



Cassia 边缘计算使用说明

V1.0

发布日期：2019 年 6 月 25 日

目录

1	名词解释.....	3
2	概述.....	3
3	注意事项.....	3
4	通过 AC 部署 Container.....	3
4.1	将 Container 上传到 AC.....	3
4.2	将 Container 批量安装到 AP.....	4
5	在 AP 上本地部署 Container.....	4
6	在 AC 上对容器的操作.....	6
6.1	查看容器状态.....	6
6.2	容器的相关操作.....	7
7	Container 中的端口转发.....	7
8	使用 SSH 登录容器.....	8
8.1	本地 SSH 登录.....	8
8.2	使用 AC 远程 SSH 登录.....	8
9	第三方 APP 打包.....	9
9.1	APP 打包命名规则.....	9
9.2	APP 包的构成.....	9
9.3	APP 打包示例.....	9
10	通过 AC 部署 APP.....	10
10.1	将 APP 上传至 AC.....	10
10.2	将 APP 批量安装到 AP.....	10
11	APP 配置定制.....	11
12	应用程序和软件包.....	12
13	Wiki 上的 APP 开发支持.....	13

1 名词解释

AC: Cassia 物联网控制器

Router/AP: Cassia 蓝牙路由器

APP: 第三方应用，用户自行开发的应用程序

Container: 容器，此处特指 Cassia 蓝牙路由器提供的 APP 安装空间

Ubuntu: 一个开源的操作系统

2 概述

从 AP 的 1.3 版本的固件开始，E1000 和 X1000 两种型号的蓝牙路由器中提供了一个容器供用户安装自行开发的 APP 使用。容器的内核为 Ubuntu 16.04.3。用户可以通过 AC 和 AP 两种方式来安装和升级容器和 APP。容器直接安装在 AP 中，与 AP 共享 CPU，存储和 RAM。该容器最多可使用 2 个 CPU 内核，最多可使用 128MB RAM 内存。

请从以下链接下载最新的容器固件,或者联系 Cassia 支持人员:

<https://www.cassianetworks.com/knowledge-base/router-gateway-firmware/>

3 注意事项

建议将容器中的内存使用率保持在 70% 以下,另外 30% 用于高峰时段和异常情况。即容器应使用少于 90MB 的内存,其中包括第三方应用程序使用的内存以及容器中运行的所有工具。

如果用户想要升级现有 APP,请确保 APP 的名称相同,版本号不同。或者,用户可以删除现有 APP 并安装新 APP。

Cassia 路由器容器内核使用的是精简版本的 Ubuntu。某些文件包可能未预先安装或者不可用。建议用户完成开发后,在 AP 的 Ubuntu 容器中生成 APP 包,生成时请保证预留足够的存储空间。

我们对用户开放容器的 SSH 和 root 访问权限。但是 Ubuntu 是作为操作系统存在的,因此无法修改它的核心,以及 sysfs 的属性,例如 /proc,是只读的。

容器重置将删除并重新创建容器。重置后只会保留 /opt 下的文件,不在 /opt 下的第三方 APP 将会被删除。

4 通过 AC 部署 Container

4.1 将 Container 上传到 AC

请在 AC 上按照以下步骤进行操作:

Maintenance 选项卡 → 容器更新 → 从 PC 上传并添加 → 浏览文件 → 确定



4.2 将 Container 批量安装到 AP

请在 AC 上按照以下步骤操作：

Routers 选项卡→选择需要安装的 AP→More→容器→安装



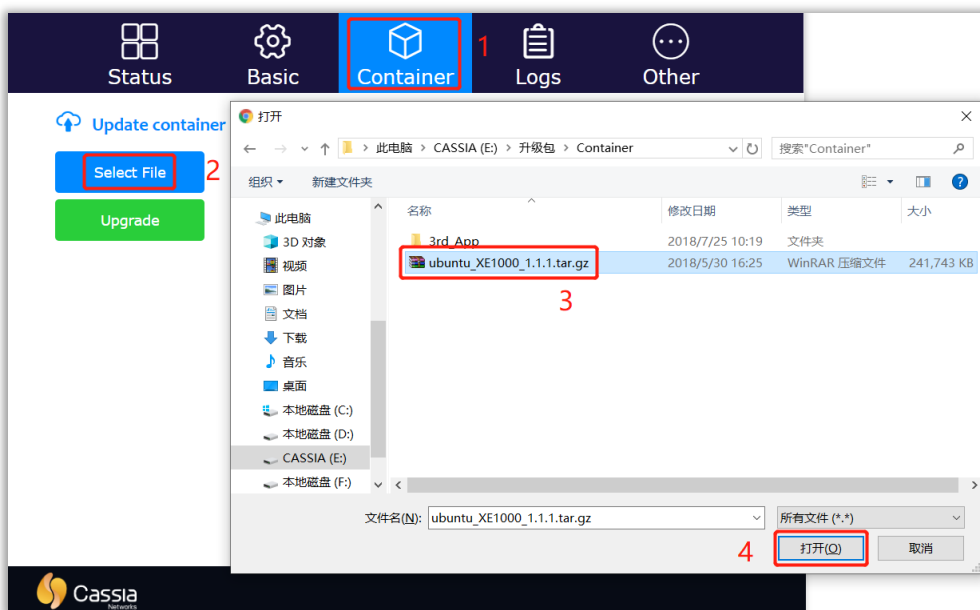
请选择容器版本，然后点击“确定”开始安装。仅可同时对相同型号的 AP 进行此操作。



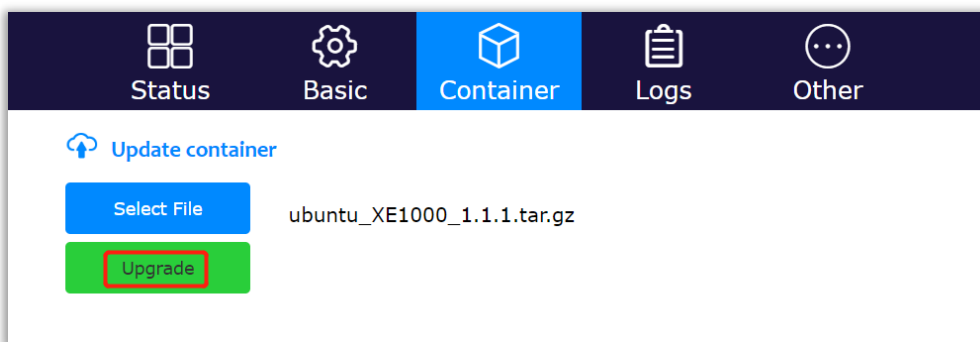
5 在 AP 上本地部署 Container

请在 AP 上，按照以下步骤操作：

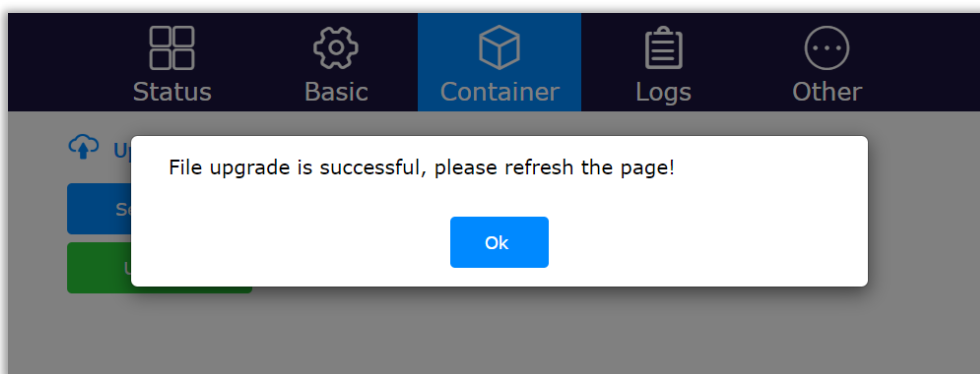
Container→选择 File→选中固件→打开。



点击“Update”开始安装

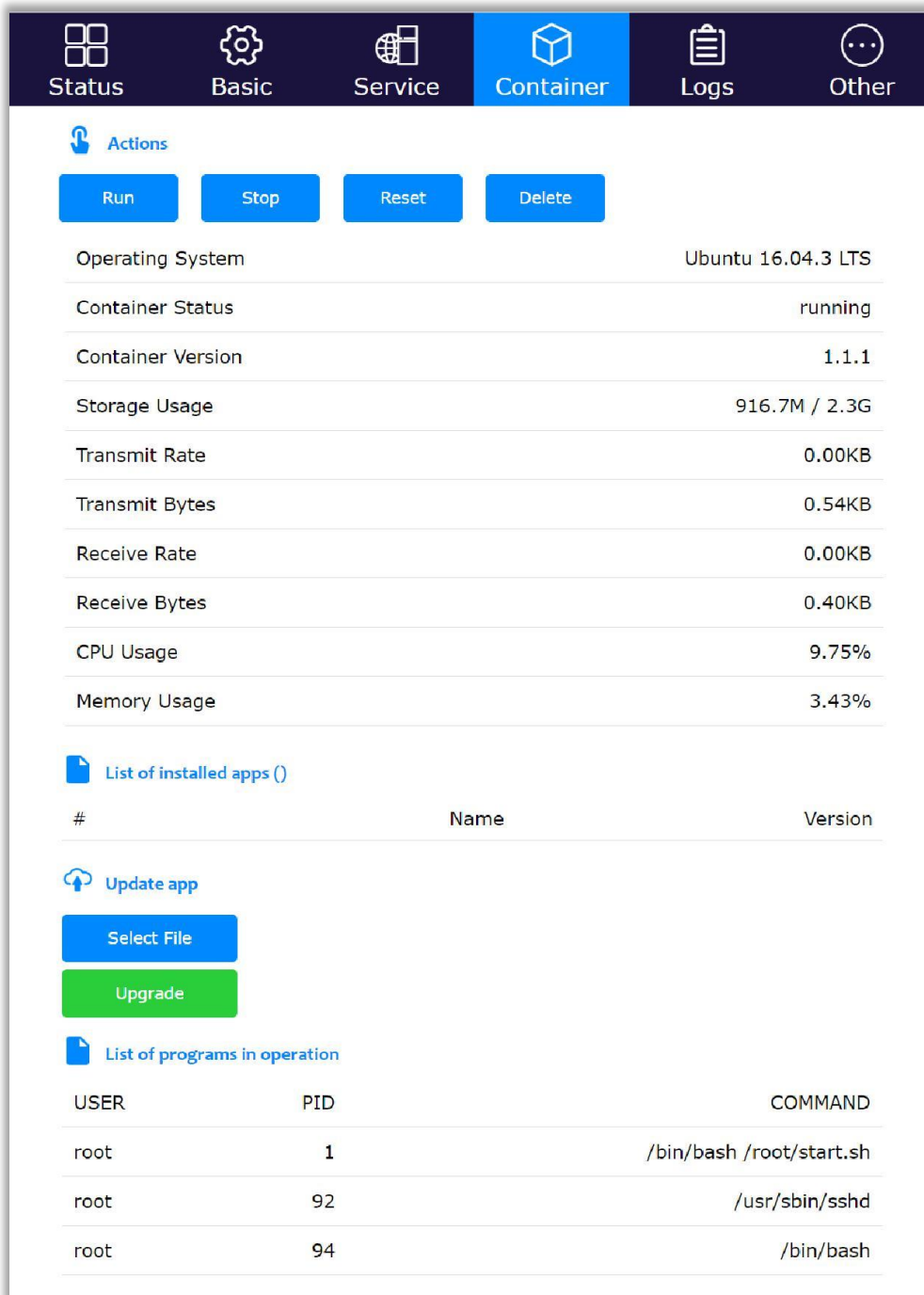


等待几分钟后，容器固件将上传并安装到 AP 中。



请刷新浏览器。然后，您将看到容器和第三方应用程序的信息。

在容器选项卡上，客户还可以上传第三方 APP 或对容器进行操作（运行，停止，重置，删除）。




6 在 AC 上对容器的操作

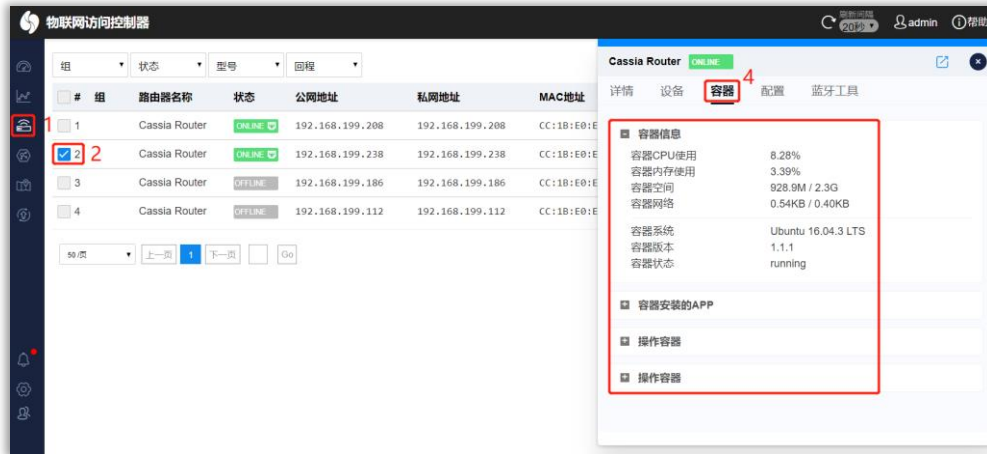
6.1 查看容器状态

用户可以在 AC 上获取并查看容器的状态和相关的信息。

请在 AC 上按照以下步骤进行操作：


Routers 选项卡 → 选择路由器 → 编辑  → 容器，即可查看容器的信息。

请注意，因固件版本不同，目前在此页面会显示两个“操作容器”选项，下文中会以“第一个”和“第二个”进行区分。



6.2 容器的相关操作

用户可以通过 AC 删除、重置、启动、停止和远程登录路由器。请在 AC 上通过以下步骤对容器进行相关操作。


Routers 选项卡 → 选择路由器 → 编辑  → 容器 → 操作容器（第一个），即可对容器进行删除、重置、停止和远程连接等操作。



7 Container 中的端口转发

从 AP 固件 1.4.1 开始，用户可以为容器的端口转发配置最多四个 TCP 或 UDP 端口。通过使用此功能，用户可以在容器中设置服务器（例如 Web 服务器），并通过路由器的专用 IP 地址和配置的端口访问它。端口范围是[60000,65525]。N/A 表示端口已关闭。

请在 AC 上通过以下步骤对容器进行相关操作。

Routers 选项卡 → 选择路由器 → 编辑  → 容器 → 操作容器（第二个），即可对容器的端口进行相关配置。

操作容器

Protocol N/A	Port <input type="text"/>
Protocol N/A	Port <input type="text"/>
Protocol N/A	Port <input type="text"/>
Protocol N/A	Port <input type="text"/>

8 使用 SSH 登录容器

8.1 本地 SSH 登录

注意：SSH 登陆使用的终端需要和 AP 在同一局域网中，使用方法请参照如下示例。x.x.x.x 是 AP 的局域网 IP，cassia/cassia 是 container 的默认用户名和密码。

使用 Linux Shell 登录示例：ssh -p 20022 cassia@x.x.x.x

使用 Xshell 登录示例：ssh cassia@x.x.x.x 20022

8.2 使用 AC 远程 SSH 登录

注意：SSH 登陆使用的终端需要和 AP 不在同一网络中，AP 需要运行在 AC 管理模式。

请在 AC 上按照以下步骤进行操作：

Routers 选项卡 → 选择路由器 → 编辑 → 容器 → 操作容器（第一个） → 远程连接



使用 SSH 登录容器后，用户可以使用 RESTful API，noble 或 dbus 控制蓝牙。也可以使用 apt-get 命令安装 Linux 工具。例如，用户可以运行“apt-get install lrzsz”来安装 lrzsz。

注意：在 AC 中，用户一次只能建立和维护一个 AP 的 SSH 会话。

9 第三方 APP 打包

9.1 APP 打包命名规则

APP 包的文件名形如 `name.major.minus.tar.gz`。

其中 `name` 为 APP 包的名称,支持的字符有大写英文字母 `A-Z`,小写英文字母 `a-z`,数字 `0-9`,下划线“`_`”,连字符“`-`”,且长度不超过 16 个字符。

`Major` 和 `minus` 为主版本号和次版本号,仅支持数组 `0-9`,且长度不超过 3 个字符。

`Tar.gz` 是固定的,声明了 APP 包的打包和压缩格式。

`name` 作为 APP 的唯一标识,具有相同 `name` 的 APP 包被程序认为是同一个 APP 的不同的版本。

`Major` 和 `minus` 区分 APP 的不同版本,版本号相同的 APP 不允许重复安装。程序只比较版本号是否相同,不比较大小。

9.2 APP 包的构成

APP 包应该包含用户需要发布的全部程序文件、配置文件和数据文件等。同时应该包含一个程序或服务启动文件 (`autorun.sh`)。

建议用户直接在 AP 的 Ubuntu 容器中制作 APP 包,制作时请保证预留足够的存储空间。

首先需要将用户希望在容器开启时自动运行的部分写入文件 `autorun.sh`。

然后将所需的其他文件按照 Ubuntu 的根文件系统层次结构,读写访问权限,用户组限,准备好

最后使用 `tar -zcvf` 命令打包。

9.3 APP 打包示例

假设用户想要生成一个名为 `abcd` 且版本为 `1.1` 的 APP。APP 包括启动脚本 `autorun.sh`,可执行程序文件 `/usr/bin/eeee`,配置文件 `/etc/ff.conf` 和文件夹 `abcd.1.1` 中的 `lib` 文件 `/usr/lib/gg.so`。

此外,从固件 `1.4.1` 开始,客户可以在文件夹 `/root/config /<app_name>/` 下添加 `meta.json` 来定义自己的 APP 配置控制台(参见第 11 章)。

文件架构如下图所示:

```
/autorun.sh
/etc/ff.conf
/usr/bin/eeee
/usr/lib/gg.so
/root/config/abcd/meta.json
```

执行打包命令 `tar -zcvf abcd.1.1.tar.gz abcd.1.1/`,系统将生成 APP 包文件 `abcd.1.1.tar.gz`。

将该 APP 安装到 AP 后，可以在容器中看到如下文件：

/etc/ff.conf
/usr/bin/eeee
/usr/lib/gg.so
/root/apps/abcd.sh
/root/apps/abcd.version
/root/config/abcd/meta.json

其中 abcd.sh 的内容和 autorun.sh 相同。

注 1： 用户的 APP 包中的文件名称不能与 ubuntu 系统中的文件同名（例如 ubuntu 系统中存在文件 /usr/bin/yes 文件，用户不能在 app 包中 /usr/bin/ 目录下添加 yes 文件）。

注 2： 用户不同的 APP 包中的文件名称不能相同（否则会替换，影响之前 APP 的使用）。

10 通过 AC 部署 APP

10.1 将 APP 上传至 AC

请在 AC 上按照以下步骤进行操作：

Maintenance 选项卡 → APP 更新 → 从 PC 上传添加 → 浏览文件 → 确定



10.2 将 APP 批量安装到 AP

请在 AC 上按照以下步骤进行操作：

Routers 选项卡 → 选择需要安装或升级的路由器 → More → 容器 → 升级 APP



在弹出的页面中，选择 APP 要安装或者升级的版本，点击确定开始安装或者升级。仅能同时操作相同型号的 AP。



11 APP 配置定制

从固件 1.4.1 开始，客户可以在 AP 的本地页面和 AC 的配置页面中添加自己的 APP 配置控制台。

首先，请在文件夹/root/config/<app_name>/下创建 meta.json。此文件定义配置项。

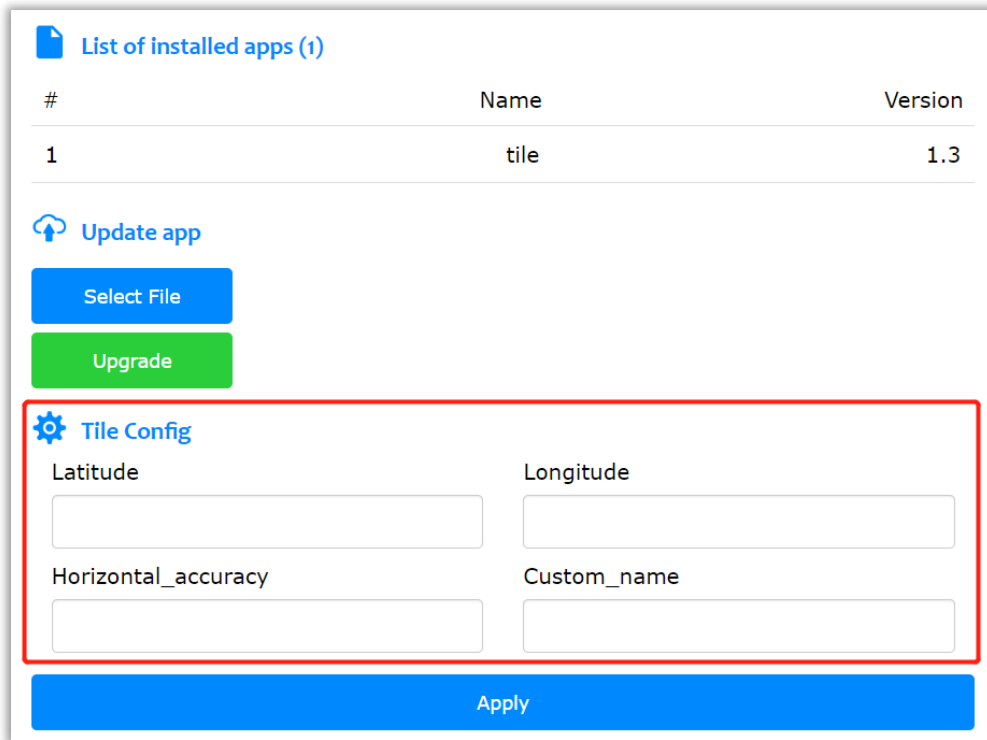
meta.json 文件格式如下：

```
{
  "name": "<app_name>",
  "config_items" : [
    {"name": "<configure_1>"},
    {"name": "<configure_2>"},
    .....
  ]
}
```

以下为某程序 meta.json 文件示例：

```
{
  "name": "Tile",
  "config_items" : [
    {"name": "Latitude"},
    {"name": "Longitude"}
    {"name": "Horizontal_accuracy"}
    {"name": "Custom_name"}
  ]
}
```

然后，在路由器的本地控制台和 AC 控制台上，客户可以找到自己的 APP 配置控制台，如下所示。



配置完成后，路由器将在 `/root/config/<app_name>/` 文件夹下生成文件 `config.json`。

12 应用程序和软件包

以下是容器版本 1.1.1 中预加载的实用工具和开发软件包的列表。

Name	Version
BlueZ	5.31
Bluetoothd	
DBus	
Python 2	2.7
Python 3	3.5.2
Python Pip	8.1.1
gobject	
dbus-python	
Nodejs	6.11.5 *
NPM	3.10.10
node-gyp	
noble	
GCC	5.4.0
G++	

* 在容器版本 1.1.1 中，Cassia 预安装了 Node.js 6.x 版本，用户可以自行升级到较新版本的 Node.js。

13 Wiki 上的 APP 开发支持

为了支持 APP 设计师构建他们的 APP，Cassia 提供了一个在线 wiki 资源：

<https://github.com/CassiaNetworks/iot-platform/wiki>。其中包括一些技术上的支持，开发环境，支持的编程语言，内置实用工具和开发包等。