



AWS Step Function

AWS Step Functions is a fully managed service that makes it easy to coordinate the components of distributed applications and microservices using visual workflows. Here's an overview of its key features:

Key Features:

1. **Visual Workflow:** Step Functions allows you to create workflows visually using a state machine diagram, making it easier to understand and manage the execution flow.
2. **State Management:** It manages the state of each step in the workflow, ensuring that the process is resilient and can handle retries, errors, and parallel execution.
3. **Integration with AWS Services:** Step Functions integrates seamlessly with other AWS services like Lambda, EC2, S3, DynamoDB, and more, allowing you to orchestrate complex processes without managing the underlying infrastructure.
4. **Error Handling and Retry:** Step Functions provides built-in error handling and retry mechanisms, allowing you to define how your application should respond to failures.
5. **Serverless Orchestration:** It's serverless, so you don't need to worry about provisioning or managing servers. You only pay for the time your workflows are running.
6. **Execution History:** You can track the execution of each workflow step with detailed logs and metrics, making it easier to debug and optimize your applications.

Use Cases:

- **Data Processing Pipelines:** Automate and coordinate the execution of data processing tasks.
- **Microservices Orchestration:** Manage the flow of microservices in an application.
- **Machine Learning:** Orchestrate and automate the steps involved in training and deploying machine learning models.
- **Long-running Processes:** Manage long-running processes like human approval workflows.

Example:

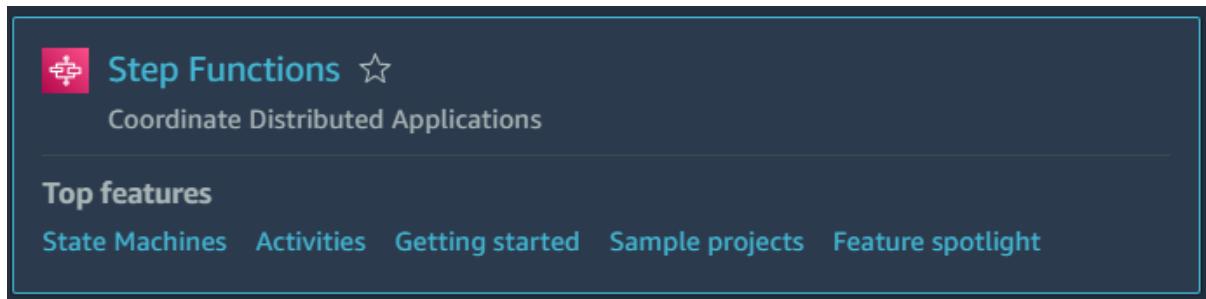
Imagine you need to process and analyze data uploaded to S3. With Step Functions, you can create a workflow that triggers a Lambda function to process the data, then another function to analyze it, and finally store the results in DynamoDB.

This service is particularly useful for creating complex applications that require multiple steps or services to work together in a coordinated way.



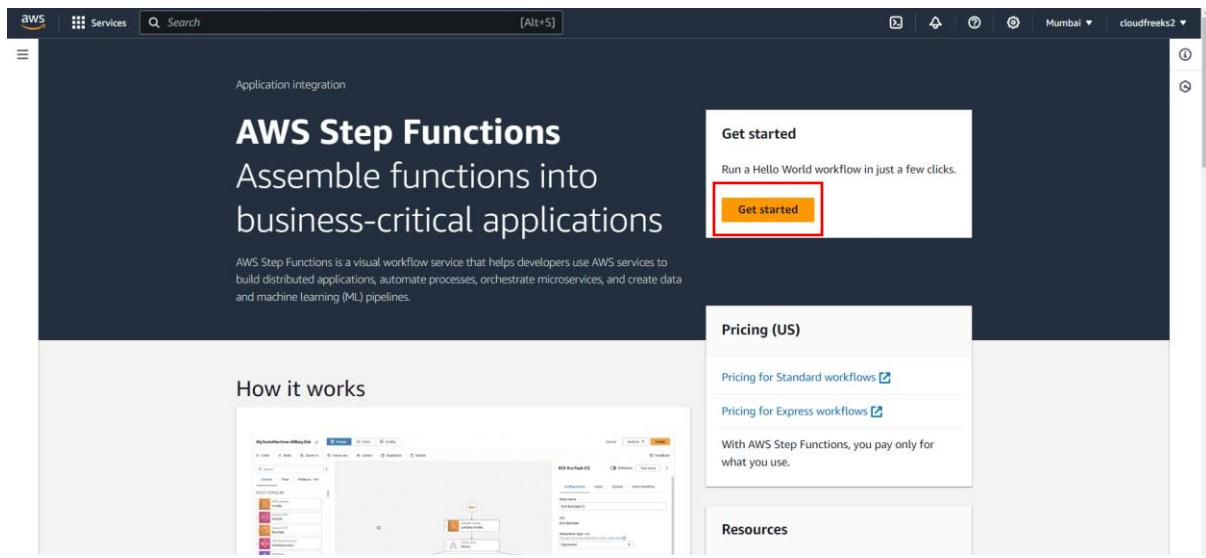
To begin with the Lab:

1. Login to AWS Console and navigate to Step Functions, choose the service accordingly.



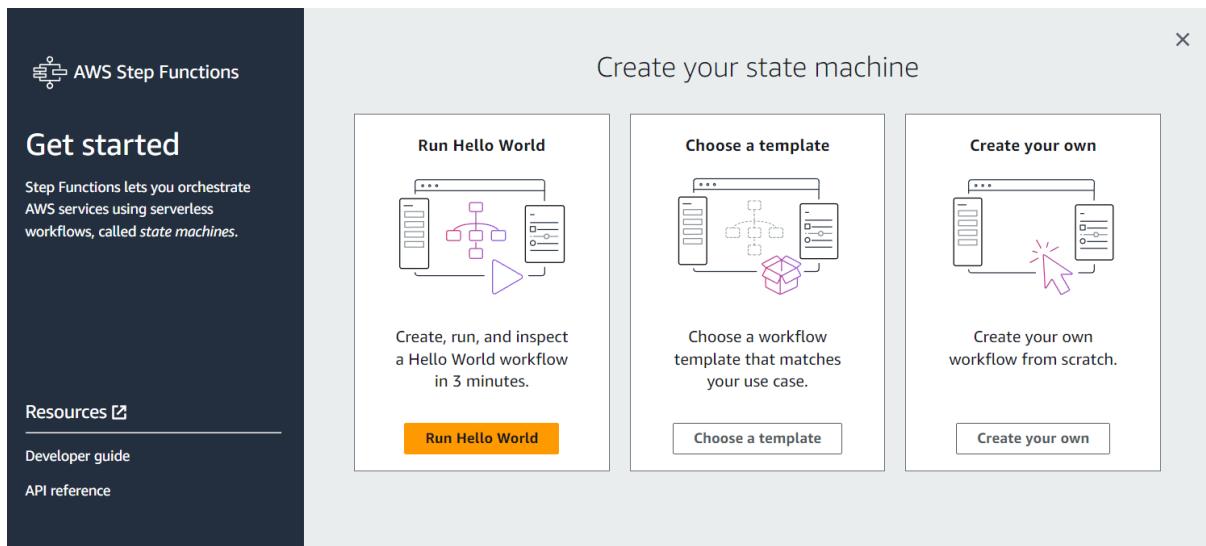
The screenshot shows the AWS Step Functions homepage. At the top, there's a logo and the text "Step Functions ☆ Coordinate Distributed Applications". Below that is a "Top features" section with links to "State Machines", "Activities", "Getting started", "Sample projects", and "Feature spotlight". The main content area has a heading "AWS Step Functions Assemble functions into business-critical applications" and a sub-section "How it works" with a screenshot of the Step Functions console. To the right, there are boxes for "Get started", "Pricing (US)", and "Resources".

2. From the dashboard of Step Functions choose Get started.



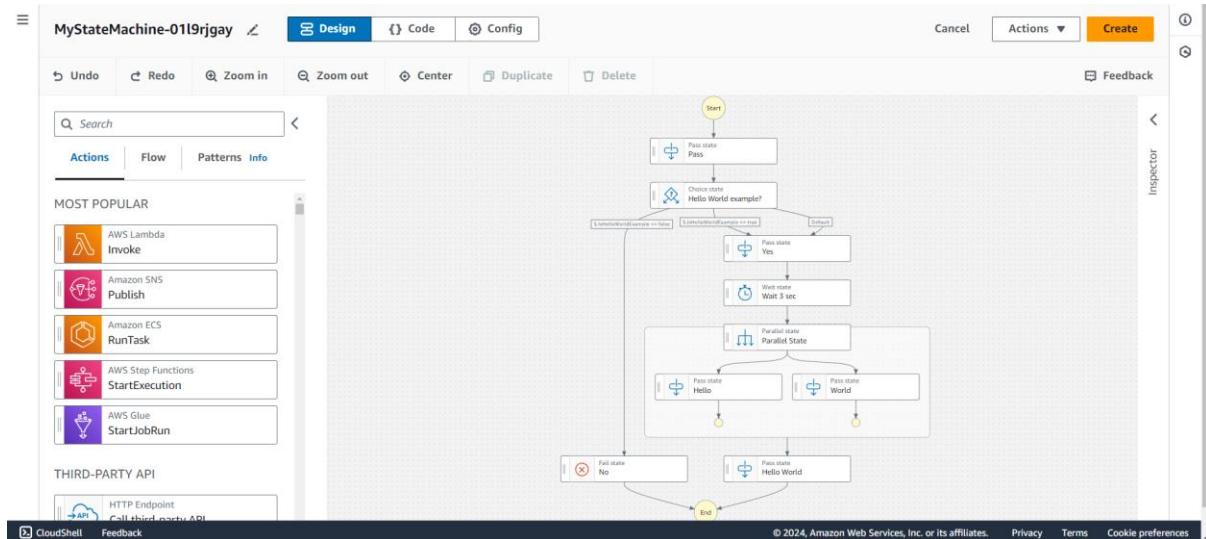
The screenshot shows the "Get started" page of the AWS Step Functions service. It features a large "Get started" button with a red border. Below it, there's a brief description: "Run a Hello World workflow in just a few clicks." To the left, there's a "How it works" section with a screenshot of the Step Functions console. To the right, there are sections for "Pricing (US)" and "Resources".

3. Then you have to click on Run Hello World.



The screenshot shows the "Run Hello World" page of the AWS Step Functions service. It features a large "Run Hello World" button with a red border. Below it, there's a brief description: "Create, run, and inspect a Hello World workflow in 3 minutes." To the left, there's a "Get started" section with a brief description: "Step Functions lets you orchestrate AWS services using serverless workflows, called state machines." To the right, there are sections for "Choose a template" and "Create your own".

4. Now you can visually see how your Hello World example will work. Then click on Create.



5. Then it will ask you to start the execution, so click on start.

Start execution

Name
873d0603-8fd2-49bd-bafb-e7d774bf27c8
Must be 1-80 characters. Can use alphanumeric characters, dashes, or underscores.

Input - optional
Enter input values for this execution in JSON format

Format JSON Export Import

```
1 {
  "Comment": "Insert your JSON here"
}
```

Start execution with latest revision

Open in a new browser tab

Cancel **Start execution**

6. You will see that the execution has been started successfully and in some time the execution status will be changed to succeed.

Execution started successfully

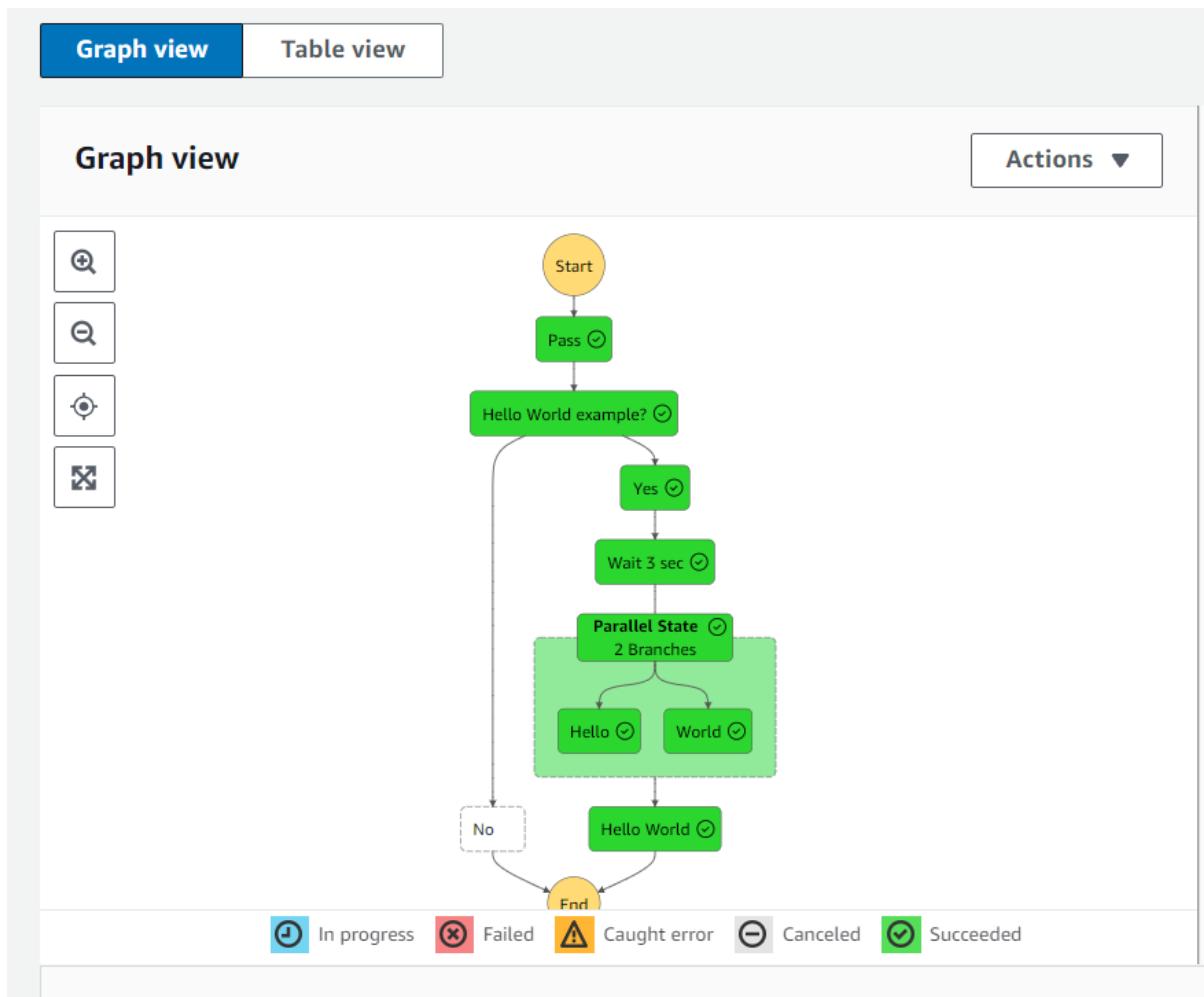
Step Functions > State machines > MyStateMachine-01l9rjgay > Execution: 873d0603-8fd2-49bd-bafb-e7d774bf27c8

Execution: 873d0603-8fd2-49bd-bafb-e7d774bf27c8

Edit state machine New execution Actions

Details		Execution input and output	Definition
Execution status <input checked="" type="radio"/> In progress	Start time Aug 23, 2024, 16:28:30.866 (UTC+05:30)		
Execution type Standard	Last event time Aug 23, 2024, 16:28:30.905 (UTC+05:30)		
Execution ARN arn:aws:states:ap-south-1:878893308172:execution:MyStateMachine-01l9rjgay:873d0603-8fd2-49bd-bafb-e7d774bf27c8	Duration 00:00:00.442		
IAM role ARN arn:aws:iam::878893308172:role/service-role/StepFunctions-MyStateMachine-01l9rjgay-role-3xi735rck	Alias -		
State transitions Learn more	Version -		
5			

7. Now scroll down and you will see the graphical representation of it.



8. Also, if you scroll down a little more then you will see that 20 total events happened. Now if you expand them then you can see the input and output parameters in more detail.

Events (20)

Filter by properties or search by keyword Filter by a date and time range

ID	Type	Step	Resource	Started After	Timestamp
▶ 9	⊖ WaitStateExited	Wait 3 sec		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 10	⊕ ParallelStateEntered	Parallel State		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 11	⊕ ParallelStateStarted	Parallel State		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 12	⊕ PassStateEntered	Hello		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 13	⊖ PassStateExited	Hello		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 14	⊕ PassStateEntered	World		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 15	⊖ PassStateExited	World		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 16	⊕ ParallelStateSucceeded	Parallel State		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 17	⊖ ParallelStateExited	Parallel State		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 18	⊕ PassStateEntered	Hello World		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 19	⊖ PassStateExited	Hello World		00:00:03.098	Aug 23, 2024, 16:28:33.964 (UTC+05:30)
▶ 20	⊕ ExecutionSucceeded			00:00:03.140	Aug 23, 2024, 16:28:34.006 (UTC+05:30)

9. Now if you come back to your step function then you will see that there is only once execution.

The screenshot shows the AWS Step Functions console with the following details:

Step Functions > State machines > State machine: MyStateMachine-01l9rjgay

MyStateMachine-01l9rjgay

Details

Arn	Type
arn:aws:states:ap-south-1:878893308172:stateMachine:MyStateMachine-01l9rjgay	Standard
IAM role ARN	Status
arn:aws:iam::878893308172:role/service-role/StepFunctions-MyStateMachine-01l9rjgay-role-3xi735rck	Active
Creation date	
Aug 23, 2024, 16:26:40 (UTC+05:30)	
X-Ray tracing	
Disabled	

Executions | Monitoring | Logging | Definition | Aliases | Versions | Tags

Executions (1)

Name	Status	Start Time (local)	End Time (local)	Duration	Version	Alias
873d0603-8fd2-49bd-bafb-e7d774bf27c8	Succeeded	Aug 23, 2024, 16:28:30	Aug 23, 2024, 16:28:34	00:00:03.140	-	-

Actions: [Edit](#) | [Actions ▾](#) | [Start execution](#)