

DOCUMENTATION FOR SETTING UP AND MANAGING HAWKBIT WITH RAUC

This documentation provides step-by-step instructions for building and starting the hawkBit Update Server, configuring RAUC with the hawkBit server, adding targets and distribution sets, uploading RAUC bundles, and enabling Raspberry Pi devices to download and install updates.

Prerequisites

- A Linux system with the following installed:
 - Java (Java 11 or higher)
 - Maven
 - Git
- A Raspberry Pi with:
 - RAUC installed and configured.
 - Minimal Yocto or any compatible Linux image.
- Network connectivity between the Raspberry Pi and the server running hawkBit.
- For Rauc installation and configuration, refer this link:
 -

https://www.konsulko.com/getting-started-with-rauc-on-raspberry-pi-2?_gl=1*j0sguk*_up*MQ..*_ga*MTQ5MDY0MTI3OC4xNzI2MTIyMTg3*_ga_91XEKLF7VE*MTcyNjEyMjl3My4xLjAuMTcyNjEyMjl3My4wLjAuMA..*_ga_G7D0YGWST4*MTcyNjEyMjE4NS4xLjEuMTcyNjEyMjl3My4wLjAuMA..#comment-615

Step 1: Build and Start hawkBit Update Server

1. Clone the hawkBit repository:

```
""bash
git clone https://github.com/eclipse-hawkbite/hawkbite.git
cd hawkbite
""
```

2. Build the hawkBit Update Server using Maven:

```
""bash
mvn clean install
""
```

3. Start the hawkBit server:

```
""bash
java -jar ./hawkbite-monolith/hawkbite-update-server/target/hawkbite-update-server-0-SNAPSHOT.jar
""
```

4. Access the hawkBit server UI: Open a browser and navigate to 'http://<server-ip>:8080'. Log in using the default credentials ('admin' / 'admin').

Step 2: Create a Target in hawkBit

To define a target (e.g., a Raspberry Pi device) in the hawkBit server:

1. Use the hawkBit Management API to create a target:

- Endpoint: 'POST /rest/v1/targets'
- Example JSON request body:

```
""json
[
  {
    "name": "RaspberryPiDevice",
    "description": "Raspberry Pi running HawkBit client",
    "controllerId": "RPI-001",
    "address": "http://<server-ip>",
    "securityToken": "<your-security-token>",
    "requestAttributes": true
  }
]
""
```

2. Verify the target:

- Endpoint: 'GET /rest/v1/targets/<target-id>'
- Example:

```
""bash
curl -X GET http://<server-ip>:8080/rest/v1/targets/RPI-001
""
```

Ensure the response includes the correct details about the target.

Step 3: Set Up the RAUC hawkBit Client on Raspberry Pi

1. Install the RAUC hawkBit updater on the Raspberry Pi:

```
""bash
git clone https://github.com/rauc/rauc-hawkbit-updater.git
cd rauc-hawkbit-updater
sudo apt update
sudo apt install -y build-essential meson ninja-build libglib2.0-dev libcurl4-openssl-dev
meson setup build
ninja -C build
sudo ninja -C build install
""
```

2. Configure the RAUC hawkBit client:

- Create the configuration file '/etc/rauc-hawkbit-updater/config.conf':

```
""ini
[client]
hawkbit_server = <server-ip>:8080
ssl = false
ssl_verify = false
tenant_id = DEFAULT
target_name = RPI-001
auth_token = <your-security-token>
```

```
bundle_download_location = /tmp/bundle.raucb
```

```
[device]
mac_address = ff:ff:ff:ff:ff:ff
hw_revision = 2
model = T1
""
```

3. Test the connection:

```
""bash
rauc-hawkbiter -c /etc/rauc-hawkbiter/config.conf
""
```

Ensure that the client successfully connects to the hawkBit server.

or

Set Up the RAUC hawkBit Client on Raspberry Pi(alternate method)

Add the rauc-hawkbiter client to the Yocto image:

->) In your Yocto local.conf file, append the following:

```
-IMAGE_INSTALL:append = " rauc-hawkbiter"
```

->) Build the Yocto image with the updated configuration:

```
-bitbake <image-name>
```

->) Flash the generated image to the Raspberry Pi.

Configure the RAUC hawkBit client on the Raspberry Pi:

Create the configuration file /etc/rauc-hawkbiter/config.conf:

```
[client]
hawkbiter_server = <server-ip>:8080
ssl = false
ssl_verify = false
tenant_id = DEFAULT
target_name = RPI-001
auth_token = <your-security-token>
bundle_download_location = /tmp/bundle.raucb
```

```
[device]
mac_address = ff:ff:ff:ff:ff:ff
hw_revision = 2
model = T1
```

->) Test the connection:

```
-rauc-hawkbiter -c /etc/rauc-hawkbiter/config.conf
```

->) Ensure that the client successfully connects to the hawkBit server.

Step 4: Create a Software Module in hawkBit

1. Navigate to the 'Software Modules' section in the hawkBit Management API.
2. Use the 'POST /rest/v1/softwaremodules' endpoint to create a new software module:

- Example JSON request body:

```
""json
[
  {
    "name": "UpdateBundle",
    "version": "1.0",
    "type": "os",
    "description": "RAUC Bundle for Raspberry Pi",
    "vendor": "Your Company",
    "encrypted": false
  }
]
""
```

3. Upload the RAUC bundle to the created software module:

- Endpoint: 'POST /rest/v1/softwaremodules/<module-id>/artifacts'
- Upload the bundle file (e.g., 'update-bundle-raspberrypi4.raucb').

Step 5: Create a Distribution Set in hawkBit

1. Use the 'POST /rest/v1/distributionsets' endpoint to create a distribution set:

- Example JSON request body:

```
""json
[
  {
    "name": "UpdateBundleDS",
    "description": "Distribution set for updating the RAUC bundle on Raspberry Pi",
    "version": "1.0",
    "locked": true,
    "modules": [
      { "id": <software-module-id> }
    ],
    "type": "os"
  }
]
""
```

2. Assign the software module to the distribution set:

- Endpoint: 'POST /rest/v1/distributionsets/<distribution-set-id>/assignedSM'
- Example JSON request body:

```
""json
[
  { "id": <software-module-id> }
]
""
```

Step 6: Assign the Distribution Set to the Target

1. Use the 'POST /rest/v1/targets/<target-id>/assignedDS' endpoint to assign the distribution set to the target:

- Example JSON request body:

```
""json
[
  {
    "id": <distribution-set-id>,
    "type": "soft",
    "schedule": "10 12 14 3 8 ? 2023",
    "duration": "00:10:00",
    "timezone": "+00:00"
  }
]
""
```

2. Verify the assignment:

- Endpoint: 'GET /rest/v1/targets/<target-id>/assignedDS'
- Ensure the response includes the correct distribution set.

Step 7: Verify and Download the Update on the Raspberry Pi

1. Run the RAUC hawkBit updater:

```
""bash
rauc-hawkbit-updater -c /etc/rauc-hawkbit-updater/config.conf
""
```

2. Monitor the logs to ensure the following steps occur:

- Connection to the hawkBit server.
- Detection of a new update.
- Download of the RAUC bundle.
- Verification of the bundle checksum.
- Installation of the update.

Step 8: Post-Update Verification

1. Check the RAUC status on the Raspberry Pi:

```
““bash
rauc status
““
```

Ensure that the boot slot and update status reflect the newly installed update.

2. Verify the target status in hawkBit:

- Endpoint: 'GET /rest/v1/targets/<target-id>/actions'
- Ensure the update action is marked as completed.

Conclusion

This documentation provides a complete walkthrough of setting up hawkBit with RAUC, adding targets, uploading bundles, creating distribution sets, and managing updates. By following these steps, you can ensure seamless updates to your Raspberry Pi or other IoT devices.
