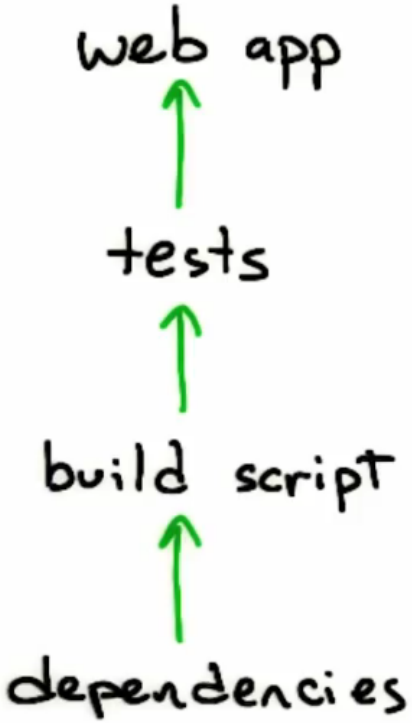
Packer Vagrant.
Machine image creation automation

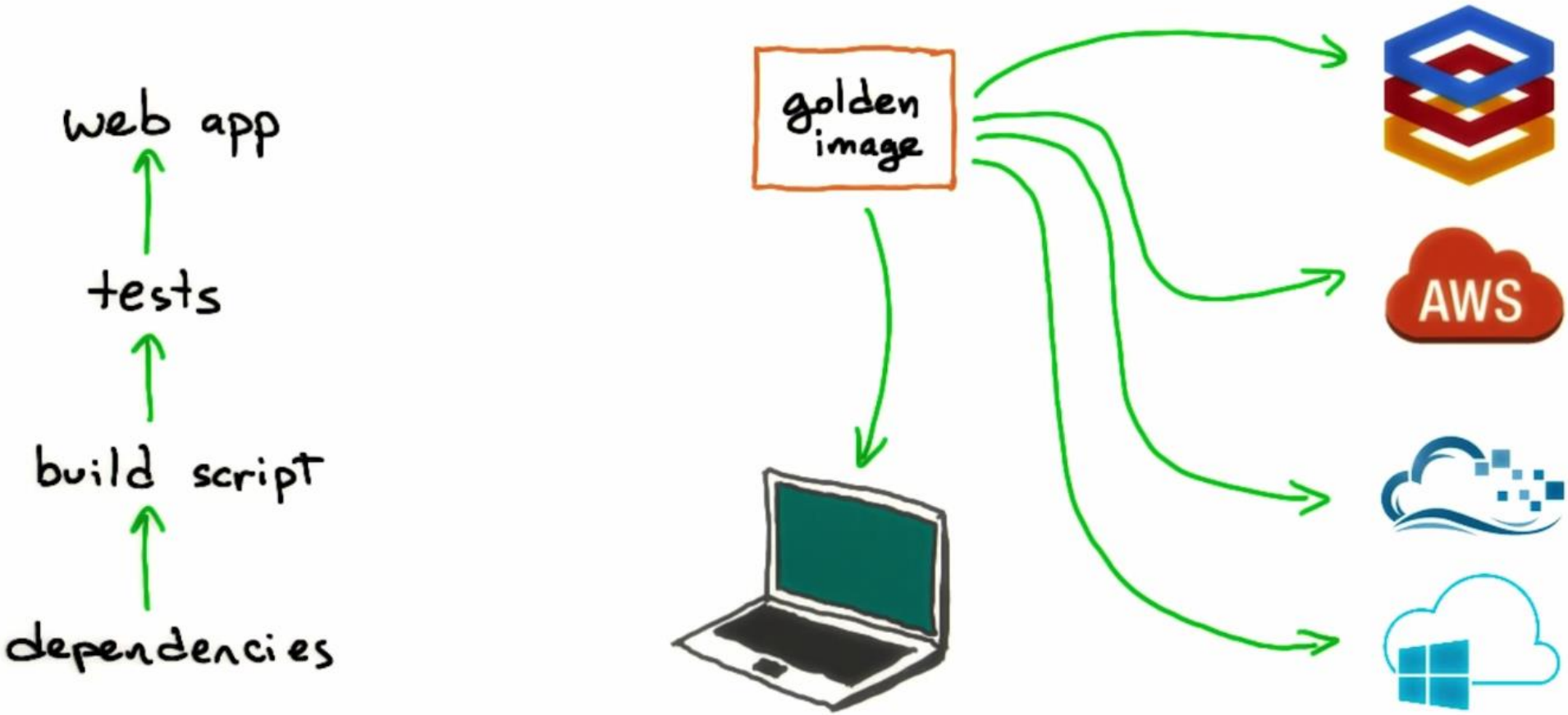
Here is an example workflow:

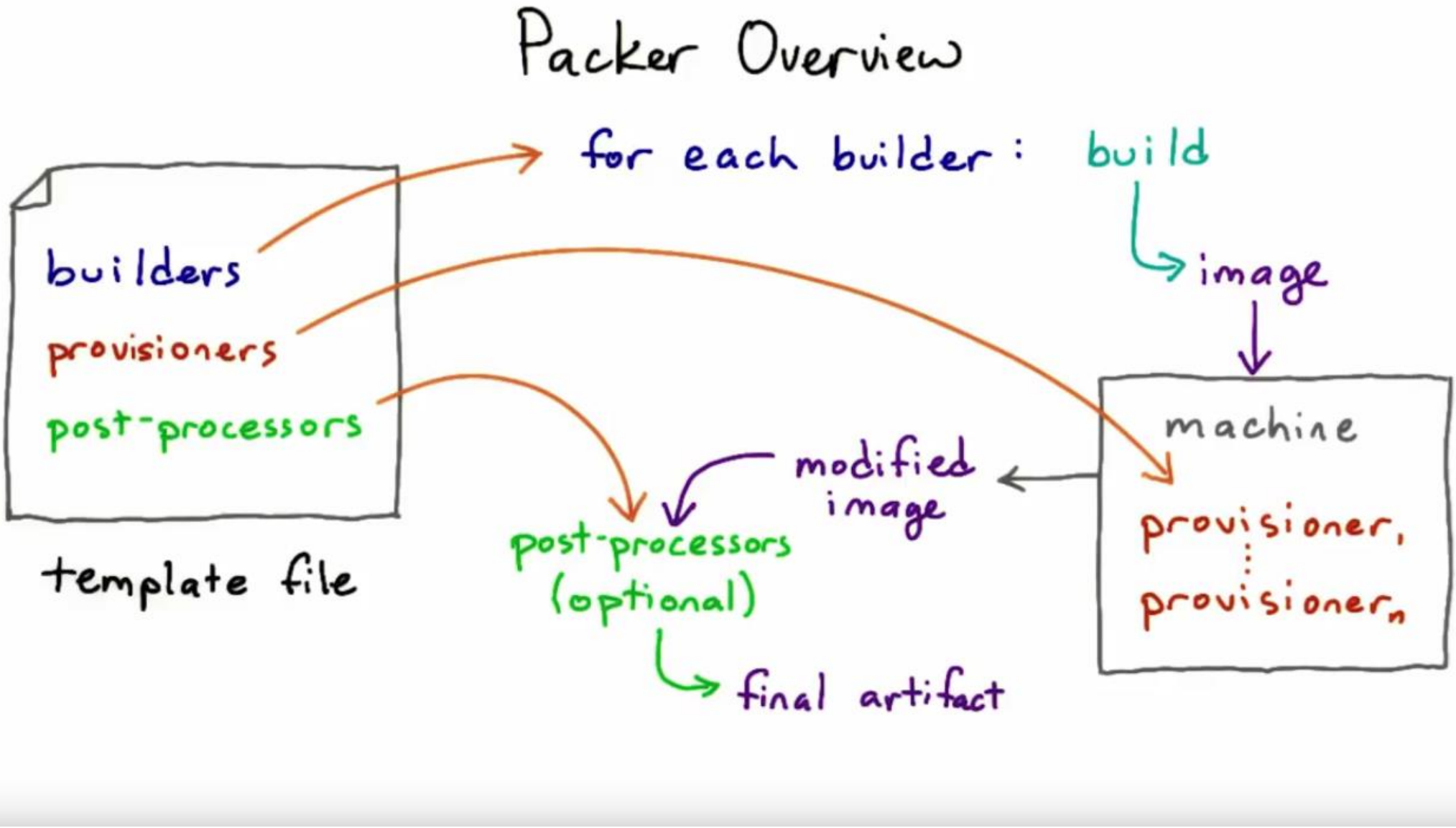
- You use Packer to build a Vagrant Box for the **virtualbox** provider
- The vagrant-cloud post-processor is configured to point to the box **hashicorp/foobar** on Vagrant Cloud via the **box_tag** configuration
- The post-processor receives the box from the vagrant post-processor
- It then creates the configured version, or verifies the existence of it, on Vagrant Cloud
- A provider matching the name of the Vagrant provider is then created
- The box is uploaded to Vagrant Cloud
- The upload is verified
- The version is released and available to users of the box

Project Overview



Project Overview




Packer Overview
for each builder :
build
image
builders
provisioners
post-processors
machine
modified
image
provisioner,
provisioner,
post-processors
(optional)
template file
final artifact

application-server.json

```
{
  "variables": {
      "PACKER_OS_FLAVOUR": "ubuntu",
      "PACKER_BOX_NAME": "ubuntu-14.04.2-server-amd64",
      "AWS_ACCESS_KEY_ID": "{{env `AWS_ACCESS_KEY_ID`}}",
      "AWS_SECRET_ACCESS_KEY": "{{env `AWS_SECRET_ACCESS_KEY`}}"
    },
    "builders": [
      {
        "type": "virtualbox-iso",
        "boot_command": [
          "<esc><wait>",
          "<esc><wait>",
          "<enter><wait>",
          "/install/vmlinuz<wait>",
          " auto<wait>",
          " console-setup/ask_detect=false<wait>",
          " console-setup/layoutcode=us<wait>",
          " console-setup/modelcode=pc105<wait>",
          " debconf/frontend=noninteractive<wait>",
          " debian-installer=en_US<wait>",
          " fb=false<wait>",
          " initrd=/install/initrd.gz<wait>",
          " kbd-chooser/method=us<wait>",
          " keyboard-configuration/layout=USA<wait>",
          " keyboard-configuration/variant=USA<wait>",
          " locale=en_US<wait>",
```

application-server.json

```
      " noapic<wait>",
      " preseed/url=http://{{ .HTTPIP }}:{{ .HTTPPort }}/preseed.cfg<wait>",
      " -- <wait>",
      "<enter><wait>"
    ],
    "boot_wait": "10s",
    "headless": false,
    "disk_size": 10140,
    "guest_additions_path": "VBoxGuestAdditions_{{.Version}}.iso",
    "guest_os_type": "Ubuntu_64",
    "http_directory": "http",
    "iso_checksum": "8acd2f56bfcba2f7ac74a7e4a5e565ce68c024c38525c0285573e41c86ae90c0",
    "iso_checksum_type": "sha256",
    "iso_url": "http://releases.ubuntu.com/trusty/{{ user `PACKER_BOX_NAME` }}.iso",
    "shutdown_command": "echo 'vagrant'|sudo -S shutdown -P now",
    "ssh_password": "vagrant",
    "ssh_port": 22,
    "ssh_username": "vagrant",
    "ssh_wait_timeout": "10000s",
    "type": "virtualbox-iso",
    "vm_name": "{{ user `PACKER_BOX_NAME` }}",
    "vboxmanage": [
      ["modifyvm", "{{.Name}}", "--memory", "1024"],
      ["modifyvm", "{{.Name}}", "--cpus", "2"]
    ],
    "virtualbox_version_file": ".vbox_version"
  },
```

application-server.json

```
68
69  "provisioners": [
70    {
71      "type": "shell",
72      "execute_command": "echo 'vagrant'|{{.Vars}} sudo -S -E bash '{{.Path}}'",
73      "scripts": [
74        "scripts/update.sh"
75      ]
76    },
77    {
78    "type": "shell",
79    "only": ["virtualbox-iso"],
80    "execute_command": "echo 'vagrant'|sudo -S -E bash '{{.Path}}'",
81    "scripts": [
82    "scripts/virtualbox.sh",
83    "scripts/vagrant.sh"
84    ]
85    },
86    {
87      "type": "shell",
88      "execute_command": "echo 'vagrant'|sudo -S -E bash '{{.Path}}'",
89      "scripts": [
90        "scripts/application.sh",
91        "scripts/cleanup.sh"
92      ]
93    }
94  ],
```

application-server.json

```
76      },
77      {
78      "type": "shell",
79      "only": ["virtualbox-iso"],
80      "execute_command": "echo 'vagrant'|sudo -S -E bash '{{.Path}}'",
81      "scripts": [
82      "scripts/virtualbox.sh",
83      "scripts/vagrant.sh"
84      ]
85      },
86      {
87        "type": "shell",
88        "execute_command": "echo 'vagrant'|sudo -S -E bash '{{.Path}}'",
89        "scripts": [
90          "scripts/application.sh",
91          "scripts/cleanup.sh"
92        ]
93      }
94    ],
95    "post-processors": [
96      [
97        {
98          "type": "vagrant",
99          "compression_level": "9",
100         "output": "{{.Provider}}/{{ user `PACKER_BOX_NAME` }}-appserver_{{.Provider}}.box"
101       }
102     ]
```

Packer Overview

How would you switch from Ubuntu to CentOS in Packer?

- [ ] Run Packer on a machine or VM running CentOS.
- [ ] Change the source in the builder configuration.
- [ ] Change the source in the provisioner configuration.
- [ ] Change the target in the post-processor configuration.
- [ ] You can't; Packer only supports Ubuntu.

Packer Overview

How would you switch from Ubuntu to CentOS in Packer?

- [ ] Run Packer on a machine or VM running CentOS.
- [x] Change the source in the builder configuration.
- [ ] Change the source in the provisioner configuration.
- [ ] Change the target in the post-processor configuration.
- [ ] You can't; Packer only supports Ubuntu.

IT Orchestration

Orchestration is the automated configuration, coordination and management of computer systems and software.

Examples of tools: Ansible, Puppet, Salt, Terraform, AWS CloudFormation.

Container Orchestration: Kubernetes software, managed services as AWS EKS, AWS ECS, Amazon Fargate.