

# Dashboards Setup

## Variables

Get the value from a tag

```
> show tag values from "snapdb" with key="db"
```

Get the value from a field

```
> select DISTINCT("vmname") from (select "ready_summation","vmname" from "vsphere_vm_cpu")
```

## Query Editor

### InfluxQL with InfluxDB v2

```
SELECT * FROM cpu WHERE time >= $__timeFrom AND time <= $__timeTo  
SELECT * FROM cpu WHERE $__timeFilter(time)  
SELECT $__dateBin(time) from cpu
```

### Time Series: vSphere Cluster CPU Usage

```
SELECT mean("usage_average")  
FROM "vsphere_host_cpu"  
WHERE ("clustername" =~ /^$clustername$/ AND "cpu" = 'instance-total') AND $timeFilter  
GROUP BY time($inter), "clustername", "cpu" fill(none)
```

### Gauge: vSphere Datastore Status

```
SELECT mean("used_latest") * (100 / mean("capacity_latest"))  
FROM "vsphere_datastore_disk"  
WHERE ("source" =~ /^$datastore$/) AND $timeFilter  
GROUP BY time($inter) fill(none)
```

### Bar gauge: vSphere Datastore Usage Capacity

```
SELECT last("used_latest") * (100 / last("capacity_latest"))
FROM "vsphere_datastore_disk"
WHERE ("source" =~ /^$datastore$/) AND $timeFilter
GROUP BY time($inter) , "source" fill(none)
```

## Stat: Uptime

```
SELECT last("uptime_latest") AS "Uptime"
FROM "vsphere_host_sys"
WHERE ("vcenter" =~ /^$vcenter$/ AND "clustername" =~ /^$clustername$/) AND $timeFilter
GROUP BY time($inter) fill(null)
```

# Flux with InfluxDB v2

## Time Series: vSphere Cluster CPU Usage

```
from(bucket: v.defaultBucket)
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "vsphere_host_cpu")
  |> filter(fn: (r) => r["_field"] == "usage_average")
  |> filter(fn: (r) => r["cpu"] == "instance-total")
  |> filter(fn: (r) => r["clustername"] =~ /^${clustername:regex}/)
  |> group(columns: ["clustername"])
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

## Gauge: vSphere Datastore Status

```
from(bucket: v.defaultBucket)
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "vsphere_datastore_disk")
  |> filter(fn: (r) => r["_field"] == "capacity_latest" or r["_field"] == "used_latest")
  |> filter(fn: (r) => r["source"] =~ /^${datastore:regex}/)
  |> group()
  |> pivot(rowKey: ["_time"], columnKey: ["_field"], valueColumn: "_value")
  |> map(fn: (r) => ({ r with _value: float(v: r.used_latest) / float(v: r.capacity_latest) * 100.0 }))
  |> group(columns: ["source", "_field"])
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
```

## Bar gauge: vSphere Datastore Usage Capacity

```
from(bucket: v.defaultBucket)
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "vsphere_datastore_disk")
  |> filter(fn: (r) => r["_field"] == "capacity_latest" or r["_field"] == "used_latest")
  |> filter(fn: (r) => r["source"] =~ /${datastore:regex}/)
  |> pivot(rowKey:["_time"], columnKey: ["_field"], valueColumn: "_value")
  |> map(fn: (r) => ({ r with _value: float(v: r.used_latest) / float(v: r.capacity_latest) * 100.0 }))
  |> group(columns: ["source", "_field"])
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
```

## Stat: Uptime

```
from(bucket: v.defaultBucket)
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "vsphere_host_sys")
  |> filter(fn: (r) => r["_field"] == "uptime_latest")
  |> filter(fn: (r) => r["vcenter"] =~ /${vcenter:regex}/)
  |> filter(fn: (r) => r["clustername"] =~ /${clustername:regex}/)
  |> group(columns: ["clustername"])
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

---

Revision #11

Created 14 July 2021 05:41:08 by Admin

Updated 8 February 2024 11:43:53 by Admin