VAGRANT

Ferdiansyah Dolot
Raisa Uku

CONTENT

BUT IT WORKED IN MY MACHINE...

Same configuration guides to be followed by the developers

Difficult to manage multiple projects

Traditional approach

Manual setup of the development environment

Difficulties in keeping the environment in sync with other team members

Lead to a difference between development and production environments

WHAT IS VAGRANT

Configuration and provisioning tool

A set of tools that **automates** all the necessary tasks to run the virtual development environment

Downloading, installing, booting VM

It supports all major virtual solutions

Configuring various resources

RAM, CPUs, network connections and shared folders

Provisioning

Shell script &
Software configuration tools

Facilitates **sharing and distributing virtual development** environments

SET UP THE DEVELOPMENT ENVIRONMENT

1. Install the PROVIDER - "back end of Vagrant"
   Vagrant sits on top of existing virtualization solutions.

2. Install Vagrant software

THREE MAIN COMPONENTS OF VAGRANT

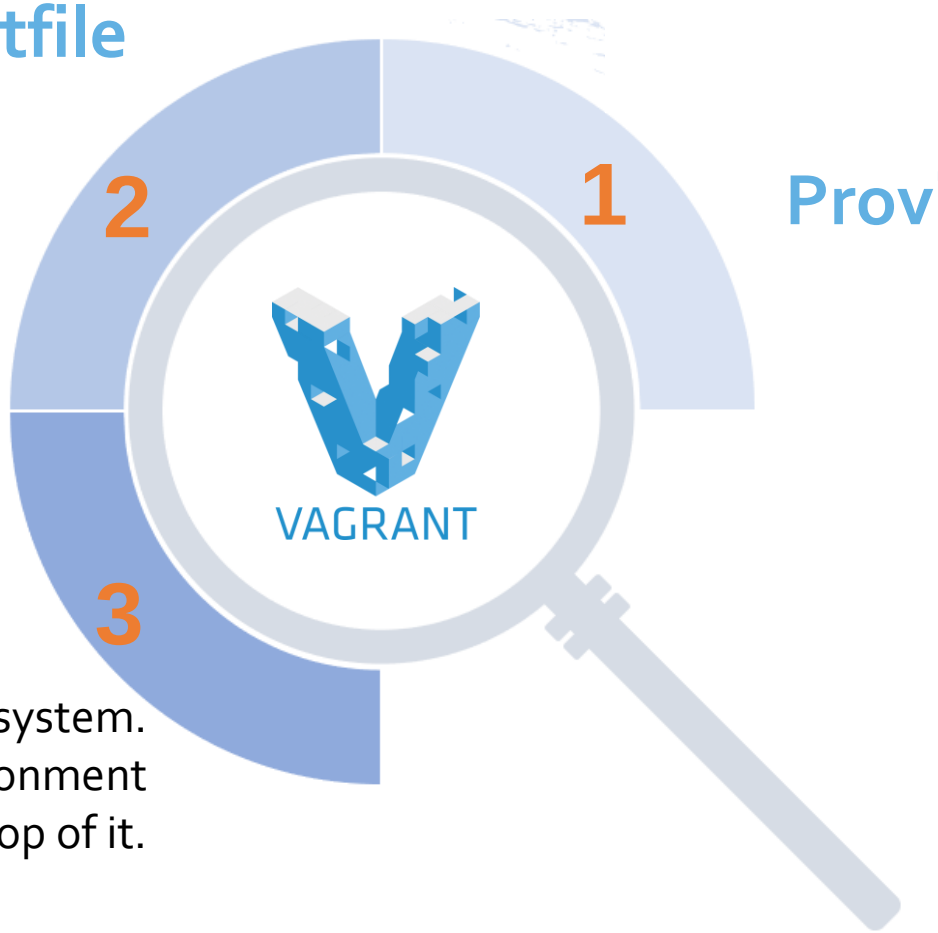
Vagrantfile

Contains the information that Vagrant requires to create the environment.

Ruby

Boxes

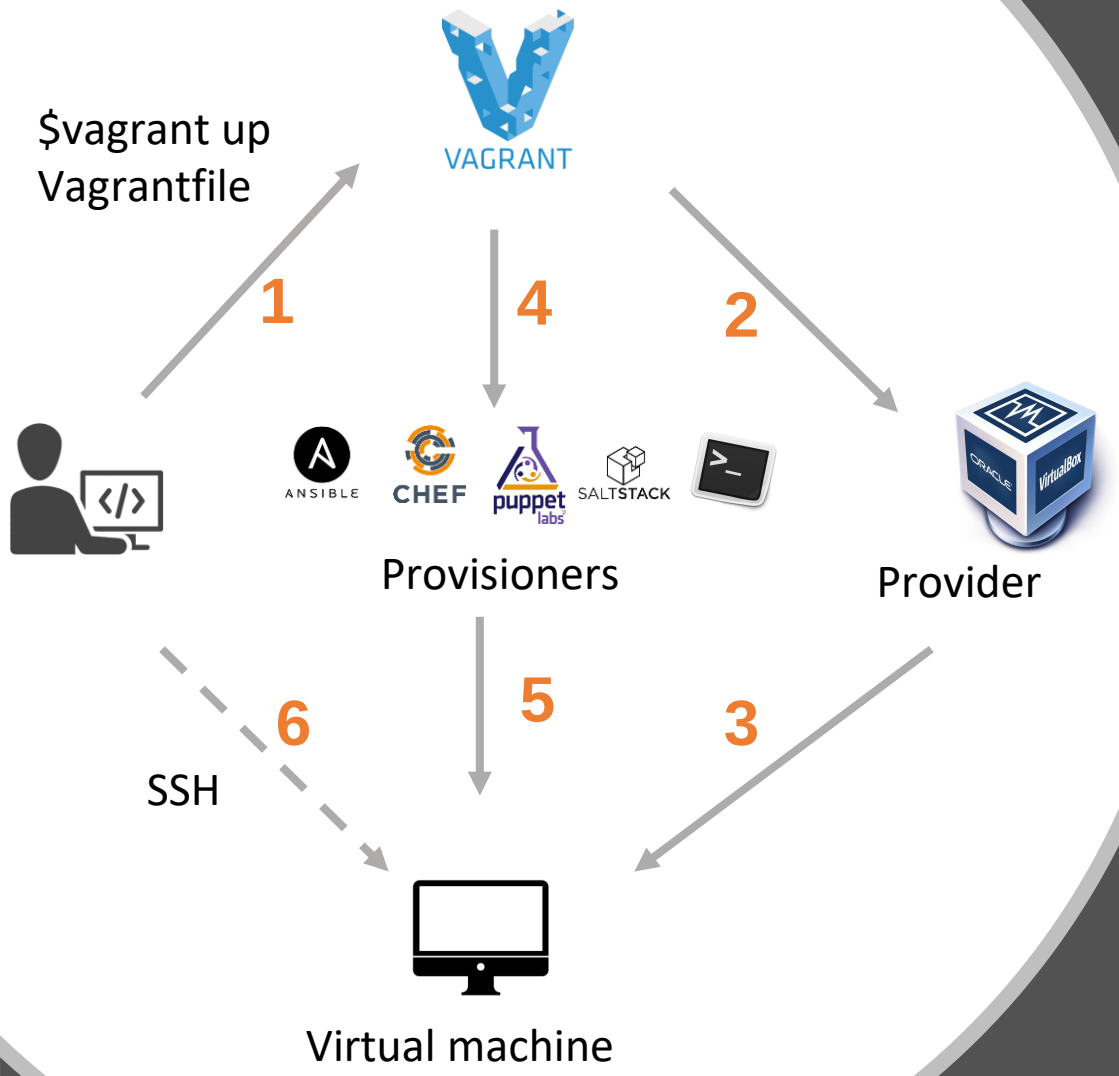
Pre-created base image of operating system.
Used to build the development environment & to run provisioning on top of it.



Providers

VAGRANTFILE AND BOXES


```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure(2) do |config|
      config.vm.box = "precise64"
      config.vm.box_url =
      "http://files.vagrantup.com/precise64.bo
x"
End
```



INITIALIZING DEVELOPMENT ENVIRONMENT

Running vagrant is quite simple:

```
vagrant init precise64
http://files.vagrantup.com/precise64.box
vagrant up
```

Access the development environment:

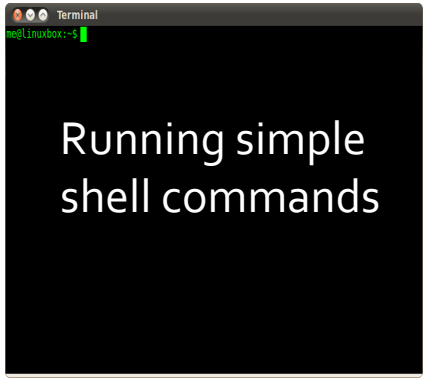
```
vagrant ssh
```

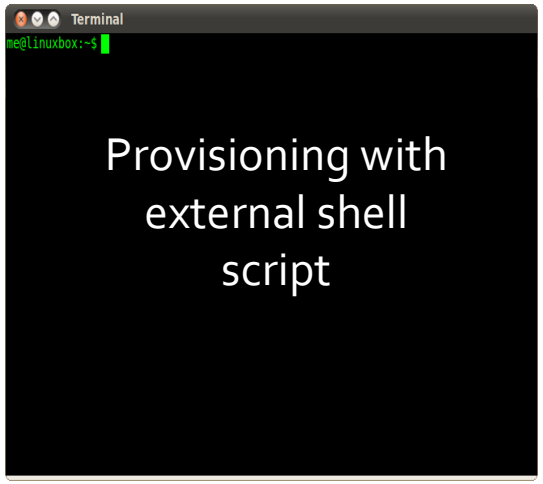
Stop the machine

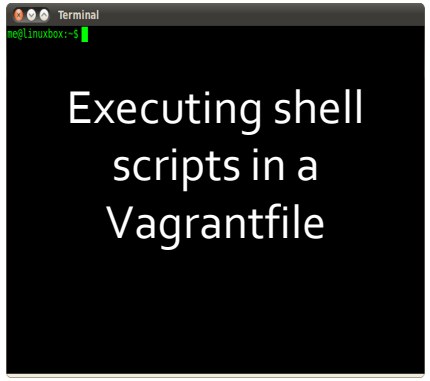
```
vagrant destroy
Vagrant halt
```

PROVISIONING VAGRANT ENVIRONMENT

Basic provisioning with Vagrant using shell scripts





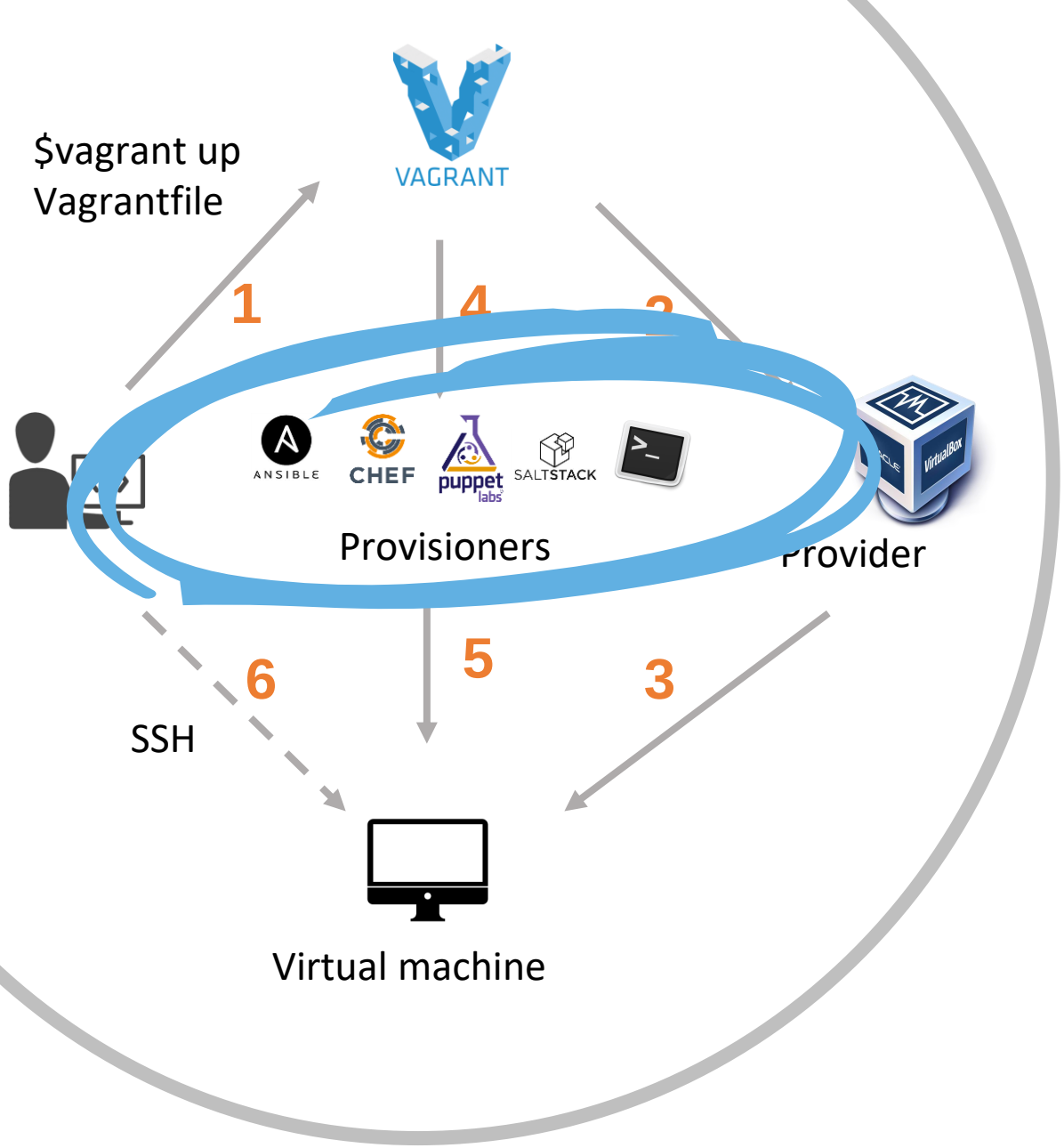


PROVISIONING VAGRANT ENVIRONMENT

Provisioning with external shell script

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure(2) do |config|
    config.vm.box = "precise64"
  config.vm.box_url = "http://files.vagrantup.com/precise64.box"
    config.vm.provision "shell", path: "nginx-install.sh"
    config.vm.network "forwarded_port", guest:80, host:8080
end
```

PROVISIONING VAGRANT ENVIRONMENT



Configuration management tool provisioning

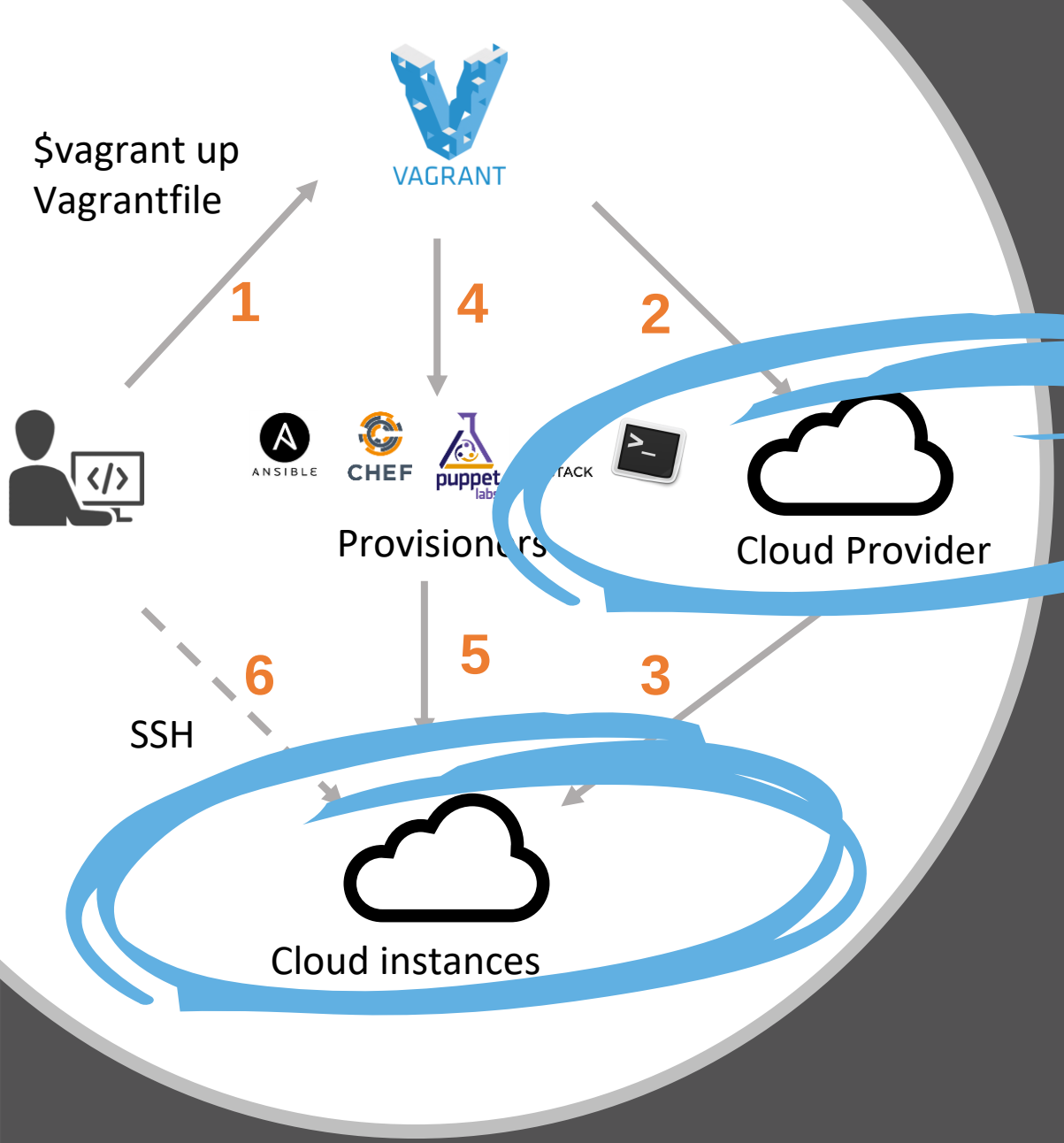
Rich feature
Easier to manage

PROVISIONING VAGRANT ENVIRONMENT

Provisioning with Ansible

```
Vagrant.configure("2") do |config|
 config.vm.provision "ansible" do |ansible|
   ansible.playbook = "playbook.yml"
 end
end
```

VAGRANT IN CLOUD



Control over cloud instances
Provisioning on cloud instances

PROVIDERS

vagrant-aws

vagrant-azure

vagrant-bhyve

vagrant-brightbox

vagrant-cloudstack

vagrant-digitalocean

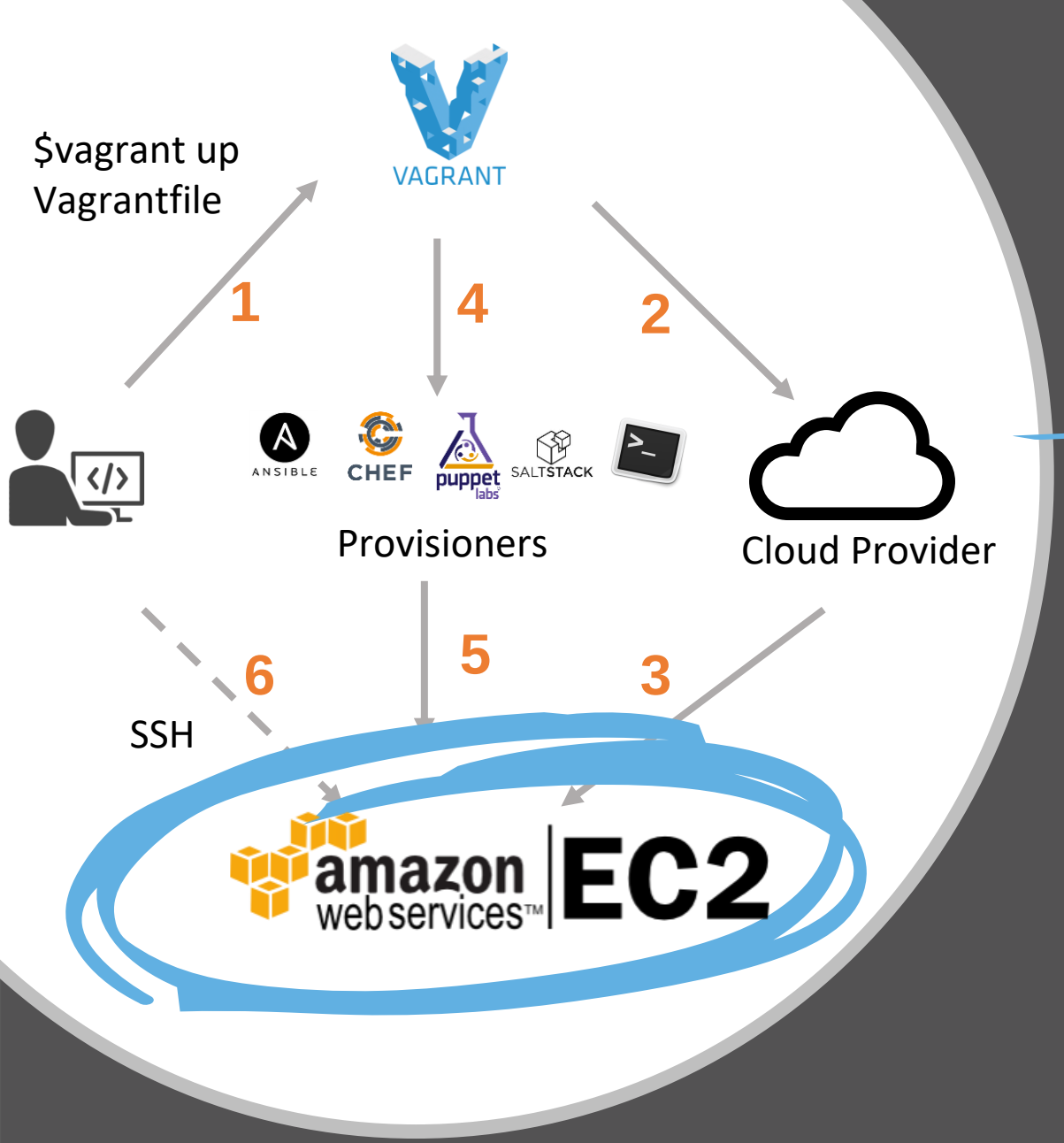
vagrant-ganeti

vagrant-google

vagrant-hp

.....

https://github.com/hashicorp/vagrant/wiki/Available-Vagrant-Plugins



VAGRANT IN AWS

Creating AWS instance
Specify AWS region
Specify security control
Specify AMI
Direct ssh access to EC2 instances

VAGRANT IN AWS

vagrant-aws plugin installation

```
$ vagrant plugin install vagrant-aws
...
$ vagrant up --provider=aws
```

Add dummy box for vm.config.box

```
$ vagrant box add dummy https://github.com/mitchellh/vagrant-
aws/raw/master/dummy.box
```

VAGRANT IN AWS

Defining Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "dummy"

  config.vm.provider :aws do |aws, override|
    aws.access_key_id = "YOUR KEY"
    aws.secret_access_key = "YOUR SECRET KEY"
    aws.keypair_name = "KEYPAIR NAME"
    aws.instance_type = “YOUR INSTANCE TYPE”
    aws.ami = “AMI CODE"
    aws.availability_zone = “YOUR REGION”
    override.ssh.username = “YOUR USERNAME"
    override.ssh.private_key_path = "PATH TO YOUR PRIVATE KEY"
  end
end
```

VAGRANT IN AWS

Instance control

`vagrant up` # starting EC2 instance

`vagrant status` # check status of EC2 instance

```
dolsky@ubuntu:~/Documents/CC/research$ vagrant up
Bringing machine 'default' up with 'aws' provider...
==> default: Warning! The AWS provider doesn't support any of the Vagrant
==> default: high-level network configurations (`config.vm.network`). They
==> default: will be silently ignored.
==> default: Launching an instance with the following settings...
==> default:  -- Type: t2.micro
==> default:  -- AMI: ami-10900f7f
==> default:  -- Region: eu-central-1
==> default:  -- Keypair: cloud-computing
==> default:  -- Security Groups: ["default"]
==> default:  -- Block Device Mapping: []
==> default:  -- Terminate On Shutdown: false
==> default:  -- Monitoring: false
==> default:  -- EBS optimized: false
==> default:  -- Source Destination check:
==> default:  -- Assigning a public IP address in a VPC: false
==> default:  -- VPC tenancy specification: default
==> default: Waiting for instance to become "ready"...
==> default: Waiting for SSH to become available...
:paranoid is deprecated, please use :verify_host_key. Supported values are exactly the same, only the name of the option has changed.
==> default: Machine is booted and ready for use!
==> default: Rsyncing folder: /home/dolsky/Documents/CC/research/ => /vagrant
dolsky@ubuntu:~/Documents/CC/research$ vagrant status
Current machine states:

default                   running (aws)

The EC2 instance is running. To stop this machine, you can run
`vagrant halt`. To destroy the machine, you can run `vagrant destroy`.
```

VAGRANT IN AWS

Instance control

`vagrant ssh` # direct ssh to EC2 instance

`vagrant destroy` # terminate EC2 instance

```
dolsky@ubuntu:~/Documents/CC/research$ vagrant ssh
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-1047-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

96 packages can be updated.
0 updates are security updates.


*** System restart required ***
ubuntu@ip-172-31-33-167:~$ hostname
ip-172-31-33-167
ubuntu@ip-172-31-33-167:~$ exit
logout
Connection to ec2-18-184-152-33.eu-central-1.compute.amazonaws.com closed.
dolsky@ubuntu:~/Documents/CC/research$
```

VAGRANT IN AWS

Other important configuration

security_groups: list of security groups of this instance

monitoring: enable detailed monitoring

private_ip_address: associate private ip address

block_device_mapping: control disk in EC2, e.g. space expansion

elb: elastic load balancer

https://github.com/mitchellh/vagrant-aws

THANK YOU