



POLITECNICO
MILANO 1863



ITECNICO
IANO 1832

AWS IoT

Parte prima

Marco D. Santambrogio – marco.santambrogio@polimi.it

Emanuele Del Sozzo – emanuele.delsozzo@polimi.it

Lorenzo Di Tucci – lorenzo.ditucci@mail.polimi.it

Giuseppe Natale – giuseppe.natale@polimi.it

Marco Rabozzi – marco.rabozzi@polimi.it

Alberto Scolari – alberto.scolari@polimi.it

Matteo Ferroni – matteo.ferroni@polimi.it

Ver. aggiornata al 18/12/2016

Outline

- **Registration**
- **The AWS Command Line Interface**
- **Setup your first “thing”**
- **Connect, publish and subscribe using an MQTT client**
- **Connect, publish and subscribe using the Python SDK**
- **MQTT over WebSocket**



Registration

<https://aws.amazon.com/it/iot/>



Internet of Things (IoT) è un termine coniato da Kevin Ashton, un pioniere tecnologico inglese che lavorando sull'identificazione a radiofrequenza (RFID) concepì un sistema di sensori universali che collegavano il mondo fisico a Internet. Sebbene le cose, Internet e la connettività siano i tre componenti di base dell'IoT, il valore è dato dal divario colmato tra il mondo fisico e quello digitale nei sistemi che si migliorano e consolidano in maniera autonoma.



Registration

<https://aws.amazon.com/it/iot/>




Credenziali per il login

Utilizza il modulo sottostante per creare le credenziali di accesso da utilizzare per AWS e Amazon.com.

| | |
|---|---|
| Il mio nome è: | <input type="text" value="Matteo Ferroni"/> |
| Il mio indirizzo e-mail è: | <input type="text" value="matteo.ferroni@polimi.it"/> |
| Inserisci di nuovo per confermare: | <input type="text" value="matteo.ferroni@polimi.it"/> |
| Inserisci una nuova password: | <input type="password" value="....."/> |
| Inserisci di nuovo per confermare: | <input type="password" value="....."/> |
| <input type="button" value="Crea account"/> | |



Registration

- Potete crearvi un account personale per iniziare a sperimentare le possibilità di AWS IoT
- Il giorno della hackathon, avrete a disposizione degli account illimitati, cortesemente offerti da 



Italiano  Esci

Accesso ad Amazon Web Services

Informazioni di contatto

☐ Account aziendale ☒ Account personale

** Campi obbligatori*

Nome e cognome* Matteo Ferroni

Paese* Italia



Registration

Perché devo fornire le informazioni di pagamento?

Chiediamo una carta di credito o di debito per agevolare il passaggio all'utilizzo dei servizi AWS a pagamento qualora l'account superi i limiti del piano gratuito di AWS. Inoltre, utilizziamo le informazioni di pagamento per verificare l'autenticità del tuo account e per prevenire attività fraudolente.



Italiano Esci

Accesso ad Amazon Web Services



Informazioni di pagamento

Inserisci le informazioni di pagamento di seguito. Potrai provare gratuitamente un'ampia gamma di prodotti AWS tramite il piano gratuito. Sulla carta di credito o debito verrà addebitato solo l'importo relativo all'utilizzo che non rientra nel piano gratuito.

» [Domande frequenti](#)

Numero della carta di credito/debito

Data di scadenza



Registration



Italiano [Esci](#)

Accesso ad Amazon Web Services



Verifica dell'identità

Un sistema automatizzato effettuerà immediatamente la chiamata e chiederà di inserire il numero PIN fornito.

1. Fornire un numero di telefono ✓

2. Chiamata in corso

Seguire le istruzioni sul telefono e digitare il seguente numero identificativo personale (PIN) sul telefono.

PIN:

Se non si è ancora ricevuta la chiamata al numero indicato sopra, attendere. Questa pagina si aggiornerà automaticamente con le azioni successive da eseguire.

3. Verifica dell'identità completata



Registration



Italiano [Esci](#)

Accesso ad Amazon Web Services

Informazioni di contatto

Informazioni di pagamento

Verifica dell'identità

Piano di supporto

Conferma

Piano di supporto

AWS Support offre una serie di piani per soddisfare le tue esigenze. Tutti i piani forniscono accesso 24 ore su 24, 7 giorni su 7 al servizio clienti, alla documentazione di AWS, ai whitepaper e ai forum di supporto. Per l'accesso al supporto tecnico e a risorse aggiuntive che aiutano a pianificare, distribuire e ottimizzare l'ambiente AWS, ti consigliamo di scegliere il piano di supporto che più si allinea al tuo utilizzo di AWS.

Selezionare un'opzione

☒ **Basic**

Descrizione: servizio clienti in caso di domande su account o fatturazione e accesso ai forum della community AWS.

Prezzo: incluso

☐ **Developer**

Caso d'uso: prova di AWS

Descrizione: il contatto principale può porre domande tecniche tramite il Centro di supporto e ricevere risposte entro 12 - 24 ore durante l'orario lavorativo locale.

Prezzo: a partire da 29 USD/mese (varia in base all'utilizzo)

☐ **Business**

Caso d'uso: utilizzo di AWS in fase di produzione



Registration

<https://aws.amazon.com/it/free/>

Il piano gratuito di Amazon Web Services (AWS) è stato ideato per consentirti di acquisire esperienza diretta dei servizi cloud di AWS. Il piano gratuito di AWS consente di usare una serie di servizi per 12 mesi a partire dalla data di registrazione più altre offerte che non scadono automaticamente dopo i 12 mesi previsti dai termini del piano gratuito.

Your AWS Account is Ready - Get Started Now



Amazon Web Services a matteo.ferroni@polimi.it

19:05

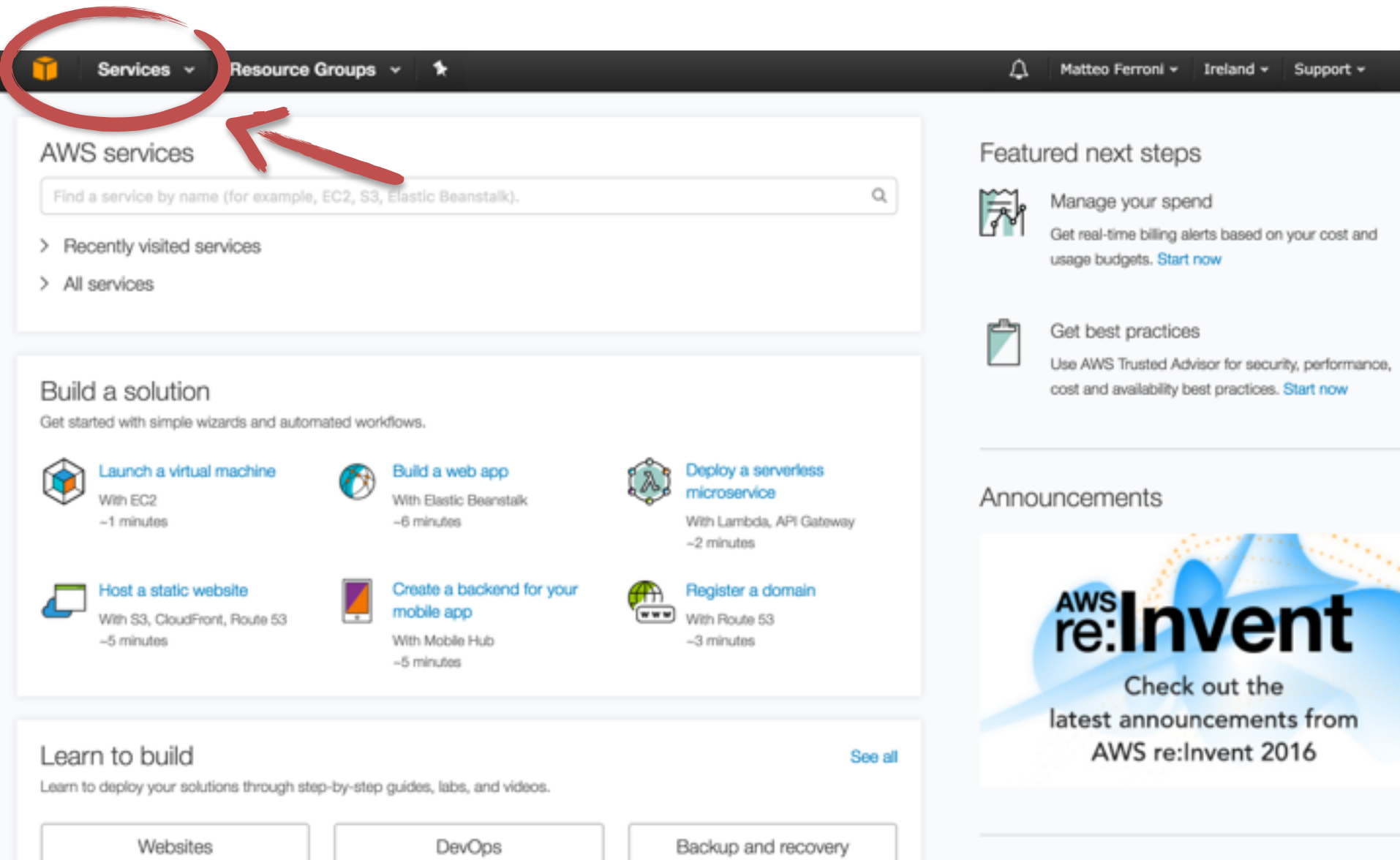


Your AWS Account is Ready

For the next 12 months, you'll have free access to core AWS compute, storage, database, and application services within the limits of the [Free Tier](#).



Services



Services ▾ Resource Groups ▾ ⭐


AWS services


Find a service by name (for example, EC2, S3, Elastic Beanstalk). 🔍


- > Recently visited services
- > All services


Build a solution


Get started with simple wizards and automated workflows.


 **Launch a virtual machine**
With EC2
~1 minutes

 **Build a web app**
With Elastic Beanstalk
~6 minutes

 **Deploy a serverless microservice**
With Lambda, API Gateway
~2 minutes

 **Host a static website**
With S3, CloudFront, Route 53
~5 minutes

 **Create a backend for your mobile app**
With Mobile Hub
~5 minutes

 **Register a domain**
With Route 53
~3 minutes

Learn to build [See all](#)


Learn to deploy your solutions through step-by-step guides, labs, and videos.


Websites

DevOps


Backup and recovery

Featured next steps

 **Manage your spend**
Get real-time billing alerts based on your cost and usage budgets. [Start now](#)

 **Get best practices**
Use AWS Trusted Advisor for security, performance, cost and availability best practices. [Start now](#)

Announcements


AWS re:Invent
Check out the latest announcements from AWS re:Invent 2016



Services

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with 'Services' selected, 'Resource Groups', and user information. On the left, a 'History' sidebar lists 'Console Home', 'IAM', 'CloudWatch', 'Kinesis', 'EMR', and 'Billing'. The main area displays a grid of service categories: Compute (EC2, EC2 Container Service, Lightsail, Elastic Beanstalk, Lambda), Developer Tools (CodeCommit, CodeBuild, CodeDeploy, CodePipeline), Analytics (Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, Data Pipeline, QuickSight), Application Services (Step Functions, SWF, API Gateway, AppStream, Elastic Transcoder), Storage (S3, Elastic File System, Glacier, Storage Gateway), Management Tools (CloudWatch, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, Trusted Advisor, Managed Services), Artificial Intelligence (Lex, Polly, Rekognition, Machine Learning), Messaging (SQS, SNS, SES), Database (RDS, DynamoDB, ElastiCache, Redshift), Business Productivity (WorkDocs, WorkMail), Security, Identity & Compliance (IAM, Inspector, Certificate Manager, Directory Service, WAF & Shield, Compliance Reports), Networking & Content Delivery (VPC, CloudFront, Direct Connect, Route 53), Internet Of Things (AWS IoT), Game Development (GameLift), Mobile Services (Mobile Hub, Cognito), and Desktop & App Streaming (WorkSpaces, AppStream 2.0). The 'Internet Of Things' category is circled in red, and a red arrow points to the 'AWS IoT' service.

Services

Resource Groups

Search services

Group A-Z

History

Console Home

IAM

CloudWatch

Kinesis

EMR

Billing

Compute

EC2

EC2 Container Service

Lightsail

Elastic Beanstalk

Lambda

Developer Tools

CodeCommit

CodeBuild

CodeDeploy

CodePipeline

Analytics

Athena

EMR

CloudSearch

Elasticsearch Service

Kinesis

Data Pipeline

QuickSight

Application Services

Step Functions

SWF

API Gateway

AppStream

Elastic Transcoder

Storage

S3

Elastic File System

Glacier

Storage Gateway

Management Tools

CloudWatch

CloudFormation

CloudTrail

Config

OpsWorks

Service Catalog

Trusted Advisor

Managed Services

Artificial Intelligence

Lex

Polly

Rekognition

Machine Learning

Messaging

SQS

SNS

SES

Database

RDS

DynamoDB

ElastiCache

Redshift

Business Productivity

WorkDocs

WorkMail

Security, Identity & Compliance

IAM

Inspector

Certificate Manager

Directory Service

WAF & Shield

Compliance Reports

Networking & Content Delivery

VPC

CloudFront

Direct Connect

Route 53

Internet Of Things

AWS IoT

Game Development

GameLift

Mobile Services

Mobile Hub

Cognito

Desktop & App Streaming

WorkSpaces

AppStream 2.0



AWS IoT - Useful references

<https://console.aws.amazon.com/>

Console di gestione AWS

<http://docs.aws.amazon.com/iot/latest/developerguide/iot-sdks.html>

AWS IoT SDKs

<https://github.com/aws/aws-iot-device-sdk-python>

AWS IoT SDK for Python

<https://github.com/dwyl/learn-aws-iot>

Learn how to use Amazon Web Services Internet of Things (IoT) service to build connected applications



Installing the AWS Command Line Interface

<http://docs.aws.amazon.com/cli/latest/userguide/installing.html>

*The **AWS Command Line Interface** is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.*

Choose an Installation Method

There are a number of different ways to install the AWS CLI on your machine, depending on what operating system and environment you are using:

- On Microsoft Windows – use the [MSI installer](#).
- On Linux, OS X, or Unix – use [pip](#) (a package manager for Python software) or install manually with the [bundled installer](#).

Note

On OS X, if you see an error regarding the version of six that came with distutils in El Capitan, use the `--ignore-installed` option:

```
$ sudo pip install awscli --ignore-installed six
```



Access key ID and secret access key

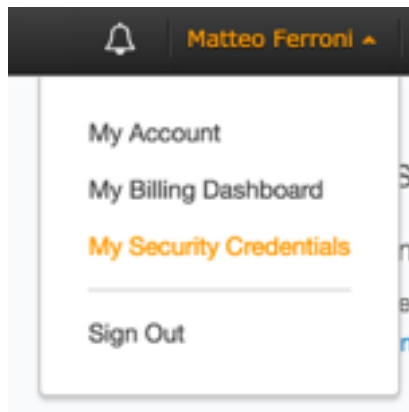
<https://console.aws.amazon.com/iam/home?#/home>

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them by using the AWS Management Console. We recommend that you use IAM access keys instead of AWS root account access keys. IAM lets you securely control access to AWS services and resources in your AWS account.



Access key ID and secret access key

<https://console.aws.amazon.com/iam/home?#/home>

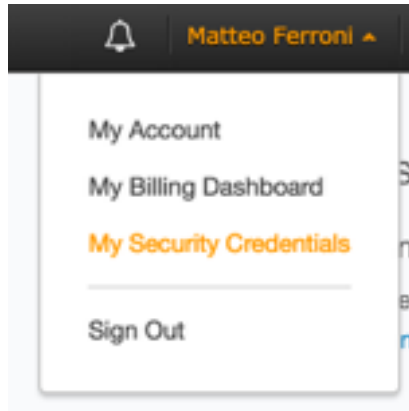


Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them by using the AWS Management Console. We recommend that you use IAM access keys instead of AWS root account access keys. IAM lets you securely control access to AWS services and resources in your AWS account.

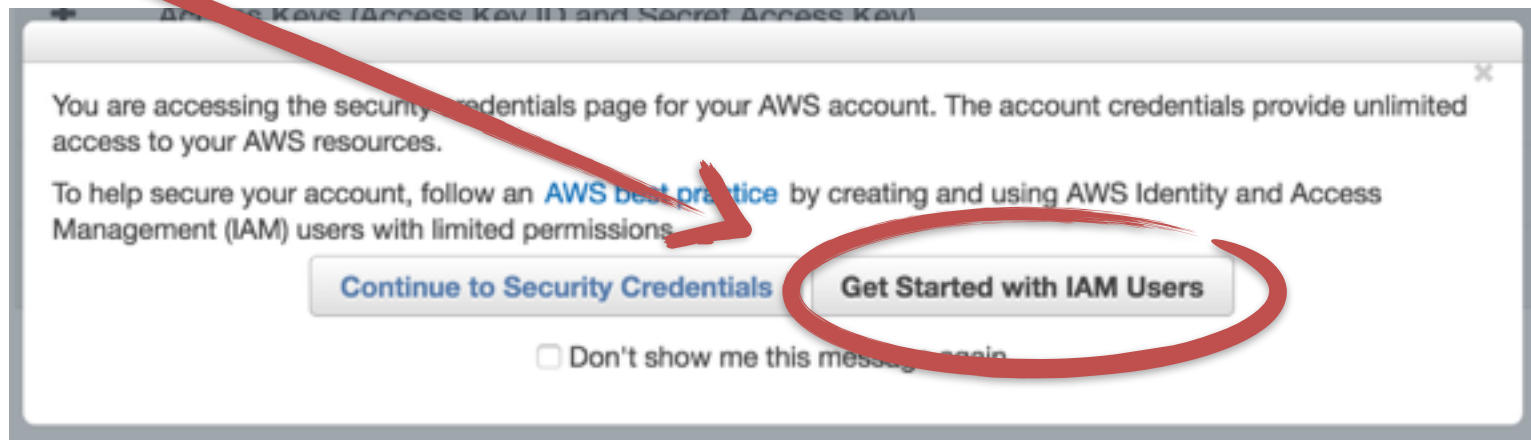


Access key ID and secret access key

<https://console.aws.amazon.com/iam/home?#/home>



Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them by using the AWS Management Console. We recommend that you use IAM access keys instead of AWS root account access keys. IAM lets you securely control access to AWS services and resources in your AWS account.



Access key ID and secret access key

1. Open the **IAM console**.
2. In the navigation pane, choose **Users**.
3. Choose your IAM user name (not the check box).
4. Choose the **Security Credentials** tab and then choose **Create Access Key**.
5. To see your access key, choose **Show User Security Credentials**. Your credentials will look something like this:
 - Access Key ID: AKIAIOSFODNN7EXAMPLE
 - Secret Access Key: wjaIrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
6. Choose **Download Credentials**, and store the keys in a secure location.

Your secret key will no longer be available through the AWS Management Console; you will have the only copy. Keep it confidential in order to protect your account, and never email it. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.



Add user - Details

Add user



Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*



Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.



AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.



Add user - Permissions

Add user

1

Details

2

Permissions

3

Review

4

Complete

Set permissions for aws-cli



Add user to group



Copy permissions from
existing user



Attach existing policies
directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Create group

Refresh

Q Search

Showing 1 result

Group ▼

Attached policies

☒ IoT-full-access AWSIoTFullAccess



Add user - Create group

Create group







Group name

Create policy

Refresh

Filter: Policy type

Showing 6 results

| | | Policy name | Type | Attachments | Description |
|-------------------------------------|---|--|-------------|-------------|---|
| <input type="checkbox"/> | ▶ |  AWSIoTLogging | AWS managed | 0 | Allows creation of Amazon CloudWatch Log groups an... |
| <input type="checkbox"/> | ▶ |  AWSIoTConfigAccess | AWS managed | 0 | This policy gives full access to the AWS IoT configurati... |
| <input type="checkbox"/> | ▶ |  AWSIoTRuleActions | AWS managed | 0 | Allows access to all AWS services supported in AWS lo... |
| <input type="checkbox"/> | ▶ |  AWSIoTConfigReadOnlyA... | AWS managed | 0 | This policy gives read only access to the AWS IoT confi... |
| <input type="checkbox"/> | ▶ |  AWSIoTDataAccess | AWS managed | 0 | This policy gives full access to the AWS IoT messaging... |
| <input checked="" type="checkbox"/> | ▶ |  AWSIoTFullAccess | AWS managed | 0 | This policy gives full access to the AWS IoT configurati... |

Cancel

Create group



Add user - Completed

Add user



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://631211024482.signin.aws.amazon.com/console>

 Download .csv

User

▶  aws-cli

Access key ID

AKIAJAIISGT3QJOMER4SQ

Secret access key

***** [Show](#)

Close



Configuring the AWS Command Line Interface

<http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html>

Regione (suggerita): ***eu-west-1***

Output (suggerito): ***json***

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: ENTER
```



Using the AWS Command Line Interface

Thing

- **Create a “thing”**

```
aws iot create-thing  
  --thing-name "TestDevice1"
```

Response:

```
{  
  "thingArn":  
    "arn:aws:iot:eu-west-1:631211024482:thing/  
    TestDevice1",  
  "thingName":  
    "TestDevice1"  
}
```



Using the AWS Command Line Interface

Thing

- **Generate a certificate**

```
aws iot create-keys-and-certificate
--set-as-active
--certificate-pem-outfile cert.pem
--public-key-outfile publicKey.pem
--private-key-outfile privateKey.pem
```

Certificate

Response:

```
{
  "certificateArn":
    "arn:aws:iot:eu-west-1:631211024482:cert/
    b0a162f3f5bfc207346e598a706c098db4003fa4330b5c1bddfd08f512
    987f31",
  "certificateId":
    "b0a162f3f5bfc207346e598a706c098db4003fa4330b5c1bddfd08f51
    2987f31",
  ...
}
```



Using the AWS Command Line Interface

- **Create a policy file: *policy.json***

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Action": ["iot:*"],  
    "Resource": ["*"]  
  }]  
}
```

Thing

Certificate

- **And load it in AWS**

```
aws iot create-policy  
  --policy-name "first-policy"  
  --policy-document file://PATH-TO-FILE
```

Policy



Using the AWS Command Line Interface

Thing

Certificate

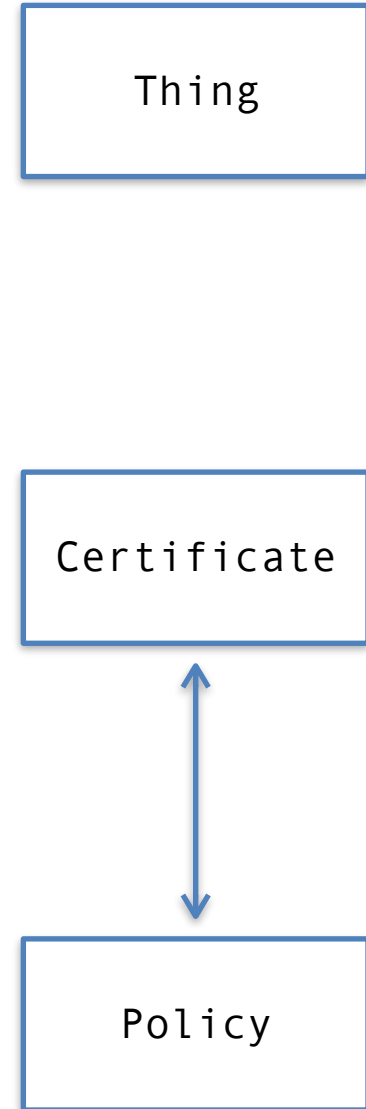
Policy



Using the AWS Command Line Interface

- **Link certificate and policy**

```
aws iot attach-principal-policy  
  --principal "ARN-CERTIFICATE"  
  --policy-name "first-policy"
```



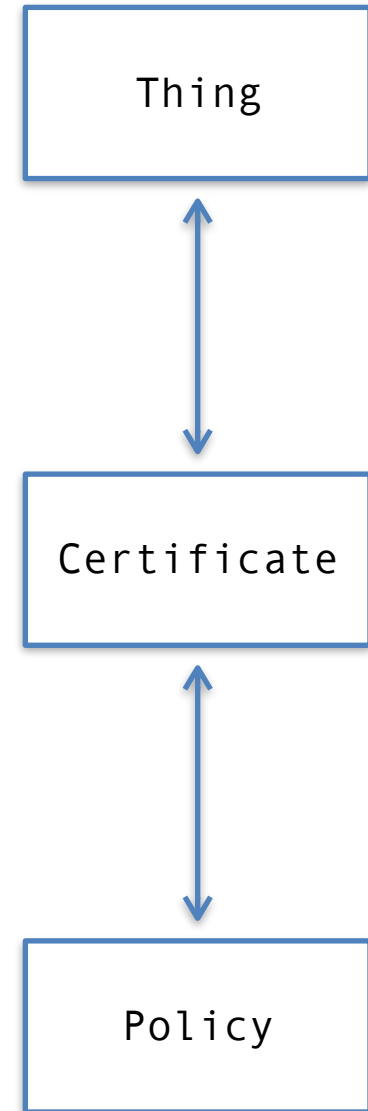
Using the AWS Command Line Interface

- **Link certificate and policy**

```
aws iot attach-principal-policy  
  --principal "ARN-CERTIFICATE"  
  --policy-name "first-policy"
```

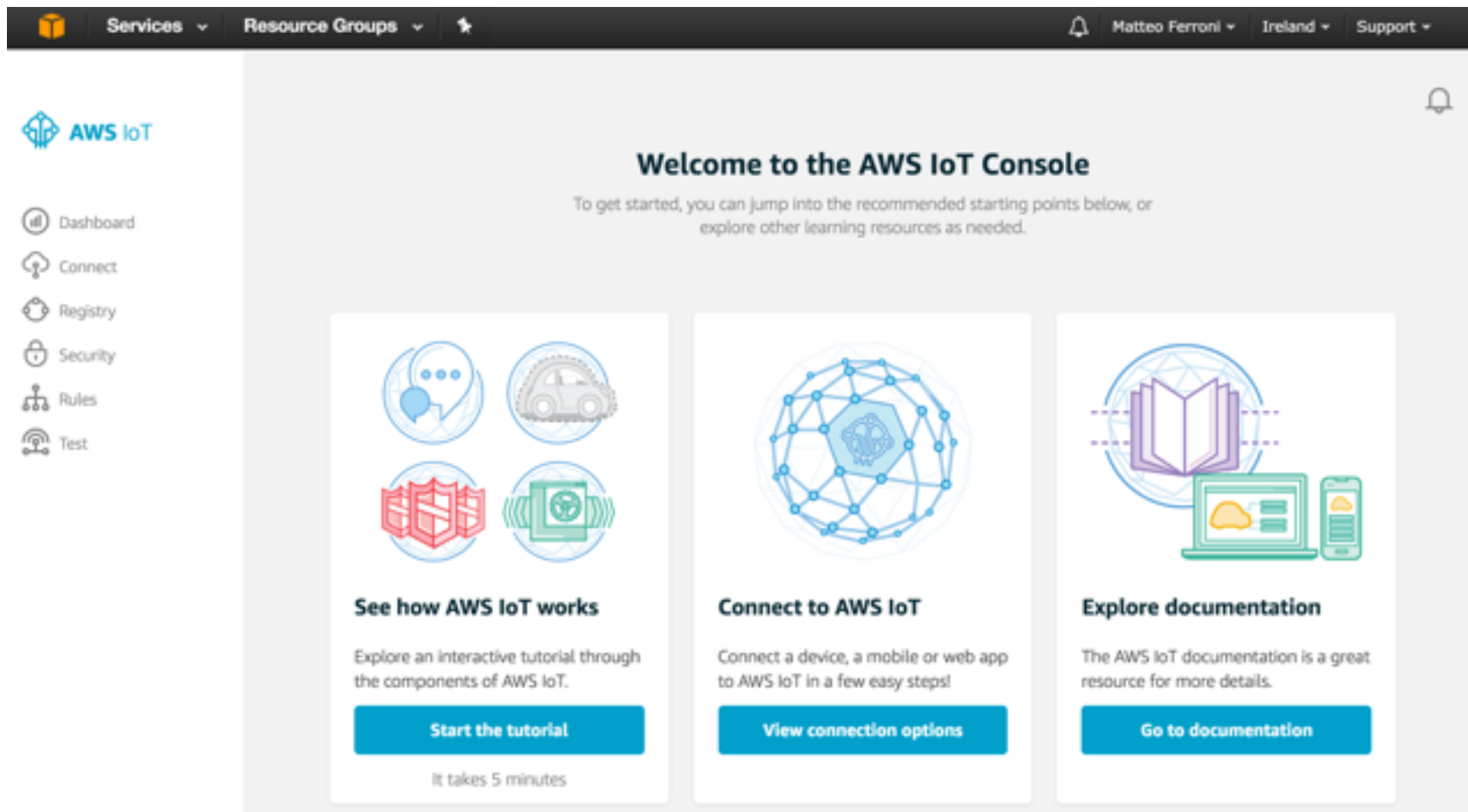
- **Link certificate and “thing”:**

```
aws iot attach-thing-principal  
  --thing-name "TestDevice1"  
  --principal "ARN-CERTIFICATE"
```



AWS IoT - Dashboard

Le stesse operazioni possono essere fatte da UI web



The screenshot shows the AWS IoT Console interface. At the top, there's a navigation bar with 'Services', 'Resource Groups', and a user profile 'Matteo Ferroni' in 'Ireland'. On the left, a sidebar lists navigation options: Dashboard, Connect, Registry, Security, Rules, and Test. The main content area is titled 'Welcome to the AWS IoT Console' and provides three recommended starting points:

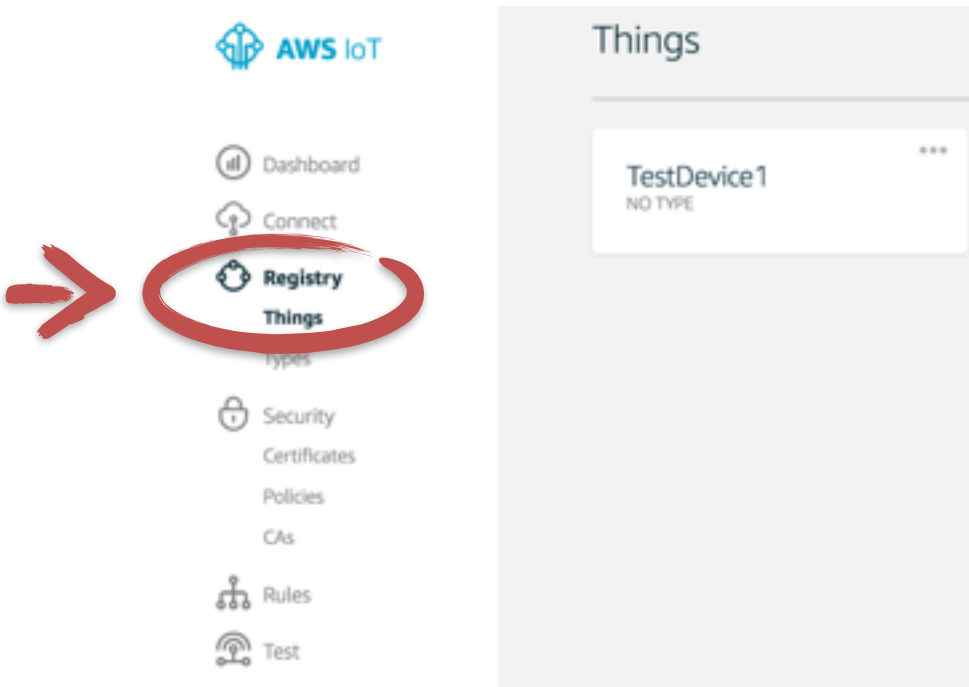
- See how AWS IoT works**: Explore an interactive tutorial through the components of AWS IoT. It takes 5 minutes. Button: [Start the tutorial](#).
- Connect to AWS IoT**: Connect a device, a mobile or web app to AWS IoT in a few easy steps! Button: [View connection options](#).
- Explore documentation**: The AWS IoT documentation is a great resource for more details. Button: [Go to documentation](#).



AWS IoT - Dashboard



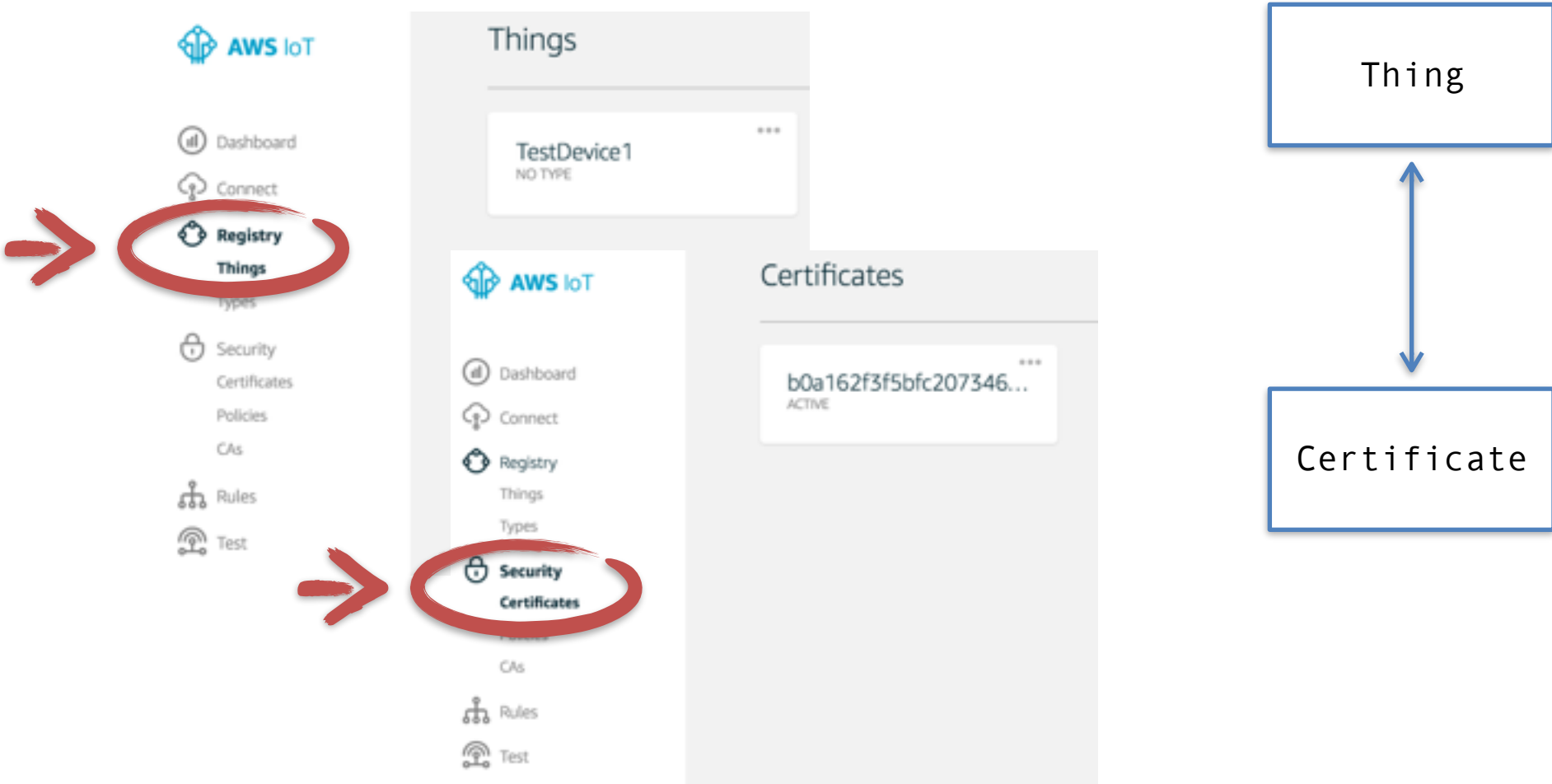
AWS IoT - Dashboard



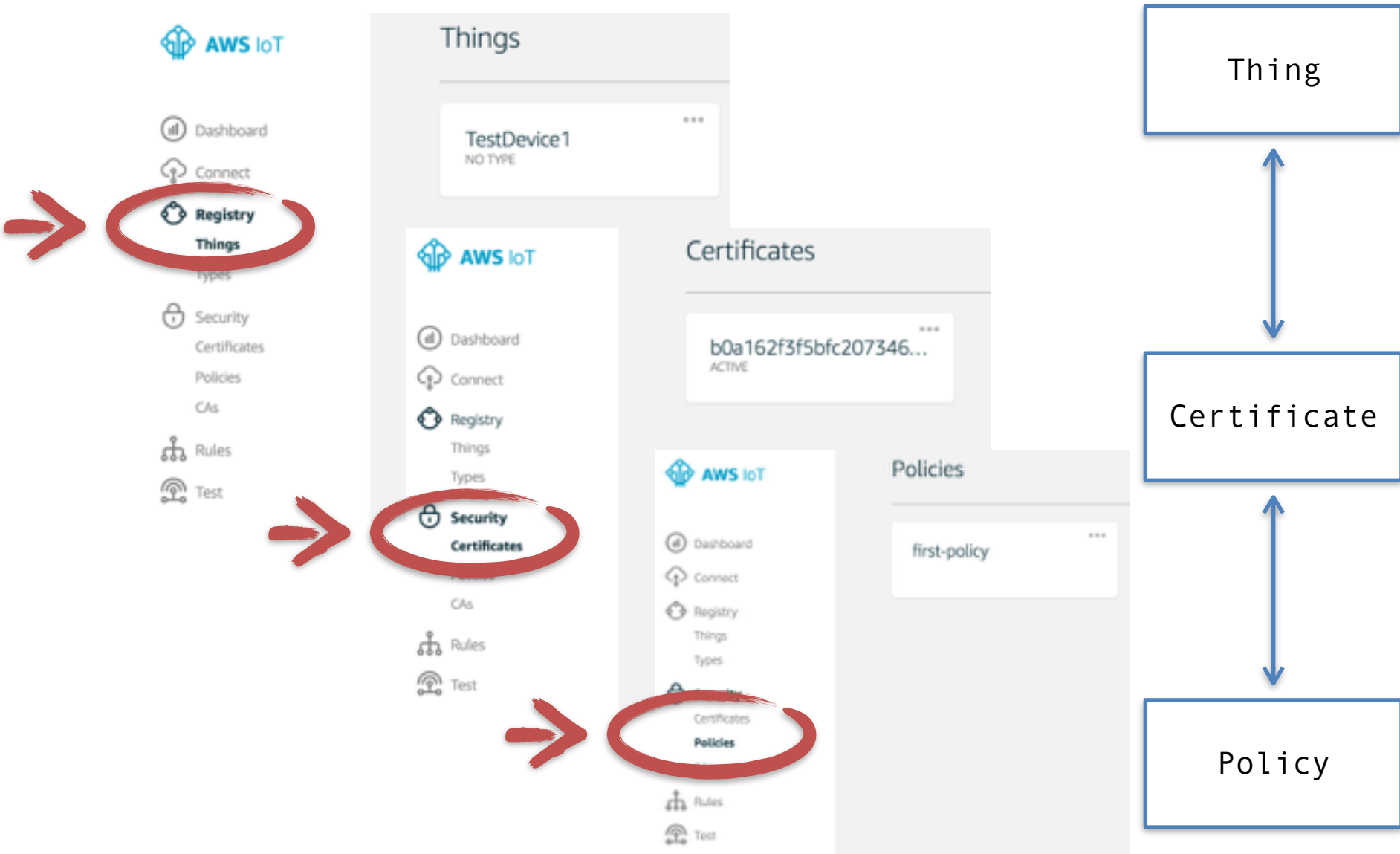
Thing



AWS IoT - Dashboard



AWS IoT - Dashboard



MQTT client example



- **Download *rootCA.pem* :**

`https://www.symantec.com/content/en/us/enterprise/verisign/roots/VeriSign-Class 3-Public-Primary-Certification-Authority-G5.pem`

- **Download *MQTT.fx* :**

`http://mqttfx.jfx4ee.org/index.php/download`

- **Trova l'indirizzo a cui connetterti:**

`aws iot describe-endpoint`

Response:

```
{  
  "endpointAddress":  
    "a1c2qwq6b794bg.iot.eu-west-1.amazonaws.com"  
}
```



MQTT client example



Connection Profile

Profile Name

Broker Address

Broker Port

Client ID

General User Credentials **SSL/TLS** Proxy Last Will and Testament

Enable SSL/TLS ☒

Protocol

☐ CA signed server certificate

☐ CA certificate file

☐ CA certificate keystore

☒ Self signed certificates

CA File

...

Client Certificate File

...

Client Key File

...

Client Key Password

PEM Formatted ☒

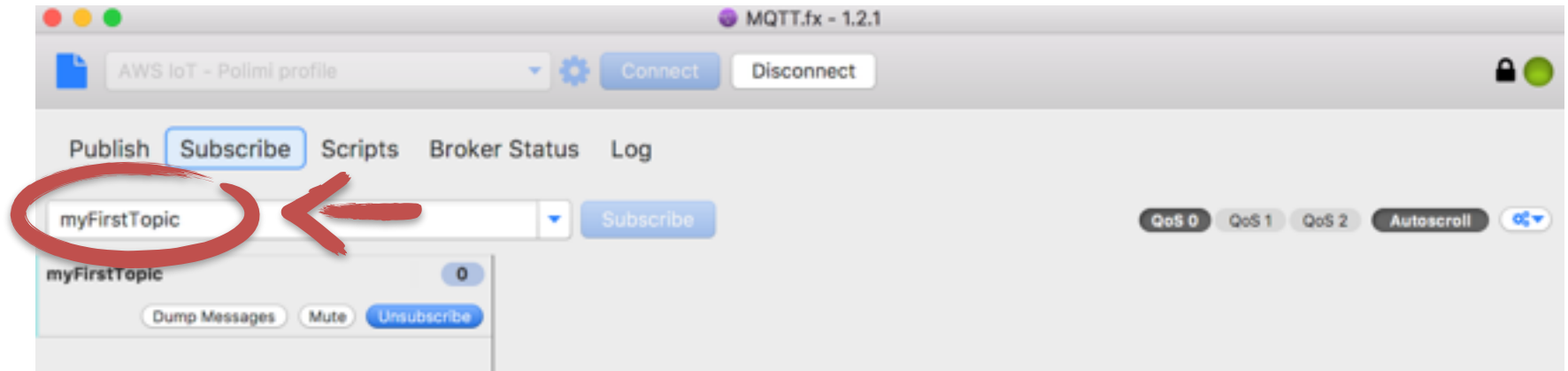
- Configura MQTT.fx usando l'indirizzo trovato e i certificati generati



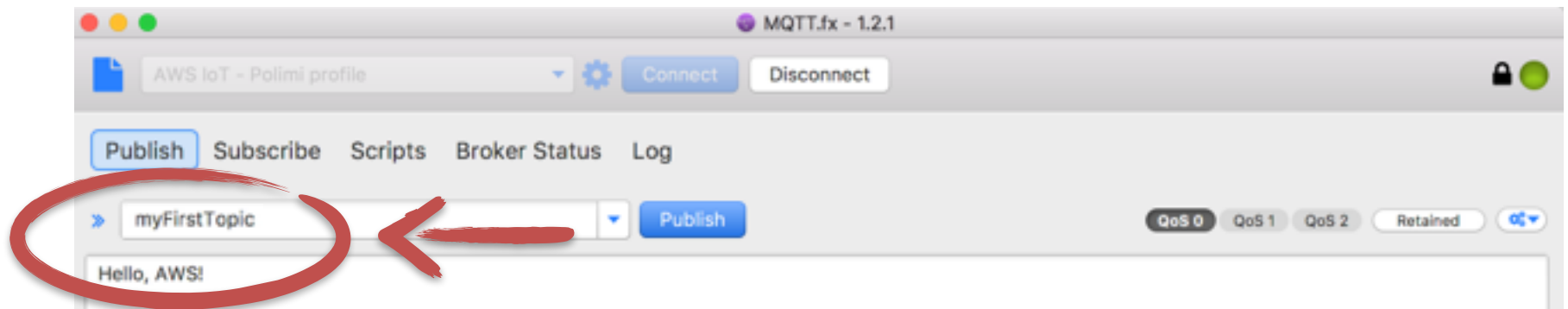
Connect, Publish and Subscribe



- Connect and subscribe



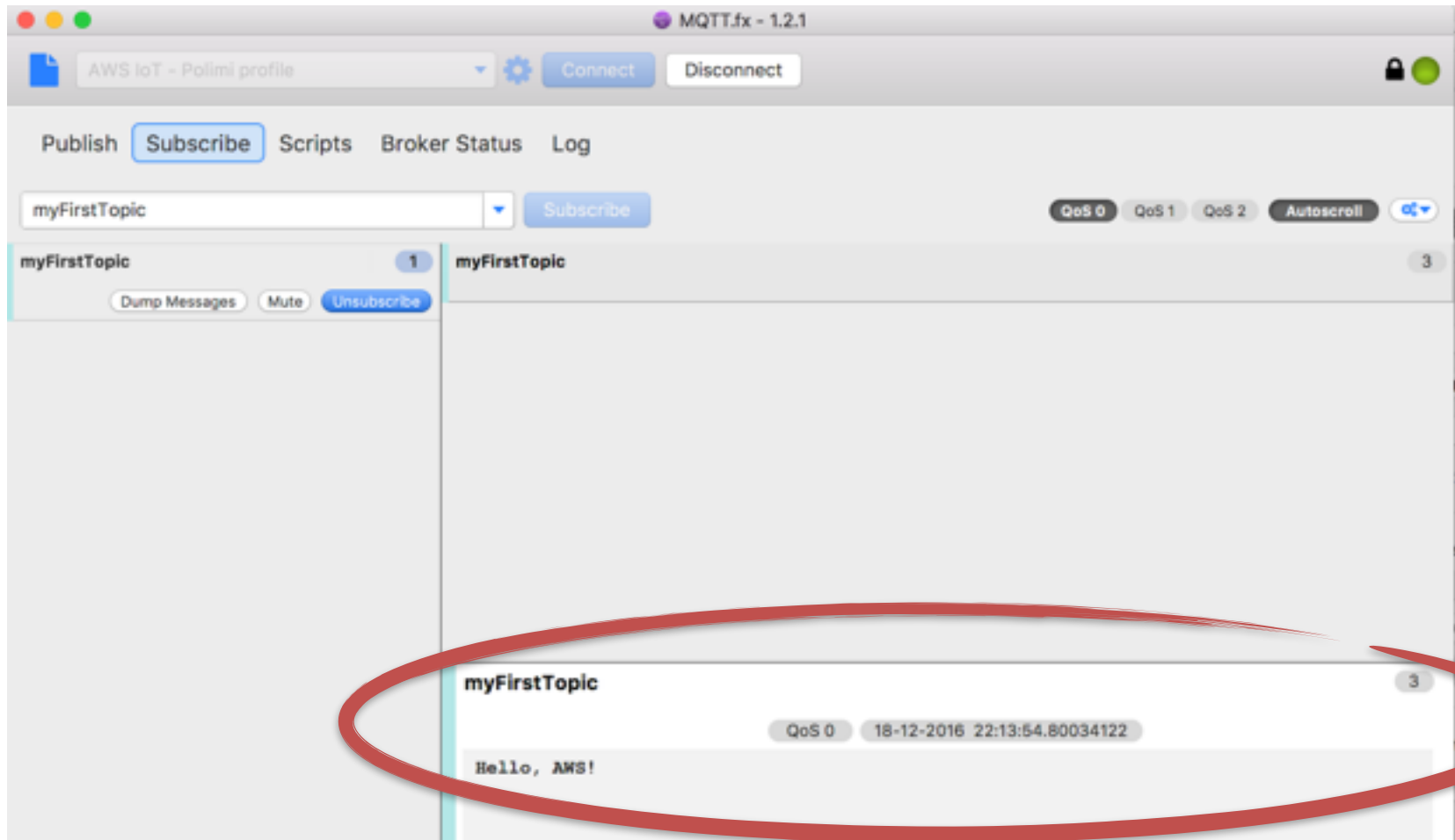
- Publish



Connect, Publish and Subscribe



- ...Message received!



Publish and subscribe using AWS Dashboard

The screenshot displays the AWS IoT MQTT client interface. At the top, the navigation bar includes 'Services', 'Resource Groups', 'AWS IoT', and user information. The left sidebar lists navigation options: Dashboard, Connect, Registry, Security, Rules, Test, Settings, and Learn. The main content area is titled 'MQTT client' and shows the user is 'Connected as iotconsole-1482095854968-3'.

The interface is divided into two main sections: 'Subscriptions' and 'Publish'.

Subscriptions Section:

- Subscribe to a topic:** This section allows users to subscribe to a specific topic. It includes a text input field for the 'Subscription topic' (currently containing 'myFirstTopic'), a 'Max message capture' input (set to 100), and a 'Quality of Service' selector (set to 0). A blue 'Subscribe to topic' button is located at the bottom of this section.
- Subscriptions list:** A list of existing subscriptions is shown, including 'myFirstTopic' with a red dot icon and a close button (X).

Publish Section:

- Publish:** This section allows users to publish a message to a specific topic. It includes a text input field for the topic (currently containing 'myFirstTopic') and a blue 'Publish to topic' button.
- Message input:** A text area for entering the message content. The example message shown is:

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

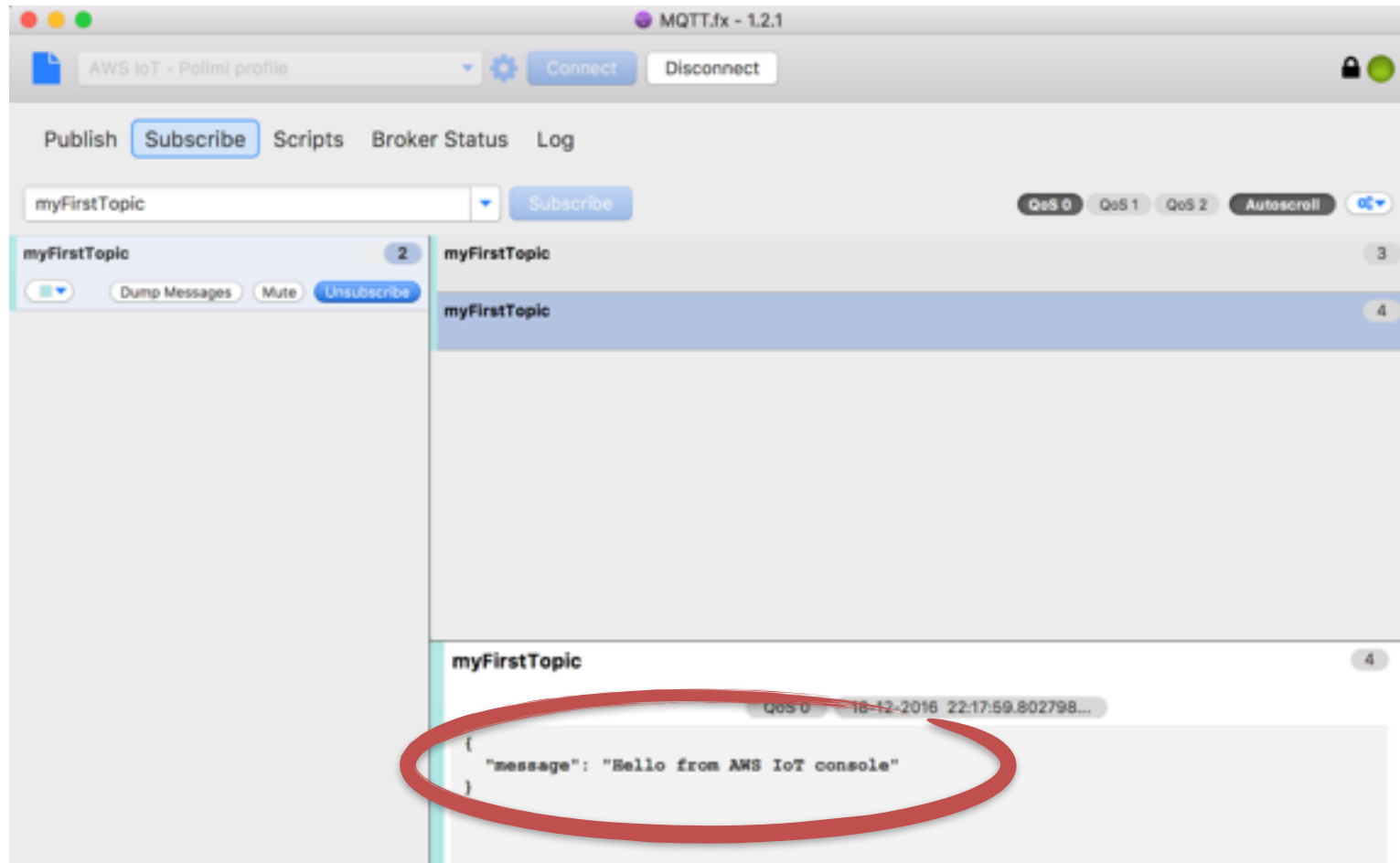
Red circles highlight the 'Publish to topic' button and the message input area in the 'Publish' section.



From the dashboard to MQTT.fx



- ...Message received!



The Python SDK - Install

<https://github.com/aws/aws-iot-device-sdk-python>

Install from pip

```
pip install AWSIoTPythonSDK
```

Build from source

```
git clone https://github.com/aws/aws-iot-device-sdk-python.git
cd aws-iot-device-sdk-python
python setup.py install
```

Download the zip file

The SDK zip file is available [here](#). Unzip the package and install the SDK like this:

```
python setup.py install
```



The Python SDK - Connection types

<https://github.com/aws/aws-iot-device-sdk-python>

Credentials

The SDK supports two types of credentials that correspond to the two connection types:

- X.509 certificate

For the certificate-based mutual authentication connection type. Download the [AWS IoT root CA](#). Use the AWS IoT console to create and download the certificate and private key. You must specify the location of these files when you initialize the client.

- IAM credentials

For the Websocket with Signature Version 4 authentication type. You will need IAM credentials: an access key ID, a secret access key, and an optional session token. You must also download the [AWS IoT root CA](#). You can specify the IAM credentials by:



Your first MQTT message using the Python SDK

```
# Import SDK packages
import ssl
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

# For certificate based connection
myMQTTClient = AWSIoTMQTTClient("TestDevice1", 4, False, True);

# Configurations
# For TLS mutual authentication
myMQTTClient.configureEndpoint("a1y3khtyn0ojiv.iot.eu-west-1.amazonaws.com", 8883)

myMQTTClient.configureCredentials("rootCA.pem", "privateKey.pem", "cert.pem")

myMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish queueing
myMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz
myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec
myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec

myMQTTClient.connect()
myMQTTClient.publish("myFirstTopic", "hello NECST!", 0)

myMQTTClient.unsubscribe("myFirstTopic")
myMQTTClient.disconnect()
```



The Polimi firewall...

Problema

- *sulle reti WiFi Polimi la **porta 8883 è chiusa!***

Soluzione per connessioni da laptop

- usa una VPN
(e.g., TunnelBear: <https://www.tunnelbear.com/>)



Soluzione per connessioni da scheda

- Usiamo **MQTT over WebSocket**
<http://docs.aws.amazon.com/iot/latest/developerguide/protocols.html#mqtt-ws>



MQTT over WebSocket - Credentials

<https://github.com/aws/aws-iot-device-sdk-python>

Credentials

The SDK supports two types of credentials that correspond to the two connection types:

- X.509 certificate

For the certificate-based mutual authentication connection type. Download the [AWS IoT root CA](#). Use the AWS IoT console to create and download the certificate and private key. You must specify the location of these files when you initialize the client.

- IAM credentials

For the WebSocket with Signature Version 4 authentication type. You will need IAM credentials: an access key ID, a secret access key, and an optional session token. You must also download the [AWS IoT root CA](#). You can specify the IAM credentials by:



MQTT over WebSocket - Credentials

- Creiamo un nuovo utente “my-board”

Add user



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://631211024482.signin.aws.amazon.com/console>

 Download .csv

| | User | Access key ID | Secret access key |
|---|------------|----------------------|---|
| ▶ | ✓ my-board | AKIAIHRT4777TIB23R6Q | RsXQsAnOS4MOV0Hq3bcLDF/tiAz6JsxP Hide |

Close



Your second MQTT message, over WebSocket

```
# Import SDK packages
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

# For Websocket connection
myMQTTClient = AWSIoTMQTTClient("my-board", useWebSocket=True)

# AWS IoT MQTT Client
myMQTTClient.configureIAMCredentials("ID", "SECRET")

# Configurations
# For Websocket
myMQTTClient.configureEndpoint("a1c2qwq6b794bg.iot.eu-west-1.amazonaws.com", 443)

# For Websocket, we only need to configure the root CA
myMQTTClient.configureCredentials("rootCA.pem")

myMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish queueing
myMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz
myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec
myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec

myMQTTClient.connect()
myMQTTClient.publish("myFirstTopic", "hello from my-board, I'm using a WebSocket!", 0)
# myMQTTClient.subscribe("test/topic", 1, customCallback)
myMQTTClient.unsubscribe("myFirstTopic")
myMQTTClient.disconnect()
```

