

# *Grafana Plugins, Past and Future Plus New Cloud Data Sources*

*Daniel Lee, Grafana Labs*

# Agenda

- Cloud Datasources
- Google Stackdriver presentation - Joy Wang
- Oracle Cloud Infrastructure presentation - Mies Hernandez van Leuffen
- Community Plugins
- Future of Grafana Plugins

# *Grafana Now Supports The 3 Major Clouds*

- Grafana Labs uses all 3 clouds and so do lots of Grafana users
- Avoid vendor lock-in - hybrid/multi-cloud
- Fits with our core vision to democratize metrics
- Helping teams better understand their systems by unifying, visualizing, and analyzing their measurement data - wherever it comes from

# *Supported Cloud Datasources*

- AWS Cloudwatch
  - Created in 2015 by Mitsuhiro Tanda
- Azure Monitor
  - First as a plugin. Included in Grafana 6.0 as a core data source
- Google Stackdriver
  - Released as beta for 5.3. Officially released in Grafana 6.0

# *Introduction to Azure Monitor*

- Supports 4 Azure Services
  - Azure Monitor
  - Application Insights
  - Azure Log Analytics
  - (Application Insights Analytics)
- Support for:
  - Infrastructure metrics
  - application metrics
  - log-based metrics

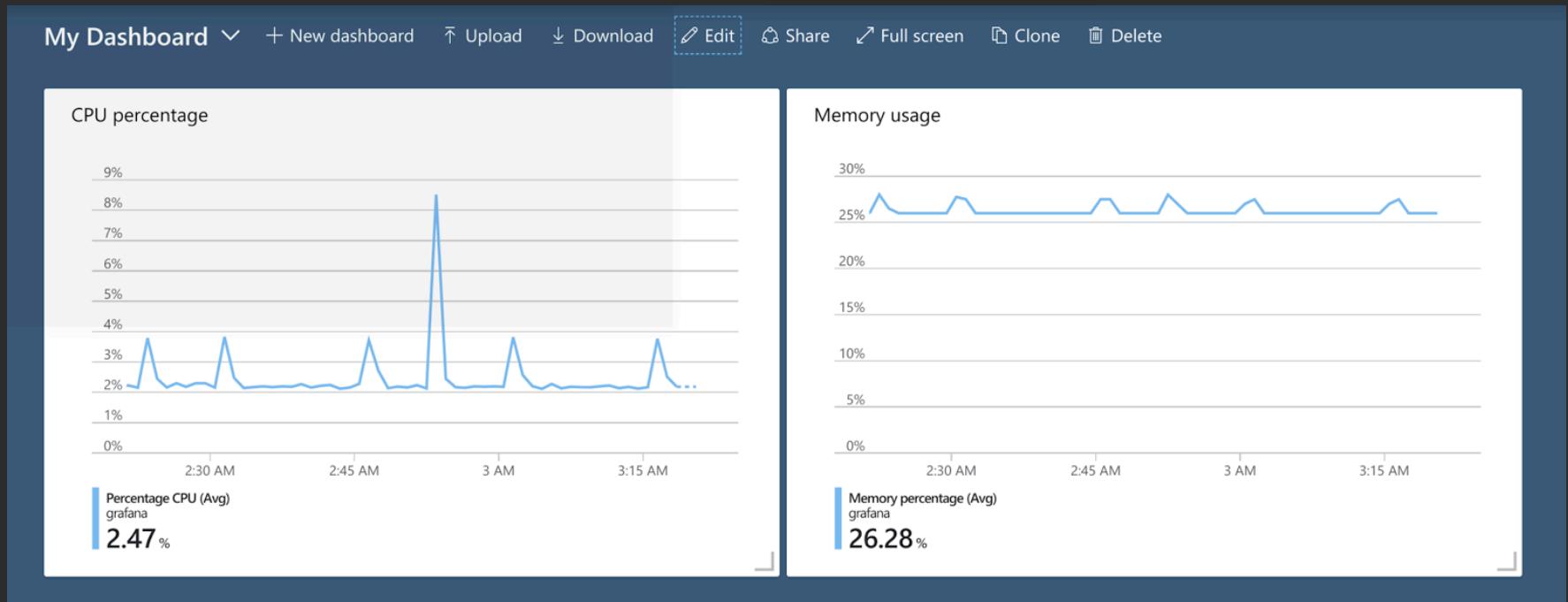
# *Azure Monitor - Why move it to Core Grafana?*

- Azure is used by lots of Grafana users
- The Grafana team is committing to a higher level of support
- Gets Grafana Alerting

# *Azure Monitor in Grafana*

Please welcome Brendan Burns from Microsoft

# Azure Portal



# Azure Monitor Dashboard in Grafana



# Azure Monitor - Support for multiple clouds

## Azure Monitor API Details

Azure Cloud	Azure China	
Subscription Id	9da108e59e55	
Tenant Id	-481a115f7cbd	
Client Id	d1ae81f44e37	
Client Secret	configured	

# Azure Monitor - Mix queries from the services

Queries to Azure Monitor ▾

Add Query    Query Inspector    ?

▼ A

Service	Azure Monitor					
Resource Group	\$rg	Namespace	Microsoft.Compute/virtualMachines	Resource Name	\$rn	
Metric	Percentage CPU	Aggregation	Average		Time Grain	1 minute
Legend Format	{{metric}}					

▼ B

Service	Application Insights					
Metric	requests/count	Aggregation	sum			
Group By	request/urlHost	Filter	your/groupby eq 'a_value'			
Time Grain	auto	Auto Interval	1m			
Legend Format	alias patterns (see help for more info)					

# Azure Monitor Query Editor

The screenshot shows the Azure Monitor Query Editor interface. At the top, there's a header with "Queries to" set to "Azure Monitor". To the right are buttons for "Add Query", "Query Inspector", and a help icon. On the left, there's a vertical sidebar with four icons: a yellow cylinder (Metrics), a blue chart (Logs), a grey gear (Metrics), and a bell (Logs).

The main area contains a query configuration panel:

- Service:** Azure Monitor
- Resource Group:** \$rg
- Metric:** Percentage CPU
- Legend Format:** {{metric}}
- Namespace:** Microsoft.Compute/virtualMachines
- Resource Name:** \$rn
- Aggregation:** Average
- Time Grain:** 1 minute

Below the configuration are time-related controls:

- Relative time: 1h
- Time shift: 1h
- Hide time info: toggle switch

# Azure Log Analytics Query Editor

Queries to Azure Monitor

Add Query Query Inspector ?

**A**

Service Azure Log Analytics

Workspace danieltest

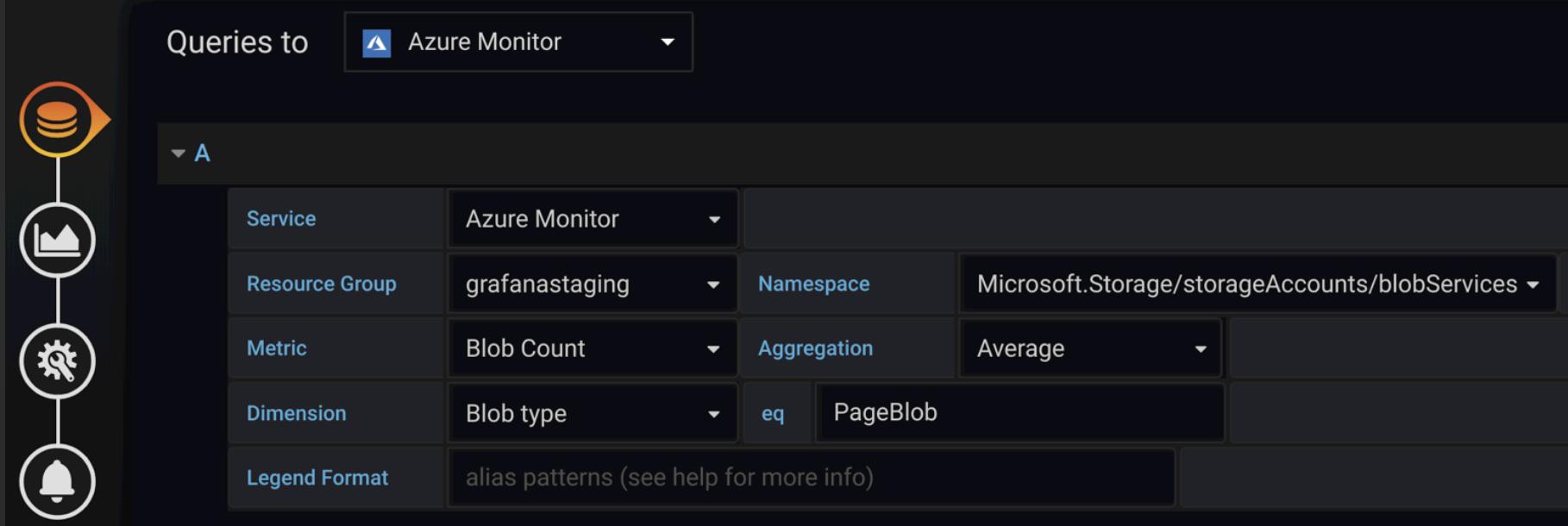
(Run Query: Shift+Enter, Trigger Suggestion: Ctrl+Space)

Run

```
AzureActivity  
| where $__timeFilter() and Category == "$Category"  
| summarize count() by Category, bin(TimeGenerated, $__interval)  
| order by TimeGenerated asc
```

Format As Time series Show Help ▾

# Azure Monitor - Dimension Filtering



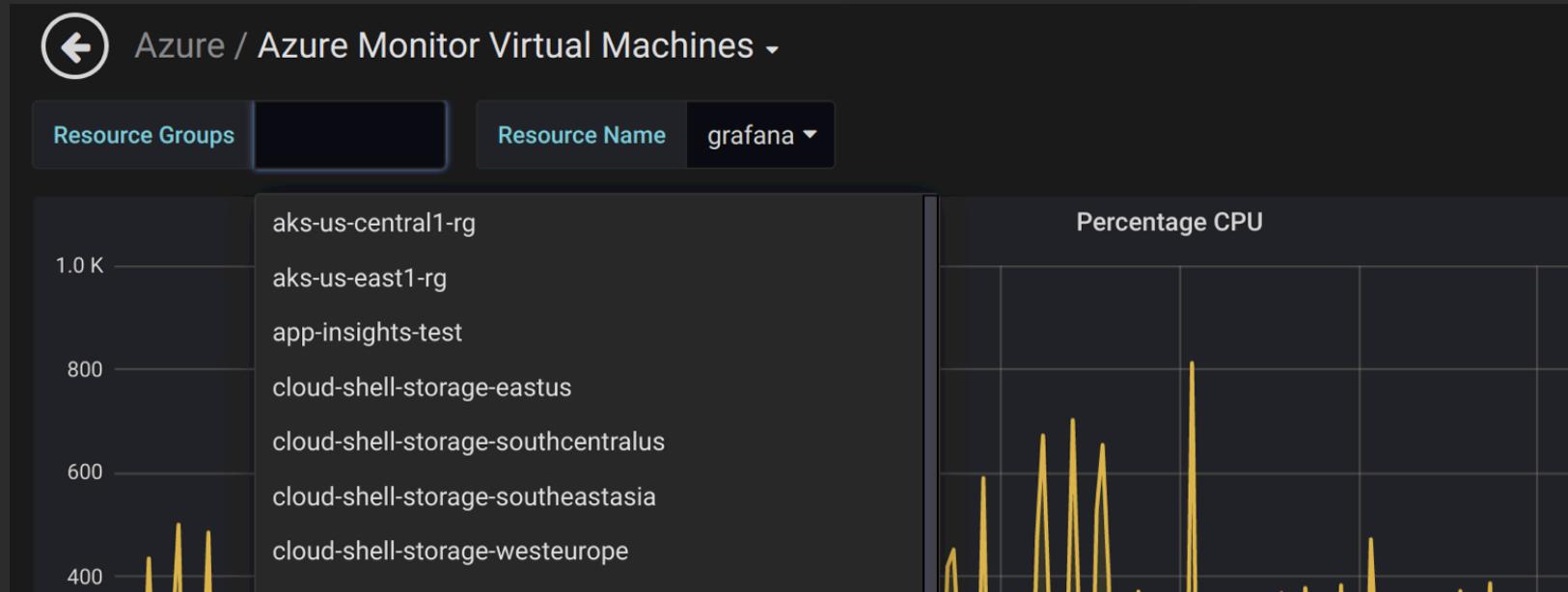
The screenshot shows the Grafana interface for querying Azure Monitor data. On the left, there's a vertical sidebar with five icons: a yellow cylinder with an orange arrow, a blue line graph, a grey gear, and a white bell. The main area has a dark background with light-colored input fields.

**Queries to:** Azure Monitor

**Panel A:**

Service	Azure Monitor		
Resource Group	grafanastaging	Namespace	Microsoft.Storage/storageAccounts/blobServices
Metric	Blob Count	Aggregation	Average
Dimension	Blob type	eq	PageBlob
Legend Format	alias patterns (see help for more info)		

# Azure Monitor - Templating



# Azure Monitor service has alerting

Alert



Rule

Name Percentage CPU Alerting alert Evaluate every 1m For 1m ⓘ

Conditions

WHEN avg () OF query (A, 24h, now) IS ABOVE 10 ⚡ +

No Data & Error Handling

If no data or all values are null	SET STATE TO	No Data ▾
If execution error or timeout	SET STATE TO	Alerting ▾

# Azure Monitor - Provisioning

```
apiVersion: 1

datasources:
  - name: gdev-azureMonitor
    type: grafana-azure-monitor-datasource
    access: proxy
    jsonData:
      azureLogAnalyticsSameAs: false
      cloudName: azuremonitor

      clientId: 999b7ed8-1aed-486e-9789-819770bbb999
      subscriptionId: 99993801-6ee6-49de-9b2d-9106972f9999
      tenantId: 9993f661-a933-4b3f-8176-51c4f982e999

      appInsightsAppId: DEMO_APP

      logAnalyticsClientId: 9992db7c-6f6f-4e88-93d9-9ae126c0a999
      logAnalyticsDefaultWorkspace: 9991b44e-3e57-4410-b027-6cc0ae6de999
      logAnalyticsSubscriptionId: 99993801-6ee6-49de-9b2d-9106972f9999
      logAnalyticsTenantId: 9993f661-a933-4b3f-8176-51c4f982e999

    securejsonData:
      clientSecret: MyVerySecretClientSecret

      appInsightsApiKey: DEMO_KEY

      logAnalyticsClientSecret: MyVerySecretLogAnalyticsClientSecret
```



# *Stackdriver in Grafana*

Please welcome Joy Wang from Google

# Stackdriver

Stackdriver is a broad suite of products to improve the development experience on GCP as well as other cloud environments:

## Monitoring

- Platform, system, app, and custom metrics
- Uptime/health checks
- Dashboards and charts
- Alerts and notifications

## Logging

- Platform, system, app, and audit logs
- Log search/view/filter
- Logs-based metrics
- Error notification and Dashboard

## APM

- Trace - Latency analysis across distributed apps
- Profiler - CPU and memory profiling
- Debug - In production debug and conditional snapshots

## IRM

- Command and control system for incidents
- Team collaborations
- Post-mortems and communications



Grafana Labs  
Incidents repository

# *Why Did We Build the Plugin Together*

We Hear Our Users' Feedback And We Took Actions

We ❤️ Open Source

We Learn From The Community

We Continuously Improve

# Stackdriver and Grafana Stackdriver Plug-in



# Stackdriver Dashboard Next Steps

The image displays three side-by-side screenshots of the Google Stackdriver Dashboard, illustrating various monitoring features and chart configurations.

**Screenshot 1: System Health - GCE Instances - CPU Utilization**

This screenshot shows the "System Health" dashboard for GCE instances. The left sidebar includes filters for "GCE", "Group", "Instance\_id", "Zone", "Project\_id", and "Metric Label". The main area displays a chart titled "Instances - CPU Utilization" showing CPU usage over time (10:30 AM to 3:30 PM). A legend indicates multiple series: GCE VM Instance (yellow), Metric: CPU Usage (orange), and GCE only by zone (blue).

**Screenshot 2: System Health - Saved views**

This screenshot shows the "System Health" dashboard with a focus on saved views. A modal window titled "Select a saved view to apply" lists several options: "Album recommendation service" (highlighted in blue), "GCE only by zone" (highlighted in yellow), "MongoDB (zone and version)", "No GAE or GCE", "Only recent deployment", and "Restricted access". The background shows the "Instances - CPU Utilization" chart from the first screenshot.

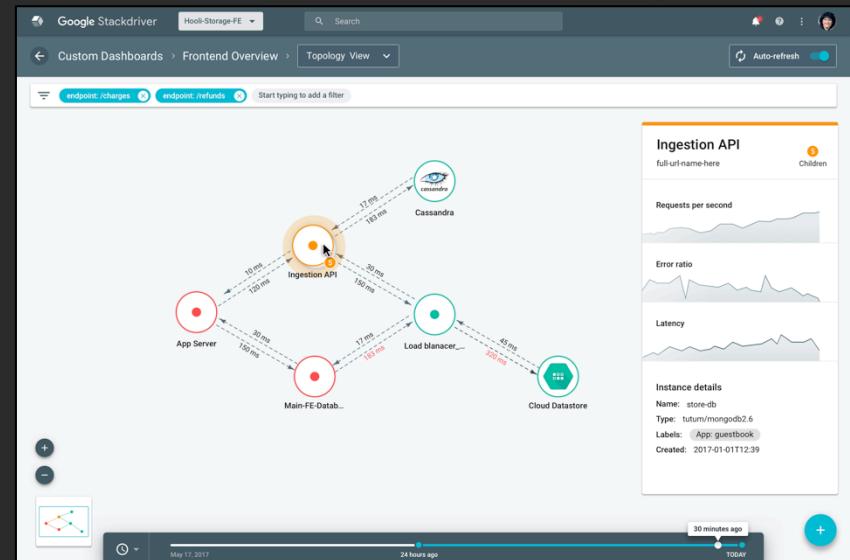
**Screenshot 3: System Health - Instances - Network Outbound Traffic**

This screenshot shows the "System Health" dashboard with a focus on network traffic. It displays two charts: "Instances - Network Outbound Traffic" and "Volumn Usage (agent)". The "Instances - Network Outbound Traffic" chart shows network traffic over time (10:30 AM to 3:30 PM) with multiple colored lines. The "Volumn Usage (agent)" chart shows volume usage over time (10:30 AM to 3:30 PM) with blue and red lines. The top navigation bar shows "Workspace Musicify Recommendation Engine".

abs

# *Stackdriver Monitoring Also Introducing*

- Kubernetes monitoring
- Expanded multi-cloud and hybrid observability
- Metric query language
- SLO monitoring
- Service graph and service dashboard
- APIs for everything
- Metric granularity and retention



# *Stackdriver in Grafana*

- Great to work with Google
- 1st datasource in React
- Can use Grafana alerting

The screenshot shows the Grafana interface for the Stackdriver plugin. At the top, there's a dropdown menu with the URL "logging.googleapis.com/user/redhat-heatmap". Below it is a section for "Template Variables" with options: \$service, \$metric, \$metriclabel, \$zone, and Metrics. The "Metrics" section is expanded, showing several metrics with descriptions:

- Log bytes ingested**: Log bytes ingested.
- Monthly log bytes ingested**: Month-to-date log bytes ingested.
- Log bytes**: Number of bytes in all log entries ingested.
- Logs-based metric errors (Deprecated)**: Number of log entries that did not contribute to user defined metrics. This metric is deprecated. Use logging.go
- Excluded log bytes**: Number of bytes in log entries that were excluded.

# *Oracle Cloud Infrastructure Datasource*

Please welcome Mies Hernandez van Leuffen from Oracle



Cloud Native Labs



# Oracle Cloud Infrastructure Data Source for Grafana

Micha “mies” Hernandez van Leuffen  
VP of Solution Development  
@mies

#OracleCloudNative  
[cloudnative.oracle.com](http://cloudnative.oracle.com)



# Safe Harbor

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

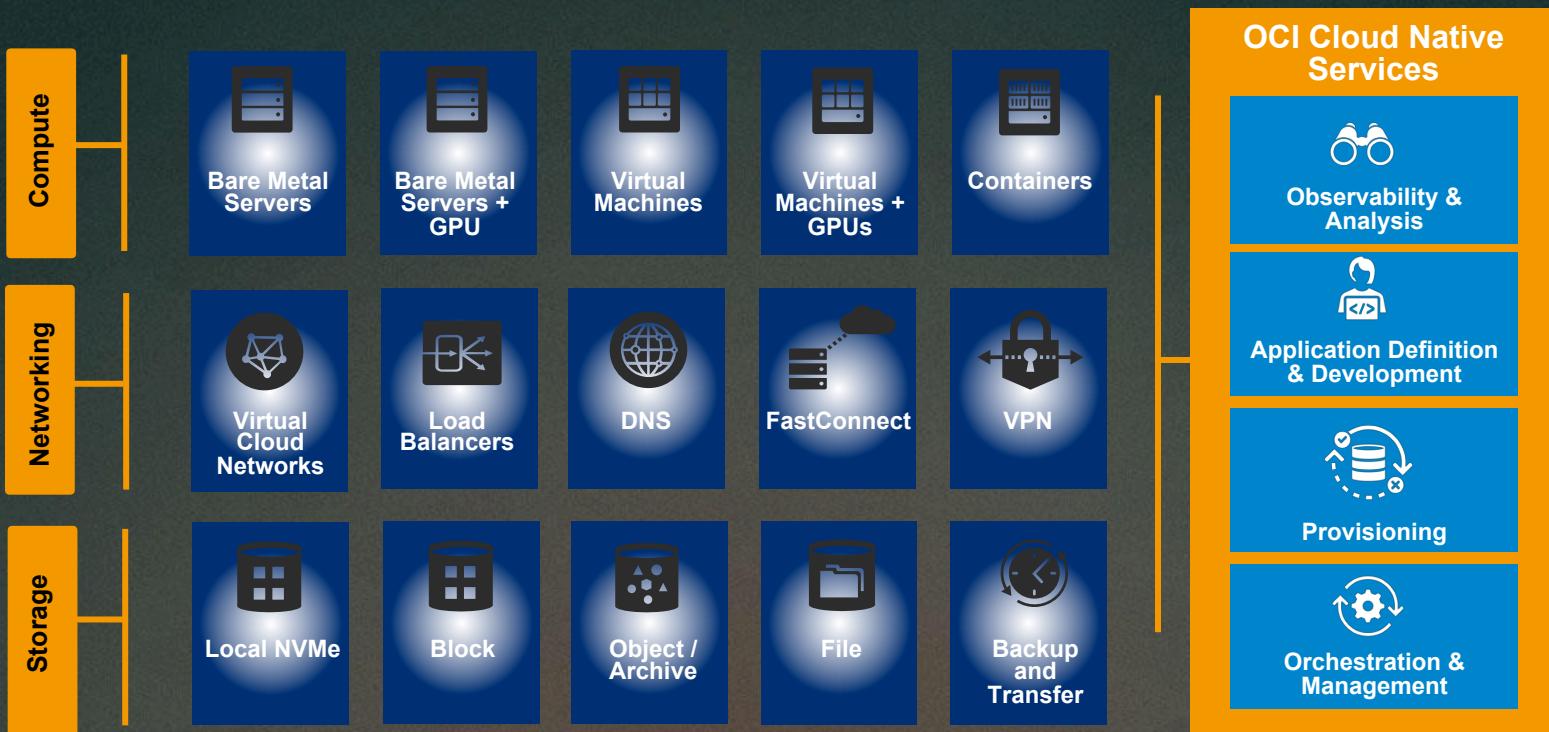
# About me and Oracle Cloud Native Labs

- Founder and CEO of Wercker; container native CI/CD
- VP of Solution Development at Oracle Cloud Native Labs
- Build customer-deployable cloud native/container native solutions to bridge the gap between Oracle Cloud Infrastructure (OCI) and open source communities.

<http://cloudnative.oracle.com>



# Oracle Cloud Infrastructure



# Oracle Cloud Native Framework – Open and Integrated



# Oracle Cloud Infrastructure Data Source for Grafana

- Data Source plug-in for the OCI Monitoring Services APIs
- Visualize data and metrics related to OCI resources
- Current metrics available:
  - Compute Agent, Block Store, LBaaS, VCN (Virtual Cloud Network)
- Supports local installation (Mac, Linux), Kubernetes & VM-based installation
- Easy configuration and setup

OCI Monitoring Services:

Telemetry service for operations staff, dashboard developers, and SRE, providing:

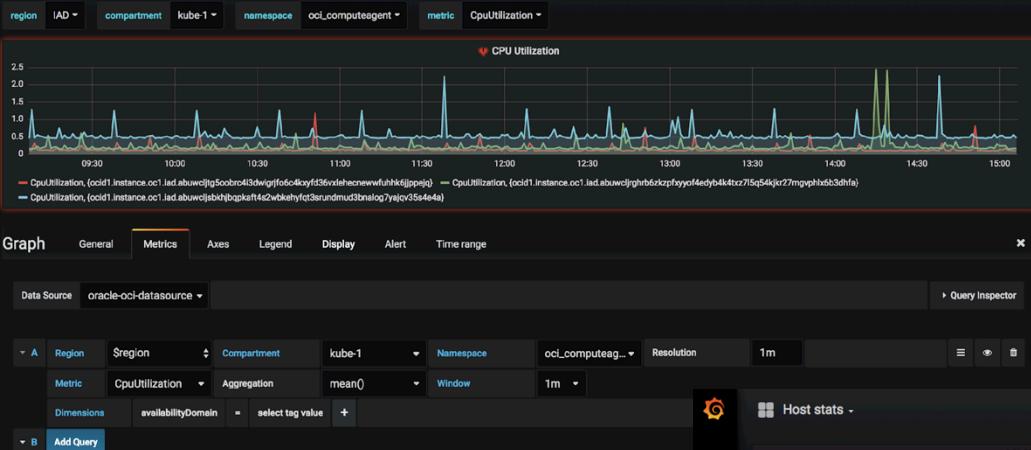
- Out of the box aggregated metrics for OCI services and resources
- Discover and retrieve metrics from Public Monitoring API
- Pre-configured dashboards in OCI console for visualization



<https://www.trybooster.com/>

Available in Grafana Marketplace and OSS at: <https://github.com/oracle/oci-grafana-plugin>

Docs : <https://github.com/oracle/oci-grafana-plugin/tree/master/docs>



# Oracle Cloud Infrastructure Data Source for Grafana

Come see more dashboards at the booth!

# Stay Connected!

Sign up for news, events and other updates on  
**Oracle Cloud Native Labs**

Scan QR code, or go to

<http://bit.ly/Oracle-CNL-GrafanaCon19>

[https://cloud.oracle.com/en\\_US/tryit](https://cloud.oracle.com/en_US/tryit) - 300 Hours Free!



Scan me

# *Grafana Plugins - Year in Review*

- Lots of new plugins
- Lots of updates and improvements to existing plugins

# *So Many New Plugins - Datasources*

- LinkSmart SensorThings (IOT)
- JSON (fork of the SimpleJson DS)
- Warp10 (IOT TSDB)
- Akumuli (new TSDB)
- DarkSky (weather)
- MetaQueries
- SumoLogic
- Thruk (Nagios, Icinga)

# *So Many New Plugins - Panels*

- ePict
- Statusmap
- Singlestat Math
- Windrose
- Boom table for Graphite
- Parity Report
- Traffic Lights
- Radar Graph
- SVG
- Multistat
- Annotations List

# *Community Plugin Highlights - Zabbix Datasource*

- Zabbix supports loadable modules
- Allows real-time history export into different databases
- InfluxDB added as Direct DB Connection datasource to plugin
- Can query history data directly from InfluxDB
- A big thanks to Gleb Ivanovsky for his Zabbix InfluxDB module - effluence

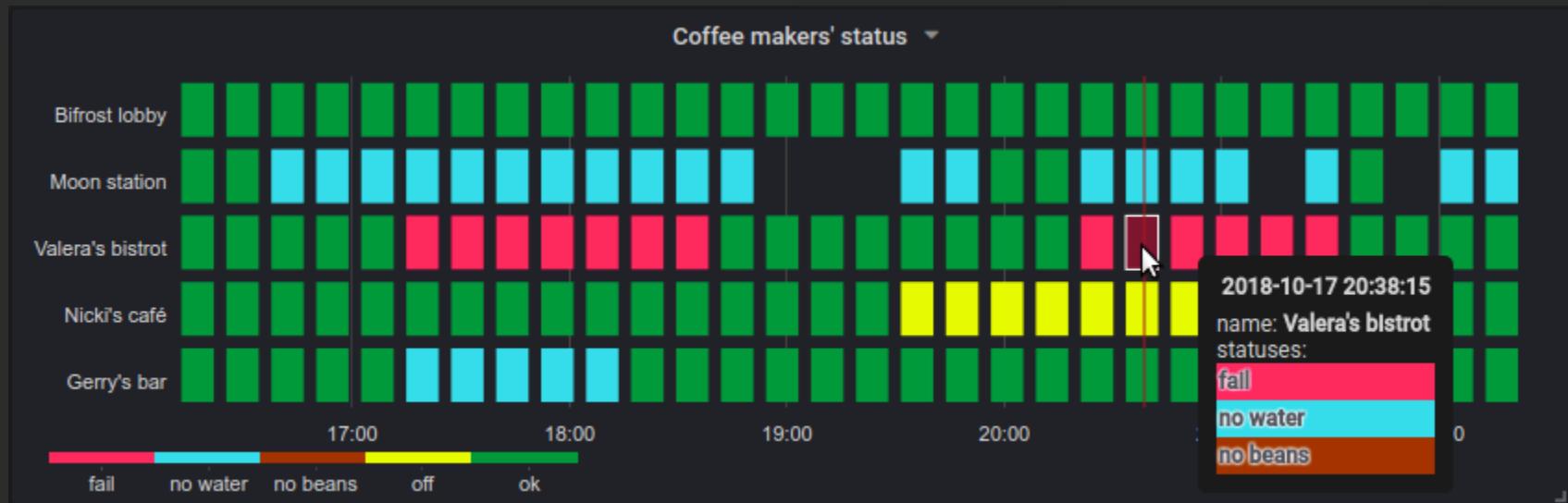
# Zabbix - new Problems Panel

Problems					
Host	Severity	Status	Problem	Time	Details
database01	Warning	PROBLEM	Processor load is too high on database01	16 Dec 2018 16:43:13	<a href="#">Details</a>
frontend01.te...	Warning	PROBLEM	Processor load is too high on frontend01	16 Dec 2018 16:17:33	<a href="#">Details</a>
backend04	Disaster	PROBLEM	Test Disaster on backend04	03 Dec 2018 20:54:47	<a href="#">Details</a>
backend04	Warning	PROBLEM	Processor load is too high on backend04	03 Dec 2018 20:54:47	<a href="#">Details</a>
backend04	Information	PROBLEM	Test Information on backend04	03 Dec 2018 20:54:47	<a href="#">Details</a>
backend04	Not classified	PROBLEM	Test Not classified on backend04	03 Dec 2018 20:54:47	<a href="#">Details</a>
backend03	Disaster	PROBLEM	Test Disaster on backend03	03 Dec 2018 20:45:32	<a href="#">Details</a>
backend03	Warning	PROBLEM	Processor load is too high on backend03	03 Dec 2018 20:45:32	<a href="#">Details</a>
backend03	Information	PROBLEM	Test Information on backend03	03 Dec 2018 20:45:32	<a href="#">Details</a>
backend03	Not classified	PROBLEM	Test Not classified on backend03	03 Dec 2018 20:45:32	<a href="#">Details</a>

Previous [Page 1 of 3](#) [10 rows](#) Next



# Multistat Panels - Statusmap



# Multistat Panels - Multistat

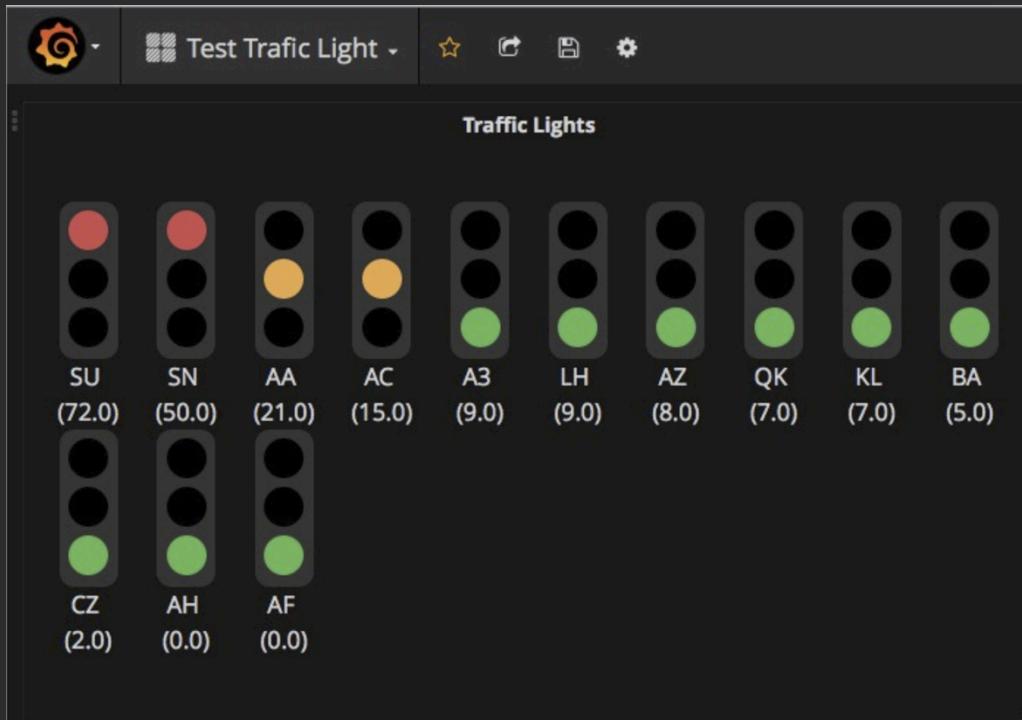


# Multistat Panels - Parity Report

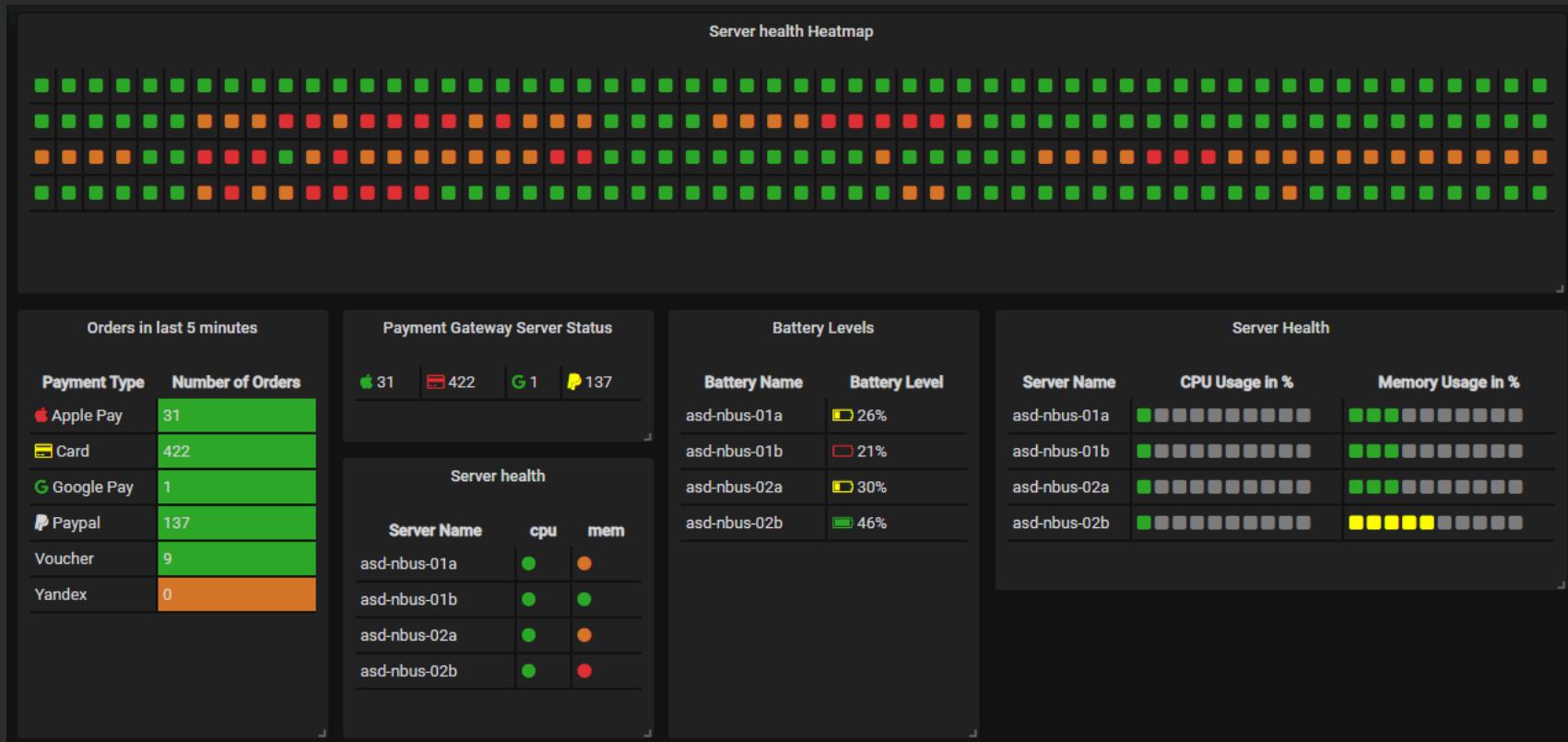
## NETWORK REPORT ▾

Name	Difference ▾	Equation
Check 1	99.99%	23.96 GB = 2.00 MB
Check 3	77.27%	5.00 MB = 22.00 MB
Check 4	76.74%	5.00 MB = 21.50 MB
Check 2	60%	5.00 MB = 2.00 MB
Check 5	60%	5.00 MB = 2.00 MB
Check 6	60%	5.00 MB = 2.00 MB

# Multistat Panels - Traffic Lights

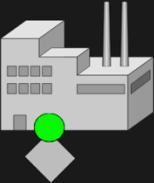


# Multistat Panels - Boom Table



# Image Panels - SVG

Panel Title



SVG General Metrics **SVG Builder** SVG Events Time range

Use SVG Builder

Canvas

Width	100%
Height	100%
Viewport X	0
Viewport Y	0
Viewport Width	1000
Viewport Height	1000

Elements

Name	ID	x	y	rotation	r-center x	r-center y	scale
plant	plant	100	100	0	0	0	2
square	square	200	600	45	75	75	1
light-green	light-green	220	500	0	0	0	1

Add new

Repository	rtdmaster
Categories	indicators
SVG	light-green

+ Add

SVG Panel - Last 6 hours  

Clock Demo



Visualization  SVG

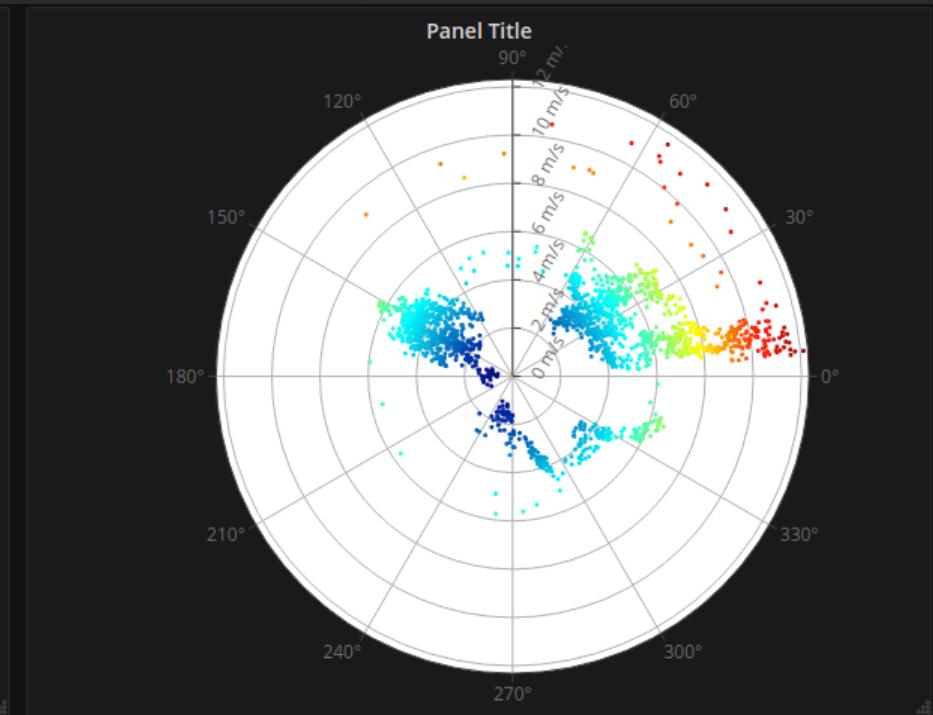
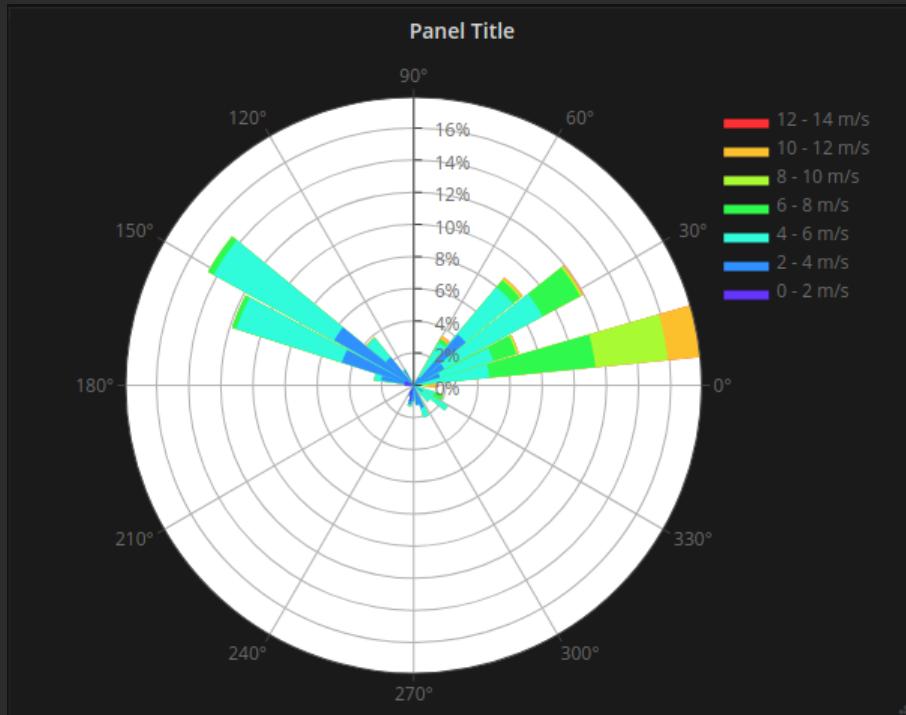
Events

JavaScript Code: `onHandleMetric(ctrl, svgnode)`

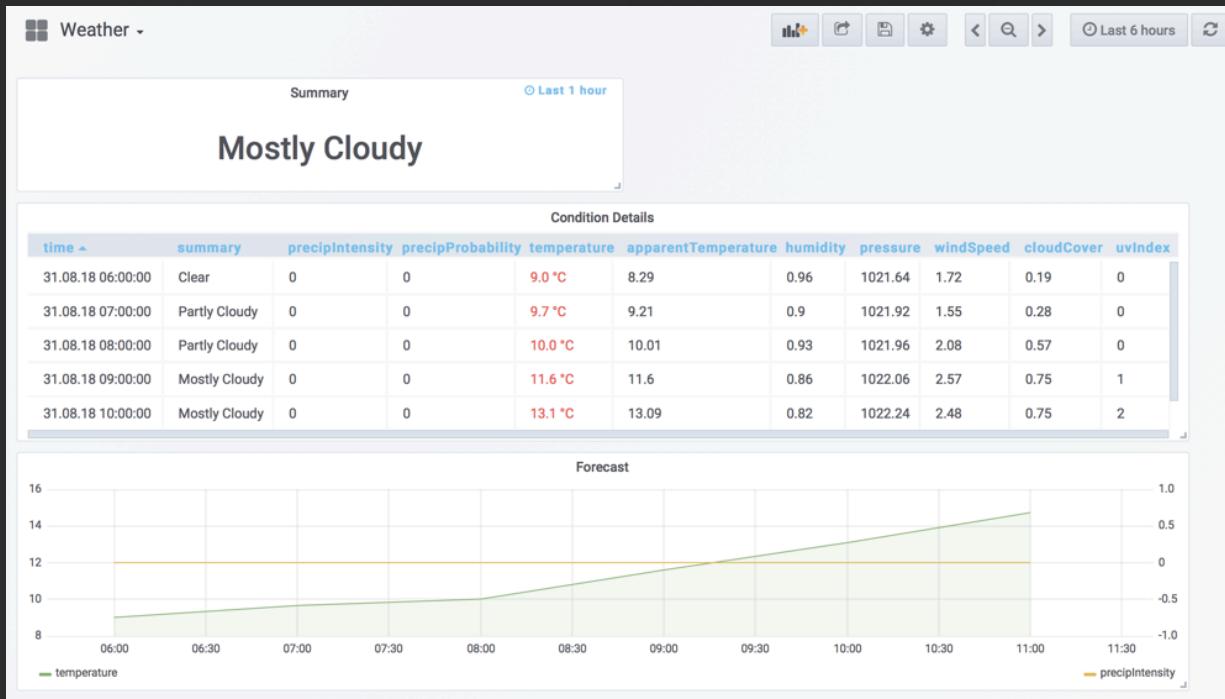
- This code is executed upon every refresh
- @param {MetricsPanelCtrl} ctrl Instance of current grafana panel object.
- @param {HTMLElement} svgnode Html DOM node containing svg data.

```
1 var now = new Date();
2 var angle = 360 * now.getHours() / 12;
3 $(svgnode).find("#hourhand").attr("transform", "rotate("+angle.toString()+" 100 100");
4 var angle = 360 * now.getMinutes() / 60;
5 $(svgnode).find("#minutehand").attr("transform", "rotate("+angle.toString()+" 100 100");
6 var angle = 360 * now.getSeconds() / 60;
7 $(svgnode).find("#secondhand").attr("transform", "rotate("+angle.toString()+" 100 100");
```

# Weather - Windrose Panel



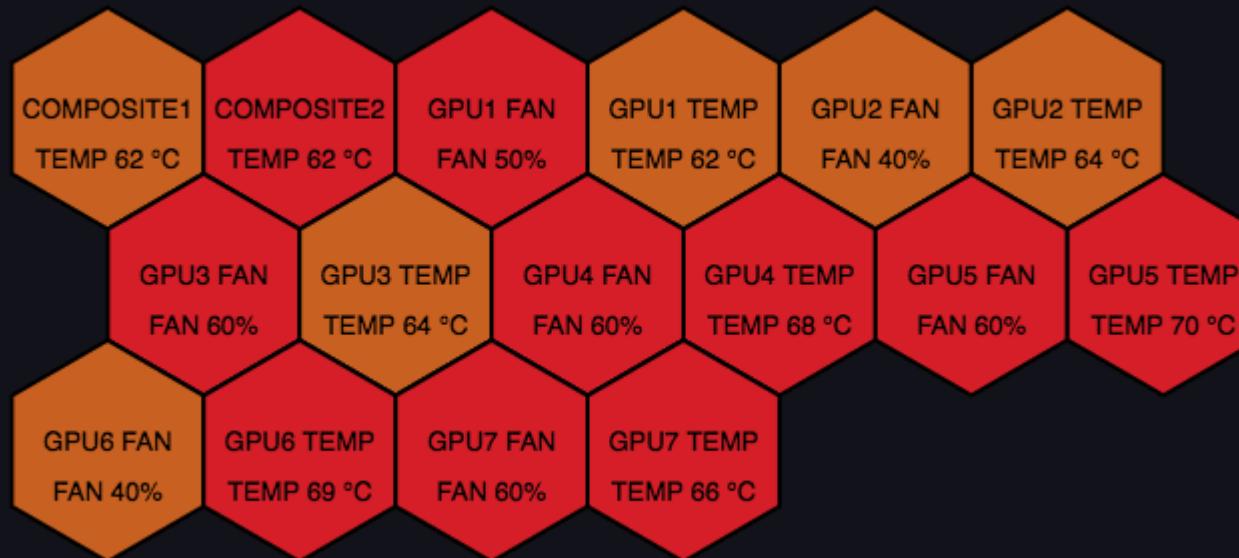
# Weather - Darksky Datasource



# *New Plugins From Grafana Labs*

- Azure Data Explorer Datasource
- Polystat Panel
- Flux (InfluxDB) Datasource
- Sensu App

# *Polystat Panel*



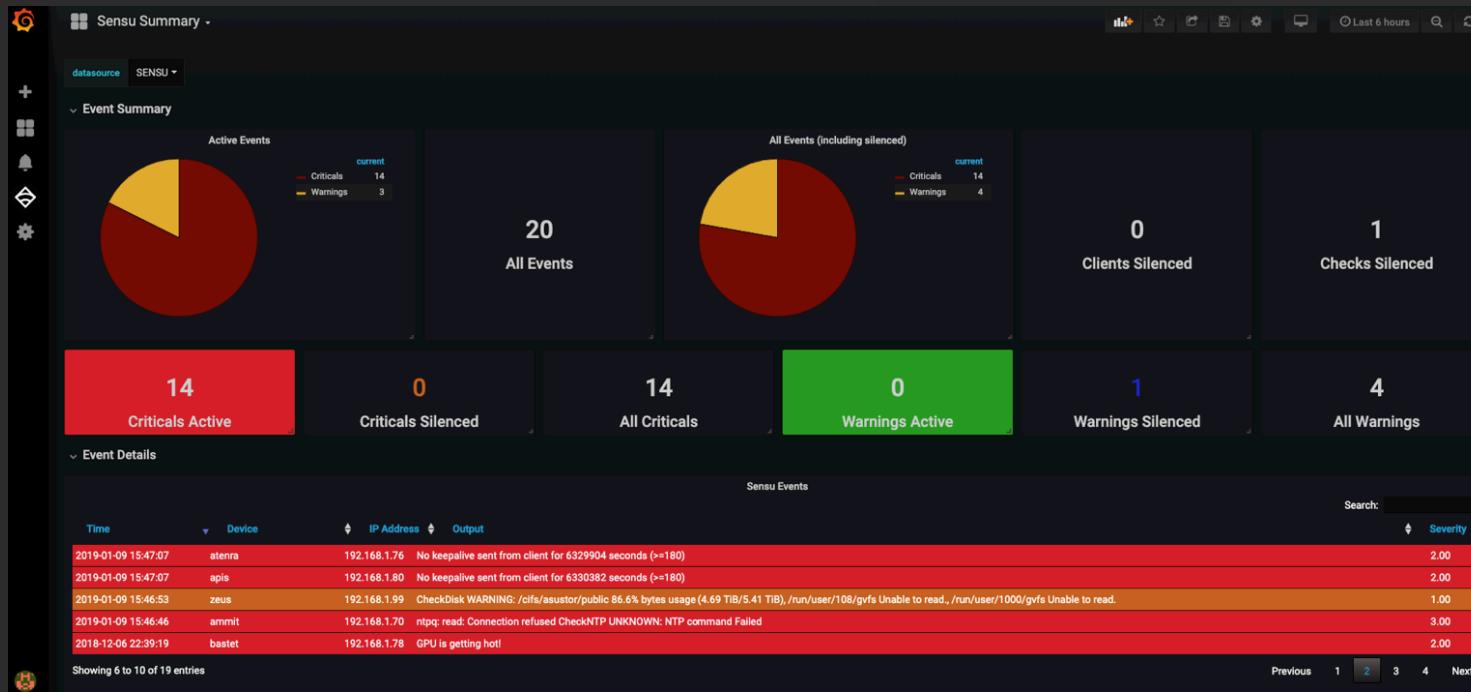
# Flux Datasource

The screenshot shows the Grafana interface with a dark theme. On the left, there is a sidebar with four icons: a yellow circle with an orange arrow pointing right, a blue line graph, a gear, and a bell. Above the sidebar, a dropdown menu says "Queries to" followed by "Flux". Below the sidebar, a section titled "A" contains a Flux query:

```
from(bucket: "telegraf")
|> range($range)
|> limit(n: 100)
|> filter(fn: (r) => r["_measurement"] == "mem")
```

At the bottom of the screen, there are several buttons: "FORMAT AS" (selected), "Time series", "Result tables", "20", "Result records" (selected), and "580".

# Sensu App



# *Improvement areas for Plugin Development*

- Plugins have been a huge success but...
- Datasources can't use Grafana alerting
- Lack of reusable components
- Lack of TypeScript typings
- No rules or guidance on building plugins
- Grafana is being converted to React, what happens with plugins?

# *The Future of Plugins - Backend Plugins*

- Hooks and extension points are incredibly useful
- Currently all plugins are frontend only
- New backend plugins
  - Backend component for datasource plugins
  - Alert Notifiers
  - Pluggable alerting engine

# *Our not-so-secret secret*

- Backend plugins for datasources is already in Grafana
  - but not quite ready
  - Documentation needs to be written
  - Continuous Integration for binaries (Linux, Darwin and Windows)
  - Some plugins are already using this feature:
    - Oracle OCI Datasource
    - Consul Datasource
    - AWS Cloudwatch Logging Datasource

# *The Future of Plugins - React and @grafana/ui*

- New UI component library being created while migrating Grafana to React
- Used for building Grafana and for React plugins
- Published to npm as an alpha last week

# *Build your own React plugin - Gauge Component*

```
import React, { PureComponent } from 'react';
import { Gauge, PanelProps } from '@grafana/ui';

interface Options {
  serverName: string;
}

interface Props extends PanelProps<Options> {}

export class Panel extends PureComponent<Props> {
  render() {
    const { options, width, height } = this.props;
    return (
      <div>
        <h2>Server {options.serverName}</h2>
        <Gauge
          value={10}
          width={width}
          height={height}
          minValue={0}
          maxValue={100}
        />
      </div>
    );
  }
}
```



# *Work with us on plugins*

- Plugins are a huge part of Grafana's future
- We want to build a larger plugins community
- Want to improve existing plugins (including core plugins)
- (And we're hiring - we are remote-first and focused on open-source)

# *Summary*

- All major cloud providers have support in Grafana
- Loads of great plugins to help you integrate with Grafana and visualize your data
- React plugins and the `@grafana/ui` library will make creating plugins easier
- New hooks and plugin types coming for the Grafana backend

# *Workshops Tomorrow*

- Extending Grafana
- Torkel and Peter will demonstrate the new @grafana/ui library