GITHUB (HTTPS://GITHUB.COM/MITCHELLH/PACKER)

DOWNLOAD (/DOWNLOADS.HTML)

INTRO (/INTRO)

DOCUMENTATION (/DOCS)

COMMUNITY (/COMMUNITY)

DOCS

Installation (/docs/installation.html)

Terminology (/docs/basics/terminology.html)

COMMAND-LINE

Introduction (/docs/command-line/introduction.html)

Build (/docs/command-line/build.html)

Fix (/docs/command-line/fix.html)

Inspect (/docs/command-line/inspect.html)

Push (/docs/command-line/push.html)

Validate (/docs/command-line/validate.html)

Machine-Readable Output (/docs/command-line/machine-readable.html)

TEMPLATES

Introduction (/docs/templates/introduction.html)

Builders (/docs/templates/builders.html)

Provisioners (/docs/templates/provisioners.html)

Post-Processors (/docs/templates/post-processors.html)

Push (/docs/templates/push.html)

Configuration Templates (/docs/templates/configuration-templates.html)

User Variables (/docs/templates/user-variables.html)

Veewee-to-Packer (/docs/templates/veewee-to-packer.html)

BUILDERS

Amazon EC2 (AMI) (/docs/builders/amazon.html)

DigitalOcean (/docs/builders/digitalocean.html)

Docker (/docs/builders/docker.html)

Google Compute Engine (/docs/builders/googlecompute.html)

Null (/docs/builders/null.html)

OpenStack (/docs/builders/openstack.html)

Parallels (/docs/builders/parallels.html)

QEMU (/docs/builders/qemu.html)

VirtualBox (/docs/builders/virtualbox.html)

VMware (/docs/builders/vmware.html)

Custom (/docs/builders/custom.html)

PROVISIONERS

Shell Scripts (/docs/provisioners/shell.html)

File Uploads (/docs/provisioners/file.html)

Ansible (/docs/provisioners/ansible-local.html)

Chef Client (/docs/provisioners/chef-client.html)

Chef Solo (/docs/provisioners/chef-solo.html)

Puppet Masterless (/docs/provisioners/puppet-masterless.html)

Puppet Server (/docs/provisioners/puppet-server.html)

Salt (/docs/provisioners/salt-masterless.html)

Custom (/docs/provisioners/custom.html)

POST-PROCESSORS

Atlas (/docs/post-processors/atlas.html)

compress (/docs/post-processors/compress.html)

docker-import (/docs/post-processors/docker-import.html)

docker-push (/docs/post-processors/docker-push.html)

docker-save (/docs/post-processors/docker-save.html)

docker-tag (/docs/post-processors/docker-tag.html)

Vagrant (/docs/post-processors/vagrant.html)

Vagrant Cloud (/docs/post-processors/vagrant-cloud.html)

vSphere (/docs/post-processors/vsphere.html)

OTHER

Core Configuration (/docs/other/core-configuration.html)

Debugging (/docs/other/debugging.html)

Environmental Variables (/docs/other/environmental-variables.html)

EXTEND PACKER

Packer Plugins (/docs/extend/plugins.html)

Developing Plugins (/docs/extend/developing-plugins.html)

Custom Builder (/docs/extend/builder.html)

Custom Command (/docs/extend/command.html)

Custom Post-Processor (/docs/extend/post-processor.html)

Custom Provisioner (/docs/extend/provisioner.html)

TEMPLATES: BUILDERS

Within the template, the builders section contains an array of all the builders that Packer should use to generate a machine images for the template.

Builders are responsible for creating machines and generating images from them for various platforms. For example, there are separate builders for EC2, VMware, VirtualBox, etc. Packer comes with many builders by default, and can also be extended to add new builders.

This documentation page will cover how to configure a builder in a template. The specific configuration options available for each builder, however, must be referenced from the documentation for that specific builder.

Within a template, a section of builder definitions looks like this:

```
{
  "builders": [
    // ... one or more builder definitions here
  ]
}
```

Builder Definition

A single builder definition maps to exactly one build (/docs/basics/terminology.html#term-build). A builder definition is a JSON object that requires at least a `type` key. The `type` is the name of the builder that will be used to create a machine image for the build.

In addition to the `type`, other keys configure the builder itself. For example, the AWS builder requires an `access_key`, `secret_key`, and some other settings. These are placed directly within the builder definition.

An example builder definition is shown below, in this case configuring the AWS builder:

```
{
  "type": "amazon-ebs",
  "access_key": "...",
  "secret_key": "..."
}
```

Named Builds

Each build in Packer has a name. By default, the name is just the name of the builder being used. In general, this is good enough. Names only serve as an indicator in the output of what is happening. If you want, however, you can specify a custom name using the `name` key within the builder definition.

This is particularly useful if you have multiple builds defined that use the same underlying builder. In this case, you must specify a name for at least one of them since the names must be unique.

PACKER (/)   A HASHICORP (HTTP://WWW.HASHICORP.COM/) PROJECT.
(HTTP://WWW.HASHICORP.COM)