Cloudera Edge Management 1.4.1

Using Asset Push Command [Tech Preview]

Date published: 2019-04-15
Date modified: 2022-07-21

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

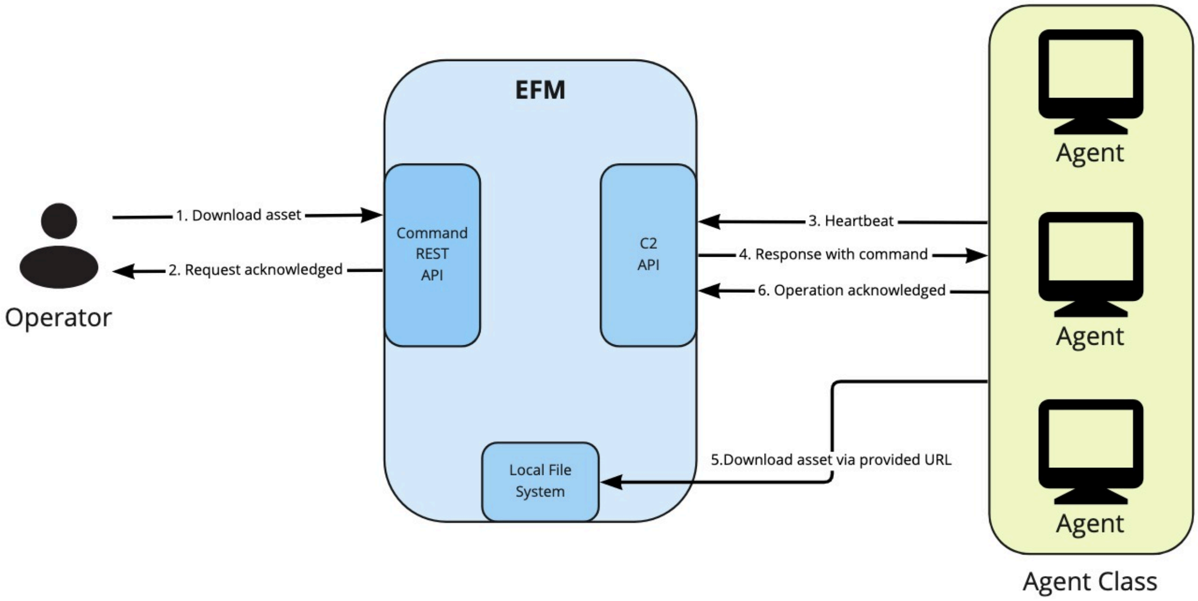
Contents

Using Asset Push command

Learn how you can trigger an asset push command to agents of a particular agent class from the Swagger UI.

You need to initiate the process by issuing an UPDATE ASSET command against an agent class through the REST API. The call is asynchronous and returns the created operation details. On the next heartbeat, the agent receives the operation (command) in the heartbeat response. The agent initiates the download through the provided URL which points to EFM. EFM proxies the URL to the respective external storage. External storage is a local filesystem in the tech preview. The external storage can be cloud object storage, database, and so on in future releases. Asset files have to be deployed on all EFM nodes under the same path and name. After the asset is downloaded, the agent acknowledges the operation.

The workflow of this process is as depicted in the following diagram:



Agent configuration

On the agent side the default location where the asset will be saved is ${MINIFI_HOME}/asset. You can customize this path with the nifi.asset.directory property.

Executing an asset push command

1. Ensure that the asset is available on all EFM nodes, which the agents download.
2. Construct the request payload.

   For example,

```
{
    "assetFileName": "test.txt",
    "assetUri": "/tmp/input.txt",
    "forceDownload": false
}
```

   Where,

   - assetFileName = The target file name to be used on the agent.

  - assetUri = Location of the asset on EFM.
  - forceDownload = Forces the agent to skip all version checks and perform asset download.

3. Send a POST request to /efm/api/commands/<testAgentClass>/update-asset with the payload created in the previous step and the agent class name.

   This triggers all agents under the agent class to download the requested asset.
4. Go to the Edge Events page and check for the UPDATE ASSET command.

   You see the UPDATE ASSET command where you can track how it goes through its lifecycle stages (for example, queued, deployed, and done). You can also check the command's current state under the commands tab in the agent details page for the agent manager.

   Result: After commands are executed, assets are available on all agents belonging to the specified agent class.

Restrictions and limitations

As the current implementation is in tech preview, be aware of the following limitations:

- If asset push fails, there is no automated recovery or retry option.
- Assets must be located on all EFM nodes; no external source is available at this point. It also means your EFM node(s) hardware capabilities can be a limitation for download speed or storage capacity for assets.
- Only CEM MiNiFi C++ Agent - 1.22.06 release supports this feature.