

[Toggle side navigation](#)

# How to build Multipass images with Packer

[Packer](#) is a utility that lets you (re)build images to run in a variety of environments. Multipass can run those images too, provided some requirements are met (namely, the image has to boot on the hypervisor in use, and [cloud-init](#) needs to be available).

## Contents:

- [The basics](#)
- [Customizing the image](#)
- [Explaining the process](#)
  - [cloud-init](#)
  - [Cleanup](#)
- [Next steps](#)

## The basics

The easiest way is to start from an existing [Ubuntu Cloud Image](#), and the base project setup follows (you can click on the filenames to see their contents, `meta-data` is empty on purpose):

```
├─ cloud-data
|   └─ meta-data
|       └─ user-data
└─ template.json
```

1 directory, 3 files

You will need to install QEMU (e.g. `sudo apt install qemu`) and from there you can run:

```
$ packer build template.json
$ multipass launch file://$PWD/output-qemu/packer-qemu --disk 5G
...
Launched: tolerant-hammerhead
$ multipass shell tolerant-hammerhead
...
ubuntu@tolerant-hammerhead ~:~$
```

## Customizing the image

Now the above works for you, delete the test instance with `multipass delete --purge tolerant-hammerhead` and edit the following section in the `template.json` file:

```
{
  "type": "shell",
  "inline": ["echo Your steps go here."]
},
```

Anything you do here will be reflected in the resulting image. You can install packages, configure services, anything you can do on a running system. You'll need `sudo` (passwordless) for anything requiring admin privileges, or you could add this to this provisioner to run the whole script privileged:

```
"execute_command": "sudo sh -c '{{ .Vars }} {{ .Path }}'",
```

To make the image useful for others, you'll need to "generalize" the image so that it then works fine on another machine, likely another network, or even under a different hypervisor. That's what the last entry in the `provisioners` list does in [template.json](#).

## Explaining the process

In short, Packer starts a virtual machine using the downloaded image and runs all the provisioner steps on the running system.

### cloud-init

The Ubuntu cloud images are set up to only allow login via SSH with key-based authentication by default. Packer's [qemu builder](#) only supports password authentication, so we use `cloud-init` to provision a `packer` user with a simple password for use only during the image build process. This is achieved by the following section in the JSON template:

```
"http_directory": "cloud-data",
"qemuargs": [
  ["-smbios", "type=1,serial=ds=nocloud-net;instance-id=packer;seedfrom=http://{{ .HTTPIP }}:{{ .HTTPPort }}"],
]
```

This directs Packer to serve the `cloud-data` directory over HTTP, and `cloud-init` to consume it via the [NoCloud datasource](#). Refer to [cloud-init's documentation](#) to understand what the `user-data` entries mean.

## Cleanup

For Multipass to be able to consume this image, `cloud-init` needs to execute another set of instructions, so it has to be reset. The `packer` user needs to be removed and some other changes need to be reverted in the image (like SSH password authentication). This happens in the last provisioner in the template, which should be self-explanatory.

Your changes are likely to require some cleanup as well, you can [mount qemu images](#) and use the [Meld](#) tool to compare the two trees, to see if all the changes are indeed on purpose.

## Next steps

Go to [Packer's documentation](#) to learn about the QEMU builder, other provisioners and their configuration as well as everything else that might come in handy. Alternatively, you could extend `user-data` with other `cloud-init` directives to provision your image.

Please come to [our discourse space](#) and let us know about your images!

---

*Last updated 3 months ago.*

 [Help improve this document in the forum.](#)

© 2022 Canonical Ltd.

[Legal information](#) • [Manage your tracker settings](#) • [Report a bug on Multipass itself](#) • [Report a bug on this site](#)