
Grafana Labs

Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share beautiful dashboards with your team and foster a data driven culture.

The easiest way to get started is with Grafana Cloud, our fully composable observability stack.

Grafana's Background

Created in 2014

Thriving community & 4M+ users

Many data connections, unified

Focused on producing excellent dashboards

Free, open-source technology

Free & paid deployment options

Single dashboarding service

Full "observability stack"

“The goal when creating Grafana was to make the things I found hard, and other people found hard, more easy and accessible. That way, more people could actually start instrumenting their applications and creating dashboards by themselves. Make the tools of observability accessible to everyone in an organization, not just the single Ops person.”

Torkel Ödegaard
CGO, Co-founder, Grafana Labs

The evolution of Grafana

Since it was started in 2014, Grafana has become one of the most popular open source projects on GitHub.

6.0

- Explore for ad-hoc data exploration and troubleshooting
- Gauge Panel
- New Panel Editor UX improvements
- Google Stackdriver Datasource released
- Azure Monitor plugin core datasource
- React plugin support
- Named colors for improved color picking
- Removal of user session storage

7.0

- New Panel Editor
- Explore new tracing UI
- Usage insights for enterprise customers
- Transformations and simple Math operations for all data sources
- New plugins platform
- Cloudwatch support
- Time zone support

8.0

- Unified alerts
- Library panels for improved dashboards
- Real-time streaming
- Fine-grained access control
- Data source query caching
- New panels for bar charts, state timeline, histogram
- Plugin catalog

9.0

- Prometheus and Loki visual query builders
- New heatmap, geomap, candlestick panels
- New Explore-to-dashboard workflow
- Command palette navigation
- Envelope encryption
- New panel search
- Alerting UX improvements

Community-driven development is at the heart of Grafana

Built on open source

- 900k+ active installs
- 49k GitHub stars

Driven by the community

- 10M+ global users
- 1,700+ contributors

https://community.grafana.com/

https://slack.grafana.com/

https://grafana.com/docs/grafana/latest/developers/contribute/?pg=oss-graf&plcmt=resources

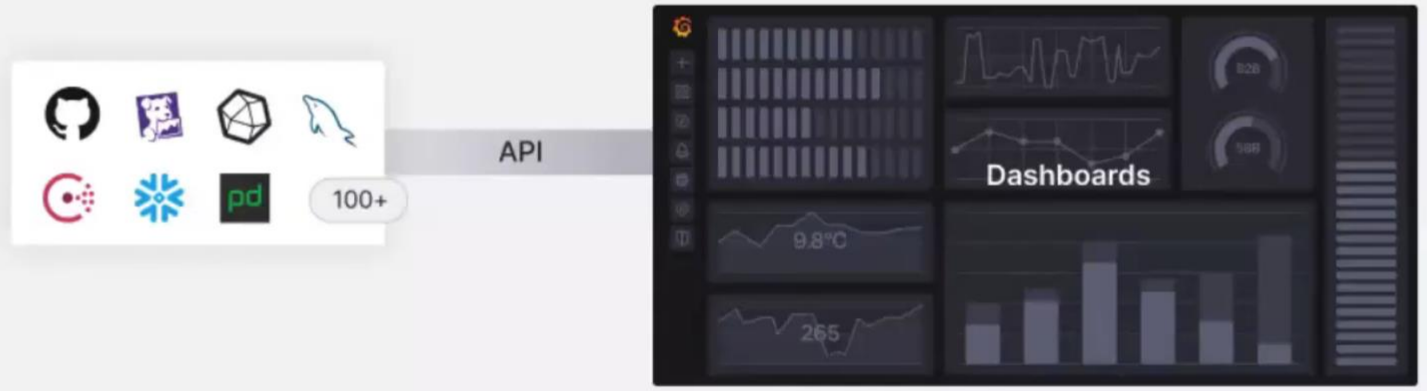
Grafana Labs Blog

News, announcements, articles, metrics & monitoring love

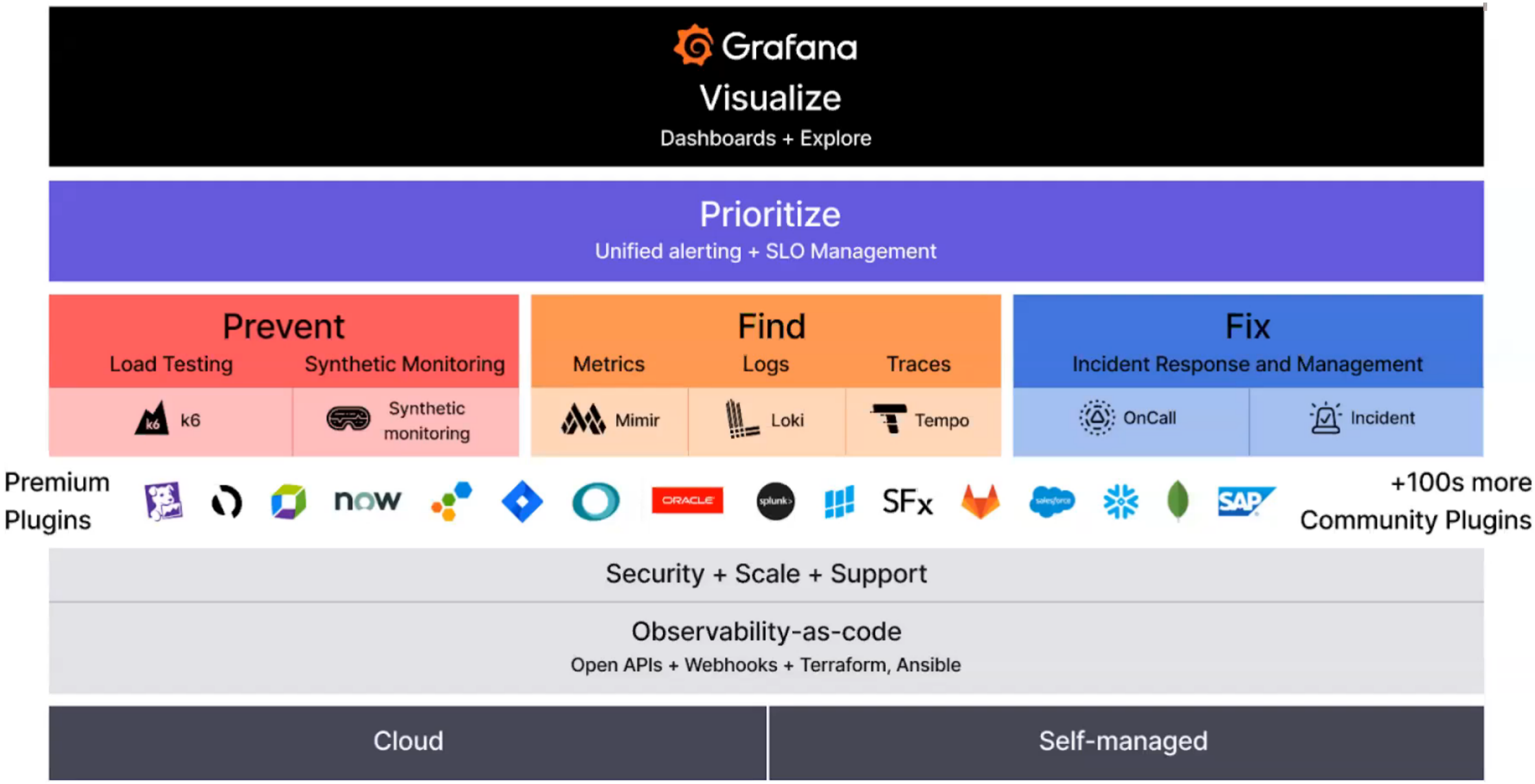
All articles →

https://grafana.com/blog/

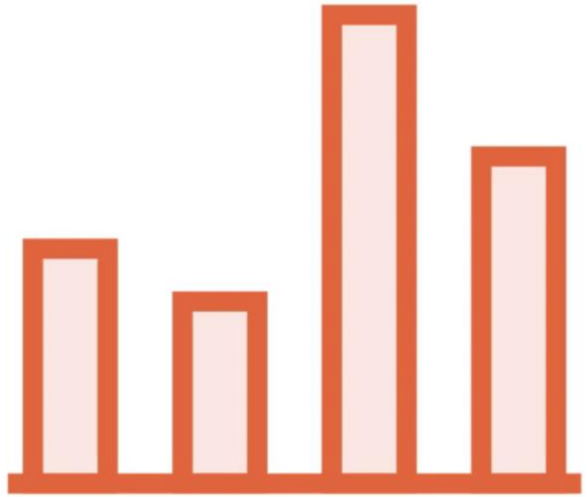
What most users think of Grafana



Your open and composable observability Stack



Grafana's Capabilities



Build rich, interactive dashboards

Built-in graphics for different data

Highly customizable

Secure web application

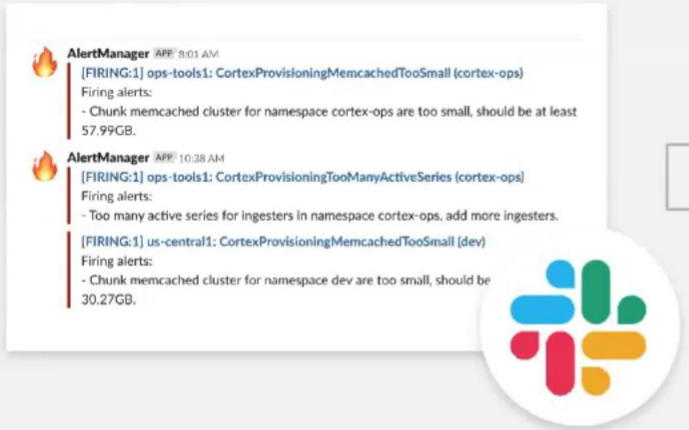
- **Set permissions**
- **Third party authentication**

Repeating panels

User selected variables

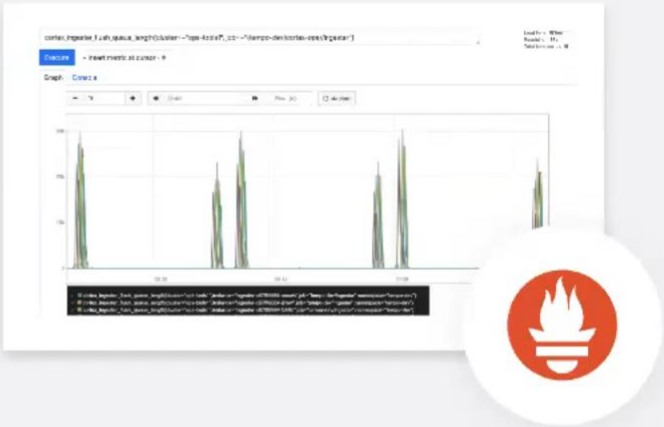
Linking to other dashboards

1. Alert



2. Dashboard

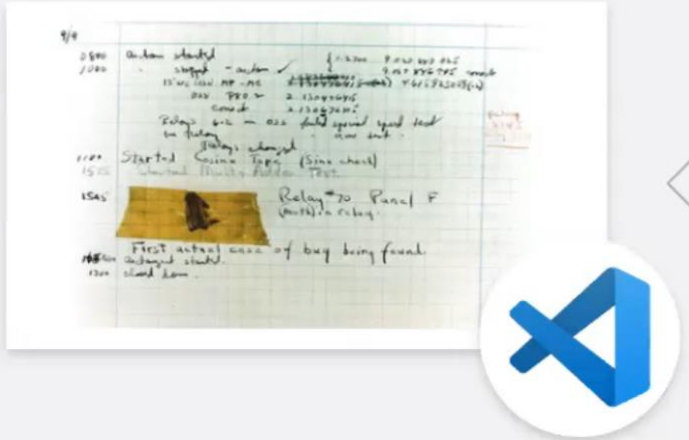
3. Adhoc Query



4. Log Aggregation

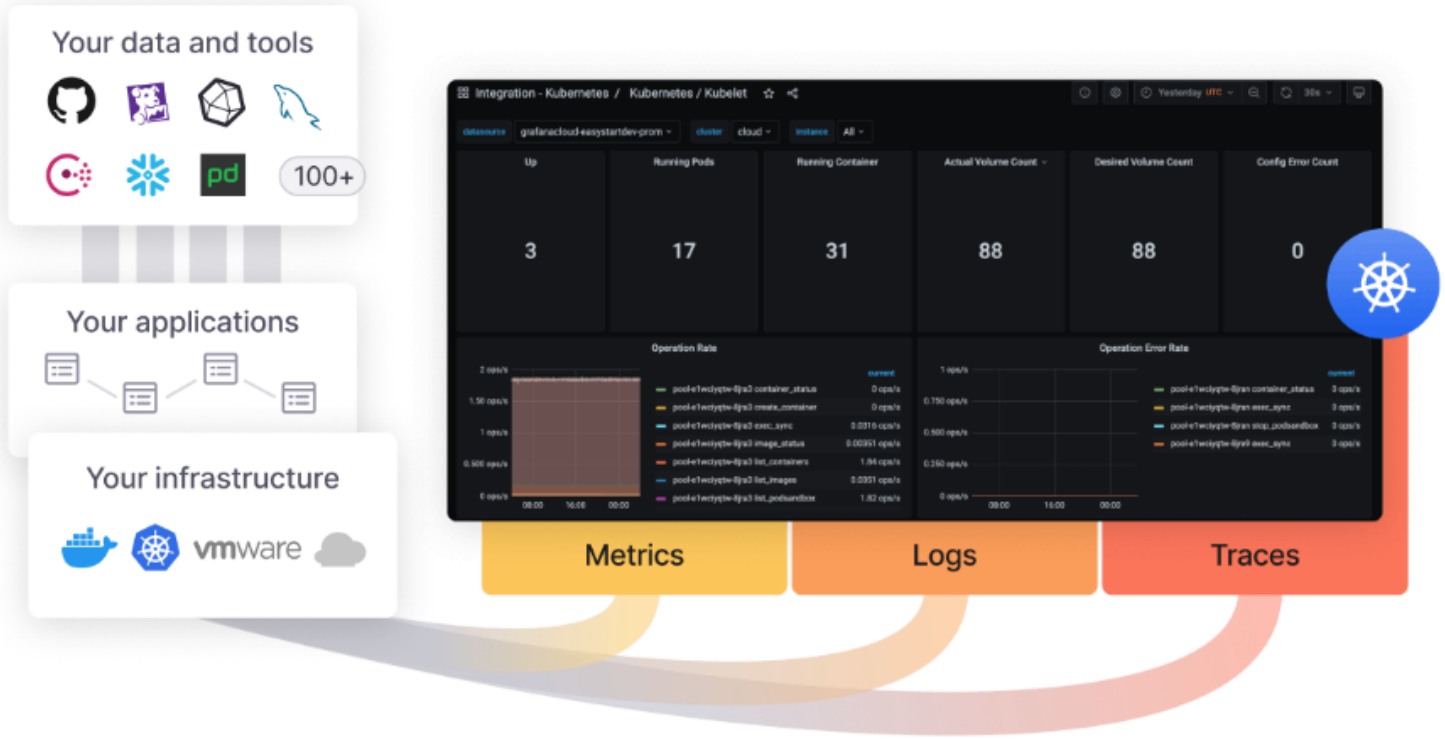
5. Distributed Tracing

6. Fix!



Your K8s monitoring stack

Operational dashboards for your data here, there, or anywhere



Your visibility stack

Operational dashboards for your data here, there, or anywhere

Your IoT visibility stack

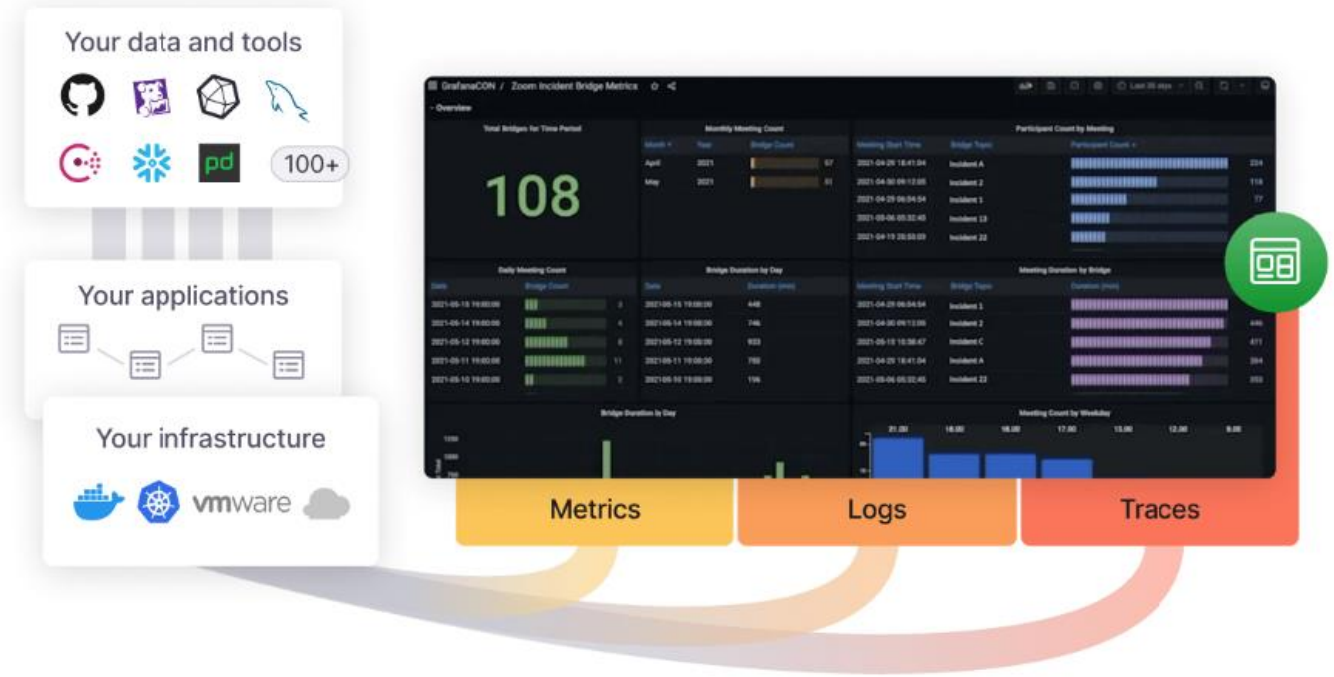
Operational dashboards for your data here, there, or anywhere

Your whatever you want to see stack

Operational dashboards for your data here, there, or anywhere

Your single pane of glass stack

Operational dashboards for your data here, there, or anywhere

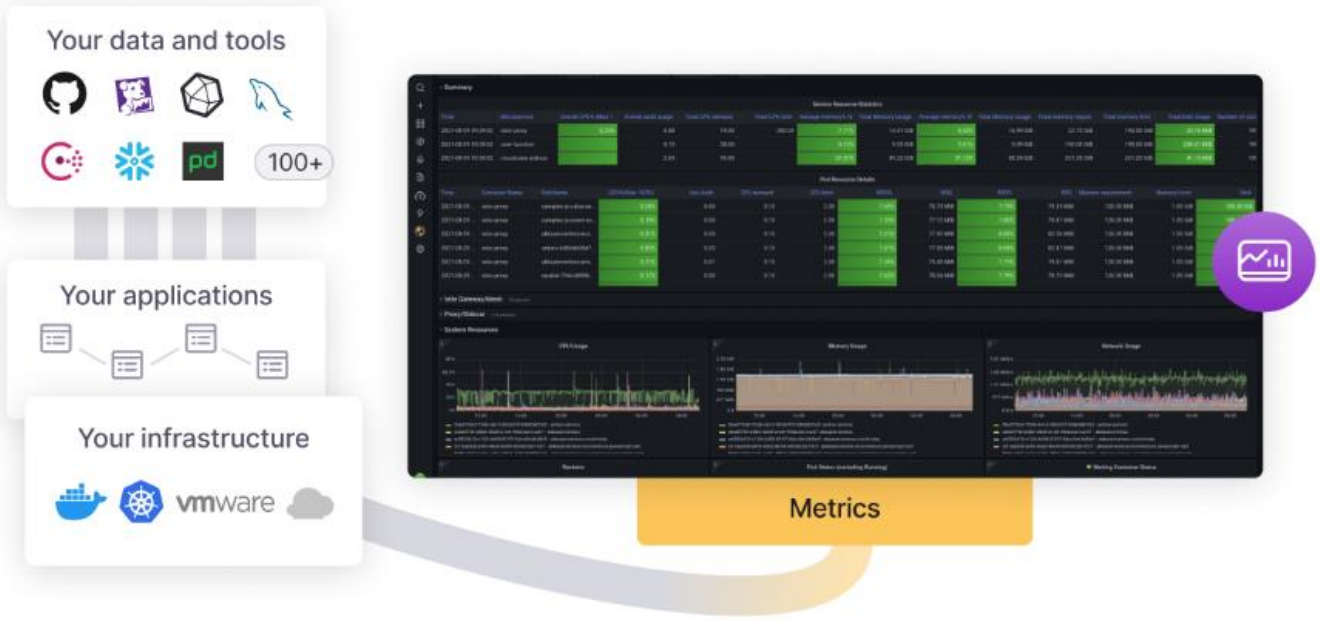


Your observability stack

Operational dashboards for your data here, there, or anywhere

Your monitoring stack

Operational dashboards for your data here, there, or anywhere



Grafana Labs as a company

1,000+ employees (50% total R&D)

950K+ active Grafana installations

2,000+ Grafana Labs customers

6M+ Grafana Labs users

$534M in VC funding raised since 2019

Open Source is at the heart of what we do at Grafana Labs:

Employ **44%** of all Prometheus Project Maintainers

Employ **100%** of Grafana Maintainers & Project Founders

Employ **100%** of the Project Founders for Loki

Employ **100%** of all Mimir Maintainers & Project Founders

The **leading contributors** to the Graphite project

Employ **100%** of the Project Founders for Tempo

Grafana Open Source

Grafana open source is open source visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics, logs, and traces no matter where they are stored. It provides you with tools to turn your time-series database (TSDB) data into insightful graphs and visualizations.

Grafana Loki

- Grafana Loki is an open source, set of components that can be composed into a fully featured logging stack.

Grafana Tempo

- Grafana Tempo is an open source, easy-to-use and high-volume distributed tracing backend.

Grafana Cloud

- Grafana Cloud is a highly available, fast, fully managed OpenSaaS logging and metrics platform. It is everything you love about Grafana, but Grafana Labs hosts it for you and handles all the headaches.

Grafana Enterprise

- Grafana Enterprise is the commercial edition of Grafana that includes additional features not found in the open source version. Building on everything you already know and love about Grafana, Grafana Enterprise adds enterprise data sources, advanced authentication options, more permission controls, 24x7x365 support, and training from the core Grafana team.

Grafana Loki

Loki is a horizontally scalable, highly available, multi-tenant log aggregation system inspired by Prometheus. It is designed to be very cost effective and easy to operate. It does not index the contents of the logs, but rather a set of labels for each log stream.

The Loki project was started at Grafana Labs in 2018, and announced at KubeCon Seattle. Loki is released under the AGPLv3 license.

Grafana Labs is proud to lead the development of the Loki project, building first-class support for Loki into Grafana, and ensuring Grafana Labs customers receive Loki support and features they need.

Why use Grafana Loki?

It's really easy to get started because you can send logs in any format, from any source, using a wide array of clients

100% persistence to object storage means you get petabyte scale, high throughput and cost-effective & durable storage

Build metrics and generate alerts from your log lines

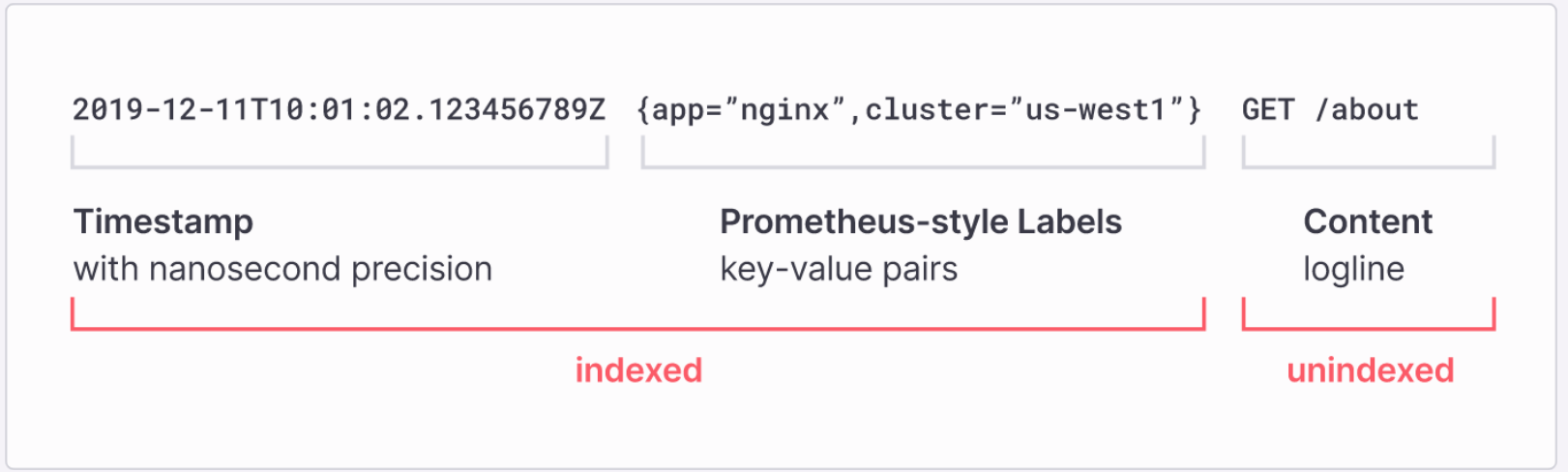
No ingestion log formatting requirements gives you more flexibility and the option to format at query time

Tail your logs in realtime to see the logs as they come into the system, update the logs after every certain time, view logs for a particular date, etc.

Natively integrates with Prometheus, Grafana and K8s so you can seamlessly move between metrics, logs and traces within a single UI

Grafana Loki

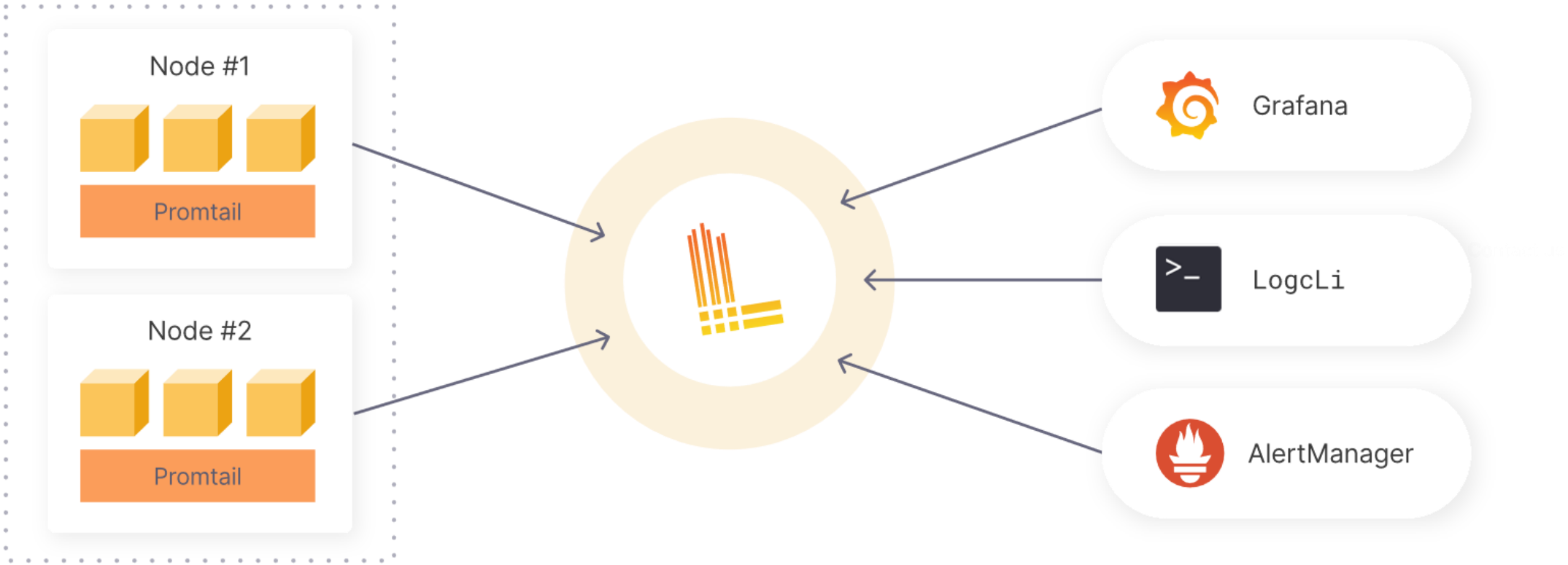
Loki takes a unique approach by **only indexing the metadata** rather than the full text of the log lines:



Loki's minimal indexing approach means that storing the same set of logs in Loki requires far less storage than with other solutions

- ✓ Log any and all formats
- ✓ Fast writes
- ✓ Tiny indexes
- ✓ Cheaper to run
- ✓ Simpler to operate
- ✓ Fast queries
- ✓ Cut and slice your logs in dynamic ways (Flexible)

Grafana Loki



Grafana Loki

Pull in any logs with Promtail
Promtail is a logs collector built specifically for Loki. It uses the same service discovery as Prometheus and includes analogous features for labeling, transforming, and filtering logs before ingestion into Loki.

Store the logs in Loki
Loki does not index the text of logs. Instead, entries are grouped into streams and indexed with labels.Not only does this reduce costs, it also means log lines are available to query within milliseconds of being received by Loki.

Use LogQL to explore
Use Loki's powerful query language, LogQL, to explore your logs. Run LogQL queries directly within Grafana to visualize your logs alongside other data sources, or with LogCLI, for those who prefer a command line experience.

Alert on your logs
Set up alerting rules for Loki to evaluate on your incoming log data. Configure Loki to send the resulting alerts to a Prometheus Alertmanager so they can then get routed to the right team.

Built on open source, driven by the community

66k+
Active users

12k+
GitHub stars

400+
Contributors

Grafana Tempo

Grafana Tempo is an open source, easy-to-use, and high-scale distributed tracing backend. Tempo is cost-efficient, requiring only object storage to operate, and is deeply integrated with Grafana, Prometheus, and Loki. Tempo can ingest common open source tracing protocols, including Jaeger, Zipkin, and OpenTelemetry.

The Tempo project was started at Grafana Labs and announced at Grafana ObservabilityCON in October 2020. It became generally available with the 1.0 release in June 2021. Tempo is released under the AGPLv3 license.

Grafana Labs is proud to lead the development of the Tempo project, building first-class support for Tempo into Grafana, and ensuring Grafana Labs customers receive Tempo support and features they need.

The Goal 🚀	Sample 100% of our read path
The Problem 😢	Scaling existing solutions
The Solution 💡	Tempo

Grafana Tempo

Traditional distributed tracing tools

- ⚠ **Challenging to scale** Requires a database like Elasticsearch or Cassandra that can be tough to manage and costly to operate at scale
- ⚠ **Expensive** Paying to index and store traces
- ⚠ **Requires manual instrumentation** Existing tools need to be manually configured

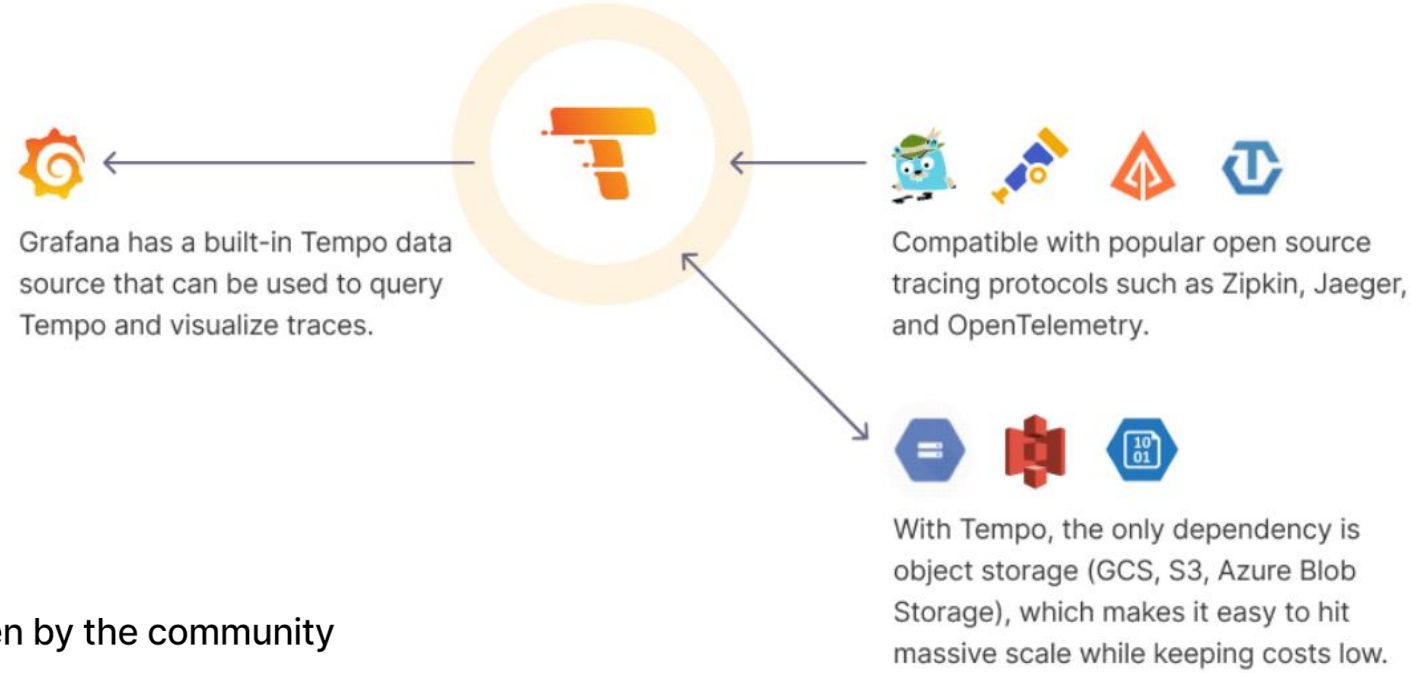
→

Tempo's approach

- ✓ **Built for massing scale** The only dependency is object storage which provides affordable long term storage of traces
- ✓ **Cost-effective** Not indexing the traces makes it possible to store orders of magnitude more trace data for the same cost
- ✓ **Strong integration with open source tools** Compatible with open source tracing protocols

Grafana Tempo

How does Grafana Tempo work?



Built on open source, driven by the community

2k+
Active users

1k+
GitHub stars

70+
Contributors

Grafana Enterprise Stack

Compose and scale observability with one or all pieces of the stack – on your infrastructure.

Request a demo →

Scalability guaranteed
Purpose-built to handle the scale of today's data volumes and tomorrow's growth

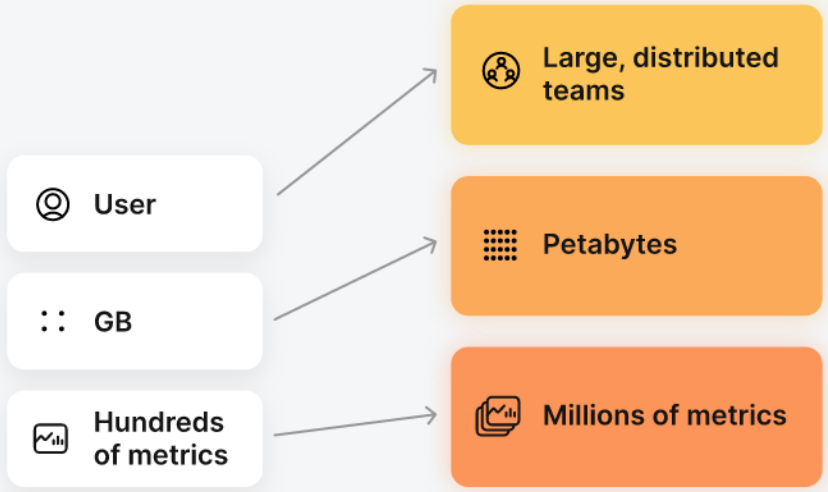
Security & authentication
Provides secure access and better data protection

Support
Backed by support, training, and consulting from the Grafana Labs team

Correlated data
Integrates metrics and logs with Grafana

Architected for scale

Built with modern distributed systems principles to grow with your applications and infrastructure. No artificial limits or barriers, just true horizontal scalability.

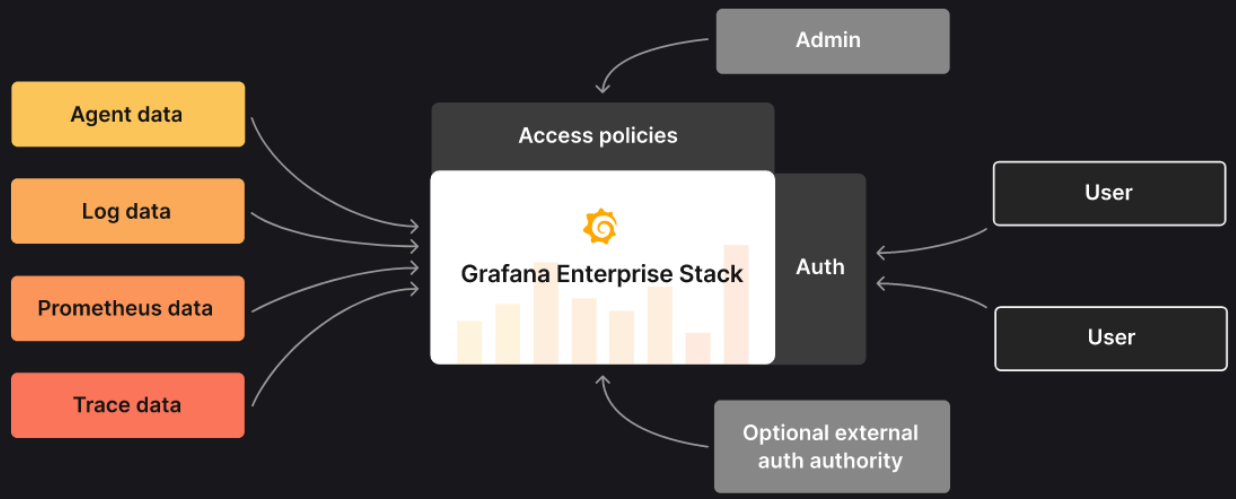


Architected for scale

- Achieve maximum potential observability since metric and log limits are no longer a barrier.
- Save money by not supporting the overhead of a monitoring infrastructure, both in resources and people time.
- Easily store application and infrastructure data in one centralized cluster, or across multiple clusters without needing a dedicated team.

Simple, organization-wide administration and security

Confidently democratize data across your entire company with advanced admin tools to govern, audit, and secure user permissions and data.



Simple to manage

Not everyone needs access to the same data. Create flexible, secure views of relevant data for each team to optimize self-service, without additional overhead.

Simple to synchronize

Centralized authentication allows you to seamlessly integrate with your existing LDAP/Active Directory platform to allow a single point of control.

Simple to customize

Best-in-class query performance means you can quickly create real-time dashboards that can be shared throughout your organization.

Simple to secure

Export detailed user event logs to a file or to Enterprise Logs, in order to track and prevent suspicious activity.

Integrations

In addition to the plugins, Grafana Cloud customers get full support for integrations with dozens of popular 3rd party providers, tools, and services.

Supported Integrations

- Apache Spark
- Apollo Server
- Asterisk
- AWX
- Caddy
- Ceph
- cert-manager
- CloudWatch
- CloudWatch logs
- CockroachDB
- Confluent Cloud
- Consul
- CoreDNS
- Dnsmasq
- Docker
- Elasticsearch
- Etcd
- Gitea
- GitHub
- Go programming language
- HAProxy
- Home Assistant
- Java Virtual Machine
- Jenkins
- JIRA
- Kafka
- Kubernetes
- Linux Node
- MacOS
- Memcached
- MinIO
- MongoDB
- MySQL
- NGINX
- Node.js
- NSQ
- PostgreSQL
- RabbitMQ
- rclone
- Redis
- Ruby Rack
- Spring Boot
- Windows
- WSO2 Enterprise Integrator
- WSO2 Streaming Integrator

5 key positive consequences

Reduce MTTR

Improve business visibility

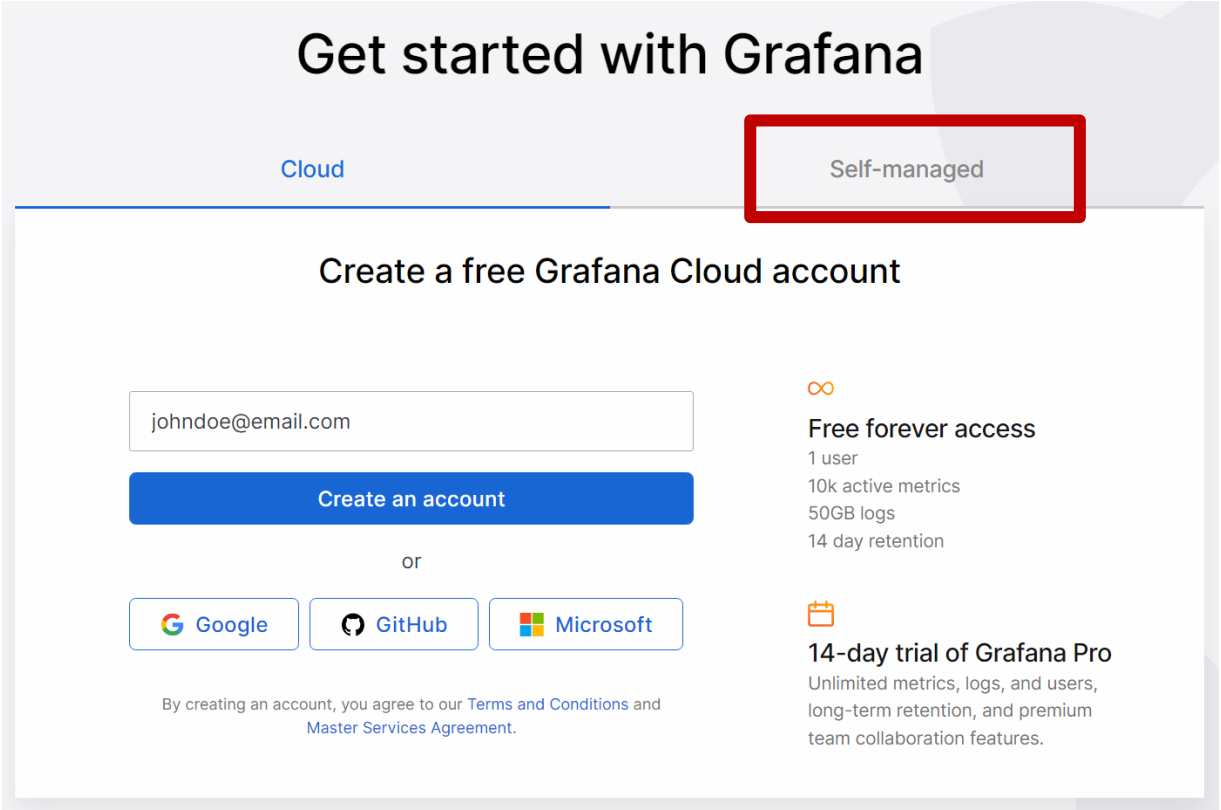
Enable self service with Governance

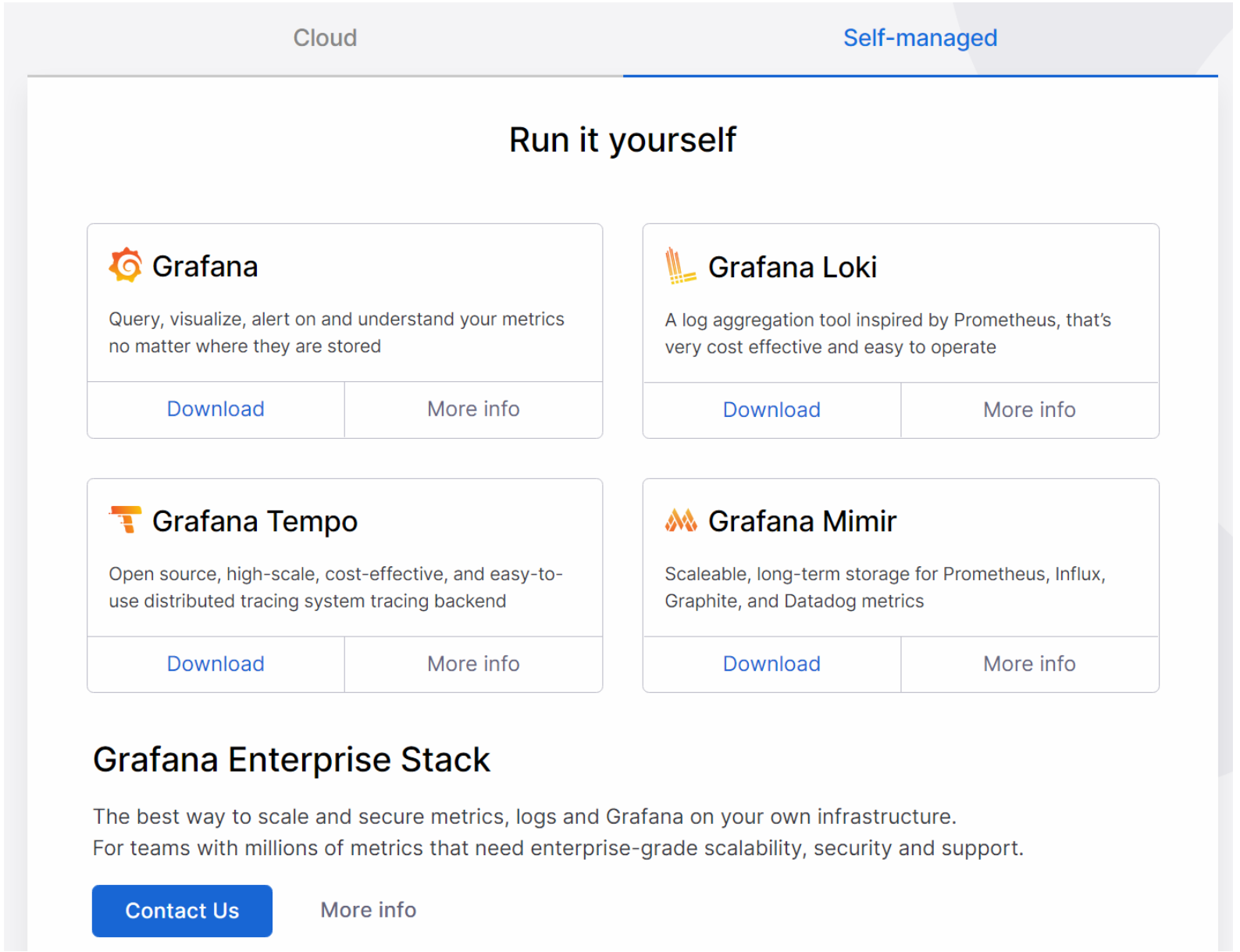
Meet SLA and Enforce best practices

Implement new use cases

Environment Setup

https://grafana.com/get/?plcmt=top-nav&cta=downloads





https://grafana.com/grafana/download?pg=get&plcmt=selfmanaged-box1-cta1

Configure Grafana:
https://grafana.com/docs/grafana/v9.0/setup-grafana/configure-grafana/

Upgrade Grafana:
https://grafana.com/docs/grafana/v9.0/setup-grafana/upgrade-grafana/

Configure Security:
https://grafana.com/docs/grafana/v9.0/setup-grafana/configure-security/

Sign in to Grafana

To sign in to Grafana for the first time:

1. Open your web browser and go to **http://localhost:3000/**. The default HTTP port that Grafana listens to is 3000 unless you have configured a different port.
2. On the signin page, enter admin for username and password.
3. Click Sign in. If successful, you will see a prompt to change the password.
4. Click OK on the prompt and change your password.

Grafana CLI Setup

Grafana CLI is a small executable that is bundled with Grafana server. It can be executed on the same machine Grafana server is running on. Grafana CLI has plugins and admin commands, as well as global options.

To list all commands and options:

grafana-cli -h

Invoking Grafana CLI

To invoke Grafana CLI, add the path to the grafana binaries in your `PATH` environment variable. Alternately, if your current directory is the `bin` directory, use `./grafana-cli` . Otherwise, you can specify full path to the CLI. For example, on Linux `/usr/share/grafana/bin/grafana-cli` and on Windows `C:\Program Files\GrafanaLabs\grafana\bin\grafana-cli.exe` .

> **Note:** Some commands, such as installing or removing plugins, require `sudo` on Linux. If you are on Windows, run Windows PowerShell as Administrator.

https://grafana.com/docs/grafana/v9.0/cli/

ub-web-01

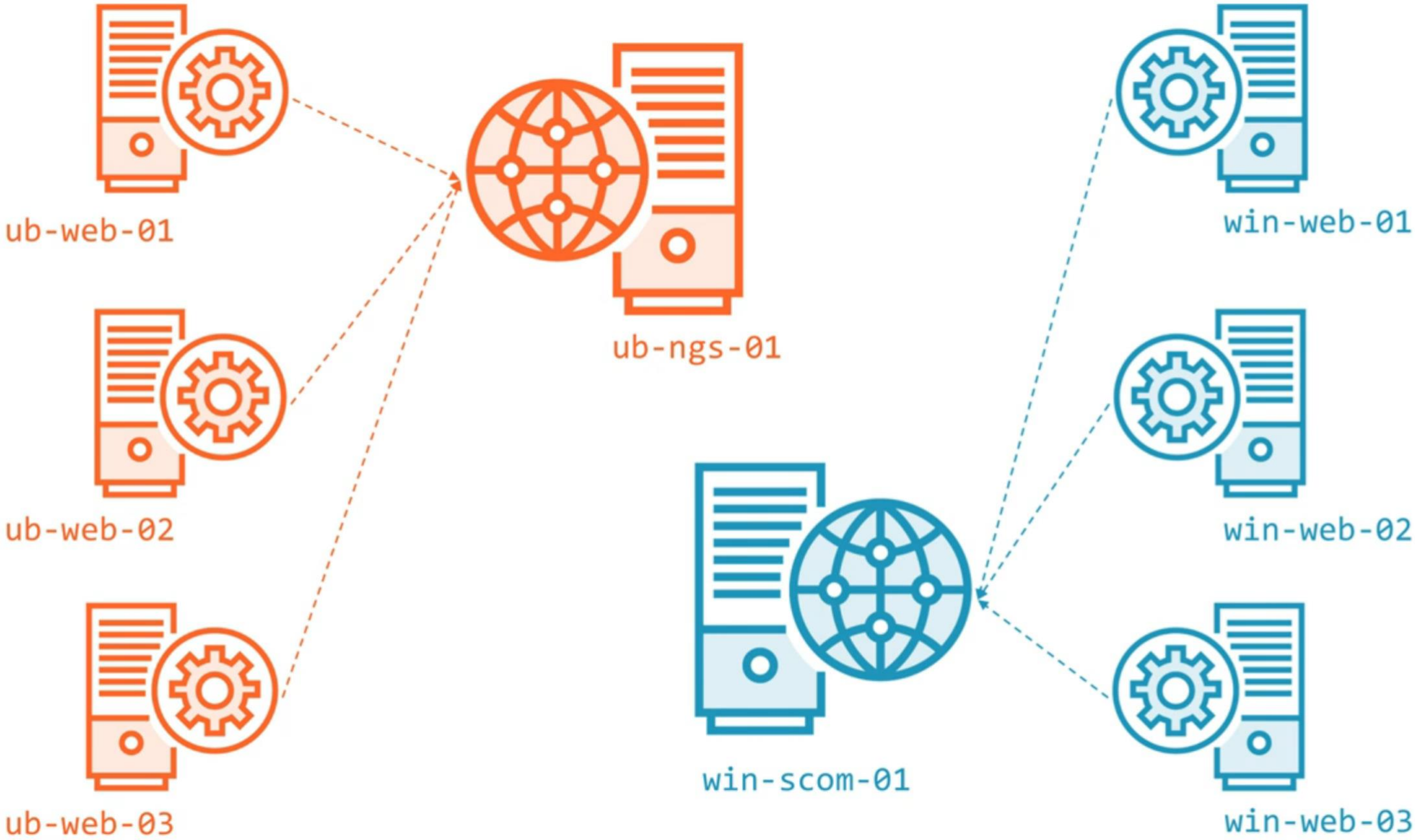
win-web-01

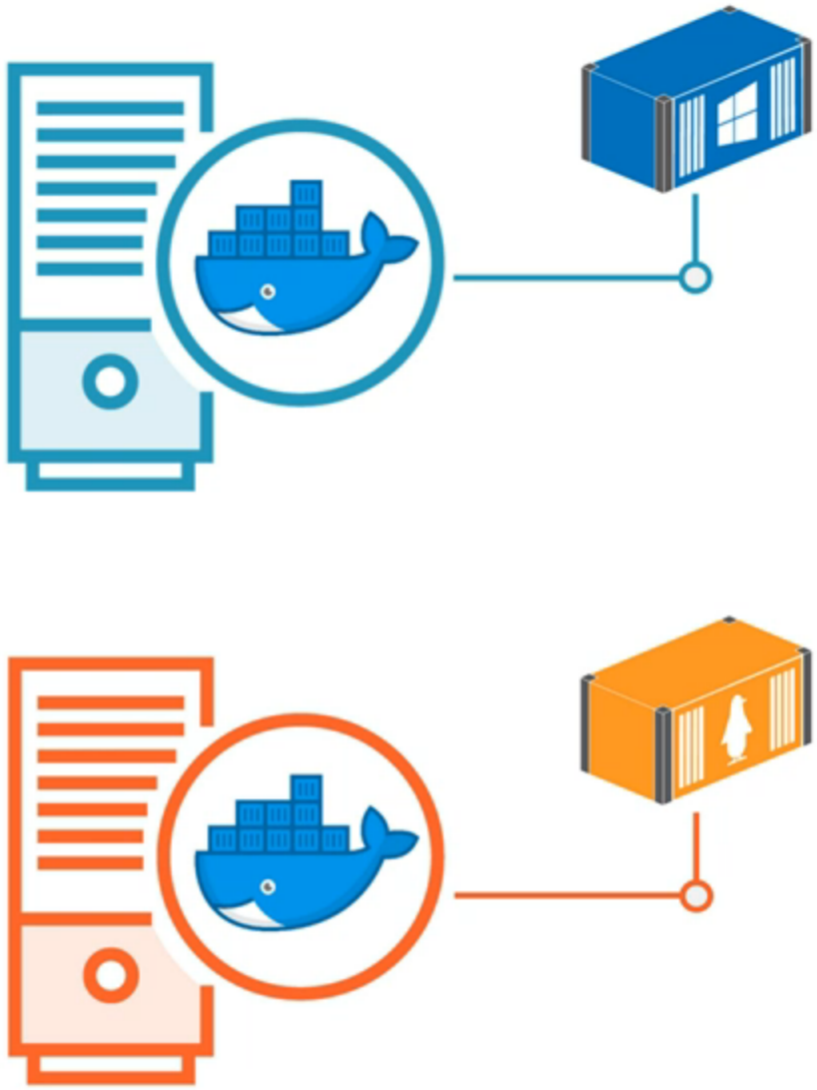
ub-web-02

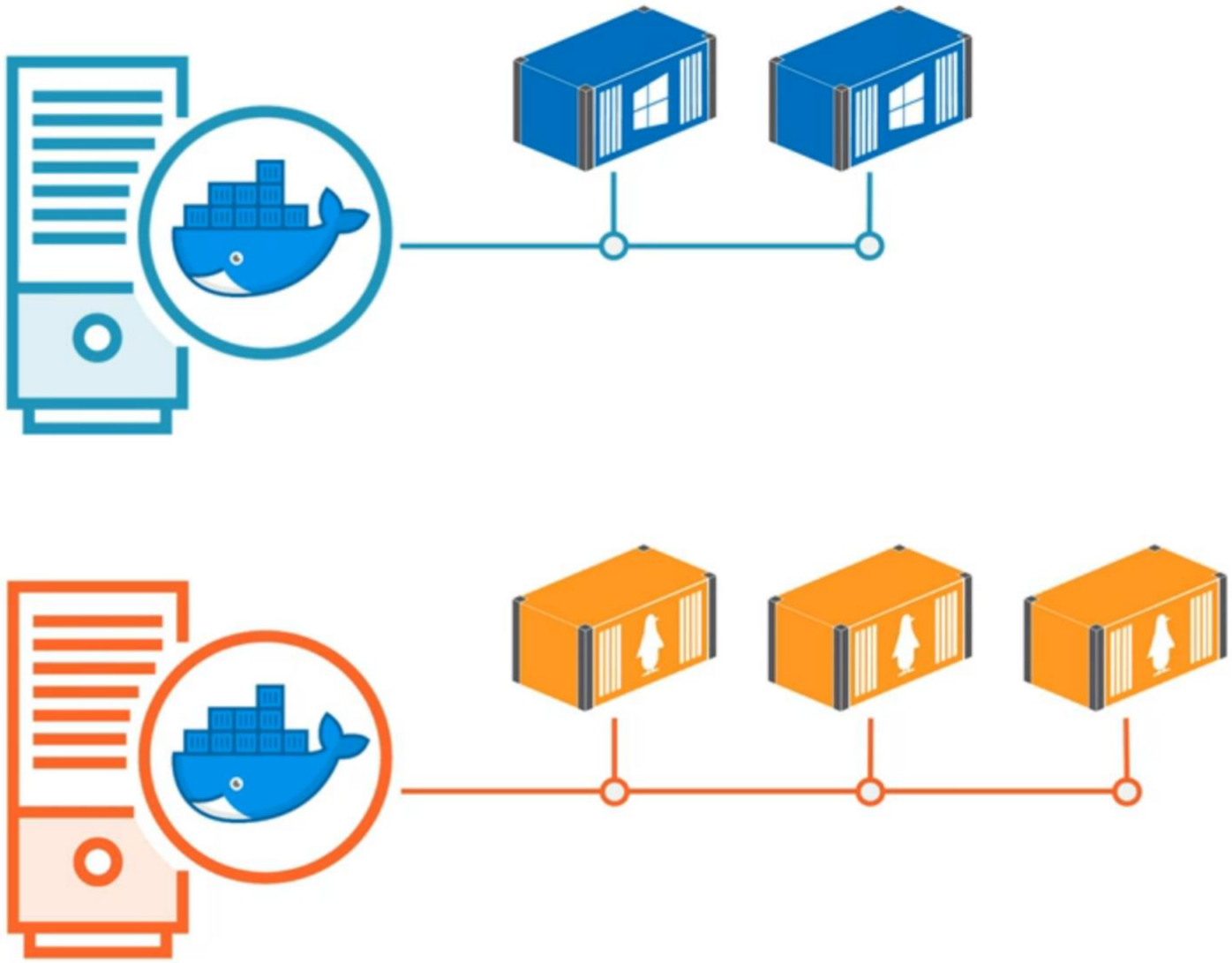
win-web-02

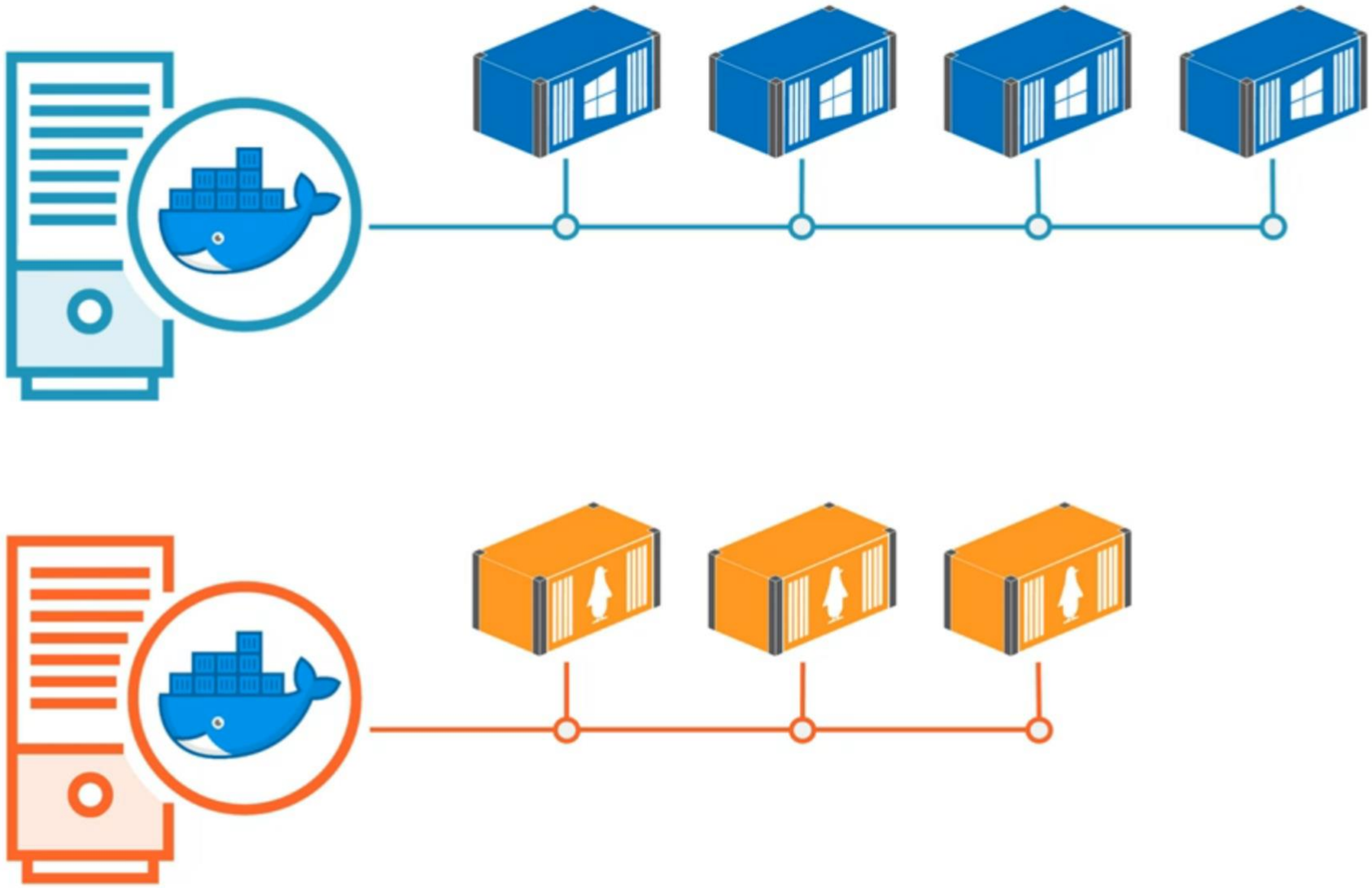
ub-web-03

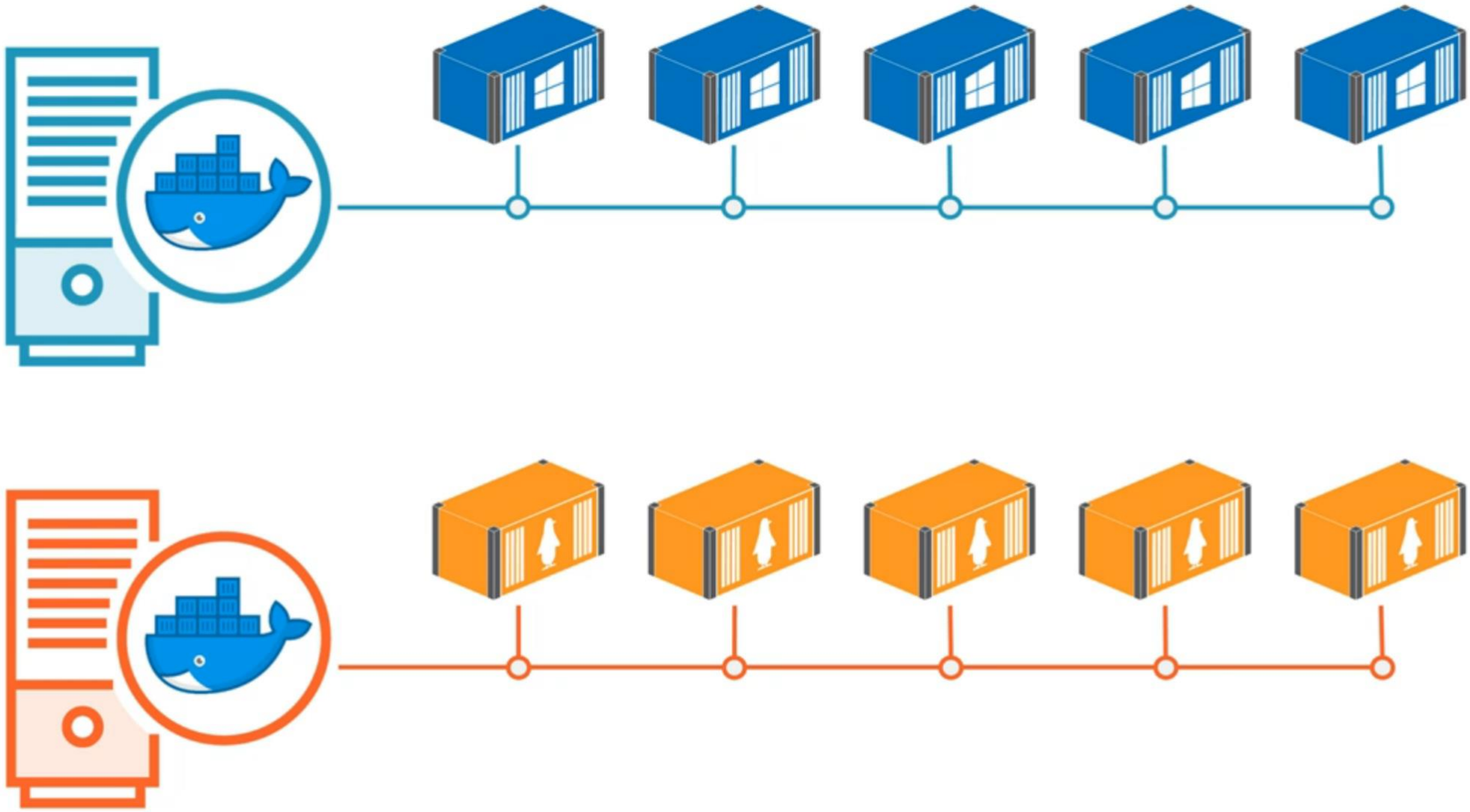
win-web-03

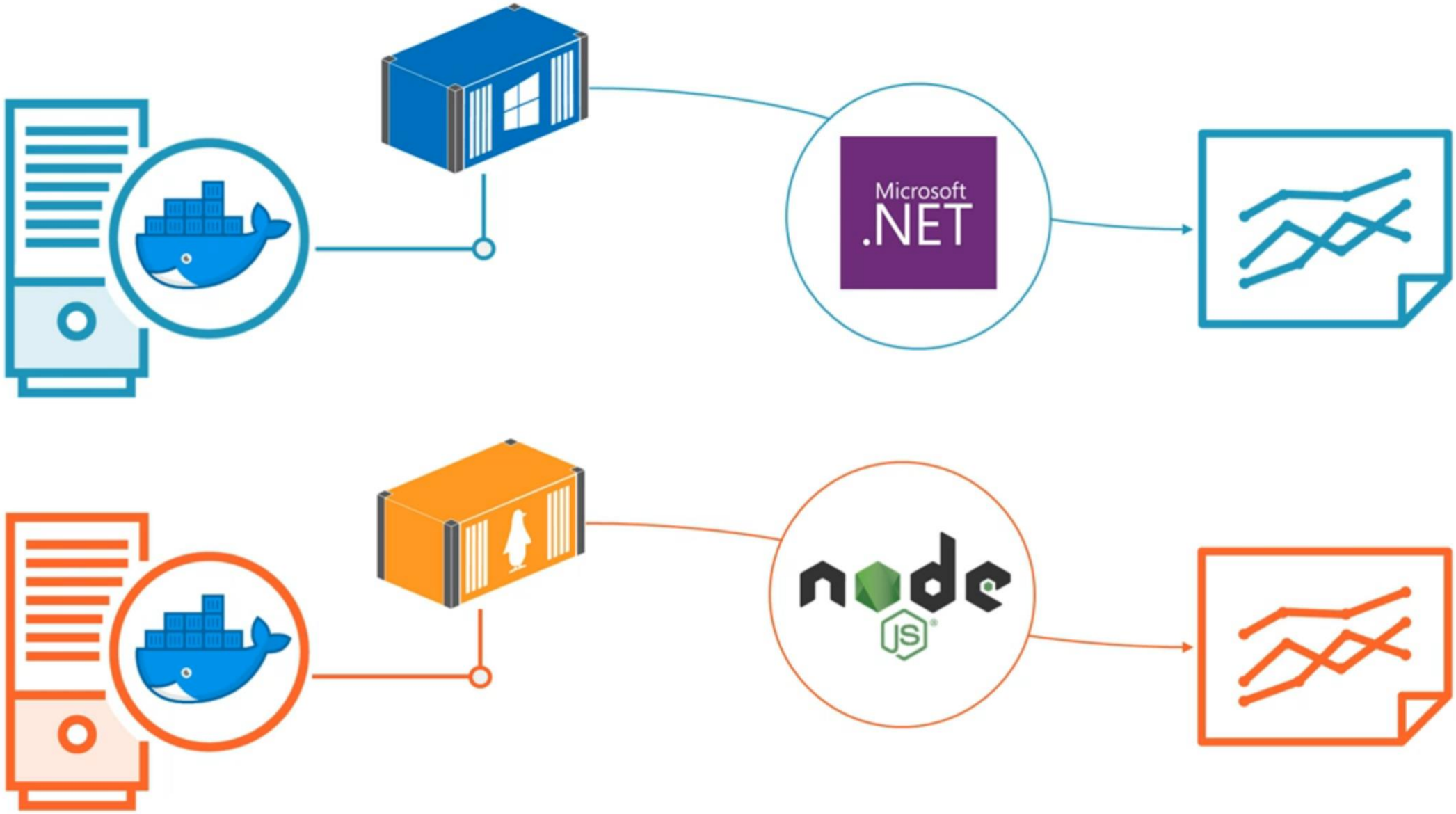

ub-web-01
ub-web-02
ub-web-03
ub-ngs-01
win-scom-01
win-web-01
win-web-02
win-web-03

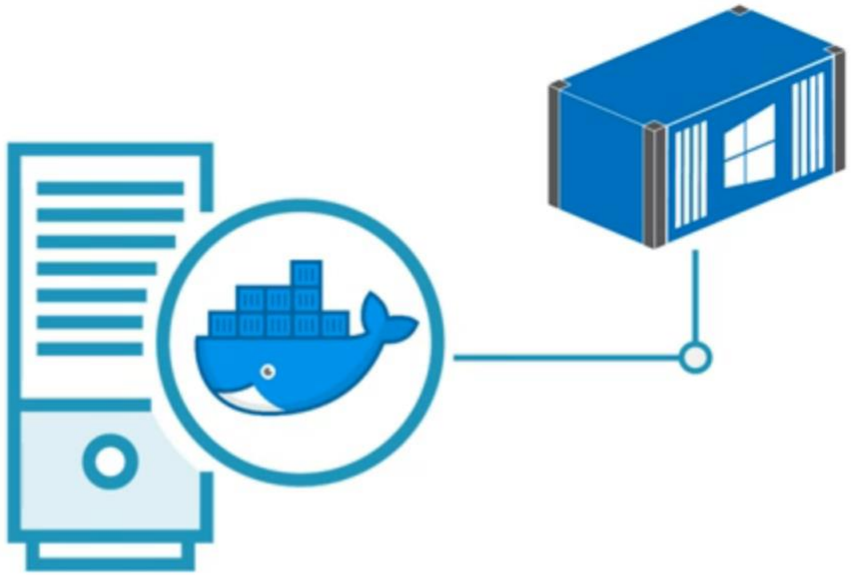






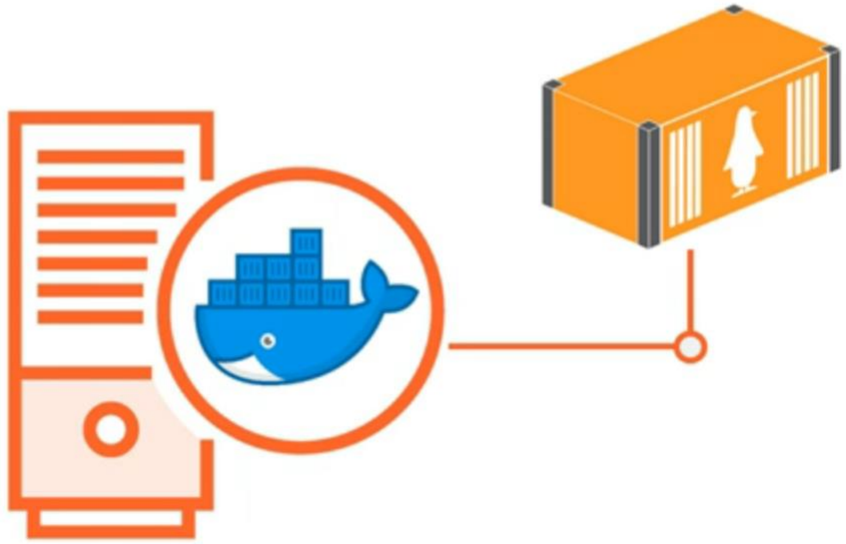



Microsoft
.NET
node
JS



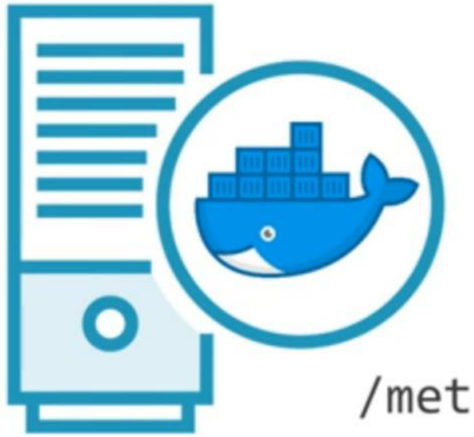
/metrics

```
# HELP process_thread_count Thread Count Perf Counter
# TYPE process_thread_count GAUGE
process_thread_count{host="EC38E36F18DE"} 21
# HELP dotnet_clr_memory_gen_0_heap_size Gen 0 heap size
# TYPE dotnet_clr_memory_gen_0_heap_size GAUGE
dotnet_clr_memory_gen_0_heap_size{host="EC38E36F18DE"} 2097152
```

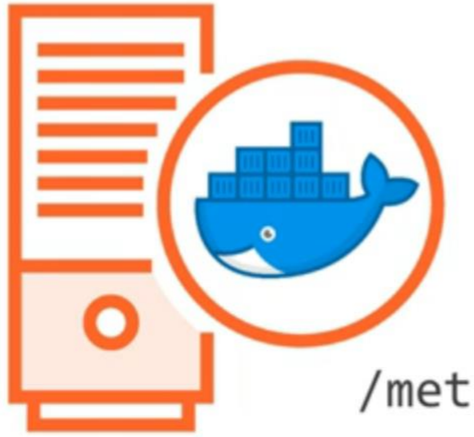


/metrics

```
# HELP jvm_threads_current Current thread count
# TYPE jvm_threads_current gauge
jvm_threads_current 37.0
# HELP jvm_memory_bytes_committed Committed bytes
# TYPE jvm_memory_bytes_committed gauge
jvm_memory_bytes_committed{area="heap",} 2.47463936E8
jvm_memory_bytes_committed{area="nonheap",} 3.8273024E7
```

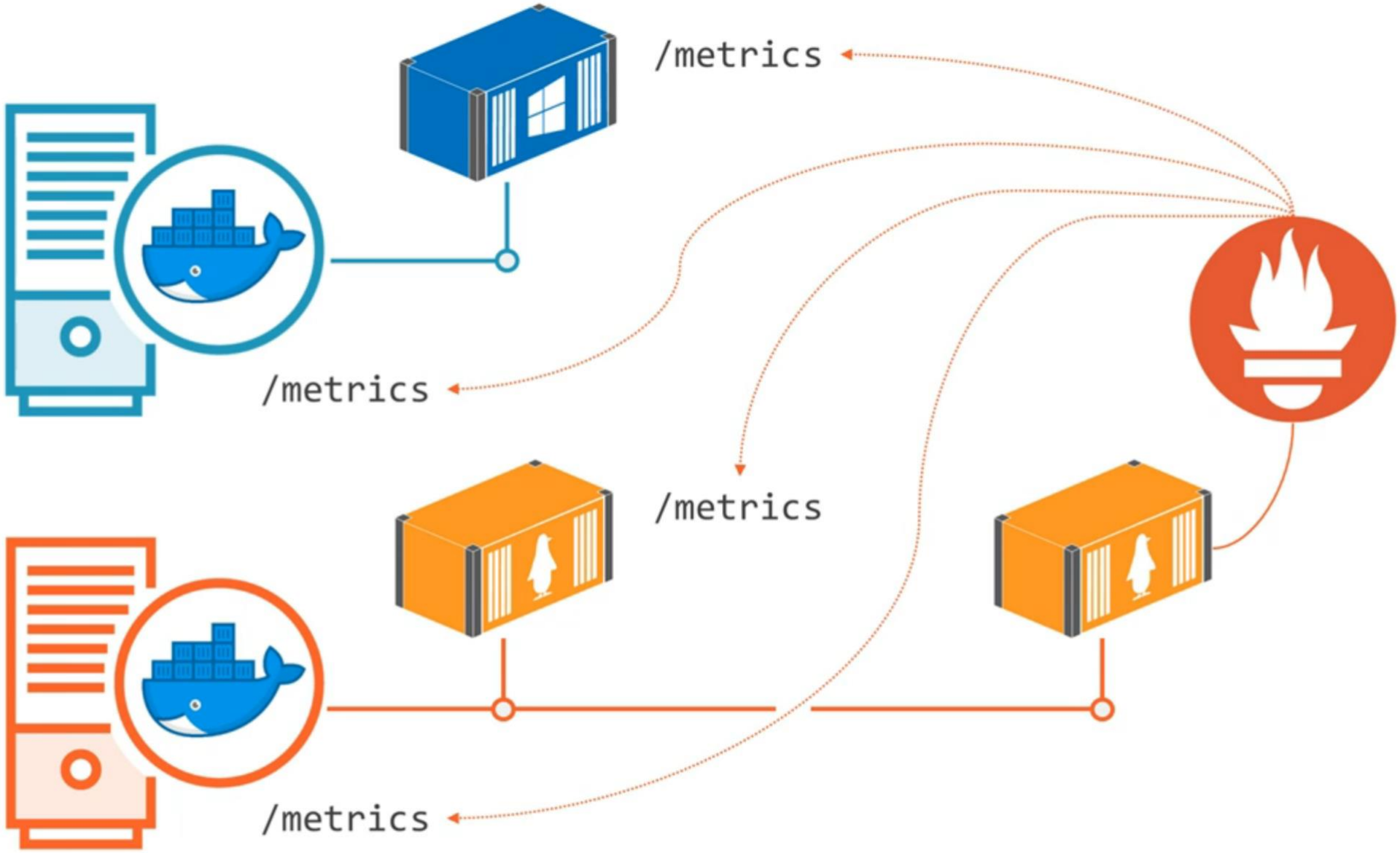


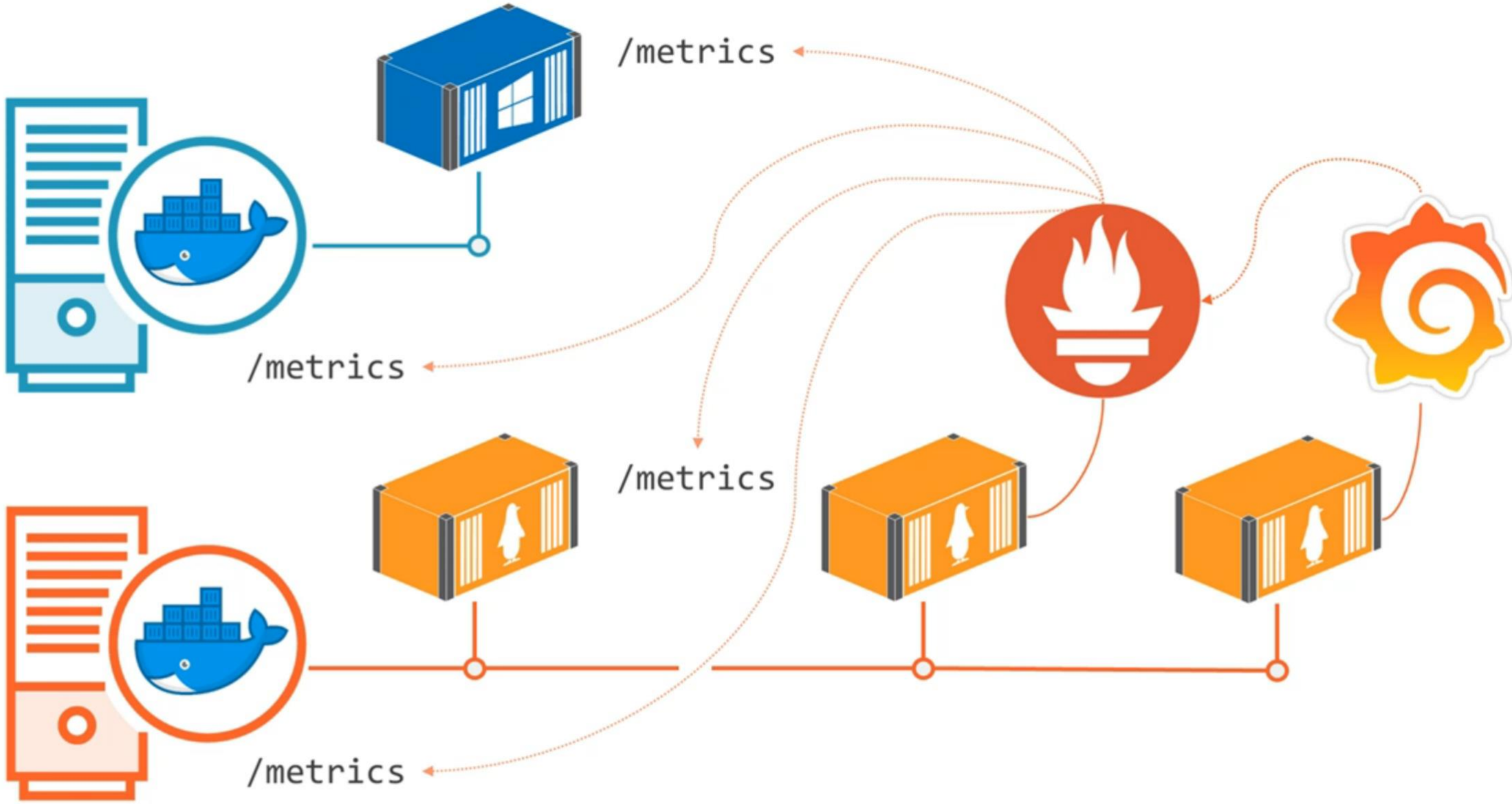
/metrics

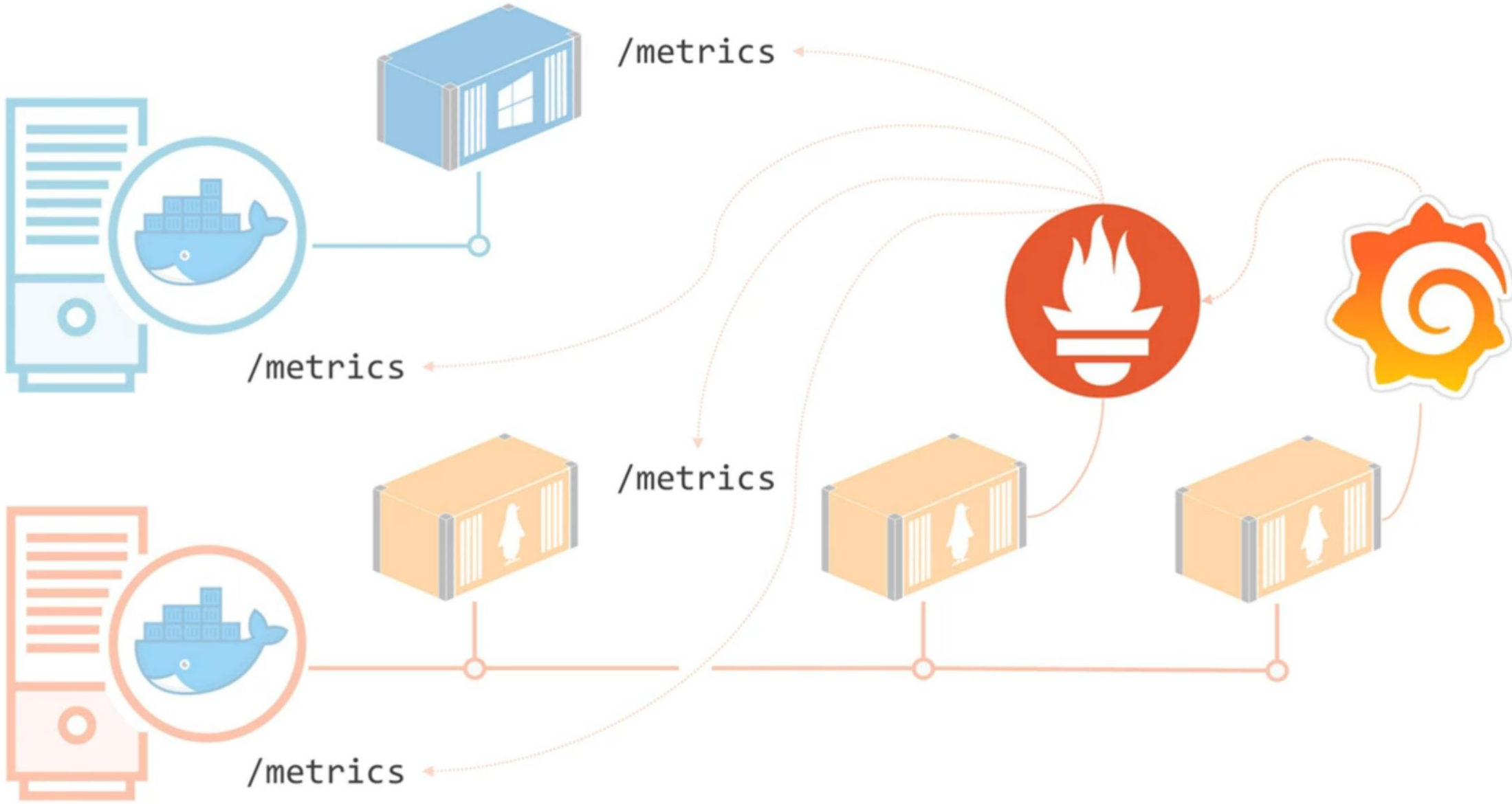


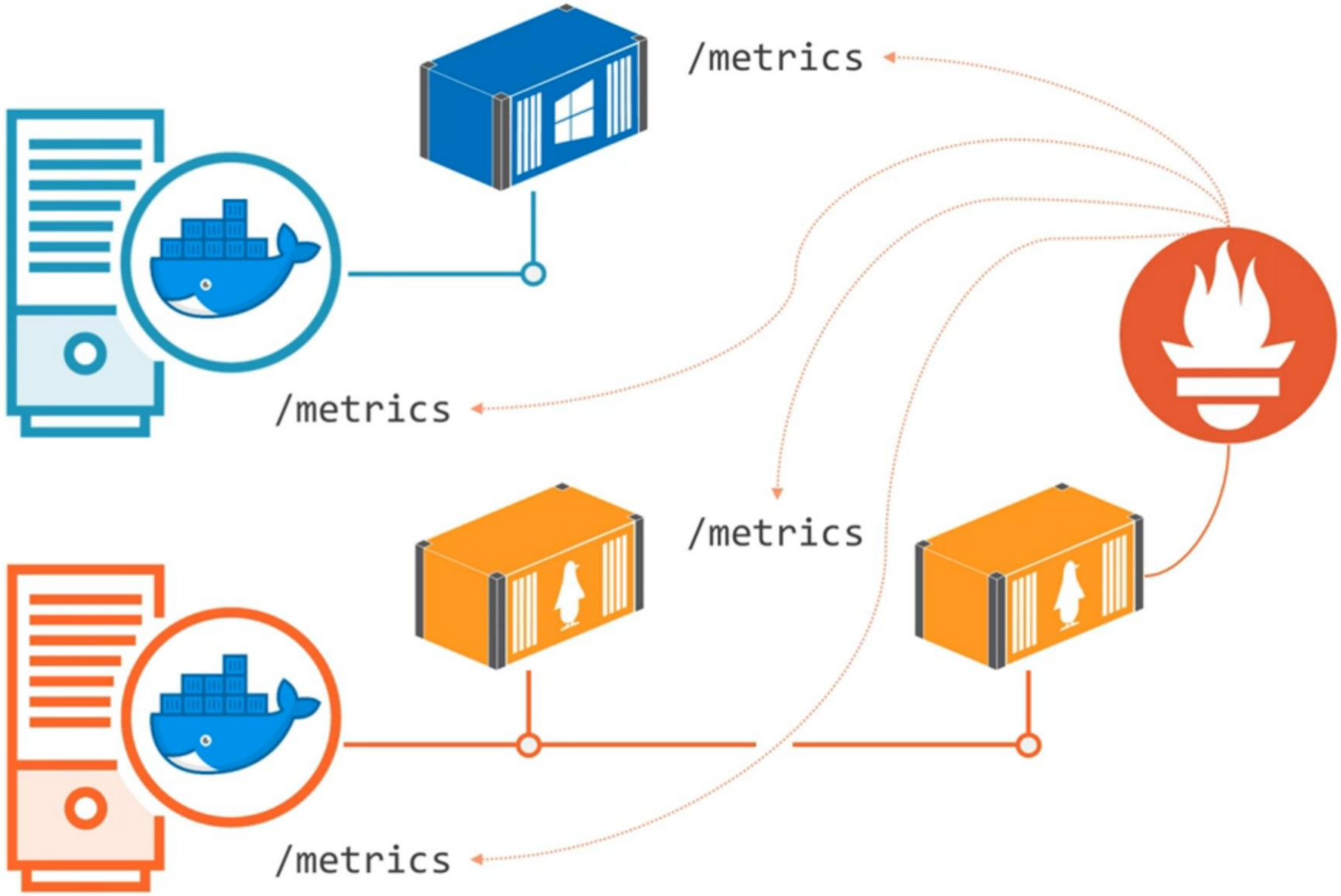
/metrics

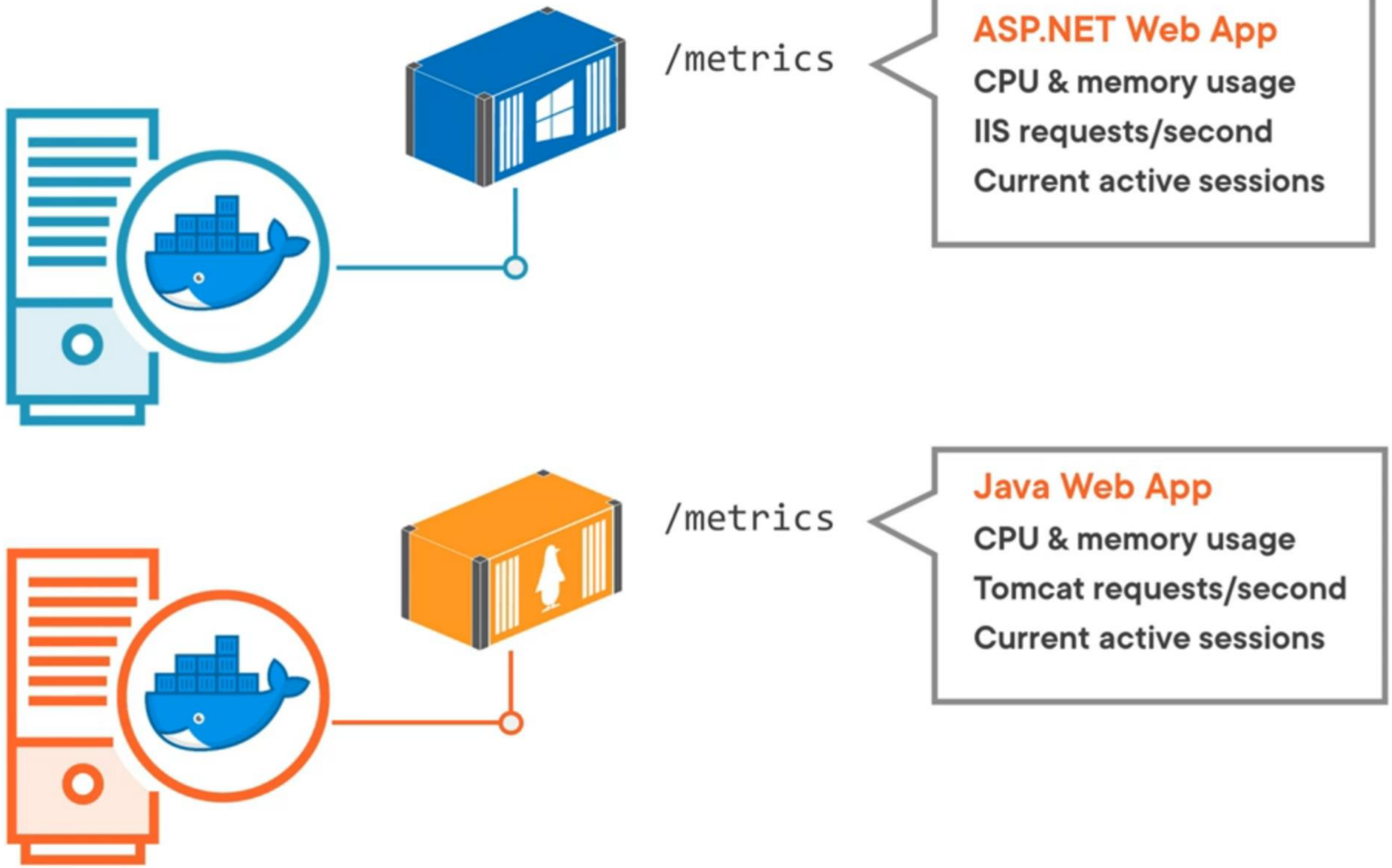
```
# HELP engine_daemon_container_states_containers The count of containers
# TYPE engine_daemon_container_states_containers gauge
engine_daemon_container_states_containers{state="paused"} 0
engine_daemon_container_states_containers{state="running"} 23
engine_daemon_container_states_containers{state="stopped"} 9
# HELP engine_daemon_engine_cpus_cpus The number of CPUs
# TYPE engine_daemon_engine_cpus_cpus gauge
engine_daemon_engine_cpus_cpus 4
```

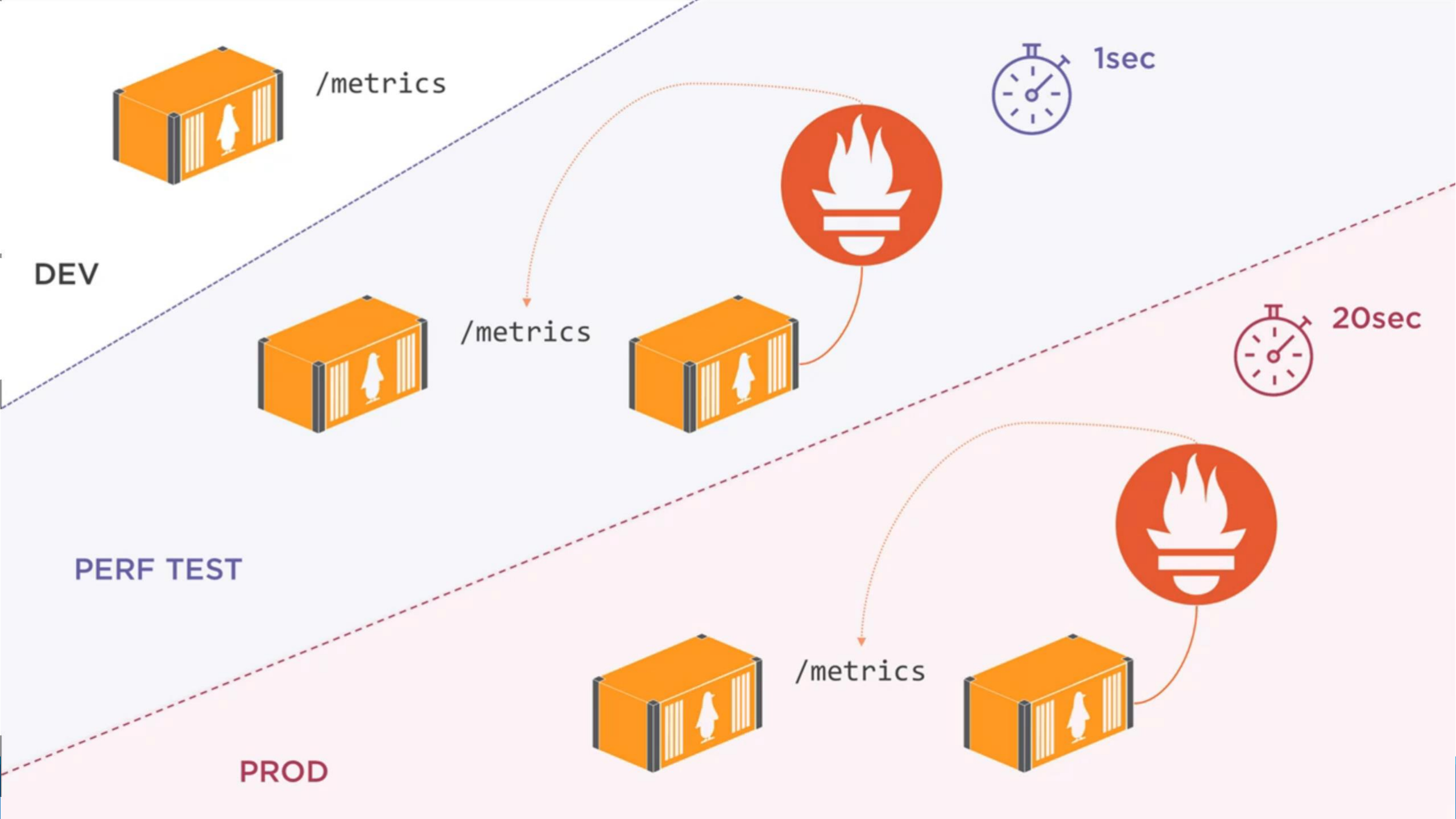

/metrics
/metrics
/metrics
/metrics


/metrics
/metrics
/metrics
/metrics


/metrics
/metrics
/metrics
/metrics


/metrics
/metrics
/metrics
/metrics


/metrics
ASP.NET Web App
CPU & memory usage
IIS requests/second
Current active sessions
/metrics
Java Web App
CPU & memory usage
Tomcat requests/second
Current active sessions


/metrics
1sec
DEV
/metrics
20sec
PERF TEST
/metrics
PROD

Approaches to Running Grafana

Shared monitoring stack

- Support scale and failover ⊕
- Single instances to manage ⊕
- Instances need high-availability ⊖
- Difficult to automate deployment ⊖
- Can't easily run the whole stack in dev ⊖

Monitoring in project stack

- Multiple instances to manage ⊖
- Run at minimum scale ⊕
- Service loss won't impact other projects ⊕
- Run the same stack in every environment ⊕
- Supports automated deployment ⊕