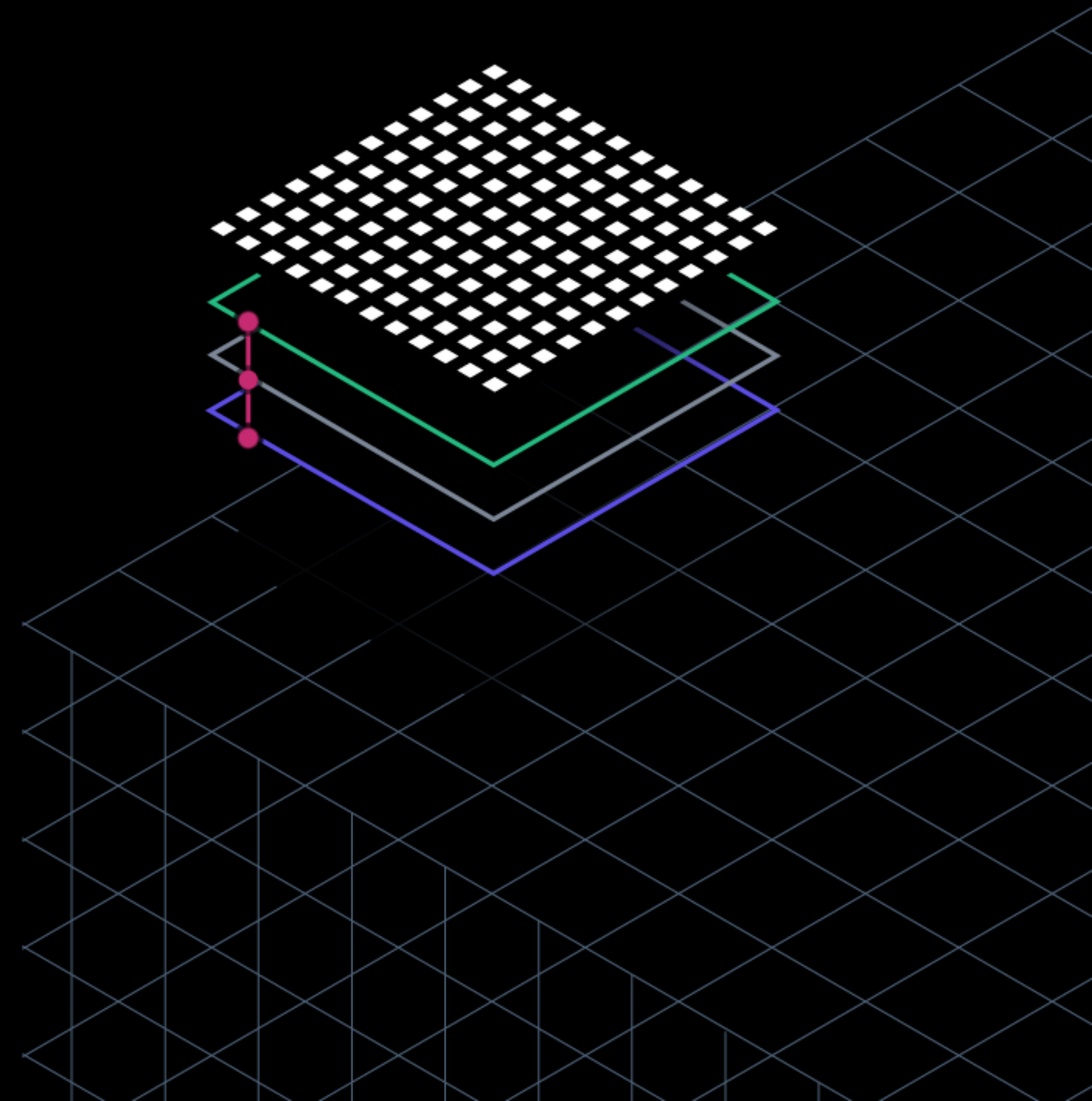
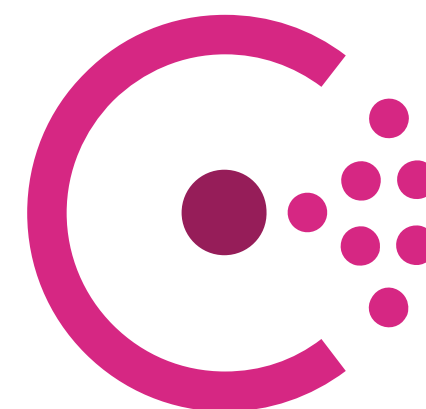
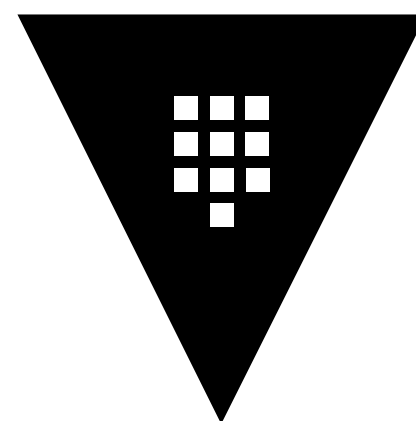




**Any Infrastructure.
Any Application.
Anywhere.**

Any Infrastructure. Any Application.
Anywhere.





HashiCorp Jobs

We are hiring

Slides and code:

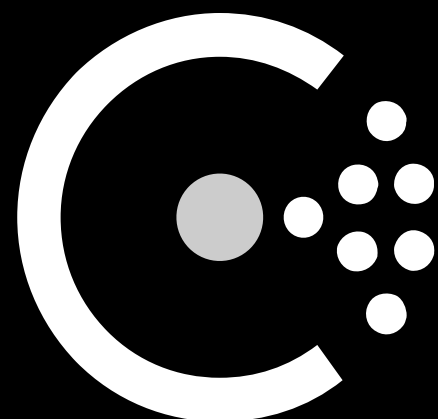
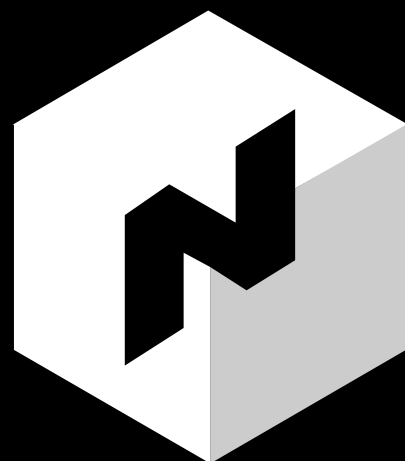
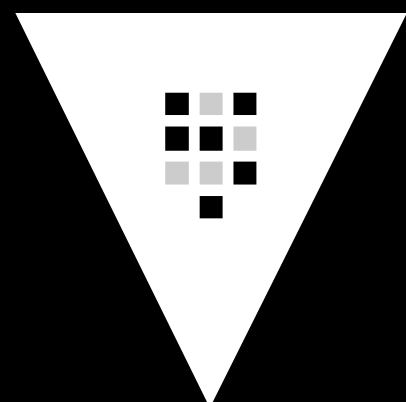
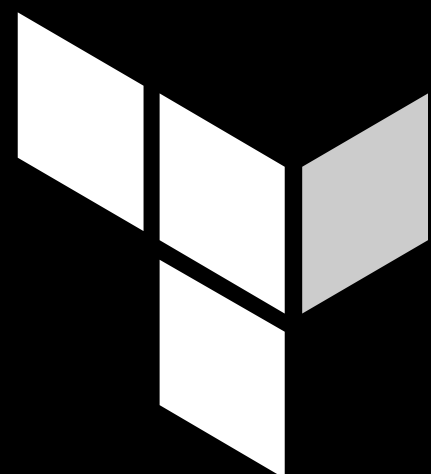
<https://github.com/kikitux/HUG-BE>



Building VMs and docker images with Packer.

Packer 101.

Packer 201.

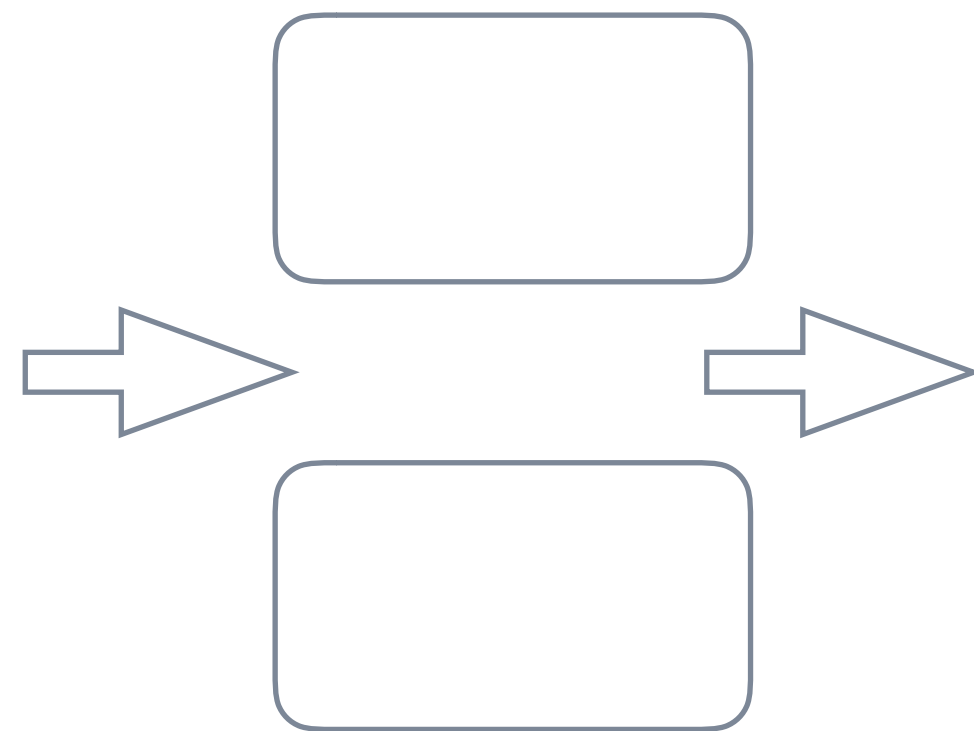


Packer Basics

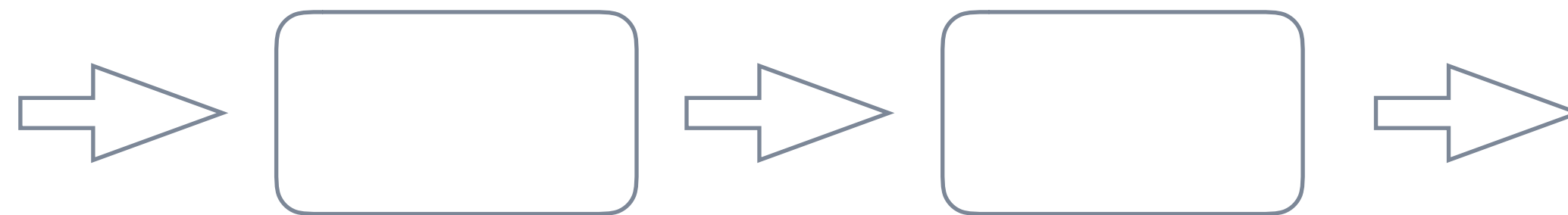
101

Parallel vs Serial

- Parallel



- Serial



JSON Template

```
{
  "variables": {},
  "builders": [
    {
      "type": "required"
    }
  ],
  "post-processors": [
    {},
    {}
  ],
  "provisioners": [
    {},
    {}
  ]
}
```


JSON Template

```
{  
  "variables": # can be ENV or text  
  "builders": [  
    { # these are run in parallel },  
    { # these are run in parallel }  
  ],  
  "post-processors": [  
    { # these are run in parallel },  
    { # these are run in parallel }  
  ],  
  "provisioners": [  
    { # these are run in series 1},  
    { # these are run in series 2}  
  ]  
}
```

JSON Template

```
{  
  "variables": # can be ENV or text  
  "builders": [  
    { # these are run in parallel },  
    { # these are run in parallel }  
  ],  
  "post-processors": [  
    [  
      { # these are run in series 1 },  
      { # these are run in series 2 }  
    ]  
  ],  
  "provisioners": [  
    { # these are run in series 1},  
    { # these are run in series 2}  
  ]  
}
```

Virtualbox-iso

Starts on an ISO 🤪

```
{
  "variables": {
    .. others ..,
    "build_name": "xenial"
  },
  "builders": [
    {
      "type": "virtualbox-iso",
      "name": "{{ user `build_name` }}-vbox",
      "vm_name": "{{ user `build_name` }}-vbox",
      "boot_command": [
        "this goes to the boot"
      ],
      "boot_wait": "5s",
      "disk_size": "{{user `disk_size`}}",
      "guest_os_type": "Ubuntu_64",
      "http_directory": "http",
      "iso_checksum": "{{user `iso_checksum`}}",
      "iso_checksum_type": "{{user `iso_checksum_type`}}",
      "iso_url": "{{user `iso_url`}}"
    }
  ],
  ..
}
```

```
{
  "variables": {
    .. others ..,
    "build_name": "xenial"
  },
  ..
  "provisioners": [
    {
      "script": "scripts/provision.sh",
      "type": "shell"
    }
  ],
  "post-processors": [
    {
      "type": "vagrant",
      "keep_input_artifact": true,
      "output": "{{.BuildName}}.box"
    }
  ]
}
```

Virtualbox-ovf

Starts on an VM 🕶️

```
{
  "variables": {
    .. others ..,
    "build_name": "xenial"
  },
  "builders": [
    {
      "type": "virtualbox-ovf",
      "name": "{{ user `build_name` }}-vbox",
      "vm_name": "{{ user `build_name` }}-vbox",
      "boot_wait": "5s",
      "checksum": "{{user `ova_checksum`}}",
      "checksum_type": "{{user `ova_checksum_type`}}",
      "source_path": "{{user `ova_url`}}"
    }
  ],
  ..
}
```

```
{
  "variables": {
    .. others ..,
    "build_name": "xenial"
  },
  ..
  "provisioners": [
    {
      "script": "scripts/provision.sh",
      "type": "shell"
    }
  ],
  "post-processors": [
    {
      "type": "vagrant",
      "keep_input_artifact": true,
      "output": "{{.BuildName}}.box"
    }
  ]
}
```


Docker

Docker!

```
{
  "variables": {},
  "builders": [
    {
      "commit": true,
      "image": "ubuntu:xenial",
      "pull": true,
      "type": "docker",
    }
  ],
  "post-processors": [
    {
      "repository": "xenial",
      "tag": "latest",
      "type": "docker-tag"
    }
  ],
  "provisioners": [
    {
      "script": "xenial-all.sh",
      "type": "shell"
    }
  ]
}
```

Nice things to know *

201 (* I wish someone told me this before)

JSON Template, filter only

```
{
  "variables": {},
  "builders": [
    {
      "type": "required1"
    },
    {
      "type": "required2"
    }
  ],
  "post-processors": [
    {"only": ["required1"],...},
    {"only": ["required2"],...}
  ],
  "provisioners": [
    {"only": ["required1"],...},
    {"only": ["required2"],...}
  ]
}
```

```
{
  "variables": {},
  "builders": [
    {
      "type": "required1",
      "name": "optional1"
    },
    {
      "type": "required2",
      "name": "optional2"
    }
  ],
  "post-processors": [
    {"only": ["required1"],...},
    {"only": ["required2"],...},
    {"only": ["optional1"],...},
    {"only": ["optional2"],...}
  ],
  "provisioners": [
    {"only": ["required1"],...},
    {"only": ["required2"],...},
    {"only": ["optional1"],...},
    {"only": ["optional2"],...}
  ]
}
```

File Provision

Upload

```
{  
  "type": "file",  
  "source": "/local/path/file.ext",  
  "destination": "/remote/path/file.ext"  
}
```

Download

```
{  
  "type": "file",  
  "source": "/tmp/slug.tar.gz",  
  "destination": "slug.tar.gz",  
  "direction": "download"  
}
```


New Use cases all the time

Question on the mailing list

My question was: should I use one package.json per Dockerfile or should I build both images from the same package.json?

I have seen we can use several builders and post-processors with Packer but of what I understand, this feature mostly targets the use case of building the same image for various platforms (e.g. Docker + Vagrant) at the same time.

For my case here, I would like to build two different images not related to each others and using different provisioning roles in a same playbook.

Usually, a unique Ansible playbook script is used to provision several images according to its inventory (here I would have the web container and db container in the inventory), but I am not sure I can do this with Packer.

Thanks in advance,

Loric

Response and example repo!

<https://github.com/kikitux/packer-multidocker>

Thank you for your precise explanation, and for having posted this code as example on GitHub.

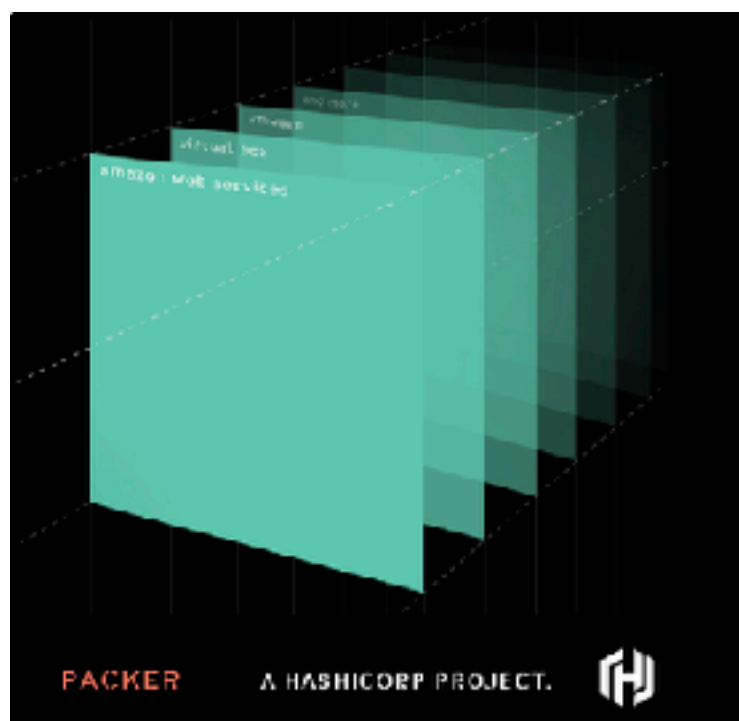
Multi Docker

Docker! Docker! Docker!

```
{
  "variables": {},
  "builders": [
    {
      "commit": true,
      "image": "ubuntu:xenial",
      "pull": true,
      "type": "docker",
      "name": "xenial-vim"
    },
    {
      "commit": true,
      "image": "ubuntu:xenial",
      "pull": true,
      "type": "docker",
      "name": "xenial-git"
    }
  ],
  ..
}
```

```
{
  ..
  "post-processors": [
    {
      "only": [
        "xenial-vim"
      ],
      "repository": "xenial-vim",
      "tag": "latest",
      "type": "docker-tag"
    },
    {
      "only": [
        "xenial-git"
      ],
      "repository": "xenial-git",
      "tag": "latest",
      "type": "docker-tag"
    }
  ],
  ..
}
```

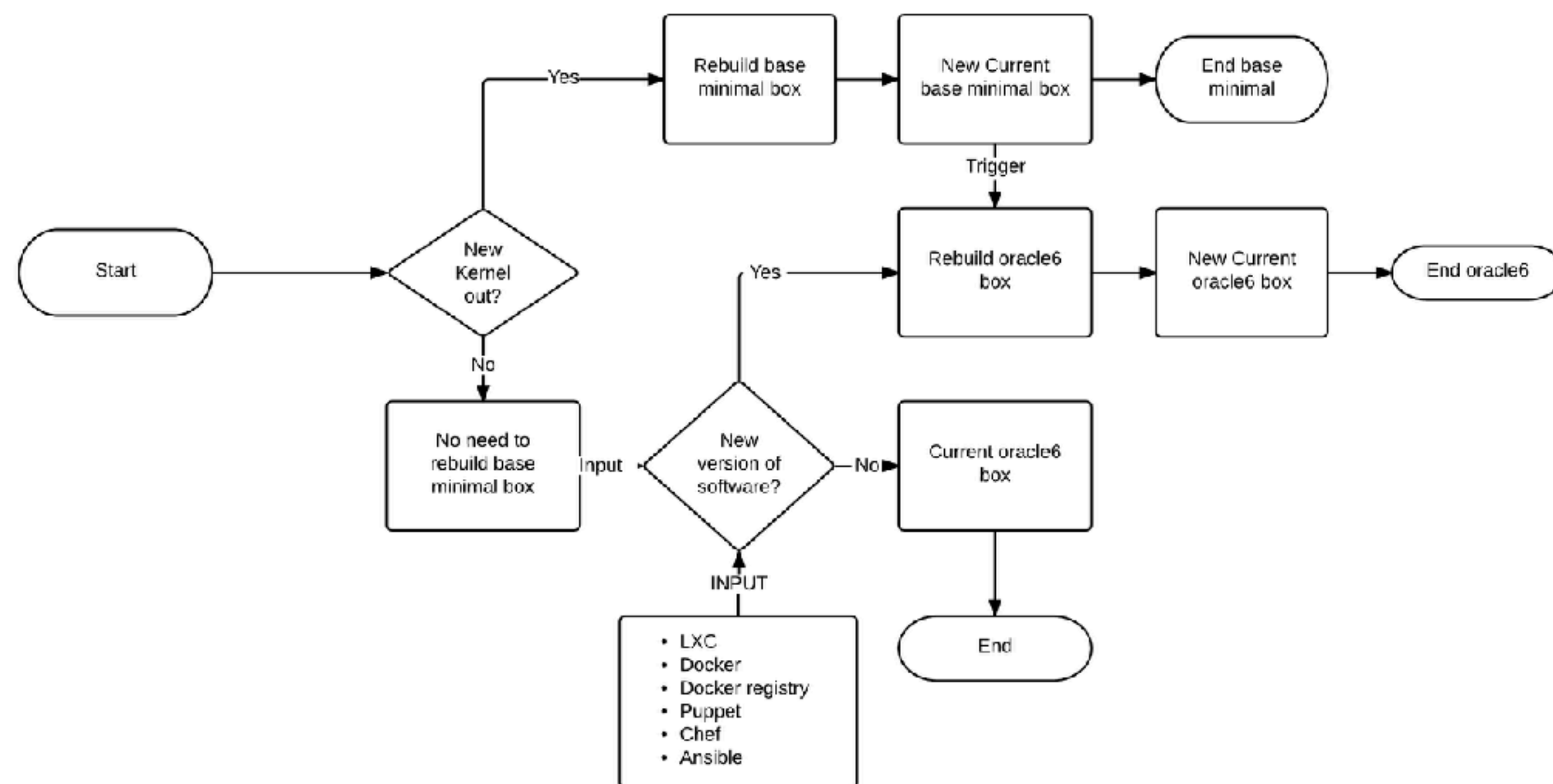
```
{
  ..
  "provisioners": [
    {
      "script": "xenial-all.sh",
      "type": "shell"
    },
    {
      "only": [
        "xenial-vim"
      ],
      "script": "xenial-vim.sh",
      "type": "shell"
    },
    {
      "only": [
        "xenial-git"
      ],
      "script": "xenial-git.sh",
      "type": "shell"
    }
  ],
}
```



Sample Packer Build

Multistate

Virtualbox-iso + Virtualbox-ovf



Thank you.



HashiCorp

www.hashicorp.com hello@hashicorp.com