Cloudera Flow Management 2.1.5

Using parameter providers

Date published: 2019-06-26
Date modified: 2022-12-13

https://docs.cloudera.com/

Legal Notice

Contents

What is a parameter provider?

Parameter provider is a new extension point that is added to the existing NiFi parameter context feature that can help to populate flow parameters on demand. Learn the features associated with parameter providers and how you can use them in your NiFi environment.

You can create parameter contexts from parameters fetched from an external source. Using parameter providers allows automatic creation of parameter contexts from external sources, for example file-based Kubernetes secrets, environment variables, or HashiCorp Vault secrets engines.

Parameter providers:

- Provide a feature in the controller settings that enable you to generate parameter contexts from external sources.
- Enable you to keep your provided parameter contexts up to date with the external source by running a Fetch Parameters operation.
- Provide an extension point for developing new custom parameter providers that can be deployed in a NiFi Archive (NAR).
- Provide a CLI command, nifi fetch-params, to fetch and apply parameters, allowing a scripted approach to keep parameter contexts up to date.

Parameter values are stored (encrypted, if sensitive) inside the flow. Parameter providers are a mechanism that automate the creation of parameter contexts and facilitate keeping them updated. They do not replace the framework mechanism to resolve parameter values during processor/controller service execution and they do not pull parameters directly from the external source at the time of usage in the flow.

Note: Parameter providers do not have an automatic mechanism to refresh parameter values from the NiFi UI. Fetching and applying parameters is a potentially disruptive operation, since it can involve stopping and starting large portions of the flow. This kind of activity could be scripted using the CLI command fetch-pa rams, but should be done with the potential for flow disruption in mind.

For more information, see *Parameter Contexts* in the Apache NiFi User Guide.

Related Information

Parameter Contexts, Apache NiFi User Guide

Example for using parameter providers

In this example you can see how a parameter provider is created and configured, how it is used to fetch parameters from an external source, and how a parameter context is created and updated using the parameters fetched from the external source.

Creating and configuring a parameter provider

Learn how to create and configure parameter providers through the NiFi controller settings.

Procedure

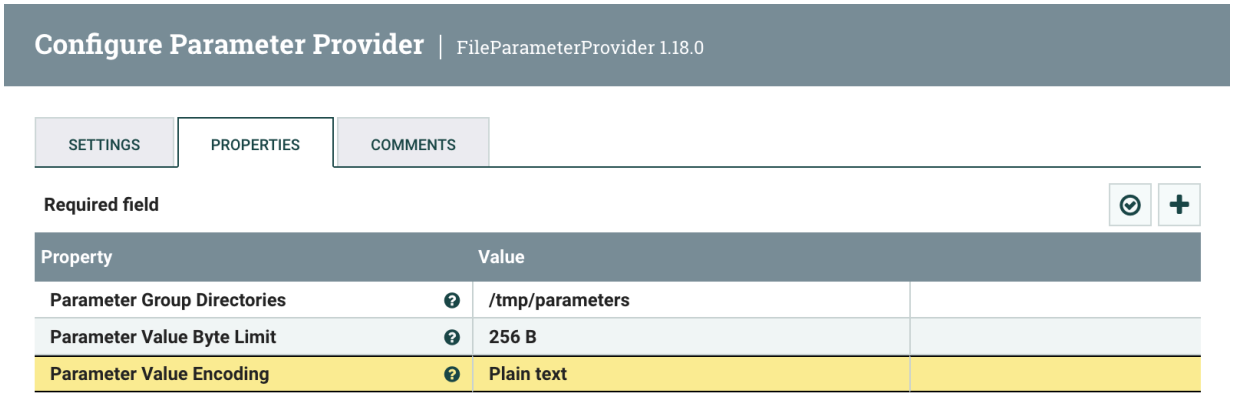
1. Select Controller Settings from the top-right Global menu in the NiFi UI.
   The **NiFi Settings** dialog opens.
2. Go to the PARAMETER PROVIDERS tab.
3. Click **+** in the top-right corner of the NiFi Settings dialog to add a new parameter provider.

4. Select one of the listed types, in this example, FileParameterProvider, and click Add.

   The parameter provider is created.

   The FileParameterProvider created in this example lets you supply parameters in key-value files inside a parameter group directory.

5. Click to edit it.
6. Enter the absolute path of the parameter groups on your file system in the Parameter Group Directories property.
7. Set the Parameter Value Encoding property to 'Plain text'.



8. Click APPLY to save the parameter provider configuration.

   The Fetch Parameters icon becomes available if the path set for the Parameter Group Directories property is readable.

Results

The parameter provider, in this example FileParameterProvider, is created and configured.

What to do next

You can now move on to fetching parameters.

Related Information

Fetching parameters

Fetching parameters

Learn how to fetch parameters from an external source into a parameter group.

Before you begin

You must have created and configured a parameter provider. See Creating and configuring a parameter provider.

Procedure

1. If you do not have any files containing key-value pairs as parameters inside the directory set in the Parameter Group Directories property of FileParameterProvider, which in this example is the tmp/parameters directory, create some files and add some content.

```
$ print admin > /tmp/parameters/sys.admin.username && \
  print password > /tmp/parameters/sys.admin.password && \
```

```
print value > /tmp/parameters/sys.other
```

In this example, each file represents a parameter in the parameters parameter group, and the contents of each file represents the parameter value.

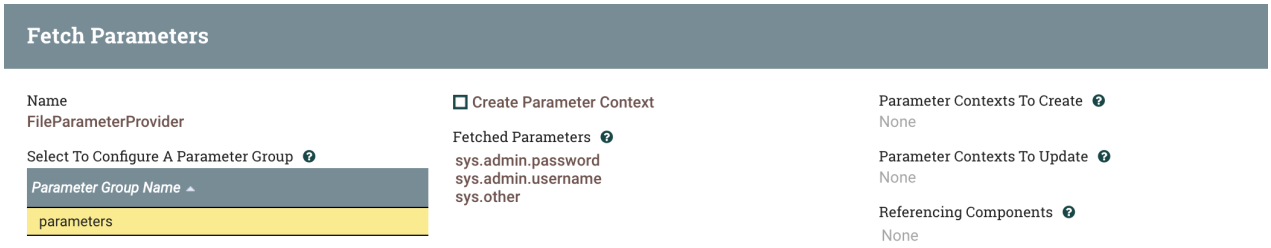
2. Select Controller Settings from the top-right Global menu in the NiFi UI.

   The NiFi Settings dialog opens.
3. Go to the PARAMETER PROVIDERS tab.
4. Click the Fetch Parameters (down arrow) icon of FileParameterProvider.

   The Fetch Parameters dialog opens.

Results

A parameter group named parameters is listed. This is the name of the directory where the parameter groups are stored on your file system. This group can be used to create a parameter context. The fetched parameters sys.admin.username, sys.admin.password, and sys.other are displayed.



What to do next

You can now move on to creating a parameter context from a parameter group.

Related Information

Creating a parameter context from a parameter group

Creating a parameter context from a parameter group

Learn how to create a parameter context from a parameter group that stores parameters fetched from an external source using a parameter provider.

Before you begin

You must have fetched parameters from an external source. See Fetching parameters.

Procedure

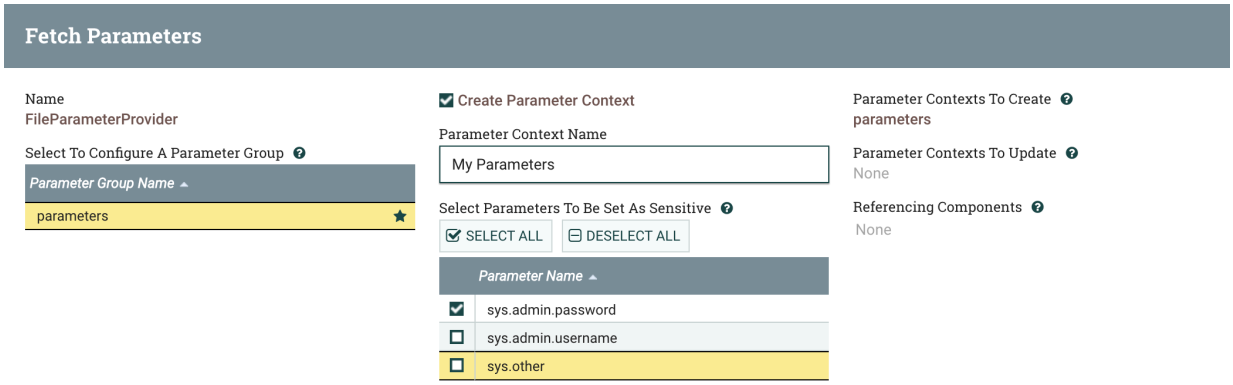
1. Select Controller Settings from the top-right Global menu in the NiFi UI.

   The NiFi Settings dialog opens.
2. Go to the PARAMETER PROVIDERS tab.
3. Click the Fetch Parameters (down arrow) icon of FileParameterProvider.

   The Fetch Parameters dialog opens.

4. Select Create Parameter Context.
   - A star appears next to the parameters parameter group. The star indicates that the parameter group has an associated parameter context.
   - Checkboxes appear next to the parameter names.
   - The Parameter Context Name field appears.

   Enter a name for your parameter context. In this example, it is My Parameters.



5. Select the parameters that you want to set as sensitive.
   In this example, only the sys.admin.password parameter needs to be set to sensitive.
6. Click APPLY to create the parameter context.
7. Click Close in the Fetch Parameters dialog.

Results

- The newly created parameter context appears under the SETTINGS tab of the Configure Parameter Provider dialog.
- The parameters appear under the PARAMETERS tab of the Update Parameter Context dialog. Only the values of the non-sensitive parameters can be read. The parameters cannot be edited. The parameter provider can add, remove, or update parameters in this parameter context.

What to do next

- You can view the new parameter context in the NiFi Parameter Contexts list by clicking its name.
- You can view the details of the parameter context in the Update Parameter Context dialog by clicking .
- You can return to the NiFi Settings by clicking the right arrow icon in the listing.

Related Information

Creating a parameter context from a parameter group

What is Parameter Context inheritence?

Updating parameter sensitivity

You may need to update the sensitivity setting of parameters of a Pprameter contexts over time.

Before you begin

- You have created a parameter context from a parameter provider.
- If the parameter context, in this example, 'My Parameters', is assigned to a group, and that group has a component that references the sys.other parameter, you cannot update its sensitivity. In order to do so, ensure that you remove any references to it.

You must have created a parameter context from a parameter provider.

Procedure

1. Select Controller Settings from the top-right Global menu in the NiFi UI.
   The NiFi Settings dialog opens.
2. Go to the PARAMETER PROVIDERS tab.
3. Click the Fetch Parameters (down arrow) icon of FileParameterProvider.
   The Fetch Parameters dialog opens.
4. Select the parameter whose sensitivity you want to change.
5. Click APPLY to update the parameter context.

Updating parameter context when the external source has changed

The parameters that you fetched from an external source may change in that external source over time. In such cases you need to fetch the parameters again using the parameter providers.

About this task

Parameters are not automatically synchronized with the external source. To update any linked parameter contexts, you can fetch the parameters again, and you need to specify the sensitivity of any new parameters.

Before you begin

There must be changes to your external source. To simulate the change in the example, delete the sys.other parameter, update the value of the sys.admin.username parameter, and add a new parameter new-parameter.

```
$ rm /tmp/parameters/sys.other && \
  print admin2 > /tmp/parameters/sys.admin.username && \
  print test > /tmp/parameters/new-parameter
```

Procedure

1. Select Controller Settings from the top-right Global menu in the NiFi UI.
   The NiFi Settings dialog opens.
2. Go to the PARAMETER PROVIDERS tab.
3. Click the Fetch Parameters (down arrow) icon of FileParameterProvider.
   The Fetch Parameters dialog opens. An asterisk appears next to newly fetched and changed parameters. To view detailed information about the change in the parameter, hover over the asterisk.
4. Set sensitivity for the newly fetched parameters.
5. Click APPLY to update the parameter context.
6. Click Close in the Fetch Parameters dialog.

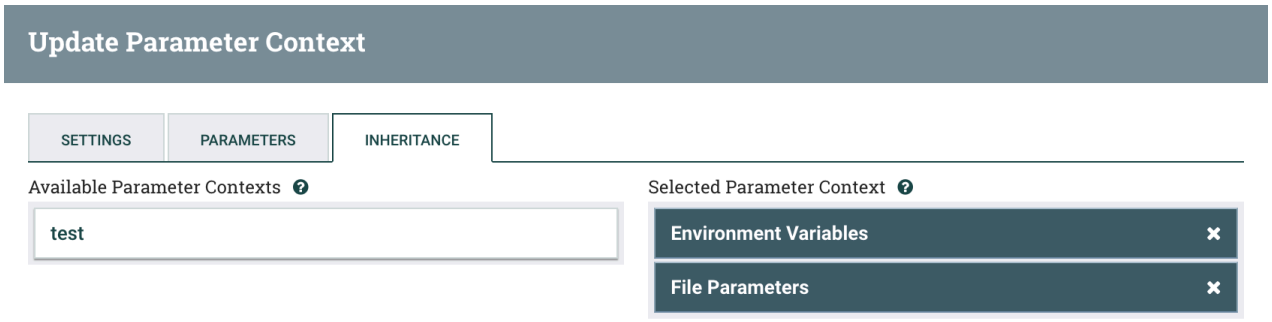
Results

The changes that took place in the parameters in the external source are now reflected in the parameter context. In this example, sys-other parameter is removed, new-parameter is added, and the value of sys.admin.username is changed. You can view the changes by clicking the Fetch Parameters (down arrow) icon.

Using parameter context inheritance to combine parameters

You can use parameter context inheritance to combine parameters from different parameter providers within one parameter context.

It is good practice to create separate parameter providers for different sources, and then compose the parameter contexts through inheritance. This allows sensitive parameters to originate from a secrets manager, like HashiCorp Vault, and non-sensitive parameters to originate from other sources like environment variables or the file system. In the following simple example, the EnvironmentVariableParameterProvider contains non-sensitive parameters and the FileParameterProvider contains sensitive parameters. Both parameter providers are added as inherited contexts to a new parameter context, 'Parameters'. For details on inherited parameter contexts, see *What is parameter context inheritance?*.



The new parameter context inherits parameters from both sources:

JENV_FORCEJAVAHOME	true	→
JENV_FORCEJDKHOME	true	→
JENV_LOADED	1	→
JENV_SHELL	zsh	→
new-parameter	Sensitive value set	→
sys.admin.password	Sensitive value set	→
sys.admin.username	Sensitive value set	→