

Creating the Vagrantfile

`vagrant init` Initialize Vagrant with a Vagrantfile and `./vagrant` directory, using no specified base image. Before you can do `vagrant up`, you'll need to specify a base image in the Vagrantfile.

`vagrant init` Initialize Vagrant with a specific box. To find a box, go shopping. When you find one you like, just replace it's name with `boxpath`. For example, `vagrant init chef/centos-6.5`.

Vagrantfile customizations

```
vagrant.configure("2") do |config|
  config.vm.box = "chef/centos-6.5"
  # guest is the VM; host is your computer end
  config.vm.network "forwarded_port", guest: 80,
  host: 8080
  config.vm.provision :shell, path:
  "my_bash_script.sh"
  # path is relative to your Vagrantfile
end
```

By default `./` on your computer is shared as `/vagrant` on the VM. Letting other people access your VM's

Boxes

`vagrant box list` List the installed boxes

`vagrant box add <name> <box path/HTTP URI>` Add the box for later use

`vagrant box remove <name> virtualbox` delete a box

`vagrant box outdated` Check for updates vagrant box update

Boxes are prebuilt VM images. You never modify your box images

Common Vagrant Commands

`vagrant up` starts vagrant environment (also provisions only on the FIRST vagrant up) Equivalent to pressing the power buttons on your servers.

`vagrant status` outputs status of the vagrant machine

`vagrant halt` stops the vagrant machine

`vagrant reload` restarts vagrant machine, loads new Vagrantfile configuration

`vagrant provision` forces reprovisioning of the vagrant machine

`vagrant ssh` connects to machine via SSH

Common Vagrant Commands (cont)

`vagrant destroy` stops and deletes all traces of the vagrant machine

`vagrant suspend` Suspends a virtual machine (remembers state)

`vagrant resume` Resume a suspended machine (vagrant up works just fine for this as well)

`vagrant reload --provision` Restart the virtual machine and force provisioning

`vagrant provision --debug` Use the debug flag to increase the verbosity of the output

Be sure that you are in the same directory as the Vagrantfile when running these commands!

Tips

`vagrant -v` Get the vagrant version

`vagrant global-status` outputs status of all vagrant machines

`vagrant global-status --prune` same as above, but prunes invalid entries

`vagrant push` Vagrant can be configured to *deploy code!*

`vagrant up --provision | tee provision.log` Runs `vagrant up`, forces provisioning and logs all output to a file

`VAGRANT_LOG=info` Use the environment variable

`vagrant up` `VAGRANT_LOG` to set verbosity