

# Cortex 101



KubeCon



CloudNativeCon

North America 2019

*Chris Marchbanks*

*@csmarchbanks*

<https://github.com/cortexproject/cortex>



# About me



KubeCon



CloudNativeCon

North America 2019

- Maintainer of Prometheus and Cortex
- Building internal observability platform for Splunk Cloud
- Passionate about all things observability
- Skiing (uphill!), hiking, and climbing whenever possible





KubeCon



CloudNativeCon

North America 2019

# Show of hands



# Kubernetes and Prometheus



KubeCon



CloudNativeCon

North America 2019

## What's awesome

- First class support
- Large community of users
- Alerting and exploration without external dependencies

## Areas of struggle

- Storage limited to local disk
- Hard to query and alert across clusters



# Why use Cortex?



KubeCon



CloudNativeCon

North America 2019

A **global store** of as many metrics as you need

With **Prometheus query and alerting** capabilities

On durable, **long term storage**

Across **multiple tenants**

# Practical Use Cases



KubeCon



CloudNativeCon

North America 2019

1. Long term storage of all, or a subset of metrics
2. Centralized system for aggregating data from many isolated clusters
3. Prometheus-as-a-Service for different organizations/products (or companies)
4. Anything else you can imagine :)

# Who uses Cortex?



KubeCon



CloudNativeCon

North America 2019



Grafana Labs



ASPEN MESH



# How does Cortex do it?



KubeCon



CloudNativeCon

North America 2019

Built on the same code as Prometheus

- Horizontally scalable
- Highly available
- NoSQL and Blob storage
- Multi-tenancy embedded throughout

Cortex is a CNCF Sandbox project

<https://github.com/cortexproject/cortex>



# Cortex: Global View

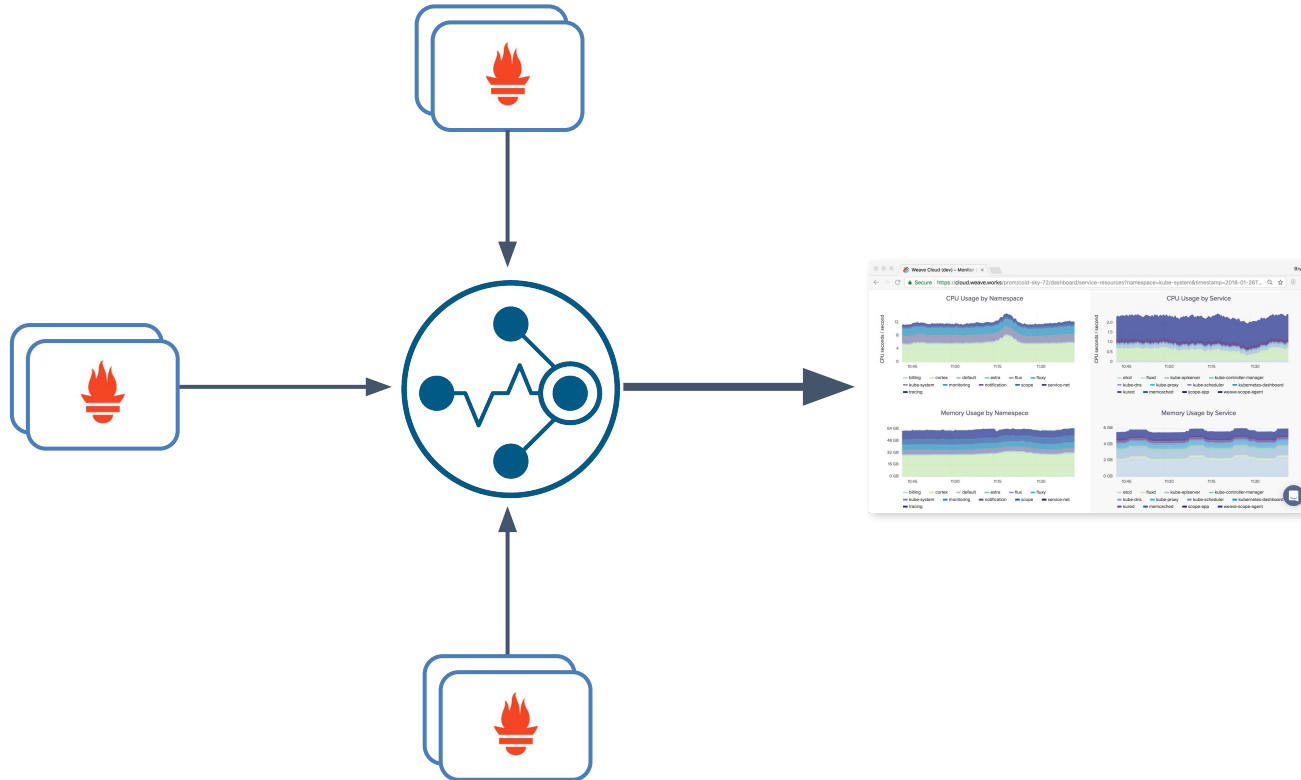


KubeCon



CloudNativeCon

North America 2019



# Cortex: Single Binary Mode



KubeCon

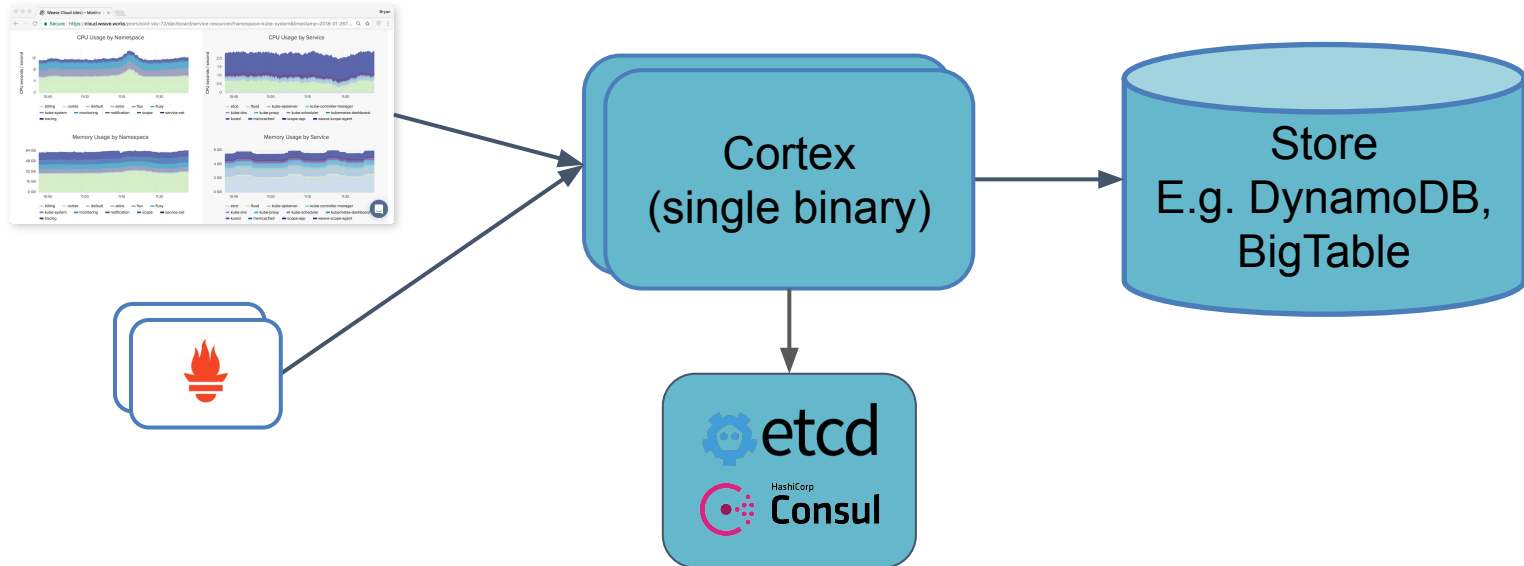


CloudNativeCon

North America 2019

Simplest way to start with clustered Cortex.

```
$ cortex -config.file=config.yaml
```



# Cortex: How to Scale?



KubeCon



CloudNativeCon

North America 2019

Cortex splits into multiple components via a command line flag.

```
-target=<component_name>
```

Scale each component independently.

Components can be considered part of three subsystems:

1. Ingest
2. Query
3. Rule/Alert evaluation

# Cortex: Ingest

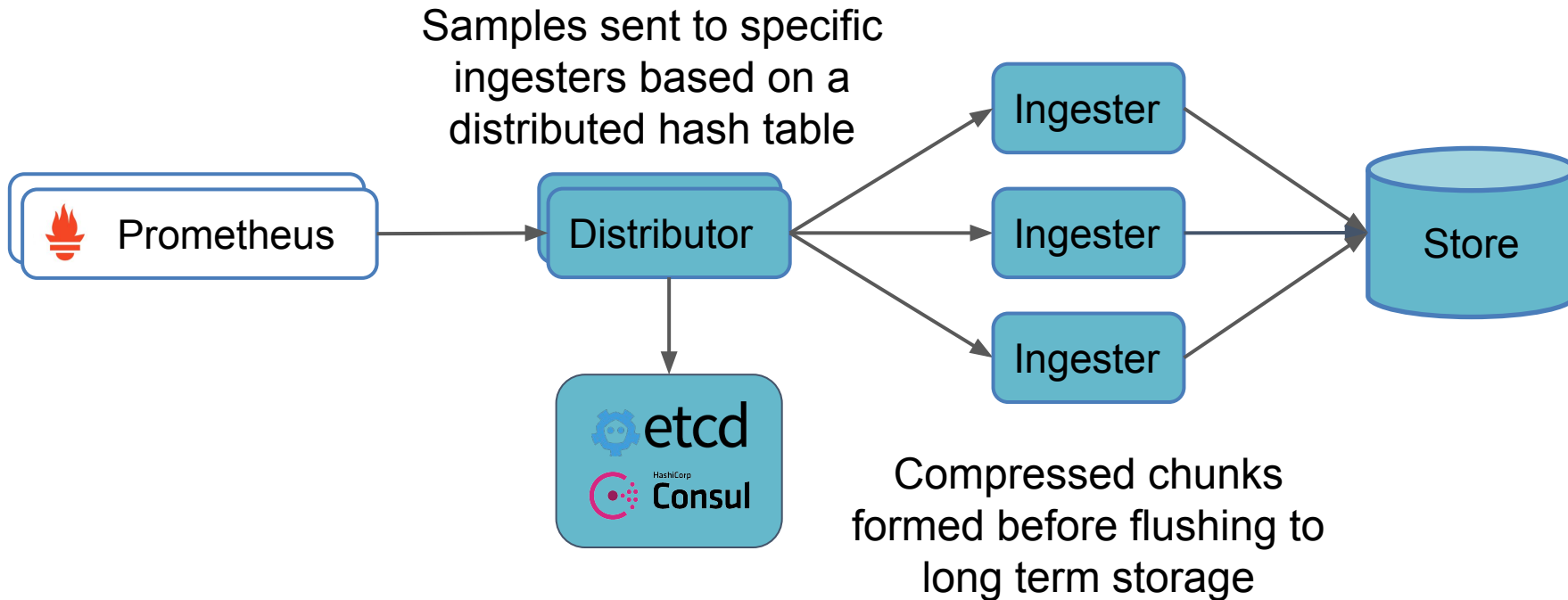


KubeCon



CloudNativeCon

North America 2019



# Cortex: Ingest Compression

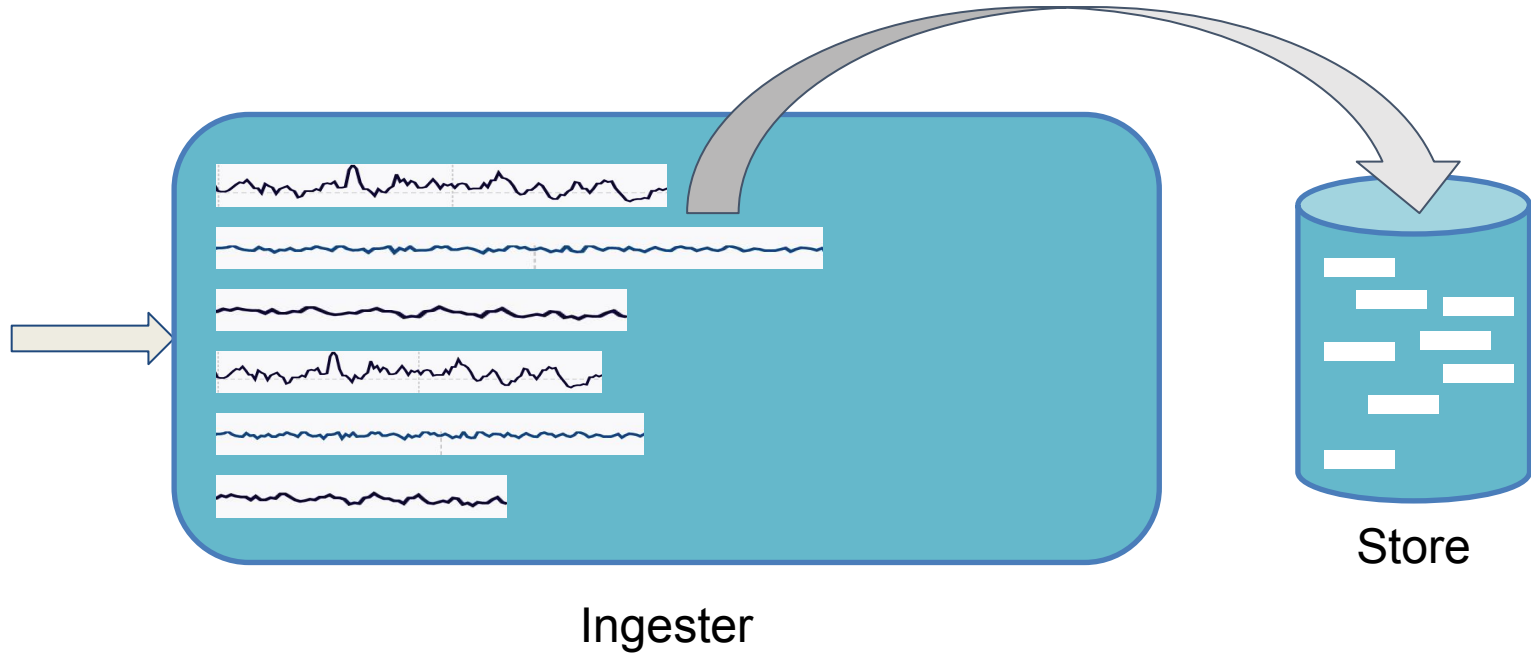


KubeCon



CloudNativeCon

North America 2019



*Gorilla compression, same as Prometheus:* <http://www.vldb.org/pvldb/vol8/p1816-teller.pdf>

# Cortex: Query

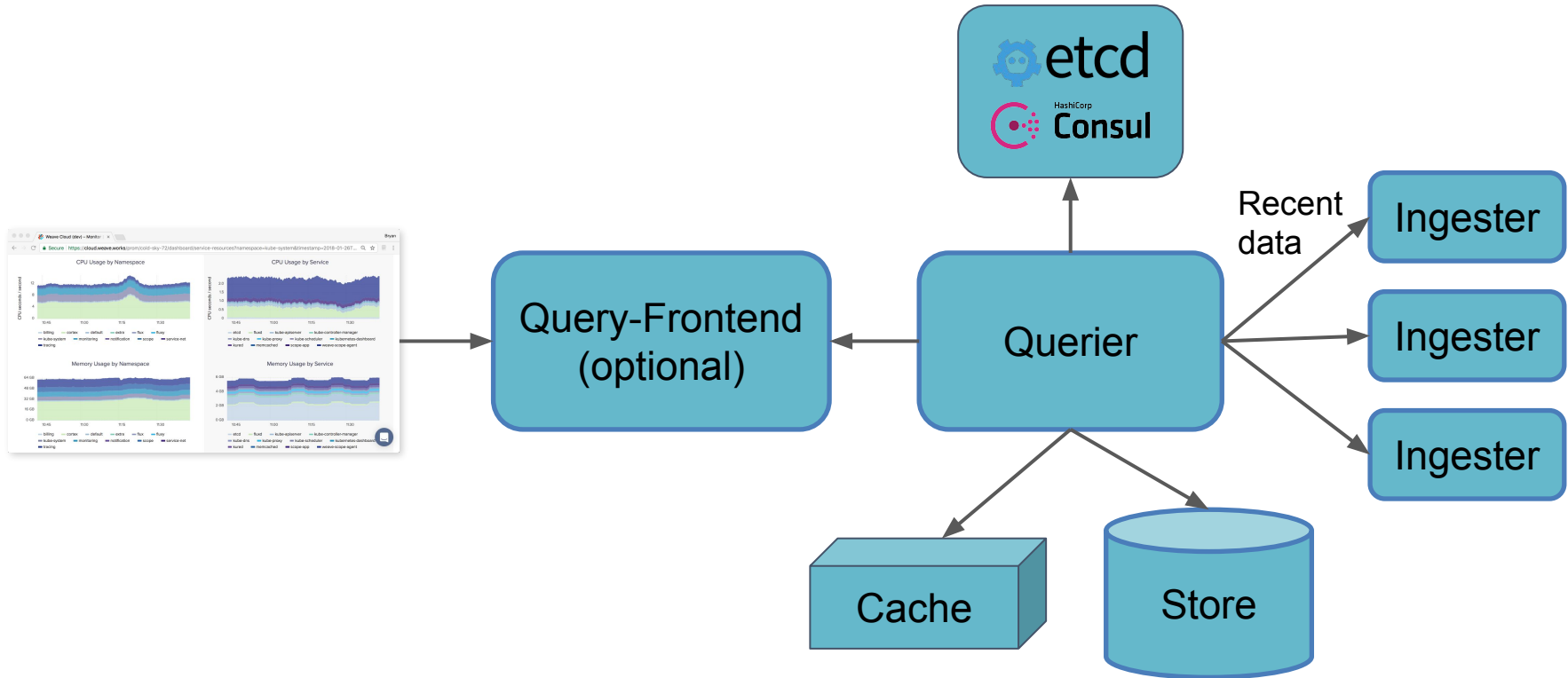


KubeCon



CloudNativeCon

North America 2019



# Cortex: Ingest and Query

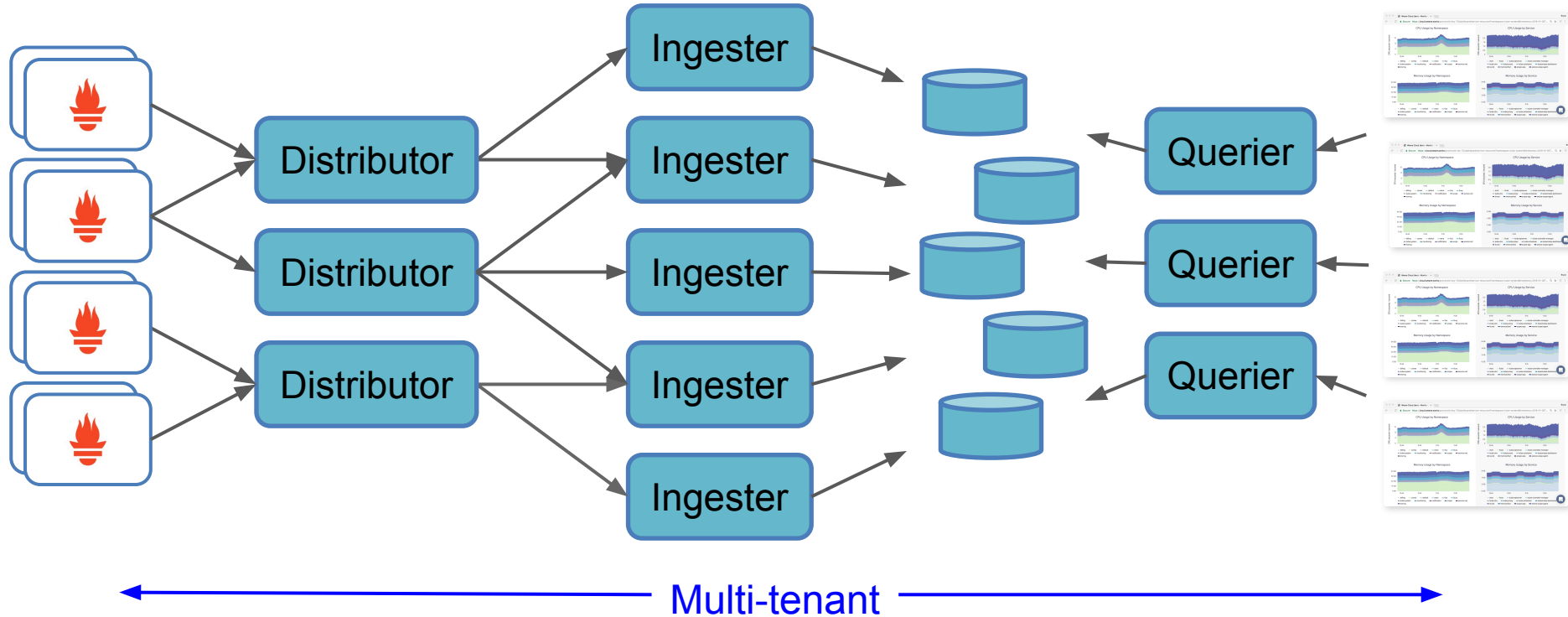


KubeCon



CloudNativeCon

North America 2019



# Cortex: Global Rule Evaluation

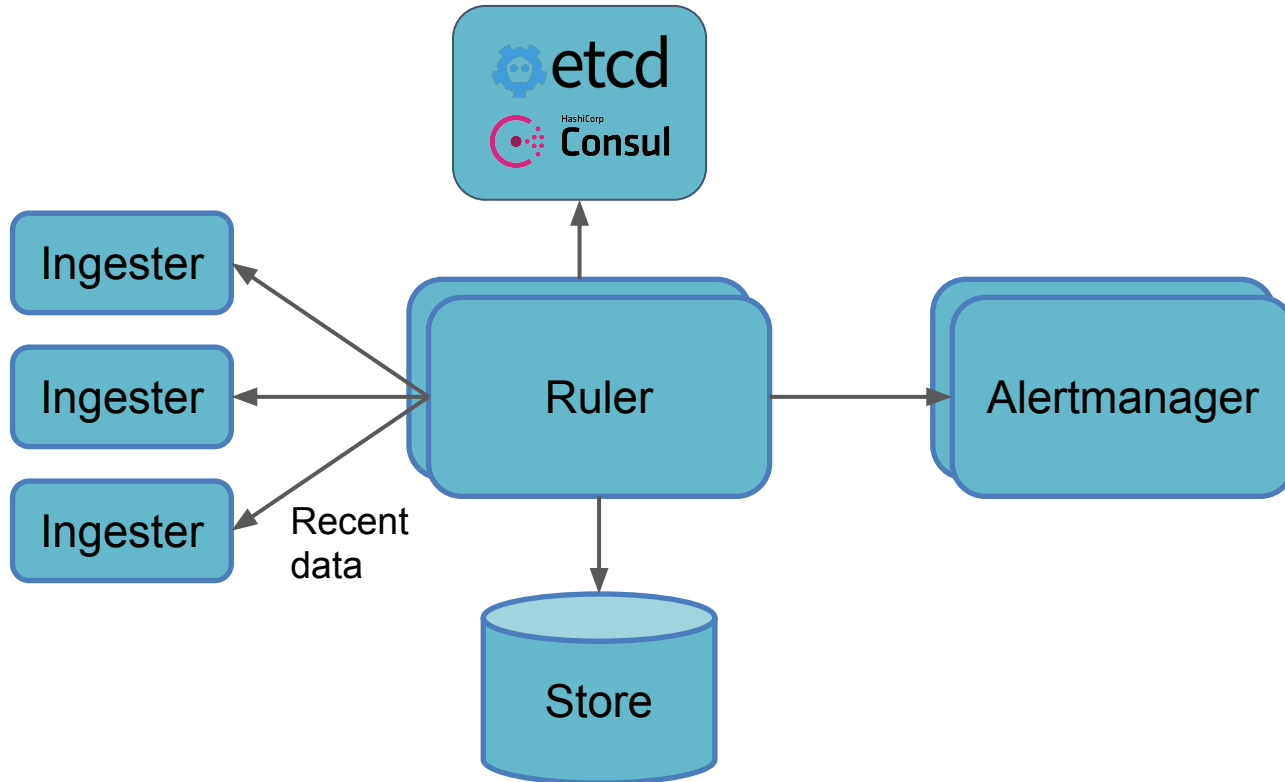


KubeCon



CloudNativeCon

North America 2019





# Cortex: Long Term Storage



KubeCon



CloudNativeCon

North America 2019



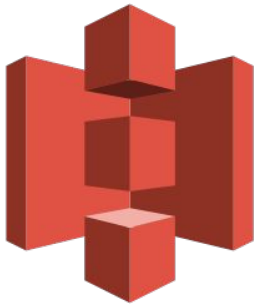
DynamoDB



Google Cloud Bigtable



*cassandra*



S3



Google Cloud Storage



**BoltDB**

# What's New?



KubeCon



CloudNativeCon

North America 2019

## Recent (experimental) features:

1. Ingesters create and ship blocks instead of chunks
  - a. Uses much of the same code as Thanos
2. Internal gossip to remove Consul/Etcd

## Upcoming work:

3. Write-ahead log for ingesters
  - a. Easier recovery when an ingester is restarted
4. Query Parallelization

# Thanos: An Alternative



KubeCon



CloudNativeCon

North America 2019

<https://thanos.io/>

- Decentralized architecture
- Only needs object storage
- Also a CNCF project



Excellent talk from PromCon comparing Cortex & Thanos:

<https://promcon.io/2019-munich/talks/two-households-both-alike-in-dignity-cortex-and-thanos/>



KubeCon



CloudNativeCon

North America 2019

# Questions?

*Chris Marchbanks*

*@csmarchbanks*

<https://github.com/cortexproject/cortex>

