Overview of Pulsar filters

Use Pulsar filters to make decisions based on performance telemetry or a route map. Pulsar filters are only available to you if you have purchased Pulsar Active Steering.

The following Pulsar filters are available in the Filter Chain:

- Pulsar Performance Sort: Sorts answers from best to worst performance.
- Pulsar Availability Sort: Sorts answers from best to worst availability.
- **Pulsar Performance Stabilize**: Keeps answers with the best performance, based on a defined performance threshold.
- **Pulsar Availability Threshold**: Keeps answers with the best availability, based on a defined availability threshold.
- Pulsar Route Map: Searches for a source IP match in at least one route map.

NOTE

The Pulsar Stabilize and Pulsar Sort filters have been deprecated. To learn more about the changes and to learn how you can transition to the performance and availability filters, review Transitioning to the new Pulsar availability and performance filters.

Important

You should set up a proper fallback strategy to ensure that other filters in NS1's Filter Chain can intelligently route traffic when there isn't sufficient performance telemetry. You should set up your Filter Chain to include these filters before each Pulsar filter. For example, you can use the Geotarget filters to fall back on a geotargeting strategy; or you could use **Shuffle** or **Cost** filters to better balance the lowest-cost traffic across your endpoints.

If your record contains a Pulsar filter, attach a Pulsar job to the metadata of every answer. If you do not set Pulsar job metadata for every answer, Pulsar is unable to act on its telemetry, and will bypass the product of the set of the set

?) Help

filter with no decision being made. Be sure to set other Pulsar metadata on each answer, and modify settings for each filter if needed to ensure that your performance metrics can be captured.

Pulsar Performance Sort

By default, answers are sorted from lowest to highest value. You can enable the **Sort from the worst value** option to sort in ascending order to account for certain metrics that require answers to be sorted from highest to lowest. If there is not enough performance data available to make a decision, answers are passed through unchanged and are logged as *Insufficient data* in the *Global decisions* dashboard.

The default behavior (lowest to highest) works best when your sort criteria is latency, because you want to send end users to endpoints that have lower latency values. If you're measuring bandwidth, you would sort from the highest value to sort in a way that places endpoints with higher bandwidth higher on the filtered answers list than lower-performing answers.

For example, assume that you have three endpoints: Endpoint A, Endpoint B, and Endpoint C. You set the latency threshold to 50%. Endpoint A has a response time of 160ms, Endpoint B has a response time of 110ms, and Endpoint C has a response time of 100ms. Of these three endpoints, Endpoint A is dropped, because it had a higher latency. Endpoints B and C are passed along in the order that they were passed into the filter.

Pulsar Availability Sort

By default, answers are sorted from most available to least available. If there is not enough availability data, answers are passed through unchanged, and the decisions are reported as *Insufficient data* in the *Global decisions* dashboard. You must associate each answer with a Pulsar job to gather the statistics that are used to sort answers.

Pulsar Performance Stabilize

By default, this filter determines which answer has the best latency relative to the other answers that have also passed through to this filter and removes answers beneath the stabilization threshold. This threshold value is a static value that you define; it is unrelated to answers until Pulsar must determine which answers fall below the threshold. This action provides the answers with the best latency while dropping those that do not meet the threshold. If there is not enough performance data available to make a decisir

? Help

are passed through unchanged, and these decisions are logged as *Insufficient data* in the *Global decisions* dashboard.

You can optionally enable the **Sort from the worst value** configuration option to reverse the sorting behavior. If there is not enough performance data available to make a decision, answers are passed through unchanged, and these decisions are logged as *Insufficient data* in the *Global decisions* dashboard. After determining the best or worst performance in each of the answers, the filter then compares its defined performance threshold to the other responses.

If sorting by the default behavior, any answer that is above the threshold is removed. This can be useful if, for example, you are setting a threshold for latency, where a higher value can slow the flow of traffic. Conversely, your threshold could be based on bandwidth. if you sort from lowest to highest, the lowest-performing answer will be discarded.

In the **Threshold to use for Pulsar stabilization** field, define a threshold relative to the best or worst answer (depending on your sort preference), which the filter uses to eliminate answers that fall outside of this threshold. The default threshold is 50ms; you can change this to match your personal performance requirements.

For example, Pulsar is evaluating three answers left in the Filter Chain to determine which answers may be removed. Endpoint A has a latency of 5ms, Endpoint B has a latency of 10ms, and Endpoint C has a latency of 50ms. You have set the stabilization threshold to 25ms. To determine which answers to drop, Pulsar finds the answer with the lowest latency (Endpoint A at 5ms) and adds the 25ms threshold. Because Endpoints A and B are still above the threshold (at 30ms and 35ms), these answers continue on in the Filter Chain and Endpoint C is dropped.



NOTE

ΝΟΤΕ

If you enter a number in the **Threshold to use for Pulsar stabilization** field, it will be interpreted **(?)** Help in milliseconds that will be applied to the best-performing value. To apply this value as a percentage, and a

Pulsar Availability Threshold

This filter determines which answers have the best availability based on a specified threshold, expressed as a percentage. This value is a static value that you define; it is unrelated to answers until Pulsar must determine which answers fall below the threshold. This action provides the most available answers while dropping those that do not meet the threshold. If there is not enough availability data, or if you do not specify a threshold in the **Apply availability threshold** field, answers pass through unchanged, and these decisions are logged as *Insufficient data* in the *Global decisions* dashboard.

For example, assume that you have three endpoints (Endpoint A, Endpoint B, and Endpoint C). You set the threshold to 95%. Endpoint A has an availability of 90%, Endpoint B has an availability of 97%, and Endpoint C has an availability of 100%. Endpoint A will be removed, because it's less than 95% available. Endpoints B and C are passed along in the filter chain in the order that they were passed in.



If you have set the threshold to an absolute (ms) value, the following image helps to show you how Pulsar would make a decision in that case:

170	
150	
130	
110	
90	
	ENDPOINT A ENDPOINT B ENDPOINT C

Set the threshold for answers in the settings for the filter; this sets a threshold that *every* answer still available in the Filter Chain must meet or be dropped. You can also set the threshold within the answer metadata. If you set the threshold in the answer metadata, this threshold takes priority for that answer. Click the menu icon to the right of the answer to open the *Answer Metadata* dialog, then select **Pulsar data** from the list of metadata options. Select a job from the job list, then select the **Threshold on** toggle and specify the threshold you want to use. You can set a different threshold per answer, but you must add Pulsar metadata before the answer can take advantage of Pulsar intelligent routing.

Pulsar Route Map

This filter searches for a source IP in at least one route map. If the IP is matched, answers with the associated targets are kept. This filter lets you control and automate routing behavior.

After you upload custom route maps, you can define the associated records to control and automate routing behavior. To activate the route map, you must attach it to at least one record. Maps without a corresponding record are inactive. The map itself associates IP ranges with at least one target that corresponds to the route_map_targets metadata value in an answer.

Filter Chain		Ungrouped Answers	
nable Client Subnet	-	1.1.1.1	
≡ Up	÷	up. CHI-1 • × route_maps: 2229 × route_map_targets: abc, def ×	
Pulsar Route Map	÷	2.2.2.2	
		up: MPLS-1	
🗑 Edit Filter Ch	nain	3.3.3.3	
		up: CHI-1 O × route_maps: 2229 × route_map_targets: jd ×	

You can attach multiple route maps to the same record. When you use more than one route map, the filter attempts to find a match within the first listed route map. If the filter encounters an issue with the route map, it tries the next map, continuing until it finds a matching IP range with associated target **Performance** answers.