Cassia Networks, Inc.
97 East Brokaw Road, Suite 130
San Jose, CA 95112
support@cassianetworks.com

Cassia AC Installation Guide

Release date: March 21, 2022

Contents

1. Server Resource Requirements

Below table shows the server resource requirements for Cassia IoT AC in production deployment.

Number of Gateways	CPU	RAM	Storage
Less than 50	3Ghz * 2 core	4GB	8GB
50 to 100	3Ghz * 4 core	4GB	16GB
100 to 500	3Ghz * 4 core	8GB	32GB
500 to 5000	3Ghz * 8 core	32GB	50GB

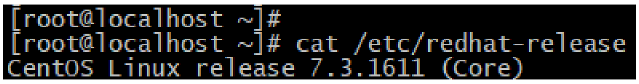
NOTE: The server resource requirements may vary depending on the way the user's application controls the Bluetooth devices, number of Bluetooth devices, the frequency of the connection setup requests, etc.

If you plan to use AWS EC2 to host your Cassia IoT AC, please select instance type T2 or M4, which uses intel CPU. For example, you can use t2.medium for an AC that manages less than 50 gateways. Please check https://aws.amazon.com/ec2/instance-types/ for all the AWS instance types.

2. Operating System

Please install CentOS 7.0 (or higher version) 64-bit Linux first. Please use the following command to check your CentOS version.

$ cat /etc/redhat-release



```
[root@localhost ~]#
[root@localhost ~]# cat /etc/redhat-release
CentOS Linux release 7.3.1611 (Core)
```

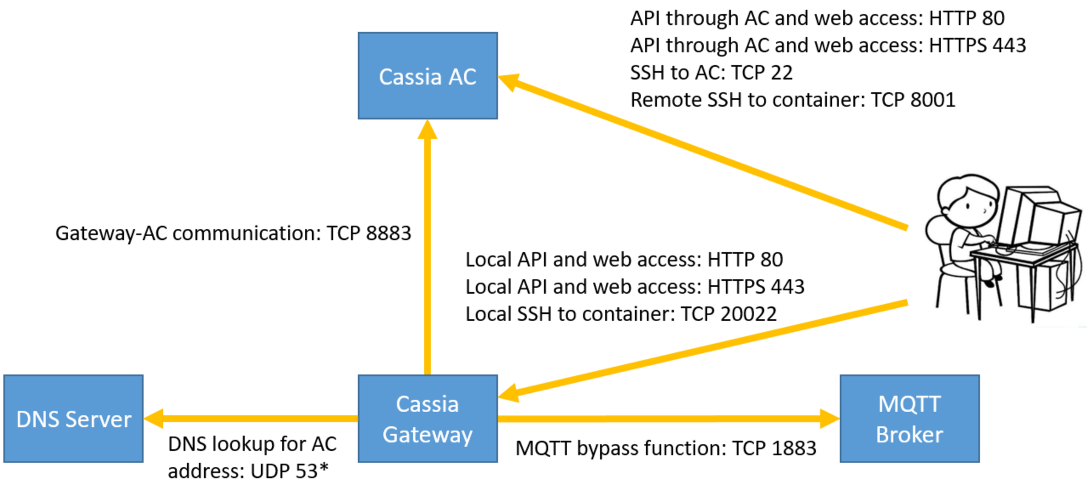
3. Port Configuration

NOTE: From version 2.0.3, a newly installed AC will support MQTT only (CAPWAP disabled by default). If the user needs to connect a version 1.4.x gateway (only supports CAPWAP) to a 2.0.3 AC, please enable the CAPWAP ports in AC side firewall and AC settings. For the AC upgraded from a lower version, both CAPWAP and MQTT will be enabled by default.

For more information about MQTT and CAPWAP, please check Cassia User Manual.

3.1. AC firewall inbound port setting

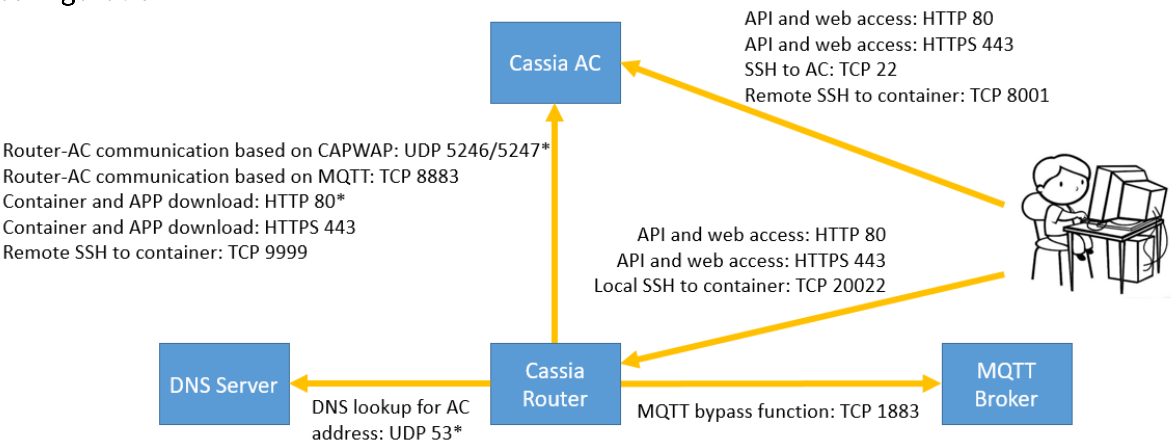
From v2.1.1, for the gateways that uses MQTT to communicate with AC (default setting), the following ports are used and required for firewall configuration. TCP ports 80, 443 and 9999 are not required anymore.


Cassia AC
API through AC and web access: HTTP 80
API through AC and web access: HTTPS 443
SSH to AC: TCP 22
Remote SSH to container: TCP 8001
Gateway-AC communication: TCP 8883
Local API and web access: HTTP 80
Local API and web access: HTTPS 443
Local SSH to container: TCP 20022
DNS Server
Cassia Gateway
MQTT Broker
DNS lookup for AC address: UDP 53*
MQTT bypass function: TCP 1883

Please open below ports inbound on the AC firewall. If you are using AWS, please go to EC2->Security groups->inbound and add the rules, e.g. allow tcp port source 0.0.0.0.

Type	Port	Description
TCP	8883	Gateway-AC communication
TCP	8001	Remote SSH to container (laptop->8001->AC<-8883<-container)

For the gateways that use CAPWAP to communicate with AC or the gateways using firmware below v2.1.1, the following ports may be used and required for firewall configuration.


Cassia AC
API and web access: HTTP 80
API and web access: HTTPS 443
SSH to AC: TCP 22
Remote SSH to container: TCP 8001
Router-AC communication based on CAPWAP: UDP 5246/5247*
Router-AC communication based on MQTT: TCP 8883
Container and APP download: HTTP 80*
Container and APP download: HTTPS 443
Remote SSH to container: TCP 9999
API and web access: HTTP 80
API and web access: HTTPS 443
Local SSH to container: TCP 20022
DNS Server
Cassia Router
MQTT Broker
DNS lookup for AC address: UDP 53*
MQTT bypass function: TCP 1883

Please open below ports inbound on the AC firewall. If you are using AWS, please go to EC2->Security groups->inbound and add the rules, e.g. allow tcp port source 0.0.0.0.

Type	Port	Description
HTTP	80	Container and APP download from AC based on HTTP. API and web access based on HTTP.
HTTPS	443	Container and APP download from AC based on HTTPS. API and web access based on HTTPS.
TCP	9999	Remote SSH to container (laptop->8001->AC<-9999<-container)
TCP	8883	Gateway-AC communication based on MQTT
TCP	8001	Remote SSH to container (laptop->8001->AC<-9999<-container)
UDP	5246, 5247	Gateway-AC communication based on CAPWAP (default port)
UDP	6246, 6247	Gateway-AC communication based on CAPWAP (backup port)

3.2. Add ports mapping on CentOS

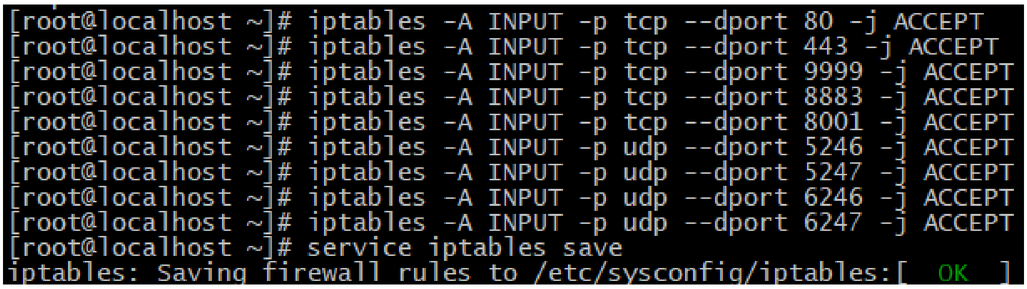
From v2.1.1, for the gateways that uses MQTT to communicate with AC (default setting), please run the below commands on the CentOS server.

```
iptables -A INPUT -p tcp --dport 8883 -j ACCEPT
iptables -A INPUT -p tcp --dport 8001 -j ACCEPT
service iptables save
```

For the gateways that use CAPWAP to communicate with AC or the gateways using firmware below v2.1.1, please run the below commands on the CentOS server.

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A INPUT -p tcp --dport 9999 -j ACCEPT
iptables -A INPUT -p tcp --dport 8883 -j ACCEPT
iptables -A INPUT -p tcp --dport 8001 -j ACCEPT
iptables -A INPUT -p udp --dport 5246 -j ACCEPT
iptables -A INPUT -p udp --dport 5247 -j ACCEPT
iptables -A INPUT -p udp --dport 6246 -j ACCEPT
iptables -A INPUT -p udp --dport 6247 -j ACCEPT
service iptables save
```

You will see below output if the above port mapping commands success.



```
[root@localhost ~]# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p tcp --dport 9999 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p tcp --dport 8883 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p tcp --dport 8001 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p udp --dport 5246 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p udp --dport 5247 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p udp --dport 6246 -j ACCEPT
[root@localhost ~]# iptables -A INPUT -p udp --dport 6247 -j ACCEPT
[root@localhost ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
```

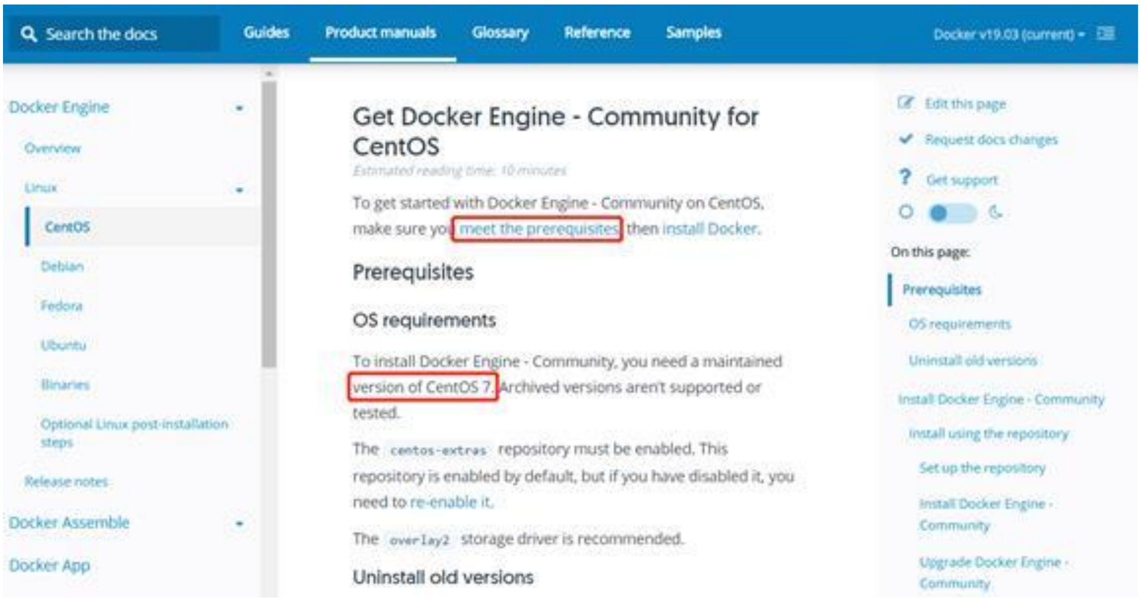
If you see error "The service command supports only basic LSB actions…", please run below commands first, and then try ports mapping again.

```
systemctl stop firewalld
systemctl mask firewalld
yum install iptables-services
systemctl enable iptables
systemctl restart iptables
```

4. Prepare to Install Docker

4.1. Make sure you meet the prerequisites

Please open https://docs.docker.com/install/linux/docker-ce/centos/ in a web browser and check if you meet the prerequisites.



Get Docker Engine - Community for CentOS

Estimated reading time: 10 minutes

To get started with Docker Engine - Community on CentOS, make sure you meet the prerequisites, then install Docker.

Prerequisites

OS requirements

To install Docker Engine - Community, you need a maintained version of CentOS 7. Archived versions aren't supported or tested.

The `centos-extras` repository must be enabled. This repository is enabled by default, but if you have disabled it, you need to re-enable it.

The `overlay2` storage driver is recommended.

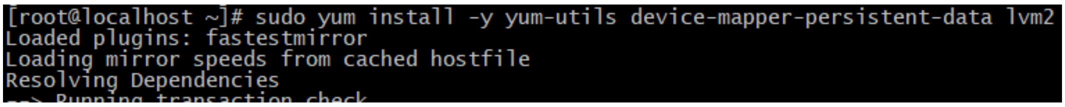
Uninstall old versions

4.2. Install Docker repository

Before you install Docker CE for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

- Install the required packages

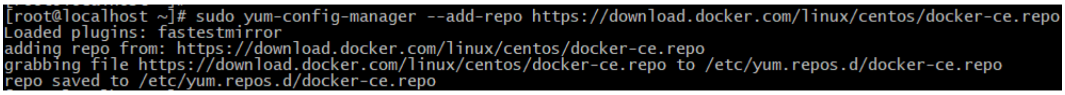
  $ sudo yum install -y yum-utils device-mapper-persistent-data lvm2



```
[root@localhost ~]# sudo yum install -y yum-utils device-mapper-persistent-data lvm2
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
```

- Setup Docker repository

  $ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo



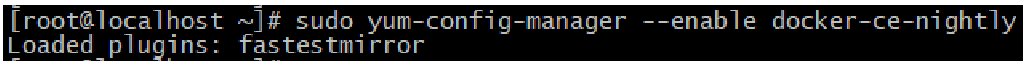
```
[root@localhost ~]# sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

- Install test software

  The test software is included in the docker.repo, but it is disabled by default. It will be used to check if Docker-CE is installed correctly.

  Please enable the test software by the below command.

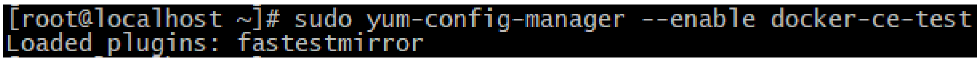
  $ sudo yum-config-manager --enable docker-ce-nightly



```
[root@localhost ~]# sudo yum-config-manager --enable docker-ce-nightly
Loaded plugins: fastestmirror
```

Please run below command to enable the test channel.

$ sudo yum-config-manager --enable docker-ce-test

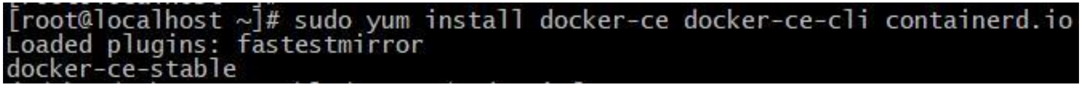


```
[root@localhost ~]# sudo yum-config-manager --enable docker-ce-test
Loaded plugins: fastestmirror
```

5. Install Docker CE

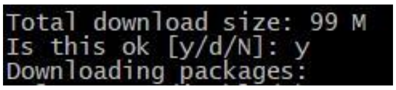
5.1. Install the latest version of Docker CE

$ sudo yum install docker-ce docker-ce-cli containerd.io

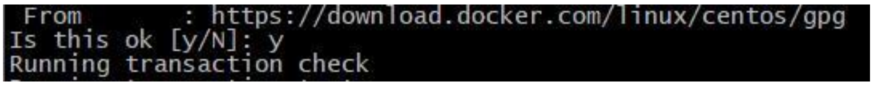


```
[root@localhost ~]# sudo yum install docker-ce docker-ce-cli containerd.io
Loaded plugins: fastestmirror
docker-ce-stable
```

Please choose Y twice during the Docker installation.



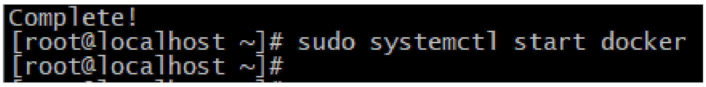
```
Total download size: 99 M
Is this ok [y/d/N]: y
Downloading packages:
```



```
 From       : https://download.docker.com/linux/centos/gpg
Is this ok [y/N]: y
Running transaction check
```

5.2. Start Docker

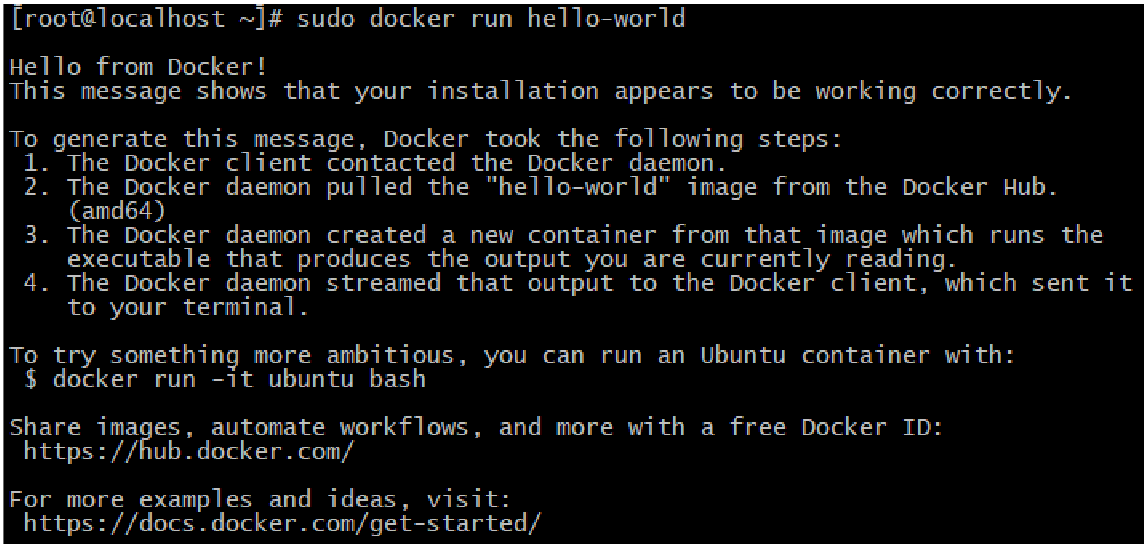
$ sudo systemctl start docker



```
Complete!
[root@localhost ~]# sudo systemctl start docker
[root@localhost ~]#
```

5.3. Verify Docker-CE is installed successfully

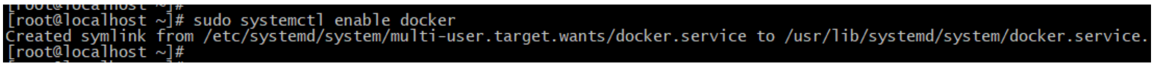
$ sudo docker run hello-world



```
[root@localhost ~]# sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

5.4. Set Docker startup with system

$ sudo systemctl enable docker



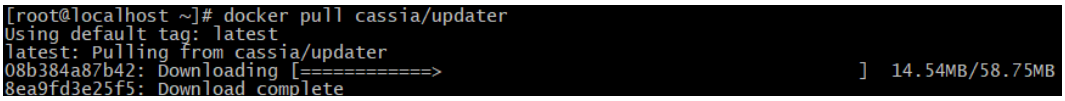
```
[root@localhost ~]# sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@localhost ~]#
```

6. Install AC

6.1. Download AC transition image (cassia-ac-updater.gz) to Docker

Option 1: direct pull the image from docker hub

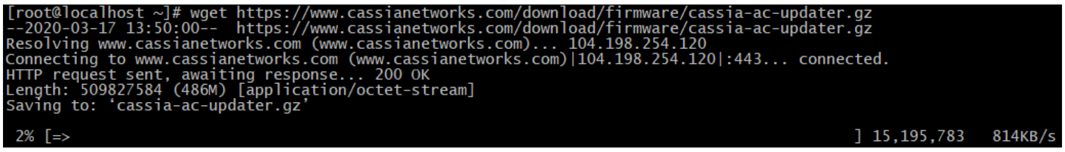
$ docker pull cassia/updater



```
[root@localhost ~]# docker pull cassia/updater
Using default tag: latest
latest: Pulling from cassia/updater
08b384a87b42: Downloading [============>                                      ]  14.54MB/58.75MB
8ea9fd3e25f5: Download complete
```

Option 2: download the image from Cassia's website

$ wget https://www.cassianetworks.com/download/firmware/cassia-ac-updater.gz



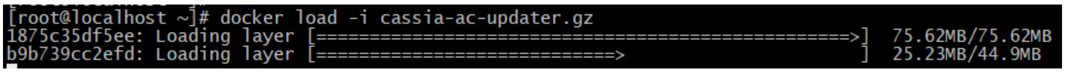
```
[root@localhost ~]# wget https://www.cassianetworks.com/download/firmware/cassia-ac-updater.gz
--2020-03-17 13:50:00--  https://www.cassianetworks.com/download/firmware/cassia-ac-updater.gz
Resolving www.cassianetworks.com (www.cassianetworks.com)... 104.198.254.120
Connecting to www.cassianetworks.com (www.cassianetworks.com)|104.198.254.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 509827584 (486M) [application/octet-stream]
Saving to: 'cassia-ac-updater.gz'

 2% [=>                                                                   ] 15,195,783   814KB/s
```

Option 3: download the image on local machine and scp to AC server

$ scp cassia-ac-updater.gz root@ip:/path/

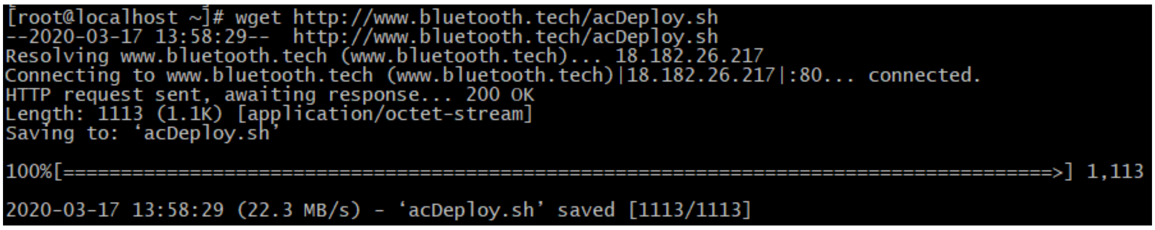
6.2. Load cassia-ac-updater image in Docker

$ docker load -i cassia-ac-updater.gz (If you used option 1 to download AC transition image, please skip this command)



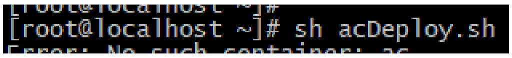
```
[root@localhost ~]# docker load -i cassia-ac-updater.gz
1875c35df5ee: Loading layer [==================================================>]  75.62MB/75.62MB
b9b739cc2efd: Loading layer [=============================>                     ]  25.23MB/44.9MB
```

$ wget http://www.bluetooth.tech/acDeploy.sh



```
[root@localhost ~]# wget http://www.bluetooth.tech/acDeploy.sh
--2020-03-17 13:58:29--  http://www.bluetooth.tech/acDeploy.sh
Resolving www.bluetooth.tech (www.bluetooth.tech)... 18.182.26.217
Connecting to www.bluetooth.tech (www.bluetooth.tech)|18.182.26.217|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1113 (1.1K) [application/octet-stream]
Saving to: 'acDeploy.sh'

100%[==========================================================================================>] 1,113

2020-03-17 13:58:29 (22.3 MB/s) - 'acDeploy.sh' saved [1113/1113]
```

$ sh acDeploy.sh

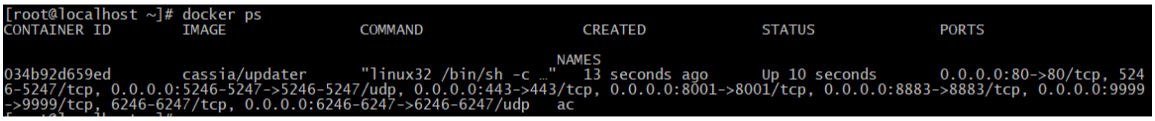


```
[root@localhost ~]#
[root@localhost ~]# sh acDeploy.sh
Error: No such container: ac
```

6.3. Check if AC is installed correctly

If you see below information after running "docker ps", it means the AC has been installed correctly.

$ docker ps

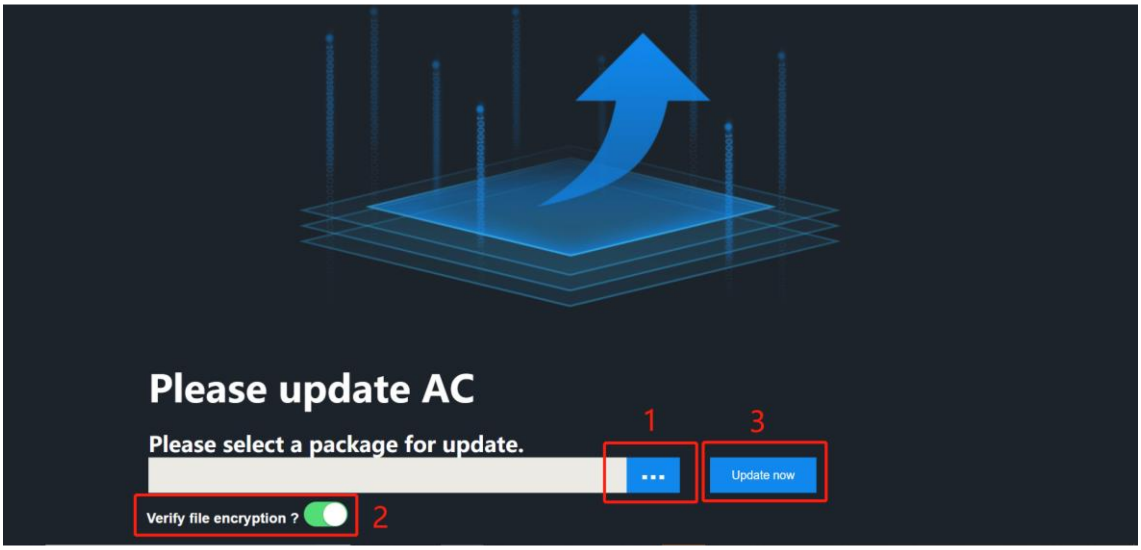


```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
                                                               NAMES
034b92d659ed        cassia/updater      "linux32 /bin/sh -c …"   13 seconds ago      Up 10 seconds       0.0.0.0:80->80/tcp, 524
6-5247/tcp, 0.0.0.0:5246-5247->5246-5247/udp, 0.0.0.0:443->443/tcp, 0.0.0.0:8001->8001/tcp, 0.0.0.0:8883->8883/tcp, 0.0.0.0:9999
->9999/tcp, 6246-6247/tcp, 0.0.0.0:6246-6247->6246-6247/udp   ac
```

6.4. Upgrade your AC to the latest version.

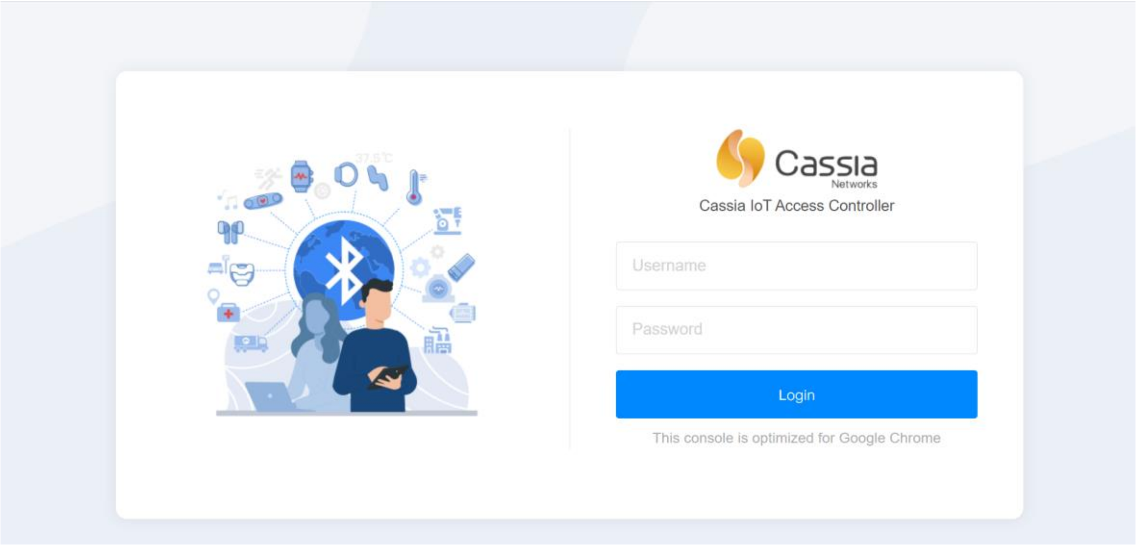
Please download the latest AC image from https://www.cassianetworks.com/support/knowledge-base/ac-server-software/ to any directory on your PC. This page is password protected, please get in touch with your Cassia sales representative for assistance.

Open “http://<server-ip>:8001/update/” on web browser to see the AC update UI. Please click “…” to select the downloaded AC image, enable “Verify file encryption?” if the image is GPG encrypted (*.gpg), and then click “Update now”. Please wait until the upgrade process finishes.



6.5. Access the AC

Now you can access your AC by entering its IP address in the web browser. The default user is **admin**, and the default password is **1q2w#E$R**. Please change the password.



Cassia
Networks
Cassia IoT Access Controller
Username
Password
Login
This console is optimized for Google Chrome

7. Configure AC

7.1. Apply License Key

If you want to manage more than three Cassia Bluetooth gateways by one Cassia IoT AC, please send below information to support@cassianetworks.com to apply License Key. The AC license key governs the number of Bluetooth gateways that can be managed by the AC and the valid time. Please set License Key in the AC setting page when you receive it.

- AC information (customer name, AC URL, etc.)
- Number of managed gateways (4 to 9999 gateways)
- Device ID (please copy from AC setting page)

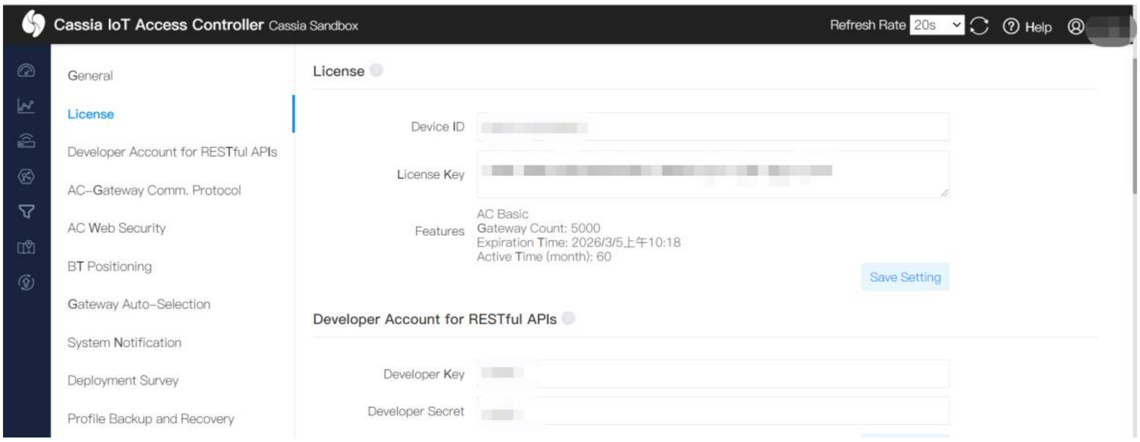
7.2. Set Developer Key and Developer Secret

Before using Cassia's RESTful API through the AC, end-users will now have to generate their own Developer Key and Developer Secret. These credentials are also intended for the end user's IoT application for OAuth 2.0 authentication towards Cassia's AC.

NOTE: For the latest version of the AC, v2.1.1, the Developer Secret Key should be between 8 to 60 characters, and must contain numbers, letters, and special characters.

For a 2.1.1 AC upgraded from older versions, the old Developer Secret key still functions, but we are strongly recommending users generate new Developer Secret Keys that match the new format stated above. Please update the Developer Secret Key used in your IoT application as well.

Please see the screenshot below for inputting the Developer Key and Developer Secret in AC setting page.


Cassia IoT Access Controller Cassia Sandbox
Refresh Rate
Help
General
License
Developer Account for RESTful APIs
AC–Gateway Comm. Protocol
AC Web Security
BT Positioning
Gateway Auto–Selection
System Notification
Deployment Survey
Profile Backup and Recovery
Save Setting

Refresh Rate: 20s

License

Device ID: [redacted]

License Key: [redacted]

Features: AC Basic
Gateway Count: 5000
Expiration Time: 2026/3/5上午10:18
Active Time (month): 60

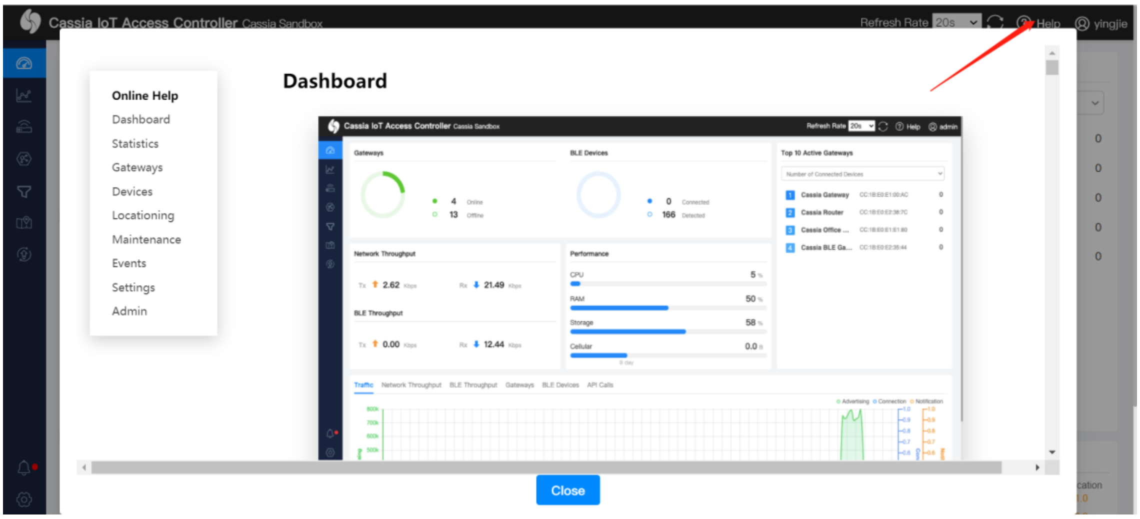
Developer Account for RESTful APIs

Developer Key: [redacted]

Developer Secret: [redacted]

7.3. Finish other AC configurations

Please finish other AC configurations according to AC online help and Cassia user manual.

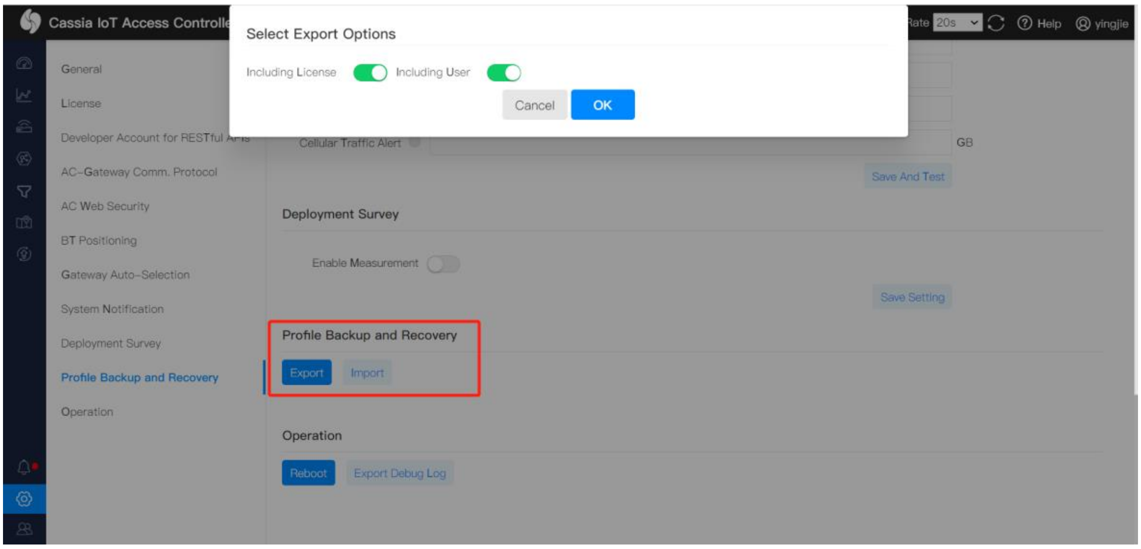


8. Backup AC Configuration

From firmware 2.0, customers can export the license, setting, user accounts, floor plan, gateway list, and roaming data of AC to a backup encrypted file. Please store the backup file in a secure manner.

Customers can recover the AC configuration by importing a backup file. **NOTE**: The backup file can’t be imported to the AC on a different server if the backup file includes AC license.

For security reasons, the Developer Key and Developer Secret is not exported to the backup file. Please input your Developer Key and Developer Secret in AC setting page after importing the backup file (you can find them in your IoT application).



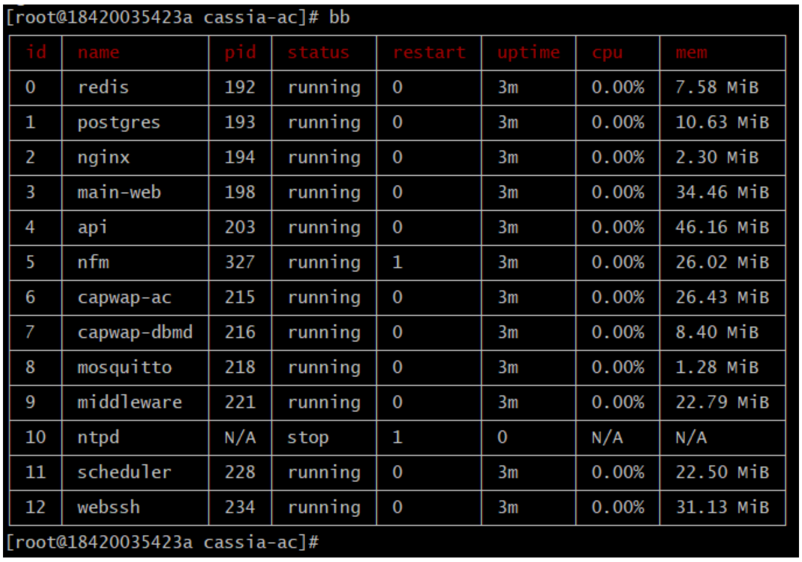
9. Trouble Shooting Tips

9.1. Check AC process

Run below command to enter the AC in Docker

$ docker exec -it acc bash

Run command “bb” to check the current running AC process. Below is an example.



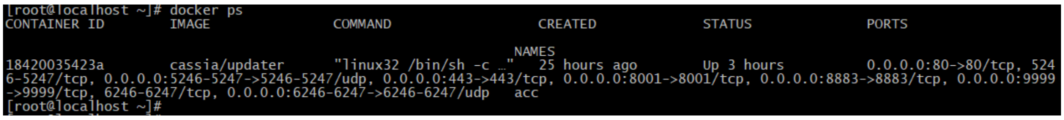
```
[root@18420035423a cassia-ac]# bb
| id | name        | pid | status  | restart | uptime | cpu   | mem       |
| 0  | redis       | 192 | running | 0       | 3m     | 0.00% | 7.58 MiB  |
| 1  | postgres    | 193 | running | 0       | 3m     | 0.00% | 10.63 MiB |
| 2  | nginx       | 194 | running | 0       | 3m     | 0.00% | 2.30 MiB  |
| 3  | main-web    | 198 | running | 0       | 3m     | 0.00% | 34.46 MiB |
| 4  | api         | 203 | running | 0       | 3m     | 0.00% | 46.16 MiB |
| 5  | nfm         | 327 | running | 1       | 3m     | 0.00% | 26.02 MiB |
| 6  | capwap-ac   | 215 | running | 0       | 3m     | 0.00% | 26.43 MiB |
| 7  | capwap-dbmd | 216 | running | 0       | 3m     | 0.00% | 8.40 MiB  |
| 8  | mosquitto   | 218 | running | 0       | 3m     | 0.00% | 1.28 MiB  |
| 9  | middleware  | 221 | running | 0       | 3m     | 0.00% | 22.79 MiB |
| 10 | ntpd        | N/A | stop    | 1       | 0      | N/A   | N/A       |
| 11 | scheduler   | 228 | running | 0       | 3m     | 0.00% | 22.50 MiB |
| 12 | webssh      | 234 | running | 0       | 3m     | 0.00% | 31.13 MiB |
[root@18420035423a cassia-ac]#
```

Process ntpd can be stopped, but all other processes should be running. If you find any process is stopped (except for ntpd) or has very high restart times, please contact Cassia support for help.

9.2. Check Docker status

Check if Docker is running with the below command.

$ docker ps



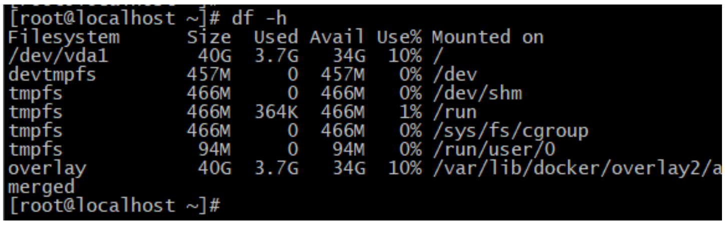
```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS

                                                               NAMES
18420035423a        cassia/updater      "linux32 /bin/sh -c …"   25 hours ago        Up 3 hours          0.0.0.0:80->80/tcp, 524
6-5247/tcp, 0.0.0.0:5246-5247->5246-5247/udp, 0.0.0.0:443->443/tcp, 0.0.0.0:8001->8001/tcp, 0.0.0.0:8883->8883/tcp, 0.0.0.0:9999
->9999/tcp, 6246-6247/tcp, 0.0.0.0:6246-6247->6246-6247/udp   acc
[root@localhost ~]#
```

If you see the above output, it means the Docker is running. IMAGE is the path of AC. PORTS are the Docker port mapping. NAME is the name of AC in Docker (please use it in other AC operations).

9.3. Check disk usage

Please use the following command to check the current disk usage

$ df -h



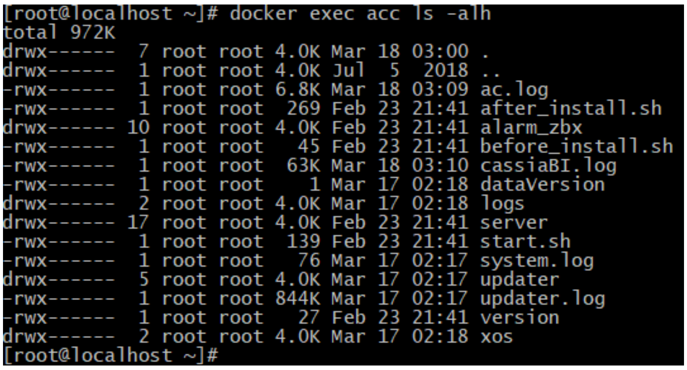
```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        40G  3.7G   34G  10% /
devtmpfs        457M     0  457M   0% /dev
tmpfs           466M     0  466M   0% /dev/shm
tmpfs           466M  364K  466M   1% /run
tmpfs           466M     0  466M   0% /sys/fs/cgroup
tmpfs            94M     0   94M   0% /run/user/0
overlay          40G  3.7G   34G  10% /var/lib/docker/overlay2/a
merged
[root@localhost ~]#
```

9.4. Clean up ac.log

In old version AC, ac.log may be very large. If the AC disk is full, you can clean up ac.log.

- Check the size of ac.log

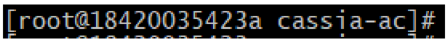
  $ docker exec acc ls -alh



```
[root@localhost ~]# docker exec acc ls -alh
total 972K
drwx------  7 root root 4.0K Mar 18 03:00 .
drwx------  1 root root 4.0K Jul  5  2018 ..
-rwx------  1 root root 6.8K Mar 18 03:09 ac.log
-rwx------  1 root root  269 Feb 23 21:41 after_install.sh
drwx------ 10 root root 4.0K Feb 23 21:41 alarm_zbx
-rwx------  1 root root   45 Feb 23 21:41 before_install.sh
-rwx------  1 root root  63K Mar 18 03:10 cassiaBI.log
-rwx------  1 root root    1 Mar 17 02:18 dataVersion
drwx------  2 root root 4.0K Mar 17 02:18 logs
drwx------ 17 root root 4.0K Feb 23 21:41 server
-rwx------  1 root root  139 Feb 23 21:41 start.sh
-rwx------  1 root root   76 Mar 17 02:17 system.log
drwx------  5 root root 4.0K Mar 17 02:17 updater
-rwx------  1 root root 844K Mar 17 02:17 updater.log
-rwx------  1 root root   27 Feb 23 21:41 version
drwx------  2 root root 4.0K Mar 17 02:18 xos
[root@localhost ~]#
```

- Enter the AC in docker

  $ docker exec -it acc bash



```
[root@18420035423a cassia-ac]#
```

  Now, the current folder changes to cassia-ac

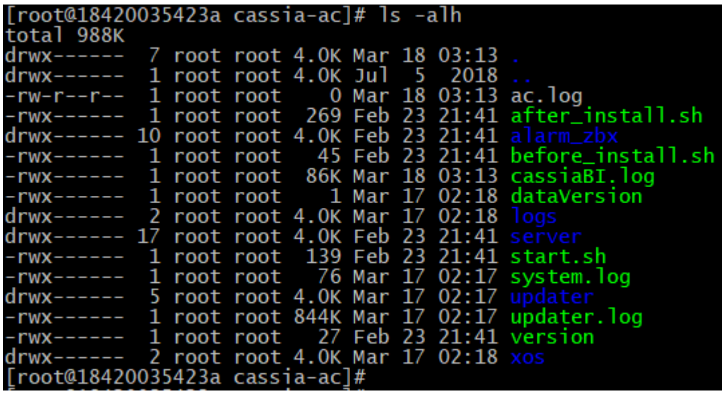
- Delete old ac.log and create a new empty ac.log

  $ rm -f ac.log

  $ >ac.log

- Check if the size of ac.log is zero

  $ ls –alh



```
[root@18420035423a cassia-ac]# ls -alh
total 988K
drwx------  7 root root 4.0K Mar 18 03:13 .
drwx------  1 root root 4.0K Jul  5  2018 ..
-rw-r--r--  1 root root    0 Mar 18 03:13 ac.log
-rwx------  1 root root  269 Feb 23 21:41 after_install.sh
drwx------ 10 root root 4.0K Feb 23 21:41 alarm_zbx
-rwx------  1 root root   45 Feb 23 21:41 before_install.sh
-rwx------  1 root root  86K Mar 18 03:13 cassiaBI.log
-rwx------  1 root root    1 Mar 17 02:18 dataVersion
drwx------  2 root root 4.0K Mar 17 02:18 logs
drwx------ 17 root root 4.0K Feb 23 21:41 server
-rwx------  1 root root  139 Feb 23 21:41 start.sh
-rwx------  1 root root   76 Mar 17 02:17 system.log
drwx------  5 root root 4.0K Mar 17 02:17 updater
-rwx------  1 root root 844K Mar 17 02:17 updater.log
-rwx------  1 root root   27 Feb 23 21:41 version
drwx------  2 root root 4.0K Mar 17 02:18 xos
[root@18420035423a cassia-ac]#
```

9.5. Restart AC in Docker

Run below commands to stop and start AC in Docker

$ docker stop acc

$ docker start acc

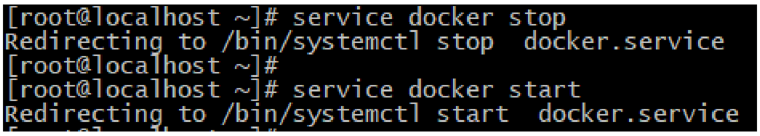
Or run below commands to restart AC in Docker

$ docker restart acc

9.6. Restart Docker service

$ service docker stop

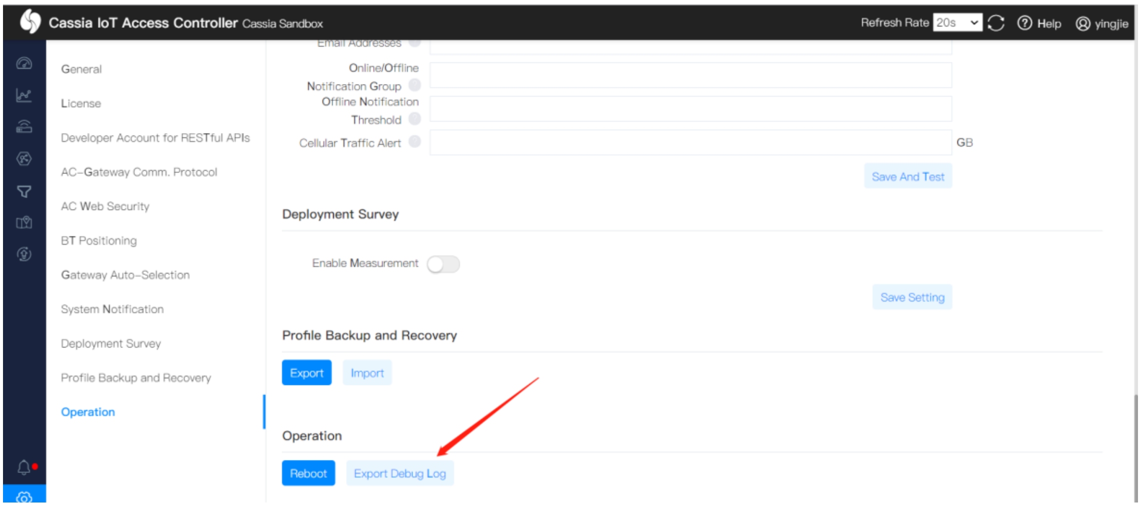
$ service docker start



```
[root@localhost ~]# service docker stop
Redirecting to /bin/systemctl stop  docker.service
[root@localhost ~]#
[root@localhost ~]# service docker start
Redirecting to /bin/systemctl start  docker.service
```

9.7. Contact Cassia Support

If you can’t fix the issue following the above tips, please contact Cassia support **support@cassianetworks.com**. Please export AC debug log and email to us.



Cassia IoT Access Controller Cassia Sandbox
Refresh Rate 20s
Help
yingjie
General
License
Developer Account for RESTful APIs
AC-Gateway Comm. Protocol
AC Web Security
BT Positioning
Gateway Auto-Selection
System Notification
Deployment Survey
Profile Backup and Recovery
Operation
Email Addresses
Online/Offline Notification Group
Offline Notification Threshold
Cellular Traffic Alert
GB
Save And Test
Deployment Survey
Enable Measurement
Save Setting
Profile Backup and Recovery
Export
Import
Operation
Reboot
Export Debug Log