Lab 10: Internet of Things

The lab is about integrating an IoT device with AWS IoT and get notified by SMS of events coming from the device using Simple Notification Service.   
  
This is done by firstly creating a digital IoT Shadow of a physical connected device and then by adding an AWS IoT rule, using the AWS IoT Rule engine. This will subscribe to the message events coming from the device and forward them all to a SNS. This SNS topic will have a subscriber which will be an email or phone (reached using an SMS message).  
  
The way to send the telemetry data to AWS is done through a custom client code that we are going to fill in. This will allow getting more knowledge about AWS IoT and the particularities of the IoT communication and processing, in general.



# Install all the necessary dependencies

## Prerequisites - code

* Clone the repository [CloudShape IoT lab](https://github.com/bproca/CloudShapeIoT) to a local directory
  + Instead of cloning, you can download a ZIP with the code from the Downloads section at the above link

git clone <https://github.com/bproca/CloudShapeIoTLab>

## Prerequisites - dependencies

* Install the Paho MQTT client library, see [paho-mqtt/installation](https://pypi.org/project/paho-mqtt/#installation).
  + You may need to additionally install python 3 and pipif they are not installed already on your machine

pip install paho-mqtt

Or with virtualenv:

virtualenv paho-mqtt

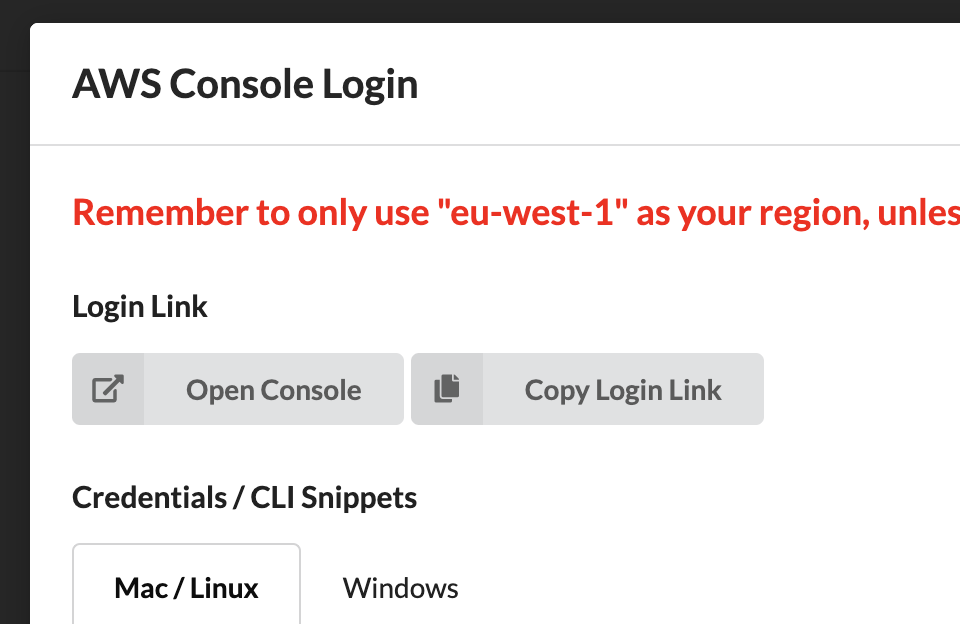
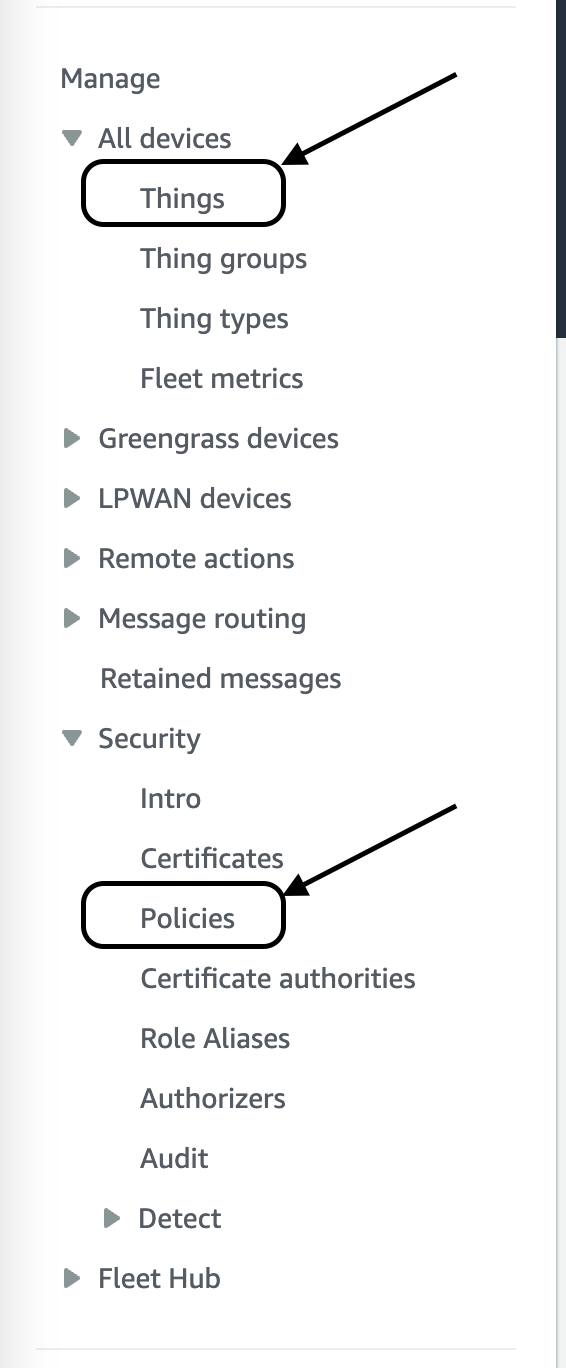
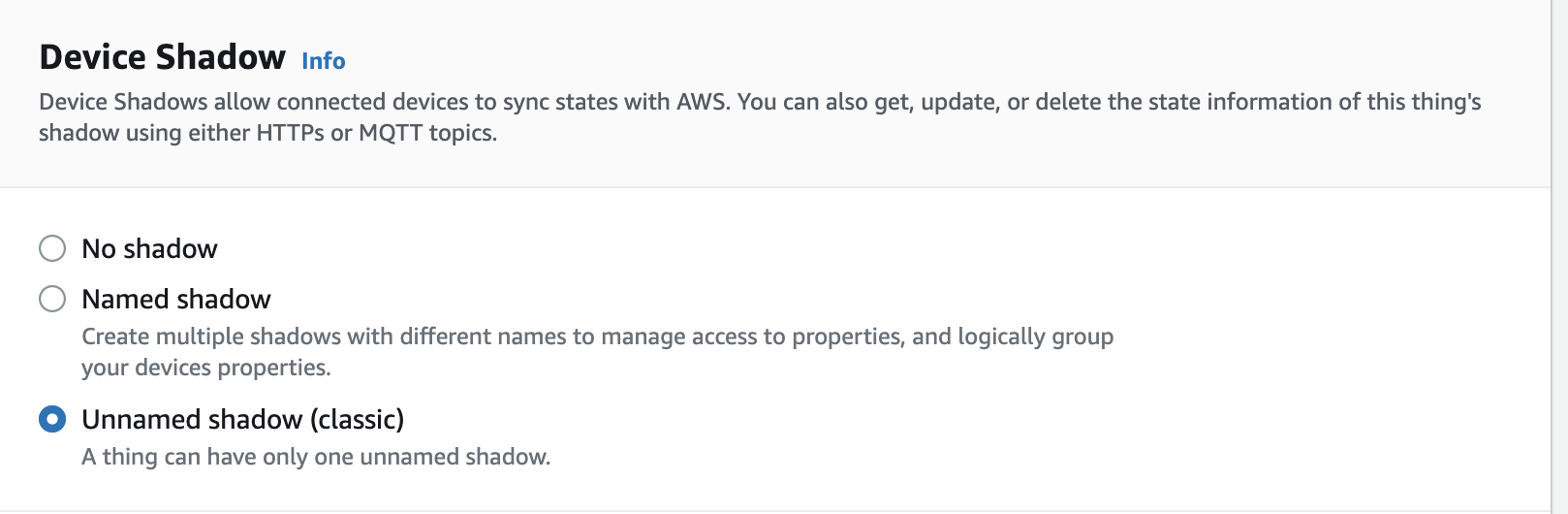
source paho-mqtt/bin/activate

pip install paho-mqtt

# Create the necessary infrastructure and code update

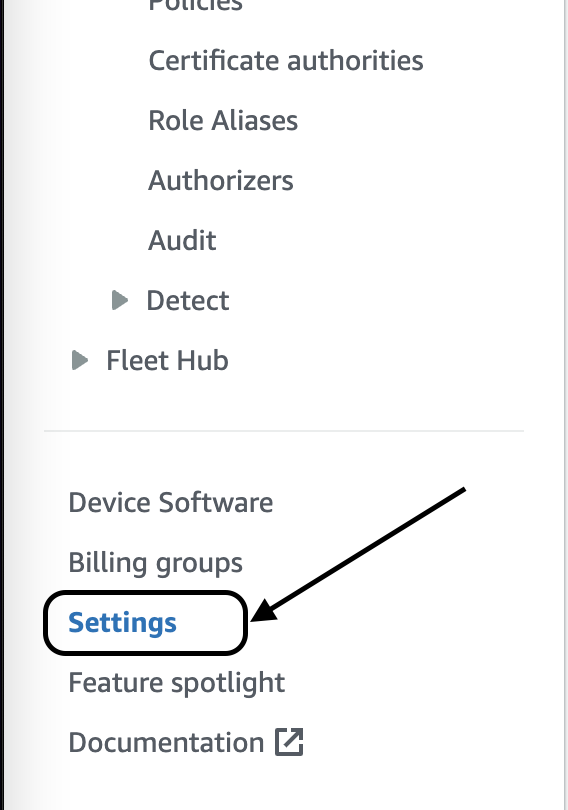
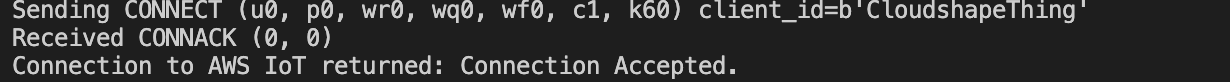
## Create an IoT Thing and Shadow in AWS IoT Core -1p

In the communication between the script and AWS IoT Core, the physical Thing is your laptop.

* Login to your AWS account
* Create an IoT shadow that will match the physical device that sends data to the cloud. Follow the tutorial from [AWS docs / Create IoT Thing](https://docs.aws.amazon.com/iot/latest/developerguide/create-iot-resources.html) (**Note**: AWS IoT is actually AWS IoT Core and the home page is <https://console.aws.amazon.com/iot/home> )
  + you can ignore the naming used in the tutorial and use names of your choice (e.g. *CloudshapeThing*)
  + create the policy and then the thing object with the created policy
  + **Note**: the menu on the left holds the policies and things links:
  + **Note**: select the unnamed shadow option:
* Go to Downloads folder on your computer and copy all the certificates downloaded and move them to the certs folder from the repo.

## Connect to the Thing endpoint from the script -1p

This section is about working on the connect phase of the MQTT communication, from the local script to AWS IoT Core.

* In the mqtt\_service.py module instantiate an MQTT client. See the [constructor](https://pypi.org/project/paho-mqtt/#constructor-reinitialise) example in the documentation.
* Update the cert\_filename and the private\_key\_filename variables with the correct values
* Complete the setup()method
  + Call the tls\_set method of the MQTT client object to setup the certificate related files. Use relative paths for the CA filename, the certificate filename and the private key filename.
  + Register on\_connect and on\_disconnect callbacks, example on the same documentation page.
  + Register also the on\_log method
* Call the connect method of the MQTT client object using as host the Thing endpoint from the AWS Console and the port 8883, see [Connect](https://pypi.org/project/paho-mqtt/#connect-reconnect-disconnect) example.
  + Go in the AWS Console to Manage > Settings > Device data endpoint to find the Thing’s API host endpoint that we are going to use later.
* Lastly, call the loop\_start() method of the MQTT client object - this will setup an event loop manager that will invoke the callbacks when needed.
* Run python script.py to test that the connect succeeds.
  + The console logging should tell that the connection was accepted.
  + Go to AWS Console > IoT Core and click on the Monitor tab. You should see the connections being shown on the graph.

## Publish a message to the Thing on a specific topic -1p

This section is about creating the sample messages and publishing them to the cloud.

* In mqtt\_service.py complete the publish\_message method that will send a payload to a specific topic. See the documentation for the publish method.
  + Complete the execute method from the script.py module by calling the mqtt\_service.py publish\_message method.
  + Use the topic base name ‘connectedcar/telemetry’

**Note**: The message generated is a vehicle telemetry object. The id represents the id of the message and is unique, while vin is the vehicle identifier, it can be a harcoded uuid for now.

**Note**: If you don’t see any message but the publish is marked as success use qos=1 in the publish method call

**Note**: You might also need to add a delay between the setup call and the publish call

* Run python script.py to test that the connection succeeds.
  + The console logging should tell that the messages were published.
  + Go to the AWS Console > IoT Core and click on the Monitor tab. You see the number of messages received in the grapth.

## Create a SNS topic -1p

Following the next steps we'll attach a rule (an action) for each time a device sends data to the cloud. Our action is to send an SMS using a AWS SNS notification topic.

* Create a role
  + Go to the AWS Console and select IAM
  + Go to Roles from the left hand side bar and click on Create Role
  + On the section "Choose the service that will use this role" select "IoT". The "next" until the end. On the "Role name" field choose IoTWorkshopRole
* Create a SNS topic
  + From the AWS console go to SNS
  + Click on topics
  + Click on "Create new topic" and create a topic called IoTWorkshopTopic. Copy the arn of the

topic, we will needed later.

**Note**: Create a Standard topic

* Click on subscriptions and then on "Create new subscription"

On the "Topic arn" field paste the arn of the topic created earlier. On the "protocol" field select

SMS. On the "endpoint" add your phone number with the prefix "+4".

**Note**: You will also have to confirm your number before adding it, just follow the wizard

## Create IoT Rule for SNS -1p

Create the IoT rule:

1. On the AWS console go to IoT Core
2. Select " Message routing" from the left hand sidebar
3. Click on “Rules”
4. Click on “Create rule”
5. On the "Rule name" field add the name IoTSNSRule. Give whatever description you want.
6. Click “Next”
7. Add the SQL statement as described in the example listed on the page to create the filter rule
   1. use the name of the MQTT topic used in the code as the filter field
8. Click “Next”
9. In the “Action1” sections
10. Select from the list Send a message as an SNS push notification.
11. In the “SNS topic” section select the target called IoTWorkshopTopic
12. On the IAM role name choose the role created earlier IoTWorkshopRole.
13. Click “Next”
14. Validate the information
15. Click Create

Now you can run again your script.

## 