vagrant up

--[no-]destroy-on-error	estroy the newly created machine if a fatal, unexpected error occurs. This will only happen on the first vagrant up. By default this is set.
--[no-]parallel	Bring multiple machines up in parallel if the provider supports it.
--provider x	Bring the machine up with the given provider. By default this is "virtualbox".
--provision	Force the provisioners to run.
--provision-with x,y,z	This will only run the given provisioners. For example, if you have a :shell and :chef_solo provisioner and run vagrant provision --provision-with shell, only the shell provisioner will be run.

This command creates and configures guest machines according to your Vagrantfile.
This is the single most important command in Vagrant, since it is how any Vagrant machine is created. Anyone using Vagrant must use this command on a day-to-day basis.

vagrant halt

-f (--force)	Don't attempt to gracefully shut down the machine. This effectively pulls the power on the guest machine.

This command shuts down the running machine Vagrant is managing.
Vagrant will first attempt to gracefully shut down the machine by running the guest OS shutdown mechanism. If this fails, or if the --force flag is specified, Vagrant will effectively just shut off power to the machine.

vagrant reload

--provision	Force the provisioners to run.
--provision-with x,y,z	This will only run the given provisioners. For example, if you have a :shell and :chef_solo provisioner and run vagrant provision --provision-with shell, only the shell provisioner will be run.

The equivalent of running a halt followed by an up.
This command is usually required for changes made in the Vagrantfile to take effect. After making any modifications to the Vagrantfile, a reload should be called.
The configured provisioners will not run again, by default. You can force the provisioners to re-run by specifying the --provision flag.

vagrant suspend

This suspends the guest machine Vagrant is managing, rather than fully shutting it down or destroying it.
A suspend effectively saves the exact point-in-time state of the machine, so that when you resume it later, it begins running immediately from that point, rather than doing a full boot.
This generally requires extra disk space to store all the contents of the RAM within your guest machine, but the machine no longer consumes the RAM of your host machine or CPU cycles while it is suspended.

vagrant resume

This resumes a Vagrant managed machine that was previously suspended, perhaps with the suspend command.

vagrant status

This will tell you the state of the machines Vagrant is managing.
It is quite easy, especially once you get comfortable with Vagrant, to forget whether your Vagrant machine is running, suspended, not created, etc. This command tells you the state of the underlying guest machine.

vagrant ssh

-c COMMAND (--command COMMAND)	This executes a single SSH command, prints out the stdout and stderr, and exits. stdin will not be functional on this executed command.
-p (--plain)	This does an SSH without authentication, leaving authentication up to the user.

This will SSH into a running Vagrant machine and give you access to a shell.
If a -- (two hyphens) are found on the command line, any arguments after this are passed directly into the ssh executable. This allows you to pass any arbitrary commands to do things such as reverse tunneling down into the ssh program.

vagrant destroy

-f (--force)	Don't ask for confirmation before destroying.

This command stops the running machine Vagrant is managing and destroys all resources that were created during the machine creation process. After running this command, your computer should be left at a clean state, as if you never created the guest machine in the first place.
This command usually asks for confirmation before destroying. This confirmation can be skipped by passing in the -f or --force flag.

Sponsored by **Readability-Score.com**
Measure your website readability!
https://readability-score.com