Atop Technologies, Inc

SE59XX Series Industrial Device Server Series

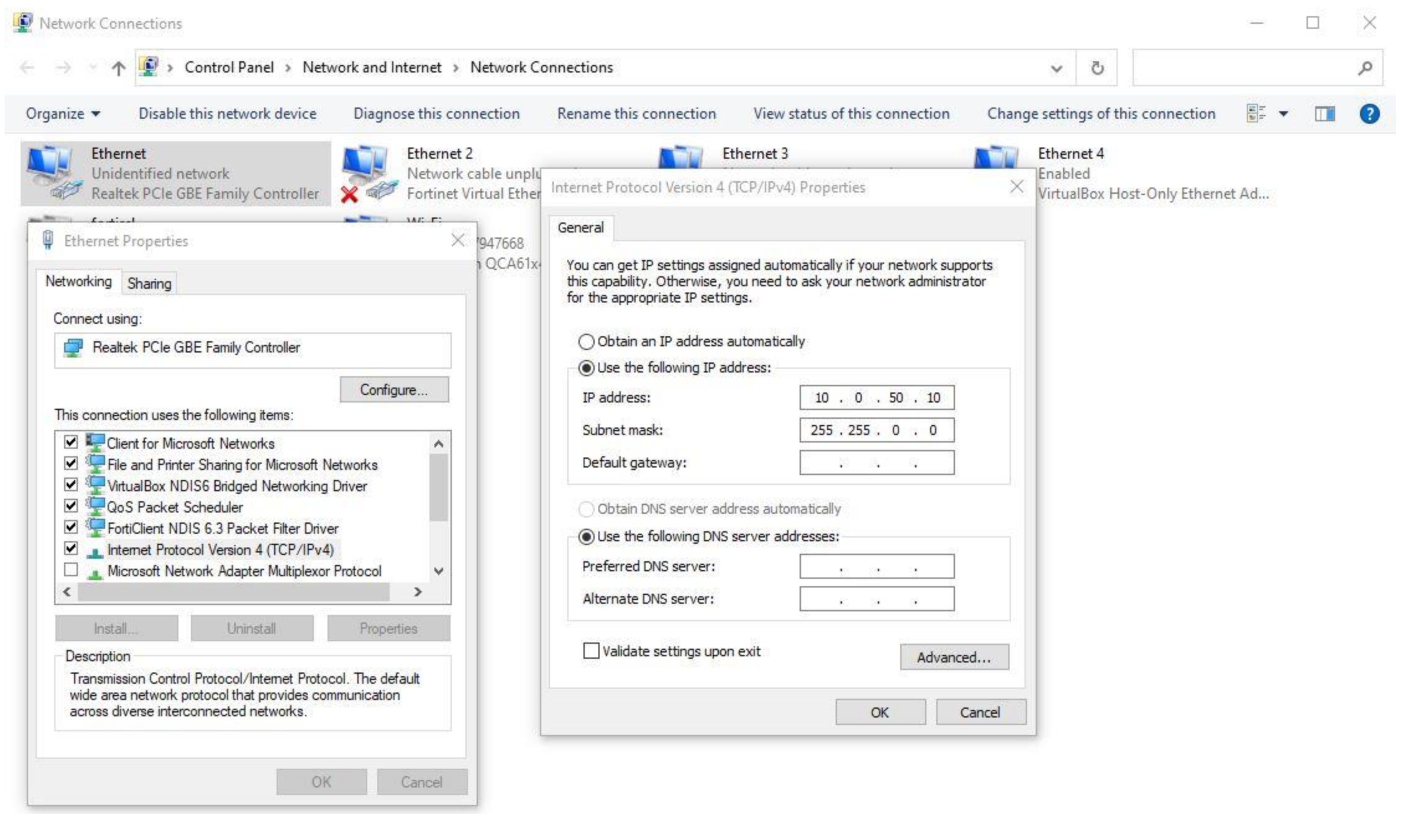
Getting Started Guide

Connecting SE59XX

Connect SE59XX to network or directly to pc.

Device comes with default static ip 10.0.50.100 and subnet mask 255.255.0.0.

For the first time accessing set your pc lan configuration in the same IP range.


Network Connections
Control Panel > Network and Internet > Network Connections
Organize
Disable this network device
Diagnose this connection
Rename this connection
View status of this connection
Change settings of this connection

Ethernet
Unidentified network
Realtek PCIe GBE Family Controller

Ethernet 2
Network cable unplu
Fortinet Virtual Ether

Ethernet 3

Ethernet 4
Enabled
VirtualBox Host-Only Ethernet Ad...

Ethernet Properties

Connect using:
Realtek PCIe GBE Family Controller

This connection uses the following items:

- [✓] Client for Microsoft Networks
- [✓] File and Printer Sharing for Microsoft Networks
- [✓] VirtualBox NDIS6 Bridged Networking Driver
- [✓] QoS Packet Scheduler
- [✓] FortiClient NDIS 6.3 Packet Filter Driver
- [✓] Internet Protocol Version 4 (TCP/IPv4)
- [ ] Microsoft Network Adapter Multiplexor Protocol

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically [✓ Use the following IP address:]

IP address:	10 . 0 . 50 . 10
Subnet mask:	255 . 255 . 0 . 0
Default gateway:	. . .

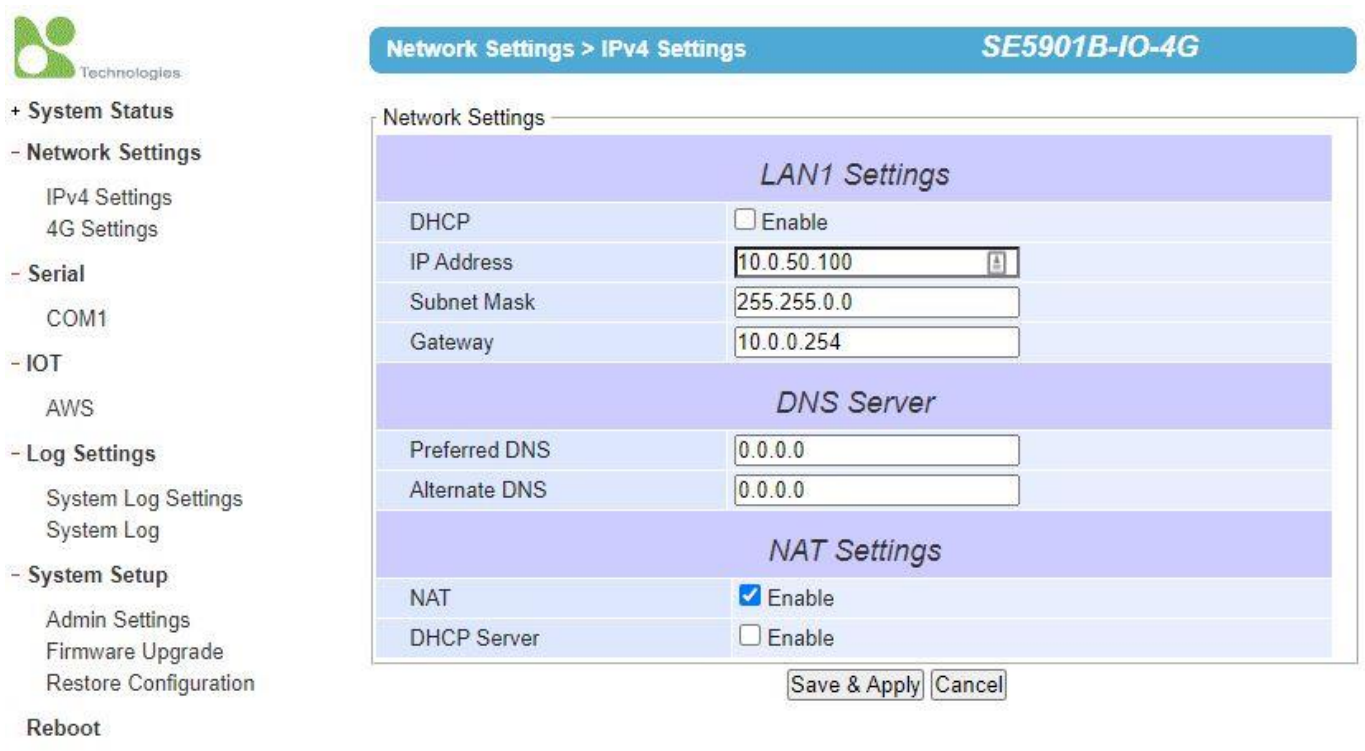
Obtain DNS server address automatically [✓ Use the following DNS server addresses:]

Preferred DNS server:	. . .
Alternate DNS server:	. . .

- [ ] Validate settings upon exit

Enter the web interface using ip 10.0.50.100 and use the default username and password provided by ATOP.

Configuring Network Settings



Network Settings

LAN1 Settings	
DHCP	☐ Enable
IP Address	10.0.50.100
Subnet Mask	255.255.0.0
Gateway	10.0.0.254
DNS Server	
Preferred DNS	0.0.0.0
Alternate DNS	0.0.0.0
NAT Settings	
NAT	✓ Enable
DHCP Server	☐ Enable

Enter the network settings by clicking on the Network Settings from the web GUI.

Enter any desired IP Address, Subnet mask and gateway required based on your preference.

Click on the Save & Apply button.

Dynamic IP assigning scheme can be used for getting the IP’s provided by the DHCP server in the network.

For this click on the DHCP Enable and do Save & Apply.

If required reboot the device using Reboot option from the web GUI.

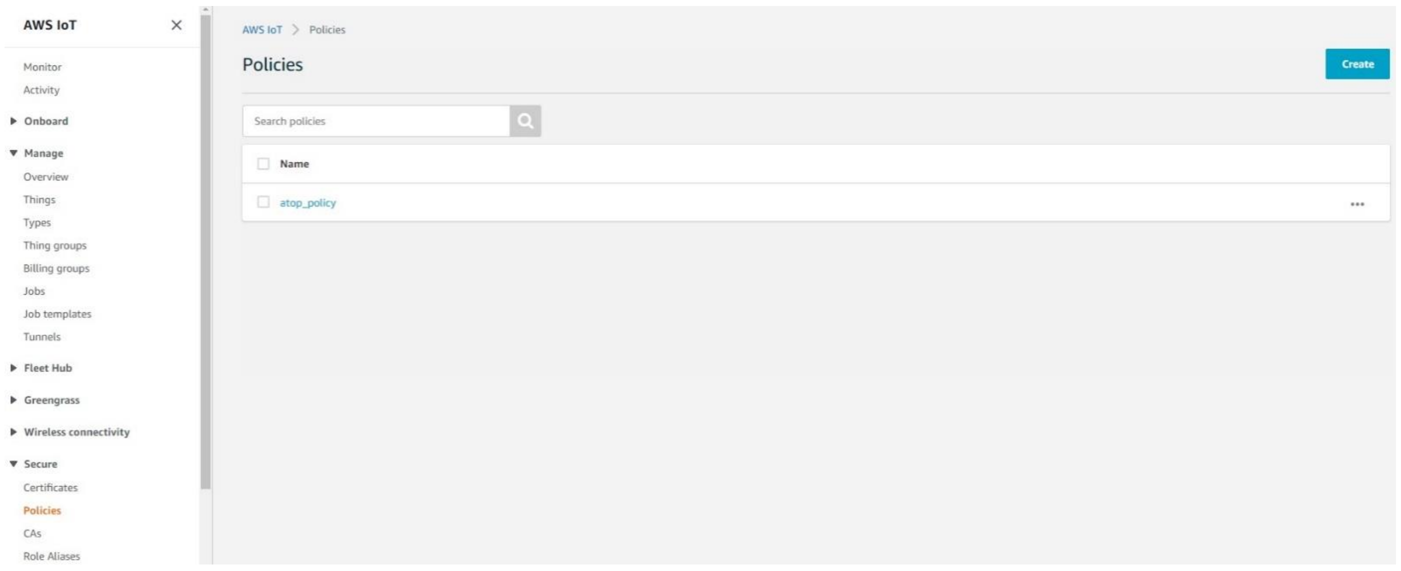
Configuring AWS IoT Core

Login to the aws console using https://aws.amazon.com.

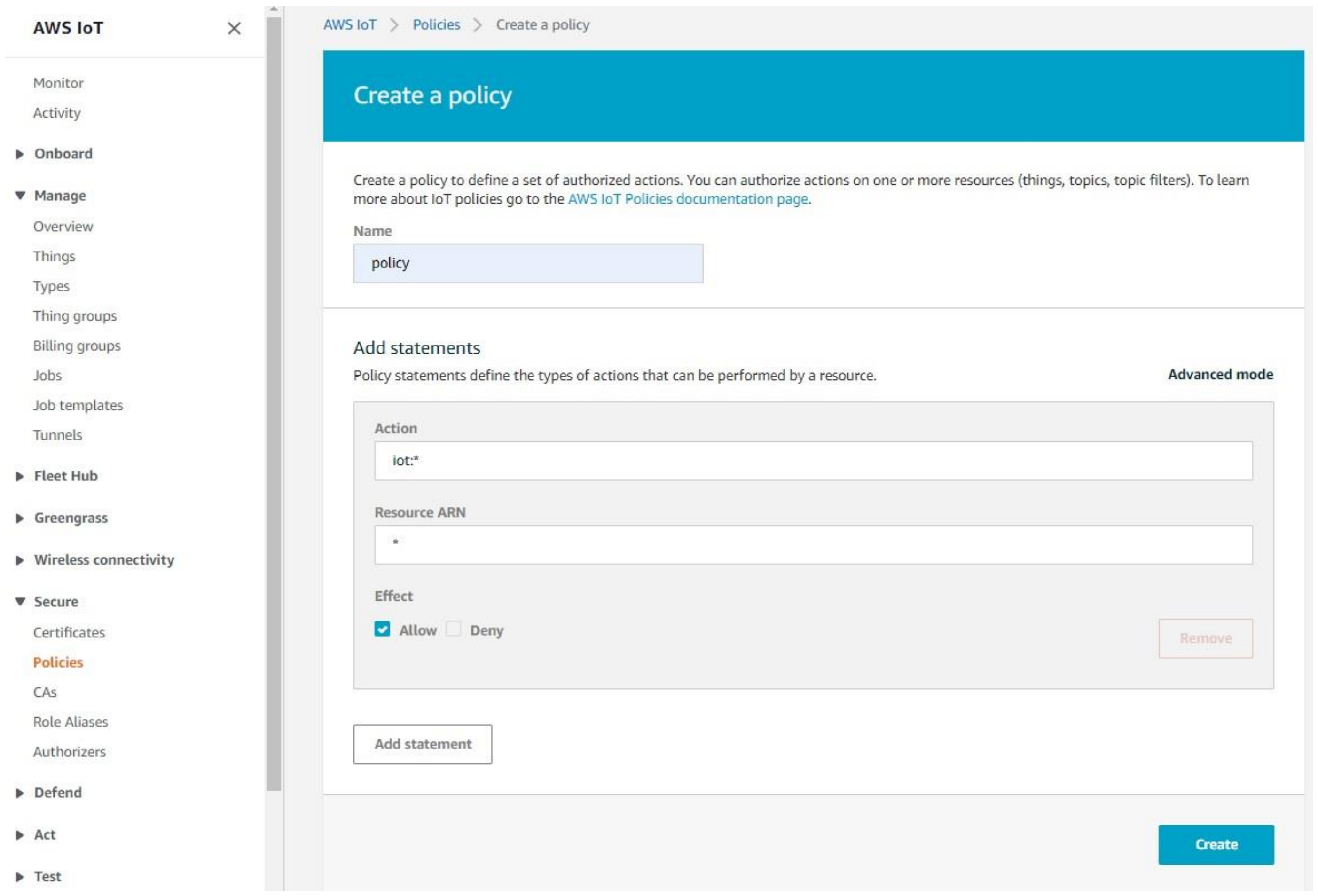
Select IoT core from the list of aws services.

Firstly a policy has to be created.

For this go to Secure menu and click on the policies.



Then create a policy by clicking on the create button.



Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

Action: iot:*

Resource ARN: *

Effect: [✓ Allow] Deny

Give any name to the policy and specify action as iot:*, so that it permits all iot actions.

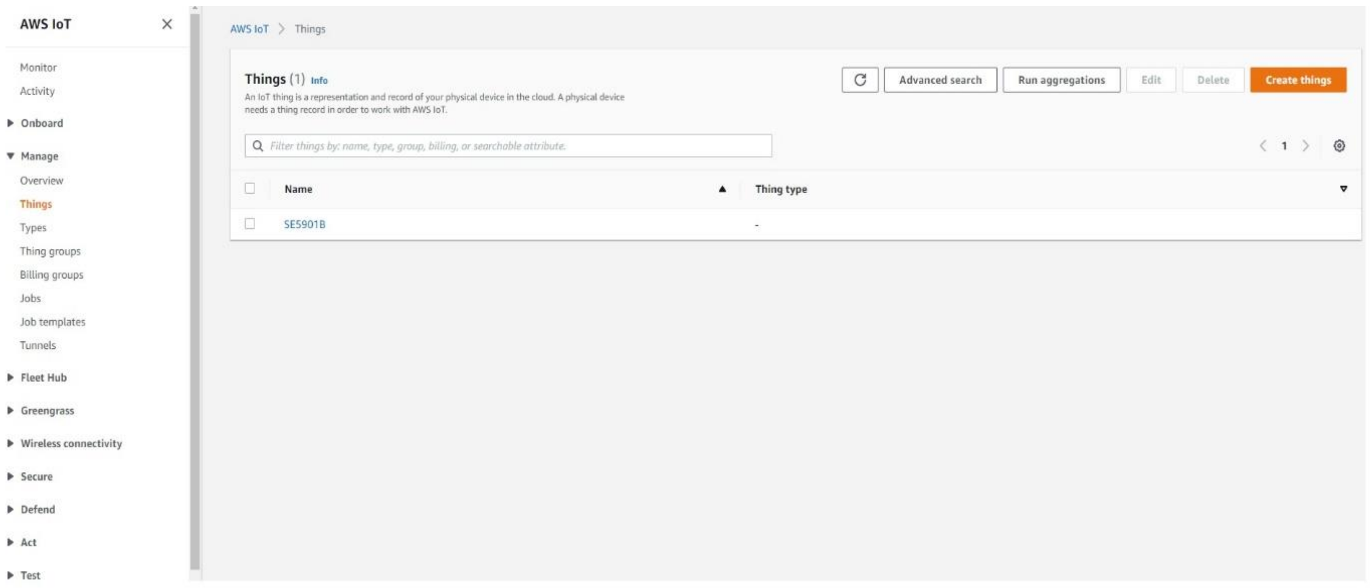
If you want to allow only specific set of action, it can be configured here.

The policy created can’t be modified later.

Resource ARN can be marked as * and effect can be marked as Allow.

Secondly a thing object has to be created in here.

In order to create a thing, go to Manage and navigate to Things.



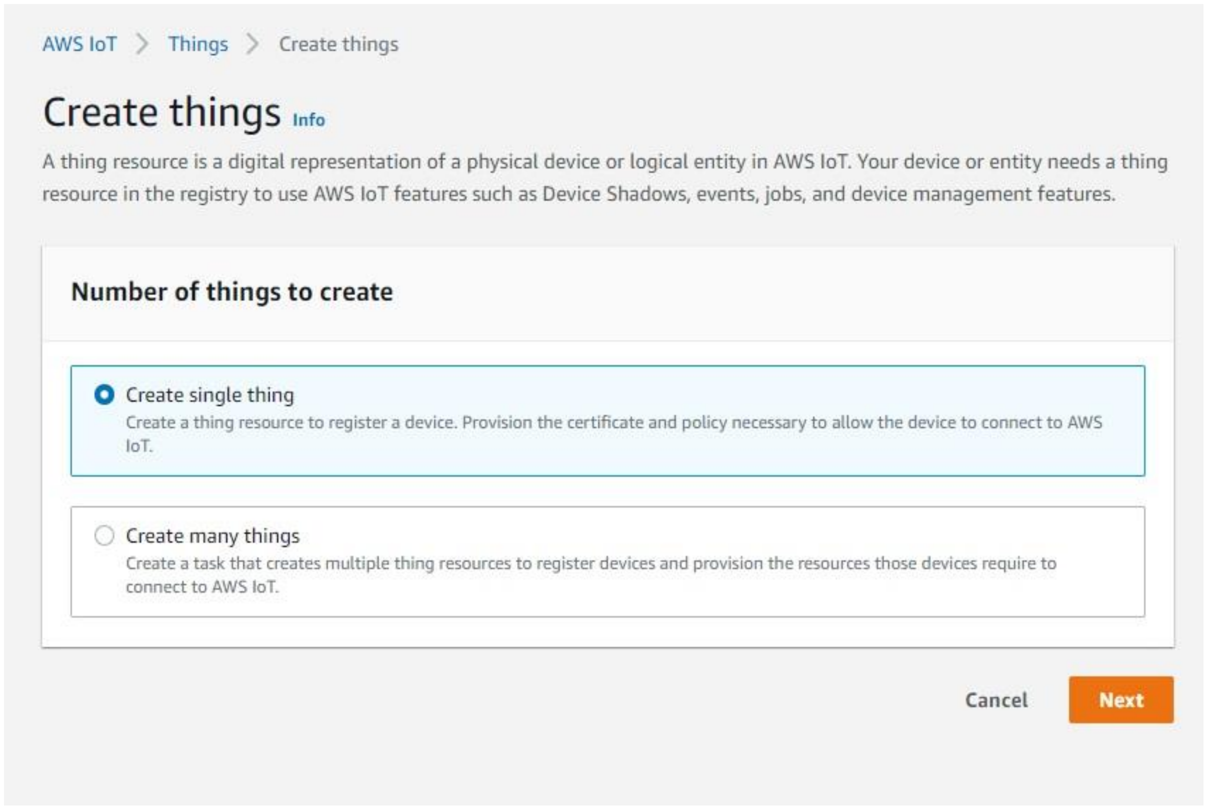
Things (1) Info

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

Name	Thing type
SE5901B	-

Then click on the Create things.

Select Create single thing.


AWS IoT > Things > Create things

Create things Info

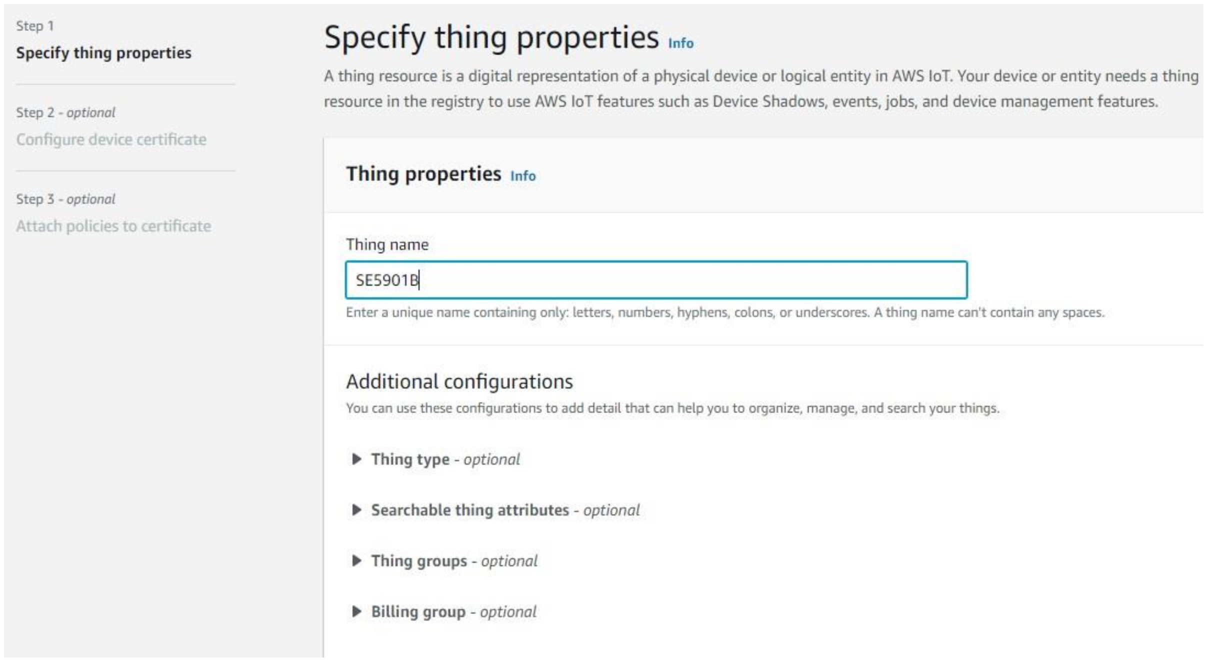
A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Number of things to create

[✓ Create single thing]
Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.

Create many things
Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

Specify thing name.



Specify thing properties Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties Info

Thing name: SE5901B

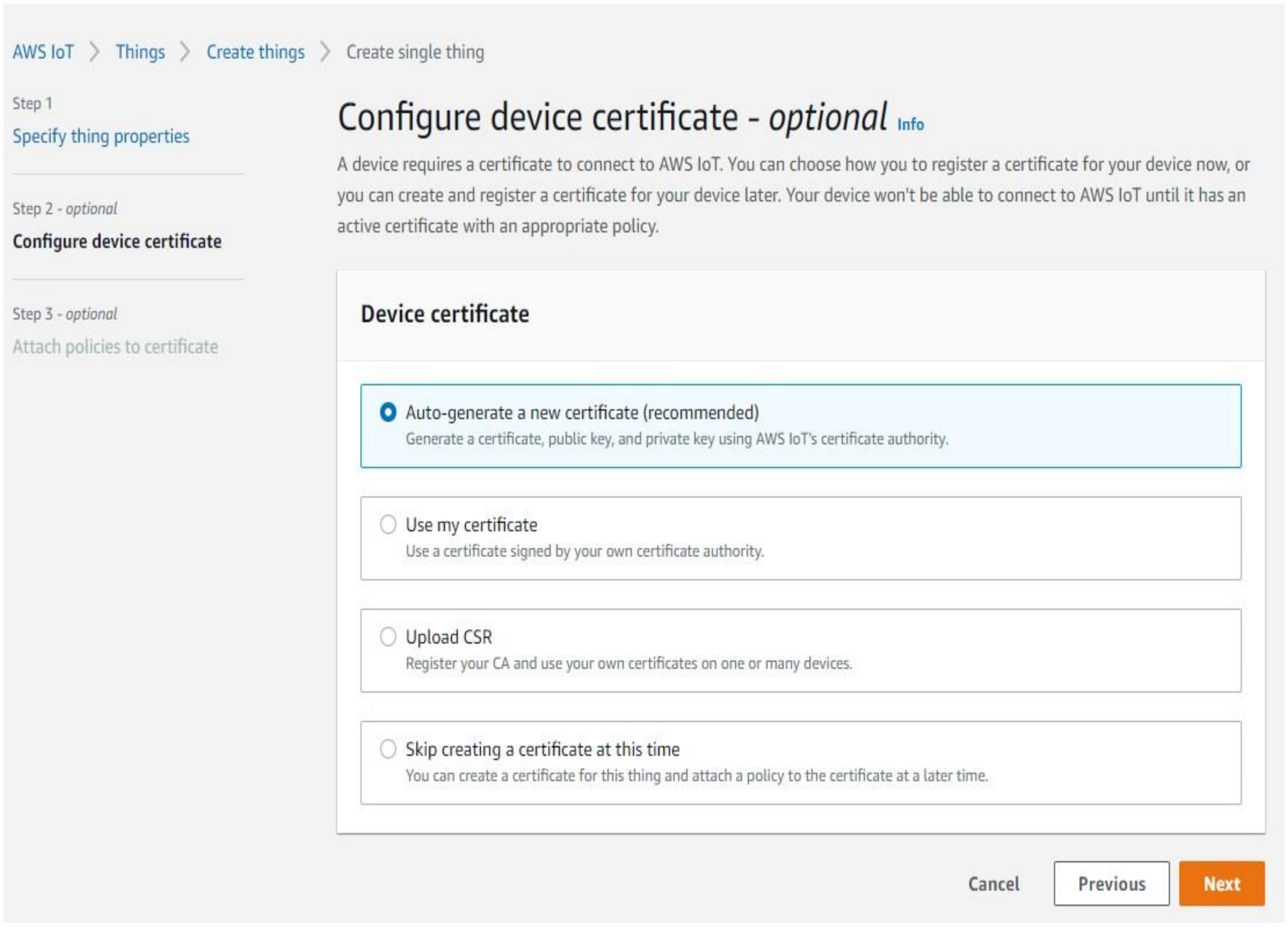
Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

▶ Thing type - optional
▶ Searchable thing attributes - optional
▶ Thing groups - optional
▶ Billing group - optional

Select Auto-generate a new certificate.



AWS IoT > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

Step 3 - *optional*
Attach policies to certificate

Configure device certificate - *optional* Info

A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

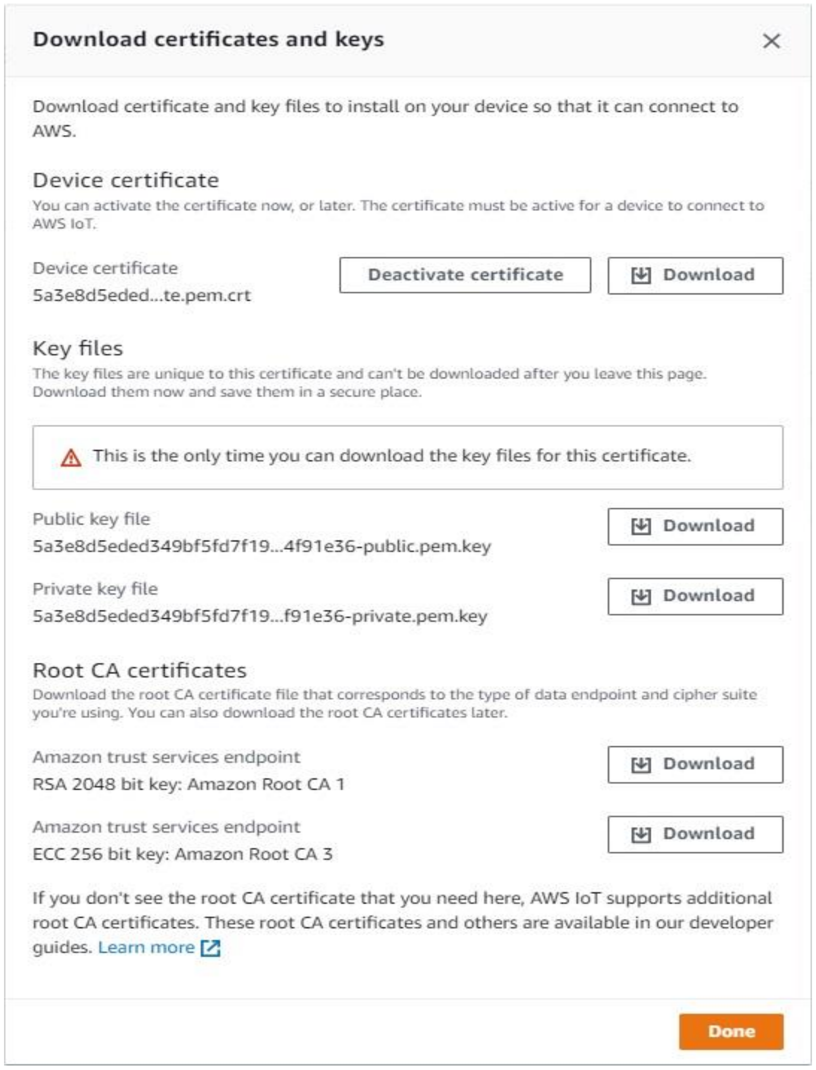
- [✓ Auto-generate a new certificate (recommended)]
  Generate a certificate, public key, and private key using AWS IoT's certificate authority.
- Use my certificate
  Use a certificate signed by your own certificate authority.
- Upload CSR
  Register your CA and use your own certificates on one or many devices.
- Skip creating a certificate at this time
  You can create a certificate for this thing and attach a policy to the certificate at a later time.

Download the device certificate, Public and Private key file.

If required activate device certificate.

Also download any one of the root ca certificates.

Without downloading all the required certificate, you won’t be able to click on Done.



Download certificates and keys

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
5a3e8d5eded...te.pem.crt

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

This is the only time you can download the key files for this certificate.

Public key file
5a3e8d5eded349bf5fd7f19...4f91e36-public.pem.key

Private key file
5a3e8d5eded349bf5fd7f19...f91e36-private.pem.key

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. Learn more

The device data end point can be found under settings.

Cross-compiling AWS Application

Download the tool-chain from the link shown below.

“www.google.com”

Cross-compile custom aws application using the tool chain.

There are two ways of providing aws parameters while doing cross-compilation.

- Defining aws parameters directly in the program

  The aws parameters can be directly defined in the programs statically.

  But keep in mind that all the certificates and key file location should be “/etc/ssl/certs/***

  Eg:
  ```
  #define AWS_MQTT_PORT 8883
  #define ROOT_CA_CERT_PATH “/etc/ssl/certs/*.crt”
  ```

- Using cmake variables.

  Another method of providing aws parameters are with the help of cmake variables.

  Some of the commonly used variables are AWS_IOT_ENDPOINT, ROOT_CA_CERT_PATH, CLIENT_CERT_PATH, DCLIENT_PRIVATE_KEY_PATH, AWS_MQTT_PORT, THING_NAME etc.

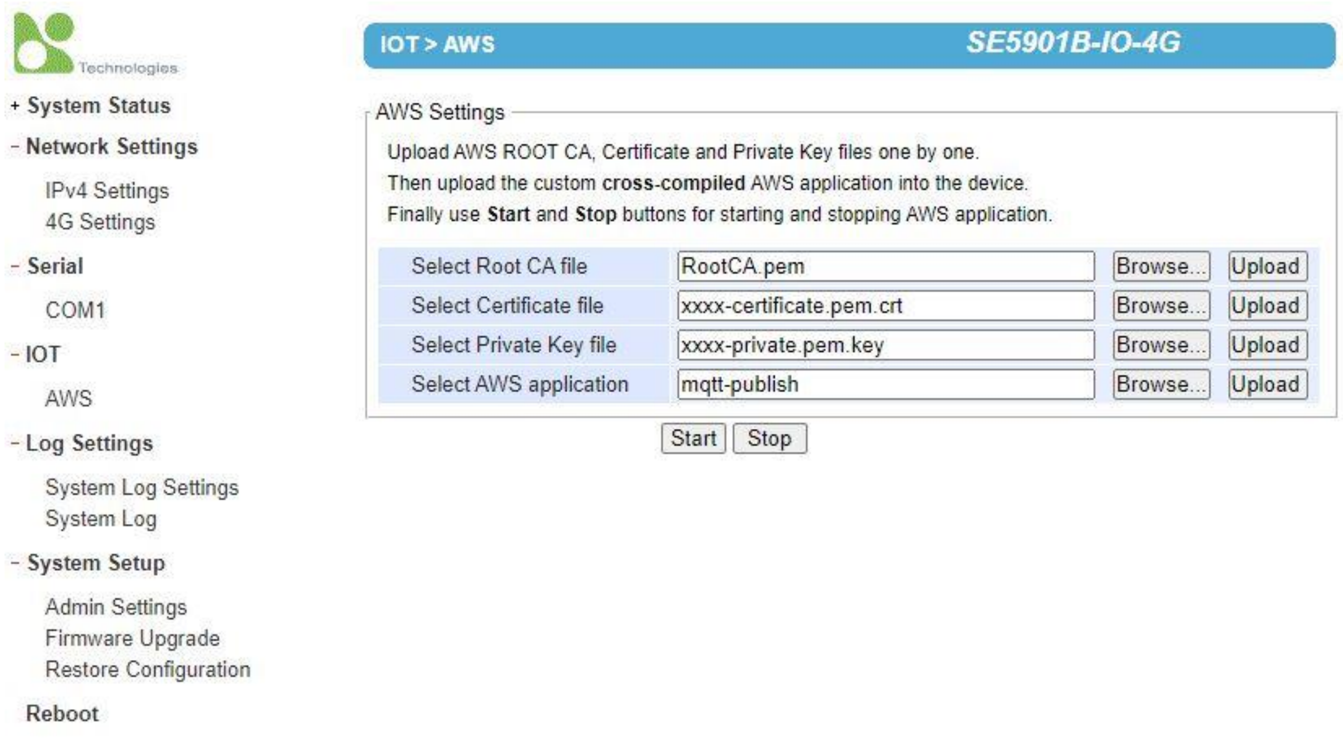
  The cmake variables can be used as follows.

  Eg:
  ```
  cmake -S. -Bbuild -DAWS_MQTT_PORT=8883 -DROOT_CA_CERT_PATH=/etc/ssl/certs/RootCA.pem
  ```

Once the compiled application is ready we need to upload it to the device along with the certificates required.

Use the device data end point from IoT core settings.

Connecting to AWS IoT Cloud



AWS Settings

Upload AWS ROOT CA, Certificate and Private Key files one by one.
Then upload the custom **cross-compiled** AWS application into the device.
Finally use **Start** and **Stop** buttons for starting and stopping AWS application.

Select Root CA file	RootCA.pem
Select Certificate file	xxxx-certificate.pem.crt
Select Private Key file	xxxx-private.pem.key
Select AWS application	mqtt-publish

To connect to AWS firstly need to upload certificates and applications into the device.

In order to do that go to IOT and navigate to AWS.

Select the Root CA file, browse and upload the file.

Similarly upload Certificate and Private key file downloaded from AWS console.

Finally upload the cross-compiled AWS application into the device.

Testing AWS Connectivity

Click on Start button to run your aws application from the device web gui.

Go to AWS IoT core and navigate to Test and select MQTT Test client.

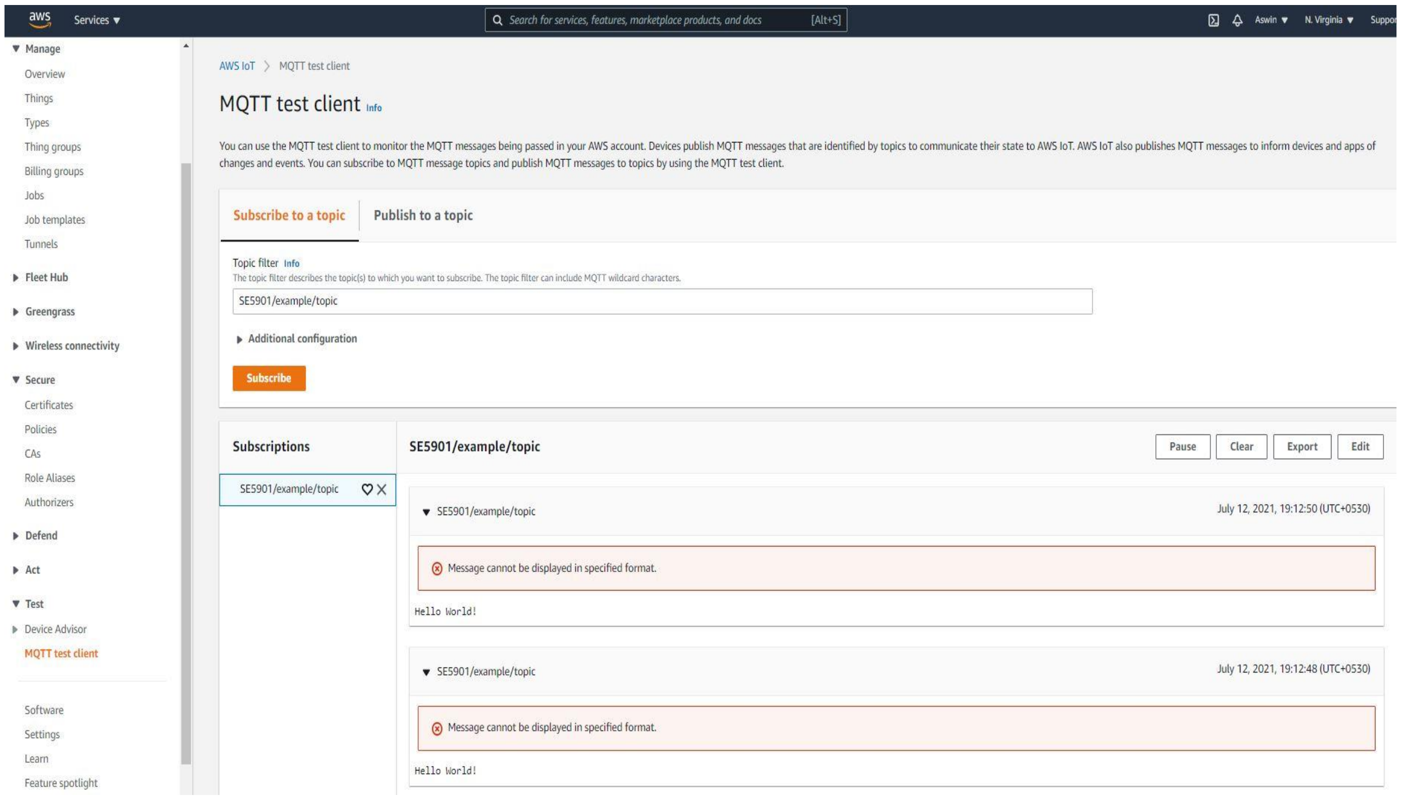
Enter the topic name specified in your program in the topic filter and click on the Subscribe button.

By default aws demo’s topic name would be thing/example/topic.

For the thing name SE5901 topic is SE5901/example/topic.

And you can see the messages published in the console.

Use stop button in the device web gui for stopping the aws application.



MQTT test client Info

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Topic filter Info
The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.
SE5901/example/topic

Subscriptions: SE5901/example/topic

SE5901/example/topic

```
SE5901/example/topic                    July 12, 2021, 19:12:50 (UTC+0530)
Message cannot be displayed in specified format.
Hello World!

SE5901/example/topic                    July 12, 2021, 19:12:48 (UTC+0530)
Message cannot be displayed in specified format.
Hello World!
```

Thank You

Connect to ATOP FAE for any further assistance