Deploying an Observability Stack in One Line of Code*

Anthony E. Nocentino

anocentino@purestorage.com

Anthony E. Nocentino

Principal Field Solution Architect @ Pure Storage

- Specialize in system architecture, performance, SQL Server, Kubernetes, Containers, Microsoft Azure and VMware
- Masters Computer Science

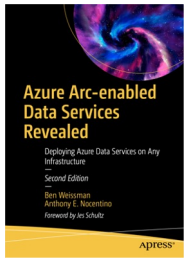
email: anocentino@purestorage.com

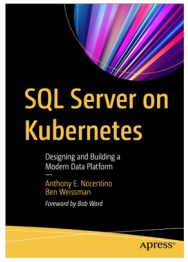
Blog: www.nocentino.com

Twitter: @nocentino

GitHub: https://github.com/nocentino/

Pluralsight Author: www.pluralsight.com





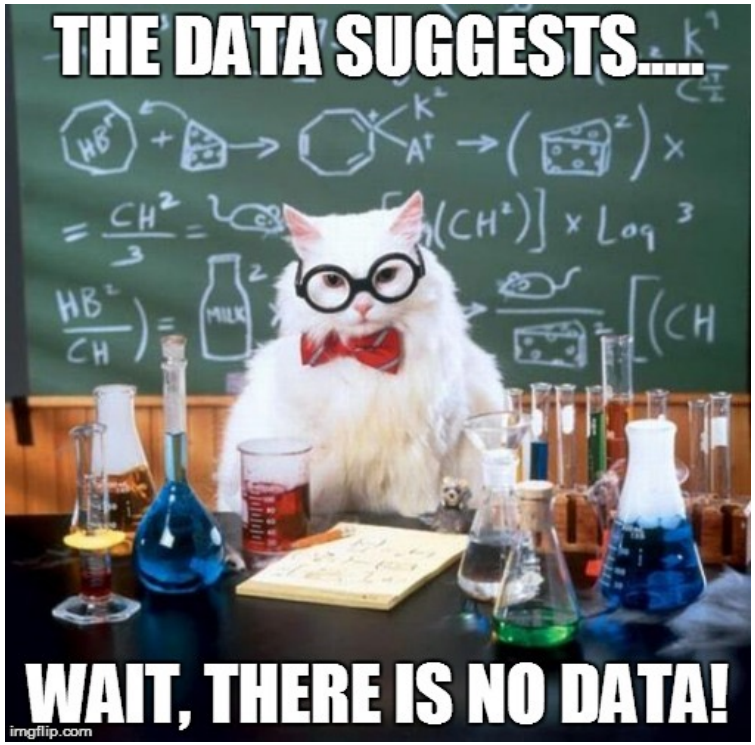
Agenda

- What's observability?
- Defining meaningful metrics and dashboards
- Architecture of our monitoring solution
- Deploying in docker
- Demos

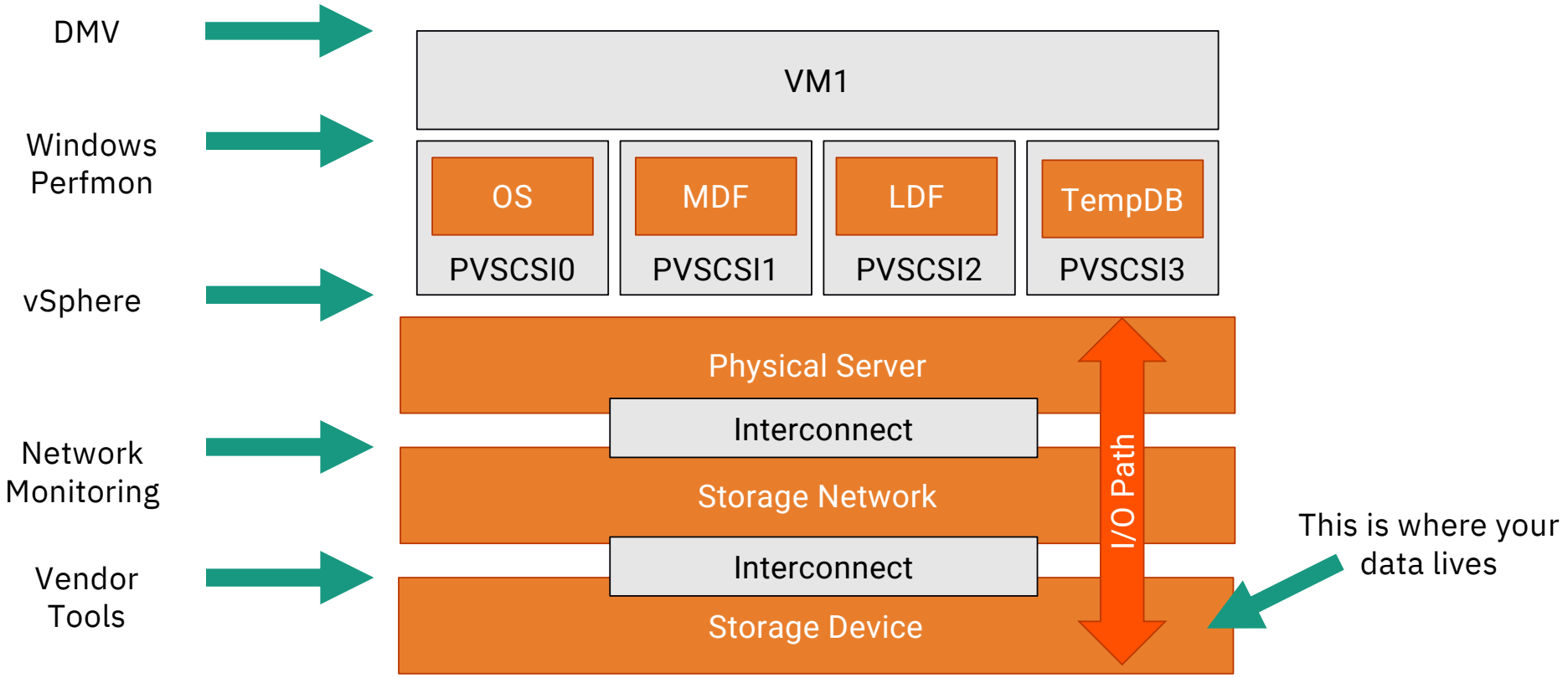
Why Did I Build This?

Customers often have no monitoring platforms ☹

If they do, they're often point solutions that only give visibility to parts of their stack

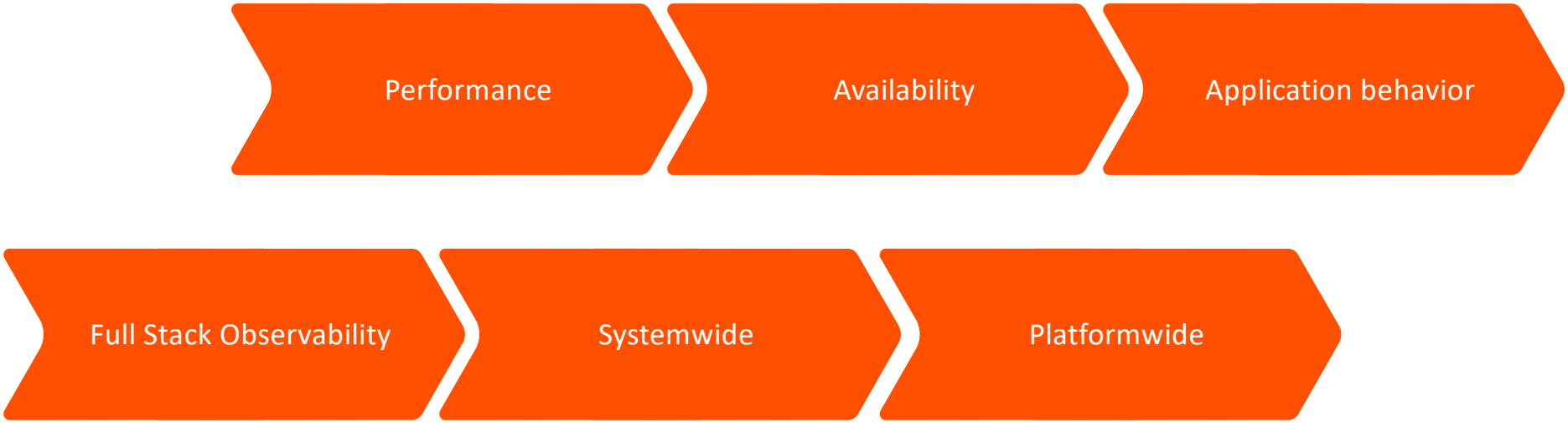


Virtualization IO Stack – SQL Server Example

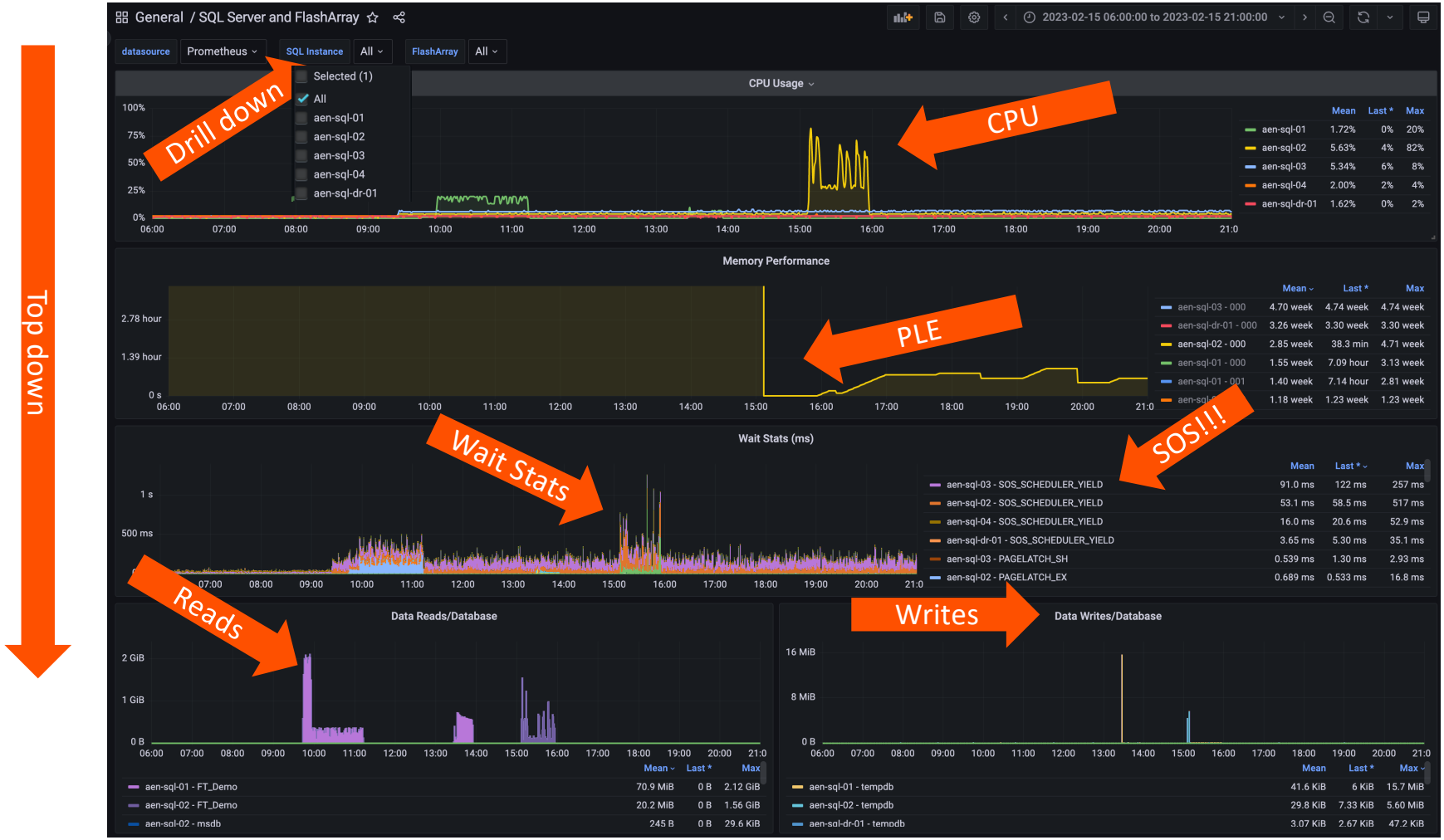


What’s Observability?

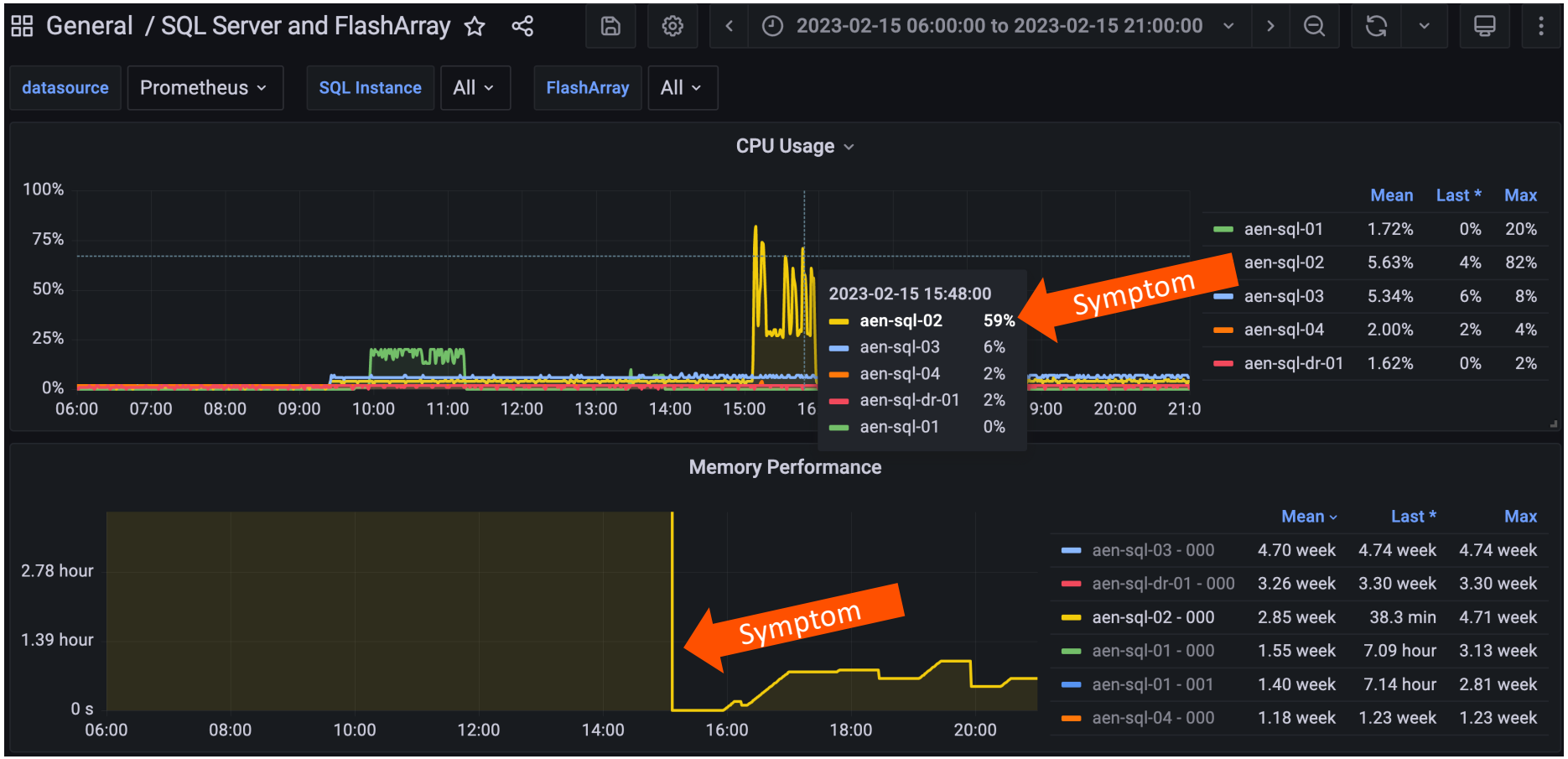
Understanding what’s happening and how it impacts the health of the system



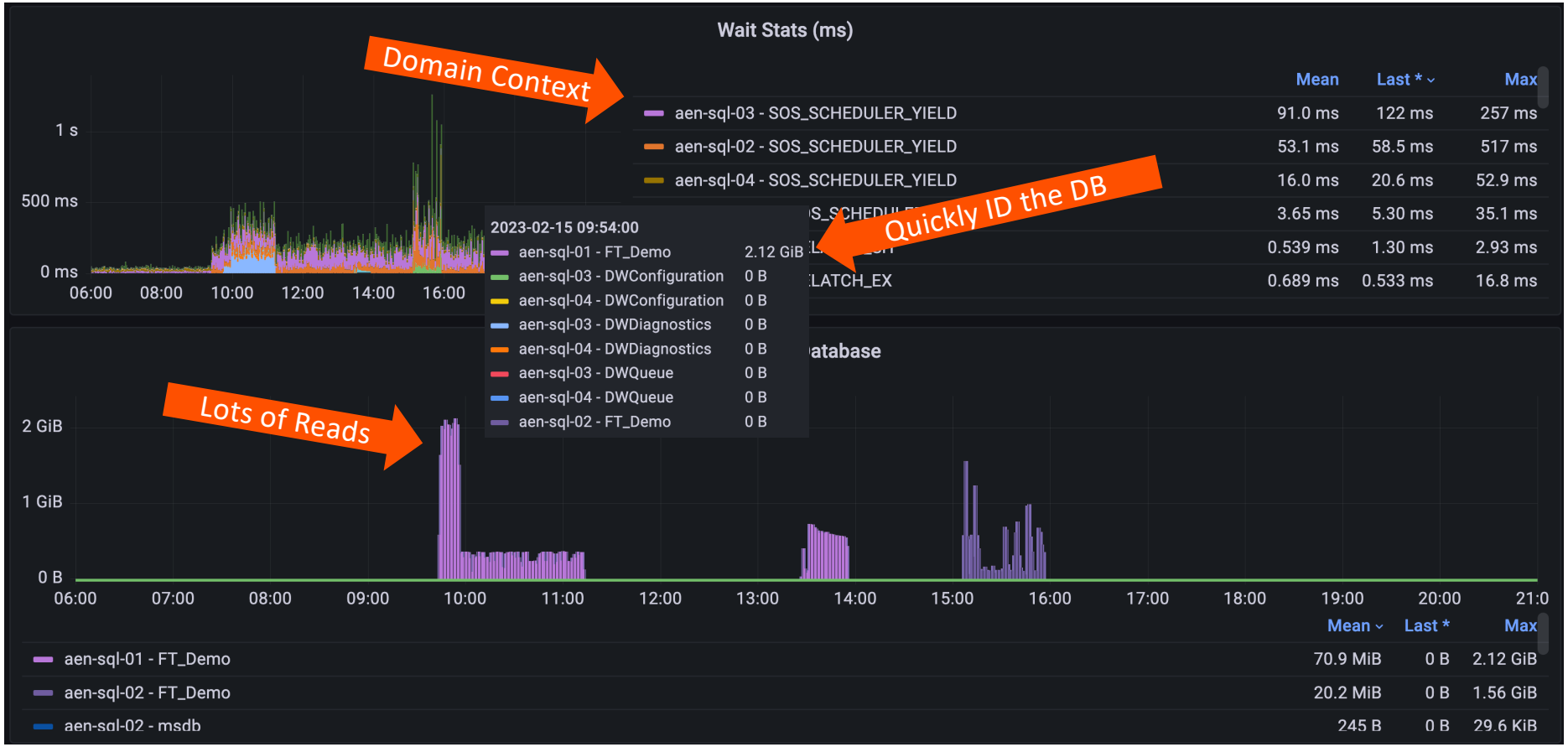
Defining Meaningful Dashboards – Convey what's important



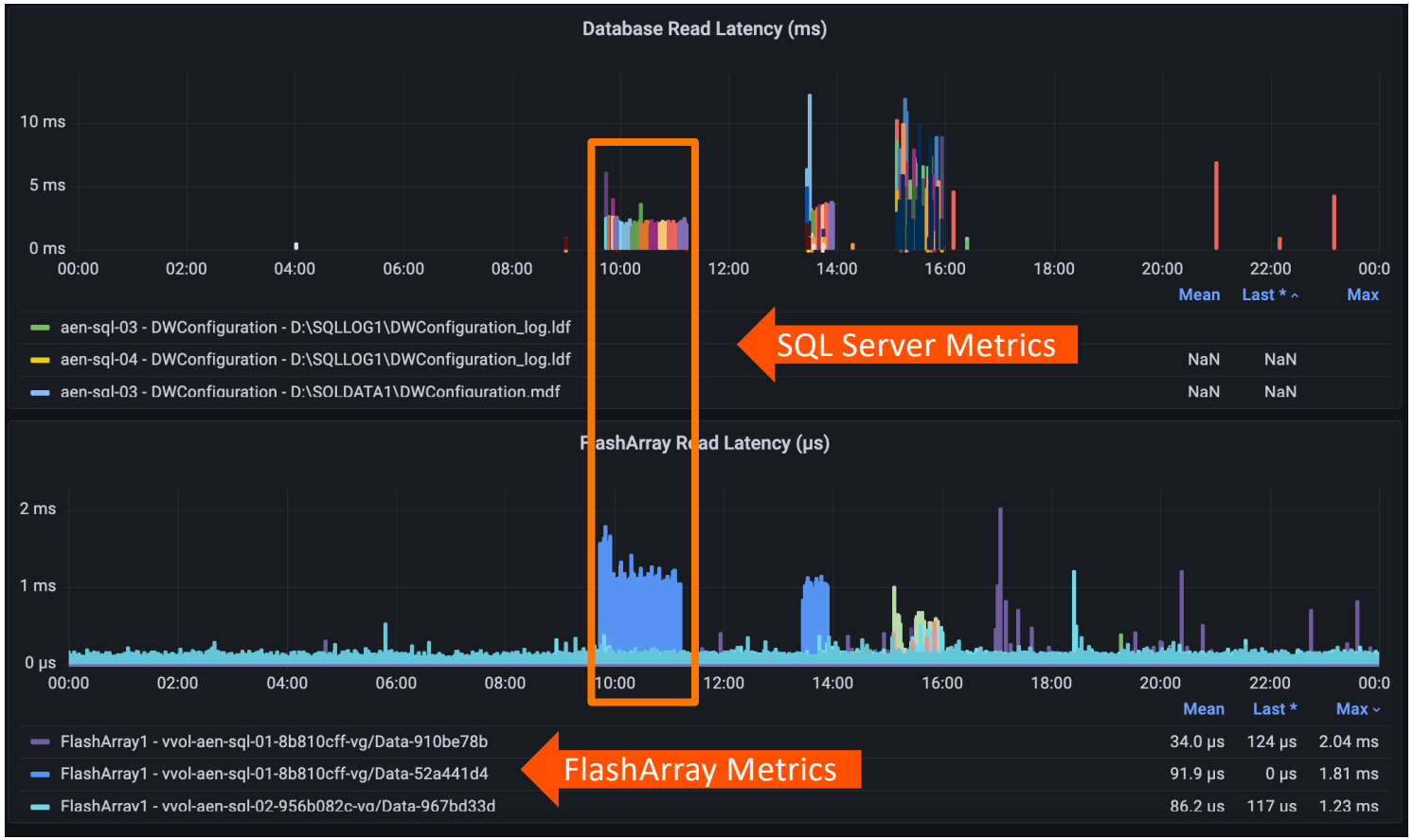
Defining Meaningful Dashboards



Defining Meaningful Dashboards



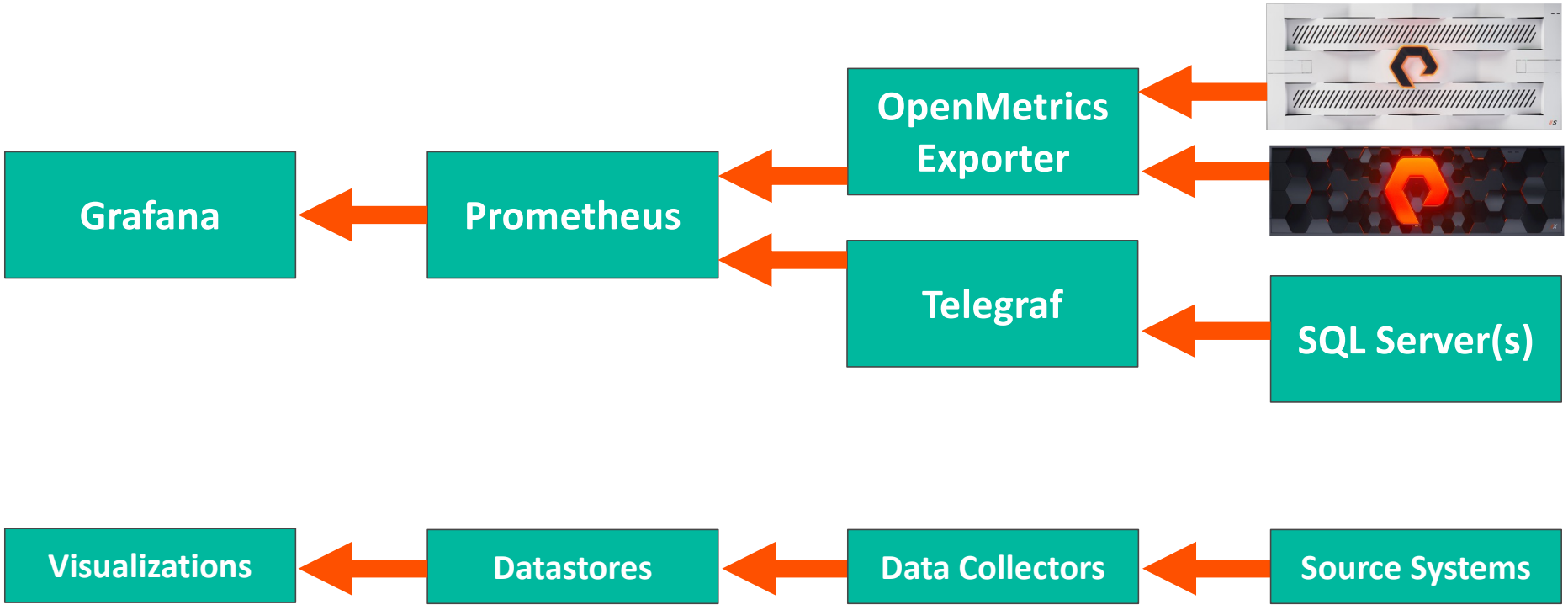
Defining Meaningful Dashboards – correlating metrics



Gain insight across platforms and domains

Monitoring Architecture

Keeping an eye on complex systems



Other Telegraf Plugins

Windows Server 2022

MySQL

vmware

SQL

https://github.com/influxdata/telegraf

Building your own Metrics Exporter

Defined design pattern to build your own

Also specifies an interface for the metric output

Gets metrics from the target via the network

Generally, HTTP/HTTPS

Another option is instrumentation in your applications

What's better a **push** or a **pull**?

https://prometheus.io/docs/guides/multi-target-exporter/

Container Based Deployment

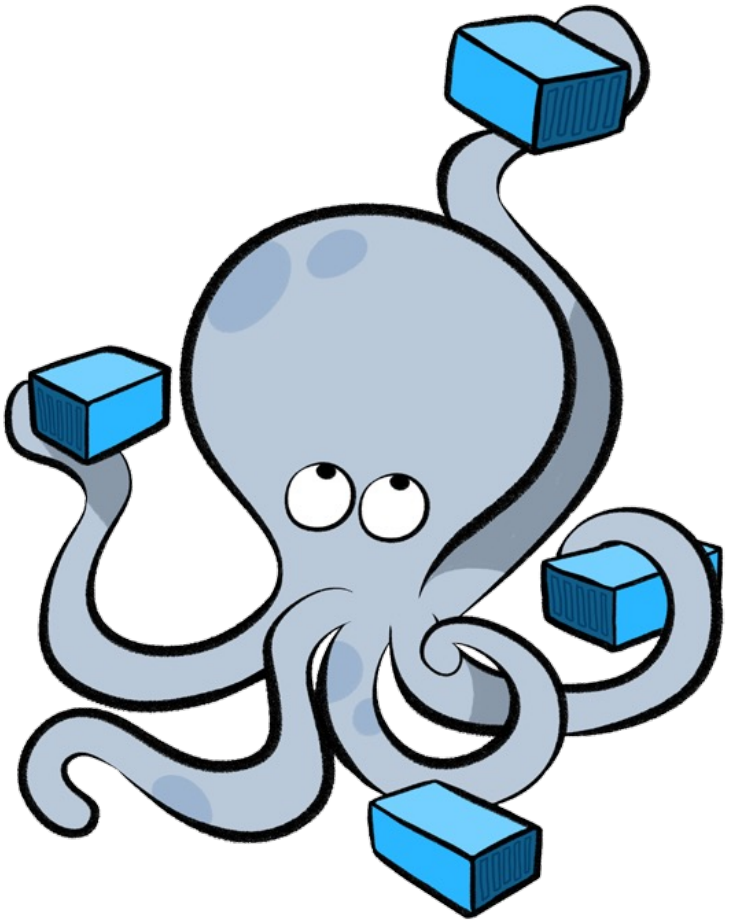
Container is a self-contained application

Docker Compose

- Starts up the containers
- Configures the applications
- Networking connecting the applications
- Expose Grafana to the network

Orchestrated solution defined in code

Can run anywhere you have Docker



System Requirements

Its just a couple containers, right?

- Docker (which includes Docker Compose)
- Can run on Linux, Windows or MacOS
- If you're building a persistent system, Linux is suggested
- A SQL Server to monitor
- Since it's a time series database we can build time correlated dashboards from different sources
  - FlashArray Volume Name – OpenMetrics Exporter
  - SQL Server Instance Name – Telegraf
- We're correlating the metrics based on the FA Volume Name containing the SQL Server instance name
  - vVols/RDM/Physical

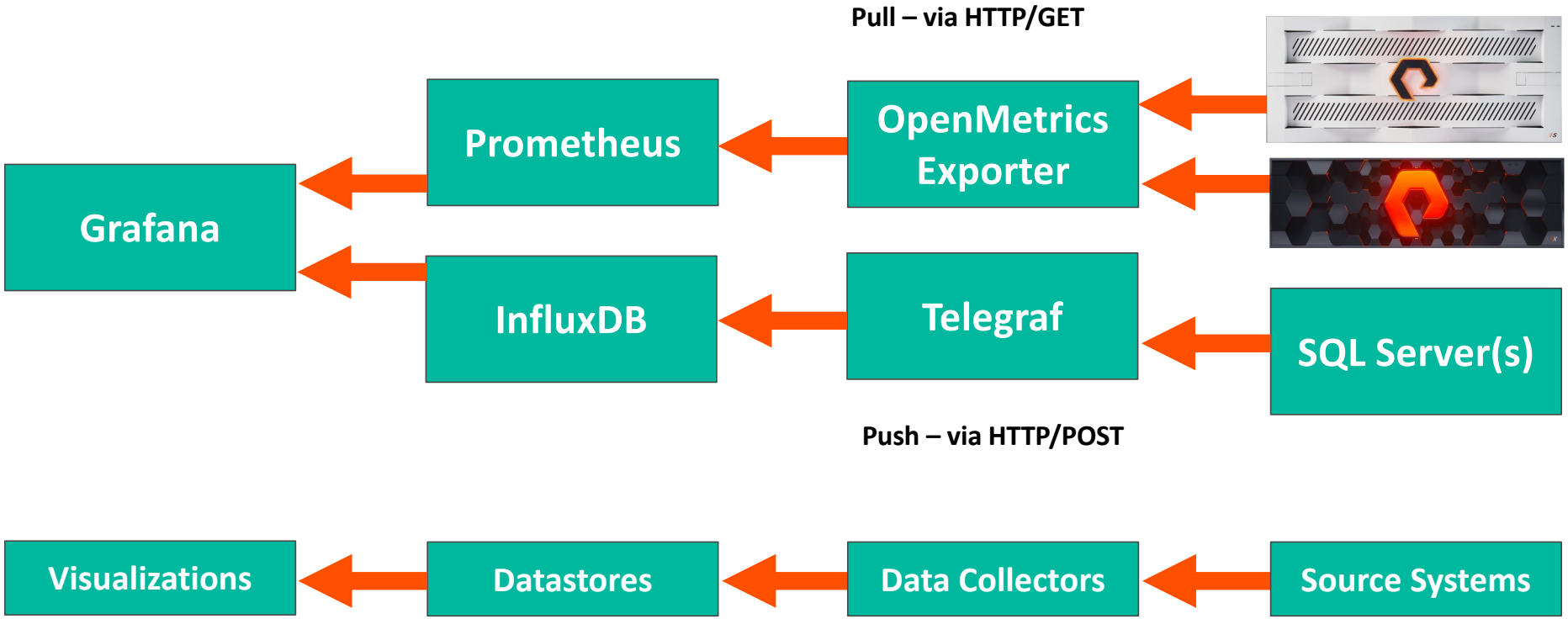
https://github.com/nocentino/pure-fa-openmetrics-exporter-sqlserver

PURESTORAGE®

Demo:
Deploying a monitoring stack in one line of code*

Monitoring Architecture – What's next

Keeping an eye on complex systems



Key Takeaways

Deploying a monitoring stack in one line of code*

- Try it out – https://github.com/nocentino/pure-fa-openmetrics-exporter-sqlserver
- https://www.nocentino.com/posts/2022-12-20-monitoring-flasharray-with-openmetrics/
- Will likely separate out the Telegraf metrics into an InfluxDB instance
- Add in VMware metrics