Extensions

Background

Added in Ambari 2.4.

An Extension is a collection of one or more custom services which are packaged together. Much like stacks, each extension has a name which needs to be unique in the cluster. It also has a version directory to distinguish different releases of the extension. Much like stack versions which go in `/var/lib/ambari-server/resources/stacks` with <stack_name>/<stack_version> sub-directories, extension versions go in `/var/lib/ambari-server/resources/extensions` with <extension_name>/<extension_version> sub-directories.

An extension can be linked to supported stack versions. Once an extension version has been linked to the currently installed stack version, the custom services contained in the extension version may be added to the cluster in the same manner as if they were actually contained in the stack version.

Third party developers can release Extensions which can be added to a cluster.

Structure

The structure of an Extension definition is as follows:

```
|_ extensions
   |_ <extension_name>
      |_ <extension_version>
         |_ metainfo.xml
         |_ services
            |_ <service_name>
               |_ metainfo.xml
               |_ metrics.json
               |_ configuration
                  |_ {configuration files}
               |_ package
                  |_ {files, scripts, templates}
```

An extension version is similar to a stack version but it only includes the metainfo.xml and the services directory. This means that the alerts, kerberos, metrics, role command order, widgets files are not supported and should be included at the service level. In addition, the repositories, hooks, configurations, and upgrades directories are not supported although upgrade support can be added at the service level.

Extension Inheritance

Extension versions can *extend* other Extension versions in order to share command scripts and configurations. This reduces duplication of code across Extensions with the following:

- add new Services in the child Extension version (not in the parent Extension version)
- override command scripts of the parent Services
- override configurations of the parent Services

For example, **MyExtension 2.0** could extend **MyExtension 1.0** so only the changes applicable to **the MyExtension 2.0** extension are present in that Extension definition. This extension is defined in the metainfo.xml for **MyExtension 2.0**:

```
<metainfo>
  <extends>1.0</extends>
```

Supported Stack Versions

Each Extension Version must support one or more Stack Versions. The Extension Version specifies the minimum Stack Version which it supports. This is included in the extension's metainfo.xml in the prerequisites section like so:

```
<metainfo>
  <prerequisites>
    <min-stack-versions>
      <stack>
        <name>HDP</name>
        <version>2.4</version>
      </stack>
      <stack>
        <name>OTHER</name>
        <version>1.0</version>
      </stack>
    </min-stack-versions>
  </prerequisites>
</metainfo>
```

Installing Extensions

It is recommended to install extensions using management packs. For more details see the instructions on packaging custom services using extensions and management packs.

Once the extension version directory has been created under the resource/extensions directory with the required metainfo.xml file, you can restart ambari-server.

Extension REST APIs

You can query for extensions by calling REST APIs.

Get all extensions

```
curl -u admin:admin -H 'X-Requested-By:ambari' -X GET 'http://<server>:<port>/api/v1/extensions'
{
  "href" : "http://<server>:<port>/api/v1/extensions/",
  "items" : [
    {
      "href" : "http://<server>:<port>/api/v1/extensions/EXT",
      "Extensions" : {
        "extension_name" : "EXT"
      }
    }
  ]
}
```

Get extension

```
curl -u admin:admin -H 'X-Requested-By:ambari' -X GET 'http://<server>:<port>/api/v1/extensions/EXT'

{
  "href" : "http://<server>:<port>/api/v1/extensions/EXT",
  "Extensions" : {
    "extension_name" : "EXT"
  },
  "versions" : [
    {
      "href" : "http://<server>:<port>/api/v1/extensions/EXT/versions/1.0",
      "Versions" : {
        "extension_name" : "EXT",
        "extension_version" : "1.0"
      }
    }
  ]
}
```

Get extension version

```
curl -u admin:admin -H 'X-Requested-By:ambari' -X GET 'http://<server>:<port>/api/v1/extensions/EXT/versions/1.0'

{
  "href" : "http://<server>:<port>/api/v1/extensions/EXT/versions/1.0/",
  "Versions" : {
    "extension-errors" : [],
    "extension_name" : "EXT",
    "extension_version" : "1.0",
    "parent_extension_version" : null,
    "valid" : true
  }
}
```

Extension Links

An Extension Link is a link between a stack version and an extension version. Once an extension version has been linked to the currently installed stack version, the custom services contained in the extension version may be added to the cluster in the same manner as if they were actually contained in the stack version.

It is only possible to link an extension version to a stack version if the stack version is supported by the extension version. The stack name must be specified in the prerequisites section of the extension's metainfo.xml and the stack version must be greater than or equal to the minimum version number specified.

Extension Link REST APIs

You can retrieve, create, update and delete extension links by calling REST APIs.

Create Extension Link

The following curl command will link an extension EXT/1.0 to the stack HDP/2.4

```
curl -u admin:admin -H 'X-Requested-By: ambari' -X POST -d '{"ExtensionLink": {"stack_name": "HDP", "stack_version":

"2.4", "extension_name": "EXT", "extension_version": "1.0"}}' http://<server>:<port>/api/v1/links/
```

Get All Extension Links

```
curl -u admin:admin -H 'X-Requested-By:ambari' -X GET 'http://<server>:<port>/api/v1/links'

{
  "href" : "http://<server>:<port>/api/v1/links/",
  "items" : [
    {
      "href" : "http://<server>:<port>:8080/api/v1/links/1",
      "ExtensionLink" : {
        "extension_name" : "EXT",
        "extension_version" : "1.0",
        "link_id" : 1,
        "stack_name" : "HDP",
        "stack_version" : "2.4"
      }
    }
  ]
}
```

Get Extension Link

```
curl -u admin:admin -H 'X-Requested-By:ambari' -X GET 'http://<server>:<port>/api/v1/link/1'
{
  "href" : "http://<server>:<port>/api/v1/links/1",
  "ExtensionLink" : {
    "extension_name" : "EXT",
    "extension_version" : "1.0",
    "link_id" : 1,
    "stack_name" : "HDP",
    "stack_version" : "2.4"
  }
}
```

Delete Extension Link

You must specify the ID of the Extension Link to be deleted.

```
curl -u admin:admin -H 'X-Requested-By: ambari' -X DELETE http://<server>:<port>/api/v1/links/<link_id>
```

Update All Extension Links

This will reread the stacks, extensions and services in order to make sure the state of the stack is up to date in memory.

```
curl -u admin:admin -H 'X-Requested-By: ambari' -X PUT http://<server>:<port>/api/v1/links/
```