Greengrass Getting starter

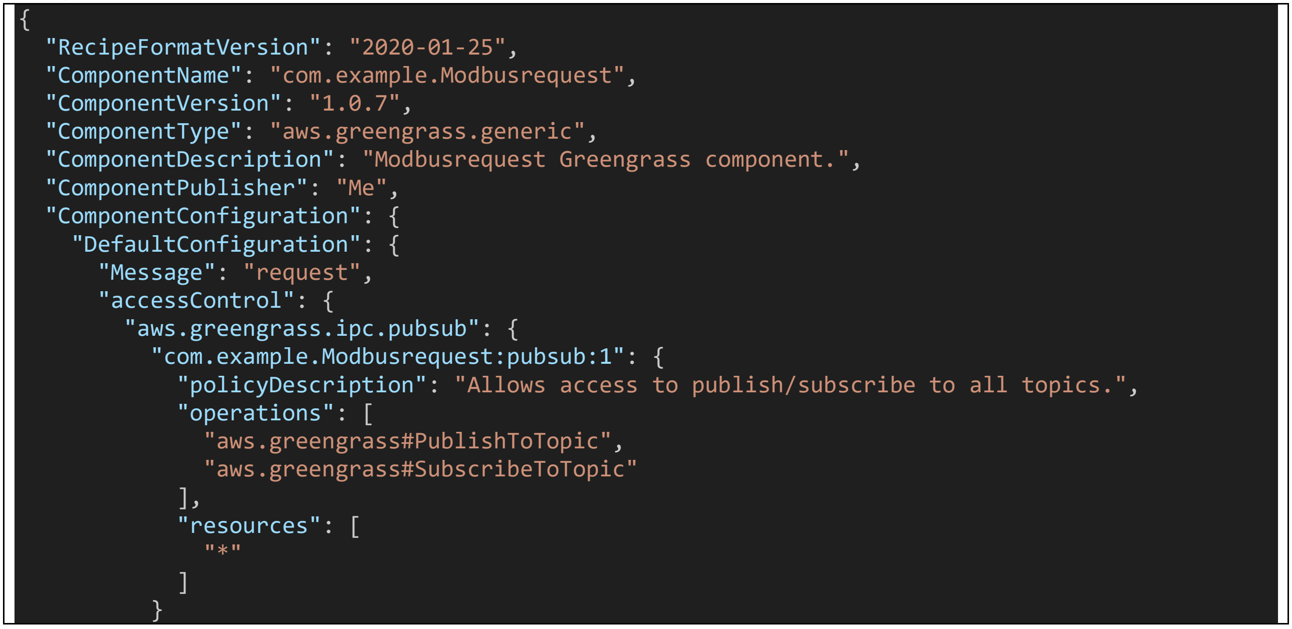
CREATE GREENGRASS CORE DEVICE:

1. Go to AWS IoT --> Greengrass devices --> Core Devices from AWS console
2. Select Set up one core device
3. Follow the instruction to install java and greengrass core:
    a. export AWS_ACCESS_KEY_ID=<insert_code>
    b. export AWS_SECRET_ACCESS_KEY=<insert_the_code>
    c. curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip > greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d GreengrassInstaller
    d. sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE -jar ./GreengrassInstaller/lib/Greengrass.jar --aws-region eu-west-1 --thing-name GreengrassQuickStartCore-1887ff5e582 --thing-group-name GreengrassQuickStartGroup --component-default-user ggc_user:ggc_group --provision true --setup-system-service true --deploy-dev-tools true

DEPLOY COMPONENTS (Guide: https://docs.aws.amazon.com/greengrass/v2/developerguide/create-first-component.html):

1. Make sure that pip is install into local machine; alternatively: *sudo yum install python3-pip*
2. Copy your script into S3 bucket thought AWS console:
   aws s3 cp .\<ARTIFACT_NAME.py>.py s3://greengrass-bucket-001/artifacts/com.example.ARTIFACT_NAME/1.0.0/ARTIFACT_NAME.py
3. Make sure your s3 bucket is in the same region of Greengrass core device (no cross-region is allowed).
4. Make sure GetObject policy is attached to the Greengrass role; alternatively follow this: https://docs.aws.amazon.com/greengrass/v2/developerguide/device-service-role.html
5. Select Components from AWS console
6. Select Create components
7. Select YAML file and paste the following. Change the name of artifact with *ARTIFACT_NAME*
8. Click create components:

(Recipe)



```
{
  "RecipeFormatVersion": "2020-01-25",
  "ComponentName": "com.example.Modbusrequest",
  "ComponentVersion": "1.0.7",
  "ComponentType": "aws.greengrass.generic",
  "ComponentDescription": "Modbusrequest Greengrass component.",
  "ComponentPublisher": "Me",
  "ComponentConfiguration": {
    "DefaultConfiguration": {
      "Message": "request",
      "accessControl": {
        "aws.greengrass.ipc.pubsub": {
          "com.example.Modbusrequest:pubsub:1": {
            "policyDescription": "Allows access to publish/subscribe to all topics.",
            "operations": [
              "aws.greengrass#PublishToTopic",
              "aws.greengrass#SubscribeToTopic"
            ],
            "resources": [
              "*"
            ]
          }
        }
      }
    }
  },
  "Manifests": [
    {
      "Platform": {
        "os": "linux"
      },
      "Name": "Linux",
      "Lifecycle": {
        "Install": {
          "script": "python3 -m pip install --user awsiotsdk"
        },
        "Run": "python3 {artifacts:path}/Modbusrequest.py '{configuration:/Message}'"
      },
      "Artifacts": [
        {
          "Uri": "s3://greengrass-bucket-
001/artifacts/com.example.Modbusrequest/1.0.7/Modbusrequest.py",
          "Digest": "JPgTpi+FYFqNMZO/a+iLnExDJiKWH4IdEh144vb092c=",
          "Algorithm": "SHA-256",
          "Unarchive": "NONE",
          "Permission": {
            "Read": "OWNER",
            "Execute": "NONE"
          }
        }
      ]
    }
  ],
  "Lifecycle": {}
}
```

(Artifact)

```
# Modbusrequest.py
import sys
import datetime
import time
import logging
import awsiot.greengrasscoreipc
from awsiot.greengrasscoreipc.model import (
    PublishToTopicRequest,
    PublishMessage,
    BinaryMessage
)
# Setup logging to stdout
logger = logging.getLogger(__name__)
logging.basicConfig(stream=sys.stdout, level=logging.DEBUG)
FUTURE_WAIT_TIME = 10
SLEEP_TIME = 5
ipc_client = awsiot.greengrasscoreipc.connect()
topic = "modbus/request/conveyer"
message = '{ "id": "TestRequest_0_7", "function": "ReadCoils", "address": 00001, "quantity": 10
}'
logger.debug("topic: " + topic)
logger.debug("message: " + message)
request = PublishToTopicRequest()
request.topic = topic
publish_message = PublishMessage()
publish_message.binary_message = BinaryMessage()
publish_message.binary_message.message = bytes(message, "utf-8")
request.publish_message = publish_message
```

```
while True:
    operation = ipc_client.new_publish_to_topic()
    operation.activate(request)
    future = operation.get_response()
    future.result(FUTURE_WAIT_TIME)
    # Append the message to the log file.
    logger.info(message)
    print("VAI A DORMIRE ver 7")
    time.sleep(SLEEP_TIME)
```

LOG THE COMPONENTS:

1. Into core device check the log: *sudo tail -f /greengrass/v2/logs/com.example.HelloWorld.log*
2. Restart a components: sudo `/greengrass/v2`/bin/greengrass-cli component restart --names `"com.example.HelloWorld"`

CONFIGURE COMPONETS TO COMPONENTS MQTT MESSAGES (https://docs.aws.amazon.com/greengrass/v2/developerguide/ipc-publish-subscribe.html)

CONFIGURE COMPONTENTS TO SEND MQTT TO AWS IOT CORE (https://docs.aws.amazon.com/greengrass/v2/developerguide/ipc-iot-core-mqtt.html):

- To use AWS IoT Core MQTT messaging in a custom component, you must define authorization policies that allow your component to send and receive messages on topics. For information about defining authorization policies (https://docs.aws.amazon.com/greengrass/v2/developerguide/interprocess-communication.html#ipc-authorization-policies)