Following best practices for using dimensions

Your environment has hundreds of independently developed services, immutable infrastructure, and frequent code pushes. You want to get the most valuable and efficient information out of your metrics monitoring, so you need information on best practices for data submission and problem detection.

Send dimensions with your datapoints

Many metrics monitoring systems make use of a pseudo-hierarchical, period-, or dot-delimited naming scheme as a way to organize and index metrics. Although Splunk Infrastructure Monitoring is compatible with this scheme, we recommend the use of explicit dimensions, sent as key-value pairs along with metric names and values. Examples of commonly used dimensions include:

- Hostname
- Datacenter
- Service
- Customer type
- Environment

Choose dimensions that are useful for filtering and aggregation

As a system that is built for flexible metric ingestion, Splunk Infrastructure Monitoring automatically creates new time series when it receives datapoints that have a combination of metric names and dimension values not previously seen. Dimensions are useful for filtering and aggregating those time series, and as such should have names that are meaningful for that purpose. Examples of commonly used dimension names include:

- Datacenter
- Environment
- Service
- Customer type
- Hostname

Using these types of dimensions, you can easily discern how the 90th percentile in latency metrics differs across datacenters or compare the resource consumption of a canary with existing production servers.

Examples of dimensions that tend not to be useful include:

- Timestamps
- Monotonically increasing values

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Using these suboptimal dimensions typically yields a very large number of very sparse time series, which do not lend themselves to useful visualization, monitoring or analysis.

Additionally, dimensions which change frequently and do not repeat - as can be the case with container IDs, for example - tend not to be useful for filtering or aggregation when dealing with dynamic infrastructure. This is an environment in which there is autoscaling, frequent cycling of containers, or frequent map reduce job starts. For more information, see Managing ephemeral infrastructure.

Do not use reserved terms in metric names, dimensions, properties, or tags

Splunk Infrastructure Monitoring reserves the prefixes of \texttt{sf\_}, \texttt{sf.num}, and \texttt{aws\_}. If you use these in custom metric names, dimension names or property names, the associated metric datapoints will not be ingested and not be available for use in any way.

Avoid large numbers of dimension names (keys)

When metric datapoints are received by Splunk Infrastructure Monitoring, they are associated with existing metric time series where possible (if the combination of dimension key-value pairs and metric name has previously been seen by our system), and trigger the creation of new time series where not.

Metric time series' dimensions and properties are stored in Elasticsearch, and are therefore subject to internal limits in Elasticsearch as to the number of unique dimension or property names that can be added per customer org. As a result, while high cardinality in dimension values (e.g. host:host1, host:host2,...host:hostN) is supported, high cardinality in dimension names (e.g. host1:true, host2:true,...hostN:true) is not. When high cardinality in dimension names is detected, new metric time series creation is halted for the customer organization in question.

In most cases, the best practice for avoiding too many dimension names is to put the variable part of the dimension name into the value instead. For example, if you are sending \texttt{foo\_splitBy\_*:true}, where * can have many values, then you should instead send \texttt{foo\_splitBy:*}.

Use tags on dimensions for set properties

In some cases, the use of many unique dimension names in the form of \texttt{foo\_splitBy\_*:true} is intended to express a kind of set property. For example, on a given metric time series, you might have all of: \texttt{foo\_splitBy:A}, \texttt{foo\_splitBy:B} and \texttt{foo\_splitBy:C}.

This usage is incompatible with the Splunk Infrastructure Monitoring metric data model, which only allows one key-value pair per key per metric time series. It is only possible to have \texttt{foo\_splitBy:A} or \texttt{foo\_splitBy:B} or \texttt{foo\_splitBy:C} but not any combination thereof with more than one such value of \texttt{foo\_splitBy}. 

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
For this use case, Splunk Infrastructure Monitoring supports set properties in the form of tags on dimensions.

1. Create a dimension with the name of foo_splitBy.
2. For the value use a sorted, comma-separated list of intended values, for example, foo_splitBy:“A,B,C”.
3. Adding the corresponding tags to the dimension. For example, foo_splitBy:A, foo_splitBy:B, and foo_splitBy:C for the foo_splitBy:“A,B,C” dimension to allow the desired behavior of filtering or aggregating by a given foo_splitBy property.

Avoid reusing dimensions for different entities over time

Splunk Infrastructure Monitoring provides three different classes of metadata to accommodate different use cases. It is common to use properties or tags for metadata coming from public cloud vendors like AWS, as such information is readily available via API calls that are distinct from those used for fetching metric data. Those properties or tags are added to dimensions, then ‘propagated’ to each metric time series that includes that dimension. This propagation allows the metric time series to be correctly aggregated or filtered using the property or tag.

For example, if a given EC2 instance is first part of the service foo, and then subsequently repurposed for service bar, then the typical metadata used for this scenario would have service:foo and service:bar set up as AWS custom tags that are imported as Splunk Infrastructure Monitoring properties, then associated with that EC2 instance (first with foo, then with bar) by adding the correct property 'service:*' to a dimension that uniquely identifies that EC2 instance. Doing so allows a user to filter or aggregate the metric time series from that instance using the property value, even as the instance is repurposed across services.

As this example shows, properties and tags are generally assumed to be mutable, and not part of the identity of a metric time series. If properties are then subsequently added to dimensions that are reused, however, then there is the potential for a collision, as dimensions are part of the identity of a time series, and meant to be associated with an actual, unique entity.

To continue the previous example, if the dimension that identifies the EC2 instance is a hostname host:baz, and that hostname is reused for another EC2 instance after the first instance is decommissioned, then it is not clear whether service:foo is supposed to be associated with the first or the second instance, and subsequently to those instance’s associated metric time series. The reuse causes properties to be incorrectly propagated, which can be apparent to the user as a problem, and is also increasingly inefficient over time for the metadata system.

Because of this potential problem, the recommended best practice is the use of unique dimensions for unique entities, for example, by adding and using an AWSUniqueld dimension on all metrics which also have a host dimension. After this is completed, syncing AWS properties to the AWSUniqueld dimension only can be done in an efficient manner and avoids the problems stated above.

For EC2 instances, in particular, it’s recommended that the AWSUniqueld is generated from <instance_id>_<region>_<AWS_account_id> because none of these are individually guaranteed to be unique, but the combination thereof is. All three of these are available on the instance itself via the AWS instance metadata API.
Next steps

These additional Splunk resources might help you understand and implement these recommendations:

- Splunk Docs: [Dimension name and value standards](#)
- Splunk Docs: [Separate dimensions from metrics names](#)
- Splunk Docs: [Metric metadata](#)