

# Packaging SAM Applications

Let's look at the packaging step that is required to deploy SAM applications after the project has been built.

## WE'LL COVER THE FOLLOWING



- Step 2: Package
  - Creating a bucket
  - Deploying to a specific AWS region
  - `sam package` command
    - SAM or CloudFormation packaging

## Step 2: Package #

### Creating a bucket #

The next step is to bundle all the files required by each function into separate ZIP archives and upload the results to S3. In order to do that, you will first need an S3 bucket to host your function packages. In continuous delivery jargon, this will be your binary artefact storage. Think of a nice bucket name and then create a new S3 bucket using the following command line (replace `BUCKET-NAME` with your chosen name):

```
aws s3 mb s3://BUCKET-NAME
```

For example, to create a bucket called `sam-project-deployment`, run the following command:

```
aws s3 mb s3://sam-project-deployment
```

S3 was one of the first services AWS launched, way back in March 2006, when they weren't yet thinking too much about conquering the world. As a result, all bucket names are unique globally, across all accounts. You will not be able to use `sam-project-deployment` because that bucket has already been claimed

by me. Think of a nice name for your deployment bucket and remember to use it instead of my bucket name when deploying your examples.

You can safely reuse the same binary artefact storage for all your projects, so you do not need to create a separate bucket for each deployment.

## Deploying to a specific AWS region

AWS command-line tools, including SAM CLI, allow users to select a specific data centre with the `--region` option. If you want to create resources in a specific region, add `--region` followed by the AWS data centre identifier to the command line. For example, to create a bucket in London, use the following command:

```
aws s3 mb s3://bucket --region eu-west-2
```

If you use a specific region for deployment, remember to add the `--region` argument to all the commands starting with `aws` or `sam` in this course.

You may create your bucket in the terminal provided below:

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

Once the bucket name is created, please save it as an environment variable next to the `BUCKET_NAME` field in the next terminal.

## sam package command #

SAM has a convenient shortcut to zip up and upload function packages to S3. Just run the following command in the directory containing the CloudFormation template ( `template.yaml` ) and remember to use your bucket name:

```
sam package --s3-bucket $BUCKET_NAME --output-template-file output.yaml
```

This command will produce another CloudFormation template, saving it to the file called `output.yaml` , as requested in the command parameter `--output-template-file` .

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		↺ ^

Look at the file contents by using `cat output.yaml` command, and you will see that it is almost identical to the original `template.yaml` . The difference will be in the `CodeUri` property of the Lambda function, which points to a remote location on S3 in the output template. That's where SAM uploaded your Lambda function code. The function definition should look similar to the following block:

```
Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: s3://sam-project-deployment/ecf4c7e8862642e4942f766736df99f0cb6b9
```

When you run a `sam package` , it will print out a deployment command. This

command is not correct for the first deployment, because it is missing the

security permissions required to set up IAM roles. Do not copy what the `sam package` printed, but instead execute the command from the next section.

### SAM or CloudFormation packaging

CloudFormation also has a `package` command, and in fact, the `sam package` is just a wrapper around it. The benefit of using the `sam` command is that you do not have to specify an input template file. This is especially useful if you want to use `sam build` for installing dependencies because SAM then uses the built copy instead of the original source. If you are not using SAM to build projects, there is no big difference between the two packaging processes.

The bucket you just created is used by the SAM packaging (actually the underlying CloudFormation packaging) to store Lambda packages while deploying. As you experiment with the examples in this course, you will create new files in that bucket with each packaging. Although S3 is a very cheap storage service, you only need these files temporarily, so you can remove them periodically. To do so, use the following command:

```
aws s3 rm s3://BUCKET_NAME --recursive
```

For example, if your intermediary bucket is called `sam-project-deployment`, the full command would be:

```
aws s3 rm s3://sam-project-deployment --recursive
```

Now it's time for deployment. You'll deploy the application in the next lesson.