Learn the **HashiCorp Suite** >

DOCS

COMMAND-LINE

TEMPLATES

BUILDERS

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

Azure Resource Manager (/docs/builders/azure-arm.html)

CloudStack (/docs/builders/cloudstack.html)

DigitalOcean (/docs/builders/digitalocean.html)

Docker (/docs/builders/docker.html)

File (/docs/builders/file.html)

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

Hyper-V (/docs/builders/hyperv.html)

Null (/docs/builders/null.html)

1&1 (/docs/builders/oneandone.html)

OpenStack (/docs/builders/openstack.html)

Parallels (/docs/builders/parallels.html)

ProfitBricks (/docs/builders/profitbricks.html)

QEMU (/docs/builders/qemu.html)

Triton (Joyent Public Cloud) (/docs/builders/triton.html)

VirtualBox (/docs/builders/virtualbox.html)

VMware (/docs/builders/vmware.html)

Custom (/docs/builders/custom.html)

PROVISIONERS

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

Local Shell (/docs/provisioners/shell-local.html)

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

PowerShell (/docs/provisioners/powershell.html)

Windows Shell (/docs/provisioners/windows-shell.html)

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

Ansible Remote (/docs/provisioners/ansible.html)

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

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

Converge (/docs/provisioners/converge.html)

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

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

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

Windows Restart (/docs/provisioners/windows-restart.html)

Custom (/docs/provisioners/custom.html)

POST-PROCESSORS

Amazon Import (/docs/post-processors/amazon-import.html)

Artifice (/docs/post-processors/artifice.html)

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

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

Checksum (/docs/post-processors/checksum.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)

Google Compute Export (/docs/post-processors/googlecompute-export.html)

Local Shell (/docs/post-processors/shell-local.html)

Manifest (/docs/post-processors/manifest.html)

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

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

OpenStack Builder

Type: `openstack`

The `openstack` Packer builder is able to create new images for use with OpenStack (http://www.openstack.org). The builder takes a source image, runs any provisioning necessary on the image after launching it, then creates a new reusable image. This reusable image can then be used as the foundation of new servers that are launched within OpenStack. The builder will create temporary keypairs that provide temporary access to the server while the image is being created. This simplifies configuration quite a bit.

The builder does *not* manage images. Once it creates an image, it is up to you to use it or delete it.

> **OpenStack Liberty or later requires OpenSSL!** To use the OpenStack builder with OpenStack Liberty (Oct 2015) or later you need to have OpenSSL installed *if you are using temporary key pairs*, i.e. don't use `ssh_keypair_name` (/docs/builders/openstack.html#ssh_keypair_name) nor `ssh_password` (/docs/templates/communicator.html#ssh_password). All major OS'es have OpenSSL installed by default except Windows.

Configuration Reference

There are many configuration options available for the builder. They are segmented below into two categories: required and optional parameters. Within each category, the available configuration keys are alphabetized.

In addition to the options listed here, a communicator (/docs/templates/communicator.html) can be configured for this builder.

Required:

- `flavor` (string) - The ID, name, or full URL for the desired flavor for the server to be created.

- `image_name` (string) - The name of the resulting image.

- `identity_endpoint` (string) - The URL to the OpenStack Identity service. If not specified, Packer will use the environment variables `OS_AUTH_URL` , if set.

- `source_image` (string) - The ID or full URL to the base image to use. This is the image that will be used to launch a new server and provision it. Unless you specify completely custom SSH settings, the source image must have `cloud-init` installed so that the keypair gets assigned properly.

- `source_image_name` (string) - The name of the base image to use. This is an alternative way of providing `source_image` and only either of them can be specified.

- `username` or `user_id` (string) - The username or id used to connect to the OpenStack service. If not specified, Packer will use the environment variable `OS_USERNAME` or `OS_USERID` , if set.

- `password` (string) - The password used to connect to the OpenStack service. If not specified, Packer will use the environment variables `OS_PASSWORD` , if set.

Optional:

- `availability_zone` (string) - The availability zone to launch the server in. If this isn't specified, the default enforced by your OpenStack cluster will be used. This may be required for some OpenStack clusters.

- `config_drive` (boolean) - Whether or not nova should use ConfigDrive for cloud-init metadata.

- `domain_name` or `domain_id` (string) - The Domain name or ID you are authenticating with. OpenStack installations require this if identity v3 is used. Packer will use the environment variable `OS_DOMAIN_NAME` or `OS_DOMAIN_ID` , if set.

- `endpoint_type` (string) - The endpoint type to use. Can be any of "internal", "internalURL", "admin", "adminURL", "public", and "publicURL". By default this is "public".

- `floating_ip` (string) - A specific floating IP to assign to this instance.

- `floating_ip_pool` (string) - The name of the floating IP pool to use to allocate a floating IP.

- `image_members` (array of strings) - List of members to add to the image after creation. An image member is usually a project (also called the “tenant”) with whom the image is shared.

- `image_visibility` (string) - One of "public", "private", "shared", or "community".

- `insecure` (boolean) - Whether or not the connection to OpenStack can be done over an insecure connection. By default this is false.

- `metadata` (object of key/value strings) - Glance metadata that will be applied to the image.

- `instance_metadata` (object of key/value strings) - Metadata that is applied to the server instance created by Packer. Also called server properties in some documentation. The strings have a max size of 255 bytes each.

- `networks` (array of strings) - A list of networks by UUID to attach to this instance.

- `rackconnect_wait` (boolean) - For rackspace, whether or not to wait for Rackconnect to assign the machine an IP address before connecting via SSH. Defaults to false.

- `region` (string) - The name of the region, such as "DFW", in which to launch the server to create the AMI. If not specified, Packer will use the environment variable `OS_REGION_NAME`, if set.

- `reuse_ips` (boolean) - Whether or not to attempt to reuse existing unassigned floating ips in the project before allocating a new one. Note that it is not possible to safely do this concurrently, so if you are running multiple openstack builds concurrently, or if other processes are assigning and using floating IPs in the same openstack project while packer is running, you should not set this to true. Defaults to false.

- `security_groups` (array of strings) - A list of security groups by name to add to this instance.

- `ssh_interface` (string) - The type of interface to connect via SSH. Values useful for Rackspace are "public" or "private", and the default behavior is to connect via whichever is returned first from the OpenStack API.

- `ssh_ip_version` (string) - The IP version to use for SSH connections, valid values are `4` and `6`. Useful on dual stacked instances where the default behavior is to connect via whichever IP address is returned first from the OpenStack API.

- `ssh_keypair_name` (string) - If specified, this is the key that will be used for SSH with the machine. By default, this is blank, and Packer will generate a temporary keypair. `ssh_private_key_file` (/docs/templates/communicator.html#ssh_private_key_file) must be specified with this.

- `tenant_id` or `tenant_name` (string) - The tenant ID or name to boot the instance into. Some OpenStack installations require this. If not specified, Packer will use the environment variable `OS_TENANT_NAME`, if set. Tenant is also called Project in later versions of OpenStack.

- `use_floating_ip` (boolean) - *Deprecated* use `floating_ip` or `floating_ip_pool` instead.

- `user_data` (string) - User data to apply when launching the instance. Note that you need to be careful about escaping characters due to the templates being JSON. It is often more convenient to use `user_data_file`, instead.

- `user_data_file` (string) - Path to a file that will be used for the user data when launching the instance.

Basic Example: DevStack

Here is a basic example. This is a example to build on DevStack running in a VM.

```
{
  "type": "openstack",
  "identity_endpoint": "http://<destack-ip>:5000/v3",
  "tenant_name": "admin",
  "domain_name": "Default",
  "username": "admin",
  "password": "<your admin password>",
  "region": "RegionOne",
  "ssh_username": "root",
  "image_name": "Test image",
  "source_image": "<image id>",
  "flavor": "m1.tiny",
  "insecure": "true"
}
```

Basic Example: Rackspace Public Cloud