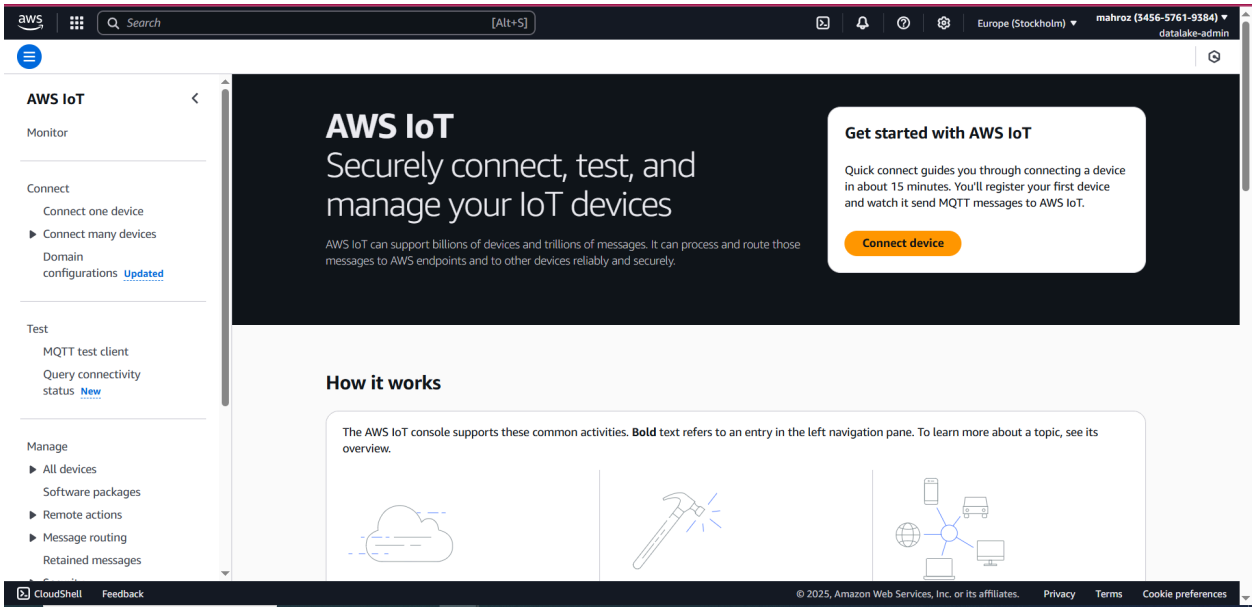
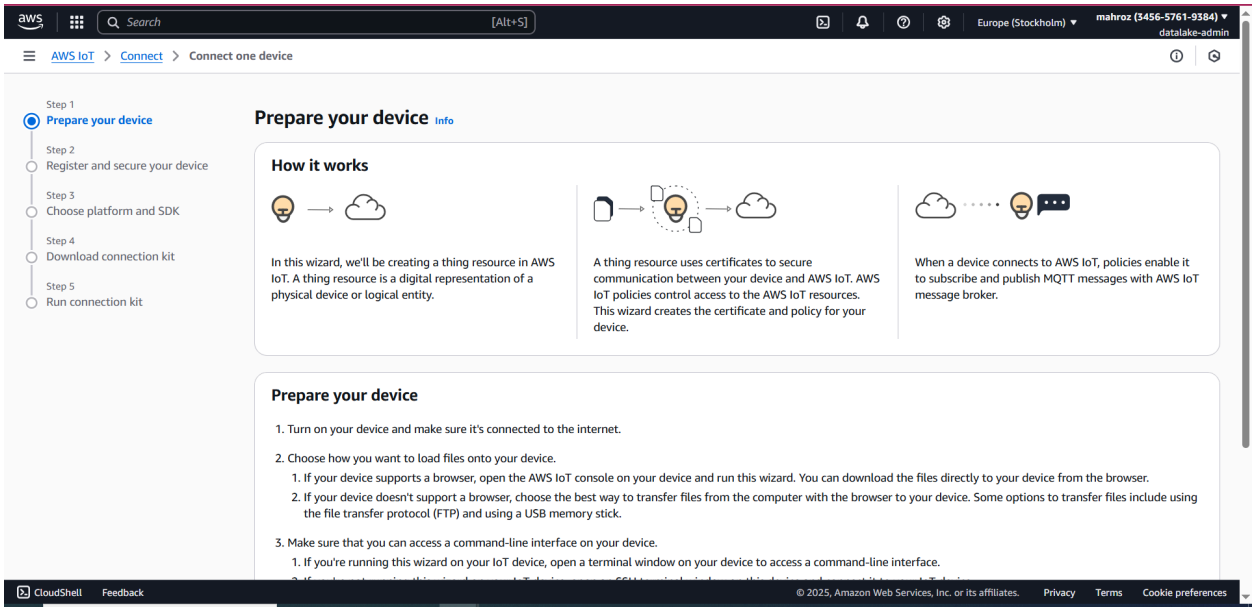
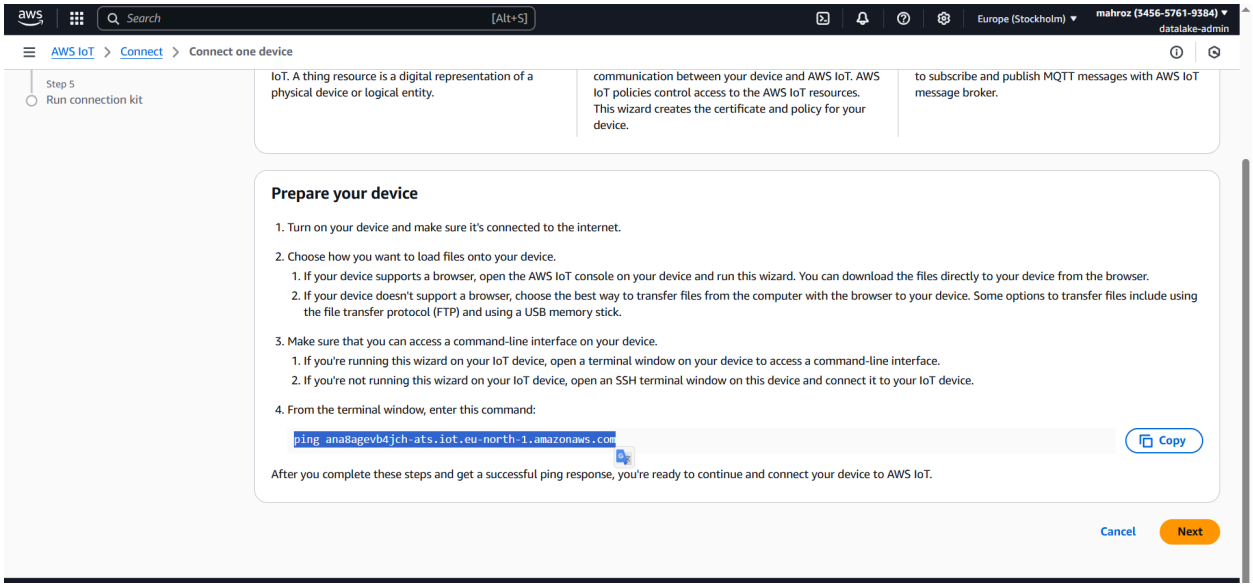
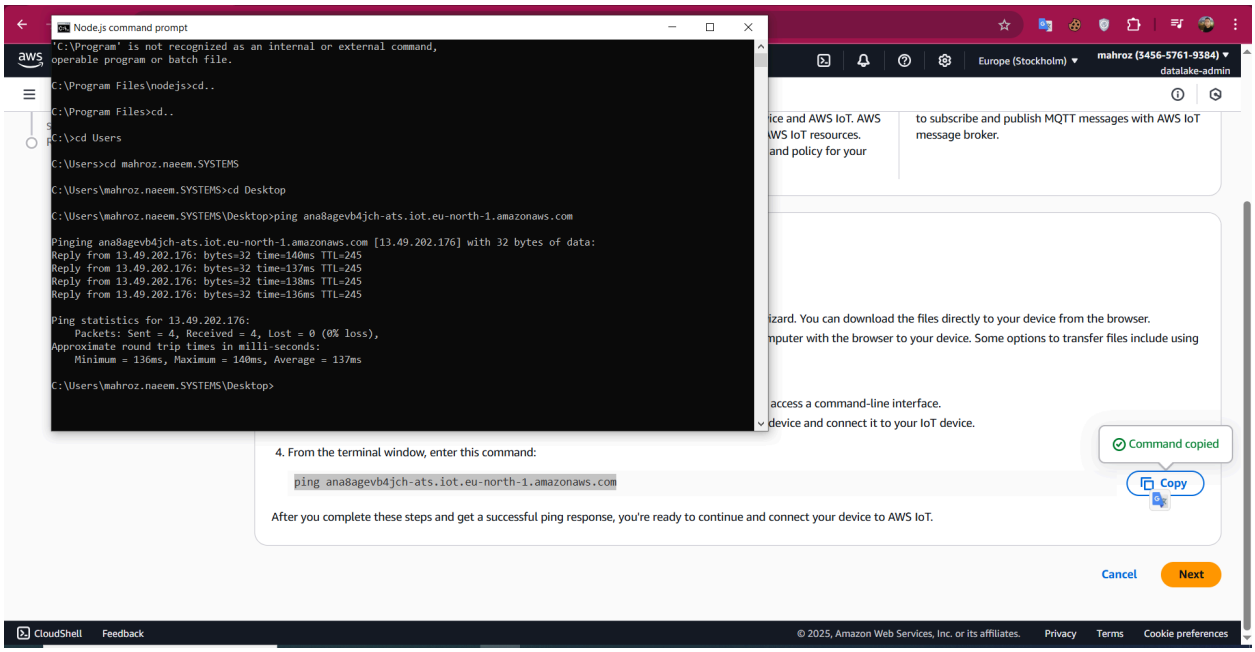
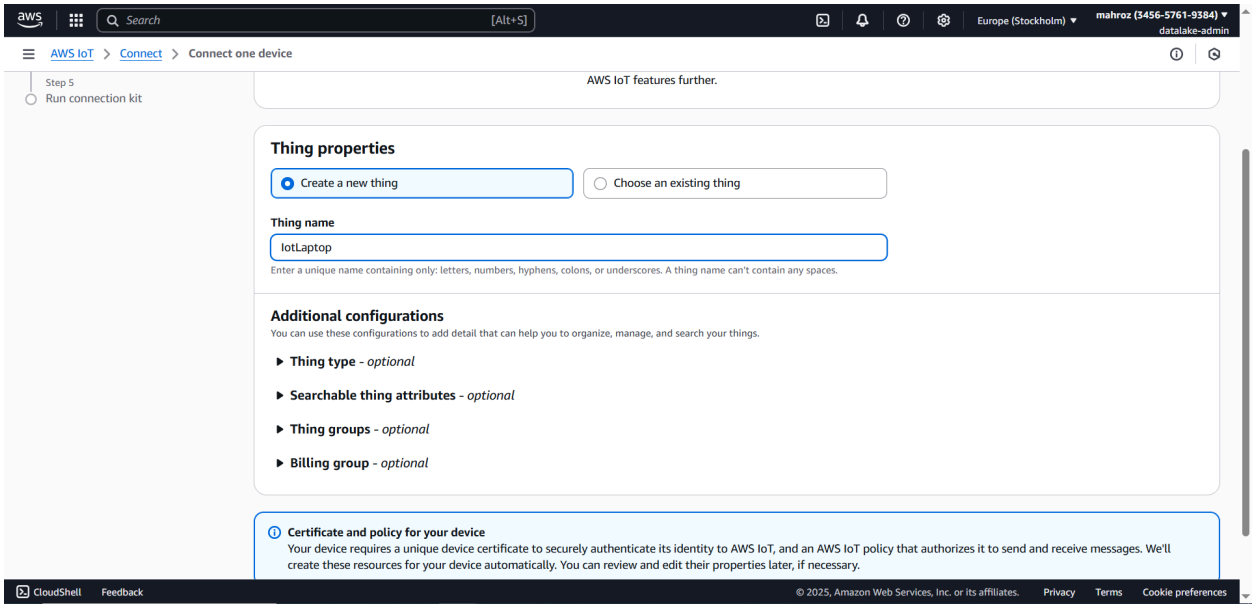
# AWS IoT

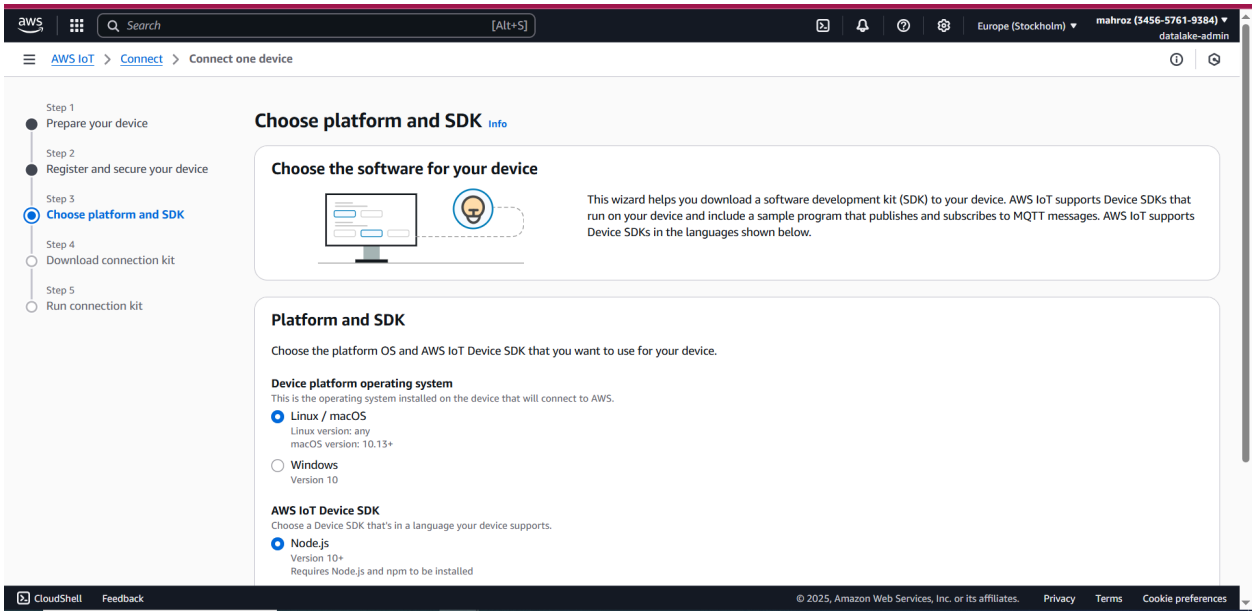

Search
[Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT
Monitor
Connect
Connect one device
Connect many devices
Domain configurations Updated
Test
MQTT test client
Query connectivity status New
Manage
All devices
Software packages
Remote actions
Message routing
Retained messages
AWS IoT
Securely connect, test, and manage your IoT devices
AWS IoT can support billions of devices and trillions of messages. It can process and route those messages to AWS endpoints and to other devices reliably and securely.
Get started with AWS IoT
Quick connect guides you through connecting a device in about 15 minutes. You'll register your first device and watch it send MQTT messages to AWS IoT.
Connect device
How it works
The AWS IoT console supports these common activities. Bold text refers to an entry in the left navigation pane. To learn more about a topic, see its overview.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

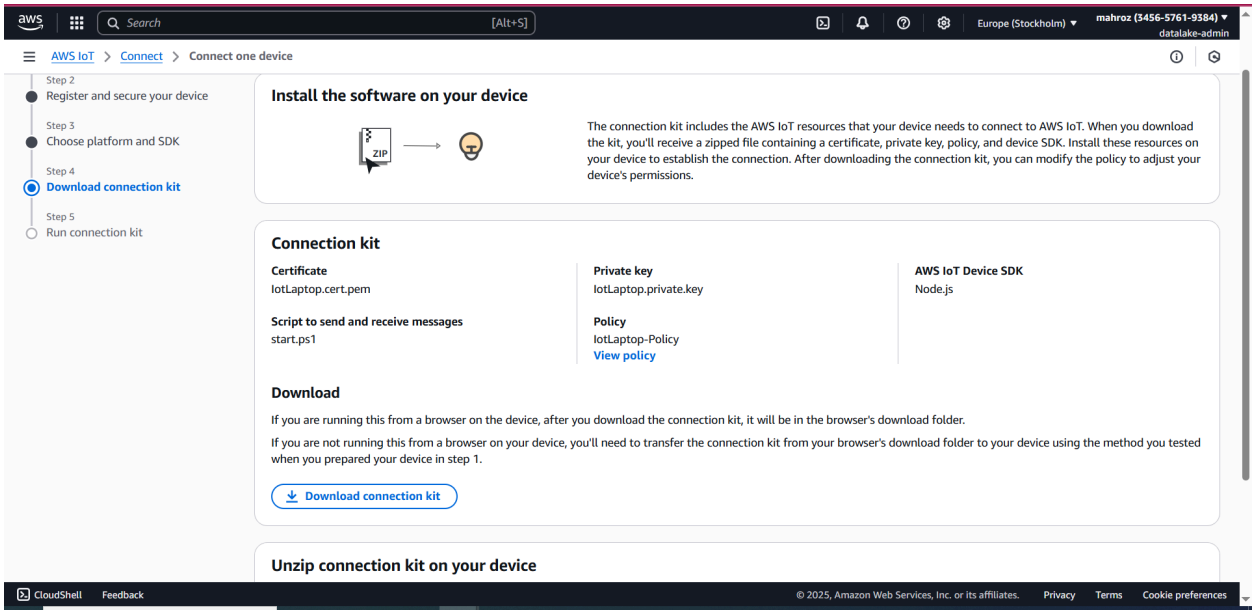

Search
[Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT > Connect > Connect one device
Step 1
Prepare your device
Step 2
Register and secure your device
Step 3
Choose platform and SDK
Step 4
Download connection kit
Step 5
Run connection kit
Prepare your device Info
How it works
In this wizard, we'll be creating a thing resource in AWS IoT. A thing resource is a digital representation of a physical device or logical entity.
A thing resource uses certificates to secure communication between your device and AWS IoT. AWS IoT policies control access to the AWS IoT resources. This wizard creates the certificate and policy for your device.
When a device connects to AWS IoT, policies enable it to subscribe and publish MQTT messages with AWS IoT message broker.
Prepare your device
1. Turn on your device and make sure it's connected to the internet.
2. Choose how you want to load files onto your device.
1. If your device supports a browser, open the AWS IoT console on your device and run this wizard. You can download the files directly to your device from the browser.
2. If your device doesn't support a browser, choose the best way to transfer files from the computer with the browser to your device. Some options to transfer files include using the file transfer protocol (FTP) and using a USB memory stick.
3. Make sure that you can access a command-line interface on your device.
1. If you're running this wizard on your IoT device, open a terminal window on your device to access a command-line interface.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

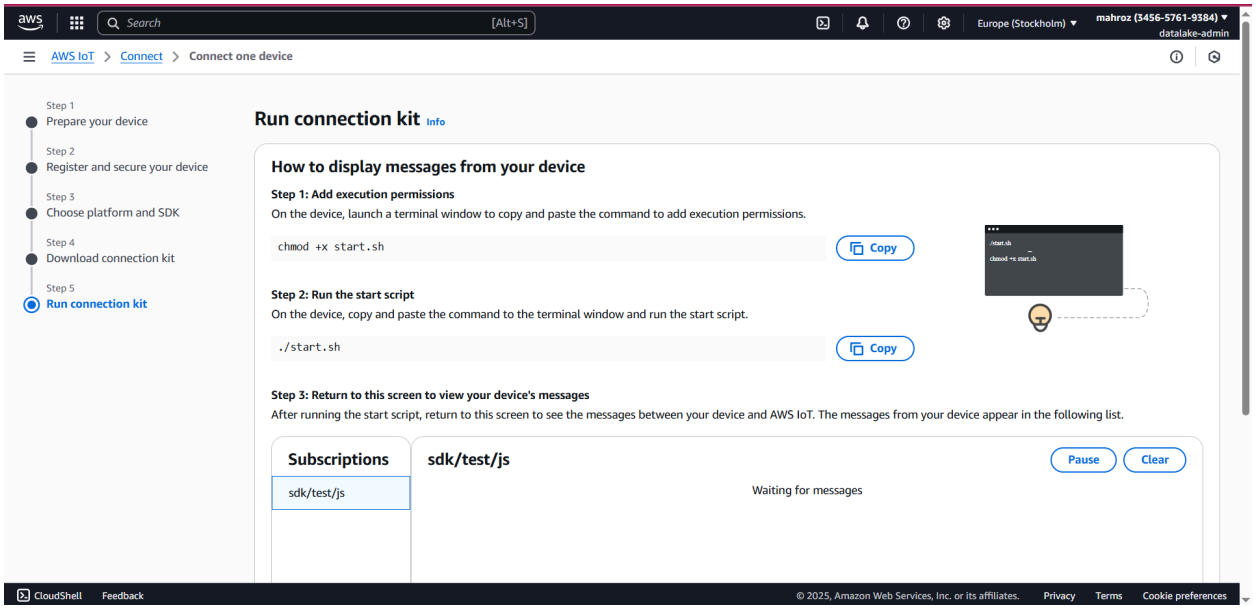

aws
Search
[Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT
Connect
Connect one device
Step 5
Run connection kit
IoT. A thing resource is a digital representation of a physical device or logical entity.
communication between your device and AWS IoT. AWS IoT policies control access to the AWS IoT resources. This wizard creates the certificate and policy for your device.
to subscribe and publish MQTT messages with AWS IoT message broker.
Prepare your device
1. Turn on your device and make sure it's connected to the internet.
2. Choose how you want to load files onto your device.
1. If your device supports a browser, open the AWS IoT console on your device and run this wizard. You can download the files directly to your device from the browser.
2. If your device doesn't support a browser, choose the best way to transfer files from the computer with the browser to your device. Some options to transfer files include using the file transfer protocol (FTP) and using a USB memory stick.
3. Make sure that you can access a command-line interface on your device.
1. If you're running this wizard on your IoT device, open a terminal window on your device to access a command-line interface.
2. If you're not running this wizard on your IoT device, open an SSH terminal window on this device and connect it to your IoT device.
4. From the terminal window, enter this command:
ping ana8agevb4jch-ats.iot.eu-north-1.amazonaws.com
Copy
After you complete these steps and get a successful ping response, you're ready to continue and connect your device to AWS IoT.
Cancel
Next


Node.js command prompt
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
C:\Program Files\nodejs>cd..
C:\Program Files>cd..
C:\>cd Users
C:\Users>cd mahroz.naeem.SYSTEMS
C:\Users\mahroz.naeem.SYSTEMS>cd Desktop
C:\Users\mahroz.naeem.SYSTEMS\Desktop>ping ana8agevb4jch-ats.iot.eu-north-1.amazonaws.com
Pinging ana8agevb4jch-ats.iot.eu-north-1.amazonaws.com [13.49.202.176] with 32 bytes of data:
Reply from 13.49.202.176: bytes=32 time=140ms TTL=245
Reply from 13.49.202.176: bytes=32 time=137ms TTL=245
Reply from 13.49.202.176: bytes=32 time=138ms TTL=245
Reply from 13.49.202.176: bytes=32 time=136ms TTL=245
Ping statistics for 13.49.202.176:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 136ms, Maximum = 140ms, Average = 137ms
C:\Users\mahroz.naeem.SYSTEMS\Desktop>
Command copied
Copy
Cancel
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

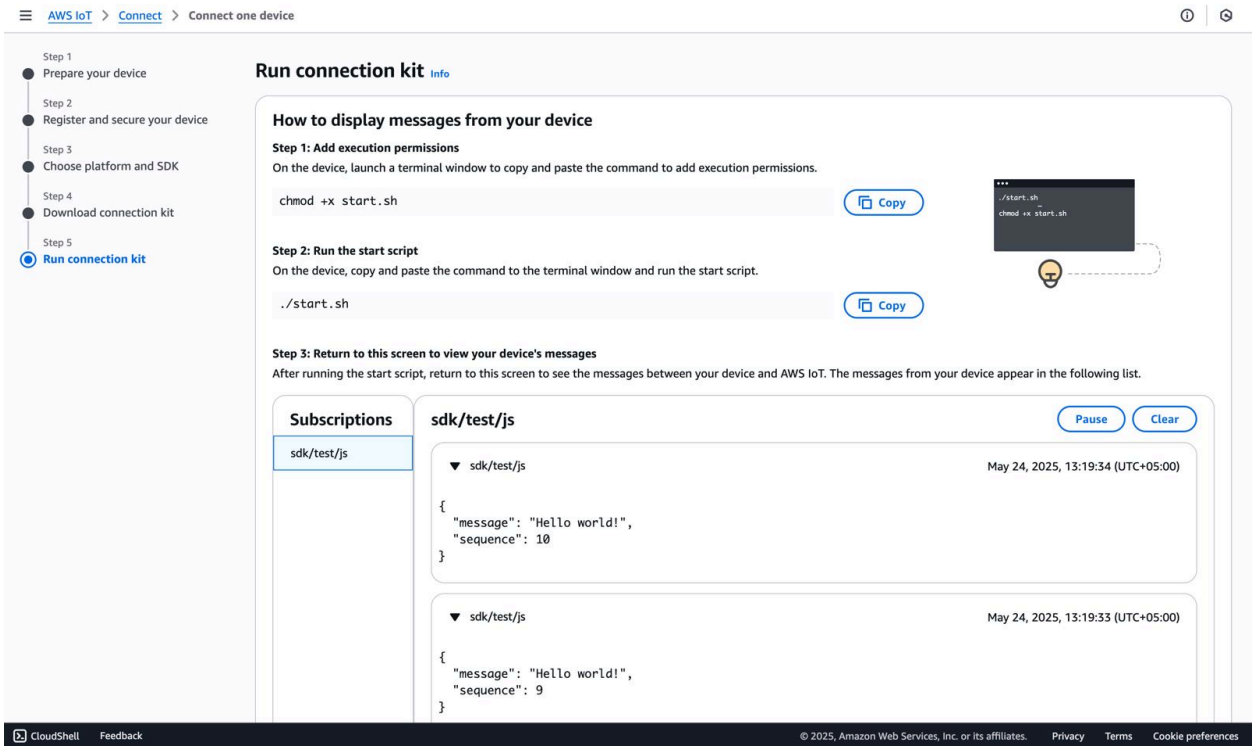

Search
[Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT
Connect
Connect one device
Step 5
Run connection kit
AWS IoT features further.
Thing properties
Create a new thing
Choose an existing thing
Thing name
IotLaptop
Enter a unique name containing only: letters, numbers, hyphens, 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
Certificate and policy for your device
Your device requires a unique device certificate to securely authenticate its identity to AWS IoT, and an AWS IoT policy that authorizes it to send and receive messages. We'll create these resources for your device automatically. You can review and edit their properties later, if necessary.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

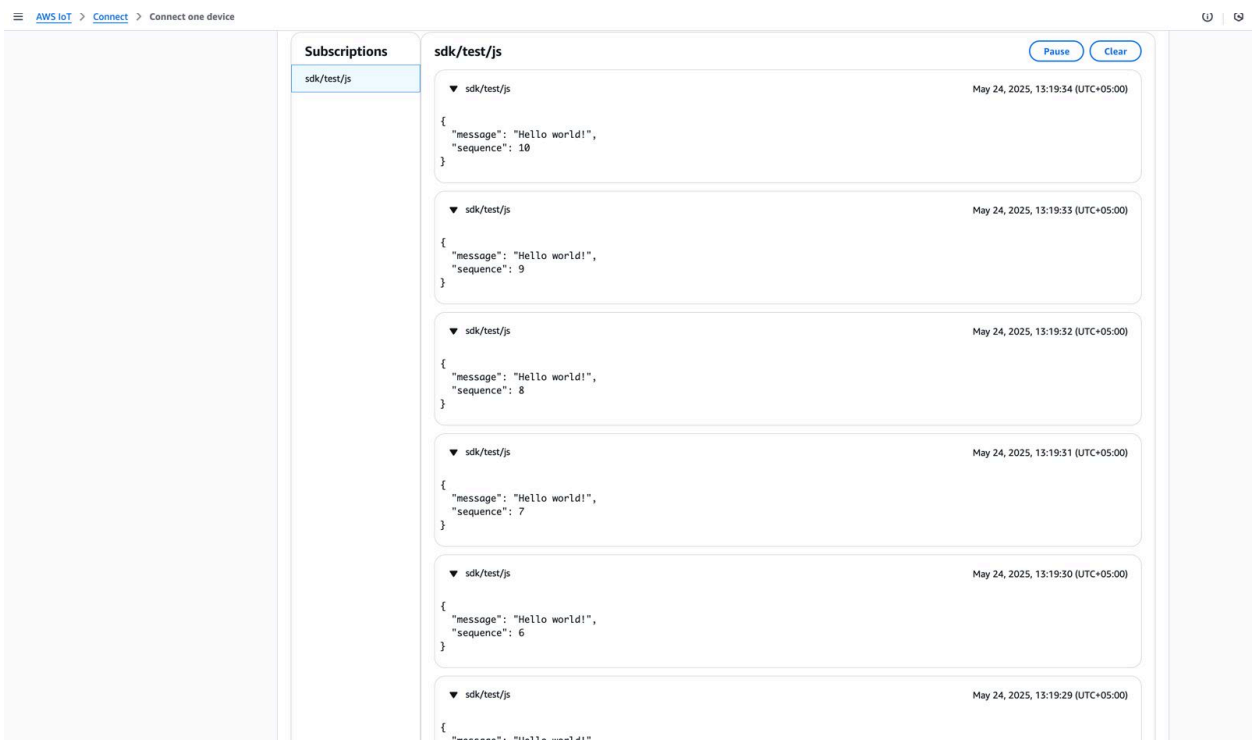

Search
[Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT
Connect
Connect one device
Step 1
Prepare your device
Step 2
Register and secure your device
Step 3
Choose platform and SDK
Step 4
Download connection kit
Step 5
Run connection kit
Choose platform and SDK
Info
Choose the software for your device
This wizard helps you download a software development kit (SDK) to your device. AWS IoT supports Device SDKs that run on your device and include a sample program that publishes and subscribes to MQTT messages. AWS IoT supports Device SDKs in the languages shown below.
Platform and SDK
Choose the platform OS and AWS IoT Device SDK that you want to use for your device.
Device platform operating system
This is the operating system installed on the device that will connect to AWS.
Linux / macOS
Linux version: any
macOS version: 10.13+
Windows
Version 10
AWS IoT Device SDK
Choose a Device SDK that's in a language your device supports.
Node.js
Version 10+
Requires Node.js and npm to be installed
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT > Connect > Connect one device
Step 2
Register and secure your device
Step 3
Choose platform and SDK
Step 4
Download connection kit
Step 5
Run connection kit
Install the software on your device
ZIP
The connection kit includes the AWS IoT resources that your device needs to connect to AWS IoT. When you download the kit, you'll receive a zipped file containing a certificate, private key, policy, and device SDK. Install these resources on your device to establish the connection. After downloading the connection kit, you can modify the policy to adjust your device's permissions.
Connection kit
Certificate
IotLaptop.cert.pem
Private key
IotLaptop.private.key
AWS IoT Device SDK
Node.js
Script to send and receive messages
start.ps1
Policy
IotLaptop-Policy
View policy
Download
If you are running this from a browser on the device, after you download the connection kit, it will be in the browser's download folder.
If you are not running this from a browser on your device, you'll need to transfer the connection kit from your browser's download folder to your device using the method you tested when you prepared your device in step 1.
Download connection kit
Unzip connection kit on your device
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
Europe (Stockholm)
mahroz (3456-5761-9384)
datalake-admin
AWS IoT > Connect > Connect one device
Step 1
Prepare your device
Step 2
Register and secure your device
Step 3
Choose platform and SDK
Step 4
Download connection kit
Step 5
Run connection kit
Run connection kit Info
How to display messages from your device
Step 1: Add execution permissions
On the device, launch a terminal window to copy and paste the command to add execution permissions.
chmod +x start.sh
Copy
Step 2: Run the start script
On the device, copy and paste the command to the terminal window and run the start script.
./start.sh
Copy
Step 3: Return to this screen to view your device's messages
After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.
Subscriptions
sdk/test/js
sdk/test/js
Pause
Clear
Waiting for messages
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
######################################################################## 100.0%
==> Fetching c-ares
==> Downloading https://ghcr.io/v2/homebrew/core/c-ares/blobs/sha256:76858a7f6dd
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/libnghttp2/manifests/1.65.0
######################################################################## 100.0%
==> Fetching libnghttp2
==> Downloading https://ghcr.io/v2/homebrew/core/libnghttp2/blobs/sha256:492e4da
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/libuv/manifests/1.51.0
######################################################################## 100.0%
==> Fetching libuv
==> Downloading https://ghcr.io/v2/homebrew/core/libuv/blobs/sha256:916f444748e9
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2025-
######################################################################## 100.0%
==> Fetching ca-certificates
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/blobs/sha256:dd
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/3/manifests/3.5.0
######################################################################## 100.0%
==> Fetching openssl@3
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/3/blobs/sha256:7bbac0e8
######################################################################## 100.0%
==> Fetching node
==> Downloading https://ghcr.io/v2/homebrew/core/node/blobs/sha256:5e18e14319326
######################################################################## 100.0%
==> Installing dependencies for node: brotli, c-ares, libnghttp2, libuv, ca-certificates and openssl@3
==> Installing node dependency: brotli
==> Downloading https://ghcr.io/v2/homebrew/core/brotli/manifests/1.1.0-2
Already downloaded: /Users/alinaeem/Library/Caches/Homebrew/downloads/72b36368116298518fd66e638dfbe8cb541d8db2d33e279683bbdb4b79c73e0d--brotli-1.1.0-2.bottle_manifest.json
==> Pouring brotli--1.1.0.arm64_sequoia.bottle.2.tar.gz
🍺  /opt/homebrew/Cellar/brotli/1.1.0: 28 files, 1.7MB
==> Installing node dependency: c-ares
==> Downloading https://ghcr.io/v2/homebrew/core/c-ares/manifests/1.34.5-1
Already downloaded: /Users/alinaeem/Library/Caches/Homebrew/downloads/c52b1c914606673127555a235f78ac192a45bff39befbba6da9aa72a17263c559--c-ares-1.34.5-1.bottle_manifest.json
==> Pouring c-ares--1.34.5.arm64_sequoia.bottle.1.tar.gz
🍺  /opt/homebrew/Cellar/c-ares/1.34.5: 176 files, 1MB
==> Installing node dependency: libnghttp2
==> Downloading https://ghcr.io/v2/homebrew/core/libnghttp2/manifests/1.65.0
Already downloaded: /Users/alinaeem/Library/Caches/Homebrew/downloads/5a72fb9f57dd728c5d6d30917849ae15111c3b440595f5694738ec9434c12427--libnghttp2-1.65.0.bottle_manifest.json
==> Pouring libnghttp2--1.65.0.arm64_sequoia.bottle.tar.gz
🍺  /opt/homebrew/Cellar/libnghttp2/1.65.0: 14 files, 767KB
==> Installing node dependency: libuv
==> Downloading https://ghcr.io/v2/homebrew/core/libuv/manifests/1.51.0
Already downloaded: /Users/alinaeem/Library/Caches/Homebrew/downloads/25bd9f2f86d047011dc8ca247bea2e4e3e5a150e29648418d48dfca4c8c136ea--libuv-1.51.0.bottle_manifest.json
==> Pouring libuv--1.51.0.arm64_sequoia.bottle.tar.gz
🍺  /opt/homebrew/Cellar/libuv/1.51.0: 20 files, 1.2MB
==> Installing node dependency: ca-certificates
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2025-
Already downloaded: /Users/alinaeem/Library/Caches/Homebrew/downloads/bc18acc15e0abddc102f828b57a29cfdbec1b6b002db37ad12bad9dbf0e9d12f--ca-certificates-2025-05-20.bottle_manifest.json
==> Pouring ca-certificates--2025-05-20.all.bottle.tar.gz
==> Regenerating CA certificate bundle from keychain, this may take a while...
🍺  /opt/homebrew/Cellar/ca-certificates/2025-05-20: 4 files, 225.8KB
==> Installing node dependency: openssl@3
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/3/manifests/3.5.0
Already downloaded: /Users/alinaeem/Library/Caches/Homebrew/downloads/d3c8243df37f69247721b1d8a6c51e1a8afd6664b832423921ffd99bb2bb381a--openssl@3-3.5.0.bottle_manifest.json
==> Pouring openssl@3--3.5.0.arm64_sequoia.bottle.tar.gz
🍺  /opt/homebrew/Cellar/openssl@3/3.5.0: 7,563 files, 35.4MB
==> Installing node
==> Pouring node--23.11.0.arm64_sequoia.bottle.1.tar.gz
🍺  /opt/homebrew/Cellar/node/23.11.0: 2,602 files, 72.7MB
==> Running `brew cleanup node`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```


AWS IoT > Connect > Connect one device
Step 1
Prepare your device
Step 2
Register and secure your device
Step 3
Choose platform and SDK
Step 4
Download connection kit
Step 5
Run connection kit
Run connection kit Info
How to display messages from your device
Step 1: Add execution permissions
On the device, launch a terminal window to copy and paste the command to add execution permissions.
chmod +x start.sh
Copy
Step 2: Run the start script
On the device, copy and paste the command to the terminal window and run the start script.
./start.sh
Copy
Step 3: Return to this screen to view your device's messages
After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.
Subscriptions
sdk/test/js
sdk/test/js
Pause
Clear
sdk/test/js
May 24, 2025, 13:19:34 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 10
}
sdk/test/js
May 24, 2025, 13:19:33 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 9
}
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


AWS IoT > Connect > Connect one device
Subscriptions
sdk/test/js
sdk/test/js
Pause
Clear
sdk/test/js
May 24, 2025, 13:19:34 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 10
}
sdk/test/js
May 24, 2025, 13:19:33 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 9
}
sdk/test/js
May 24, 2025, 13:19:32 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 8
}
sdk/test/js
May 24, 2025, 13:19:31 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 7
}
sdk/test/js
May 24, 2025, 13:19:30 (UTC+05:00)
{
"message": "Hello world!",
"sequence": 6
}
sdk/test/js
May 24, 2025, 13:19:29 (UTC+05:00)