

ASSIGNMENT ON STEP FUNCTIONS

Ques-Create a step function with the below states:

1. Starting state -> a pass type state with user input as 'start', or 'stop' or 'terminate' or 'exit'
2. Choice state -> 3 choices : Based on the input value, it will execute 3 lambda functions (start ec2 lambda, stop ec2 lambda, terminate ec2 lambda).
3. Also add a fail state which will be executed if you enter 'exit' in your input json.
4. End state

Sol-

Steps to create the step functions:

Step 1 First of all, launch an ec2 instance with given role

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Instances' page is active, displaying a list of EC2 instances. The instance named 'step' is selected, and its details are shown in the 'Description' tab.

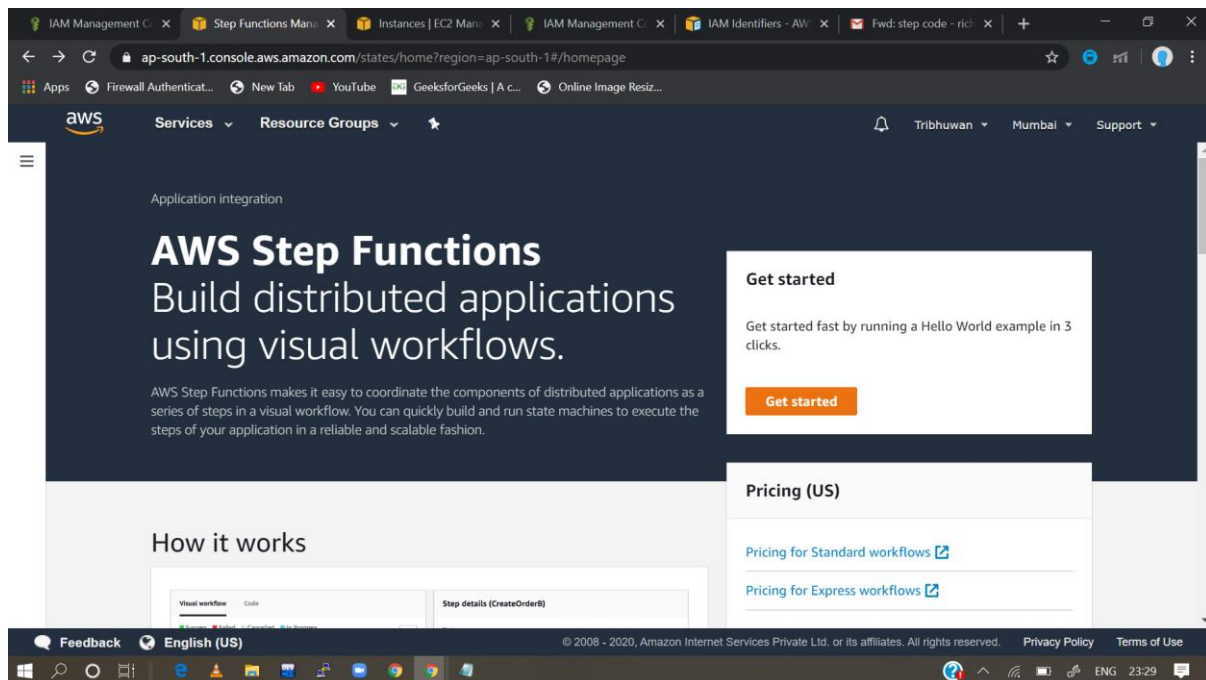
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-0a7ede802af24d4e8	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-233-
step	i-09d6532a75b71b794	t2.micro	ap-south-1a	running	Initializing	None	ec2-13-234-
	i-0efa874e6680541a5	t2.micro	ap-south-1a	running	2/2 checks ...	None	ec2-13-126-

Instance: **i-09d6532a75b71b794 (step)** Public DNS: **ec2-13-234-122-204.ap-south-1.compute.amazonaws.com**

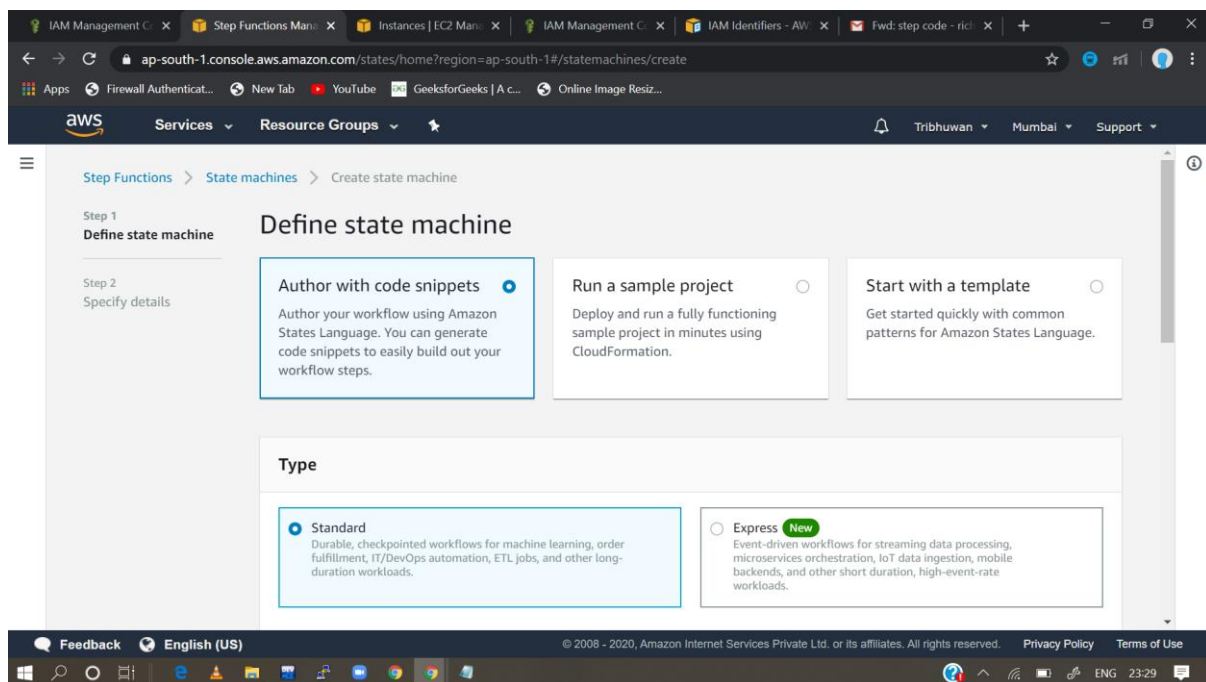
Description | Status Checks | Monitoring | Tags

Instance ID	i-09d6532a75b71b794	Public DNS (IPv4)	ec2-13-234-122-204.ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	13.234.122.204
Instance type	t2.micro	IPv6 Public IP	-

Step 2 Then go to step functions



Step 3 Then go to state machine



Step 4 Now generate the code snippet

The screenshot shows the AWS Step Functions console in the 'Definition' editor. The left pane contains the JSON definition for a state machine, and the right pane shows a visual state transition diagram.

```
12     },
13     {
14       "Variable": "$.instancetype",
15       "StringEquals": "STOP",
16       "Next": "stop_state"
17     },
18     {
19       "Variable": "$.instancetype",
20       "StringEquals": "TERMINATE",
21       "Next": "terminate_state"
22     }
23   ],
24   "Default": "exit_state"
25 },
26 "start_state": {
27   "Type": "Task",
28   "Resource": "arn:aws:lambda:ap-south-1a:124042716410:function:ec2-start",
29   "End": true
30 },
31 "stop_state": {
32   "Type": "Task",
33   "Resource": "arn:aws:lambda:ap-south-1a:124042716410:function:ec2-stop",
34   "End": true
35 },
36 "terminate_state": {
37   "Type": "Task",
38   "Resource": "arn:aws:lambda:ap-south-1a:124042716410:function:ec2-terminate",
39   "End": true
40 },
41 "exit_state": {
42   "Type": "Task",
43   "Resource": "arn:aws:lambda:ap-south-1a:124042716410:function:ec2-exit",
44   "End": true
45 }
```

The state transition diagram on the right shows a flow starting from a 'Start' node to an 'instancetype' node, which then branches into four parallel states: 'start_state', 'stop_state', 'terminate_state', and 'exit_state'. All four states converge at an 'End' node.

Step 5 Now the state machine successfully created

The screenshot shows the AWS Step Functions console with a green banner at the top stating 'State machine successfully created'. The page displays details for a state machine named 'MyStateMachine'.

State machine successfully created

Step Functions > State machines > MyStateMachine

MyStateMachine [Edit] [Start execution] [Delete] [Actions]

Details

ARN	arn:aws:states:ap-south-1:124042716410:stateMachine:MyStateMachine
IAM role ARN	arn:aws:iam::124042716410:role/service-role/StepFunctions-MyStateMachine-role-8e177d8e
Type	Standard
Creation date	Apr 20, 2020 11:37:39.303 PM

Executions | Logging | Definition | Tags