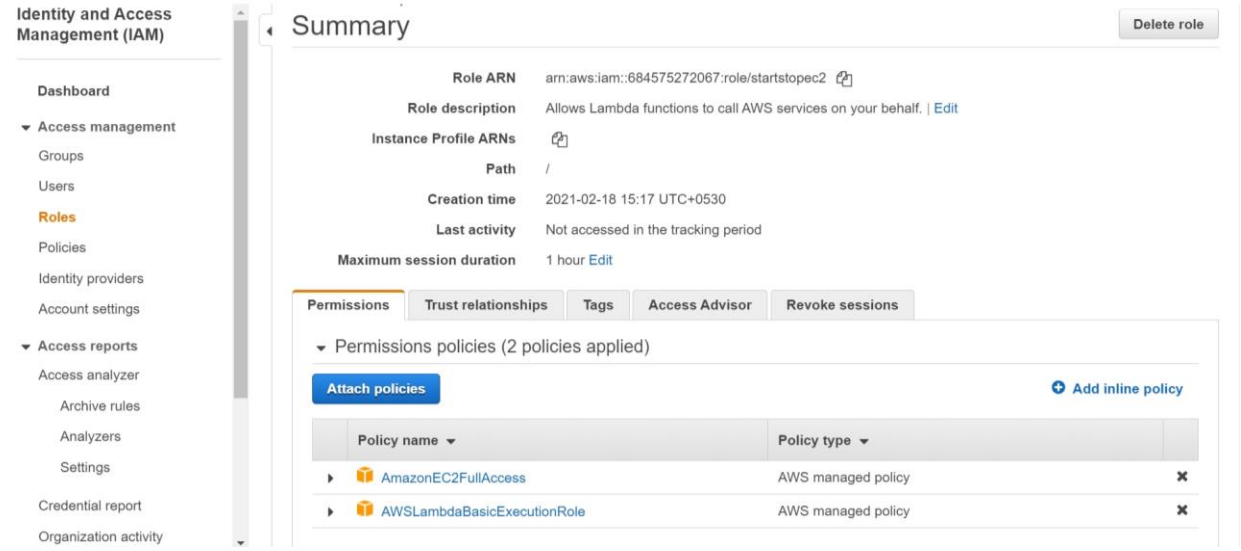
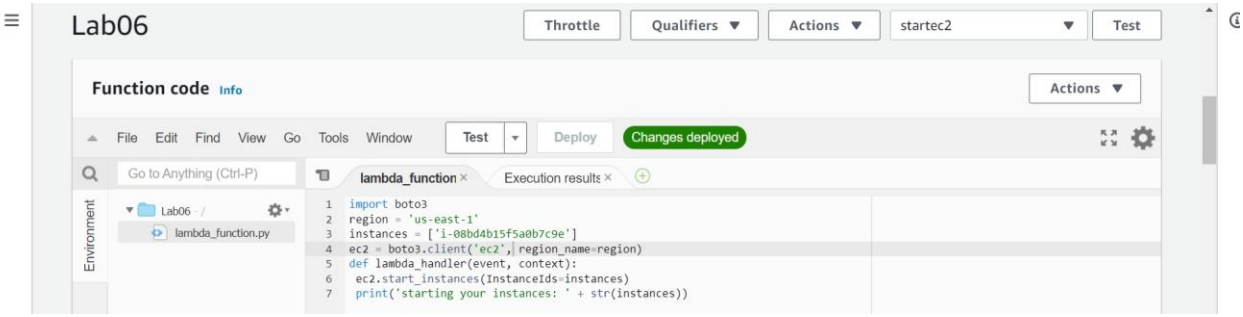
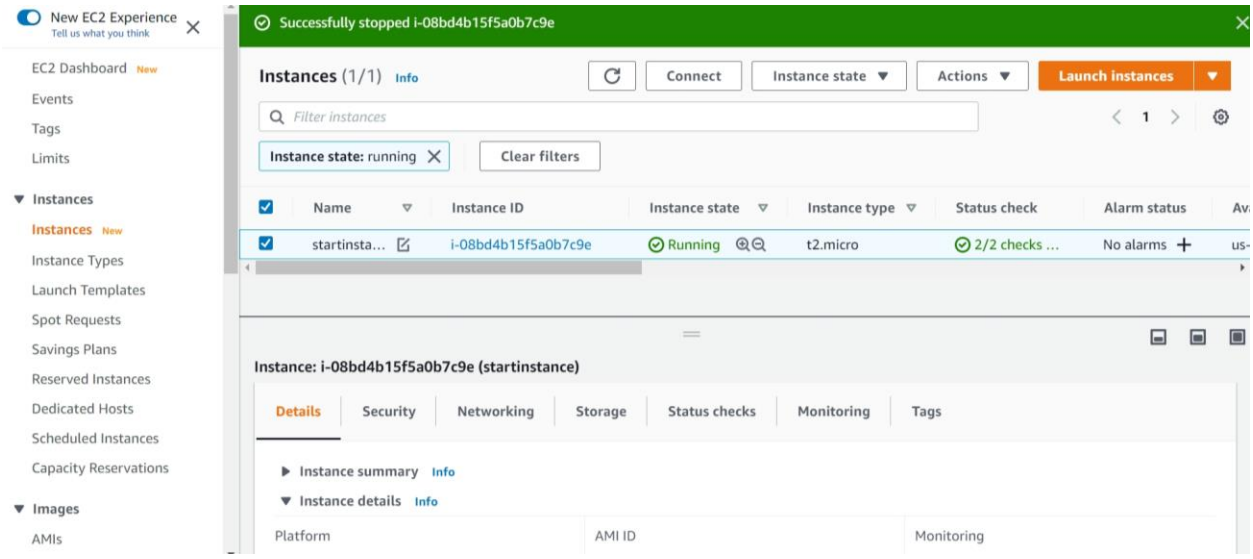
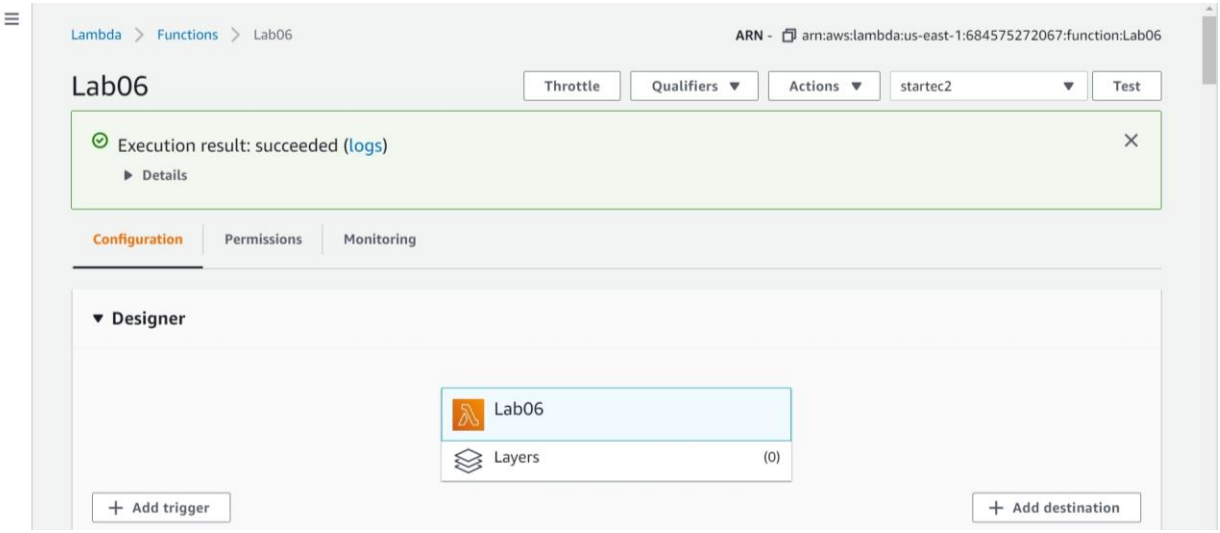
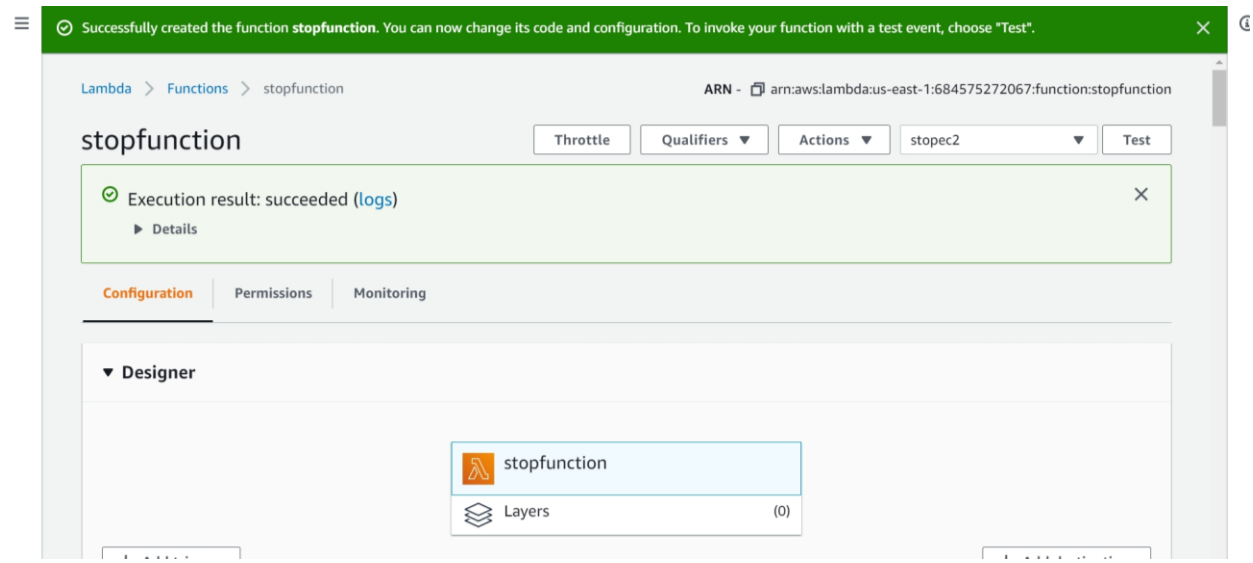
Arnav Bansal
E18CSE028
Batch EB01
Lab 6

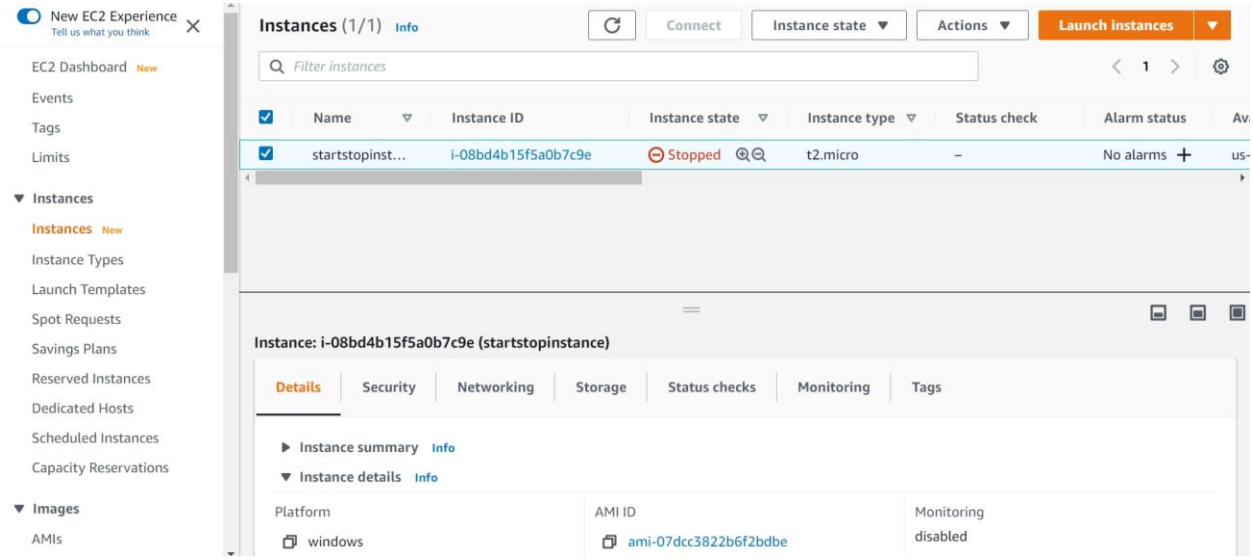


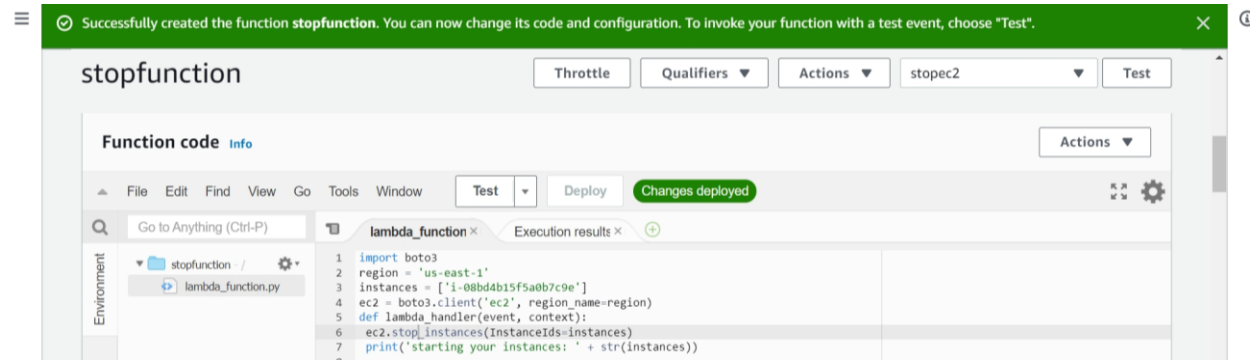


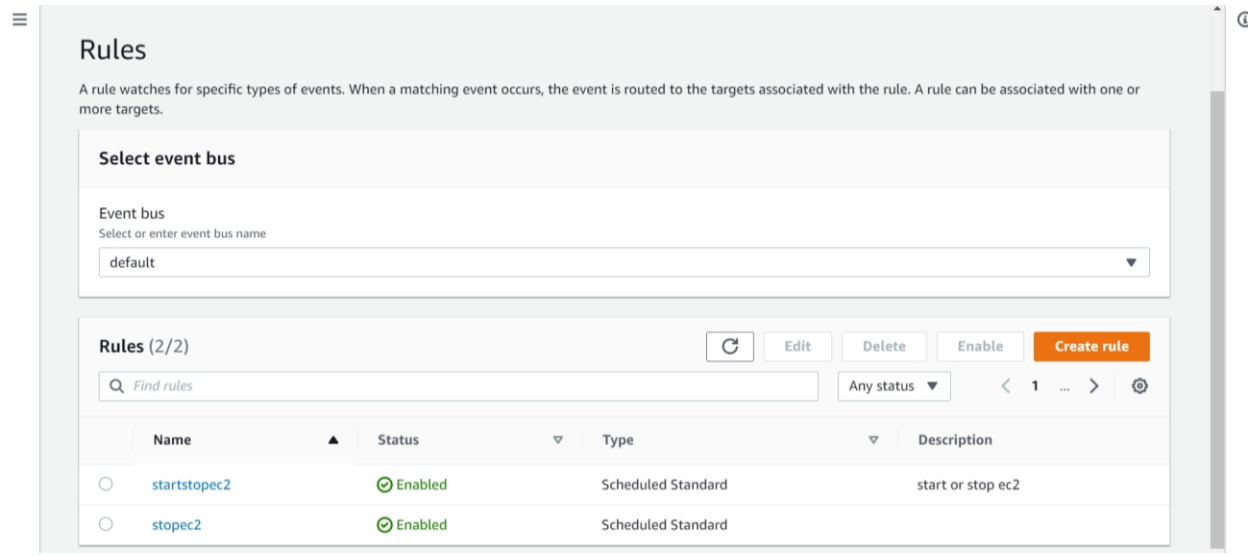

New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Successfully stopped i-08bd4b15f5a0b7c9e
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Instance state: running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
startinsta...
i-08bd4b15f5a0b7c9e
Running
t2.micro
2/2 checks ...
No alarms
Instance: i-08bd4b15f5a0b7c9e (startinstance)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance details Info
Platform
AMI ID
Monitoring

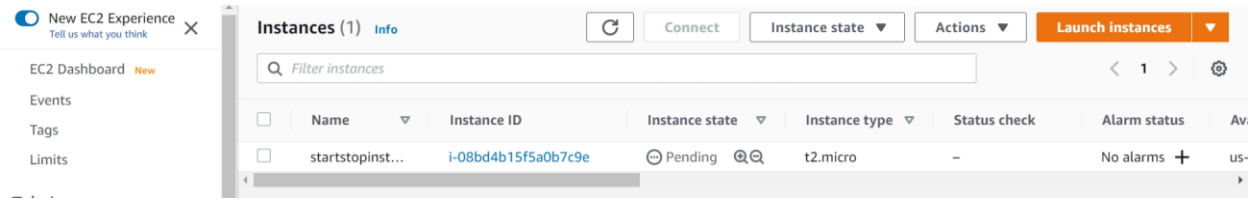

Lambda > Functions > Lab06
ARN - arn:aws:lambda:us-east-1:684575272067:function:Lab06
Lab06
Throttle
Qualifiers
Actions
startec2
Test
Execution result: succeeded (logs)
Details
Configuration
Permissions
Monitoring
Designer
Lab06
Layers
(0)
Add trigger
Add destination

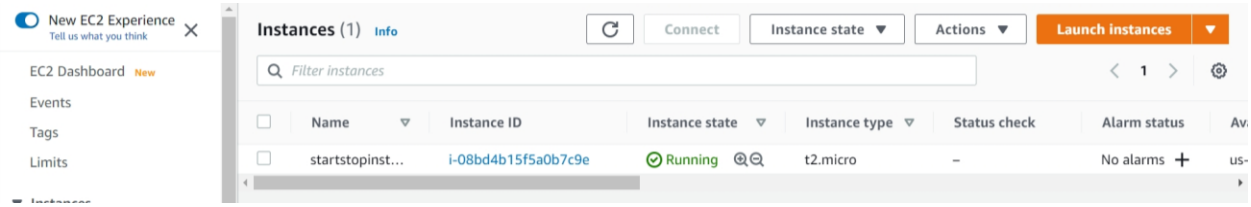

Successfully created the function stopfunction. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > stopfunction
ARN - arn:aws:lambda:us-east-1:684575272067:function:stopfunction
stopfunction
Throttle
Qualifiers
Actions
stopec2
Test
Execution result: succeeded (logs)
Details
Configuration
Permissions
Monitoring
Designer
stopfunction
Layers
(0)

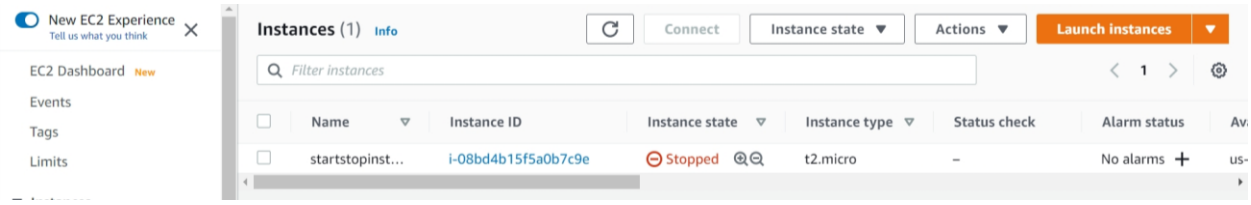

New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
startstopinst...
i-08bd4b15f5a0b7c9e
Stopped
t2.micro
–
No alarms
Instance: i-08bd4b15f5a0b7c9e (startstopinstance)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance details Info
Platform
windows
AMI ID
ami-07dcc3822b6f2bdbe
Monitoring
disabled

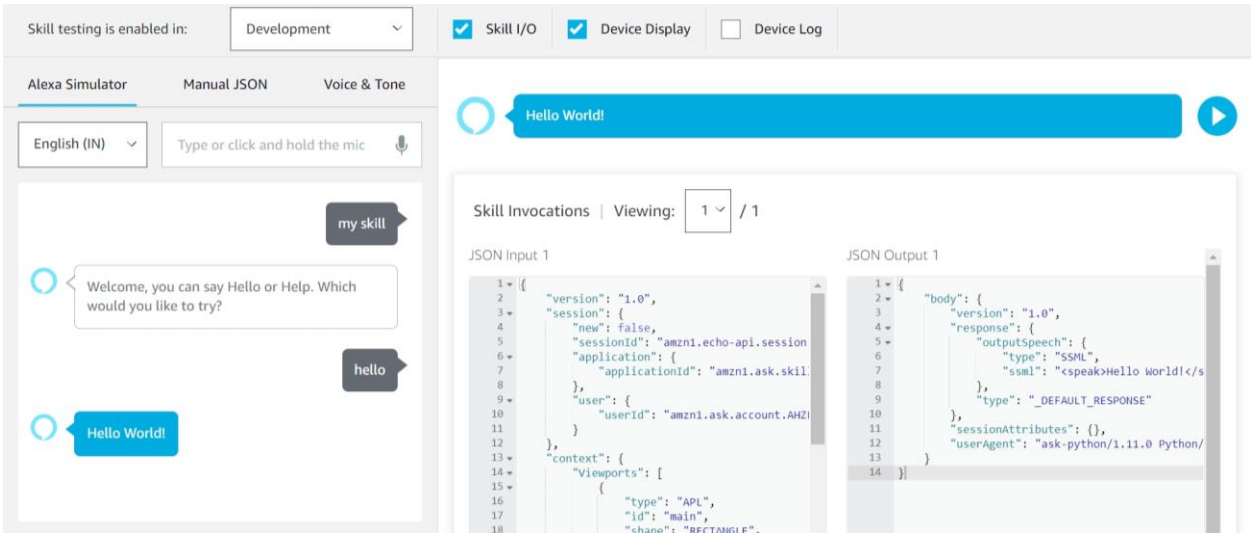

Successfully created the function stopfunction. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
stopfunction
Throttle
Qualifiers
Actions
stopec2
Test
Function code Info
Actions
File Edit Find View Go Tools Window
Test
Deploy
Changes deployed
Go to Anything (Ctrl-P)
lambda_function
Execution results
Environment
stopfunction
lambda_function.py
import boto3
region = 'us-east-1'
instances = ['i-08bd4b15f5a0b7c9e']
ec2 = boto3.client('ec2', region_name=region)
def lambda_handler(event, context):
ec2.stop_instances(InstanceIds=instances)
print('starting your instances: ' + str(instances))

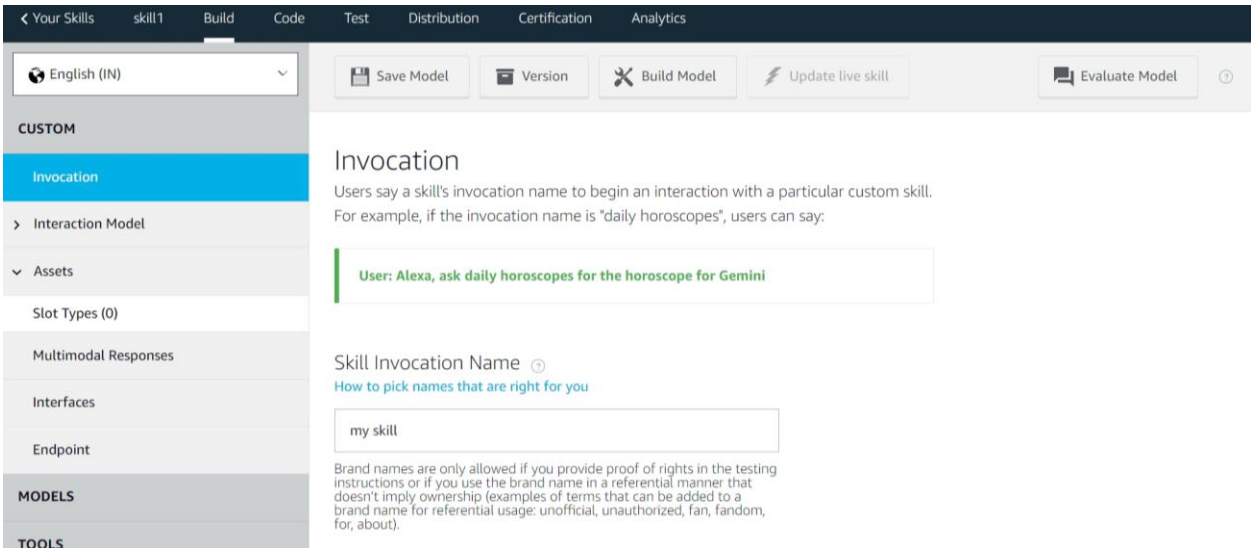

Rules
A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.
Select event bus
Event bus
Select or enter event bus name
default
Rules (2/2)
Edit
Delete
Enable
Create rule
Find rules
Any status
1
Name
Status
Type
Description
startstopec2
Enabled
Scheduled Standard
start or stop ec2
stopec2
Enabled
Scheduled Standard

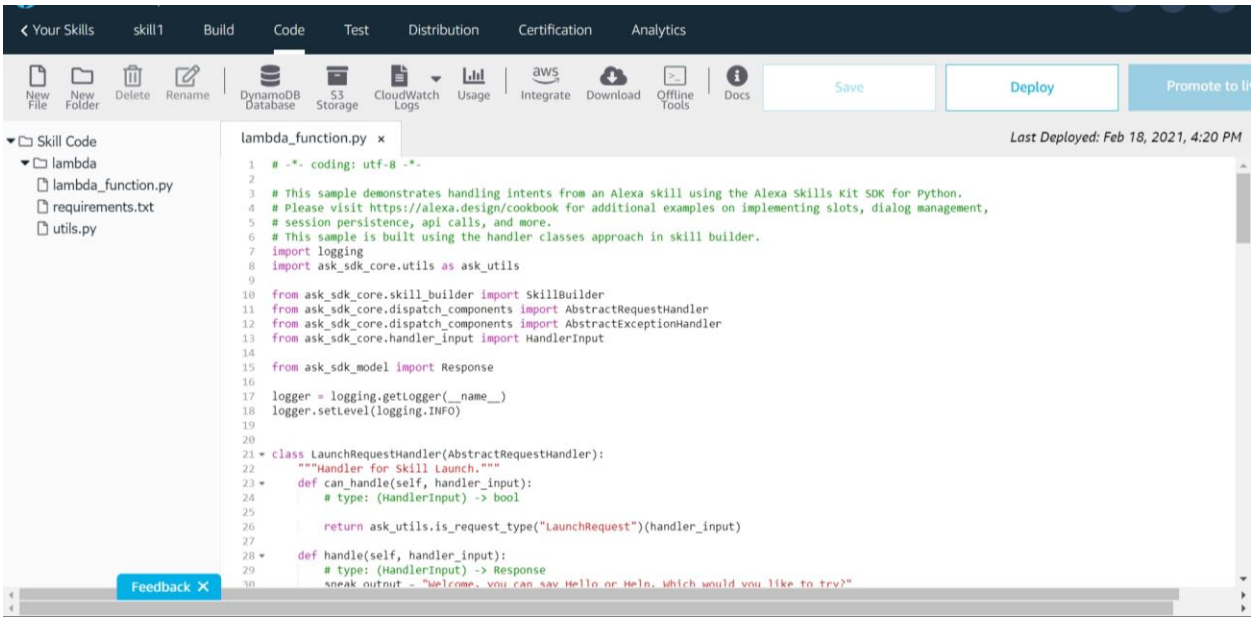

New EC2 Experience
Tell us what you think
EC2 Dashboard
New
Events
Tags
Limits
Instances (1)
Info
Connect
Instance state
Actions
Launch instances
Filter instances
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
startstopinst...
i-08bd4b15f5a0b7c9e
Pending
t2.micro
–
No alarms


New EC2 Experience
Tell us what you think
EC2 Dashboard
New
Events
Tags
Limits
Instances (1)
Info
Connect
Instance state
Actions
Launch instances
Filter instances
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
startstopinst...
i-08bd4b15f5a0b7c9e
Running
t2.micro
–
No alarms


New EC2 Experience
Tell us what you think
EC2 Dashboard
New
Events
Tags
Limits
Instances (1)
Info
Connect
Instance state
Actions
Launch instances
Filter instances
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
startstopinst...
i-08bd4b15f5a0b7c9e
Stopped
t2.micro
–
No alarms


Skill testing is enabled in:
Development
Skill I/O
Device Display
Device Log
Alexa Simulator
Manual JSON
Voice & Tone
English (IN)
Type or click and hold the mic
my skill
Welcome, you can say Hello or Help. Which would you like to try?
hello
Hello World!
Hello World!
Skill Invocations | Viewing: 1 / 1
JSON Input 1
JSON Output 1


Your Skills
skill1
Build
Code
Test
Distribution
Certification
Analytics
English (IN)
Save Model
Version
Build Model
Update live skill
Evaluate Model
CUSTOM
Invocation
Interaction Model
Assets
Slot Types (0)
Multimodal Responses
Interfaces
Endpoint
MODELS
TOOLS
Invocation
Users say a skill's invocation name to begin an interaction with a particular custom skill.
For example, if the invocation name is "daily horoscopes", users can say:
User: Alexa, ask daily horoscopes for the horoscope for Gemini
Skill Invocation Name
How to pick names that are right for you
my skill
Brand names are only allowed if you provide proof of rights in the testing instructions or if you use the brand name in a referential manner that doesn't imply ownership (examples of terms that can be added to a brand name for referential usage: unofficial, unauthorized, fan, fandom, for, about).


Your Skills
skill1
Build
Code
Test
Distribution
Certification
Analytics
New File
New Folder
Delete
Rename
DynamoDB Database
S3 Storage
CloudWatch Logs
Usage
Integrate
Download
Offline Tools
Docs
Save
Deploy
Promote to li
Skill Code
lambda
lambda_function.py
requirements.txt
utils.py
lambda_function.py ×
Last Deployed: Feb 18, 2021, 4:20 PM
# -*- coding: utf-8 -*-
# This sample demonstrates handling intents from an Alexa skill using the Alexa Skills Kit SDK for Python.
# Please visit https://alexa.design/cookbook for additional examples on implementing slots, dialog management,
# session persistence, api calls, and more.
# This sample is built using the handler classes approach in skill builder.
import logging
import ask_sdk_core.utils as ask_utils
from ask_sdk_core.skill_builder import SkillBuilder
from ask_sdk_core.dispatch_components import AbstractRequestHandler
from ask_sdk_core.dispatch_components import AbstractExceptionHandler
from ask_sdk_core.handler_input import HandlerInput
from ask_sdk_model import Response
logger = logging.getLogger(__name__)
logger.setLevel(logging.INFO)
class LaunchRequestHandler(AbstractRequestHandler):
"""Handler for Skill Launch."""
def can_handle(self, handler_input):
# type: (HandlerInput) -> bool
return ask_utils.is_request_type("LaunchRequest")(handler_input)
def handle(self, handler_input):
# type: (HandlerInput) -> Response
Feedback