

AWS Step Functions Terminology.

AWS Step Functions is a reliable service to coordinate distributed components and analyze the flow of your distributed workflow.

Step Functions is based on the concepts of tasks and state machines. You define state machines using the JSON-based Amazon States Language.

Step Functions provides a graphical console to arrange and visualize the components of your application as a series of steps. This makes it simple to build and run multi-step applications. Step Functions automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected.

Read the AWS Step Functions Documentation here: <https://aws.amazon.com/step-functions/getting-started/>^[↗]

State Types

States can perform a variety of functions in your state machine:

Do some work in your state machine (a Task state)

Make a choice between branches of execution (a Choice state)

Stop an execution with a failure or success (a Fail or Succeed state)

Simply pass its input to its output or inject some fixed data (a Pass state)

Provide a delay for a certain amount of time or until a specified time/date (a Wait state)

Begin parallel branches of execution (a Parallel state)

Dynamically iterate steps (a Map state)

Any state type other than the Fail type have the full control over the input and the output. You can control those using the "InputPath", "ResultPath" and "OutputPath". **A path is a string beginning with \$ that you can use to identify components within JSON text.** Using the "InputPath", you can determine which portion of the data sent as an input to the state to send into the processing of that state. For example, that could be a Lambda function. Then, you can insert the result from that Lambda function in a node inside of the input. This is useful when you want to be able to keep the data from the input as well as the result of Lambda function without having to do it within the code of the Lambda. Thus, keeping your flow separate from the Lambda microservice. Finally, you can apply another filter from that combination of data by using an "OutputPath" to decide which node you want to keep. You can find an example of how these three works together here:

<https://docs.aws.amazon.com/step-functions/latest/dg/input-output-example.html>^[↗]

Sample Projects

There are many sample projects provided through the AWS Documentation which you can find here:

<https://docs.aws.amazon.com/step-functions/latest/dg/create-sample-projects.html>