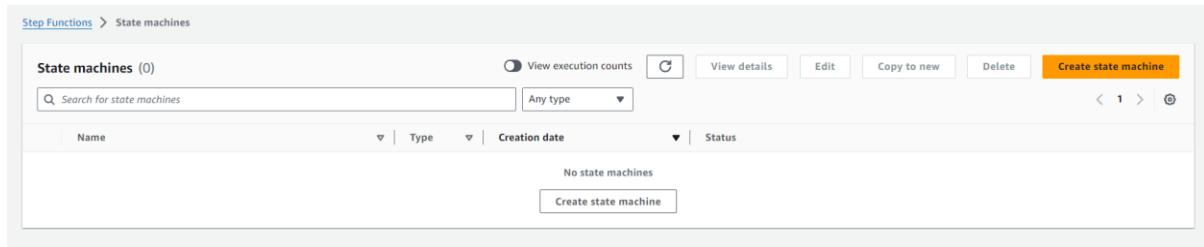


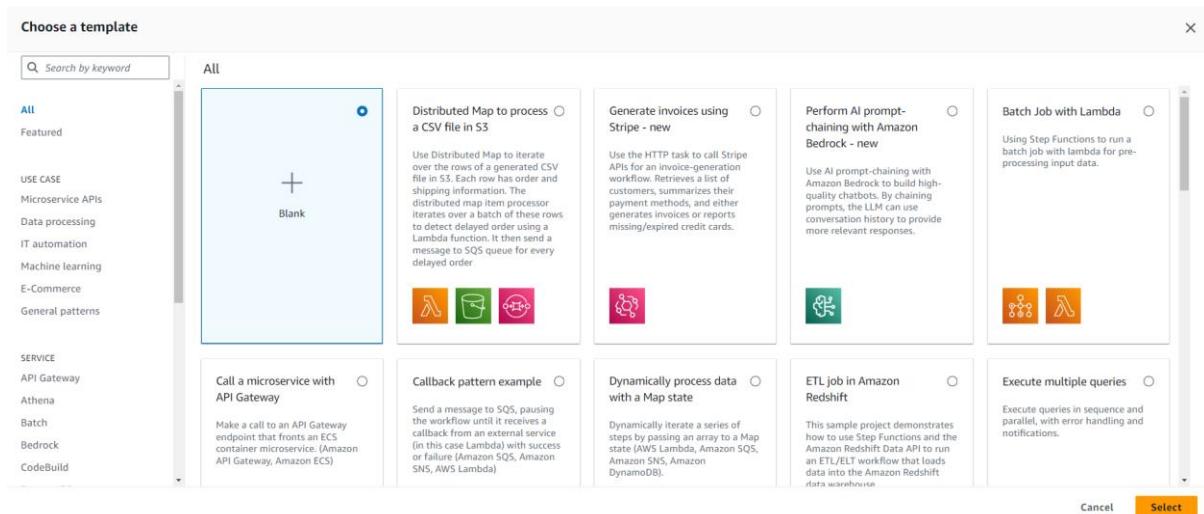


# Creating State Machine from Scratch with AWS Step Functions Workflow Studio

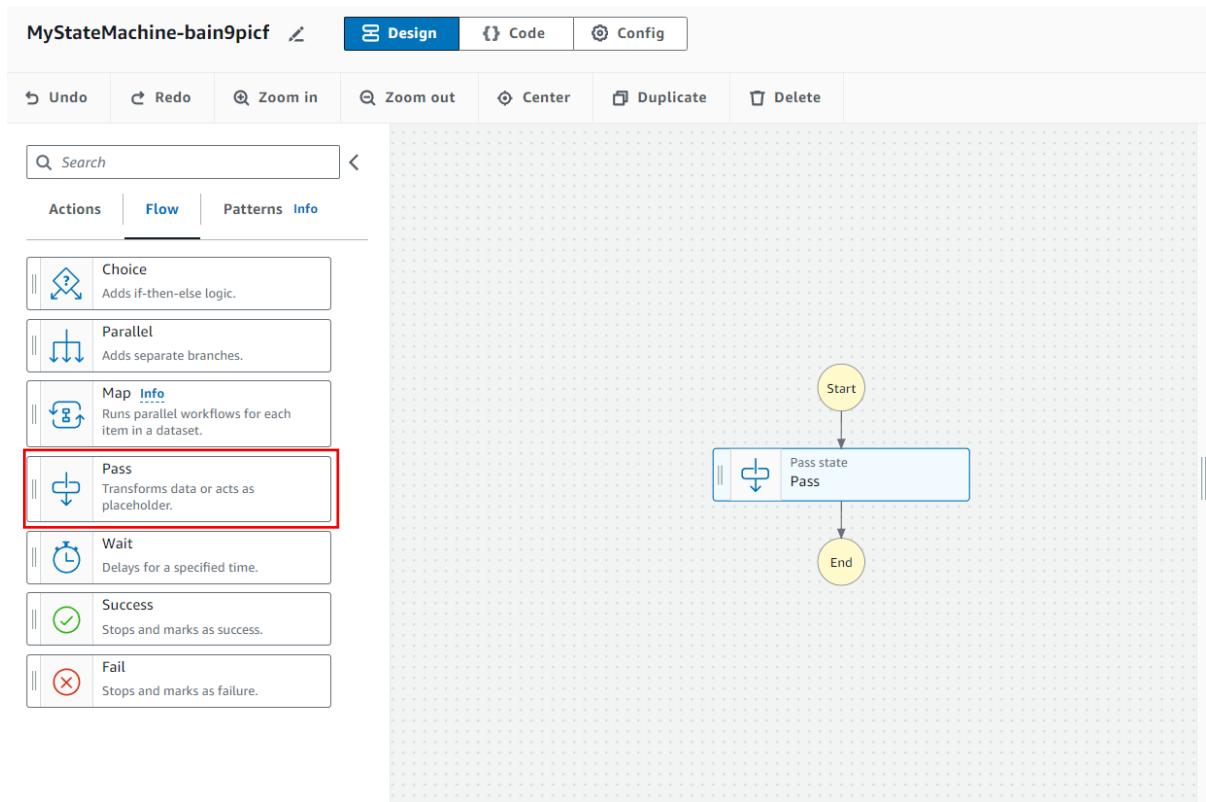
1. Go to the Step function and click on Create State Machine.



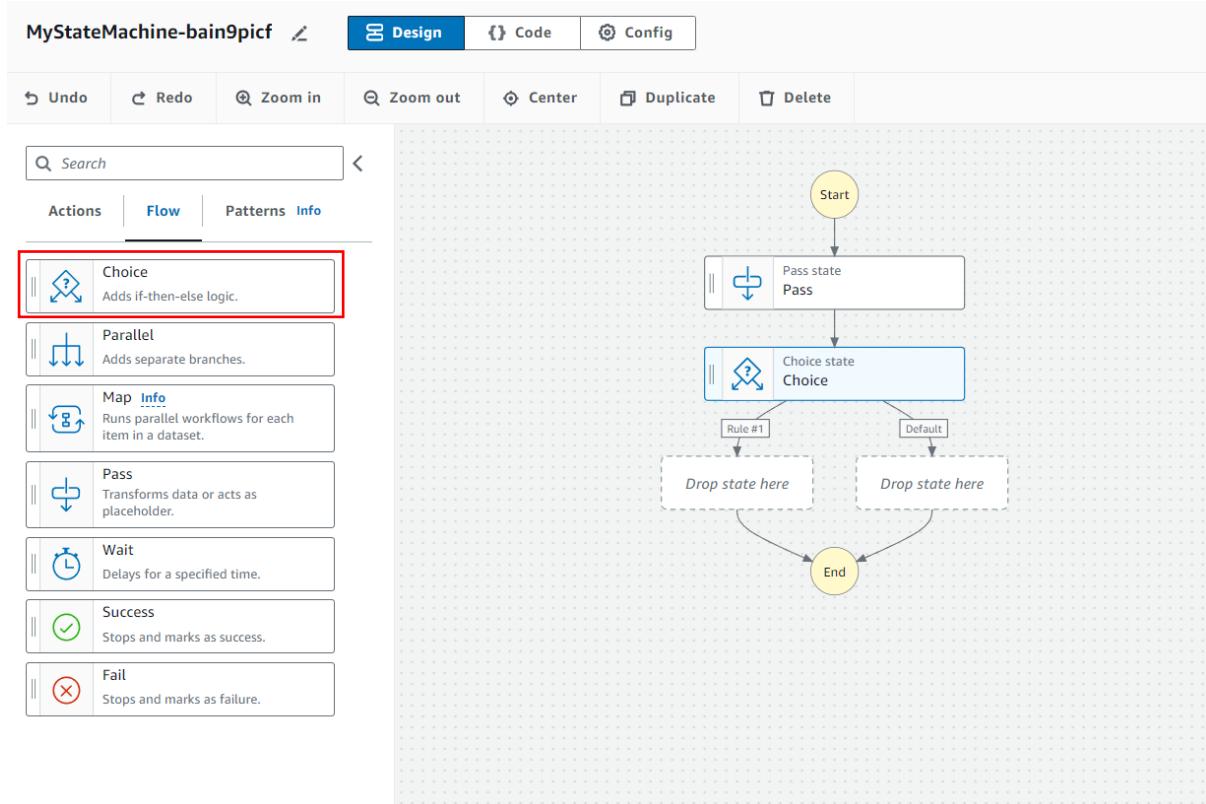
2. Then choose blank and click on Select.

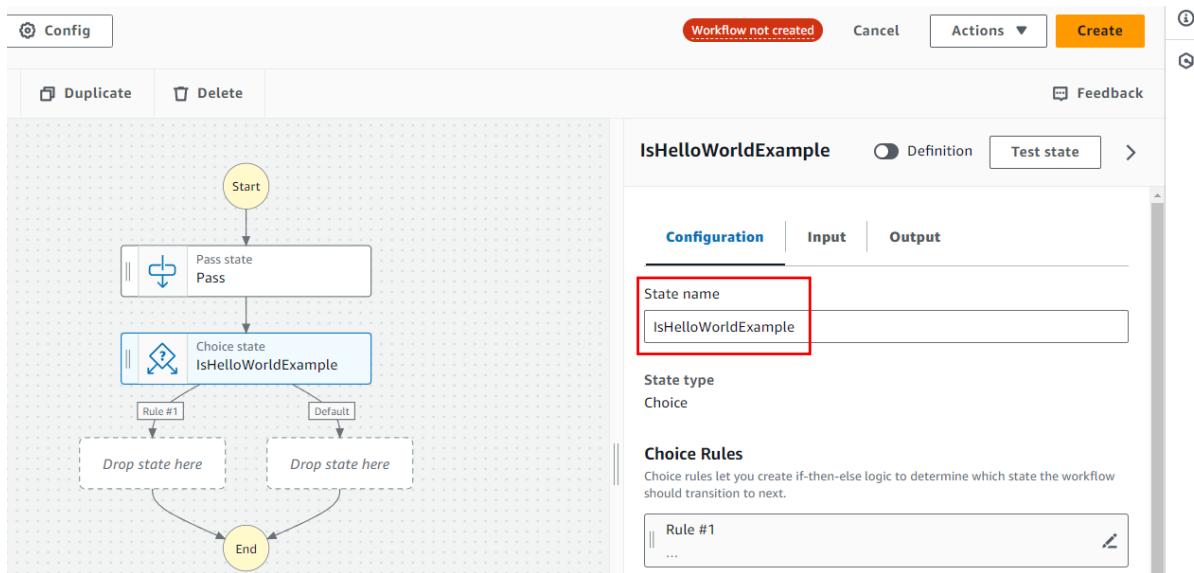


3. First you need to choose Flow, drag pass, and drop it on the middle of the screen as you can see below.

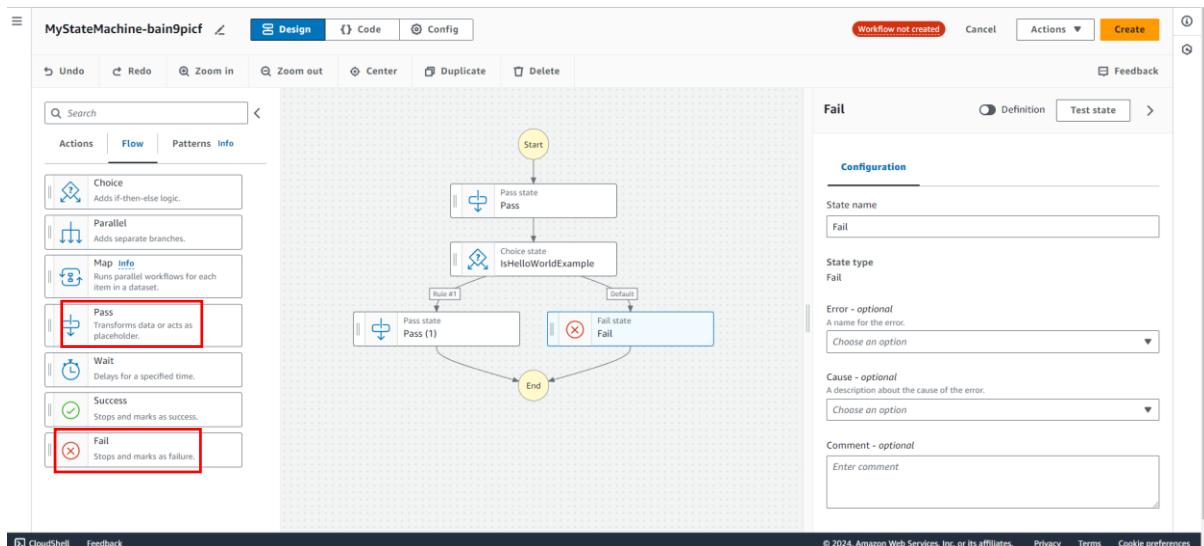


- After the Pass step drag and drop the choice step. Then you can change the name of your choice step.

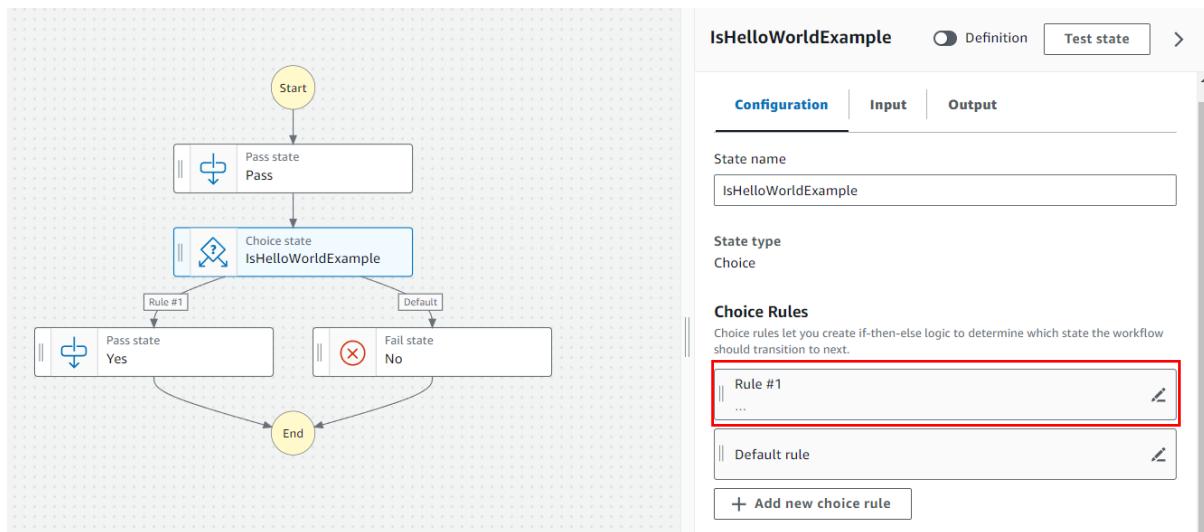




- Then from the flow you need to drag and drop the pass and fail step in the choice state. Then we are going to change their name to Yes and No.



- After that if you choose your choice state and scroll down in the configuration tab then you can add some rules. So, click on to add a rule.



7. Then you must click on add conditions. After that you need to specify the condition, so, in the Variable you have to write the name of your choice step in the same format as you can see below. Then for the operator choose is equal to and in the value choose Boolean constant and set it to true. Click on Save condition.

### **\$.IsHelloWorldExample**

#### **Choice Rules**

Choice rules let you create if-then-else logic to determine which state the workflow should transition to next.

**Rule #1** X Close

---

If these conditions are true:

**Add conditions**

**Conditions for rule #1** X

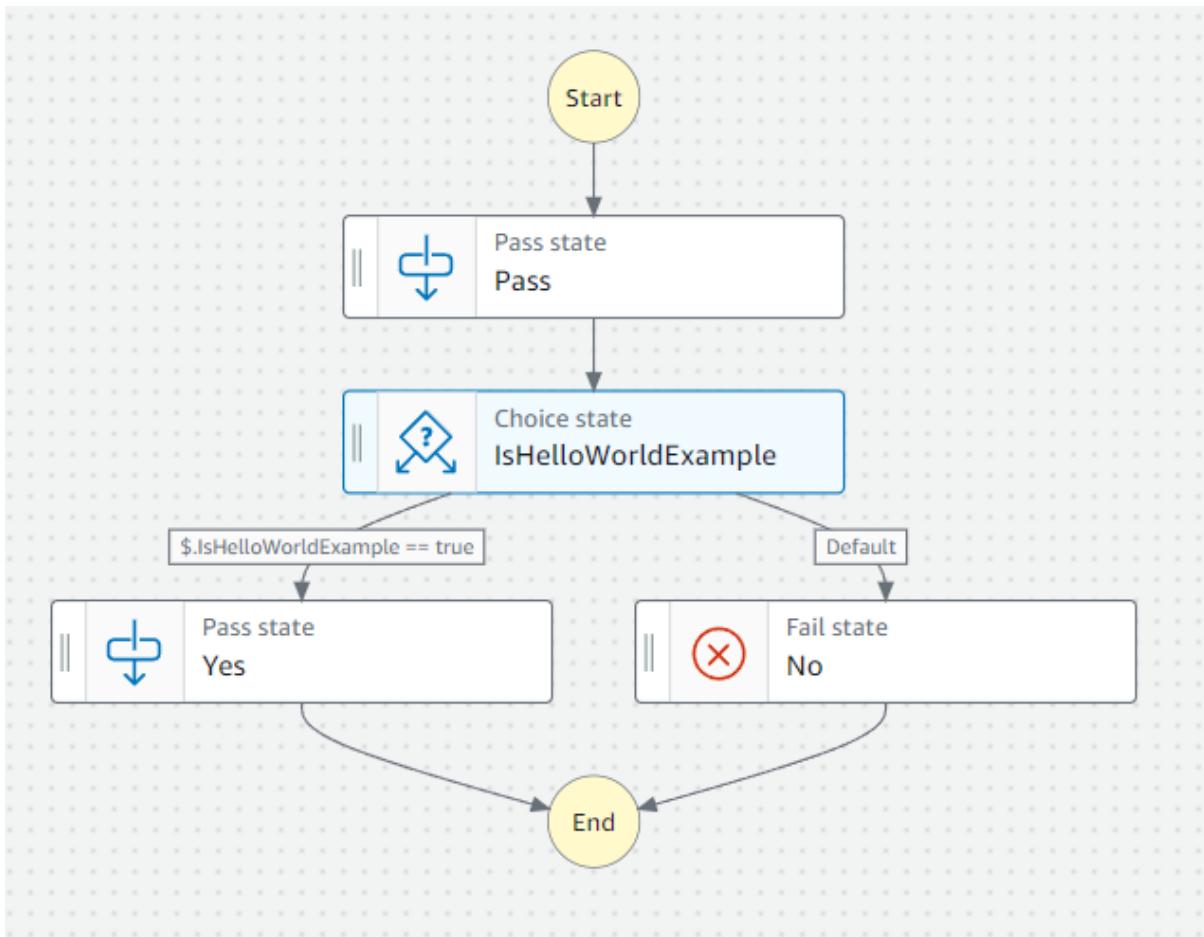
Choice rules contain conditional statements, which are used to evaluate one or more node values (called variables) in your state's JSON input. [Learn more](#)

Simple Evaluates a single conditional statement.

Not	Variable	Operator	Value
▼	\$.IsHelloWorldExam	is equal to	Boolean constant
true			

Must use JsonPath.

8. As you can see in the diagram below, we have added a choice for pass state now we are going to add a choice for fail state also.



9. So, if you scroll down below the default rule you will see that you can add a new choice rule. Click on it.

## Choice Rules

Choice rules let you create if-then-else logic to determine which state the workflow should transition to next.

Rule #1

`$.IsHelloWorldExample == true`



Default rule  Close

Default state

The state to transition to when no rule evaluates to true

No 

**Clear**

**+ Add new choice rule**

10. Then you must choose the same parameters for rule 2 but in the end, choose False and click on Save condition.

Conditions for rule #2 

Choice rules contain conditional statements, which are used to evaluate one or more node values (called variables) in your state's JSON input. [Learn more](#)

Simple Evaluates a single conditional statement. 

Not	Variable	Operator	Value
▼	\$.IsHelloWorldExam	is equal to	Boolean constant  false 

Must use JsonPath. 

Cancel 

11. Then you have to clear your default rule and in rule 2 choose the next state as NO.

Rule #2

If these conditions are true:

```
$.IsHelloWorldExample == false
```

[Edit conditions](#)

Then next state is:

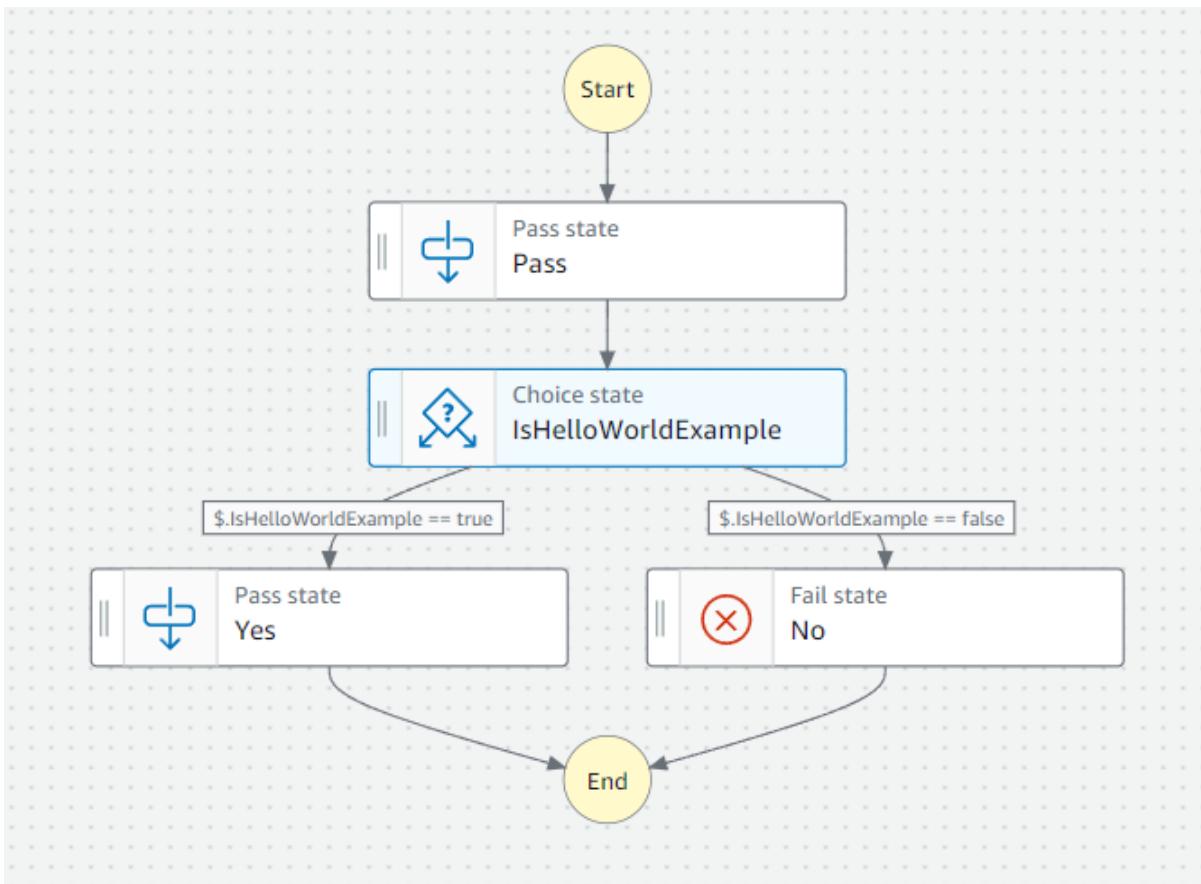
No

Comment - optional  
Provide a short description of the rule, which will appear in the graph.

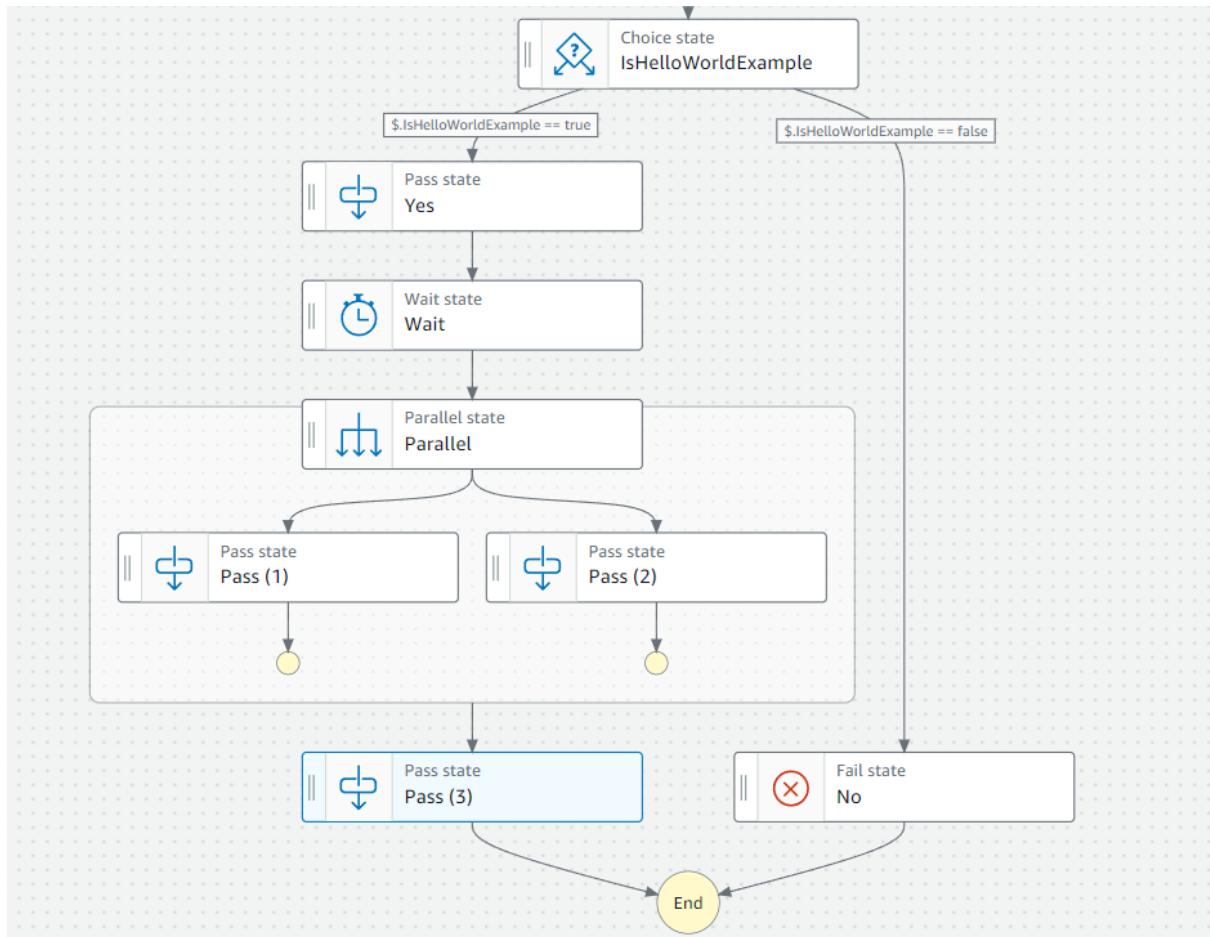
[Enter comment](#)

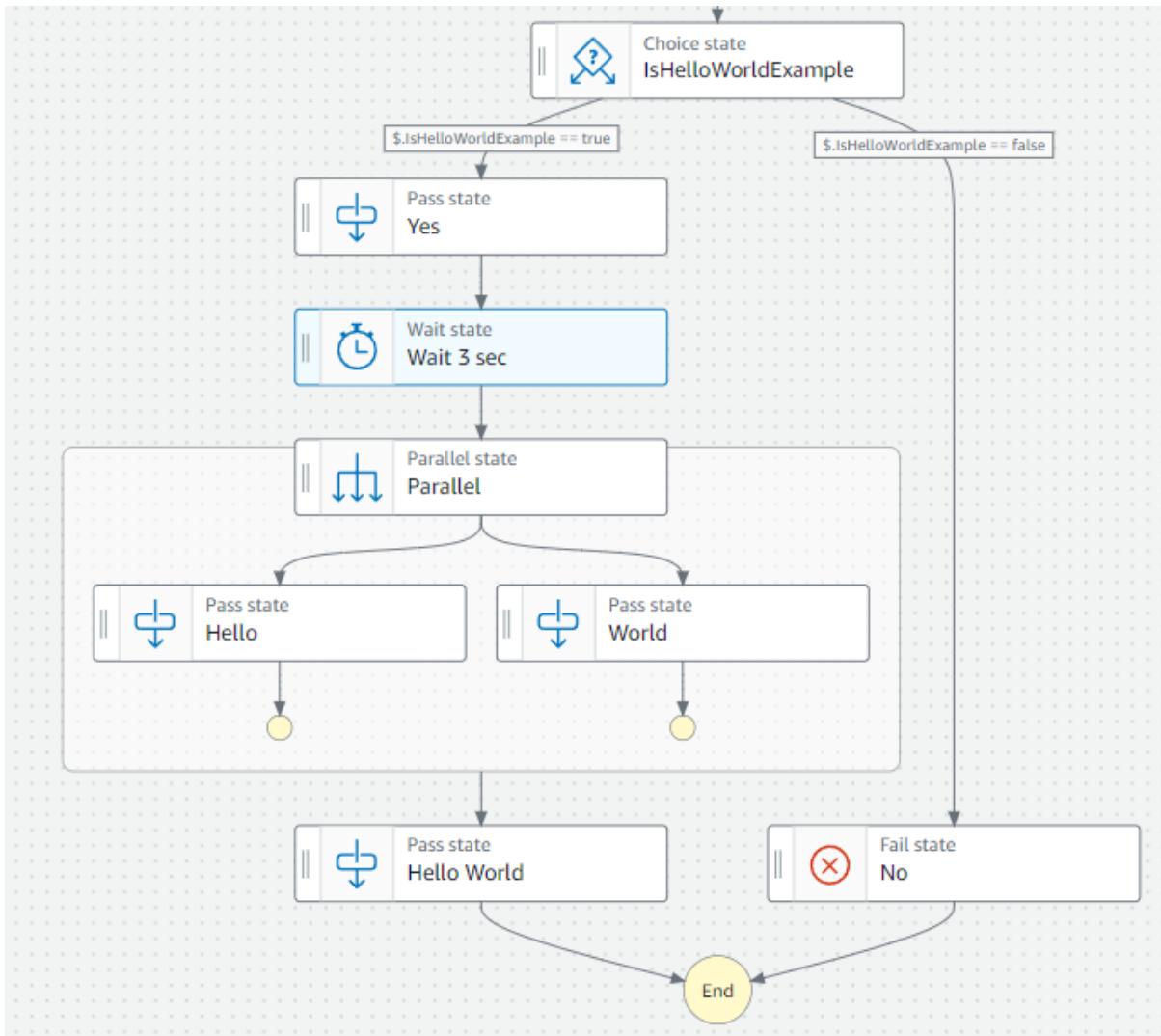
[Remove](#)

12. Below you can see your diagram. It is half done.

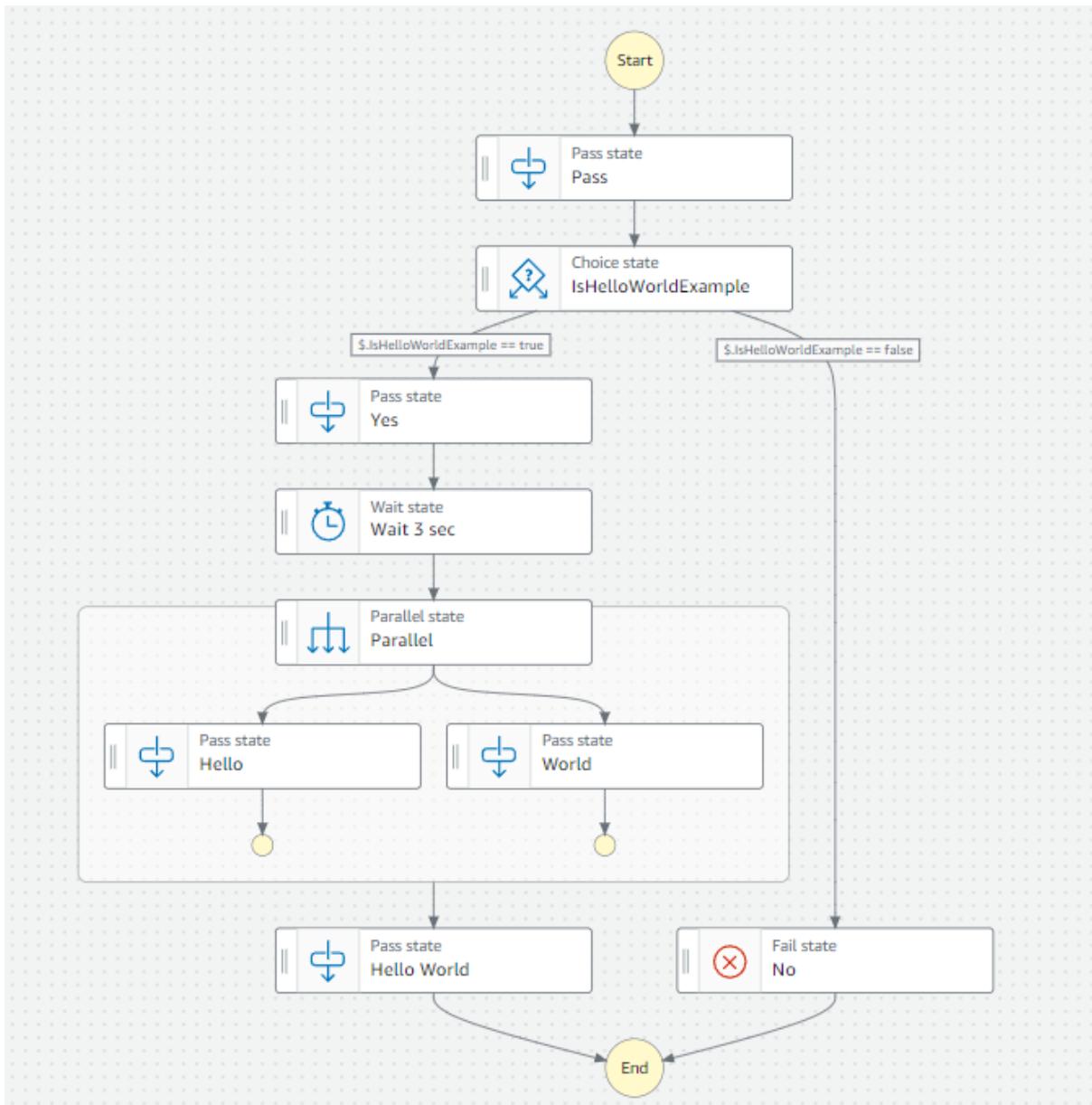


13. Below you can see the complete diagram, what happened here is that after the pass state Yes, we have added a wait state then under it we added a parallel state then in the parallel state we added two more pass states and below parallel state we again added a pass state.
14. Now I am going to name them properly and you must use the same names. Just remember in the wait state choose 3 second as the waiting period.

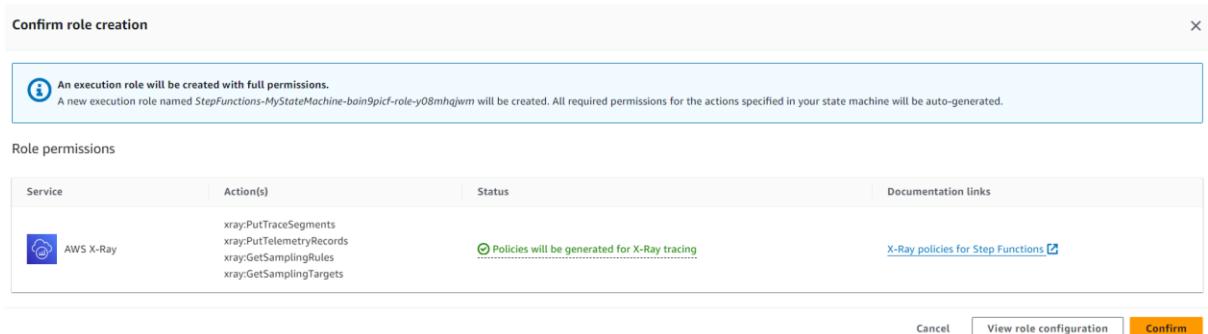




15. Below you can see the complete flow diagram. Now click on create.



16. Then click on confirm the role creation.



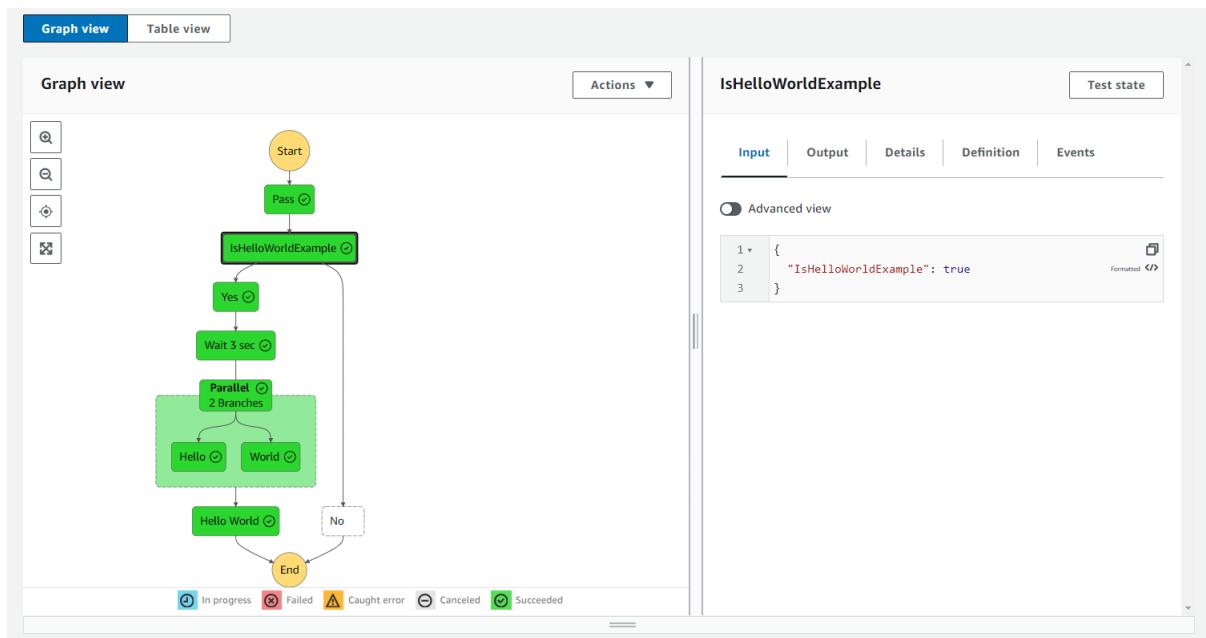
17. Below you can see that your state machine has been created but to start the execution you need to click on start execution.

18. Now you need to write the Input statement and click on start execution.

```
{
  "IsHelloWorldExample": true
}
```

19. Below you can see that the execution succeeded.

20. You can also view the graph view of your execution.



21. Now if you choose the input statement as false then you will see that the flow graph failed.

## Start execution

Name

889a2375-231b-4b9a-8bae-6bc8c4a0044b

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](#)

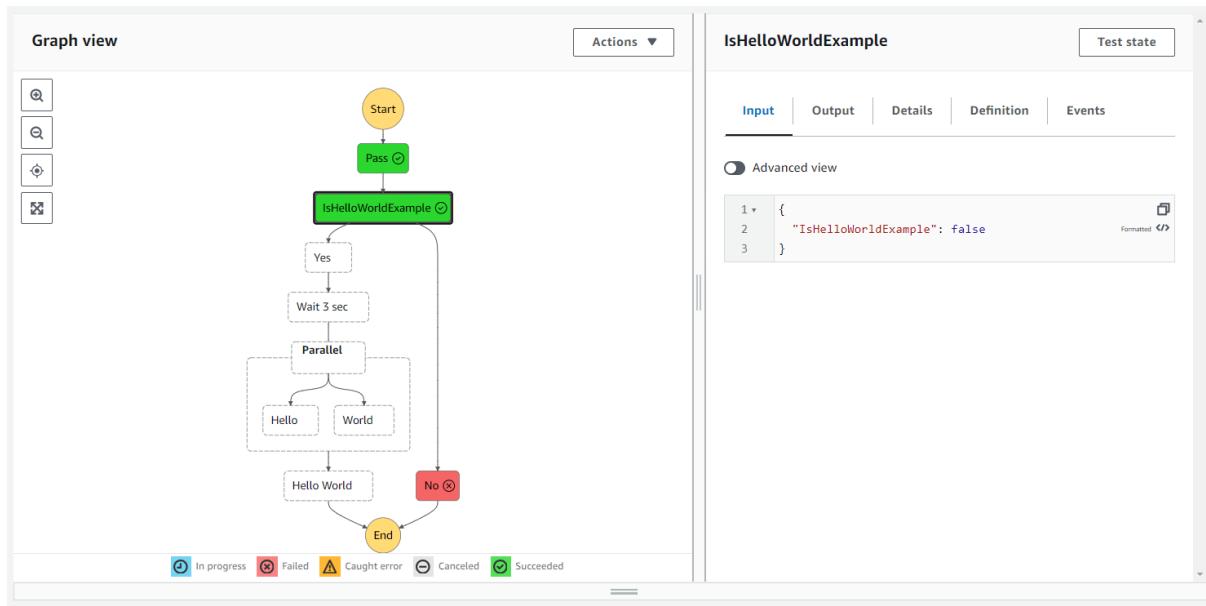
[Export](#)

[Import](#)

```
1 {  
2   "IsHelloWorldExample": false  
3 }
```

[ⓘ Start execution with latest revision](#)

Open in a new browser tab



22. You can also view the events.

Events (7)						
Filter by properties or search by keyword			Filter by a date and time range			
ID	Type	Step	Resource	Started After	Timestamp	
1	ExecutionStarted		0		Aug 23, 2024, 17:51:15.959 (UTC+05:30)	
2	PassStateEntered	Pass		00:00:00.038	Aug 23, 2024, 17:51:15.997 (UTC+05:30)	
3	PassStateExited	Pass		00:00:00.038	Aug 23, 2024, 17:51:15.997 (UTC+05:30)	
4	ChoiceStateEntered	IsHelloWorldExample		00:00:00.038	Aug 23, 2024, 17:51:15.997 (UTC+05:30)	
5	ChoiceStateExited	IsHelloWorldExample		00:00:00.038	Aug 23, 2024, 17:51:15.997 (UTC+05:30)	
6	FailStateEntered	No		00:00:00.038	Aug 23, 2024, 17:51:15.997 (UTC+05:30)	
7	ExecutionFailed			00:00:00.077	Aug 23, 2024, 17:51:16.036 (UTC+05:30)	