Welcome to the DevOpsLibrary, episode 4, Salt States. Hopefully if you’re watching this you’re already a little familiar with Salt, if not feel free to watch our SaltStack tutorial first. Today we’re going to start with 3 Ubuntu VMs, one acting as a Salt master, and two salt-minions. One will be named prod minion, and the other dev minion. If you’d like to get started quickly, just download our Vagrant File.

First we’re going to talk about salt states. If you remember from before, we were able to run a command on all minions by doing

**Salt star command dot run followed by the name of the command**

But what if we want to write a script that does multiple things, or ensure several packages are installed?

We just need to create a salt state file. To do so, on the master go to SRV slash salt. If that directory doesn’t exist, create it. Now create a new file named apache.sls. All salt state files have the extension SLS and are typically in **yamil** format.

Now type apache2 colon.

Package colon.

Dash installed.

What this file is saying is that anytime it’s called by Salt, ensure that the package named apache2 is installed.

Now go ahead and save the file.

Now run the following command:

salt star state.sls apache

Salt will now automatically install apache on both of the minions. After it finishes go ahead and run the same command one more time.

This time, Salt is just verifying that apache is still installed, if it isn’t it will reinstall it, but for now you should just see two messages saying that the package is already installed.

Now let’s look into organizing things a bit. In your SRV slash salt folder, create a new directory named apache. Move the apache.sls file into that directory. You can now create as many SLS files as you’d like in that directory. To reference them, use the name of the folder followed by a period, then the name of the SLS file without the extension. So as an example let’s run our Apache module again now that we’ve moved it.

Salt star state.sls apache.apache

Now let’s CD into the apache folder, and rename apache.sls to init.sls.

init.sls is called by Salt if we don’t reference the file directly, so now we can go back to calling salt star apache.

That should be enough to get you going on creating states, but let’s go ahead and create one more real quick, it’ll help us demonstrate high states a little bit later. This time we’re going to create a salt state for managing bashrc.

Begin by creating a second folder under SRV slash salt named bashrc. Now copy the bashrc file from root to this folder.

COPY root slash bashrc

to

SRV salt slash bashrc

Note we left off the period intentionally so that it doesn’t hide the file by default. Now create a new file under SRV slash salt bashrc named init.sls

We’re just going to add a few lines for ensuring that the bashrc file is managed on our minions.

Add root slash dot bashrc colon

On the next line type file dot managed colon, followed by dash user root, dash group root, dash mode 0644, and lastly

dash source salt colon Slash Slash bashrc Slash bashrc

Go ahead and save the file, exit init.sls and open up bashrc.

We’re going to scroll down and enable force color prompt.

Now save the file.

If we ran salt star state.sls bashrc right now, it would ensure that all of our minions would have exactly the same bashrc file, with a color text prompt enabled. Don’t worry about running it right now though, we’re going to talk about Salt’s High State functionality!

Salt by default looks for a file named top.sls under SRV slash salt. Let’s go ahead and create that file now.

Inside this file, start out by typing base colon at the top. Under that let’s add star colon. Everything under this will apply to all minions, and for now, let’s just add dash bashrc.

Now under that section, add in quotes ‘prodminion’, with apache underneath it.

Now save the file, and exit.

Run salt star state.highstate.

What Salt is now going to do is go through your top.sls file, and begin targeting each minion with the salt states that we’ve specified. Both minions are going to have their bashrc file updated if it doesn’t match the one that we’ve specified, and prodminion will also check to ensure that apache is installed.

The power of using highstate is that at any time we can tell all of our servers to configure themselves exactly how we want them to be in top.sls. You can also run highstate on a specific minion, or groups of minions at a time. Another option is to target minions using grains, which we’ll cover in a later episode. For now, we’ll show you one final trick related to states. Hop on one of the minions. Become root and run ‘crontab dash e’.

We’re going to add two lines to the crontab file.

The first one is just AT reboot salt-call state.highstate

The second line is: 00 00 \* \* \* salt-call state.highstate

Go ahead and save and exit. From now on, anytime this server is restarted, a salt state.highstate will be called, and will also be triggered automatically at midnight.

Well, that’s plenty to cover for now--I hope you’re starting to see how easy it is to manage servers with Salt, look for our videos in the future as we’ll be covering Pillars, Grains, Salt-Mines and more. Thank you!

**SneakyPhil** said thanks and asked if we could do a tutorial on Grains, Pillars, Reactor, or the Salt Mine in the future.