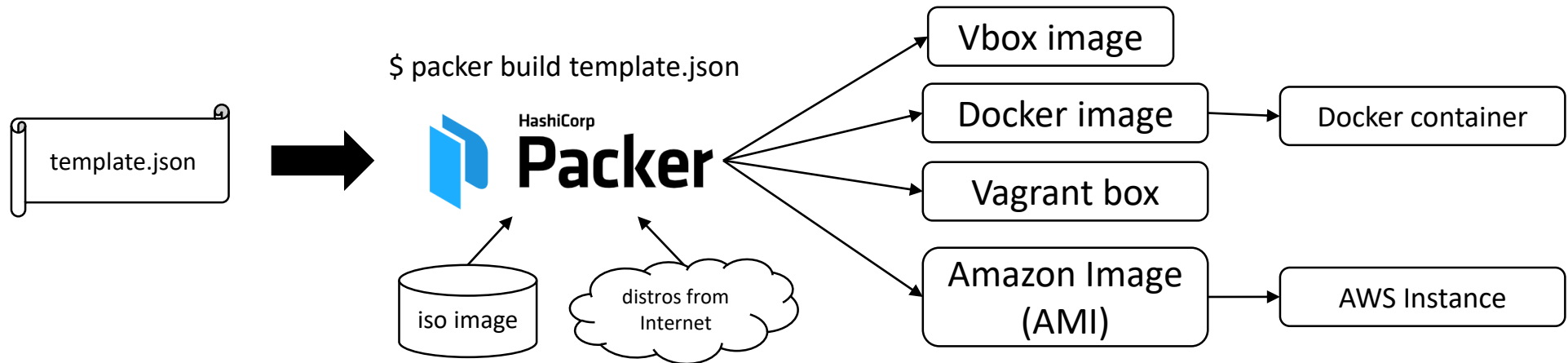


Packer.

Machine image creation automation

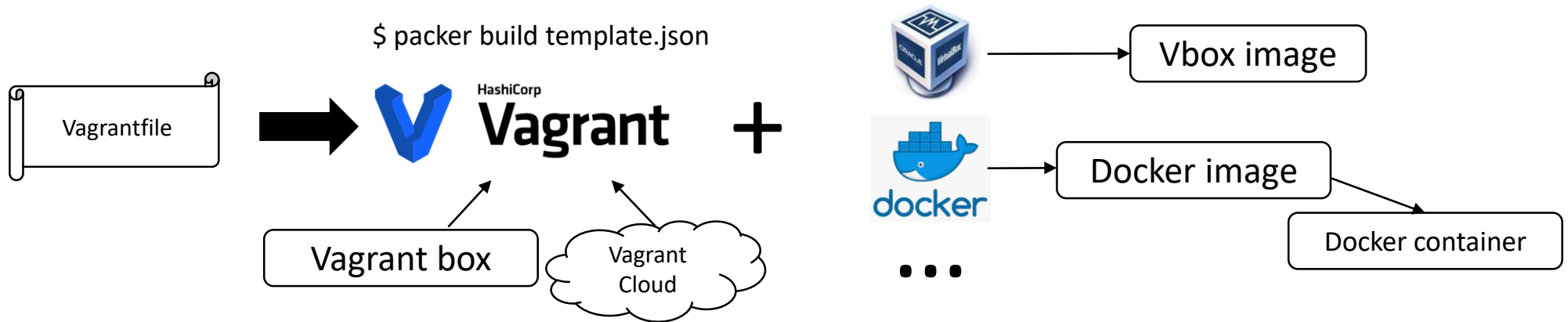


Here is an example workflow:

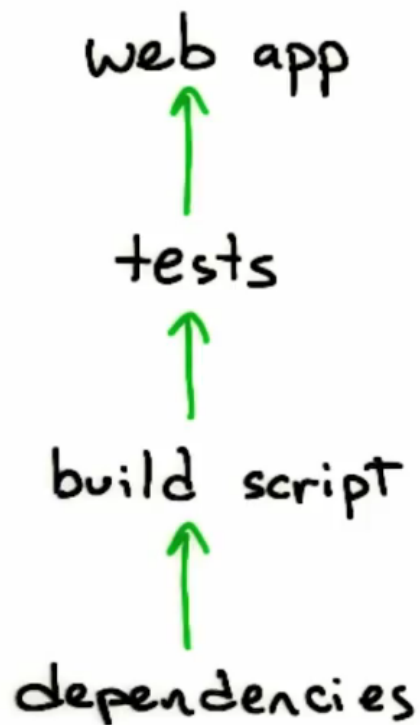
- You use Packer to build a Vagrant Box for the **virtualbox** provider
- The vagrant-cloud post-processor is configured to point to the box **hashicorp/foobar** on Vagrant Cloud via the **box_tag** configuration
- The post-processor receives the box from the vagrant post-processor
- It then creates 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

Vagrant.

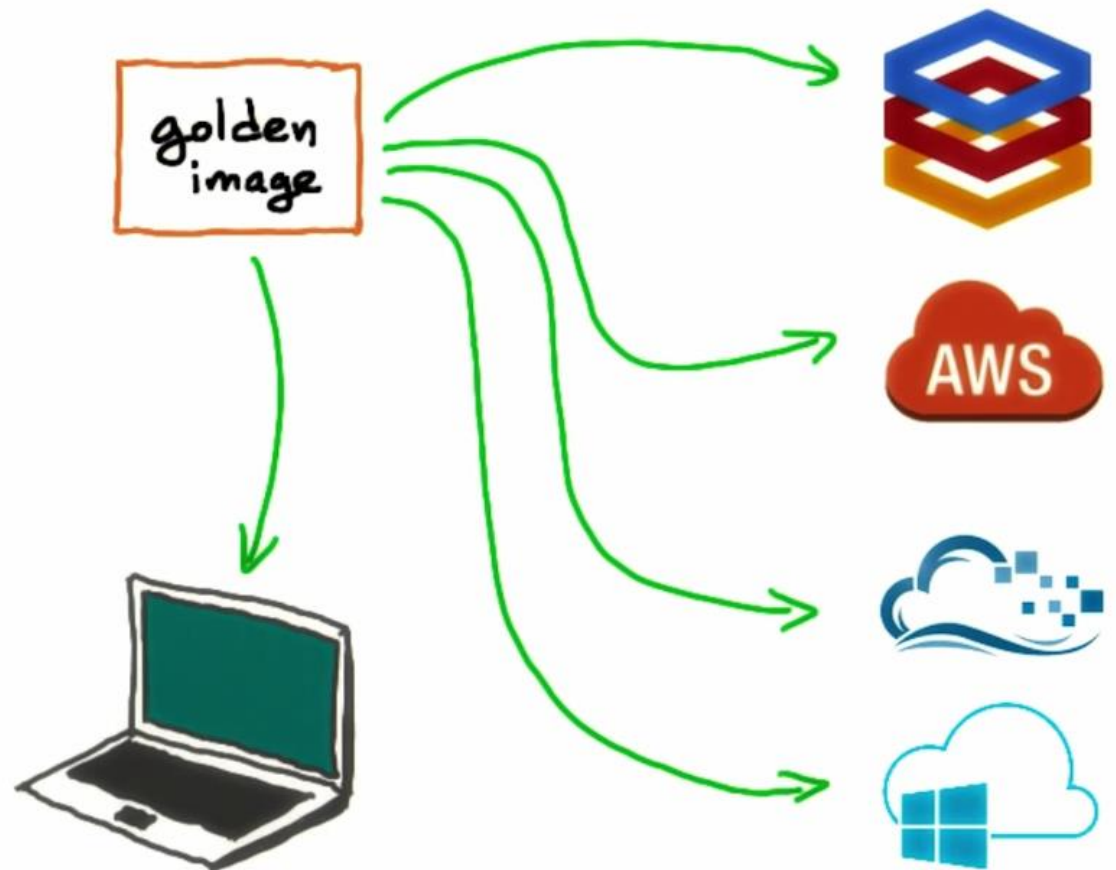
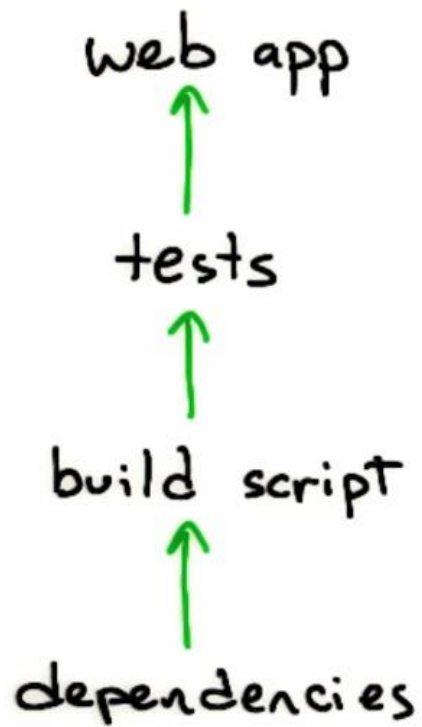
Machine image management and creation automation



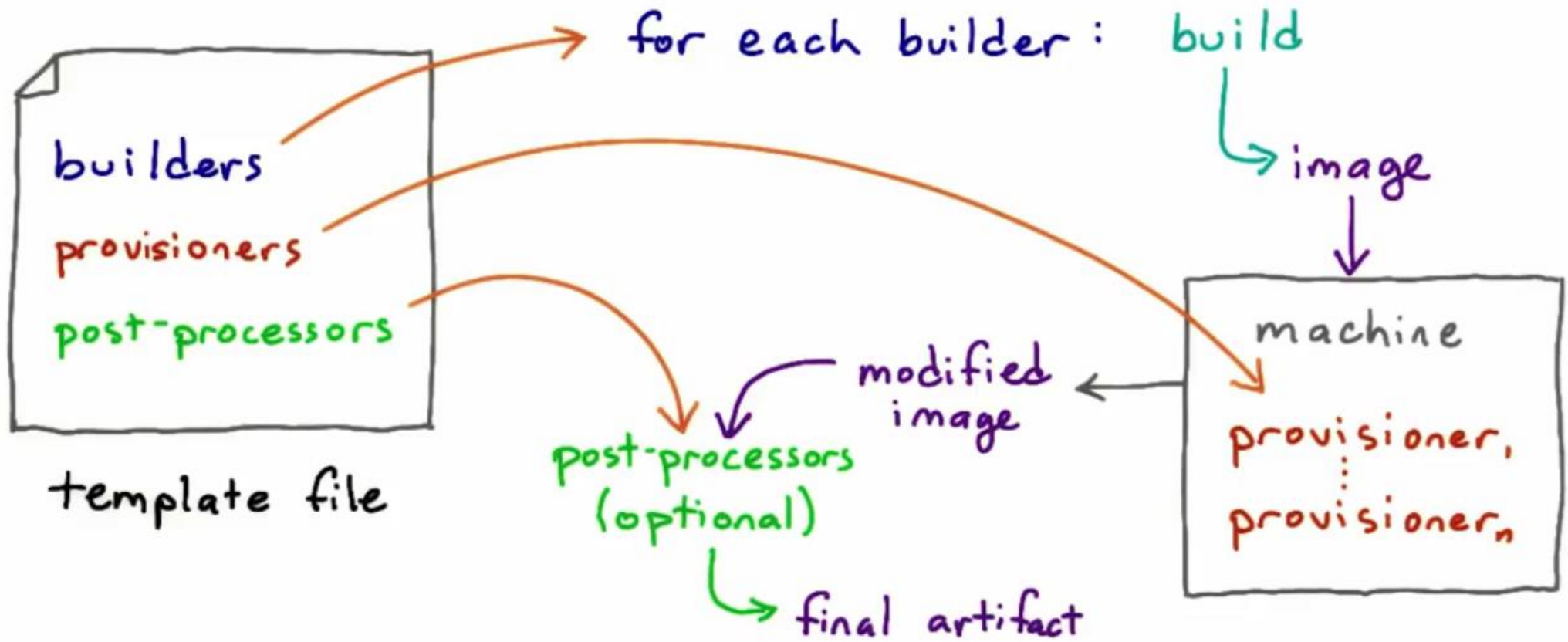
Project Overview



Project Overview



Packer Overview



```
1 {
2   "variables": {
3     "PACKER_OS_FLAVOUR": "ubuntu",
4     "PACKER_BOX_NAME": "ubuntu-14.04.2-server-amd64",
5     "AWS_ACCESS_KEY_ID": "{{env `AWS_ACCESS_KEY_ID`}}",
6     "AWS_SECRET_ACCESS_KEY": "{{env `AWS_SECRET_ACCESS_KEY`}}"
7   },
8   "builders": [
9     {
10      "type": "virtualbox-iso",
11      "boot_command": [
12        "<esc><wait>",
13        "<esc><wait>",
14        "<enter><wait>",
15        "/install/vmlinuz<wait>",
16        " auto<wait>",
17        " console-setup/ask_detect=false<wait>",
18        " console-setup/layoutcode=us<wait>",
19        " console-setup/modelcode=pc105<wait>",
20        " debconf/frontend=noninteractive<wait>",
21        " debian-installer=en_US<wait>",
22        " fb=false<wait>",
23        " initrd=/install/initrd.gz<wait>",
24        " kbd-chooser/method=us<wait>",
25        " keyboard-configuration/layout=USA<wait>",
26        " keyboard-configuration/variant=USA<wait>",
27        " locale=en_US<wait>",
```



```
30     " noapic<wait>",
31     " preseed/url=http://{{ .HTTPIP }}:{{ .HTTPPort }}/preseed.cfg<wait>",
32     " -- <wait>",
33     "<enter><wait>"
34 ],
35 "boot_wait": "10s",
36 "headless": false,
37 "disk_size": 10140,
38 "guest_additions_path": "VBoxGuestAdditions_{{.Version}}.iso",
39 "guest_os_type": "Ubuntu_64",
40 "http_directory": "http",
41 "iso_checksum": "8acd2f56bfcba2f7ac74a7e4a5e565ce68c024c38525c0285573e41c86ae90c0",
42 "iso_checksum_type": "sha256",
43 "iso_url": "http://releases.ubuntu.com/trusty/{{ user `PACKER_BOX_NAME` }}.iso",
44 "shutdown_command": "echo 'vagrant'|sudo -S shutdown -P now",
45 "ssh_password": "vagrant",
46 "ssh_port": 22,
47 "ssh_username": "vagrant",
48 "ssh_wait_timeout": "10000s",
49 "type": "virtualbox-iso",
50 "vm_name": "{{ user `PACKER_BOX_NAME` }}",
51 "vboxmanage": [
52     ["modifyvm", "{{.Name}}", "--memory", "1024"],
53     ["modifyvm", "{{.Name}}", "--cpus", "2"]
54 ],
55 "virtualbox_version_file": ".vbox_version"
56 },
```

```
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 ],
```



```
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.
- ☒ 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.