Cassia Wirepas Gateway Configuration Instructions

Release date: Oct 6, 2024

Contents

Version: EN-20241006-YJ

Introduction

1) Currently, all Cassia Wirepas enabled gateways (E1000-WP, X2000-WP, and ATX2000-WP) come with two Bluetooth radio chips. For the Wirepas enabled gateway, Cassia offers two different models, e.g.
    a. E1000-WP1 / X2000-WP1 / ATX2000-WP1 gateway: one radio chip (chip1) supports Wirepas protocol and another radio chip (chip 0) supports Bluetooth protocol.
    b. E1000-WP2 / X2000-WP2 / ATX2000-WP2 gateway: both Bluetooth radio chips are converted to support Wirepas protocol.

2) To obtain a Cassia Wirepas gateway, please contact your Cassia sales representative. At this time, customers cannot self-upgrade their existing gateway to support Wirepas protocol. This option will be available in the near future. Currently, Wirepas stack and BLE stack can only be loaded by Cassia engineers in Cassia China office and factories before shipping, therefore, it is not possible to switch the gateway between Wirepas function and Bluetooth Low Energy function in the field.

3) The Cassia Wirepas gateway will come with preconfigured dedicated firmware and container APP. **Please ONLY use the gateway firmware listed in Step 3, see below. Please don't delete or reset the container and don't delete the APP.**

4) When you require technical support, please assign this gateway to a Cassia IoT Access Controller (AC) server hosted on the Cloud. This will allow Cassia's support team to remotely troubleshoot your Wirepas gateway.

Step 1. Connect Gateway's Wi-Fi Hotspot and Login Gateway's Local Console

Please connect to the gateway's Wi-Fi hotspot with SSID: cassia-xxxxxx (the xxxxxx corresponds to the last 6 digits of the gateway's MAC address). The password of the Wi-Fi hotspot is the same as the SSID. For example, if the gateway's MAC address is "CC:1B:E0:E0:96:DC", the Wi-Fi hotspot SSID and its default password will be "cassia-E096DC".

Please open Google Chrome on your laptop, enter the gateway's default IP address 192.168.40.1, and then log in. The first time you log in, you need to set the gateway's login password. The password should include numbers, letters, and special characters. The password length should be between 8 to 20 characters. Proceed to logging into the gateway's web page.

Step 2. Network Connection Setup

Version: EN-20241006-YJ

By default, a gateway has been preconfigured to DHCP for Ethernet connection. Please plug in an Ethernet cable to the Cassia gateway and check the Ethernet IP.

In case of the Wi-Fi or cellular connection configuration, please refer to the Cassia Installation Guide:
(https://www.cassianetworks.com/download/docs/Cassia_Gateway_Installation_Guide.pdf).

Step 3. Conduct Version Check the Firmware, Container, and App

Please check whether the correct version of firmware, container, and App are installed. Note that resetting the container will remove the App and configurations.

Please download the gateway firmware of E1000 and X2000/ATX2000 from the link below:
https://www.cassianetworks.com/support/knowledge-base/router-gateway-firmware/

Please download the Wirepas APP from the link below:
https://www.cassianetworks.com/support/knowledge-base/wirepas-applications/

Please download the container version 2.0.1 from the link below:
https://www.cassianetworks.com/download/firmware/container/ubuntu_XE1000_2.0.1.tar.gz

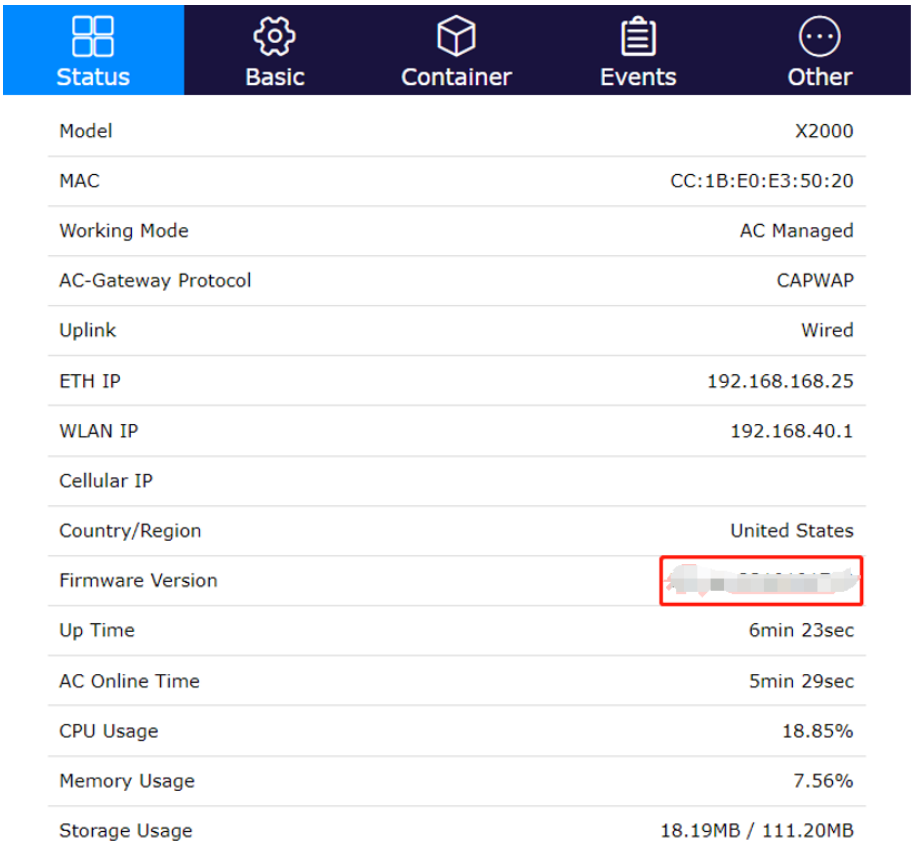

Figure 1 Check Firmware

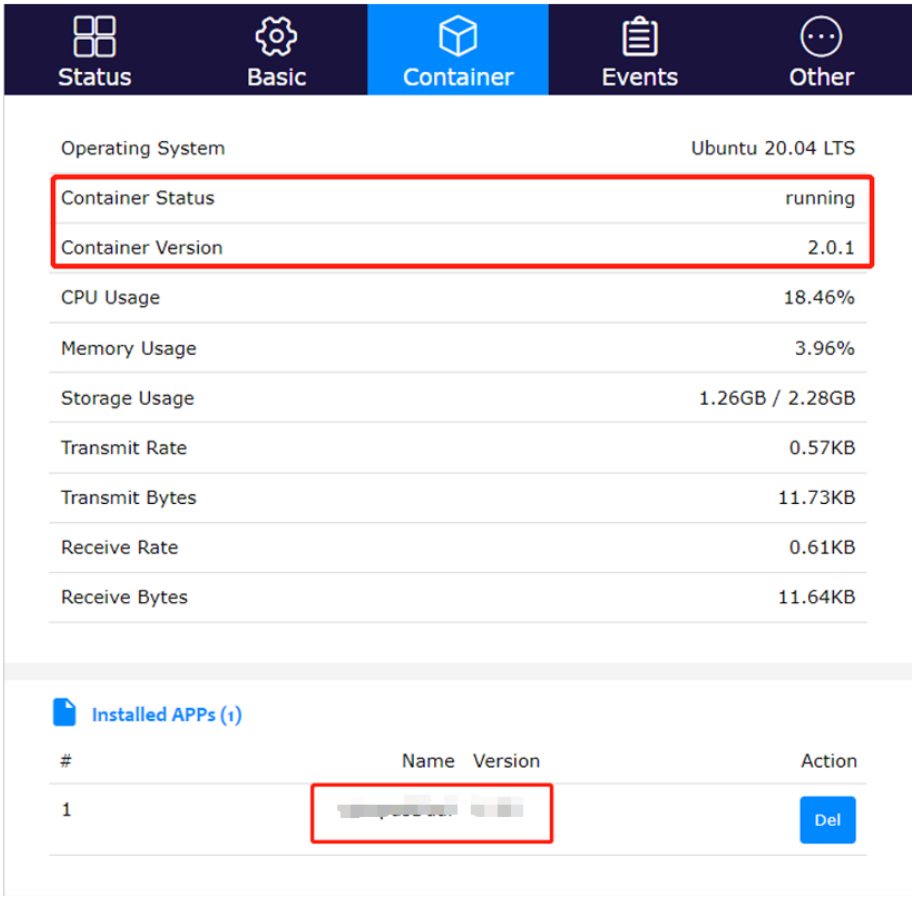

Figure 2, Check container and App

Step 4. Wirepas Configuration

Please configure Wirepas-related parameters in the Wirepas configuration section in the container tab of the AC/gateway console.

Table 1: Parameter description

Parameter	Description
Mqtt_hostname	The IP address of the Wirepas MQTT broker
Mqtt_port	The MQTT communication port to the Wirepas background. The default value is 8883 (secure) or 1883 (local)
Mqtt_username	The MQTT username
Mqtt_password	The MQTT password
Gateway_id	Each gateway in a Wirepas mesh network has a unique ID. This parameter should be a decimal number
Network_address [1,2]	The network address is used by the radio to detect valid transmissions and to filter out both the noise and other

	transmissions that do not belong to the same network. A network address must be identical for all nodes within the same network. The available Network addresses range is between 1 and 5570559, between 5636096 and 11141119, and between 11206656 and 16777215.
Network_discovery_channel [1,2]	The network channel must be identical for all nodes within the same network. The radio channel range is between 1 and 40.
Node_address [1,2]	Each node in the network must have a unique node address within the network. The two chips should have different node addresses (Node_address_2 and Node_address_2). The node address range is between 1 and 2147483647 and between 2164260864 and 4294967293.
Node_roles [1,2]	It can be configured as “sink csma-ca” or “sink”. SINK: A device that is usually connected to a server backbone. This is the final destination for all the data packets sent to the AnySink address. Similarly, all diagnostic data generated by the network itself are transmitted to a sink device. CSMA-CA mode sink: When this is enabled, the sink keeps the receiver enabled all the time when it is not transmitting. Then, the latency on sending data to the sink is way faster with the expense of higher power consumption. Intended to be used only with mains-powered devices.
Authentication_key, Cipher_key, Ca_certs	Security related parameters. When the user set parameter Authentication_key, Cipher_key to thirty-two F letters (which means FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF) the original configuration will be cleared
Output_power	It can be configured as FCC (20 dbm) for North America or ETSI (10 dbm) for Europe
Extra	For secure mode (TLS cert file provided), please configure this parameter as “--mqtt_reconnect_delay 20”. Please set the Mqtt_port parameter accordingly. For unsecured mode (TLS cert file not provided), please configure this parameter as “--mqtt_reconnect_delay 20 --mqtt_force_unsecure”. Parameter mqtt_force_unsecure will disable the TLS handshake and establish connections to the unsecured port (default: 1883). Please remember to set the Mqtt_port parameter accordingly

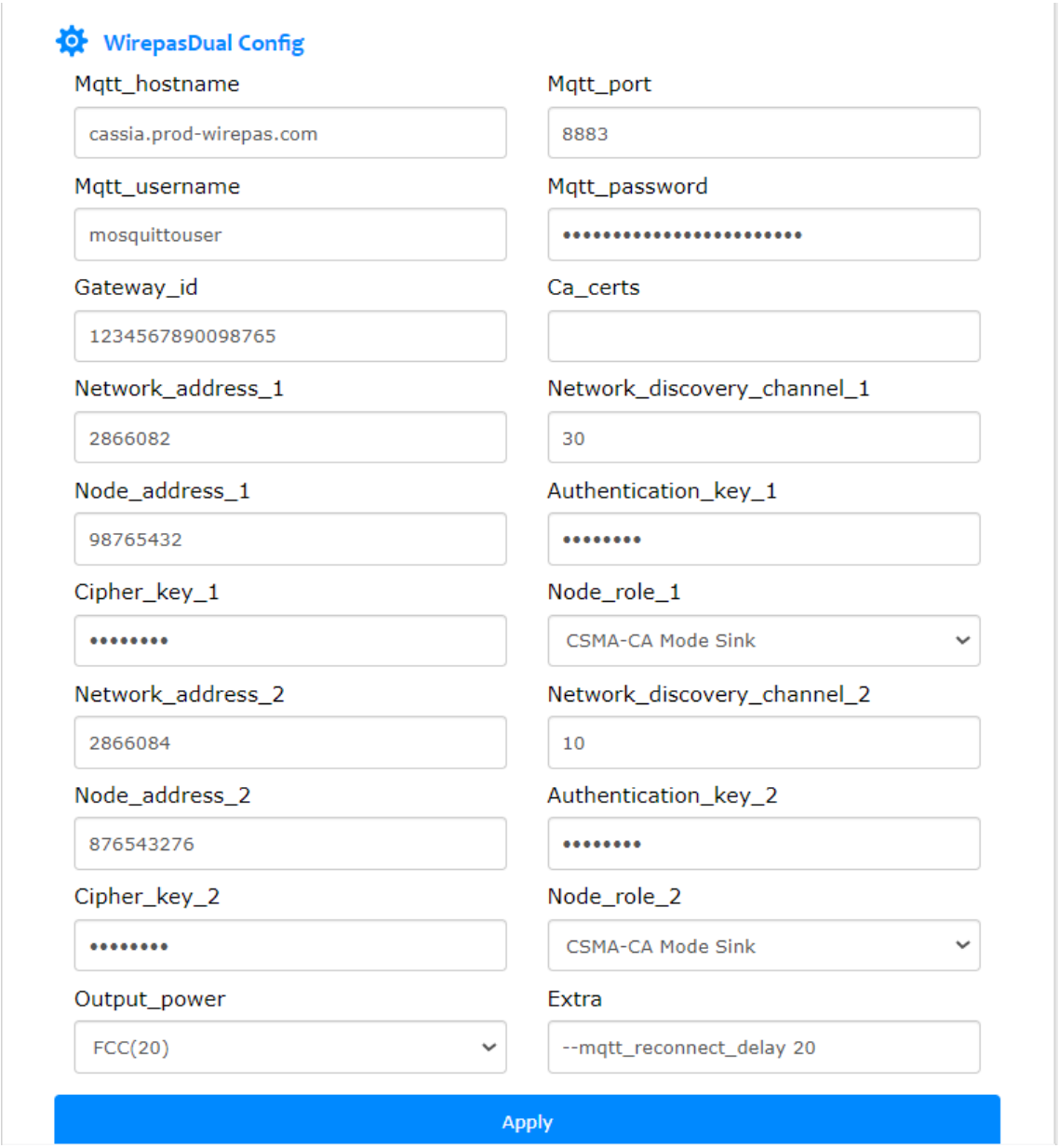

Figure 3, Dual Wirepas chip version gateway example (WP2)

Step 5. Check Sensor Data in the Wirepas Network Tool (WNT)

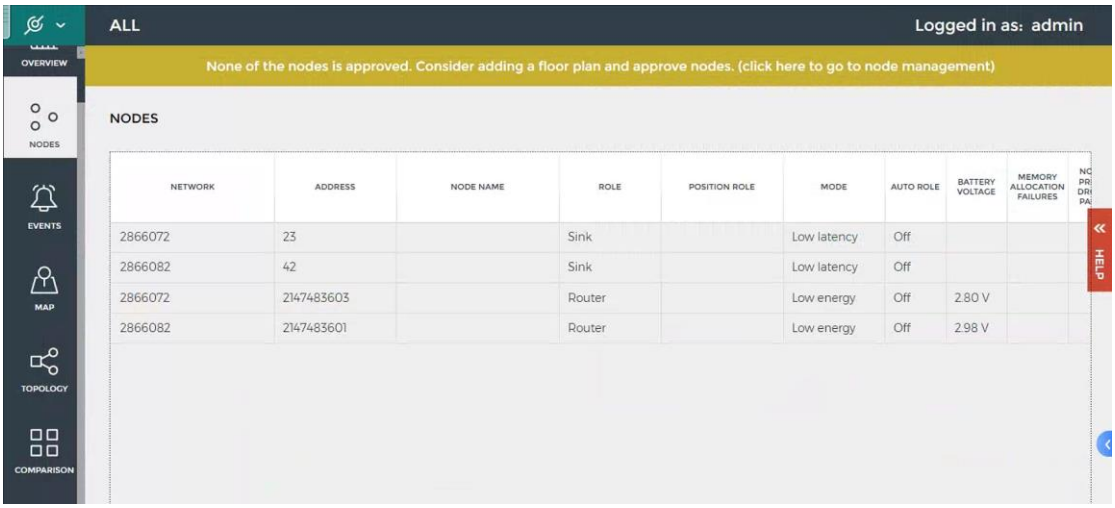

Figure 4, WNT Tool, Nodes view

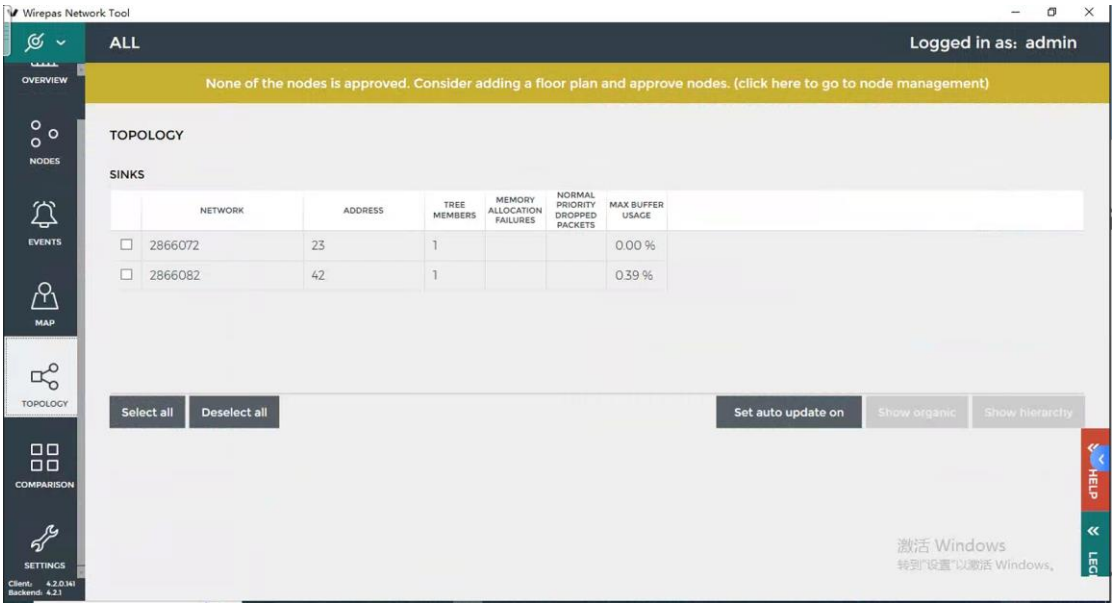

Figure 5, WNT Tool, Topology view

Please refer to link https://developer.wirepas.com/support/solutions/articles/77000499190-wirepas-network-tool-v4-client-user-guide#Nodes-View-Screen for WNT details.

Troubleshooting

- **If a customer resets the gateway by mistake**
  Please login to the local console and reconfigure the network configuration, review Step 1 and Step 2 above. Resetting the gateway will not remove or reset the Wirepas container APP.

- **If a customer deletes the container by mistake**
  Please download container 2.0.1 and then re-install the container (https://www.cassianetworks.com/download/firmware/container/ubuntu_XE1000_2.0.1.tar.gz).

  For container installation instructions, please check Chapter 5 of the Custom Application Deployment Instructions (https://www.cassianetworks.com/download/docs/Cassia-Custom_Application_Deployment_Instructions_v2.0.1.pdf)

- **If a customer resets the container, or deletes the container App**
  Please download the Wirepas container App first (see link in Step 3), install the App again, and configure parameters according to Step 4. Check the gateway installation guide (https://www.cassianetworks.com/download/docs/Cassia_Gateway_Installation_Guide.pdf).

Appendix A: Self-Signed Certificate Configuration for WNT4

1. Install Oracle WM VirtualBox 7.0

2. Install Ubuntu 20.04 on WM VirtualBox
   Please refer to Wirepas Services Installer for WNT4 User Guide for details:
   https://developer.wirepas.com/support/solutions/articles/77000499208-wirepas-services-installer-for-wnt4-user-guide

3. Install WNT backend version 4.3 in Ubuntu
   - Copy the WNT backend package to Ubuntu
     scp <name of the installer>.tar.gz YourVirtualMachineUsername@YourVirtualMachineIpaddress:/home/YourVirtualMachineUsername/
   - Uncompress the package
     tar -xzf <name of the installer>.tar.gz
   - Edit ansible/setup_host.yml file with nano text editor (command given below):
     - Set the username in the field **"instance_admin_user"** corresponding to your Ubuntu machine username. This user must have admin rights. A Ubuntu User usually has the correct rights to execute the Wirepas Service installer commands.
     - Set **'self_signed'** in the certificates field.
     - No email is needed for this type of installation
     - If missing the WPE license, set the field **"wpe_version"** to "**None**".

```
weijun@weijun:~/ansible/vars$ cat setup_host.yml
---
 ansible_python_interpreter: /usr/bin/python3

 instance_admin_user: "weijun" # your remote install admin user
 ansible_ssh_private_key_file: "./keys.pem" # private key to access remote

 # If you wish to skip wnt or wpe installation set wnt/wpe_version to None or empty
 wnt_version: "4.3"
 wpe_version: "None"

 # report and credentials
 encrypt_local_files: false
 wnt_storage: "{{playbook_dir}}/wnt" # wnt files will be copied here
 wpe_storage: "{{playbook_dir}}/wpe" # wpe files will be copied here

 # These variables are populated from your host name. They assume it
 # follows the pattern: name.domain.country
 instance_split: "{{inventory_hostname.split('.')}}"
 instance_name: "{{instance_split[0]}}"
 instance_dns_zone: "{{ '.'.join(instance_split[1:]) }}"
 instance_domain_name: "{{inventory_hostname}}"

 # Letsencrypt: Default solution installs cert automatically for your backend generated by letsencrypt.org
 # Pregenerated means that you provide cert bundle which you have bought or generated yourself and placed in
 # the path described in wnt_keychain: parameter
 #
 # Self signed certs are not recommended for production env.
 # It is much faster and easier in terms of configuration to use trusted certificate from
 # trusted certificate provider.
 #
 # It creates a new self-signed certificate to server and client and binds the created bundle.pem to
 # wnt_keychain.
 # With self signed backend you need to copy the client_bundle.pem from WNT backend into your WIN machine
 # and add it as trusted root certificate authorities in Microsoft Management Console
 # This enables WNT-Client to be able to connect to this this backend with self signed certificate.
 # NOTE: you should modify playbooks/roles/certs/defaults/main.yml according to the your
 # backend server and organization
 #
 # No cert means that backend is unsecure and no cert is needed. Not for production!

 # valid choises are [letsencrypt, pregenerated, self_signed, no_cert]
 certificate: "self_signed"
 # your domain certificate to allow haproxy's TLS handshake
 # letsencrypt and self_signed cert options owerwrite this variable
 wnt_keychain: "./extwirepasbundle.pem"

 # this field is mandatory when letsencrypt certificate is selected. owner for the domain certificate
 owner_email:
```

Figure 6, Configurations 1

- *Edit vars/aws_account.yml with nano*
  (# fill aws_access_key_id/aws_secret_access_key).

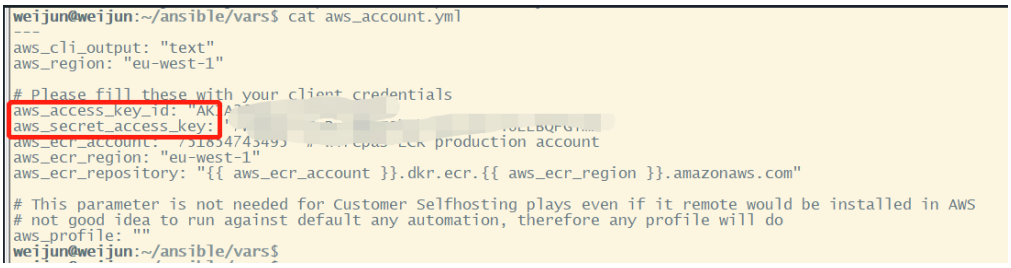

Figure 7, Configurations 2

- *sudo apt install -y ansible*
- *ansible-playbook --connection=local -i yourdomainforthisinstance, cli_setup_host.yml --tags wm-host --ask-become-pass -v*
- (#Enter your user password, setup is done, machine reboots, reconnect with ssh and continue installation)
- *cd ansible/*
- *ansible-playbook --connection=local -i yourdomainforthisinstance, cli_setup_host.yml --tags services -v*
- *docker ps* (#ensure everything is running)

NOTE: Other files except for aws_account.yml and setup_host.yml , will use the Wirepas default configuration.

4. Download customer_report.rst and client_bundle.pem from WNT Backend；
   User should execute Windows command. Command shown as:
   - ✧ *scp username@instanceipaddress:/home/username/wnt/customer_report.rst destinationFolder*
   - ✧ *scp username@instanceipaddress:/home/username/ansible/ temp_certs_***/client_bundle.pem destinationFolder*

5. WNT Client Windows configuration
   - Importing certificate

   When using a self-signed certificate and TLS, the **WNT client will refuse to establish a connection**. To allow a connection to be established, User must add the root certification authority (under cert_output_dir) to windows certificate store (client_bundle.pem). User can also enable this function from the Windows administrator command prompt with following command:

   certutil -addstore -f -enterprise -user root client_bundle.pem

where the client_bundle.pem is the generated bundle file from temp_certs_* folder.

This enables the WNT-Client to be able to connect to the WNT backend with self-signed certificate.

- Configure domain

Edit hosts in C:\Windows\System32\drivers\etc and add domain for WNT backend. i.e.

10.100.157.212 exampleserver.anydomain.com

6. Configure the Cassia Gateway Container
   - SSH into Ubuntu Container of E1000 or X2000 as cassia user
     ssh –p 20022 cassia@gateway_ip
     Password is cassia_xxxxxx (Gateway MAC address last 6 digits in lower case)
   - vi /etc/hosts,
     add '10.100.157.212 exampleserver.anydomain.com'

```
root@ubuntu:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       ubuntu
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.100.157.212    exampleserver.anydomain.com
```

7. Configure Wirepas Parameters in the Cassia Container Tab

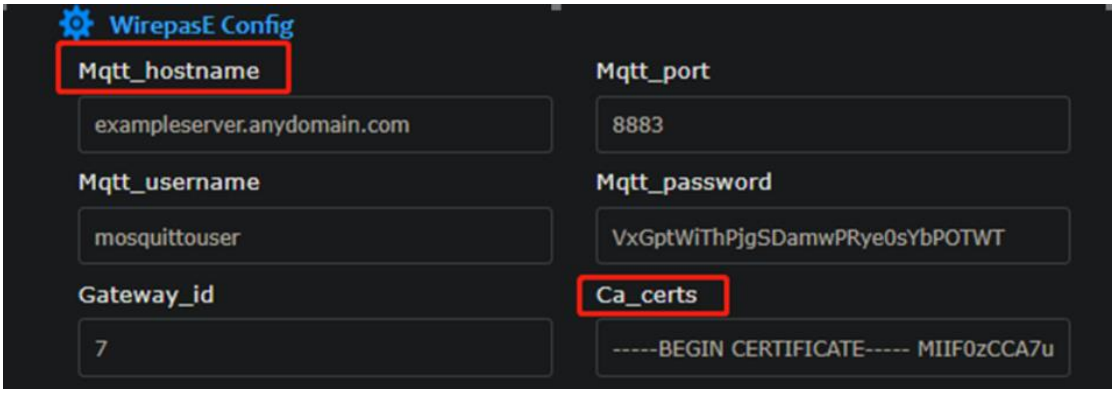

Figure 8, WirepasE Config

Mqtt_hostname： exampleserver.anydomain.com
Ca_certs：copy the content of client_bundle.pem into the box.

Appendix B: Pre-generated Certificate Configuration for WNT4

Configuration is the same with ‘self_signed’ certificate except the followings,

1. Generate certificate:
   https://gist.github.com/fntlnz/cf14feb5a46b2eda428e000157447309

2. Upload certificate to WNT backend:
   - ✧ Create a new directory with user with root permission：
     mkdir cert
   - ✧ Upload certificate to the directory:
     scp bundle.pem YourVirtualMachineUsername@YourVirtualMachineIpaddress:/home/YourVirtualMachineUsername/cert/.

3. Edit setup_host.yml file
   - ✧ Set ‘pregenerated’ in the certificates field.
   - ✧ Overwrite the Variable “wnt_keychain” with “/home/weijun/cert/bundle.pem”.

```
weijun@weijun:~/ansible/vars$ cat setup_host.yml
---
  ansible_python_interpreter: /usr/bin/python3

  instance_admin_user: "weijun" # your remote install admin user
  ansible_ssh_private_key_file: "./keys.pem" # private key to access remote

  # If you wish to skip wnt or wpe installation set wnt/wpe_version to None or empty
  wnt_version: "4.3"
  wpe_version: "None"

  # report and credentials
  encrypt_local_files: false
  wnt_storage: "{{playbook_dir}}/wnt" # wnt files will be copied here
  wpe_storage: "{{playbook_dir}}/wpe" # wpe files will be copied here

  # These variables are populated from your host name. They assume it
  # follows the pattern: name.domain.country
  instance_split: "{{inventory_hostname.split('.')}}"
  instance_name: "{{instance_split[0]}}"
  instance_dns_zone: "{{ '.'.join(instance_split[1:]) }}"
  instance_domain_name: "{{inventory_hostname}}"

  # Letsencrypt: Default solution installs cert automatically for your backend generated by letsencrypt.org
  # Pregenerated means that you provide cert bundle which you have bought or generated yourself and placed in
  # the path described in wnt_keychain: parameter
  #
  # Self signed certs are not recommended for production env.
  # It is much faster and easier in terms of configuration to use trusted certificate from
  # trusted certificate provider.
  #
  # It creates a new self-signed certificate to server and client and binds the created bundle.pem to
  # wnt_keychain.
  # With self signed backend you need to copy the client_bundle.pem from WNT backend into your WIN machine
  # and add it as trusted root certificate authorities in Microsoft Management Console
  # This enables WNT-Client to be able to connect to this this backend with self signed certificate.
  # NOTE: you should modify playbooks/roles/certs/defaults/main.yml according to the your
  # backend server and organization
  #
  # No cert means that backend is unsecure and no cert is needed. Not for production!

  # valid choises are [letsencrypt, pregenerated, self_signed, no_cert]
  certificate: "pregenerated"
  # your domain certificate to allow haproxy's TLS handshake
  # letsencrypt and self signed cert options owerwrite this variable
  wnt_keychain: "/home/weijun/ca/bundle.pem"

  # this field is mandatory when letsencrypt certificate is selected. owner for the domain certificate
  owner_email:
```

Figure 9, Setup_host

Appendix C: Configuration Letsencrypt Certificate for WNT4

WNT backend on cloud for Letsencrypt certificate are provided by Wirepas.

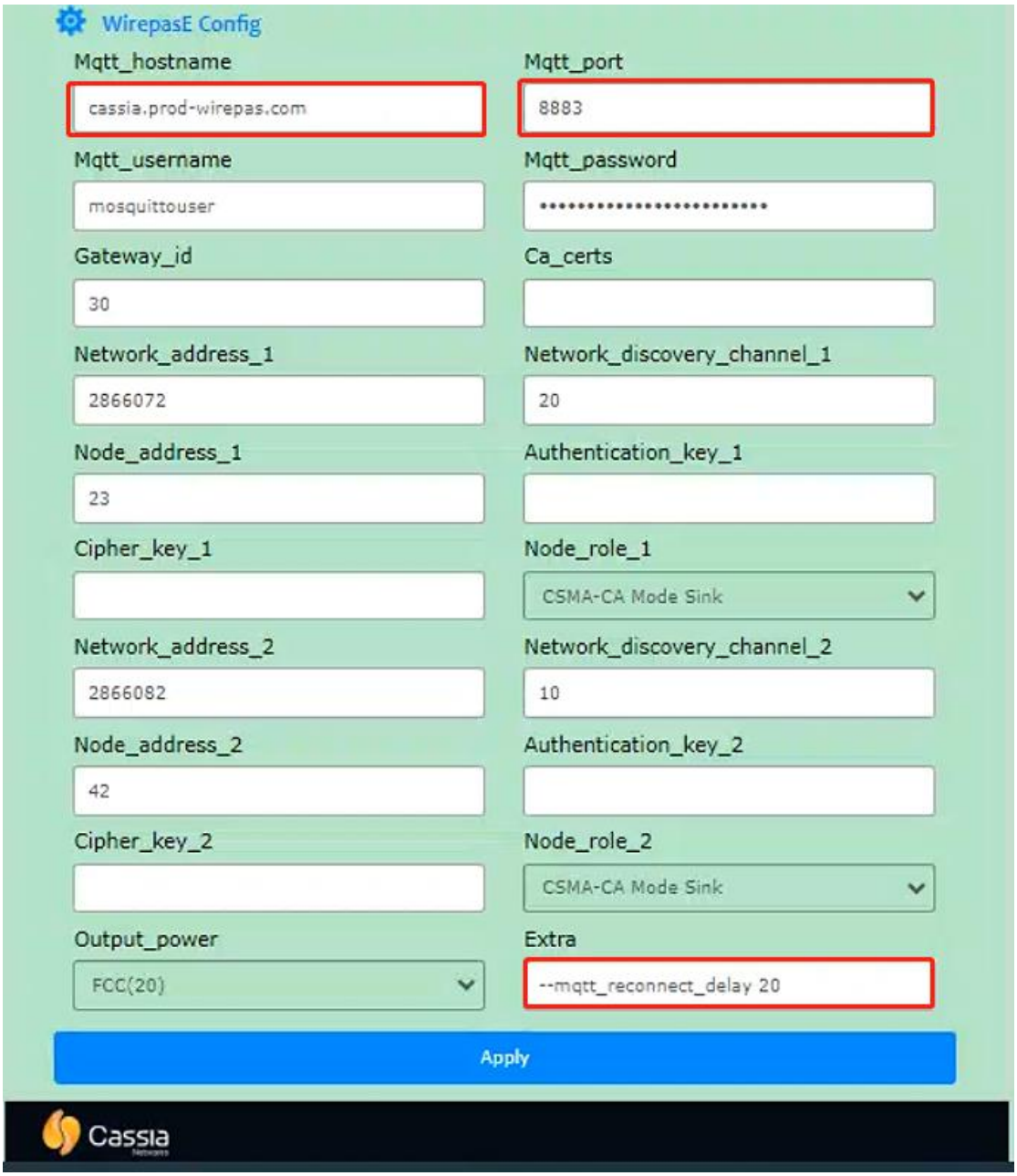

Figure 10, Gateway Configuration