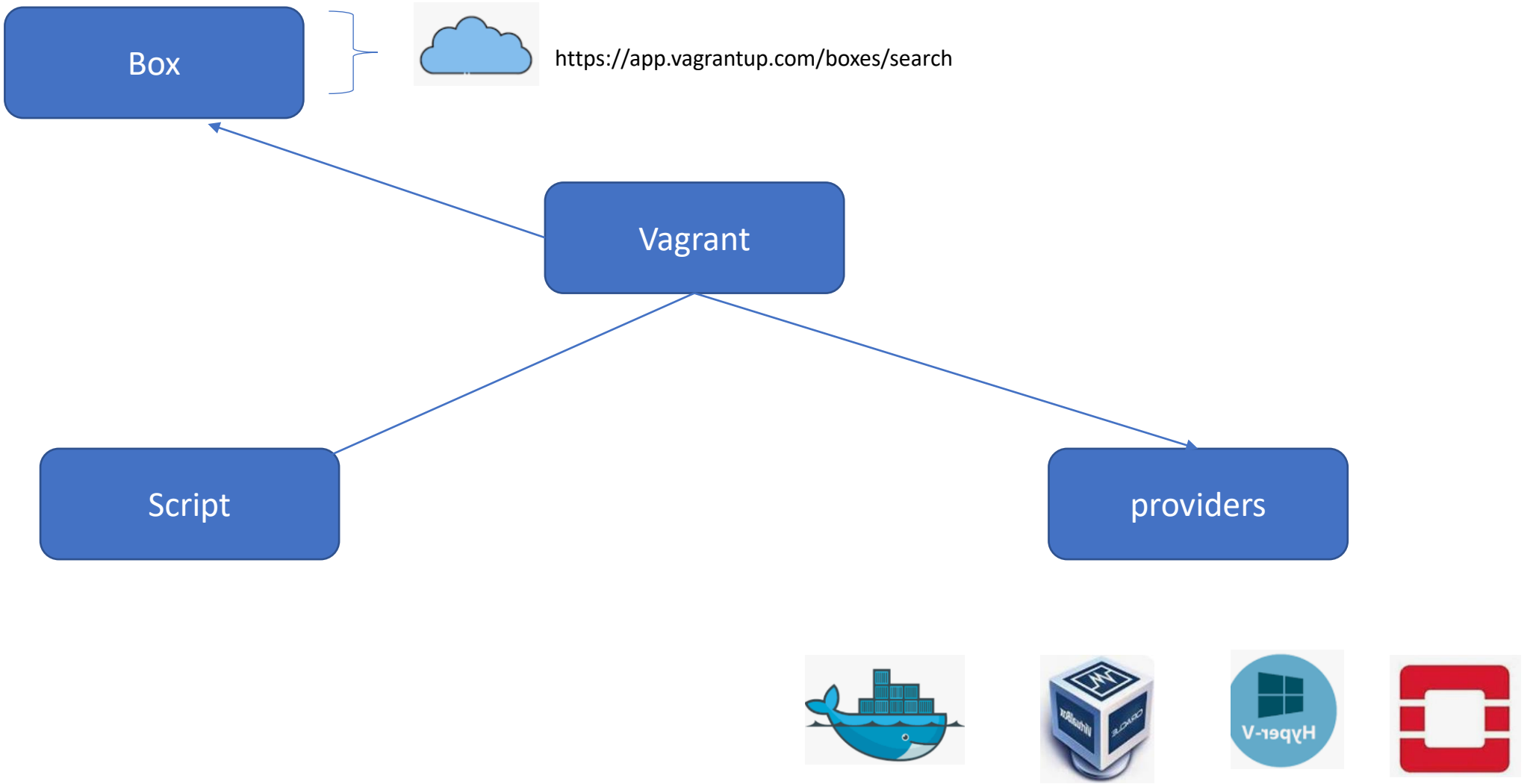
Objectifs

- Etre de capable d’automatiser une infrastructure
- Etre capable de distinguer les principales pratiques devops et sysops

Vagrant : comment ça marche ?



Vagrant Terminologie

- **Vagrant Box** : est considérer comme une image, un template depuis lequel nous allons déployer nos futures machines virtuelles. Les Boxes sont disponible sur le cloud de vagrant

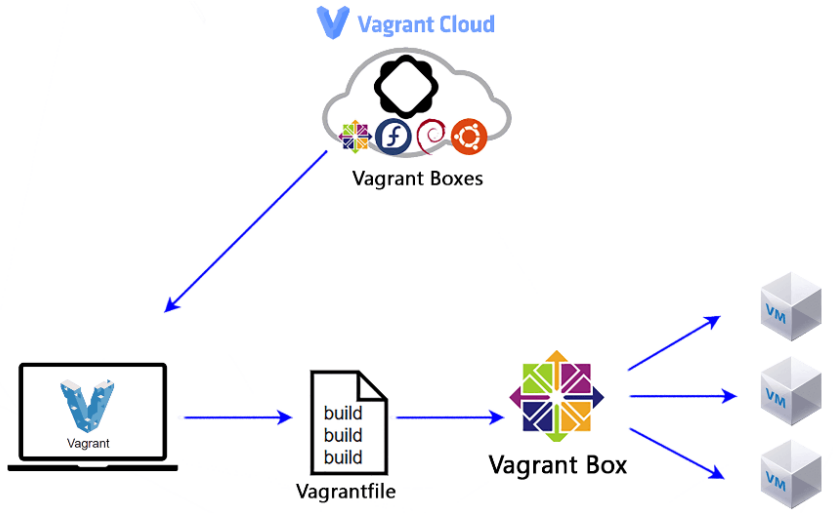
- **Packer** : est un outil qui permet la construction des images *from code* , nommé **Packer Build**

- **Vagrantfile** : fichier décrivant la configuration utilisée par **Vagrant** pour *déployer* et *configurer* un ensemble de machines virtuelles selon **les spécifications** déclarer dans ce fichier **Vagrantfile**

- **Provider** : est la plus basse couche permettant de déployer notre environement :
  - Virtualbox
  - Hyper-V
  - VMware
  - Libvirt
  - AWS

- **Provisioner** : Une fois la VM déployer, le **Provisionner** va déclencher la procédure d'installation et configuration *automatique* des VMs depuis :
  - Script Shell
  - Ansible
  - ..

Vagrant Génération des VMs



https://www.vagrantup.com/docs

Vagrant : exemple Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
require 'yaml'
require 'fileutils'

Vagrant.configure(“2” do |config|
  # Box settings
  config.vm.box = "ubuntu/trusty64"

   # Provider Settings
   config.vm.provider "virtualbox" do  |vb|
     vb.name = "vagrant-ubuntu-trust64«
     vb.memory = "1024«
     vb.cpus = "2"
   end
```

R

Vagrant Commandes utiles

➢ Créer une machine virtuelle :

vagrant init : Initialise Vagrant avec un répertoire Vagrantfile et ./.vagrant, en utilisant aucune image de base spécifiée.

vagrant init <chemin vers l'image/box> : initialise Vagrant avec une boîte spécifique.

Exemple : vagrant init hashicorp/precise64

- Démarrer un VM :

  vagrant up : commence l'environnement virtuel Vagrant

  vagrant reload : edémarre la machine, charge la nouvelle configuration de Vagrantfile

  Vagrand reload --provision : redémarre la machine virtuelle et force l'approvisionnement

- Se connecter à une VM :

  vagrant ssh : se connecte à la machine via SSH

- Arrêter une VM :

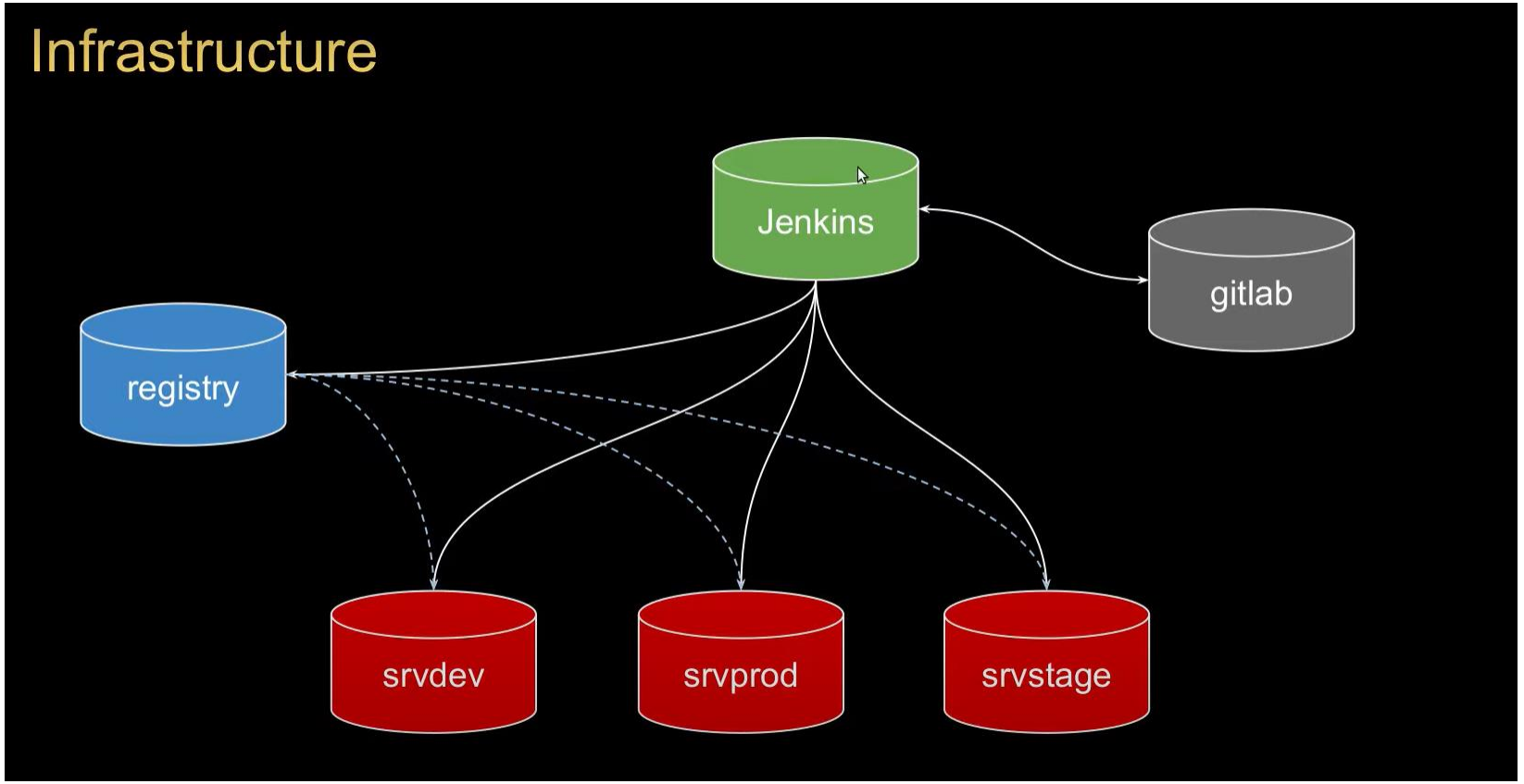
  vagrant halt : arrête la machine virtuelle

  vagrant suspend : suspend une machine virtuelle (sauvegarde l'état)

- Détruire une VM :

  vagrant destroy : arrête et supprime toute trace de la machine virtuelle

Infrastructure



Infrastructure

Vagrant : serveurs applicatif x 3

- 3 environnements : dev / stage / prod(master)
- HW : 1 cpu / 512G ram
- debian buster
- sans docker
- sans java
- connexions ssh user : vagrant / mdp : vagrant

Lien utiles :

- Installation Windows :
https://developer.hashicorp.com/vagrant/downloads

- Installation git bash pour windows :
https://git-scm.com/downloads

Vérification de la version :
vagrant –v

- Récupération projet de démarrage:
git clone https://github.com/fpicot31/Jenkins-vagrant.git

Mettre en place un cluster Kubernetes en local et déployé automatiquement avec Vagrant.

Installation par paquets :
wget https://releases.hashicorp.com/vagrant/2.2.14/vagrant_2.2.14_x86_64.deb

Vérification de la version :
vagrant -v