

# Deploying a SAM Application

Get ready to deploy the SAM application in this lesson!

## WE'LL COVER THE FOLLOWING



- Step 3: Deploy
  - SAM or CloudFormation deployment

## Step 3: Deploy #

You now have a single CloudFormation template ( `output.yaml` ) which describes the entire infrastructure and links to a packaged version of the function code. To create a running instance of your application, you'll need to create a new CloudFormation stack based on this template. To do that, run the following command in the directory that contains the packaged template ( `output.yaml` ):

```
sam deploy --template-file output.yaml --stack-name sam-test-1 --capabilities CAPABILITY_IAM
```

This command has three arguments:

- `--template-file` is the template to deploy. You want to use the result of the `package` command here, hence `output.yaml`.
- `--stack-name` is a name for the target stack. If a stack with the specified name does not yet exist in your AWS account, CloudFormation will create it. If you use the same name the next time you deploy, CloudFormation will update the existing resources instead of creating new ones.
- `--capabilities CAPABILITY_IAM` allows CloudFormation to create IAM policies. SAM needs this to let the API Gateway call Lambda functions. Note that these are not security capabilities of the Lambda function, but instead the privileges required by the deployment process.

**Note that you will use `sam-test-1` as the stack name to run all the commands in this course.**

Environment Variables		^
Key:	Value:	
LANG	C.UTF-8	
LC_ALL	C.UTF-8	
AWS_ACCESS_KEY_ID	Not Specified...	
AWS_SECRET_ACCE...	Not Specified...	
BUCKET_NAME	Not Specified...	
AWS_REGION	Not Specified...	
Terminal		↺ ^

In a few moments, you will see that SAM has started deploying your project:

```
$ sam deploy --template-file output.yaml --stack-name sam-test-1 --capabilities CAPABILITY_IAM

Waiting for changeset to be created..
Waiting for stack create/update to complete
Successfully created/updated stack - sam-test-1
```

If you see permission errors instead of a successful result, make sure that you have configured access credentials for the command line tools correctly, and that your AWS access user has permissions to modify AWS resources. You can find instructions on configuring access credentials in [Chapter 2](#).

## SAM or CloudFormation deployment

The `sam deploy` command is more than just an alias for `aws cloudformation deploy`. It can store build parameters for your project in a temporary directory for easier reuse, and even run a guided deployment where it will interactively prompt you for the necessary inputs. You can run `sam deploy --guided` for an interactive guided mode.

That's it for deploying a SAM application. In the next lesson, you'll inspect the stack you created.