Aspire Parcel and Service How to Configure

Step 1: Set up Cloudera Manager

Step 1a: Add the Aspire parcel repository to the Cloudera Manager

1. If you haven't create a temporary parcel repository with the Aspire Cloudera Parcel binaries.
2. In the Cloudera Manager click on the parcels icon to go to the parcels page.
3. In parcels page click on the configuration button.
4. Scroll until you see the option to add a remote parcel repository.
5. Click con the + button and then copy the url of the Aspire parcel repository.
6. Finally click on the Save Changes button.

> **ⓘ Local Repository**
>
> If for some reasons the remote repository cannot be used, download the parcel from it and follow the steps inside this link to configure a local repository

### Step 1b: Add the Aspire service to the Cloudera Manager

1. Download the jar of the Aspire-4.0 service.
2. Put the jar in the CSD path configured in Cloudera Server, the default is /opt/cloudera/csd
3. Restart the Cloudera Manager so the new service is picked up

```
service cloudera-scm-server stop
service cloudera-scm-server start
```

Step 2: Download, Distribute and Activate the Parcel

1. Go to the *Cloudera Manager Parcels* management page and look for the Aspire parcel that should be available to download.
2. Click **Download** and wait until download process is done.
3. Click **Distribute**.
4. After the distribution is done, click **Activate**.
   - The parcel should be activated on all instances of the Cloudera Cluster.
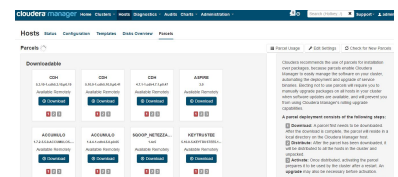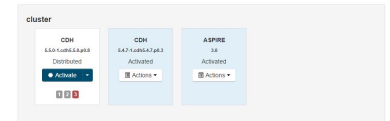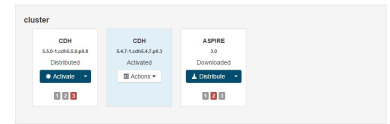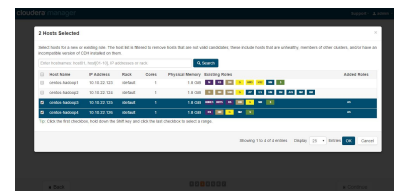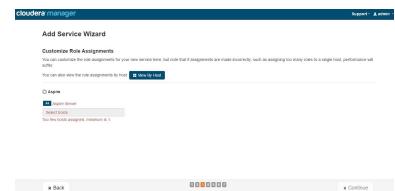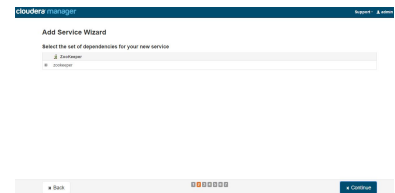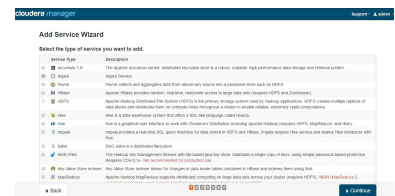
Step 3: Add the New Service

On the cluster, click **Add Service** to go to the add service wizard.

## Step 3a: Add Aspire as a service

In the wizard, Aspire should be listed as a service to install.

1. Select **Aspire** from the available services.
2. If not selected automatically, select the Zookeeper service to use.
3. Select the hosts where Aspire Server will be installed.

Step 3b: Configure Aspire

1. Common settings
   - **Aspire Version**: The Aspire version
   - **Maven username**: Enter the username of the maven account
   - **Maven password**: Enter the password of the maven account

2. No SQL settings. The provider is the repository used by the different Aspire components to store any required data used by their processes.
   a. **DB provider implementation**: Select the implementation of the no SQL db provider to use. The are two supported implementations Mongo and HBase.

3. HBase settings, these will be ignored if the provider is not HBase. For more details about how these properties work check the HBase provider documentation
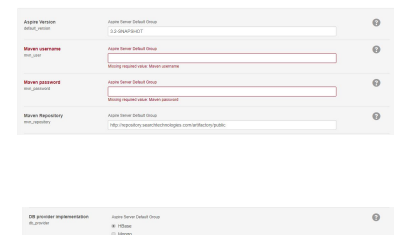
- **HBase Properties**: Specify the required Hbase properties to use in the format: NAME:VALUE. Use these only if the Hadoop configuration files are not available to use or if any additional properties besides the included in the files are required
- **HBase config path**: The path of the directory that contains the HBase config files. The required files are the hbase-site.xml and the hdfs-site.xml/core-site.xml files. If not used add any required properties in the **HBase Properties** field to add any required configuration property (ex: `hbase.zookeeper.quorum`)
- **HBase Kerberos User:** The HBase principal to use with Kerberos. Configure only if Kerberoes authentication is used.
- **HBase Kerberos Keytab:** The HBase keytab to use with Kerberos. Configure only if Kerberoes authentication is used.

4. Mongo settings, these will be ignored if the provider is not Mongo
   - **Mongo DB server**: The Mongo DB server to use.

5. Authentication settings. Authentication used in order to access the Aspire Admin Console UI
   - **Authentication type**: The type of authentication used by the Aspire Admin Console UI. It can be None (no authentication) or Ldap.

6. Ldap settings. These will be ignored if the authentication type is set to None. For more details about how these properties work check the Aspire security documentation
   - **LDAP server**: The ldap server to validate against.
   - **LDAP Authentication**: The ldap authentication to use (anonymous/simple/DIGEST-MD5).
   - **LDAP Search Base**: The search base for ldap queries for the user's dn.
   - **LDAP Administrator User DN**: Holds the dn of one user that may access Aspire with Administration role
   - **LDAP Administrator Group DN**: Holds the dn of one group that may access Aspire with Administration role. NOTE: currently, the users must be direct members of this group (not a member of a group that is a member of this group).
   - **LDAP Developer User DN**: Holds the dn of one user that may access Aspire with Developer role.
   - **LDAP Developer Group DN**: Holds the dn of one group that may access Aspire with Developer role. NOTE: currently, the users must be direct members of this group (not a member of a group that is a member of this group).
   - **LDAP Groups Hold Members**: Indicates that ldap group objects hold the membership information. By default, user objects are expected to hold group membership.
   - **LDAP User DN Query**: The ldap query used to find the dn for the user that logged in.
   - **LDAP Member Attribute**: The attribute of the ldap object (be it user or group) that holds information about group membership.

7. Other Aspire settings
   - **System Properties**: Specify the required Aspire system properties to use in the format: NAME:VALUE. This properties will be available to use on the installed Aspire Components by referencing them as ${PROPERTY_NAME}.
   - **Bundle Versions**: Specify the required bundle versions to use in the following format: ARTIFACT_ID:VERSION. Use this for components that require a different version from the default one.

8. Felix settings. Settings related with the felix properties file used by Aspire. Use only if there properties required that are not included by the default felix.properties file.
   - **Additional Felix Properties**: Add any additional Felix properties to use, format: PROPERTY=VALUE.
   - **Extra System Packages**: Appends packages to the default set of exported system packages.
   - **Boot Delegation Packages**: The following property makes specified packages from the class path available to all bundles. You should avoid using this property.

9. JVM settings
   - **Initial Memory**: Aspire initial memory.
   - **Maximum Memory**: Aspire maximum memory
   - **Maximum Metaspace**: Aspire maximum Metaspace
   - **JVM Additional Options**: Add any additional JVM options to use, format: OPTION=VALUE.

Step 3c: Finish the installation

Continue to the next tab on the wizard and wait for Aspire to start running, after that installation process is done.
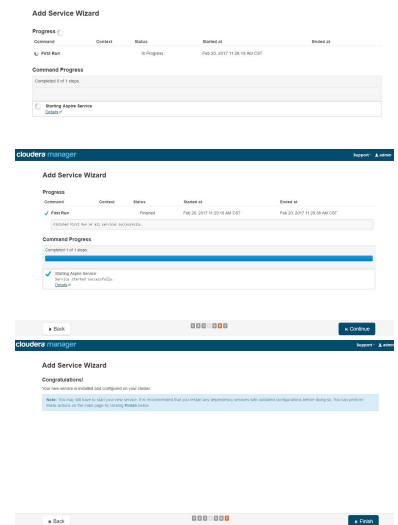
Step 4: Check the newly installed service

On the cluster, you can check that Aspire is listed along the other services.

If you click on the service you will see multiple tabs, with the following being relevant:

- **Status**: Here you can check the health of the service and it's instances.
- **Instances:** You can check and manage the instances where the service is instances.
- **Configuration:** Tab to make changes to the configuration of the service, any change made will be deployed to all Aspire instances when the service is restarted.
- **Commands:** It shows the history of command executed on the service.

> (i) If you click on an specific instance in the instances tab you will see the same tabs mentioned before, but only with information related to the selected instance, any configuration changes made here will only apply to that instance instead of all instances of the service.