 Installation Guide

1. Download RabbitMQ Installation media จาก official website

<https://www.rabbitmq.com/download.html>

A screenshot of a computer

Description automatically generated with medium confidence

1. Download Erlang จาก official website

<https://www.erlang.org/downloads>

A screenshot of a website

Description automatically generated with low confidence

1. ติดตั้ง Erlang โดยกด Next ตามขั้นตอนจนเสร็จ

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with low confidence

1. ติดตั้ง RabbitMQ โดยกด Next ตามขั้นตอนจนเสร็จ หลังติดตั้งเสร็จเลือก Start RabbitMQ service ด้วย

A screen shot of a computer

Description automatically generated with medium confidence

1. หลังติดตั้งเสร็จเข้าไปที่ Administrative Tools -> Services เพื่อ Make Sure ว่า RabbitMQ service อยู่ในสถานะ Start Automatically

A screenshot of a computer

Description automatically generated

1. เข้าไปที่ Start Menu -> RabbitMQ Server -> RabbitMQ Command Prompt (sbin) เพื่อเปิด Console สำหรับใส่ Command เพื่อจัดการ Configuration และ Plugin ต่างๆ

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated | A computer screen shot of a black screen  Description automatically generated with low confidence |

โดยรูปแบบคำสั่งสำหรับ Enable และ Disable plugin ดังนี้

และ

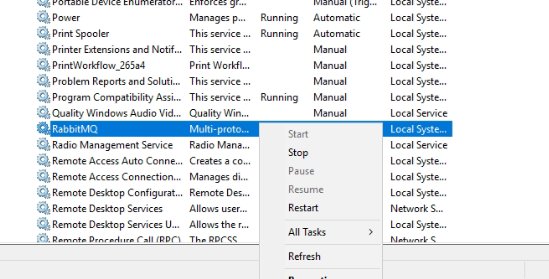
rabbitmq-plugins disable <plugin-name>

rabbitmq-plugins enable <plugin-name>

โดยเราต้อง Enable Plugin ที่สำคัญดังนี้

|  |  |  |
| --- | --- | --- |
| Plugin | วัตถุประสงค์ | คำสั่งในการ Enable |
| Web Management | ใช้ Monitoring Queue, Channel แบะ Performance ต่างๆของระบบผ่าน Web UI | rabbitmq-plugins enable rabbitmq\_management |
| AMQP | AMQP 1.0 protocol | rabbitmq-plugins enable rabbitmq\_amqp1\_0 |
| MQTT | MQTT Protocol | rabbitmq-plugins enable rabbitmq\_mqtt |
| Web MQTT | MQTT over web socket | rabbitmq-plugins enable rabbitmq\_web\_mqtt |

1. หลังติดตั้ง Plugin ทั้งหมดเสร็จ สั่ง Restart RabbitMQ Server โดยเข้าไปที่ Administrative Tools -> Services -> RabbitMQ คลิกขวา -> Restart



1. ทดลองเข้า Management UI ที่ [http://localhost:15762 เพื่อ](http://localhost:15762%20เพื่อ)ทดสอบ Management WebUI

A screenshot of a computer

Description automatically generated with medium confidence

โดยใช้ User : guest และ Password : guest ในการ Login เข้าใช้งาน

1. ใน Tab Overview จะเห็น Protocol ที่เราเปิดใช้ไปแล้วรวมถึง Port ที่เรียกใช้งานได้

A screenshot of a computer

Description automatically generated with medium confidence

1. ในการเข้าถึง Protocol ต้องการ เราต้องทำการเปิด Windows Firewall ใน Port ต่างๆด้วย
2. ไฟล์ Configuration ของ RabbitMQ จะอยู่ที่ %APPDATA%\RabbitMQ\

มี 2 File ที่ใช้ Config คือ

* advanced.config เป็น Default Configuration File ของ RabbitMQ เป็น Format ของ Erlang คล้ายๆ JSON
* rabbitmq.conf เป็น Configuration File รูปแบบใหม่ของ RabbitMQ เป็น Format คล้ายๆ INI ซึ่งปกติจะไม่มีใน %AppData%\RabbitMQ ตั้งแต่แรก ต้องสร้างใหม่

A screenshot of a computer

Description automatically generated with low confidence

ไฟล์ Config ของ RabbitMQ คือ advanced.config และ rabbitmq.conf

A picture containing text, screenshot, font

Description automatically generated

รูปแบบ Configuration ของ rabbitmq.conf รูปแบบใหม่ คล้ายๆ INI โดย 1 บรรทัดคือ 1 option

A screen shot of a computer

Description automatically generated with medium confidence

รูปแบบ Configuration ของ rabbitmq.conf รูปแบบเดิม (Classic) ตามสไตล์ erlang คล้ายๆ JSON

1. การอนุญาตให้ User เข้าใช้งาน Service ได้จากใน Network ต้อง set loopback user โดย copy code ข้างล่างนี้ไปใส่ใน File advance.config

|  |
| --- |
| [  {rabbit,  [  {loopback\_users, []}  ]  }  ]. |

ซึ่งใน advance.config ควรมีแค่นี้ ส่วนการ setup อย่างอื่นควรไป set ที่ rabbitmq.conf จะดูง่ายกว่า

**การสร้าง Server Certificate**

โดยปกติ RabbitMQ จะรองรับ protocol MQTT3.1.1 และ WS (ปราศจาก SSL) โดยปริยาย สำหรับการเชื่อมต่อแบบ MQTT ปกติ ทั้งในงาน IOT และ WebSocket โดยที่ฝั่ง Client นั้น Subscribe เข้ามาโดย protocol ws (เป็น http ธรรมดา ไม่ได้ใช้ https ) สามารถเชื่อมต่อได้เลย ไม่ยุ่งยาก

แต่กรณีที่ Web Client ใช้ https การเชื่อมต่อระหว่าง Javascript กับ RabbitMQ Broker จำเป็นต้องเชื่อมต่อด้วย WSS (Web Secure Socket) เท่านั้น เราจึงต้อง Config TLS/SSL บนฝั่งของ RabbitMQ ก่อน

1. ติดตั้ง Chocolately ลงสู่ Powershell โดยเปิด Windows Powershell แบบ Run as administrator แล้วพิมพ์

|  |
| --- |
| Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1')) |

หลังติดตั้งเสร็จ Restart Powershell

1. ติดตั้ง python จาก Windows Store หรือติดตั้งผ่าน Powershell โดยพิมพ์

|  |
| --- |
| Choco install -y python |

1. ติดตั้ง Make ผ่าน PowerShell

|  |
| --- |
| Choco install make |

1. Download RabbitMQ tls-gen จาก Github โดยการ Clone หรือ Download ก็ได้

|  |
| --- |
| https://github.com/rabbitmq/tls-gen |

1. ใน Folder ที่ดาวน์โหลด เข้าไปที่ \Basic แล้วเปิด PowerShell หรือ Git Bash โดยพิม์คำสั่ง

|  |
| --- |
| make |

หลังจากนั้น Windows จะเรียกคำสั่ง Python เพื่อทำการ Generate SSL Certificate ซึ่ง SSL ที่ได้จะทำการ Configuration ตาม Environment ของระบบให้โดยอัตโนมัติมีอายุการใช้งาน 10 ปี โดย SSL ที่ได้จะถูก Generate ลงใน 4 Folder คือ

* Result : จะเก็บ SSL ทั้งหมดลงในนี้ที่เดียว
* Client : เป็น Certificate ฝั่ง Client ซึ่งจะใช้ตอนที่ Client สั่ง Connect มายัง Server
* Server : เป็น Certificate ฝั่ง Server ซึ่งใช้ติดตั้งที่ Server
* TestCA : อันนี้ไม่จำเป็น

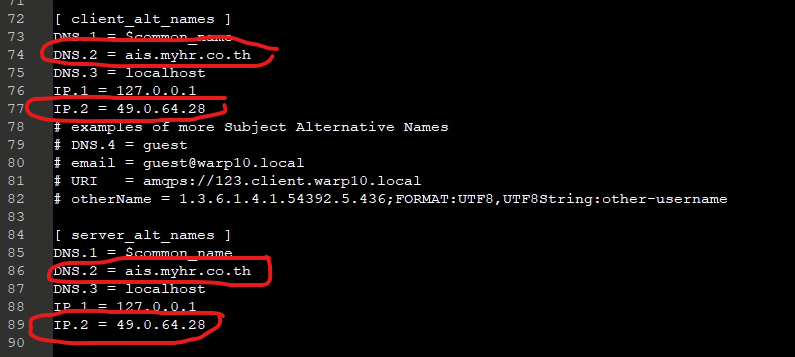
กล่าวง่ายๆคือ Certificate ทั้ง Server และ Client จะถูกรวมอยู่ที่ Folder Result

หมายเหตุ : ในระหว่างที่ สั่ง Make อาจเจอ Error บางอย่าง

1. การเรียก python3 ไม่เจอ เนื่องจาก ใน configuration ของ TLS-GEN เรียกใช้ python3 แต่ใน Windows Shell รู้จักในนาม python จึงต้องสั่ง Link windows shell

|  |
| --- |
| mklink "C:\path\to\Python3\python3.exe" "C:\path\to\Python3\python.exe" |

1. ใน TLS-GEN สามารถกำหนด alternate extension ของ Certificate ได้ ซึ่งหากไม่ได้กำหนด ข้อมูลบางอย่างเช่น ลิสต์ของ DNS, IP ให้ตรงกับข้อมูล Server ที่ใช้งานจริง หากข้อมูลใน Certificate กับ Server ไม่ตรงกันตอนเรียกใช้งานเชื่อมต่อจริงจะไม่สามารถใช้งานได้ จึงต้องกำหนดข้อมูล Alternate Extension ให้ Certificate ด้วย โดยกำหนดในไฟล์ TLS-GEN/Basic/openssl.cnf ถึงค่อยสั่ง make โดยหลังจากสั่ง make เสร็จสิ้น certificate ที่ได้จึงจะสามารถใช้งานบน Server จริงได้



ซึ่งจะเห็นได้ว่าทั้ง DNS และ IP ของ Server สามารถกำหนดได้หลายค่า ดังนั้นการ Gen ครั้งเดียวสามารถใช้ได้หลาย Server

ข้อควรสังเกต

1. CommonName (CN) ของ Server Certificate และ Client Certificate ต้องตรงกัน
2. DNS, IP ใน Alternate Extension ของทั้ง Client และ Server Certificate ต้องตรงกับ Server จริงหรือมี DSN และ IP ของ Server จริงอยู่ในลิสต์

โดยเราสามารถตรวจสอบข้อมูลภายในของทั้ง Server และ Client Certificate ได้ด้วยคำสั่ง

|  |
| --- |
| openssl x509 -in cert.pem -noout -text |

จะเห็นข้อมูลภายในของ Certificate ทั้งหมด และสามารถ Validate Private Key ได้ด้วยคำสั่ง

|  |
| --- |
| openssl rsa -in privateKey.key -check |

**การกำหนด TLS/SSL ใน RabbitMQ**

หลังจากที่เรา ได้ SSL ไว้ใช้งานต้องทำการกำหนด Configuration TLS บน RabbitMQ โดยการเปิด rabbitmq.conf ขึ้นมา ซึ่งไฟล์นี้จะถูกเก็บที่ APPDATA%\RabbitMQ\ หากไม่พบไฟล์นี้ ให้สร้างขึ้นมา

A screenshot of a computer

Description automatically generated

รูปแบบการ Config ของไฟล์ rabbitmq.conf คล้ายๆ ini โดย 1 บรรทัดคือ 1 Parameter เครื่องหมาย # นำหน้าคือ Comment

โดยให้กำหนด Config ดังนี้

|  |
| --- |
| #listeners.tcp = none  # สำหรับ SSL ของ Protocol AMQP และทั้งระบบ  listeners.ssl.default = 5671  # กำหนด Path ของ CA บนเครื่อง Server  ssl\_options.cacertfile = C:/Program Files/RabbitMQ Server/rabbitmq\_server-3.12.0/etc/Certificate/ca\_certificate.pem  # กำหนด Path ของ Server Cert ที่ได้จาก TLS-GEN  ssl\_options.certfile = C:/Program Files/RabbitMQ Server/rabbitmq\_server-3.12.0/etc/Certificate/server\_T-WindowsVM\_certificate.pem  # กำหนด Path ของ Server Key ที่ได้จาก TLS-GEN  ssl\_options.keyfile = C:/Program Files/RabbitMQ Server/rabbitmq\_server-3.12.0/etc/Certificate/server\_T-WindowsVM\_key.pem  ssl\_options.versions.1 = tlsv1.3  ssl\_options.verify = verify\_peer  ssl\_options.fail\_if\_no\_peer\_cert = true  ssl\_options.ciphers.1 = TLS\_AES\_256\_GCM\_SHA384  ssl\_options.ciphers.2 = TLS\_AES\_128\_GCM\_SHA256  ssl\_options.ciphers.3 = TLS\_CHACHA20\_POLY1305\_SHA256  ssl\_options.ciphers.4 = TLS\_AES\_128\_CCM\_SHA256  ssl\_options.ciphers.5 = TLS\_AES\_128\_CCM\_8\_SHA256  # TLS-enabled port for WSS connections at URI wss://Server:Port/ws  web\_mqtt.ssl.port = 15676  web\_mqtt.ssl.backlog = 1024  # กำหนด Path ของ CA ที่ได้จาก TLS-GEN  web\_mqtt.ssl.cacertfile = C:/Program Files/RabbitMQ Server/rabbitmq\_server-3.12.0/etc/Certificate/ca\_certificate.pem  # กำหนด Path ของ Server Cert ที่ได้จาก TLS-GEN  web\_mqtt.ssl.certfile = C:/Program Files/RabbitMQ Server/rabbitmq\_server-3.12.0/etc/Certificate/server\_T-WindowsVM\_certificate.pem  # กำหนด Path ของ Server Key ที่ได้จาก TLS-GEN  web\_mqtt.ssl.keyfile = C:/Program Files/RabbitMQ Server/rabbitmq\_server-3.12.0/etc/Certificate/server\_T-WindowsVM\_key.pem  #web\_mqtt.ssl.honor\_cipher\_order = true  #web\_mqtt.ssl.honor\_ecc\_order = true  #web\_mqtt.ssl.client\_renegotiation = false  #web\_mqtt.ssl.secure\_renegotiate = true  web\_mqtt.ssl.versions.1 = tlsv1.2  web\_mqtt.ssl.versions.2 = tlsv1.1  #web\_mqtt.ssl.ciphers.1 = ECDHE-ECDSA-AES256-GCM-SHA384  #web\_mqtt.ssl.ciphers.2 = ECDHE-RSA-AES256-GCM-SHA384  #web\_mqtt.ssl.ciphers.3 = ECDHE-ECDSA-AES256-SHA384  #web\_mqtt.ssl.ciphers.4 = ECDHE-RSA-AES256-SHA384  #web\_mqtt.ssl.ciphers.5 = ECDH-ECDSA-AES256-GCM-SHA384  #web\_mqtt.ssl.ciphers.6 = ECDH-RSA-AES256-GCM-SHA384  #web\_mqtt.ssl.ciphers.7 = ECDH-ECDSA-AES256-SHA384  #web\_mqtt.ssl.ciphers.8 = ECDH-RSA-AES256-SHA384  #web\_mqtt.ssl.ciphers.9 = DHE-RSA-AES256-GCM-SHA384  # TLS-enabled port for MQTT connections ar URI mqtts://Server:Port  mqtt.listeners.ssl.default = 8883  mqtt.listeners.tcp.default = 1883 |

มีข้อสังเกตคือใน rabbitmq.conf ไม่สามารถใช้ \ ได้ แต่เราสามารถกำหนด path ของ ssl ด้วย / แทน

หลังจากนั้นสั่ง Restart Service ของ RabbitMQ รอจนกว่า Web Management (localhost:15672) สามารถใช้งานได้

A screenshot of a computer

Description automatically generated

ตาม Config ที่ได้กำหนดจะพบว่า Port ที่ใช้ Connect ได้มีทั้ง

|  |  |  |
| --- | --- | --- |
| Protocol | Port | URI |
| AMQP | 5672 | amqp://Server:5672 |
| AMQPS | 5671 | amqps://Server:5671 |
| MQTT | 1883 | mqtt://Server:1883 |
| MQTTS | 8883 | mqtts://server:8883 |
| WS | 15675 | ws://server:15675/ws |
| WSS | 15676 | wss://Server:15676/ws |

หากเราต้องการเปลี่ยน Port ของแต่ละ Service Protocol สามารถทำได้โดยแก้ที่ rabbitmq.conf แล้วสั่ง Restart Service

1. ตรวจสอบความเรียบร้อยของระบบทั้งหมดด้วย RabbitMQ CLI โดยพิมพ์คำสั่ง

|  |
| --- |
| rabbitmqctl status |

จะต้องแสดง Information ทั้งหมดได้ โดยไม่ Error

1. หาก Error ส่วนใหญ่จะเป็นที่ erlang cookie ของ windows service และ erlang cookie ของ Current User ไม่ตรงกัน แก้ปัญหาด้วยการ เข้าไป Copy cookie จาก C:\Windows\System32\config\systemprofile\.elang.cookie ไปวางทับ %HOMEDRIVE%%HOMEPATH%\.erlang.cookie และลองสั่งเชค Status ใหม่

**วิธีการ Connect จากฝั่ง Client**

1. กรณีที่มีการเปิด Windows Firewall บนฝั่ง Server หากต้องการให้ Client Connect ได้ต้องปิด Inbound Port สำหรับ Protocol นั้นด้วย
2. การ Connect จาก Protocol ที่ไม่ใช้ TLS/SSL สามารถ Connect ได้โดยตรงจาก Client เลย

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with low confidence

1. ส่วนการ Connect ด้วย Protocol ที่ต้องการ TLS/SSL ได้แก้ mqtts และ wss ต้องกำหนด Configuration เพิ่มเติม ตามรูป

A screenshot of a computer

Description automatically generated with low confidence

ในบางครั้ง หาก connect ด้วย TLS/SSL ไม่ได้ ให้ลองเปลี่ยน TLS/SSL Version ดู ซึ่งหากไม่กำหนด Version จะเป็น Auto โดยปริยาย และส่วนใหญ่จะ Connect ไม่ได้ จากที่ลองให้ใช้ Version TLS1.1 จะสามารถ Connect ได้