# Working on a Team

In this lesson, you will learn about the different ways in which people can use AWS Lambda while working on the same team.

Modern cloud-based applications usually rely on a whole host of platform services, requiring remote resources for reliable integration testing. When a team of developers works on the same application, it's necessary to somehow isolate remote resources, especially for testing purposes, so people can avoid overwriting each other's work.

## Isolating resources #

There are several possible ways of isolating resources with AWS SAM:

- Create multiple stacks in a single account.
- Use different virtual private clouds.
- Use different AWS accounts.

### Using multiple stacks #

CloudFormation brings up an entire application using a single command, including remote resources and all the required configurations. To create an individual copy of the application, just change the name of the stack in the `sam deploy` command, and SAM will bring up a completely new instance of everything instead of updating an old environment. Using separate stacks for development, testing, and production is a very common way to organise resources for a small team. This approach also makes it easy for developers to set up their own copies of the application for testing, in effect creating a CloudFormation stack for each developer.

## Use different virtual private clouds

Small teams usually keep everything under the same AWS account, sometimes even using the same access keys for development and deployment. That is easy and convenient but doesn't really prevent people from messing up. AWS supports complex authentication and authorisation policies, so larger organisations often want to isolate developer resources directly on AWS. Many organisations want to isolate production environment access, so that developer keys can't be used to arbitrarily poke around the deployed application for end-users. The usual reasons for this are data security and auditability. Even if you do not work in a regulated environment requiring strong data protection, it's good practice to isolate production keys to prevent accidental errors.

For container-based applications, organisations often set up different access keys for each environment or even each developer, and then apply access policies to those keys so people can update only their own resources. This is relatively difficult to do with CloudFormation and SAM. Because CloudFormation generates resource names automatically, it's not easy to set up name-based access policies for resources. Using SAM will ensure that each stack you deploy is completely isolated, but it will not prevent someone from unintentionally deleting the wrong stack.

## Using multiple AWS accounts

A much more convenient solution for isolation with CloudFormation and SAM is to set up separate AWS account deployments. Each AWS account has completely separate resources, without even the need to apply different IAM policies. Until relatively recently it was difficult to manage multiple accounts, but you can now easily create sub-accounts from the AWS Organisations Console. To deploy a stack using a specific account, create a profile using `aws configure` then just add the profile name after `--profile` to `sam package` and `sam deploy`. The `sam build` works locally, so you do not have to run it separately for different profiles.

In the next lesson, you will have a look at how to set up deployment pipelines. Stay tuned!