



## **10 tips when using AWS CloudFormation (plus a bonus tip)**

1. Get familiar with the semantics of a template construction. It is JSON but you do need to know how to refer to an ec2 instance for example, how to add it to a security group etc. Obvious but worth stating up front. Use the example templates & documentation to help
2. Use IAM wherever you can to apply least privilege access.
3. Validate your templates at least once a month. AWS takes security seriously so if you are using a base AMI that has any 'critical' security advisories it may be retired. If you need to stick to a particular version of a base AMI then derive a custom AMI from it. You must take responsibility for ensuring that 'critical' security advisories are resolved as well as the expected updating of patches etc which is your responsibility as the customer.
4. Place your CloudFormation template into your version control system. Treat your templates as you would source code.
5. Do not create large monolithic templates create smaller templates that make logic sense to be treated as individual units.
6. If creating a template that uses VPC create the VPC as a separate template then deploy any instances, RDS, ELB's etc into the VPC via separate templates. Remember unless you are using hardware gateway at the customer end or using Elastic IP's (required if using NAT instances) then there is no charge for deploying a VPC.
7. Use parameters copiously this will allow you to change components without having to modify the base template
8. If you are intending to use the template across regions then ensure that you have copies of the template and any embedded templates it references in a bucket in the appropriate region
9. Bootstrap your instances by using the CloudFormation helpers to download files from public repositories, install packages and run commands. This works for both Linux & windows AMI's
10. Use `cfn-validate-template` maybe in conjunction with a more visual syntax checker like `jsonlint` to verify the syntax of the template before using it. This will save time. Reading JSON is not easy with all those curly brackets
11. Regular expressions adhere to the Java regular expression syntax <http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html> escaping `"\"` takes some getting used to so remember to add an additional backslash to any backslash characters in your regular expression, or JSON will interpret these as escape characters.