HashiCorp User Group
AMSTERDAM

https://youtu.be/aJzA-e_3tDA

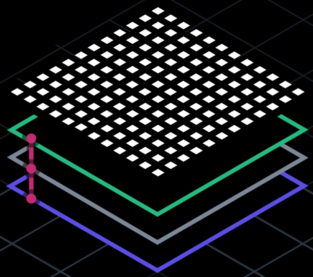
https://github.com/bbaassssiiee/vagransible

@bbaassssiiee

Bas Meijer

DevOps Lead, Schiphol Amsterdam Airport

Wasted decades on late-night hacking



?@bbaassssiiee

HashiCorp

HashiCorp

Image Building

Packer

Vagrant

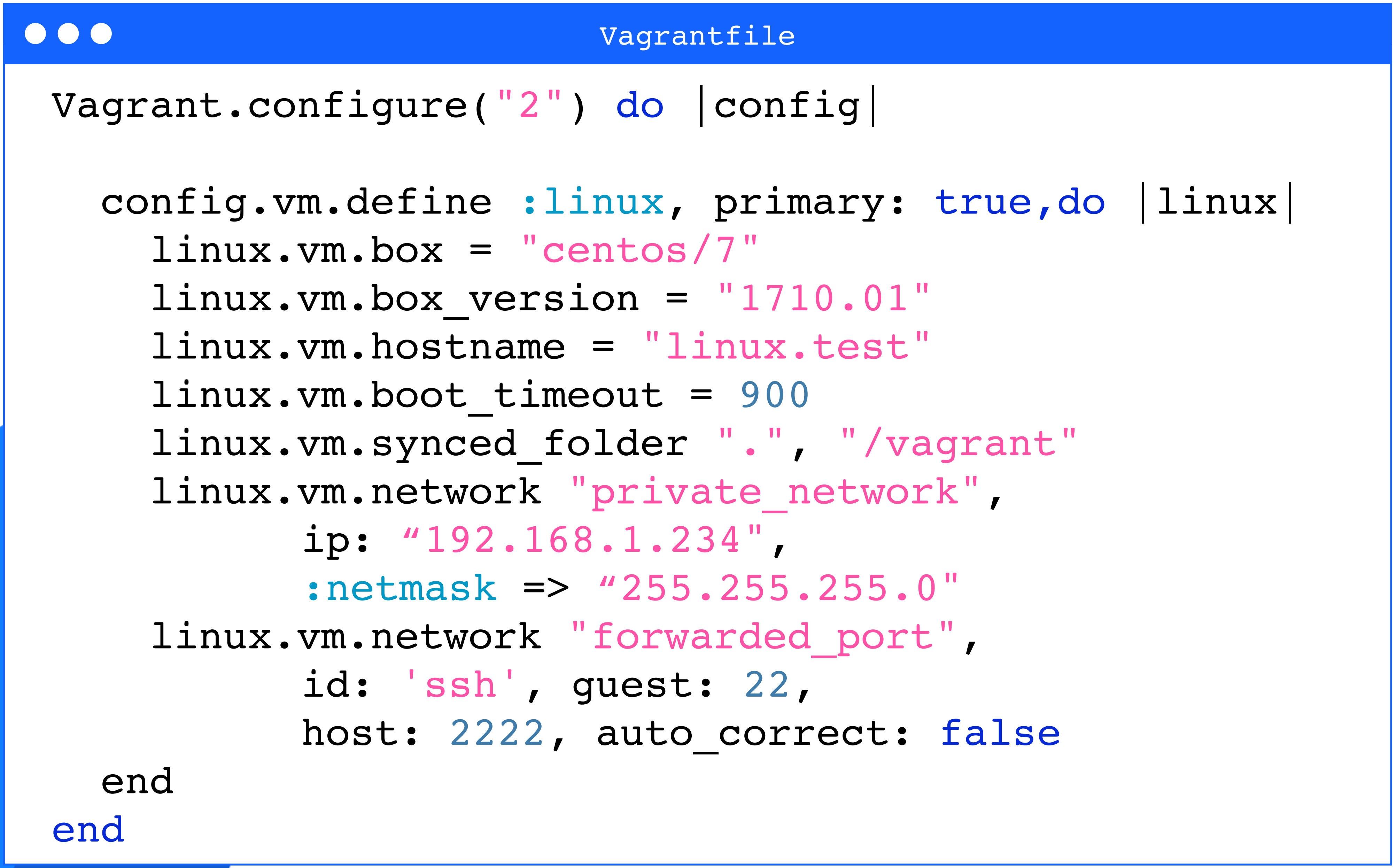
Vagrant fakes IT, until IT makes it!

- Simple idea, but very powerful
- VM`s on laptop model the production environment
- Reproducible Workflows
- Automation with Vagrant

Vagrant

Automate VM workflows

- Vagrantfile in top-level directory
- CLI interface
- Various options
- Multiple VMs
- Multiple Hypervisors
- Virtualbox - VMWare - KVM

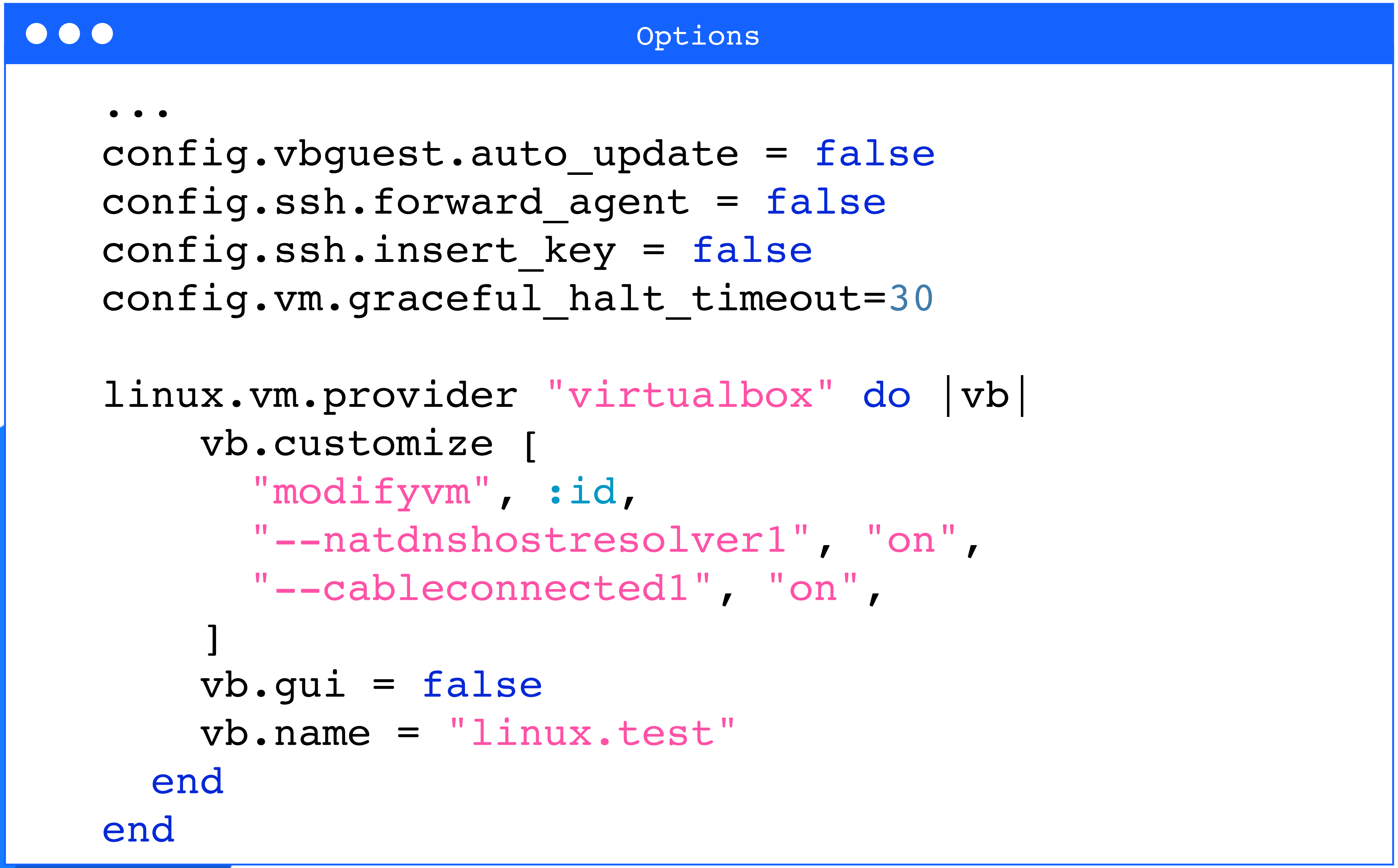


```
Vagrant.configure("2") do |config|

  config.vm.define :linux, primary: true,do |linux|
    linux.vm.box = "centos/7"
    linux.vm.box_version = "1710.01"
    linux.vm.hostname = "linux.test"
    linux.vm.boot_timeout = 900
    linux.vm.synced_folder ".", "/vagrant"
    linux.vm.network "private_network",
          ip: “192.168.1.234",
          :netmask => “255.255.255.0"
    linux.vm.network "forwarded_port",
          id: 'ssh', guest: 22,
          host: 2222, auto_correct: false
  end
end
```

@bbaassssiiee

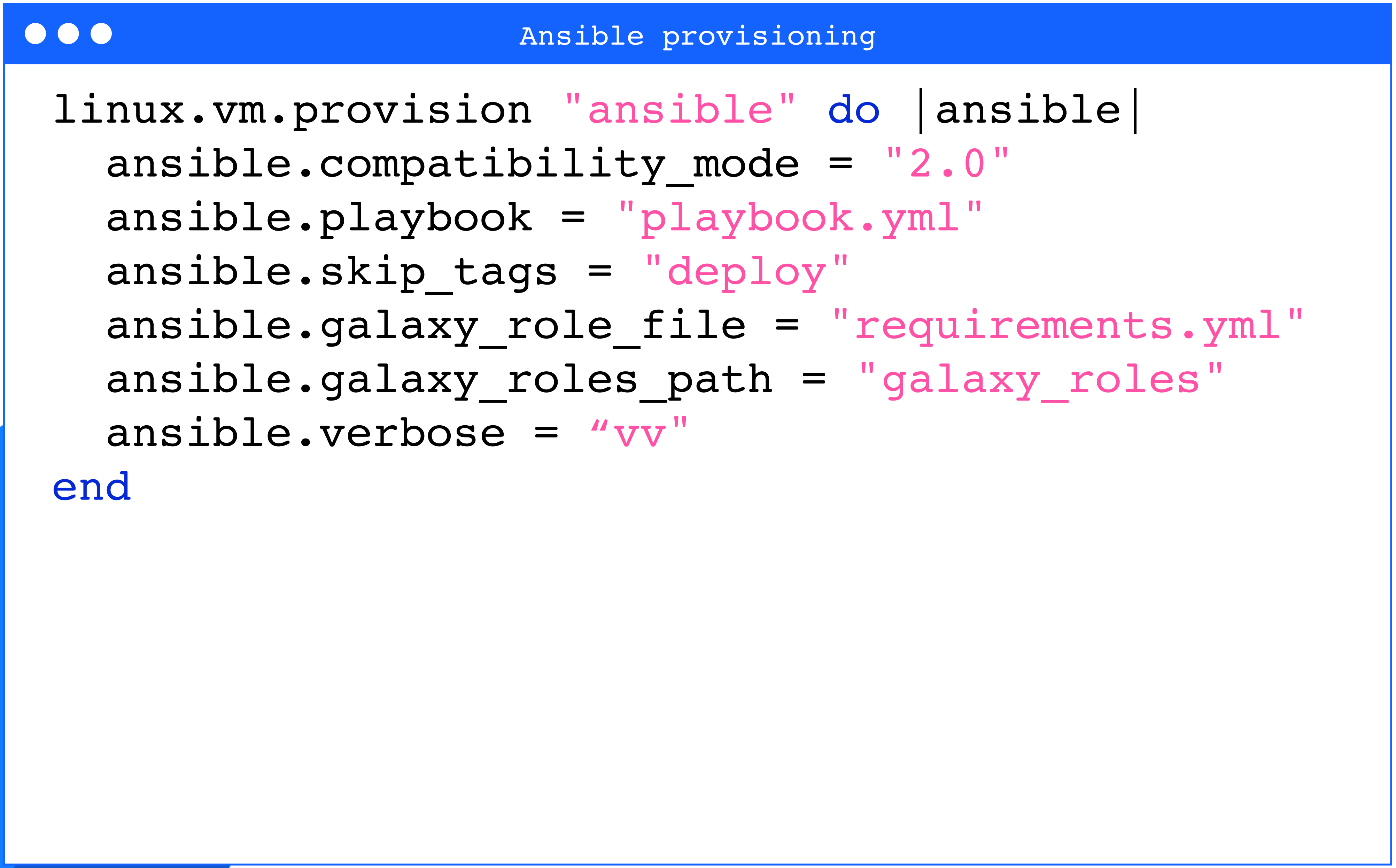
HashiCorp



```
...
config.vbguest.auto_update = false
config.ssh.forward_agent = false
config.ssh.insert_key = false
config.vm.graceful_halt_timeout=30

linux.vm.provider "virtualbox" do |vb|
    vb.customize [
      "modifyvm", :id,
      "--natdnshostresolver1", "on",
      "--cableconnected1", "on",
    ]
    vb.gui = false
    vb.name = "linux.test"
  end
end
```

@bbaassssiiee



```
linux.vm.provision "ansible" do |ansible|
  ansible.compatibility_mode = "2.0"
  ansible.playbook = "playbook.yml"
  ansible.skip_tags = "deploy"
  ansible.galaxy_role_file = "requirements.yml"
  ansible.galaxy_roles_path = "galaxy_roles"
  ansible.verbose = “vv"
end
```

```
$ vagrant up

$ vagrant ssh

$ vagrant suspend

$ vagrant destroy -f

$ vagrant up

$ vagrant halt
```

Packer

VM`s hide problems with

hardware

Clouds: other people’s hypervisors

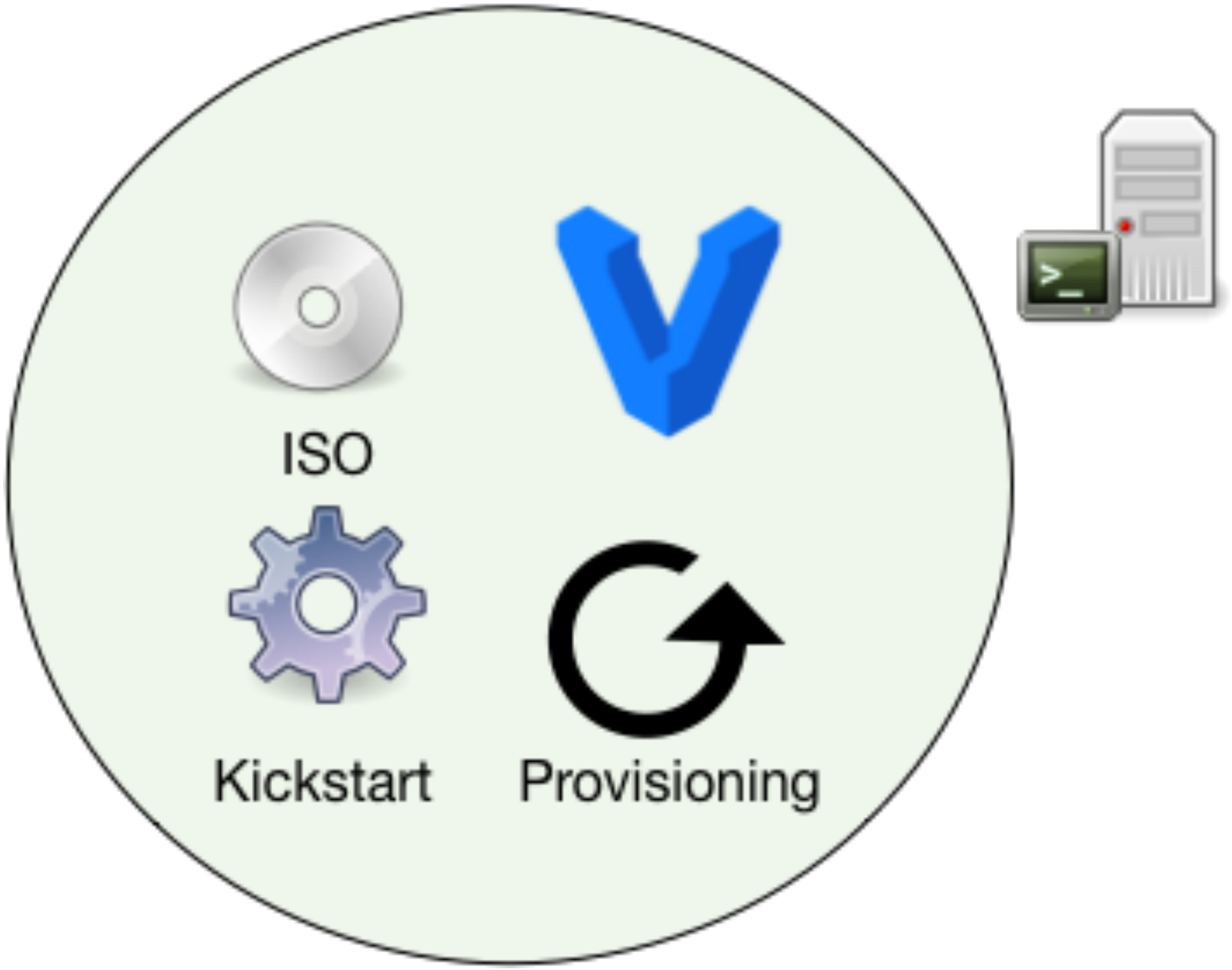
Each cloud has its own problems

Packer

Packer has Builders!

Azure	VMWare
AWS	DigitalOcean
ESX	Google
HYPER-V	OpenStack
CloudStack	VirtualBox

Vagrant VM Image Workflow



Packer

Provisioner agnostic

Ansible

Bash

Chef

Puppet

Packer

What do you need?

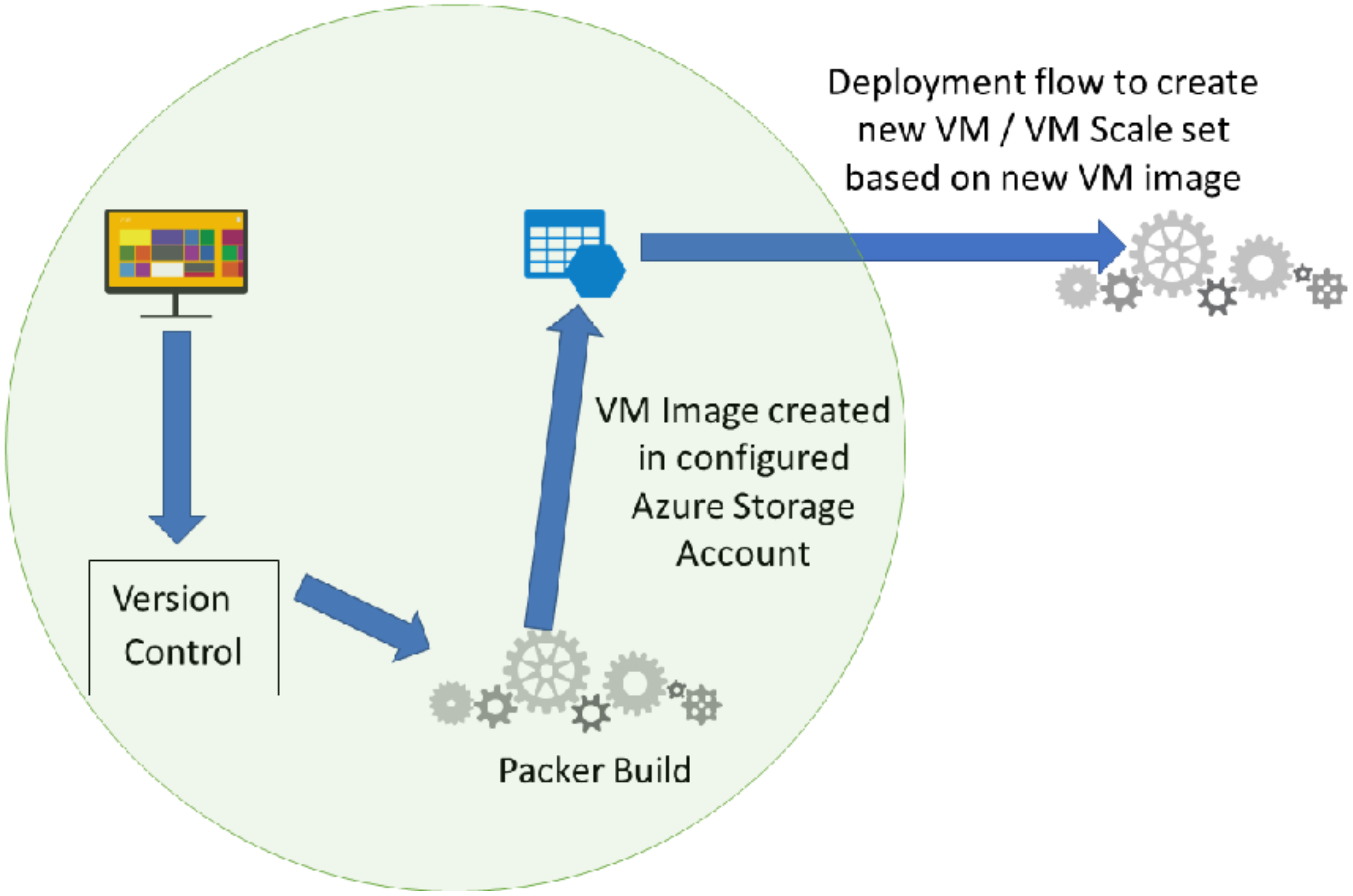
- Packer
- Vagrant
- Ansible
- Access to cloud
- Patience

What does Azure need?

Azure

- subscription_id
- client_id
- tenant_id
- resource_group
- storage_account
- client_secret

Azure VM Image Workflow



Deployment flow to create new VM / VM Scale set based on new VM image

VM Image created in configured Azure Storage Account

Version Control

Packer Build

@bbaassssiiee

```
      {
        "boot_command": [
          "<tab> append initrd=initrd.img inst.text inst.ks=http://
{{ .HTTPIP }}:{{ .HTTPPort }}/centos7.ks<enter><wait>"
        ],
        "boot_wait": "20s",
        "guest_additions_path": "VBoxGuestAdditions.iso",
        "guest_os_type": "RedHat_64",
        "headless": false,
        "http_directory": "kickstart",
        "iso_checksum":
"27bd866242ee058b7a5754e83d8ee8403e216b93d130d800852a96f41c34d86a",
        "iso_checksum_type": "sha256",
        "iso_urls": [
          "file:///Users/Shared/CentOS-7-x86_64-Minimal-1611.iso",
        ],
        "output_directory": "output-virtualbox-iso",
        "shutdown_command": "echo 'vagrant' | sudo -S /sbin/shutdown -h 0",
        "ssh_password": "vagrant",
        "ssh_port": 22,
        "ssh_username": "root",
        "ssh_wait_timeout": "10000s",
        "type": "virtualbox-iso",
        "virtualbox_version_file": ".vbox_version",
        "vm_name": "dockpack/centos7"
      }
```

@bbaassssiiee

HashiCorp

Provisioning with DOCKPACK Ansible Roles

```
"provisioners": [
  {
   "execute_command": "echo 'vagrant' | {{.Vars}} sudo -S -E bash
'{{.Path}}'",
   "scripts": [
    "scripts/vagrant.sh",
    "scripts/vmtools.sh"
   ],
   "type": "shell"
  },{
   "playbook_file": "ansible/packer.yml",
   "role_paths": [
    "ansible/roles/dockpack.base_docker",
    "ansible/roles/dockpack.base_java8"
   ],
   "type": "ansible-local"
  },
  {
   "execute_command": "echo 'vagrant' | {{.Vars}} sudo -S -E bash
'{{.Path}}'",
   "only": [
    "vmware-iso"
   ],
   "script": "scripts/cleanup.sh",
   "type": "shell"
  }
 ]
```

```
{
  "azure_tags": {
    "environment": "{{user `env`}}",
    "organization": "snbv",
    "product": "centos7image"
  },
  "client_id": "{{user `client_id`}}",
  "client_secret": "{{user `client_secret`}}",
  "image_offer": "CentOS",
  "image_publisher": "OpenLogic",
  "image_sku": "7.3",
  "location": "West Europe",
  "managed_image_name": "centos7",
  "managed_image_resource_group_name": "{{user `resource_group`}}",
  "os_type": "Linux",
  "subscription_id": "{{user `subscription_id`}}",
  "tenant_id": "{{user `tenant_id`}}",
  "type": "azure-arm",
  "vm_size": "Standard_DS2_v2"
}
```

Provisioning with DOCKPACK Ansible Roles

```
"provisioners": [
    {
      "execute_command": "echo 'vagrant' | {{.Vars}} sudo -S -E bash
'{{.Path}}'",
      "scripts": [
        "scripts/ansible.sh"
      ],
      "type": "shell"
    },{
      "playbook_file": "ansible/packer.yml",
      "role_paths": [
        "ansible/roles/dockpack.base_docker",
        "ansible/roles/dockpack.base_java8"
      ],
      "type": "ansible-local"
    },{
      "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo -E sh
'{{ .Path }}'",
      "inline": [
        "/usr/sbin/waagent -force -deprovision+user && export HISTSIZE=0
&& sync"
      ],
      "inline_shebang": "/bin/sh -x”,
      "type": "shell"
    }
  ]
```

@bbaassssiiee

Azure

Using the VM image

```
az vm create \
--resource-group my-demo-lab \
 --name my-demo-vm \
--image my-centos7-image \
--admin-username bbaassssiiee \
--generate-ssh-keys
```

@bbaassssiiee

```
resource "azurerm_virtual_machine" "virtual_machine" {
  count                 = "${var.machine_count}"
  name                  = "${local.machine_identifier}-${count.index +
1}"
  location              = "$
{azurerm_resource_group.resource_group.location}"
  resource_group_name   = "$
{azurerm_resource_group.resource_group.name}"
  vm_size               = "${var.machine_size}"
  network_interface_ids = ["$
{element(azurerm_network_interface.network_interface.*.id,
count.index)}"]

  os_profile_linux_config {
    disable_password_authentication = true

    ssh_keys {
      key_data = "${file("${var.ssh_key_path}")}"
      path     = "/home/${var.admin_username}/.ssh/authorized_keys"
    }
  }

  storage_os_disk {
    name          = "${local.machine_identifier}-osdisk1"
    image_uri     = "https://myinfratmp.blob.core.windows.net/system/
Microsoft.Compute/Images/images/packer-osDisk.850f2a37-a2eb-491e-
ab8c-0f8282b4f97e.vhd"
    vhd_uri       = "https://$
{azurerm_storage_account.storage_account.name}.blob.core.windows.net/
vhds/${local.machine_identifier}-osdisk.vhd"
    os_type       = "Linux"
    caching       = "ReadWrite"
    create_option = "FromImage"
  }
```

HashiCorp User Group
AMSTERDAM

https://youtu.be/aJzA-e_3tDA

https://github.com/bbaassssiiee/vagransible

@bbaassssiiee