
Snaplex Capacity Tuning Guide

Introduction	2
Snaplex Planning	3
Snaplex Network Requirements	4
Pipeline Execute	4
Ultra Pipelines	5
Pipeline Load Balancing	6
Snaplex Resource Management	7
Capacity Planning	7
Capacity Tuning	8
Memory Configuration thresholds	10
Snaplex Alerts	10

Introduction

This document serves as a guide to help you optimally configure and tune your Snaplex environments for efficient processing of integration workloads.

The content and recommendations within this document are intended for a SnapLogic **Administrator** or **Architect**, in addition to any individuals who may have an influence on the deployment of nodes and configurations of the Snaplex.

Authors: SnapLogic Enterprise Architecture team

Snaplex Planning

Snaplexes are a grouping of co-located nodes which are treated as a single logical entity for the purpose of Pipeline execution. The SnapLogic Control plane automatically performs load balancing of Pipeline workload within a Snaplex. Nodes in Snaplexes should be homogeneous, with the same CPU/memory/disk sizing and network configurations per node type (i.e. JCC / FeedMaster). The JCC and Feedmaster nodes in a Snaplex can be of different sizes.

Examples of recommended configurations:

Snaplex configurations
JCC node count - 4 JCC node size for each node - Large
Feedmaster node count - 2 Feedmaster node size for each node - Medium
JCC node count - 4 JCC node size for each node - X-Large
Feedmaster node count - 2 Feedmaster node size for each node - Large

Object	Definition
Node	A Node is a JVM (Java Virtual Machine) process which is installed on a server such as Windows or Linux.
JCC Node	The JCC node is responsible for: <ul style="list-style-type: none"> - Preparation, validation, and execution of Pipelines. - Send heartbeat to the Snaplogic Control plane indicating the health of the node.

FeedMaster Node	The Feedmaster node acts as an interface between the JCC nodes and the client. The main functions of a Feedmaster node are: <ul style="list-style-type: none"> - Manage message queues - Send heartbeat to the Snaplogic Control plane indicating the health of the node.
-----------------	---

When setting up Snaplexes, it is recommended to plan out the number of Snaplexes to configure along with the usage criteria to achieve isolation across workloads. Snaplexes can be organized in various ways such as:

- **Pipeline Workload** - Organize Snaplexes by workload type: Batch, Low latency, and On-demand.
- **Business Unit** - Organize Snaplexes by business units.
- **Geographical location** - Organize Snaplexes by data center or geographic location.

The recommendation is to use a combination of the above to optimize resource usage and achieve workload isolation.

Snaplex Network Requirements

Snaplexes should have the below network characteristics:

Within a Snaplex:

- Less than 10 ms round trip latency between Snaplex nodes.
- Greater than 40 MB/sec throughput between Snaplex nodes.

Snaplex to Control Plane:

- Less than 50 ms round trip latency to the SnapLogic Control plane.
- Greater than 20 MB/sec throughput to the SnapLogic Control plane.

Pipeline Execute

Pipeline execution using the **Pipeline Execute** Snap, nodes communicate with each other using HTTPS on port 8081. There is some resiliency to network failures and HTTPS requests are retried in the case of failures. Even though requests are retried, high network latency and dropped connections can result in Pipeline execution failures.

Regular Pipeline executions run within a node, requiring no communication with other nodes in the Snaplex. When a **Pipeline Execute** snap is used to run child Pipelines, there are three options:

OPTION	COMMENTS
LOCAL_NODE	This option is recommended when the child Pipeline is being used for Pipeline structuring and reuse rather than Pipeline workload distribution. Use this option for most regular child Pipeline executions.
LOCAL_SNAPLEX	The network communication is optimized for streaming data processing since the child Pipeline is on the local Snaplex. Use this option only when workload distribution within the Snaplex is required.
SNAPLEX_WITH_PATH	This has high dependency on the network. The network communication is optimized for batch data processing since the child Pipeline is on a remote Snaplex. Use this option only when the child Pipeline has to run on a different Snaplex, either because of endpoint connectivity restrictions or for workload distribution.

Ultra Pipelines

The JCC nodes communicate with the Feedmaster nodes over TCP with SSL on port 8084 when executing Ultra Pipelines. The communication between nodes is based on a message queue.

This communication is not resilient to network failure, so a reliable network is required between the Snaplex nodes for Ultra Pipeline processing. In case of any network failures, the currently processing Ultra requests will be retried or in some instances fail with errors.

If there is a communication failure between the JCC and Feedmaster nodes, then the request will be retried for up to five times. This is controlled by the `ultra_max_redelivery_count` Snaplex configuration. There is an overall 15-minute timeout for an Ultra request to the Feedmaster that is configurable at the request level using the `X-SL-RequestTimeout` HTTP request header or at the Snaplex level by using the `lfeed.request_timeout` config setting.

Note that both `ultra_max_redelivery_count` and `lfeed.request_timeout` are configured under Node Properties -> Global Properties for GroundPlexes. *You can submit a support request to configure these properties for your Cloudplexes.*

Pipeline Load Balancing

The Control plane performs load balancing for Pipeline execution requests on a Snaplex. The following table lists the configurations that are involved:

Property / Threshold	Where configured	Default value	Comments
Maximum Slots	Node properties tab of the Snaplex	4000	<p>One slot = One Snap = One active thread on the node</p> <p>A percentage of slots (configurable with the Reserved slot % property) are reserved for interactive Pipeline executions and validations thru the Designer tool.</p> <p>Pipelines will be queued if the threshold is reached.</p> <p>Some Snaps such as Pipeline Execute, Bulk loaders, and Snaps performing input/output, can use a higher number of threads compared to other Snaps.</p>
Maximum memory %	Node properties tab of the Snaplex	85 (%)	Threshold at which no more Pipelines will be assigned to a node
Snaplex node resources (CPU, FDs, Memory)	Node server configurations	Configurable	<p>If the Control plane detects that there are not enough resources available on the Snaplex, then the Pipeline execution requests will be queued up on the control plane, and resume when resources are available.</p> <p>The Control plane dispatches the Pipeline to the node which has the most available capacity in terms of CPU/memory and file descriptors. For child Pipeline executions using the <i>Pipeline Execute</i> Snap, there is a preference given for running the child</p>

			on the local node to avoid the network transfer penalty.

Table 1.0 Configurations for Pipeline load balancing

Snaplex Resource Management

Capacity Planning

This section provides some guidelines for Snaplex capacity planning and tuning.

Configuration / Use-case	Comments
Workload isolation	Isolate workloads across Snaplexes based on workload type, geographic location, and business unit.
Node sizing	Size the node (CPU, RAM, disk space) in a Snaplex based on Pipeline workload type. Batch data processing needs larger nodes while Streaming/API processing can use smaller nodes.

Maximum Slots	<p>One slot = One Snap = One active thread on the node</p> <p>A percentage of slots (configurable with the Reserved slot % property) are reserved for interactive Pipeline executions and validations thru the Designer tool.</p> <p>Pipelines will be queued if the threshold is reached.</p> <p>Some Snaps such as Pipeline Execute, Bulk loaders, and Snaps performing input/output, can use a higher number of threads compared to other Snaps.</p> <p>The general recommendation is to configure this property based on the node memory configuration. Example:</p> <p>8 GB - 2000 Slots 16 GB - 4000 Slots</p>
Batch Workloads	<p>The general guideline is that a maximum of 5 concurrent batch Pipelines can execute 8 GB of RAM. Therefore, a 16 GB RAM machine should run at most 10 concurrent batch Pipelines.</p>
API Workloads	<p>For API workloads, the rule of thumb is to have 100 active ultra API calls per 8 GB of RAM, or 20 active triggered API calls per 8 GB of RAM. So a 16 GB node can have 200 active ultra API calls or 40 active triggered API calls.</p>
Node sizing	<p>The number of nodes in a Snaplex can be estimated based on the count of batch and streaming Pipelines.</p> <p>The number of Feedmaster nodes can be half of the JCC node count, with a minimum of two recommended for high availability.</p> <p>For active Pipeline count estimates, error Pipelines can be excluded from the count since they do not consume resources under the normal workload.</p>

Table 1.1 Configurations for Snaplex capacity planning

Capacity Tuning

Below are some best practices for Snaplex capacity tuning:

Configuration / Use-case	Comments
Slot counts	<p>The Maximum slot count can be tuned based on the alerts and dashboard events. It is not required to restart the nodes for this configuration to take effect.</p> <p><i>Queued Pipelines</i> - Increase slot count by 25% <i>Busy nodes</i> - Reduce slot count by 25%</p> <p>The slot count should not be set to more than 50% above the recommended value for the node configuration. e.g.</p> <p>The recommended slot count on a node with 16 GB RAM is 4000. Setting it to higher than 6000 is not advisable.</p> <p>If you observe high CPU / memory consumption on the node despite lowering the slot count by 25%, then consider allocating additional resources to the Snaplex nodes.</p>
Workloads	<p><i>Batch Workloads:</i> Expand the node memory up to 64 GB, and deploy additional nodes for increased capacity.</p> <p><i>API Workloads:</i> Deploy additional nodes instead of expanding the memory on the current node.</p>
Active Pipelines	<p>As a general rule, it's suggested to maintain fewer than 500 active Pipeline instances on a single node. Exceeding this threshold can lead to communication bottlenecks with the Control plane.</p> <p>If the number of active Pipeline instances exceeds 500, then the advisable course of action is to consider the addition of more nodes.</p>
CPU	<p>CPU consumption can be optimized by setting the <i>Pool size</i> and <i>Batch size</i> options on <i>Pipeline Execute</i> Snaps.</p>
Memory	<p>See Table 3.0 below</p> <p>Additional Reference: Optimizations for Swap Memory</p>

Table 2.0 Configurations for Snaplex capacity tuning

Memory Configuration thresholds

Property / Threshold	Where configured	Default value	Comments
Maximum memory %	Node properties tab of the Snaplex	85 (%)	Threshold at which no more Pipelines will be assigned to a node
Pipeline termination threshold	Internal (Can be configured by setting the feature flag at the org level <code>com.snaplogic.cc.snap.common.SnapThreadStatsPoller.MEMORY_HIGH_WATERMARK_PERCENT</code>)	95 (%)	Threshold at which the active Pipeline management feature kicks in and terminates pipelines when the node memory consumption exceeds the threshold. Ideal range: 75-99
Pipeline restart delay interval	Internal (Can be configured by setting the feature flag at the org level <code>com.snaplogic.cc.snap.common.SnapThreadStatsPoller.PIPELINE_RESTART_DELAY_SECS</code>)	30 (seconds)	One Pipeline is terminated every 30 seconds until the node memory goes below the threshold (i.e. goes below 95%) Range: 75-99

Table 3.0 Snaplex node memory configurations

The above thresholds can be optimized to minimize Pipeline terminations due Out-of-Memory exceptions. Note that the memory thresholds are based on the Physical memory on the node, and not the Virtual / Swap memory.

Snaplex Alerts

SnapLogic supports alerts and notifications through email and Slack channels. These can be configured in the Manager interface under *Settings*. The recommended alerts are listed in the table below.

Alert type	Comments
Snaplex status alerts	Status alerts can be created at the org level or the Snaplex level (in the Snaplex properties). These allow notifications to be sent when the Snaplex node is unable to communicate with the SnapLogic control plane or there are other issues detected with the Snaplex.
Snaplex Resource usage alerts	Set up alerts for these event types: <i>Snaplex congestion</i> <i>Snaplex load average</i> <i>Snaplex node memory usage</i> <i>Snaplex node disk usage</i>

Table 4.0 Recommended Snaplex Alerts

Reference:

[Alerts](#)

[Slack notifications](#)