Management & Governance: CloudFormation

Defining AWS resources with CloudFormation service and our rule of thumb of what services to manage by Cloudformation and what services should be avoided is going to be discussed in this lesson.

We'll cover the following

When using AWS, you almost always want to use some CloudFormation (or a similar tool). It lets you create and update the things you have in AWS without having to click around on the console or write fragile scripts. It takes a while to get the hang of it, but the time savings pay off the initial investment almost immediately. Even for development, the ability to tear down everything cleanly and recreate your AWS set up in one click is extremely valuable

Defining AWS resources with CloudFormation

With CloudFormation;

- You define your AWS resources as a YAML script (or JSON, but we find YAML to be much easier to read and modify).
- Then you point CloudFormation to your AWS account, and it creates all the resources you defined.
- If you run the script again without making any changes, CloudFormation won't do anything (it's idempotent).

- If you make a change to one resource, it will change only that resource, plus any other resources that depend on the modified one (if necessary).
- If you change your mind about an update, you can safely tell CloudFormation to roll it back.
- You can also tell CloudFormation to tear down everything it created, and it will give you your AWS account back in the original state (with a few exceptions).

Q

CloudFormation lets you create and update the things you have in AWS without having to click around on the console or write fragile scripts.

Retake Quiz

Things to avoid in CloudFormation

Things to avoid in CloudFormation

All of that works exceptionally well. However, a trap that people often fall into is to use CloudFormation a little bit too much! There are some things you will likely want to keep out of there. The problems arise when you modify something manually that should be under CloudFormation’s control because when you do that, you can expect unpredictable behavior. Sometimes it’s okay. Sometimes it’s an unrecoverable outcome with catastrophic consequences.

When you’ve touched something manually, and you run your CloudFormation script again, it will often try to revert your changes back to how they were. Sometimes it will manage to do so, but you wouldn’t have wanted it to. Sometimes it will try to reconcile but become stuck in an endless loop.

Rule of thumb

Our rule of thumb is to let CloudFormation deal with all the AWS things that are either static or change very rarely. The table below lists down the things that should be managed by CloudFormation and the things that are better managed elsewhere. You may want to handle some of these things directly from your application, or you could have another simple script that sets them up separately.

↓ OUR RECOMMENDATIONS ↓

Managed by CloudFormation	Managed Elsewhere
VPC configurations	DynamoDB tables
Security groups	Kinesis streams
Load balancers	Autoscale settings
Deployment pipelines	S3 buckets
IAM roles	

Then there are some things that are so infrequently touched and so hard to automate that it just doesn’t make sense to script them. For example:

- Route 53 domain registrations and hosted zones
- Certificate creation and validation from the Certificate Manager.
- and so on.

> The test for whether your infrastructure-as-code is good enough is whether you feel confident that you can tear down your stack and bring it all up again in a few minutes without any mistakes. Spending an unbounded amount of time in pursuit of scripting everything is not advisable.

In the next lesson, we will take a look at SQS and it’s different attributes.