



# Cassia Wirepas Gateway Configuration Instructions

Release date: Dec 5, 2023

## Contents

Introduction.....	2
Step 1. Connect Gateway's Wi-Fi Hotspot and Login Gateway's Local Console .....	2
Step 2. Network Connection Setup.....	3
Step 3. Conduct Version Check the Firmware, Container, and App .....	3
Step 4. Wirepas Configuration.....	5
Step 5. Check Sensor Data in the Wirepas Network Tool (WNT) .....	8
Troubleshooting.....	8
Appendix A: Self-Signed Certificate Configuration for WNT4 .....	10
Appendix B: Pre-generated Certificate Configuration for WNT4.....	13
Appendix C: Configuration Letsencrypt Certificate for WNT4 .....	14

## 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

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](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 and the APP from the link below:

<https://home.mycloud.com/action/share/bebdc792-2220-45a9-8a43-6f98bd59390b>

- E1000-WP gateway firmware: E1000\_2.1.2303wp2.2308171812.gz.gpg
- X2000-WP and ATX2000-WP gateway firmware: X2000\_2.1.wp.2311160813.gz.gpg
- APP version for dual chip model: wirepasDual.1.24.tar.gz
- APP version for one chip model: wirepasOne.1.22.tar.gz

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](https://www.cassianetworks.com/download/firmware/container/ubuntu_XE1000_2.0.1.tar.gz)

<div style="display: flex; justify-content: space-between; align-items: center; padding: 5px;"> <span>Status</span> <span>Basic</span> <span>Container</span> <span>Events</span> <span>Other</span> </div>	
Model	X2000
MAC	CC:1B:E0:E3:50:20
Working Mode	AC Managed
AC-Gateway Protocol	CAPWAP
Uplink	Wired
ETH IP	192.168.168.25
WLAN IP	192.168.40.1
Cellular IP	
Country/Region	United States
Firmware Version	2.1.wp.2310191713
Up Time	6min 23sec
AC Online Time	5min 29sec
CPU Usage	18.85%
Memory Usage	7.56%
Storage Usage	18.19MB / 111.20MB

Figure 1 Check Firmware

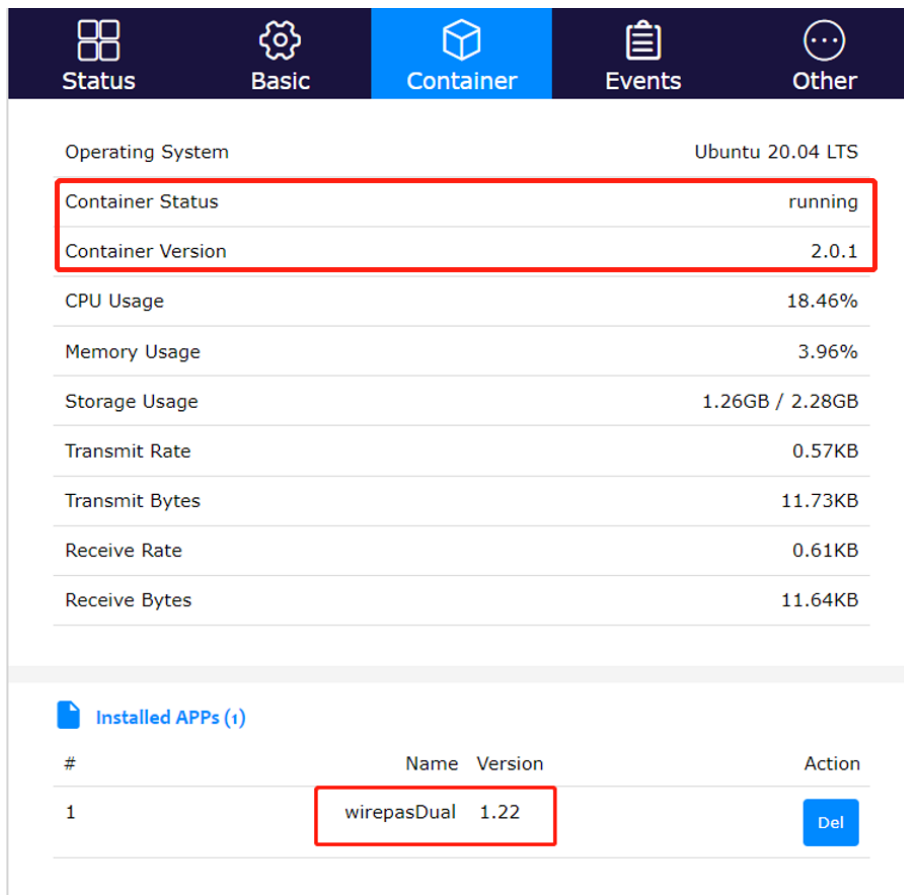


Figure 2, Check container and App

## 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	The network address is used by the radio to detect valid

[1,2]	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 chip should have different node addresses (Node_address_1 and Node_address_2). The node address range is between 1 and 2147483647 and between 2164260864 and 4294967293.
Node_roles [1,2]	<p>It can be configured as “sink csma-ca” or “sink”.</p> <p>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.</p> <p>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.</p>
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 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF) 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	<p>For secure mode (TLS cert file provided), please configure this parameter as “--mqtt_reconnect_delay 20”. Please set the Mqtt_port parameter accordingly.</p> <p>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</p>

 WirepasDual Config

Mqtt\_hostname

cassia.prod-wirepas.com

Mqtt\_username

mosquittouser

Gateway\_id

1234567890098765

Network\_address\_1

2866082

Node\_address\_1

98765432

Cipher\_key\_1

.....

Network\_address\_2

2866084

Node\_address\_2

876543276

Cipher\_key\_2

.....

Output\_power

FCC(20)

Mqtt\_port

8883

Mqtt\_password

.....

Ca\_certs

Network\_discovery\_channel\_1

30

Authentication\_key\_1

.....

Node\_role\_1

CSMA-CA Mode Sink

Network\_discovery\_channel\_2

10

Authentication\_key\_2

.....

Node\_role\_2

CSMA-CA Mode Sink

Extra

--mqtt\_reconnect\_delay 20

Apply

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

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

NETWORK	ADDRESS	NODE NAME	ROLE	POSITION ROLE	MODE	AUTO ROLE	BATTERY VOLTAGE	MEMORY ALLOCATION FAILURES	NC PR DRI PA
2866072	23		Sink		Low latency	Off			
2866082	42		Sink		Low latency	Off			
2866072	2147483603		Router		Low energy	Off	2.80 V		
2866082	2147483601		Router		Low energy	Off	2.98 V		

Figure 4, WNT Tool, Nodes view

NETWORK	ADDRESS	TREE MEMBERS	MEMORY ALLOCATION FAILURES	NORMAL PRIORITY DROPPED PACKETS	MAX BUFFER USAGE
<input type="checkbox"/> 2866072	23	1			0.00 %
<input type="checkbox"/> 2866082	42	1			0.39 %

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](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](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](https://www.cassianetworks.com/download/docs/Cassia_Gateway_Installation_Guide.pdf)).

## Appendix A: Self-Signed Certificate Configuration for WNT4

1. Install Oracle VM 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
  - o Copy the WNT backend package to Ubuntu  
`scp <name of the installer>.tar.gz  
YourVirtualMachineUsername@YourVirtualMachineIpaddress:/home/YourVirtualMachineUsername/`
  - o Uncompress the package  
`tar -xzf <name of the installer>.tar.gz`
  - o 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 choices 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 overwrite 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 ).

```

weijun@weijun:~/ansible/vars$ cat aws_account.yml
---
aws_cli_output: "text"
aws_region: "eu-west-1"

# Please fill these with your client credentials
aws_access_key_id: "AKIA..."
aws_secret_access_key: "..."
aws_ecr_account: "5194743493" # wirepas ECR production account
aws_ecr_region: "eu-west-1"
aws_ecr_repository: "{{ aws_ecr_account }}.dkr.ecr.{{ aws_ecr_region }}.amazonaws.com"

# This parameter is not needed for Customer Selfhosting plays even if it remote would be installed in AWS
# not good idea to run against default any automation, therefore any profile will do
aws_profile: ""
weijun@weijun:~/ansible/vars$

```

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
```

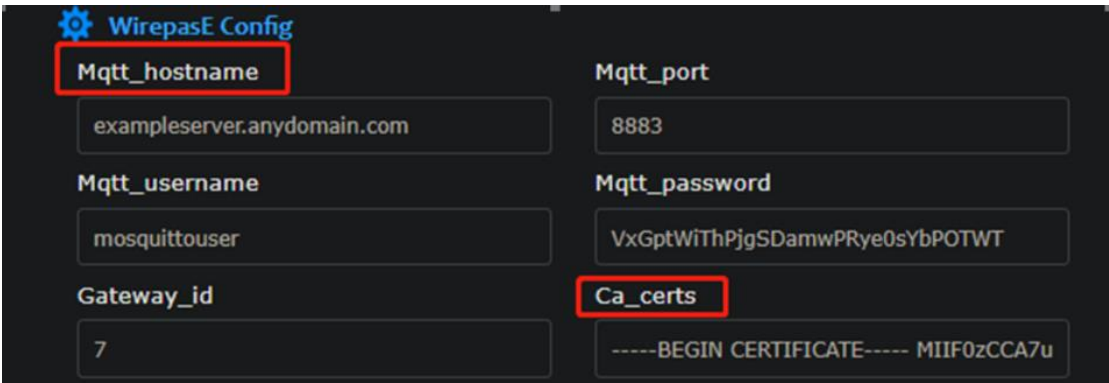
```
ff00::0 ip6-mcastprefix
```

```
ff02::1 ip6-allnodes
```

```
ff02::2 ip6-allrouters
```

```
10.100.157.212  exampleserver.anydomain.com
```

#### 7. Configure Wirepas Parameters in the Cassia Container Tab



The screenshot shows the 'WirepasE Config' interface with several fields. The 'Mqtt\_hostname' field is highlighted with a red box and contains 'exampleserver.anydomain.com'. The 'Mqtt\_port' field contains '8883'. The 'Mqtt\_username' field contains 'mosquittouser'. The 'Mqtt\_password' field contains 'VxGptWiThPjgSDamwPRye0sYbPOTWT'. The 'Gateway\_id' field contains '7'. The 'Ca\_certs' field is highlighted with a red box and contains '-----BEGIN CERTIFICATE----- MIIF0zCCA7u'.

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: "{{instance_split[0]}}"

# 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 choices are [letsencrypt, pregenerated, self_signed, no_cert]
certificate: "pregenerated"
# your domain certificate to allow haproxy's TLS handshake
# letsencrypt and self signed cert options overwrite 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.

The image shows the 'WirepasE Config' interface with various configuration fields. The fields are arranged in two columns. The left column contains: Mqtt\_hostname (cassia.prod-wirepas.com), Mqtt\_username (mosquittouser), Gateway\_id (30), Network\_address\_1 (2866072), Node\_address\_1 (23), Cipher\_key\_1 (empty), Network\_address\_2 (2866082), Node\_address\_2 (42), Cipher\_key\_2 (empty), and Output\_power (FCC(20)). The right column contains: Mqtt\_port (8883), Mqtt\_password (masked with dots), Ca\_certs (empty), Network\_discovery\_channel\_1 (20), Authentication\_key\_1 (empty), Node\_role\_1 (CSMA-CA Mode Sink), Network\_discovery\_channel\_2 (10), Authentication\_key\_2 (empty), and Node\_role\_2 (CSMA-CA Mode Sink). At the bottom right, there is an 'Extra' field containing '--mqtt\_reconnect\_delay 20'. A blue 'Apply' button is located at the bottom center. The Cassia Networks logo is at the bottom left.

Field Name	Value
Mqtt_hostname	cassia.prod-wirepas.com
Mqtt_port	8883
Mqtt_username	mosquittouser
Mqtt_password	.....
Gateway_id	30
Ca_certs	
Network_address_1	2866072
Network_discovery_channel_1	20
Node_address_1	23
Authentication_key_1	
Cipher_key_1	
Node_role_1	CSMA-CA Mode Sink
Network_address_2	2866082
Network_discovery_channel_2	10
Node_address_2	42
Authentication_key_2	
Cipher_key_2	
Node_role_2	CSMA-CA Mode Sink
Output_power	FCC(20)
Extra	--mqtt_reconnect_delay 20

Figure 10, Gateway Configuration