

Deployment Strategies for Serverless

Terminology

To understand the deployment strategies for serverless applications, we will first cover the terminology of versions, aliases, and traffic shifting. Each AWS Lambda function can have any number of versions and aliases associated with them.

Versions are snapshots of a function that includes the code and configuration, and it is a good practice to publish a new version each time you update your function code. When you invoke a specific version (using the function name and version number combination) you will get the same code and configuration regardless of the state of the function. This protects you against accidentally updating production code. To use versions, you should create an alias, which is a pointer to a version.

Aliases have a name and an Amazon Resource Number (ARN) similar to the function and are accepted by the Invoke APIs. If you invoke an alias, Lambda will in turn invoke the version that the alias is pointing to. In production, you would first update your function code, publish a new version, and invoke the version directly to run tests against it. After you are satisfied, you would change the alias to point to the new version.

Traffic shifting can shift incoming traffic between two versions of a Lambda function based on pre-assigned weights. You can use this feature to gradually shift traffic between two versions, helping you reduce the risk of new Lambda deployments. You can also change your Lambda function's code without affecting other upstream dependencies that rely on the alias.

Deploying serverless applications

If you use AWS SAM to create your serverless application, it comes built-in with AWS [CodeDeploy](#) to provide gradual Lambda deployments. With a few lines of configuration, AWS SAM does the following for you:

- Deploys new versions of your Lambda function, and automatically creates aliases that point to the new version.
- Gradually shifts customer traffic to the new version until you're satisfied that it's working as expected, or you roll back the update.
- Defines pre-traffic and post-traffic test functions to verify that the newly deployed code is configured correctly and your application operates as expected.
- Rolls back the deployment if Amazon CloudWatch alarms are generated.

Deployment options

Now that you know about AWS SAM, you can learn about your deployment options. The following list describes other traffic-shifting options that are available:

- **Canary:** Traffic is shifted in two increments. You can choose from predefined canary options. The options specify the percentage of traffic that's shifted to your updated Lambda function version in the first increment, and the interval, in minutes, before the remaining traffic is shifted in the second increment.
- **Linear:** Traffic is shifted in equal increments with an equal number of minutes between each increment. You can choose from predefined linear options that specify the percentage of traffic that's shifted in each increment and the number of minutes between each increment.
- **All-at-once:** All traffic is shifted from the original Lambda function to the updated Lambda function version at once.

An all-at-once deployment will shift traffic instantly from one version to another, while canary and linear, are much safer, more gradual deployment options.