Top 10 Tips: Chef on AWS

Chef is a configuration management solution, written in Ruby, that provides you with the ability to automate the configuration of your systems and the applications that sit on top of it. It is a client/server application where clients pull the configuration from the chef server, and all the work to transform the configuration into an instance that serves a function takes place on the instance itself.

A Chef *node* is an Instance that is registered with Chef.

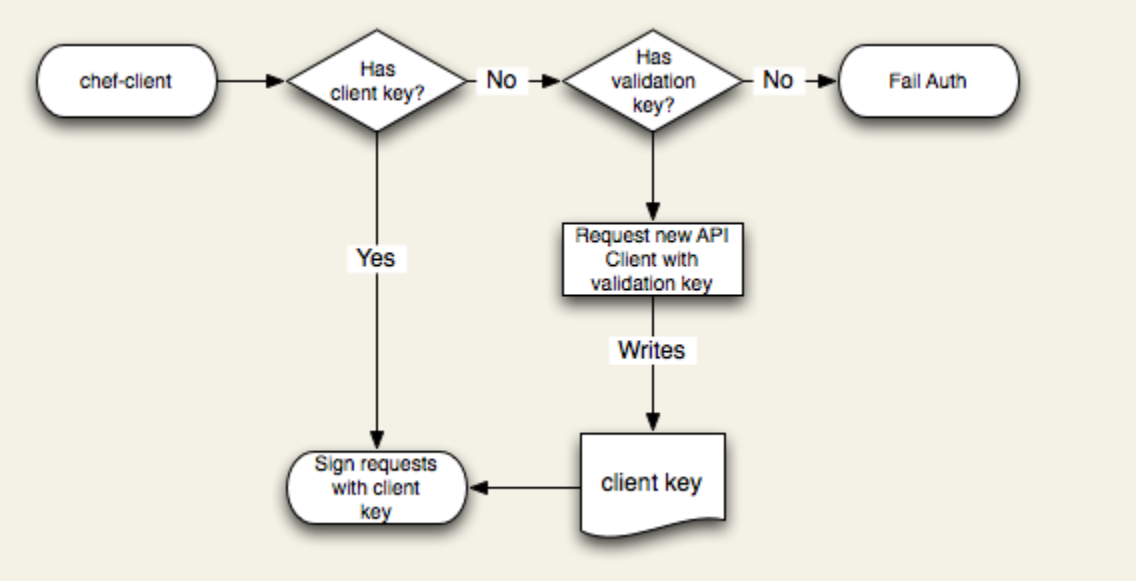
Chef comes in three flavours:

Hosted Chef – This is probably the best way for AWS users to get up to speed quickly with Chef, and it's "Chef as a service". You get up to 5 nodes (registered chef clients) for free, and this approach is great for getting started because you can delete your nodes from the Chef Server without affecting the running instance. It's also the easiest way to run Chef in production because it avoids the need to feed and water your own Chef configuration

Private Chef – A Chef appliance that can be used on-premise with the benefits of the support model that Hosted Chef supplies

Open Source Chef – Build your own Chef server.

All communication between the chef-client and server are via SSH. The diagram below illustrates the registration process of a node



Chef is used in conjunction with CloudFormation to provide an extremely powerful single-click (or command) way to instantiate an AWS stack, and then configure the applications on the server.

Top 10 Chef Tips

Now you have a vague idea what Chef is, what are the Top 10 tips are for an AWS environment?

1. Key management – In particular management of the validation key. The validation key basically grants access to the organisation[i]. This key needs to be protected, so make certain that the key is removed from the node once it has registered with Chef. (The key is not needed after initial registration). It's easy to delete the key as part of the instance's initial boot up sequence.
2. Keep the base AMI as thin as possible, and let Chef do the heavy lifting for you (for example, let Chef install and configure software packages).
3. Windows takes longer to boot up & install the Chef client (known as a "gem"), so we recommend that you pre-install the Chef-client.
4. Use a source control repository to store your chef recipes & cloud formation scripts – treat these scripts/templates like you do code.
5. Chef Recipes should be small & reusable, so keep them simple. Identify the tasks required to complete the configuration and create an individual recipe for each task: e.g. mounting EBS volumes is a distinct task and thus an ideal candidate for a standalone recipe.
6. Use CloudFormation & Chef together. All you need to do is pass the role or recipe as user-data to the instance, with a start-up script baked into the AMI that installs Chef if it's not already pre-installed. *Disclaimer: you can pass start-up commands via CloudFormation; however this is unwieldy thanks to all the quoting required. Passing a single line as user data means the same approach works for both Linux & windows. Furthermore it's immediately obvious what you are trying to do, and easier to understand how it all hangs together.*
7. Check out the community Linux and Windows cookbooks. They are a good resource, a great way to get started. Do not fret if you do not know Ruby you'll soon get the hang of it, and most likely you'll decide to take time out and start playing with Ruby anyway.
8. Use S3 as part of your *Chef with AWS* strategy. For example, keep SSH keys and packages that need to be downloaded from build outputs, etc. in Amazon S3. You can then use Identity and Access Management, signed URLs, and Bucket policies as appropriate to access S3.
9. If you do need to shell out to write a little shell script, use PowerShell, or to do something that is just easier to accomplish via shell-out, make certain that it adheres to the Imdepotency ideal. That is, put in a check as part of the Chef DSL that calls the script to make sure you do not need to have to run the action if it has already run and nothing has changed. This is particularly useful from Windows. (Yes, there is a PowerShell provider.)
10. Chef has a powerful search facility, based on SOLR. Take the time to learn how to use it to provide detailed information on your Chef managed instances.

Note: Puppet and Chef are similar in ideals; albeit with significant differences. The author believes that Chef is more Microsoft Windows™ friendly.

---

[i] Organisation – This is basically what it says. You create an organisation when you sign up for Chef, and the organisation then groups together your cookbooks, nodes, user accounts, etc.