用 Vagrant 與 Docker 拯救世界

Gea-Suan Lin
KKBOX Technologies

Vagrant

VAGRANT

```
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
    config.vm.box = "hashicorp/precise64"

    config.vm.define "db1" do |db|
        db.vm.hostname = "db1"
        db.vm.provision "shell", path: "db1.sh"
        db.vm.network "private_network", ip: "192.168.50.101"
    end

    config.vm.define "db2" do |db|
        db.vm.hostname = "db2"
        db.vm.provision "shell", path: "db2.sh"
        db.vm.network "private_network", ip: "192.168.50.102"
    end
end
```

用 Ruby 寫的

虛擬化管理工具

包括了...

Image

Network

Environment

Script

. . .

早期只支援 VirtualBox

後來支援 KVM、

VMware 等等方案

包括了今天要提到的

Docker

好處

可重複測試

每次環境相同

設定簡單

Docker


docker

Linux 上輕量級的容器管理工具

(微軟也打算要在

Windows 上實作)

http://www.zdnet.com/article/docker-container-support-coming-to-microsofts-next-windows-server-release/

Microservices

優點

啟動速度快

節省資源

缺點

目前只有 Linux 平台

無法指定容器的
IP 位置

https://github.com/docker/docker/issues/6743

Vagrant + Docker

用途

開發測試環境

Continuous Integration

設定範例

```
ENV["VAGRANT_DEFAULT_PROVIDER"] = "docker"
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
    config.vm.define "db1" do |db|
        db.vm.provision "shell", path: "db1.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker, override|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end

    config.vm.define "db2" do |db|
        db.vm.provision "shell", path: "db2.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end
    end
end
```

```
ENV["VAGRANT_DEFAULT_PROVIDER"] = "docker"
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
    config.vm.define "db1" do |db|
        db.vm.provision "shell", path: "db1.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker, override|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end

    config.vm.define "db2" do |db|
        db.vm.provision "shell", path: "db2.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end
    end
end
```

```
ENV["VAGRANT_DEFAULT_PROVIDER"] = "docker"
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
    config.vm.define "db1" do |db|
        db.vm.provision "shell", path: "db1.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker, override|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end
    end

    config.vm.define "db2" do |db|
        db.vm.provision "shell", path: "db2.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end
    end
end
```

```
ENV["VAGRANT_DEFAULT_PROVIDER"] = "docker"
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
    config.vm.define "db1" do |db|
        db.vm.provision "shell", path: "db1.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker, override|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end

    config.vm.define "db2" do |db|
        db.vm.provision "shell", path: "db2.sh"
        db.vm.synced_folder "shared/", "/srv/shared"
        db.vm.provider "docker" do |docker|
            docker.image = "npoggi/vagrant-docker"
            docker.has_ssh = true
        end
    end
end
```

Summary

We’re hiring

recruit@kkbox.com