Schedule-based Autoscaling for Cloudera Data Hub Clusters Using Impala (Preview)

Date published: 2023-11-09
Date modified: 2024-12-04

Legal Notice

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

Contents

Schedule-based autoscaling for Data Hub clusters using Impala

Autoscaling is a feature that adjusts the capacity of cluster nodes running YARN and/or Impala by automatically increasing or decreasing, or suspending and resuming, the nodes in a host group.

Generally speaking, you can enable autoscaling based either on a schedule that you define (schedule-based autoscaling), or the real-time demands of your workloads (load-based autoscaling). However, for cluster nodes that run only Impala, such as the Data Mart clusters, you are limited to only **schedule-based autoscaling policies**.

Important! This document focuses on schedule-based autoscaling for clusters running Impala, as both load and schedule-based autoscaling for YARN is covered in Autoscaling clusters.

Schedule-based autoscaling for Impala scales the number of nodes in an executor host group up or down based upon a schedule that you define. Schedule-based autoscaling is useful if workload demands tend to be high or low on a fairly regular, consistent basis.

When you configure a schedule-based autoscaling policy, you define a target node count, which is the number of nodes that you want to scale up or down to at a particular time. You finalize the schedule by entering a CRON expression and selecting the desired timezone. When the particular time/date that you define in the CRON expression occurs, the cluster is upscaled or downscaled to your target node count. The time taken to add nodes to a cluster varies by cloud provider and the configuration of the nodes (for example, recipes can have an impact on the time it takes to add a node).

Typically, schedules (especially those configured to scale-up) should be defined in a manner to factor in the time it takes to add new nodes. For example, workloads starting at 9pm will typically set a scale-up schedule for 8:45 pm, assuming it takes around 15 minutes to add a node.

Note: Schedule-based autoscaling for Impala is available for clusters provisioned in AWS and Azure with Cloudera Runtime versions 7.2.15 or newer. This is a technical preview feature under entitlement. This feature is available by default in the Data Mart cluster definitions. Schedule-based autoscaling can also be configured for clusters with custom templates that include Impala.

Autoscaling behavior

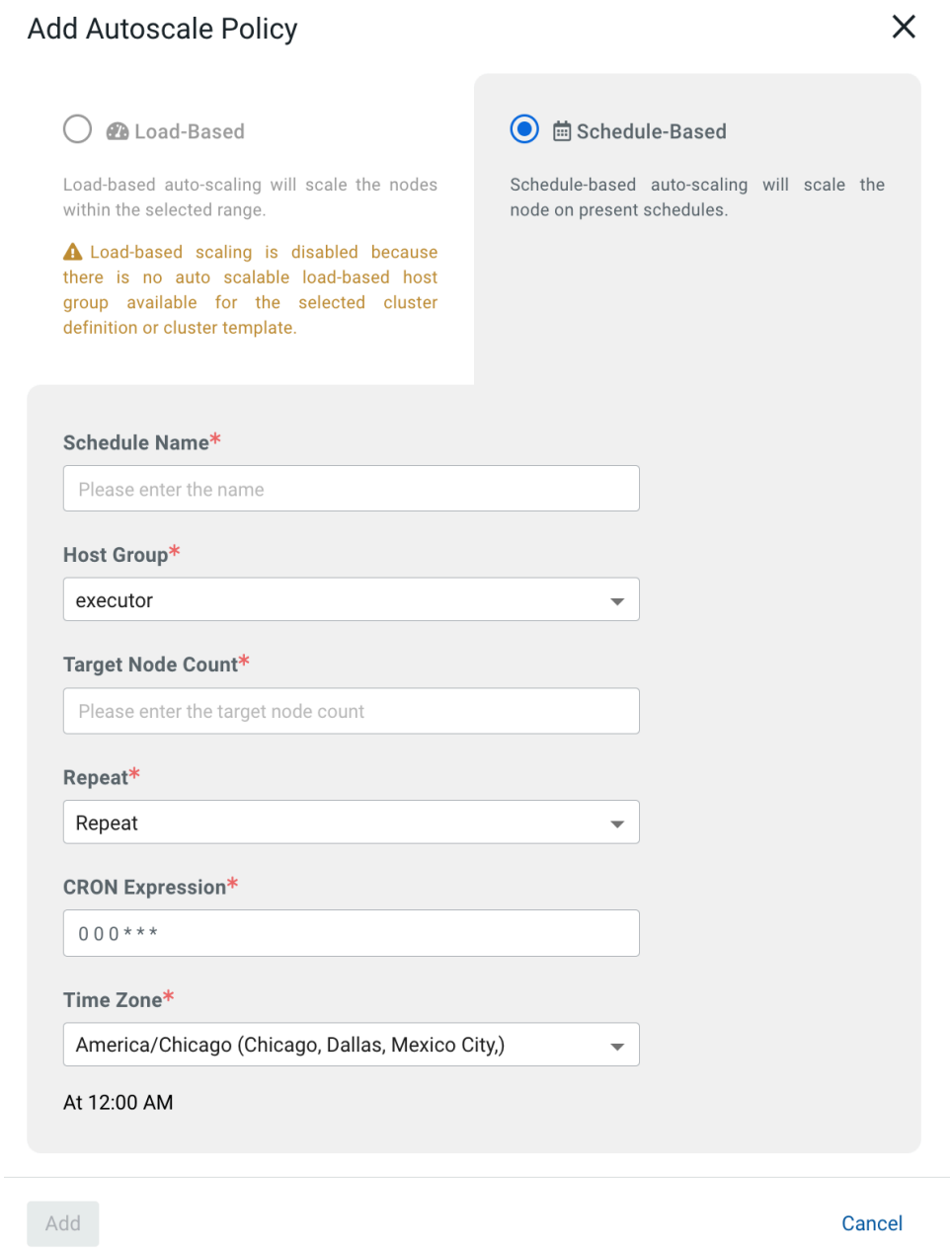
Before you define an autoscaling policy, note the following autoscaling behaviors:

- Schedule-based autoscaling for Impala is available for executor host groups only on clusters running Impala.
- During scale-down, nodes are decommissioned randomly. Note that if YARN is also running in addition to Impala, YARN takes priority and nodes are decommissioned based on the recommendations of YARN.
- Clusters can perform one upscale or downscale operation at a time.
- A cluster will continue to accept jobs while it is running, regardless of any in-progress upscale or downscale operations.
- You can create just one or multiple schedule-based policies for a host group.
- If there are not enough nodes available to match the requested scale operation, the operation will proceed on however many nodes are available (for example, during a request for a 10 node scale-up, if the cluster loses 1 node, the operation will proceed with scaling-up 9 nodes instead of 10).
- Autoscaling will be disabled if the Cloudera Manager cluster node fails.

Configuring schedule-based autoscaling for Cloudera Data Hub clusters using Impala

To configure autoscaling for clusters using Impala, add a schedule-based policy to a cluster and define the policy parameters.

1. From the Cloudera Management Console, click **Data Hub Clusters** and then select the cluster that you want to add an autoscaling policy to. You can add autoscaling policies to clusters after they have been created, not during the cluster creation process.
2. From the cluster details page, select the **Autoscale** tab and then click the slider button to enable autoscaling. Autoscaling is disabled by default.
3. Click **Add Autoscale Policy** and select **Schedule-Based**.
4. Define the policy parameters:



Parameter	Description
Schedule Name	Enter a unique name for the schedule.
Hostgroup	Select the host group that you want to scale. The list of available host groups is determined by which host groups include services that can be scaled.
Target Node Count	Enter the number of nodes that you want to upscale or downscale the host group to.

Repeat	Currently, only repeating schedules are supported.
CRON Expression	Enter a CRON expression string to define the details of the schedule that you want to create.
Time Zone	Select the appropriate timezone on which to base the CRON expression.

5. Click **Add**. The policy appears under Auto Scaling on the **Provision Data Hub** page.

Configuring schedule-based autoscaling with the CDP CLI for Cloudera Data Hub clusters using Impala

Use the CDP CLI to add a schedule-based autoscaling policy for Impala clusters and define the policy parameters.

To create an autoscaling policy, use the CLI command `create-auto-scale-rules:`

```
create-auto-scale-rules
--cluster-name <value>
--auto-scale-policies <value>
[--enabled | --no-enabled]
[--cli-input-json <value>]
[--generate-cli-skeleton]
```

Option	Description
`--cluster-name (string)`	Name or CRN of cluster on which to create the autoscaling policy.
`--auto-scale-policies (array)`	JSON array containing the actual policy values.
`[--enabled | --no-enabled] (boolean)`	Enables or disables the autoscaling policy on the cluster.
`[--cli-input-json] (string)`	Performs service operation based on the JSON string provided. The JSON string

	follows the format provided by `--generate-cli-skeleton`. If other arguments are provided on the command line, the CLI values will override the JSON provided values.
`[--generate-cli-skeleton] (boolean)`	Prints a sample input JSON to standard output. Note the specified operation is not run if this argument is specified. The sample input can be used as an argument for `--cli-input-json`.

Example JSON syntax for a schedule-based policy:

```
cdp datahub create-auto-scale-rules --cli-input-json \
"{
    \"clusterName\": \"cluster_name\",
    \"autoScalePolicies\": [
            {
                \"hostGroups\": \"host_group\",
                \"scheduleBasedPolicy\": {
                    \"schedules\": [
                        {
                            \"configuration\": {
                                \"trigger\": {
                                    \"cronExpression\": \"cron_expression\",
                                    \"timeZone\": \"time_zone\"
                                },
                                \"action\": {
                                    \"resourceAdjustmentType\": \"ABSOLUTE_COUNT\",
                                    \"resourceAdjustmentValue\": resource_count
                                }
                            },
                            \"identifier\": \"policy_name\",
                            \"description\": \"description\"
                        }
                    ]
                }
             }
        ],
    \"enabled\": true,
    \"useStopStartMechanism\": false
}"
```

Other autoscaling CLI commands include:

- list-scaling-activities
- describe-scaling-activity
- delete-auto-scale-rules
- describe-auto-scale-rules
- list-auto-scale-history
- update-auto-scale-rules

See the Data Hub CDP CLI documentation and Managing autoscaling for details.