

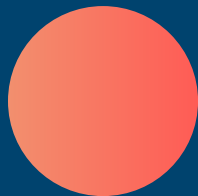


# How to Manage 600 Prometheus Instances?

**Geoffrey Beausire**

SRE @ Criteo

[g.beausire@criteo.com](mailto:g.beausire@criteo.com)





Who are we?

- We do personalized recommendation in advertisements (aka “retargeting”)

On the technical side:

- More than 40 000 servers across 10 datacenters
- 2 large Hadoop clusters (100K vcpus / 200 PB each)
- 6M requests per second (<100 ms)

# Observability at Criteo

- Dedicated team of 4 persons
- Maintain observability stack:
  - Metrics
  - Logs
  - Tracing
- Prometheus:
  - 642 instances
  - 3M samples per second
  - Most common resolution is one minute
  - More than 300 committers on the configurations files



**How do we manage 600  
instances of Prometheus?**



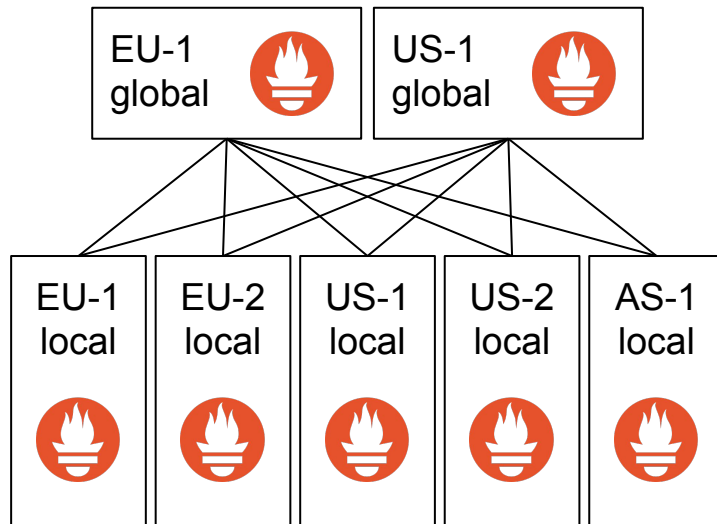
**We don't!**

**Each team is responsible for  
their own instances**

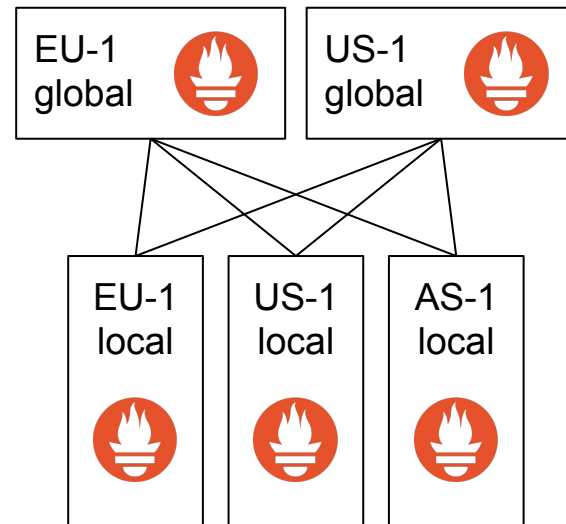
# Using perimeters

- One team has one perimeter
- Scrape services owned by the team
- Global/Local topology
- Running Mesos

Perimeter: Observability



Perimeter: NoSQL



# Why?

## Advantages:

- Low maintenance cost
- Isolation between teams
- Freedom of usage
- Clear ownership separation

## Disadvantages:

- Added workload to the client teams
- High entry cost:
  - learning how Prometheus works
  - how to use it in Criteo



**How to make it easier for  
teams to use Prometheus?**



# Reducing the workload by providing shared services

- Alertmanager
  - Generic routes are provided and preconfigured (send pages, messages via Slack given specific labels)
- Common exporters (e.g. Blackbox exporter)
- Graphite Remote Adapter

# Reducing the workload with efficient tooling

- Reliable Meta monitoring:
  - Users can choose how to be alerted
  - Alerts are actionable and well-documented
- Tooling is available to debug more complex issues (e.g. out of order errors)
- Grafana dashboard for Prometheus stats

# Reducing the workload by acting as a consulting team

The Observability team:

- Give users advice on how best to monitor their applications
- Dig deeper into complicated issues
- Take care of upgrades
- Provide workshops to onboard new users

# Reducing the entry cost through automation and self-service

- Creating a new perimeter is easy thanks to Jenkins  
(Creates all the configurations and staging pipelines)
- Prometheus is simple to test locally (one script to execute)
- Configuration is tested when creating a review:
  - Rules syntax is checked (thanks to promtool)
  - Documentation on alerts is enforced
  - Slack channels and OpsGenie receivers are validated

## Some general advice

- Start small
- Automate progressively
- Listen to the users' feedback



# Thank you

**Criteo is hiring!**

[g.beausire@criteo.com](mailto:g.beausire@criteo.com)

criteo.