

CheatSheet

Vagrant

BOXES:

Boxes are the package format for Vagrant environments. A box can be used by anyone on any platform that Vagrant supports to bring up an identical working environment. The vagrant box utility provides all the functionality for managing boxes.

Adds a box with the given address to Vagrant (download and install):

```
#vagrant box add matyunin/centos7
```

Vagrant command info:

```
#vagrant box add -h
```

Lists all the boxes that are installed into Vagrant:

```
# vagrant box list
```

Tells you whether or not the box you're using in your current Vagrant environment is outdated:

```
# vagrant box outdated
```

Removes a box from Vagrant that matches the given name:

```
# vagrant box remove NAME
```

Repackages the given box and puts it in the current directory so you can redistribute it:

```
# vagrant box repackage NAME PROVIDER
```

Updates the box for the current Vagrant environment:

```
# vagrant box update
```

INIT:

Initializes the current directory to be a Vagrant environment by creating an initial Vagrant file if one doesn't already exist:

```
#vagrant init [box-name] [box-url]
```

Will overwrite any existing Vagrant file:

```
#vagrant init [box-name] [box-url] --force
```

Minimal Vagrant file will be created:

```
#vagrant init [box-name] [box-url] --minimal
```

Will output the Vagrant file to the given file:

```
#vagrant init [box-name] [box-url] --output FILE
```

UP:

Creates and configures guest machines according to your Vagrantfile:

```
#vagrant up [SERVER]
```

STATUS:

Tells you the state of the machines Vagrant is managing:

```
#vagrant status
```

HALT:

Shuts down the running machine Vagrant is managing:

```
# vagrant halt
```

Don't attempt to gracefully shut down the machine:

```
# vagrant halt -force
```

Yannick Moeyersons
<https://github.com/YannickMoeyersons>

RELOAD:

Equivalent of running a halt followed by an up:

#vagrant reload

DESTROY:

Stops the running machine Vagrant is managing and destroys all resources that were created during the machine creation process:

vagrant destroy

Don't ask for confirmation before destroying:

vagrant destroy --force

PROVISION:

Runs any configured provisioners against the running Vagrant managed machine:

#vagrant provision

SSH:

Will SSH into a running Vagrant machine and give you access to a shell:

#vagrant ssh [SERVER]

SSH-CONFIG:

Will output valid configuration for an SSH config file to SSH into the running Vagrant machine from SSH directly:

#vagrant ssh-config --host NAME

SUSPEND:

Suspends the guest machine Vagrant is managing, rather than fully shutting it down or destroying it:

#vagrant suspend [SERVER]

RESUME:

Resumes a Vagrant managed machine that was previously suspended:

#vagrant resume

PACKAGE:

Packages a currently running *Virtual Box* environment into a re-usable box:

#vagrant package --output NAME

GLOBAL-STATUS:

Tells you the state of all active Vagrant environments on the system for the currently logged in user:

vagrant global-status

PLUGIN:

Installs a plugin with the given name or file path:

`#vagrant plugin install <name>...`

Installs a license for a proprietary Vagrant plugin:

`#vagrant plugin license <name> <license-file>`

Lists all installed plugins and their respective installed versions:

`#vagrant plugin list`

Uninstalls the plugin with the given name:

`#vagrant plugin uninstall <name> [<name2> <name3> ...]`

Updates the plugins that are installed within Vagrant:

`#vagrant plugin update [<name>]`

LOGIN:

Used to authenticate with a Vagrant Cloud server:

`#vagrant login`

Will check if you're logged in:

`#vagrant login --check`

Will log you out if you're logged in:

`#vagrant login --logout`

CONNECT:

Compliments the share command by enabling access to shared environments:

`# vagrant connect NAME`

SHARE:

Initializes a Vagrant Share session, allowing you to share your Vagrant environment with anyone in the world:

`#vagrant share`

RDP:

Will start an RDP client for a remote desktop session with the guest:

`#vagrant rdp`

ADVANCED COMMANDS:

View all subcommands, including the non-primary subcommands:

`#vagrant list-commands`

Resources:

- <https://docs.vagrantup.com/v2/cli/index.html>