

Define job scaling and redundancy settings

Last Modified on 01/12/2021 12:51 pm EST

V9.3

Overview

Scaling enables higher job throughput and improves system performance. You can set up the scaling for Service Bus Listener and Kafka Subscriber jobs. With scaling you can have multiple JES instances running the same job.

Service Bus Listener Job Definition ?

Job Command

Name *
TEST

Description

Scaling *

Single Instance
 Limited Instances
 Infinite Instances

Job is enabled

Job Instance
Select the server where the job will be executed
Any Host

Job supports redundancy

Back Next Cancel

Only Service Bus Listener and Kafka Subscriber jobs support scaling.

Scaling options

You create and manage JES jobs on the Administration site, at **Jobs Management>Add job**.

NOTE

Starting with V9.6, the job history in the Jobs Management displays the hostname for a job. This helps you manage redundant jobs efficiently.

The scaling options are available only when you set up the Service Bus Listener or Kafka Subscriber activities.

Scaling option	Description
Single Instance	<p>No scale. Only one JES runs the job.</p> <p>Select this option to define redundancy settings. Redundancy ensures that jobs run even when the JES host is down.</p> <p>You can select on which server the job will be executed:</p> <ul style="list-style-type: none"> • Any Host: The job can run on any of the available hosts. JES allocates a host, based on availability and other efficiency parameters. • Specific host: As long as the selected host is available, the job will always run on this host. If the selected host fails, JES allocates a different host based on availability and other efficiency parameters. <p>All job types support redundancy.</p>
Limited Instances	<p>Set the number of JES instances that can run the job. All JES instances will run the job until they reach the limit number. You can change this number after the job is created.</p> <p>Valid input values range from 2 to 2147483646.</p>
Infinite Instances	<p>When you select this option, every JES will run the job.</p>

NOTE

You cannot change the selected scaling option after job creation.

V9.2 and earlier

Overview

Job redundancy ensures that jobs run even when the hosting Job Execution Service (JES) is down. When the JES that hosts the job is down, the redundant job is reassigned to a JES running on a different machine, consequently, the job keeps its availability and ability to run.

You can define job redundancy and load balancing in systems that include more than one server running a JES.

All job types support redundancy.

Configure scaling

1. To set up redundancy, go to **Administration>Jobs Management>Add job**.

2. Set one of the following options:

Selection	Result
<p>Job Host Select the server where the job will be executed</p> <p><input type="text"/></p> <p><input checked="" type="checkbox"/> Job supports redundancy</p>	<p>The job can run on any of the available hosts. JES allocates a host, based on availability and other efficiency parameters.</p>
<p>Job Host Select the server where the job will be executed</p> <p>HostName <input type="text"/></p> <p><input checked="" type="checkbox"/> Job supports redundancy</p>	<p>As long as the selected host is available, the job will always run on this host. If the selected host fails, JES allocates a different host based on availability and other efficiency parameters.</p>
<p>Job Host Select the server where the job will be executed</p> <p>HostName <input type="text"/></p> <p><input type="checkbox"/> Job supports redundancy</p>	<p>The job can only run on the selected host. If the selected host fails, the job does not execute.</p>
<p>Job Host Select the server where the job will be executed</p> <p><input type="text"/></p> <p><input type="checkbox"/> Job supports redundancy</p>	<p>Not a valid option.</p>

IMPORTANT

You cannot change redundancy and host selection settings after job creation.

NOTE

Redundancy is available for new jobs only. Customers with existing jobs need to run a script to apply redundancy to existing jobs.

You can download the required script from [here](#).

