

Kubernetes advanced

Agenda

- Docker
- Advanced Kubernetes
- Helm & Prometheus
- EFK
- Harness/spinnaker
- Grafana
- Service mesh - Istio

Training Format

- Concepts
- Hands-on exercises
- Assignments
- Interactive

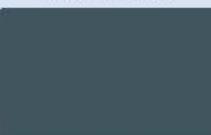
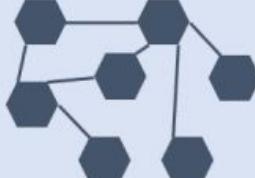
Timing

- Start time at 5.30pm
- Break 7 pm to 7.15 pm
- End at 8.30 pm

Infra setup

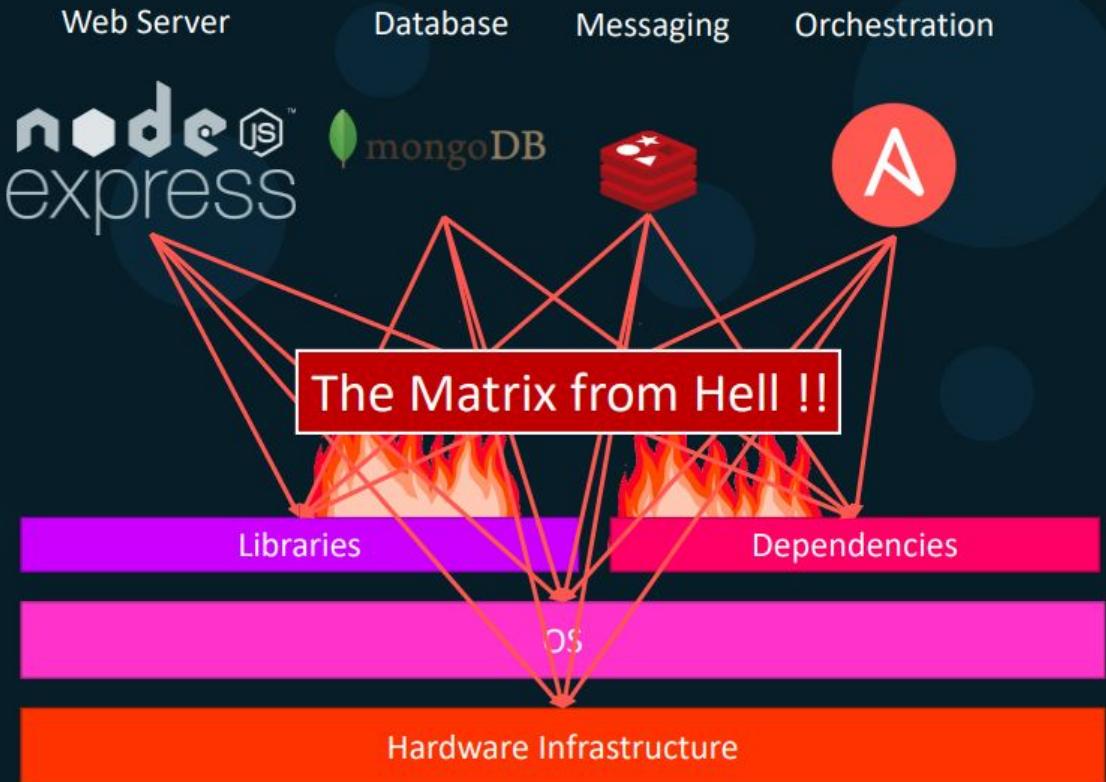
- VM
- Docker
- Kubernetes

Evolution of Computing

Development Process	Application Architecture	Deployment and Packaging	Application Infrastructure
Waterfall 	Monolithic 	Physical Server 	Datacenter 
Agile 	N-Tier 	Virtual Servers 	Hosted 
DevOps 	Microservices 	Containers 	Cloud 

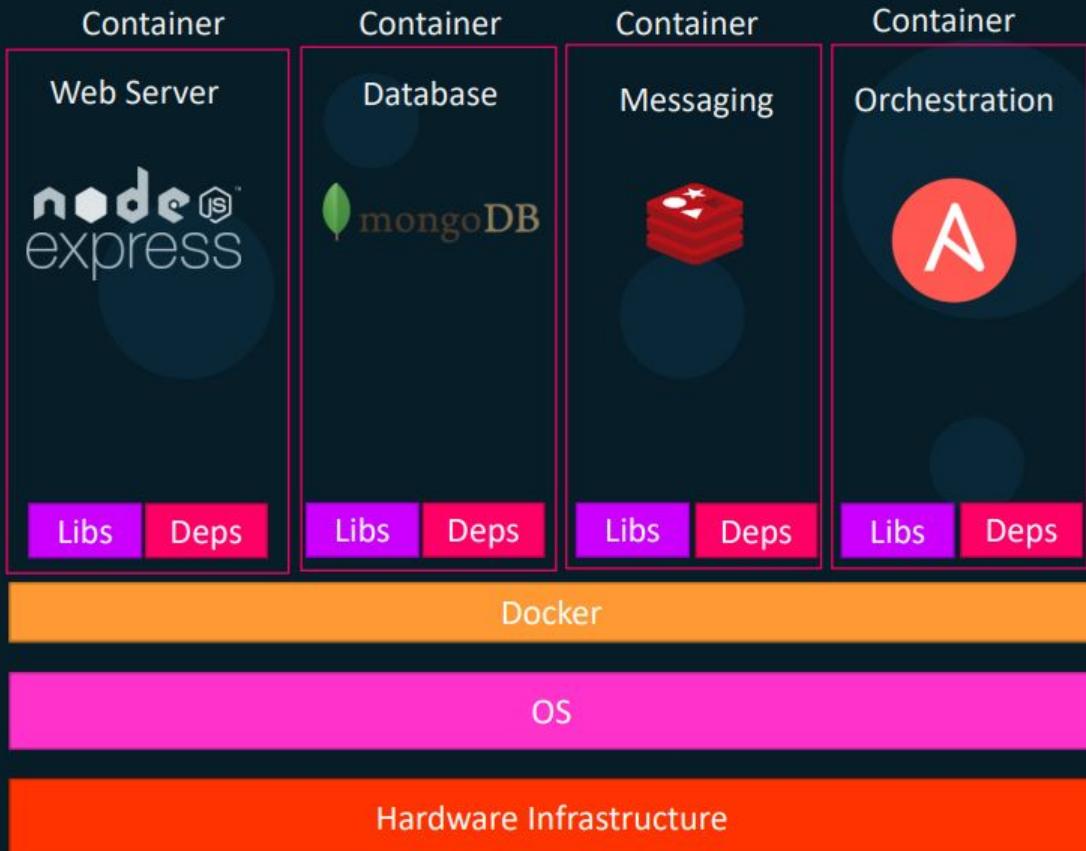
Why do you need docker?

- Compatibility/Dependency
- Long setup time
- Different Dev/Test/Prod environments

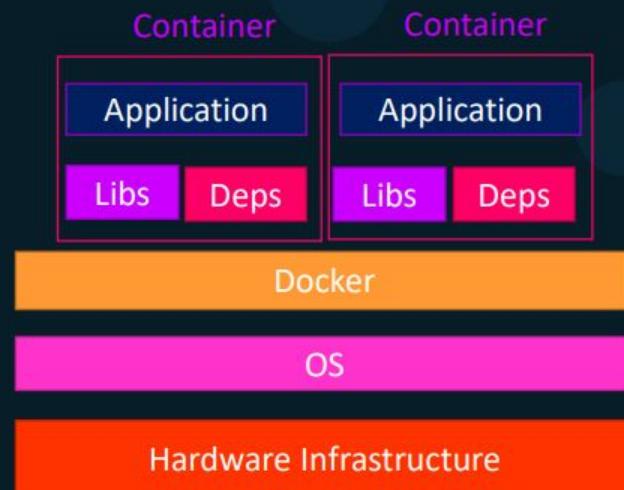
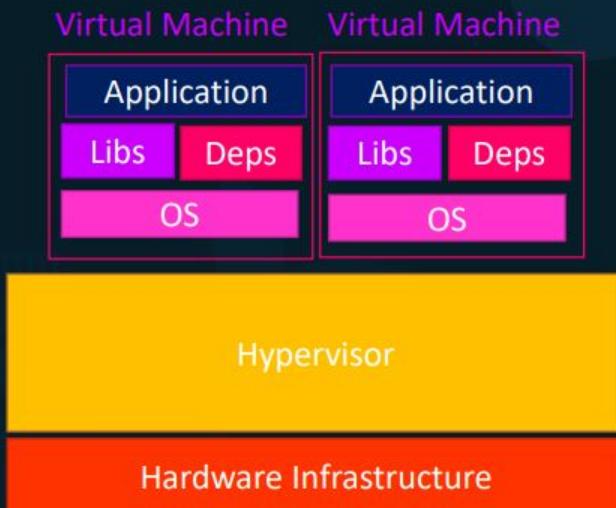
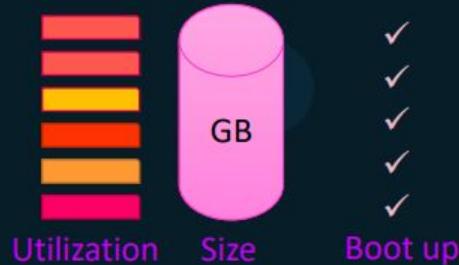


What can it do?

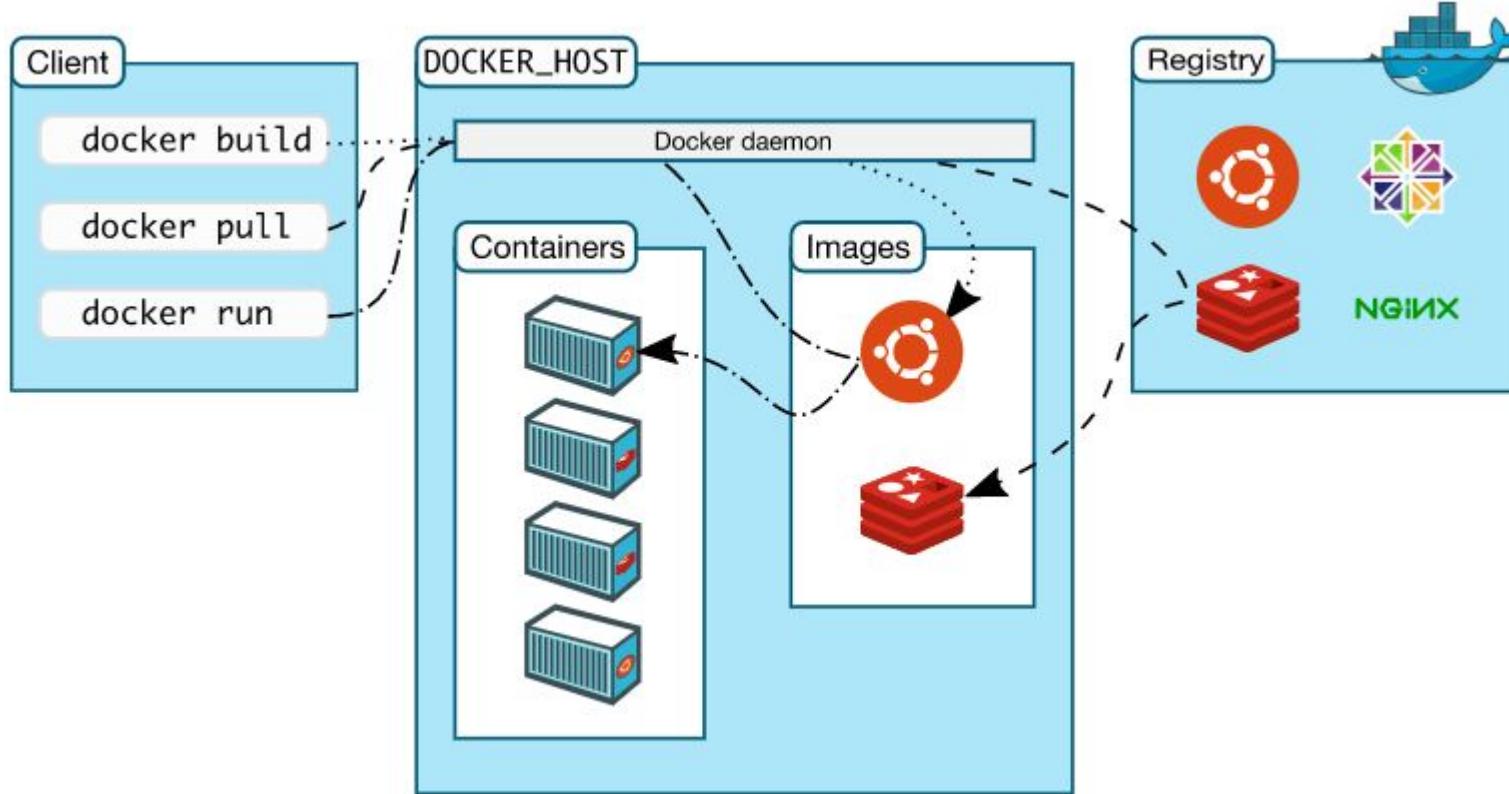
- Containerize Applications
- Run each service with its own dependencies in separate containers



Containers vs Virtual Machines



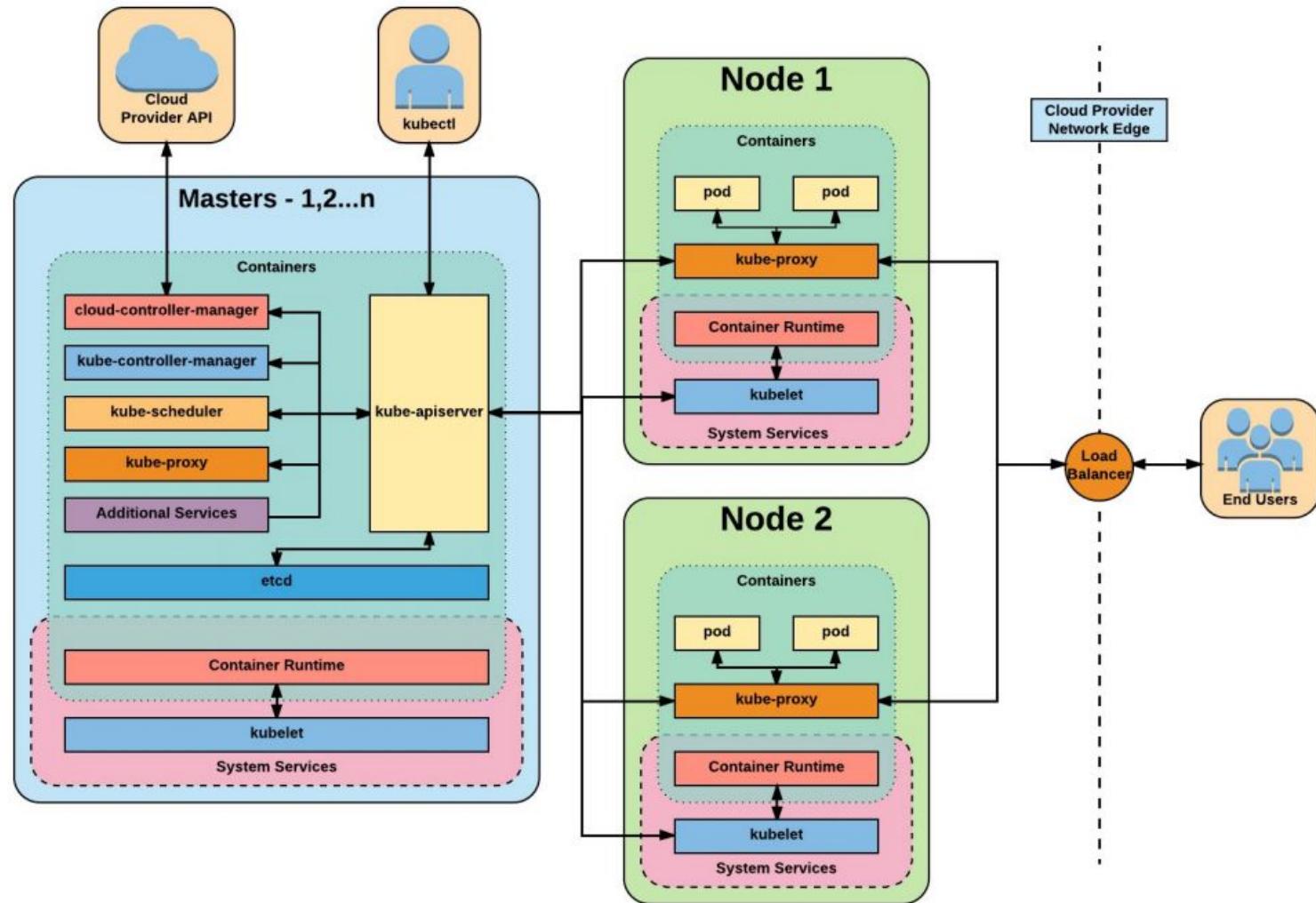
Docker architecture



Kubernetes

Kubernetes or K8s was a project spun out of Google as a open source next-gen container scheduler designed with the lessons learned from developing and managing Borg and Omega.

Kubernetes was designed from the ground-up as a loosely coupled collection of components centered around deploying, maintaining, and scaling applications.



Monitoring

Helm

Helm

- Helm the best way to find, share and use software built for Kubernetes
(definition from <https://helm.sh/>)
- Helm is a **package manager** for Kubernetes
- It helps you to manage Kubernetes **applications**
- Helm is maintained by the **CNCF - The Cloud Native Computing Foundation** (together with Kubernetes, fluentd, linkerd, and others)
 - It is now maintained in collaboration with **Microsoft, Google, Bitnami** and the **helm contributor community**

<https://helm.sh/>

Helm - charts

- Helm uses a packaging format called **charts**
 - A **chart** is a **collection** of **files** that **describe** a set of Kubernetes **resources**
 - A single chart can **deploy an app**, a piece of software, or a database
 - It can have dependencies, e.g. to install wordpress chart, you need a mysql chart
 - You can write **your own chart** to deploy your application on Kubernetes using helm

Helm - Common commands

Command	Description
helm init	Install tiller on the cluster
helm reset	Remove tiller from the cluster
helm install	Install a helm chart
helm search	search for a chart
helm list	list releases (installed charts)
helm upgrade	upgrade a release
helm rollback	rollback a release to the previous version

commands

Install on ubuntu

```
curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -
```

```
sudo apt-get install apt-transport-https --yes
```

```
echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee  
/etc/apt/sources.list.d/helm-stable-debian.list
```

```
sudo apt-get update
```

```
sudo apt-get install helm
```

<https://helm.sh/docs/intro/install/>

commands

helm

helm search hub

helm search hun mysql

helm repo add stable <https://charts.helm.sh/stable>

helm search repo stable

helm repo list

helm repo update

commands

```
helm install stable/mysql --generate-name
```

```
kubectl get all
```

```
helm install myairflow stable/airflow
```

Helm ls

```
helm uninstall airflow
```

```
helm create mychart
```

```
cd mychart
```

Install Promethues

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

```
helm repo add kube-state-metrics https://kubernetes.github.io/kube-state-metrics
```

```
helm repo update
```

```
helm install promethues prometheus-community/prometheus
```

Main Component: Prometheus Server

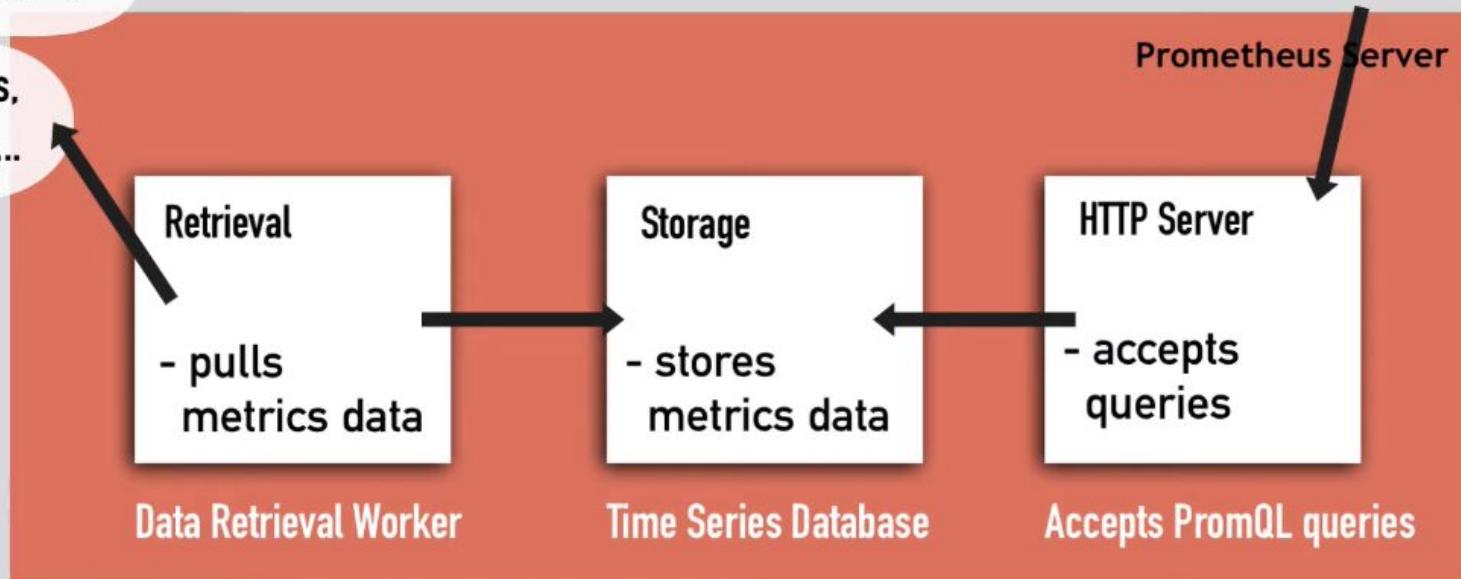
- does the actual monitoring work

Applications

Services,
Servers...

Prometheus
Web UI

Grafana
etc.



Targets and Metrics

What does Prometheus monitor?

Linux/Windows
Server

Single
Application

Apache Server

Service, like
Database

Which units are monitored of
those targets?

CPU Status

Requests Count

Request
Duration

Memory/Disk
Space Usage

Exceptions
Count

Metrics

- Format: **Human-readable** text-based
- Metrics entries: **TYPE** and **HELP** attributes

HELP

👉 description of what the metrics is

TYPE

👉 3 metrics types

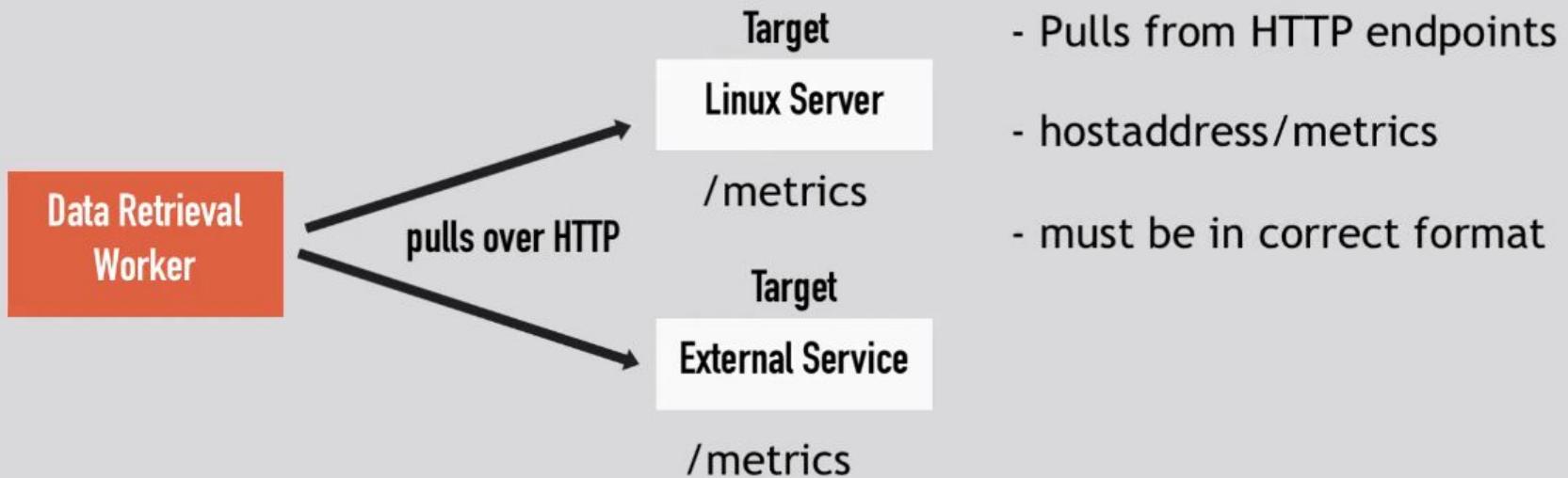
1) Counter

...how many times x happened

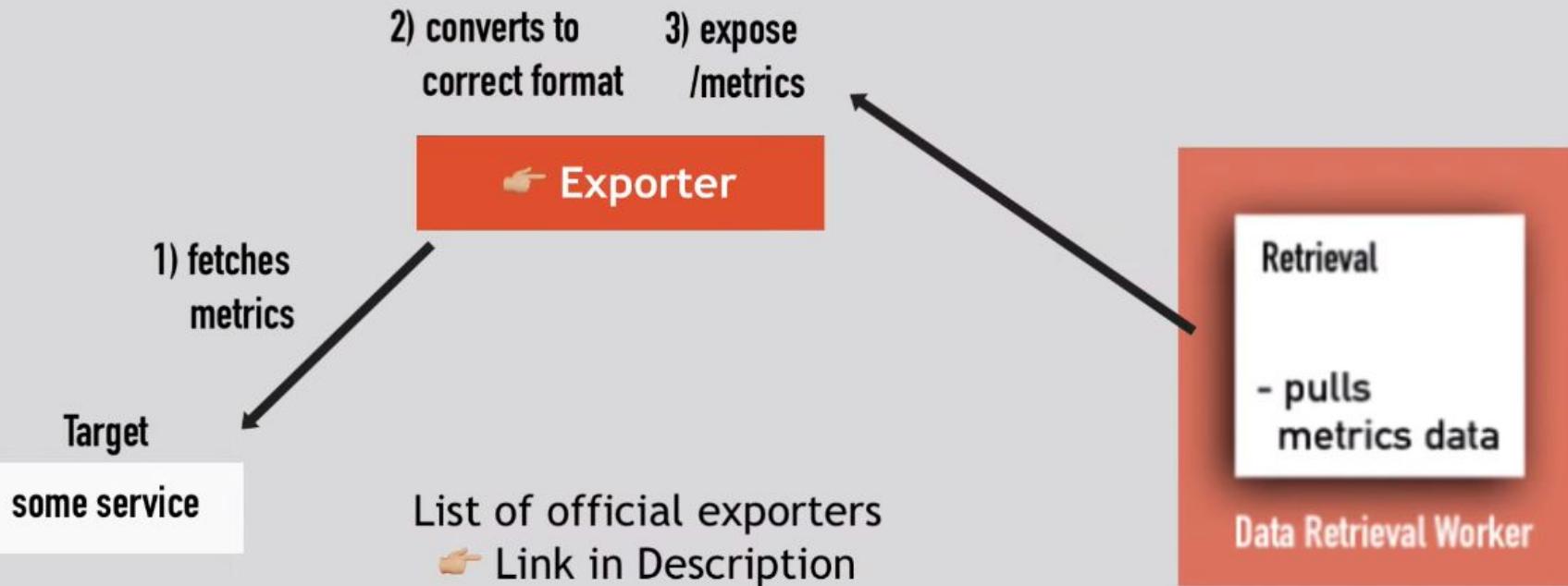
2) Gauge

...what is the current value of x now?

Collecting Metrics Data from Targets



Target Endpoints and Exporters

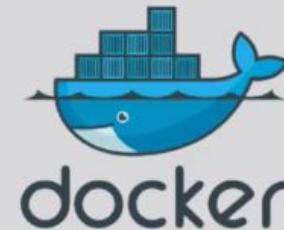


Target Endpoints and Exporters

Monitor a Linux Server?

- download a node exporter
- untar and execute
- converts metrics of the server
- exposes /metrics endpoint
- configure  to scrape this endpoint

Exporters available as
Docker images



exporter



mysql db

Alert Manager

How does Prometheus trigger the alerts?

Who receives the alerts?

etc.

Email

Slack

..reading the alert rules of config file

Prometheus Server

Retrieval

- pulls
metrics data

Storage

- stores
metrics data

HTTP Server

- accepts
queries

Data Retrieval Worker

Time Series Database

Accepts PromQL queries

push alerts

Alertmanager

notify

EFK

Elastic

Fluent

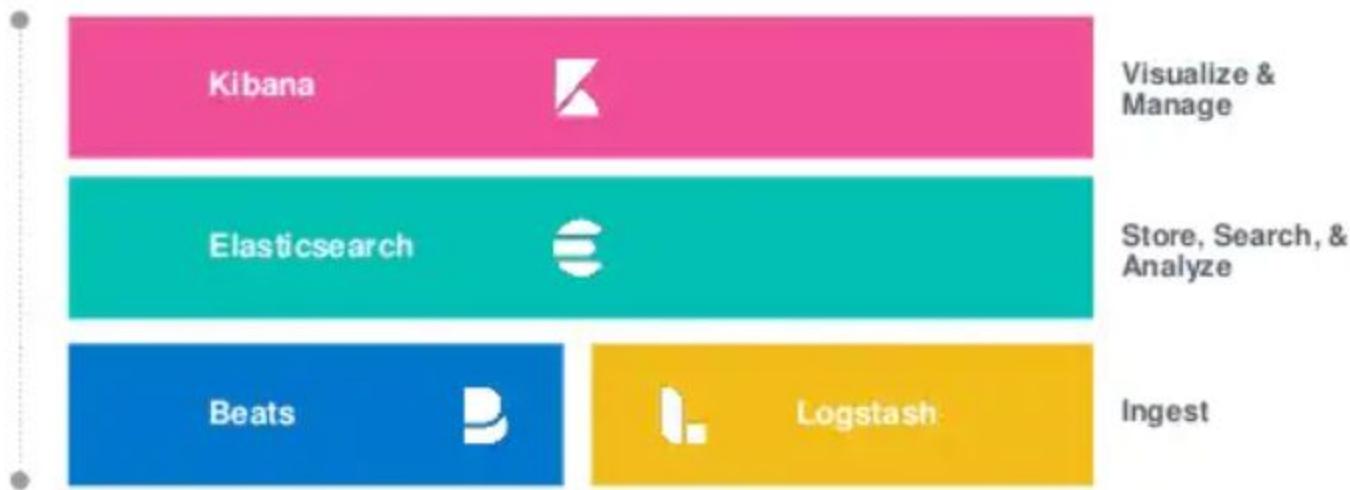
Kibana

Elastic Stack

SOLUTIONS



Elastic Stack



SaaS



Elastic cloud

SELF-MANAGED



Elastic cloud
Enterprise



Standalone



elastic

Technology **differentiation**



SCALE

Distributed by design

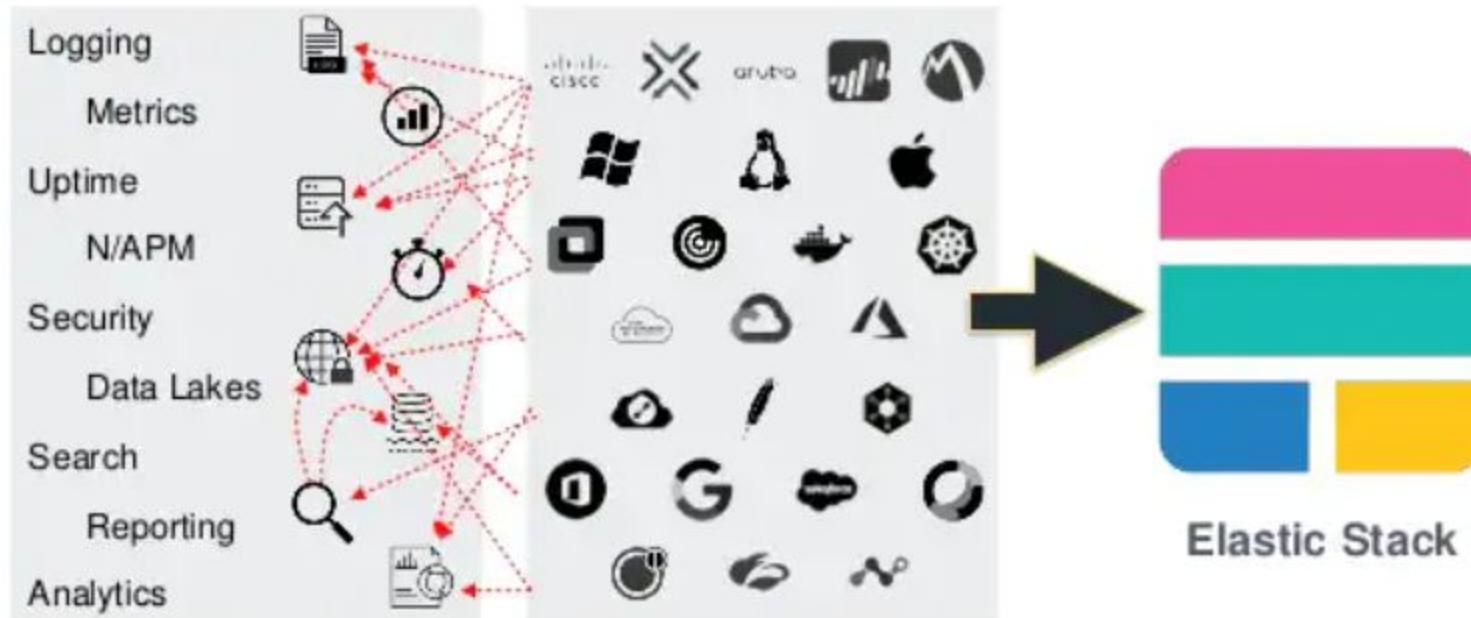
SPEED

Find matches in milliseconds

RELEVANCE

Get highly relevant results

reducing complexity



Solutions



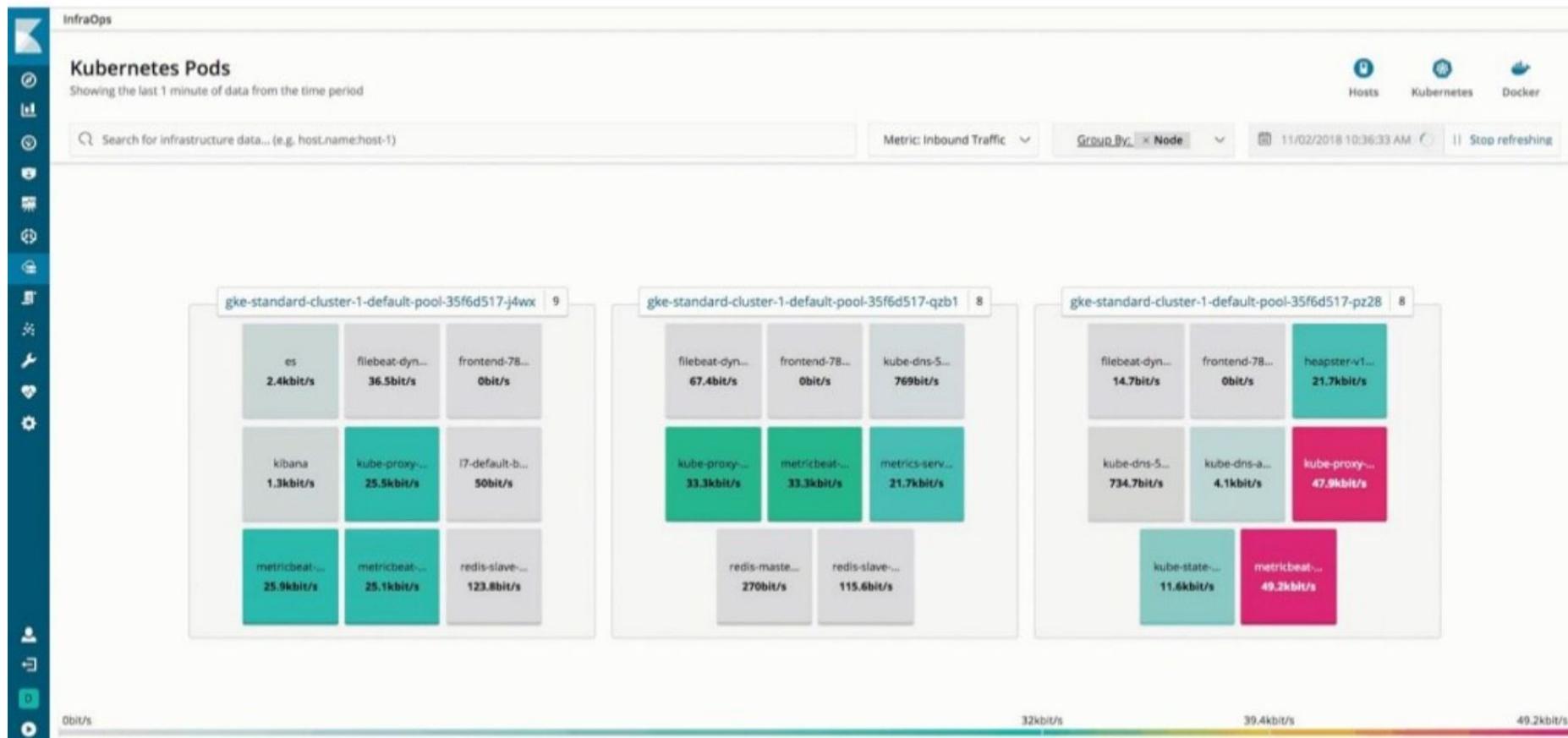
Elastic Stack



SaaS

SELF-MANAGED

Why Solutions?



Logging

Your logs, your way. At any scale, with speed.

Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

[Add APM](#)

Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

[Add log data](#)

Metrics

Collect metrics from the operating system and services running on your servers.

[Add metric data](#)

Security analytics

Centralize security events for interactive investigation in ready-to-go visualizations.

[Add security events](#)

Add sample data

Load a data set and a Kibana dashboard

Upload data from log file

Import a CSV, NDJSON, or log file

Use Elasticsearch data

Connect to your Elasticsearch index

Visualize and Explore Data

Manage and Administer the Elastic Stack

Logging

Your logs, your way. At any scale, with speed.

Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

[Add APM](#)

Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

[Add log data](#)

Metrics

Collect metrics from the operating system and services running on your servers.

[Add metric data](#)

Security analytics

Centralize security events for interactive investigation in ready-to-go visualizations.

[Add security events](#)

Add sample data

Load a data set and a Kibana dashboard

Upload data from log file

Import a CSV, NDJSON, or log file

Use Elasticsearch data

Connect to your Elasticsearch index

Visualize and Explore Data

Manage and Administer the Elastic Stack

Infrastructure Metrics

Unify your infrastructure monitoring (logs, metrics, and traces) in one place



Loading Kibana

App Search

Powerful search, built for developers



geolocal



Reset Demo

Query Tester



Overview

Analytics

Query Tester

MANAGE

Documents

Schema

API Logs

SEARCH SETTINGS

Synonyms

Curations

Relevance Tuning



ACCESS

Credentials

Documents

Filter content...

ID doc-5b73676fb5b9e1539d912ada

live_music	=> false
phone_number	=> (212) 477-8100
tags	=> Cocktail Bars
rating	=> 5
location	=> 40.7386178790617, -73.9834206895483
full_bar	=> true
created_at	=> 2018-05-24T23:36:15+00:00
street_address	=> 294 3rd Ave
updated_at	=> 2018-08-12T23:36:15+00:00
business_name	=> Ampersand
city	=> New York
business_id	=> 85775cf9-83fc-4e0f-b89b-5a4c40d99ecb
accepts_reservations	=> true

ID doc-5b73676fb5b9e1539d912ba6

live_music	=> false
phone_number	=> (415) 642-1383
tags	=> Chinese

Deployment Options



Elastic Cloud

Elastic Software as a Service, managed completely by the people who know it best.



Elastic Cloud Enterprise

Orchestration tools & integration with major cloud platforms provide the best of both worlds



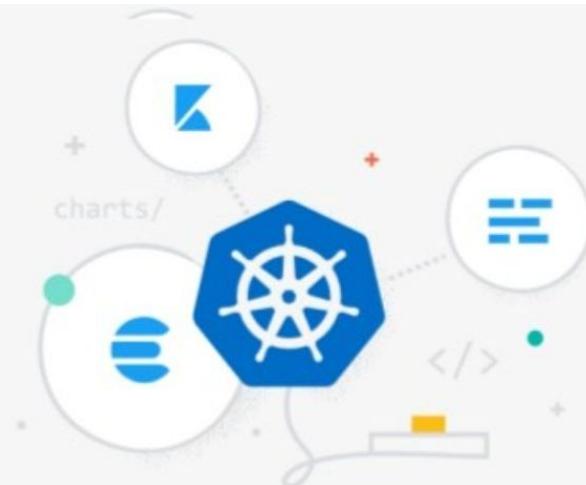
On Premise

Leverage your data centers and engineering teams to get the most from the Elastic solutions

ELASTICSEARCH & KUBECTL

Why Elasticsearch and Kubernetes?

Kubernetes has become the go-to open source choice for automating the deployment, scaling, and management of containerized architectures. Through our [commitment to cloud native technologies](#) and being where our users are, we're making it easier with Elastic Cloud on Kubernetes to deploy and run Elasticsearch and Kibana on Kubernetes.



AUTOMATE & ORCHESTRATE

Elastic Cloud on Kubernetes

Built on the Kubernetes Operator pattern, Elastic Cloud on Kubernetes extends the basic Kubernetes orchestration capabilities to support the setup and management of Elasticsearch and Kibana on Kubernetes.

With Elastic Cloud on Kubernetes, simplify the processes around deployment, upgrades, snapshots, scaling, high availability, security, and more for running Elasticsearch in Kubernetes.



What is elasticsearch?

ElasticSearch is a free and open source distributed inverted index created by Shay Banon.

Build on top of Apache Lucene

- Lucene is a most popular java-based full text search index implementation.

First public release version v0.4 in February 2010.

Developed in Java, so inherently cross-platform.

Which companies use elasticsearch?



LinkedIn

accenture
High performance. Delivered.

stackoverflow

GitHub

NETFLIX

f

The New York Times

Symantec

orange

ebay

cisco

Microsoft



Why Elasticsearch?

Easy to scale

Everything is one JSON call away (RESTful API)

Unleashed power of Lucene under the hood

Excellent Query DSL

Multi-tenancy

Support for advanced search features (Full Text)

Configurable and Extensible

Document Oriented

Schema free

Conflict management

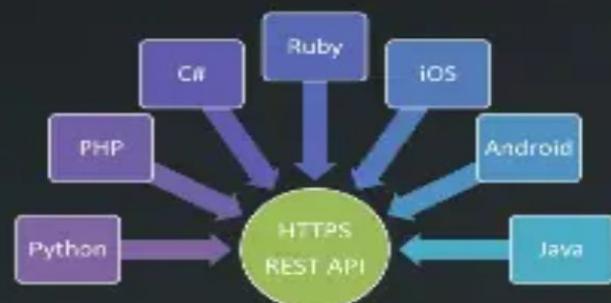
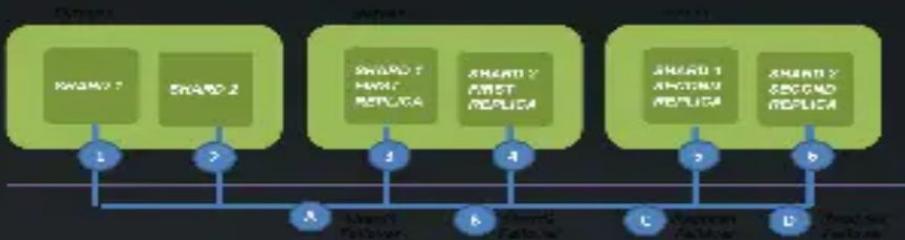
Active community



Easy to Scale

Elasticsearch allows you to start small, but will grow with your business. It is built to scale horizontally out of the box.

As you need more capacity, just add more nodes, and let the cluster reorganize itself to take advantage of the extra hardware.



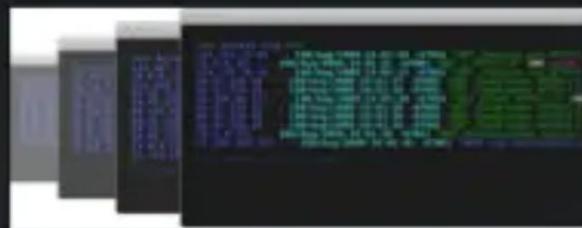
RESTful API

Elasticsearch is API driven. Almost any action can be performed using a simple RESTful API using JSON over HTTP. An API already exists in the language of your choice.

Responses are always in JSON, which is both machine and human readable.

Per-operation Persistence

Elasticsearch puts your data safety first: Document changes are recorded in transaction logs on multiple nodes in the cluster to minimize the chance of any data loss.



Excellent Query DSL

The REST API exposes a very complex and capable query DSL, that is very easy to use. Every query is just a JSON object that can practically contain any type of query, or even several of them combined.

Using filtered queries, with some queries expressed as Lucene filters, helps leverage caching and thus speed up common queries, or complex queries with parts that can be reused.

Faceting, another very common search feature, is just something that upon-request is accompanied to search results, and then is ready for you to use.

Multi-tenancy

You can host multiple indexes on one Elasticsearch installation - node or cluster. Each index can have multiple "types", which are essentially completely different indexes.

The nice thing is you can query multiple types and multiple indexes with one simple query. This opens quite a lot of options.

Multi-tenancy

```
$ curl -XPUT http://localhost:9200/kintchy  
$ curl -XPUT http://localhost:9200/elasticsearch
```

Support for advanced search features (Full Text)



Elastic | search is beautiful

Elasticsearch uses Lucene under the covers to provide the most powerful full text search capabilities available in any open source product.

Search comes with multi-language support, a powerful query language, support for geolocation, context aware did-you-mean suggestions, autocomplete and search snippets.

script support in filters and scorers

Configurable and Extensible

Many of Elasticsearch configurations can be changed while Elasticsearch is running, but some will require a restart (and in some cases, reindexing). Most configurations can be changed using the REST API too.

Elasticsearch has several extension points - namely site plugins (let you serve static content from ES - like monitoring javascript apps), rivers (for feeding data into Elasticsearch), and plugins that let you add modules or components within Elasticsearch itself. This allows you to switch almost every part of Elasticsearch if so you choose, fairly easily.

If you need to create additional REST endpoints to your Elasticsearch cluster, that is easily done as well.

Document oriented:

```
$ curl -XPUT http://localhost:9200/twitter/user/kimchy -d
{
  "name": "Stay Bacon"
}
$ curl -XPUT http://localhost:9200/twitter/tweet/1 -d '{
```

Document Oriented

Store complex real world entities in Elasticsearch as structured JSON documents. All fields are indexed by default, and all the indices can be used in a single query, to return results at breath taking speed.

Schema free

Elasticsearch allows you to get started easily. Toss it a JSON document and it will try to detect the data structure, index the data and make it searchable. Later, apply your domain specific knowledge of your data to customize how your data is indexed.

Conflict management

Optimistic version control can be used where needed to ensure that data is never lost due to conflicting changes from multiple processes.

Active community

The community, other than creating nice tools and plugins, is very helpful and supporting. The overall vibe is really great, and this is an important metric of any OSS project.

There are also some books currently being written by community members, and many blog posts around the net sharing experiences and knowledge.



Cluster :

A cluster consists of one or more nodes which share the same cluster name. Each cluster has a single master node which is chosen automatically by the cluster and which can be replaced if the current master node fails.

Node :

A node is a running instance of elasticsearch which belongs to a cluster. Multiple nodes can be started on a single server for testing purposes, but usually you should have one node per server.

At startup, a node will use unicast (or multicast, if specified) to discover an existing cluster with the same cluster name and will try to join that cluster.

Index :

An index is like a 'database' in a relational database. It has a mapping which defines multiple types.

An index is a logical namespace which maps to one or more primary shards and can have zero or more replica shards.

Type :

A type is like a 'table' in a relational database. Each type has a list of fields that can be specified for documents of that type. The mapping defines how each field in the document is analyzed.



Document:

A document is a JSON document which is stored in elastic search. It is like a row in a table in a relational database. Each document is stored in an index and has a type and an id.

A document is a JSON object (also known in other languages as a hash / hashmap / associative array) which contains zero or more fields, or key-value pairs. The original JSON document that is indexed will be stored in the _source field, which is returned by default when getting or searching for a document.

Field:

A document contains a list of fields, or key-value pairs. The value can be a simple (scalar) value (eg a string, integer, date), or a nested structure like an array or an object. A field is similar to a column in a table in a relational database.

The mapping for each field has a field 'type' (not to be confused with document type) which indicates the type of data that can be stored in that field, eg integer, string, object. The mapping also allows you to define (amongst other things) how the value for a field should be analyzed.

Mapping:

A mapping is like a 'schema definition' in a relational database. Each index has a mapping, which defines each type within the index, plus a number of index-wide settings. A mapping can either be defined explicitly or it will be generated automatically when a document is indexed.



Shard :

A shard is a single Lucene instance. It is a low-level "worker" unit which is managed automatically by Elasticsearch. An index is a logical namespace which points to primary and replica shards.

Elasticsearch distributes shards amongst all nodes in the cluster, and can move shards automatically from one node to another in the case of node failure, or the addition of new nodes.

Primary Shard :

Each document is stored in a single primary shard. When you index a document, it is indexed first on the primary shard, then on all replicas of the primary shard. By default, an index has 5 primary shards. You can specify fewer or more primary shards to scale the number of documents that your index can handle.

Replica Shard :

Each primary shard can have zero or more replicas. A replica is a copy of the primary shard, and has two purposes:

- 1) increase failover: a replica shard can be promoted to a primary shard if the primary fails.
- 2) increase performance: get and search requests can be handled by primary or replica shards.



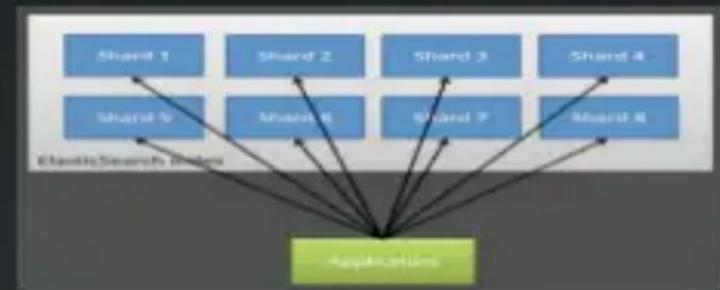
ElasticSearch Routing

All of your data lives in a primary shard, somewhere in the cluster. You may have five shards or five hundred, but any particular document is only located in one of them. Routing is the process of determining which shard that document will reside in.

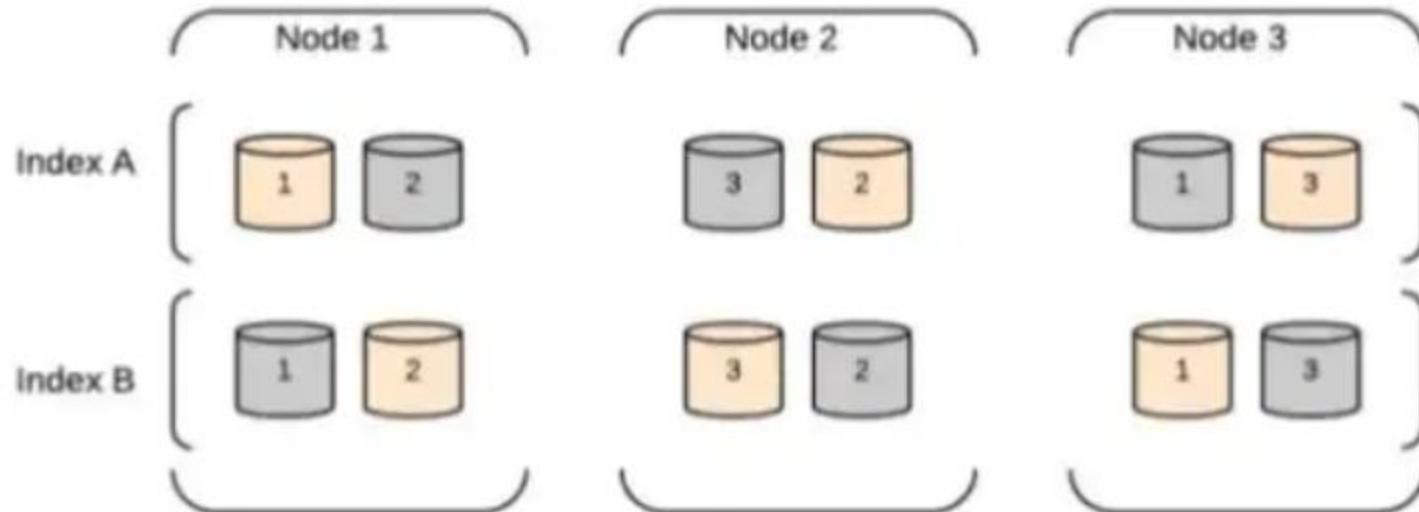
Elasticsearch has no idea where to look for your document. All the docs were randomly distributed around your cluster, so Elasticsearch has no choice but to broadcast the request to **all** shards. This is a non-negligible overhead and can easily impact performance.

Wouldn't it be nice if we could tell Elasticsearch which shard the document lived in? Then you would only have to search one shard to find the document(s) that you need.

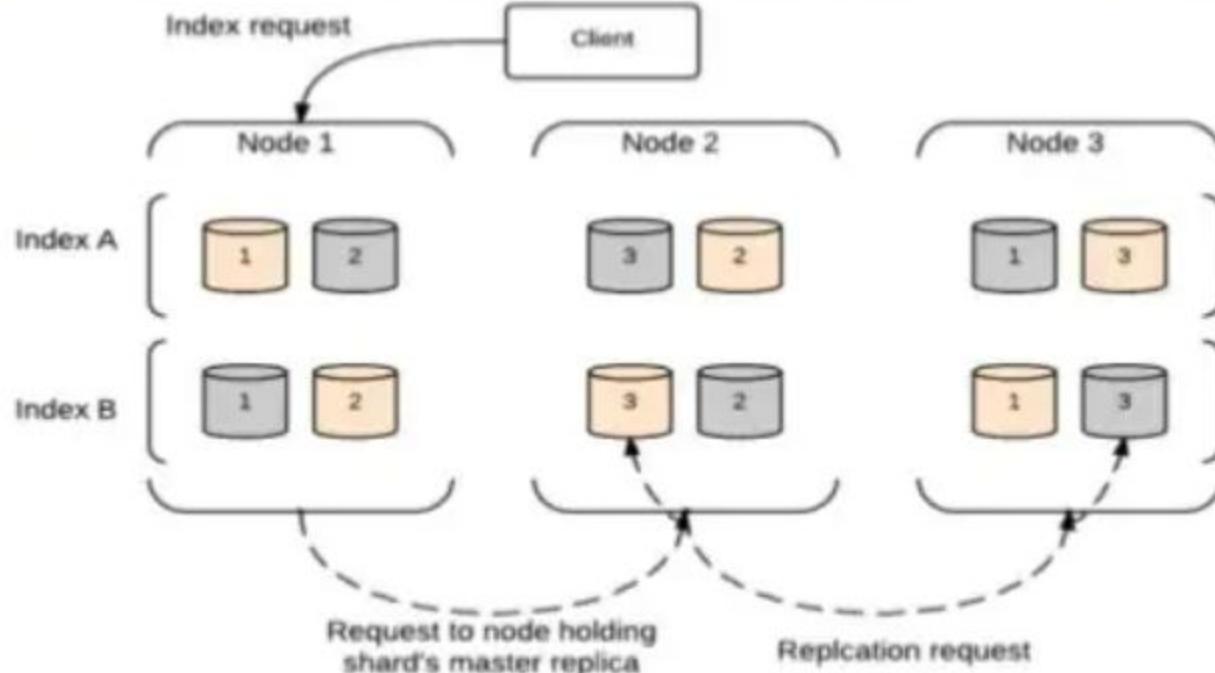
Routing ensures that all documents with the same routing value will locate to the same shard, eliminating the need to broadcast searches.



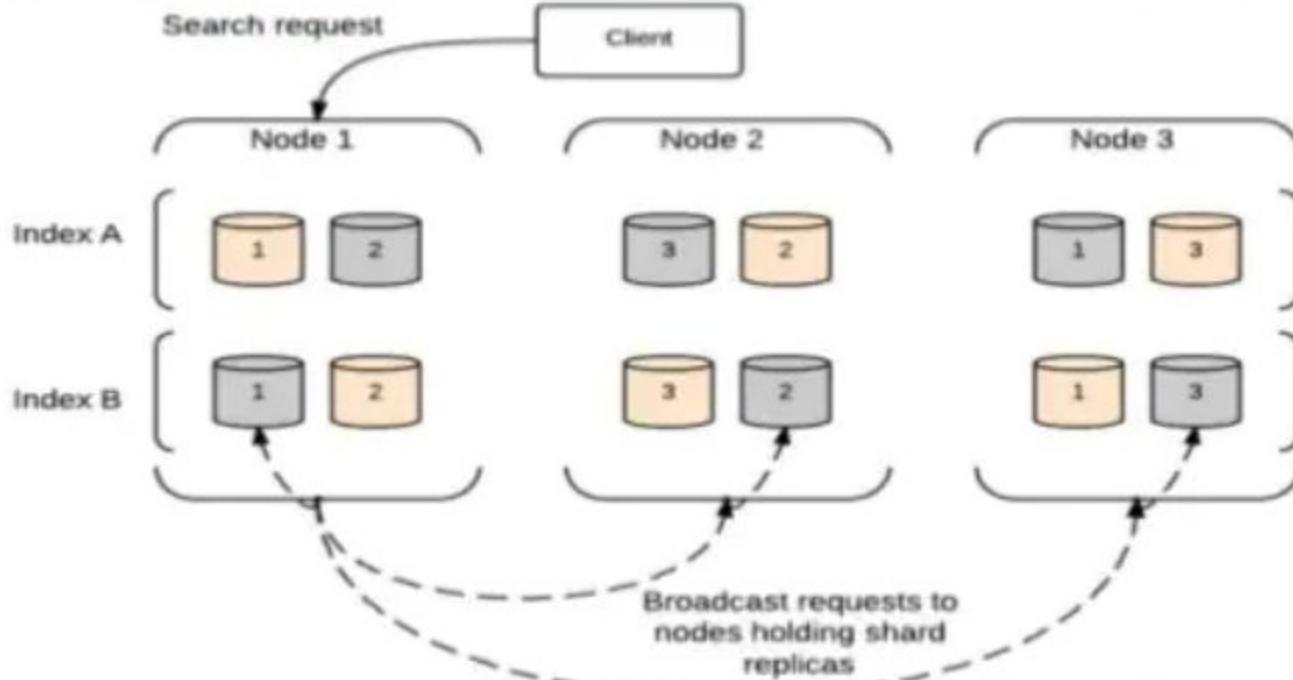
Cluster Architecture



Index Request



Search Request



Generate

Collect

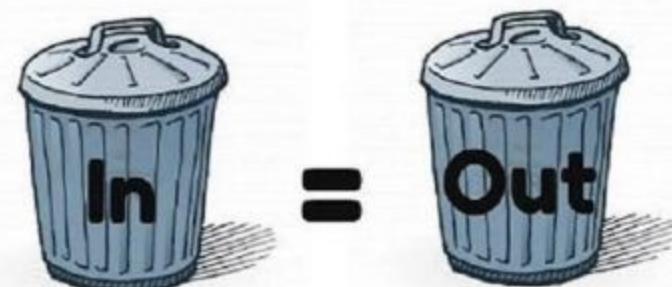
Transport

Store

Analyze

Alert

- Providing useful information, seems hard!
- Common Log Formats
 - W3C, Common Log Format, Combined Log Format
 - used for:
 - Proxy & Web Servers
- Agree upon Application Log Formats
 - Do not forget -> Log levels!
- Data security
 - Do not log passwords or privacy related data



Some seriously useful log message :)

- "No need to log, we know what is happening"
- "Something happened not sure what"
- "Empty log message"
- "Lots of sh*t happing"
- "It works b****"
- "How did we end up here?"
- "Okay i am getting tired of this error message"
- "Does this work?"
- "We hit a bug, still figuring out what"
- "Call 911 we have a problem"

Logging considerations

- Logging means more code
- Logging is not free
- Consider feedback to the UI instead of logging
- **The more you log, the less you can find**
- Consider to log only the most evil scenarios (log exceptions)
- Agree on levels like FATAL, ERROR, WARN, DEBUG, INFO, TRACE ...

LOG ALL THE THINGS



© 2013 Benji Marzi

Generate

Collect

Transport

Store

Analyze

Alert

- Syslog / Syslog-ng
- Files -> multiple places (/var/log)
 - Near realtime replication to remote destinations
- Stdout
 - Normally goes to /dev/null



In container based environments logging
to "Stdout" has the preference

Generate

Collect

Transport

Store

Analyze

Alert

- Specialized transporters and collectors available using frameworks like:
 - Logstash, Flume, Fluentd
- Accumulate data coming from multiple hosts / services
 - Multiple input sources
- Optimized network traffic
 - Pull / Push



Generate

Collect

Transport

Store

Analyze

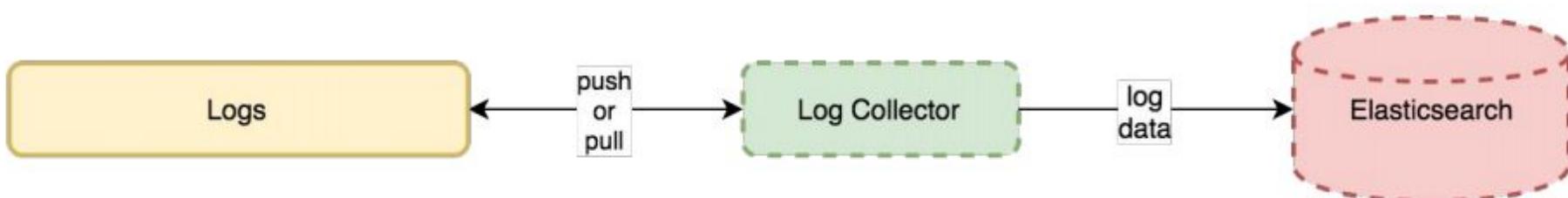
Alert

- Where should it be stored?

- Short vs Long term
- Associated costs
- Speed of data ingestion & retrieval
- Data access policies (who needs access)

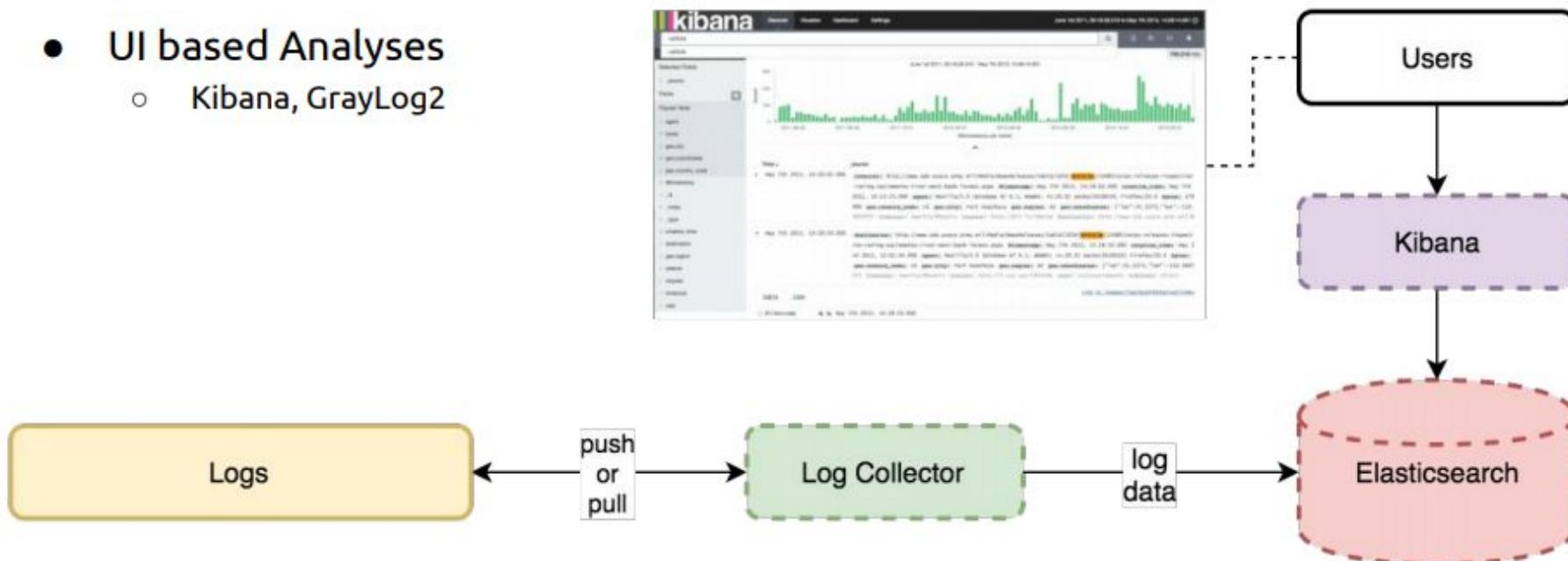
- Example storage options:

- S3, Glacier, Tape backup
- HDFS, Cassandra, MongoDB or ElasticSearch





- Batch processing of log data
 - HDFS, Hive, PIG → MapReduce Jobs
- UI based Analyses
 - Kibana, GrayLog2





```
graph LR; A[Generate] --> B[Collect]; B --> C[Transport]; C --> D[Store]; D --> E[Analyze]; E --> F[Alert]
```

- Based on patterns or “calculated” metrics → send out events
 - Trigger alert and send notifications
- Logging != Monitoring
 - Logging -> recording to diagnose a system
 - Monitoring -> observation, checking and recording

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

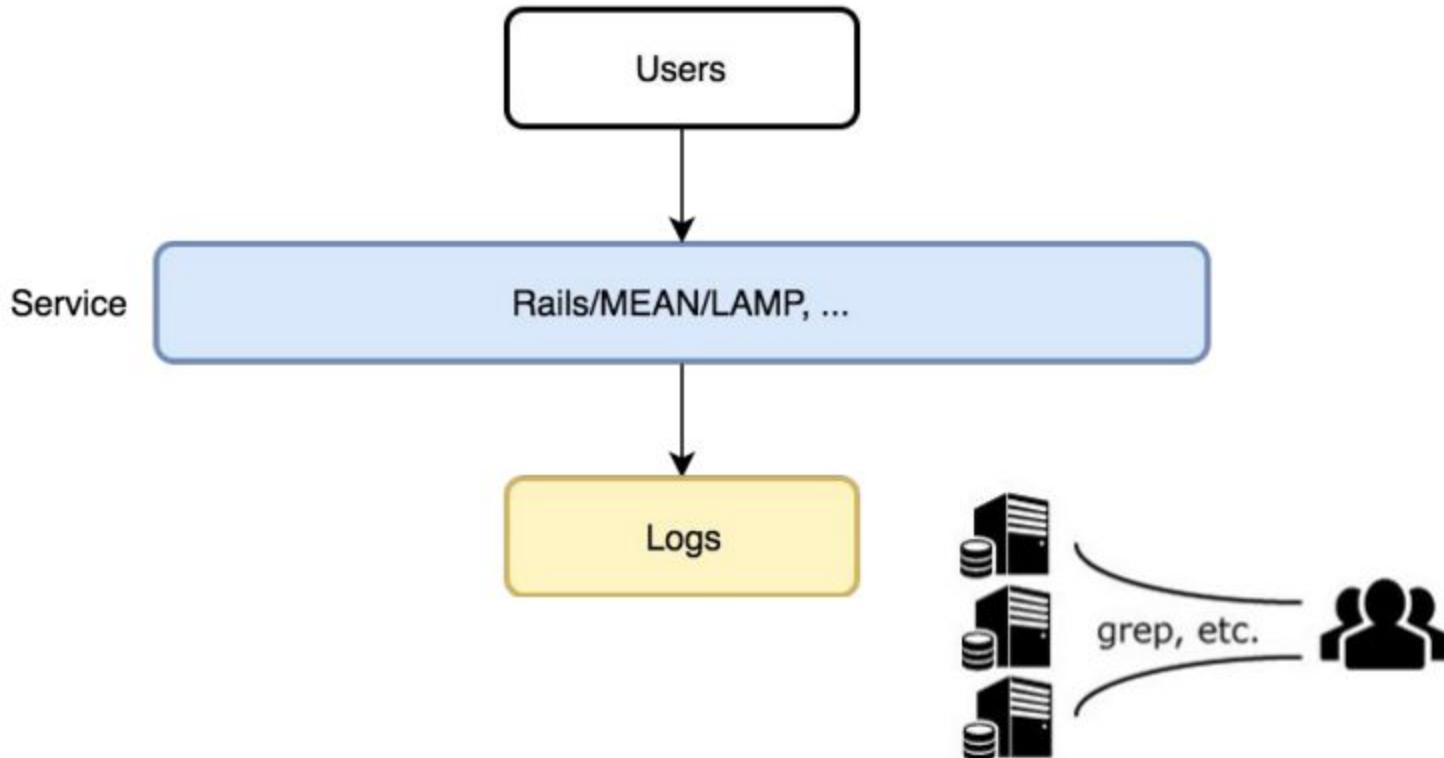
```
http_requests_total{method="post",code="200"} 1027 1395066363000
```

*“In a containerized world,
we must think differently
about logging.”*

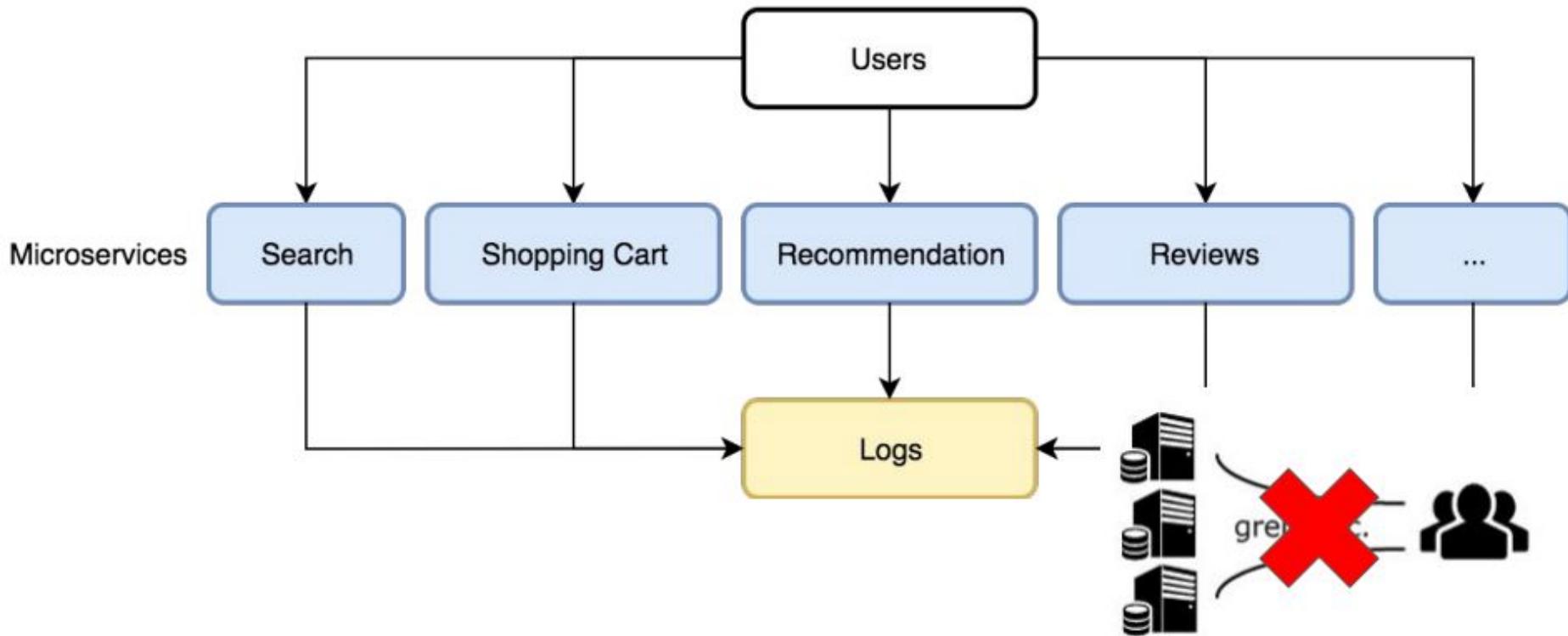
Label data at the source

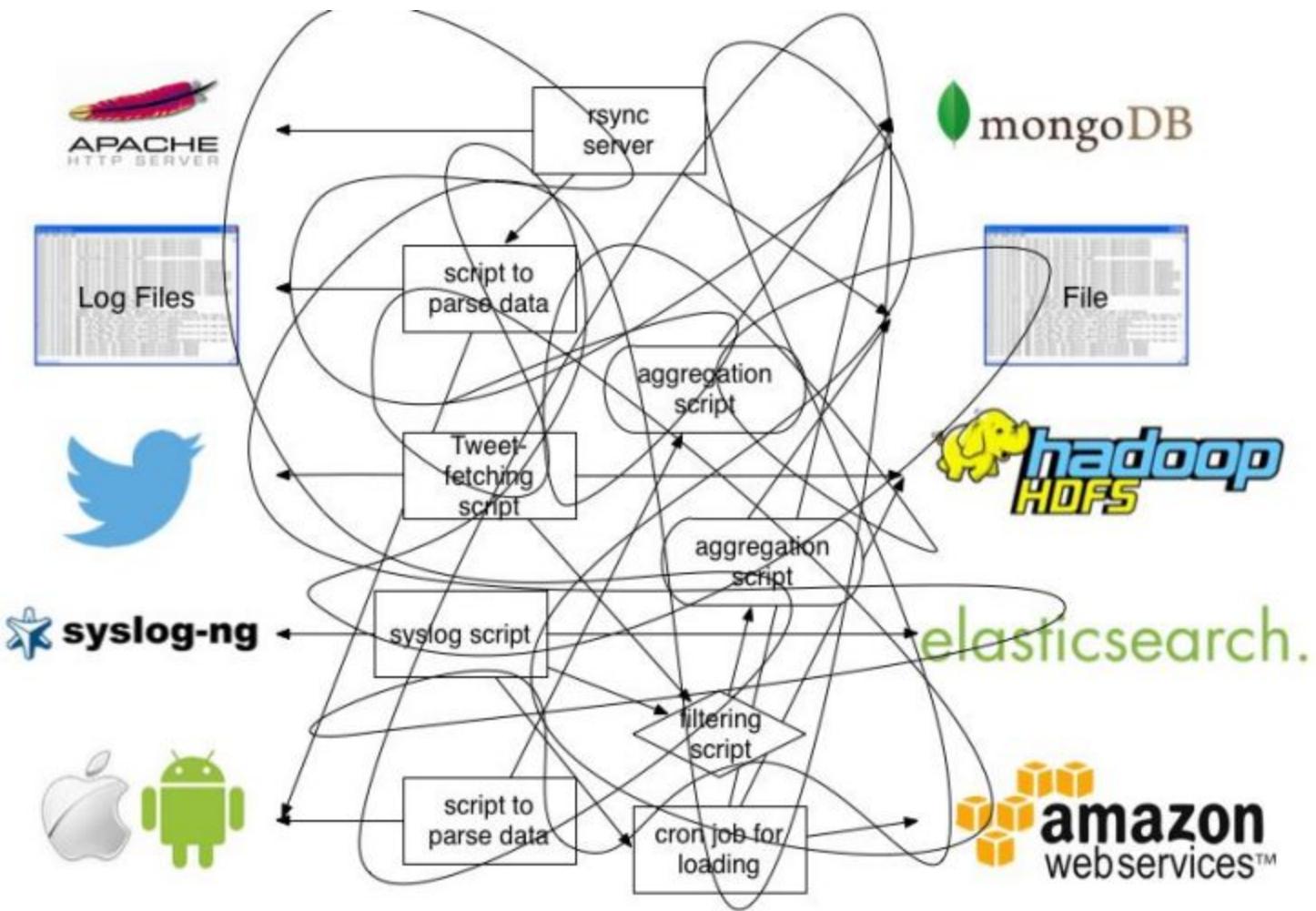
Push data and parse it as soon as possible

Logging

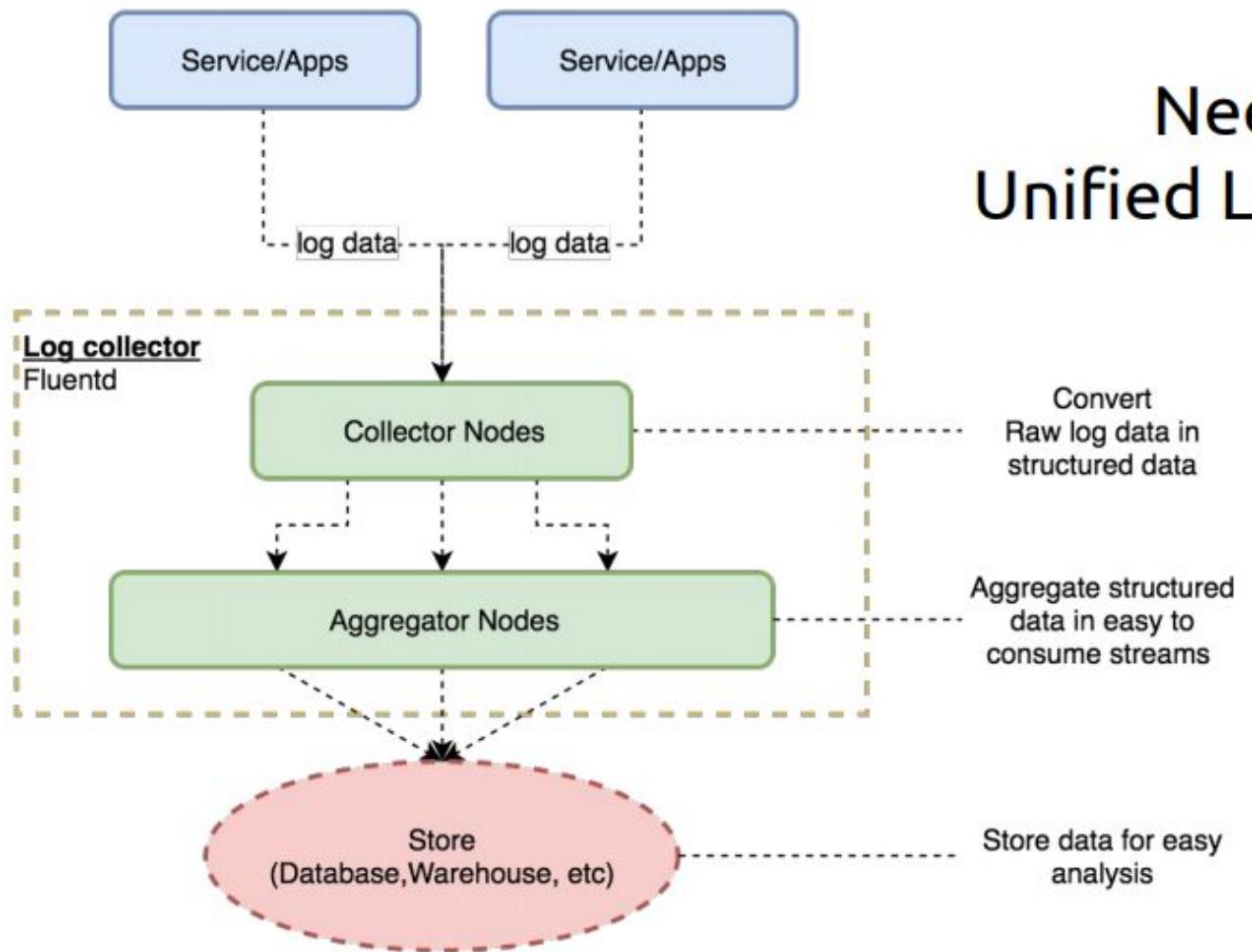


Distributed Logging





Need for a Unified Logging Layer



Access logs

Apache

App logs

Frontend

Backend

System logs

syslogd

Databases

Alerting

Nagios

Analysis

MongoDB

MySQL

Hadoop

Archiving

Amazon S3



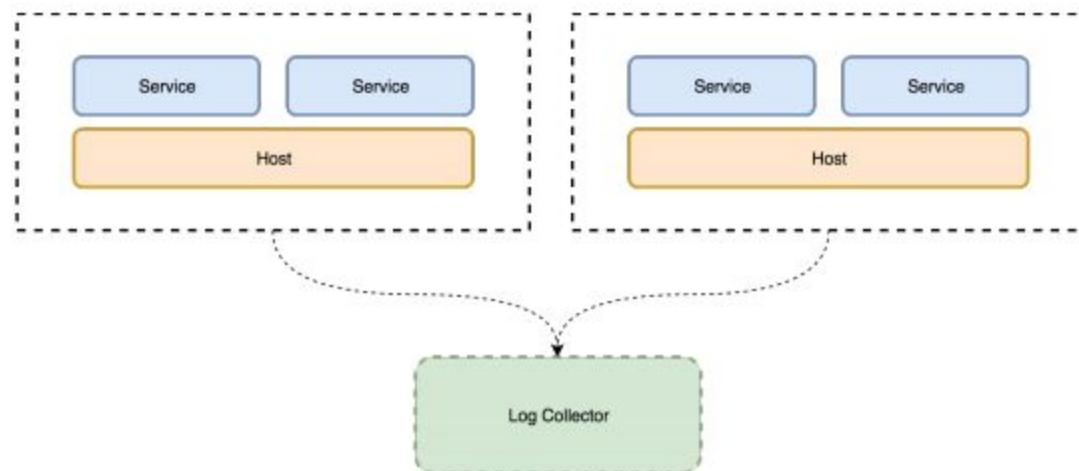
fluentd



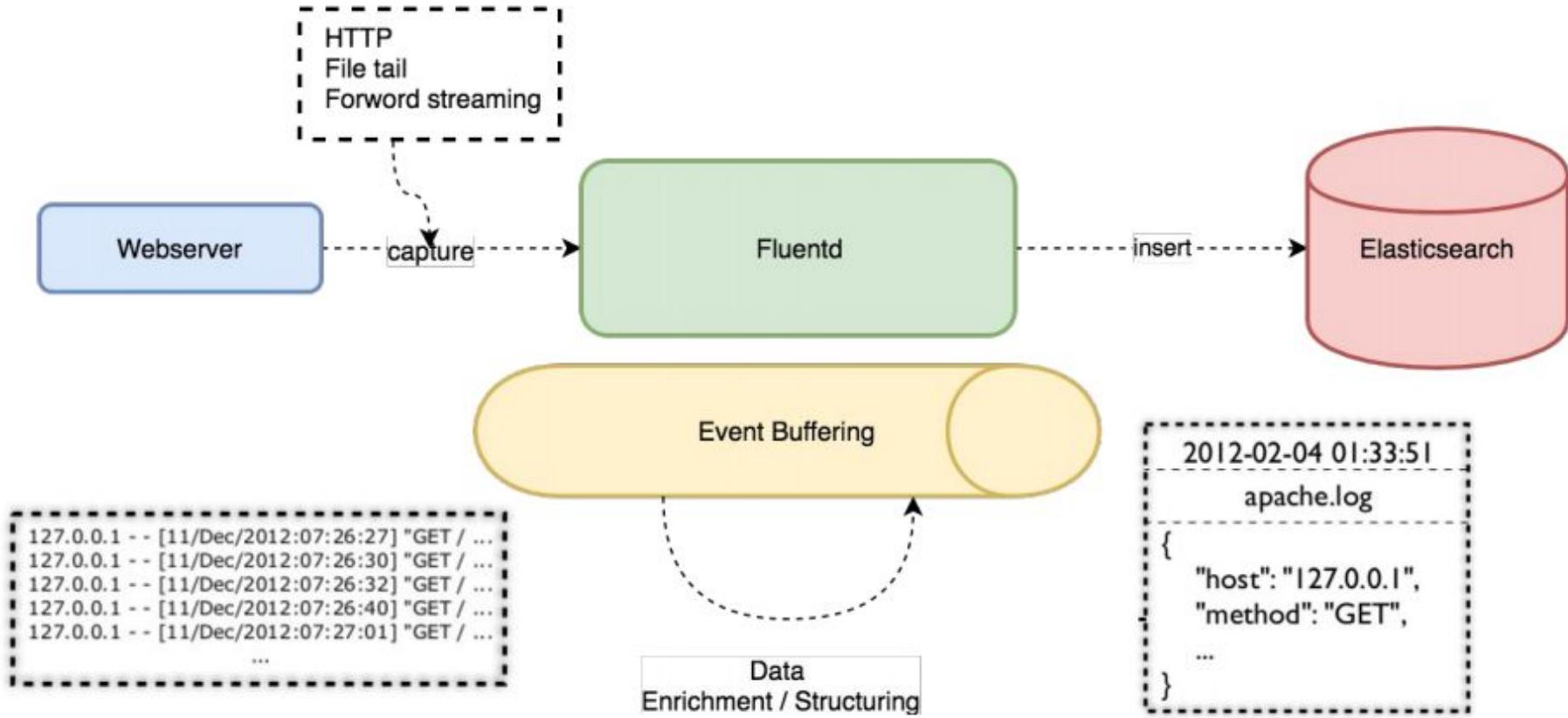
filter / buffer / routing

Fluentd

- Open source log collector written in Ruby
- Reliable, scalable and easy to extend
 - Pluggable architecture
 - Rubygem ecosystem for plugins
- Reliable log forwarding

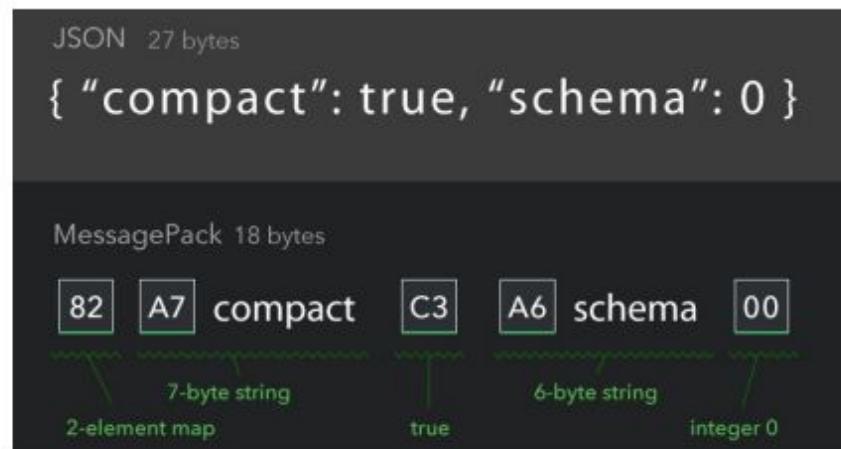


Example



Event structure

- Tag
 - Where an event comes from, used for message routing
- Time
 - When an event happens, Epoch time
 - Parsed time coming from the datasource
- Record
 - Actual log content being a JSON object
 - Internally MessagePack



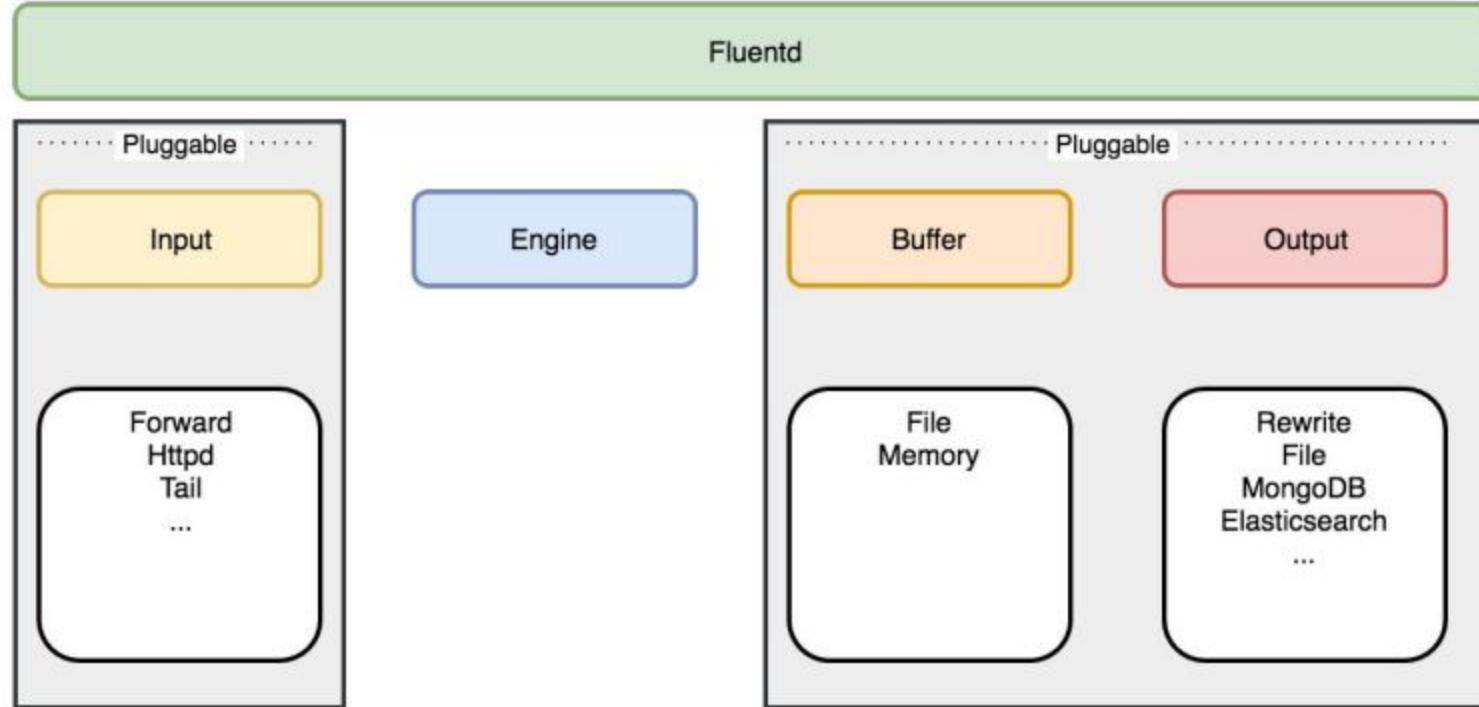
Event example

```
192.168.0.1 - - [28/Feb/2013:12:00:00 +0900] "GET / HTTP/1.1" 200 777
```



```
tag:: apache.access # set by configuration
time: 1362020400    # 28/Feb/2013:12:00:00 +0900
record: {"user": "-", "method": "GET", "code": 200, "size": 777, "host": "192.168.0.1", "path": "/"}
```

Pluggable Architecture



<http://www.fluentd.org/plugins>

Configuration

- Driven by a simple text based configuration file
 - fluent.conf

```
<source><source/>
```

→ Tell where the data comes from (input)

```
<match></match>
```

→ Tell fluentd what to do (output)

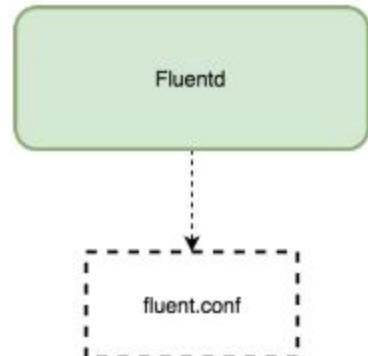
```
<filter></filter>
```

→ Event processing pipeline

```
<label></label>
```

→ Groups filter and output for internal routing

```
source -> filter 1 -> ... -> filter N -> output
```



```
# receive events via HTTP

<source>
  @type http
  port 9880
</source>
```

```
# read logs from a file

<source>
  @type tail
  path /var/log/httpd.log
  format apache
  tag apache.access
</source>
```

```
# save alerts to a file

<match alert.*>
  @type file
  path /var/log/fluent/alerts
</match>
```

```
# save access logs to MongoDB

<match apache.access>
  @type mongo
  database apache
  collection log
</match>
```

```
# forward other logs to servers

<match **>
  type forward
  <server>
    host 192.168.0.11
    weight 20
  </server>
  <server>
    host 192.168.0.12
    weight 60
  </server>
</match>
```

```
# add a field to an event  
  
<filter myapp.access>  
  @type record_transformer  
  
  <record>  
    host_param "#{Socket.gethostname}"  
  </record>  
</filter>
```

```
# grouping and internal routing  
  
<source>  
  @type forward  
  port 24224  
  bind 0.0.0.0  
  @label @SYSTEM  
</source>  
  
<label @SYSTEM>  
  <filter var.log.middleware.**>  
    @type grep  
    # ...  
  </filter>  
  <match **>  
    @type s3  
    # ...  
  </match>  
</label>
```

Generate Collect Transport Store Analyze Alert

SpringBoot
Apache Httpd

Service/Apps

Structure
data



Collector Nodes

Aggregator Nodes

RAW
log data

mongoDB

elasticsearch

STRUCTURED
log data

Store
(Database, Warehouse, etc)

query

Visualize

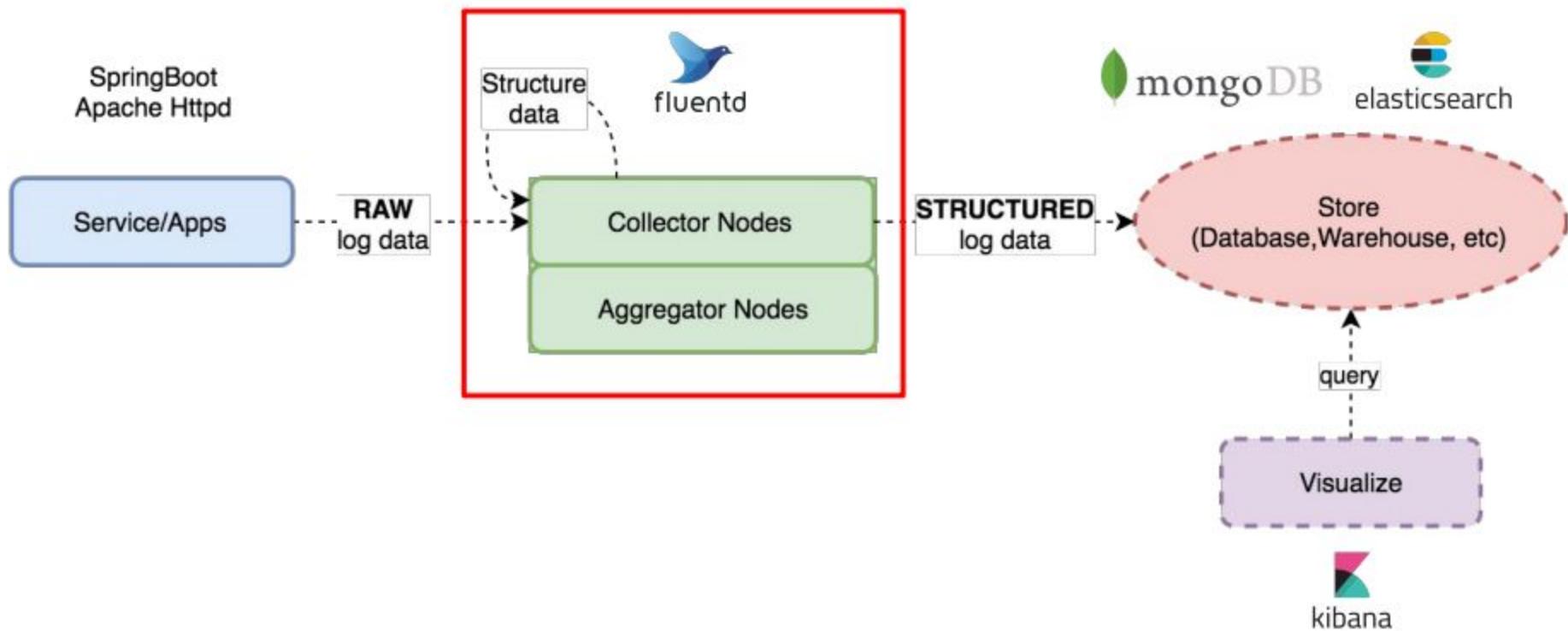


Collect & Transport

Demo Setup

Store & Analyze

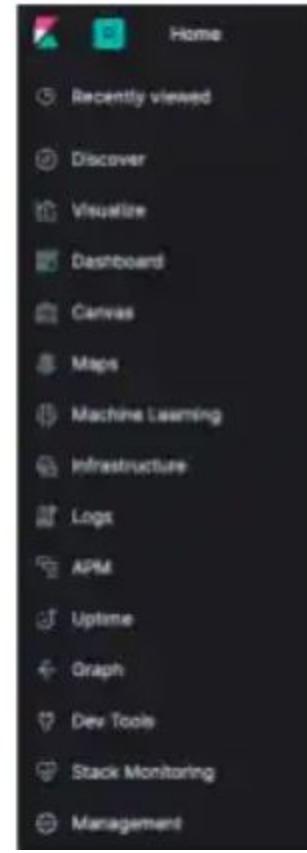
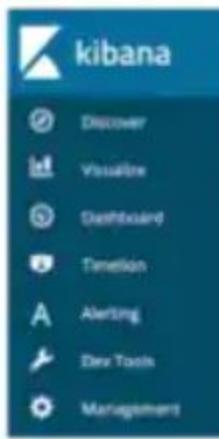
Demo: Run Fluentd



Users! What does your Kibana look like?



Users! What does your Kibana look like?



Application Performance Monitoring (APM)

Amplify your observability with application transaction and tracing

APM / Services

APM feedback Auto-refresh Last 24 hours

APM

Setup Instructions

Search transactions and errors... (E.g. transaction.duration.us > 300000 AND context.response.status_code >= 400)

Services Traces

Name ↑	Agent	Avg. response time	Trans. per minute	Errors per minute
apm-server	go	980 ms	252.5 tpm	0 err.
opbeans-go	go	27 ms	81.2 tpm	0.3 err.
opbeans-java	java	49 ms	157.7 tpm	14.6 err.
opbeans-node	nodejs	25 ms	158.5 tpm	9.9 err.
opbeans-python	python	453 ms	146.4 tpm	11.8 err.
opbeans-ruby	ruby	19 ms	104.0 tpm	11.5 err.





Spinnaker

Cloud Native Continuous Delivery
Fast, safe, repeatable deployments for every enterprise

Who should use Spinnaker?

Spinnaker provides application management and deployment to help you release software changes with high velocity and confidence. Spinnaker is an open-source, multi-cloud continuous delivery platform that combines a powerful and flexible pipeline management system with integrations to the major cloud providers. If you are looking to standardize your release processes and improve quality, Spinnaker is for you.

Supported Cloud Providers



Google Cloud Platform



kubernetes

ORACLE®

Cloud Infrastructure

Spinnaker is an open source, multi-cloud continuous delivery platform for releasing software changes with high velocity and confidence.

Created at Netflix, it has been battle-tested in production by hundreds of teams over millions of deployments. It combines a powerful and flexible pipeline management system with integrations to the major cloud providers.

Automated Releases

Create deployment pipelines that run integration and system tests, spin up and down server groups, and monitor your rollouts. Trigger pipelines via git events, Jenkins, Travis CI, Docker, CRON, or other Spinnaker pipelines.

Built-in Deployment Best Practices

Create and deploy immutable images for faster rollouts, easier rollbacks, and the elimination of hard to debug configuration drift issues. Leverage an immutable infrastructure in the cloud with built-in deployment strategies such as red/black and canary deployments.

Multi-Cloud

Deploy across multiple cloud providers including AWS EC2, Kubernetes, Google Compute Engine, Google Kubernetes Engine, Google App Engine, Microsoft Azure, Openstack, Cloud Foundry, and Oracle Cloud Infrastructure, with DC/OS coming soon.

Features List

Role-based Access Control

Restrict access to projects or accounts by hooking into your internal authentication system using OAuth, SAML, LDAP, X.509 certs, Google Groups, Azure Groups, or GitHub Teams.

CI Integrations

Listen to events, collect artifacts, and trigger pipelines from Jenkins or Travis CI. Triggers via git, cron, or a new image in a docker registry are also supported.

Chaos Monkey Integration

Test that your application can survive instance failures by terminating them on purpose.

CLI for Setup and Admin

Install, configure, and update your Spinnaker instance with halyard, Spinnaker's CLI tool.

Features List

Deployment Strategies

Configure pipelines with built-in deployment strategies such as highlander and red/black, with rolling red/black and canary in active development, or define your own custom strategy.

Restricted Execution Windows

Restrict the execution of stages to certain windows of time, making sure deployments happen during off-peak traffic or when the right people are on hand to monitor the roll-out.

Manual Judgments

Require a manual approval prior to releasing an update with a manual judgement stage.

Monitoring Integrations

Tie your releases to monitoring services Datadog, Prometheus, Stackdriver, SignalFx, or New Relic using their metrics for canary analysis.

Notifications

Set up event notifications for email, Slack, HipChat, or SMS (via Twilio).

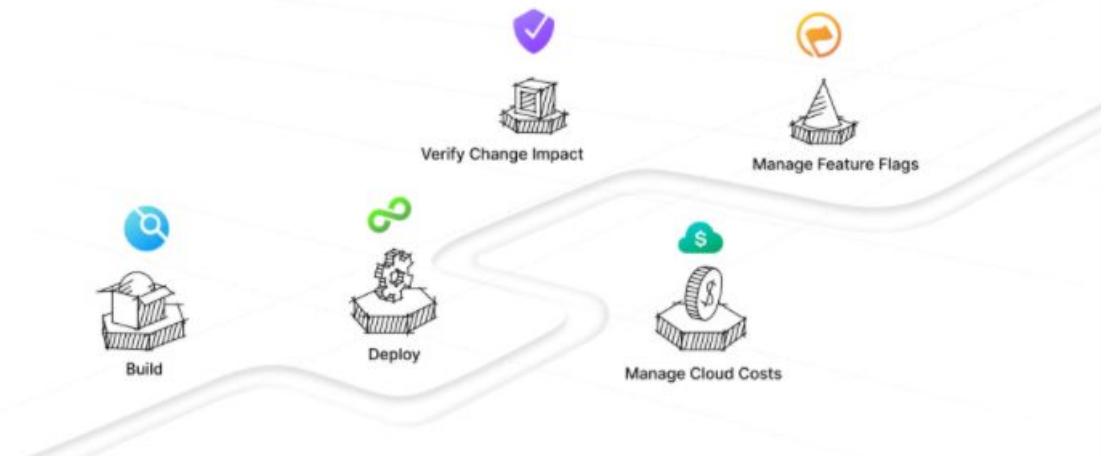
VM Bakery

Bake immutable VM images via Packer, which comes packaged with Spinnaker and offers support for Chef and Puppet templates.

The Modern Software Delivery Platform

Loved by Developers

Trusted by Businesses



Intelligent Software Delivery.

Empower engineers to deliver software on-demand.

Module

Continuous Delivery

Allow engineers to deploy on-demand to production using fast, repeatable and safe pipelines.

Build Blue/Green and Canary Deployments in Mins

Build, template, and scale complex deployments across engineering teams using a visual builder, or native GitOps experience.

Automated Deployment Verification and Rollback

Auto-detect performance and quality regressions for every deployment, and rollback in seconds to the previous working version.

Supports Cloud-Native and Traditional Applications

Harness supports containers, Helm, Kubernetes, and ECS for microservices, along with Serverless and 'vintage' stacks like WebLogic and WebSphere.

Click the ring to explore



or [Learn More >](#)

Intelligent Software Delivery.

Empower engineers to deliver software on-demand.

Module

Continuous Integration

Self-service cloud-native CI without the engineering TOIL of scripts, dependencies and maintenance.

Native GitOps

Easily build new CI pipelines with a graphical editor, no experience required.

Visual Pipeline Builder

Build and visualize complex CI pipelines in minutes, no experience required.

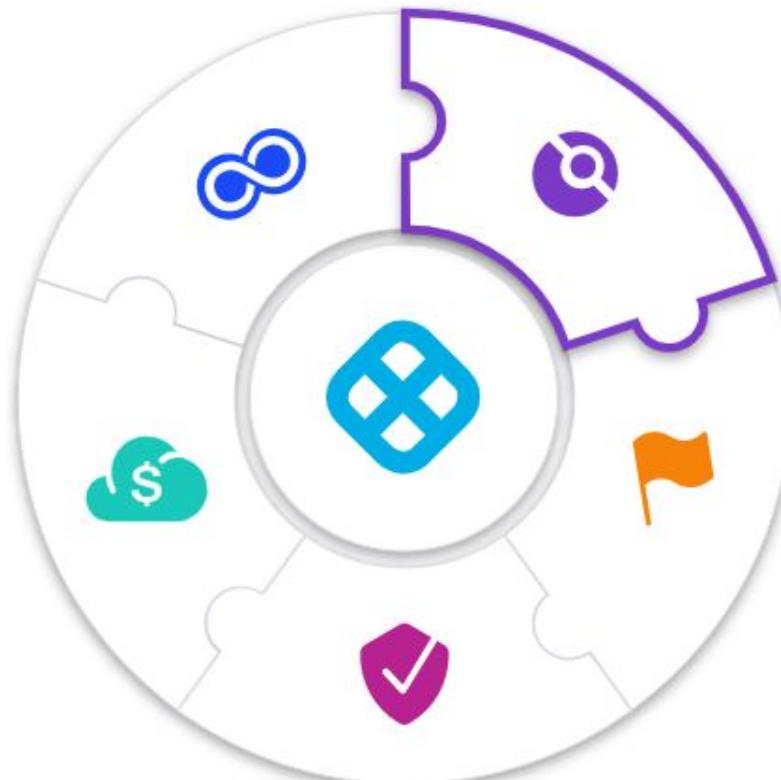
AI/ML Optimized Pipelines

Automate build and test cycles by 80% so you spend less time waiting and more time coding.

Try Free

or [Learn More >](#)

Click the ring to explore



Intelligent Software Delivery.

Empower engineers to deliver software on-demand.

Module

Continuous Features

Empower product development teams to quickly release new features, with minimal risk. Test with a specified subset, or roll out to all customers. The choice is yours.

Feature Workflows

Schedule flag rollouts, send for approval, and verify feature health with selected target audiences.

Feature Verifications

Ensure flags don't impact performance and end-user experience, before a full rollout.

Feature Canvas

Visibility into software delivery, from code merge to deployment to rollout, in a single consolidated view.

Click the ring to explore



Try Free

or [Learn More >](#)

Module

Continuous Verification

85% of failures are due to change. In minutes, Harness tells you which changes led to failure so you can immediately repair your builds and deployments.

Change Detection

Auto-detects every change and correlates them with failures in CI/CD pipelines.

AI/ML-Driven Deployment Verification

Intelligent analysis using data from your monitoring, APM and log tools.

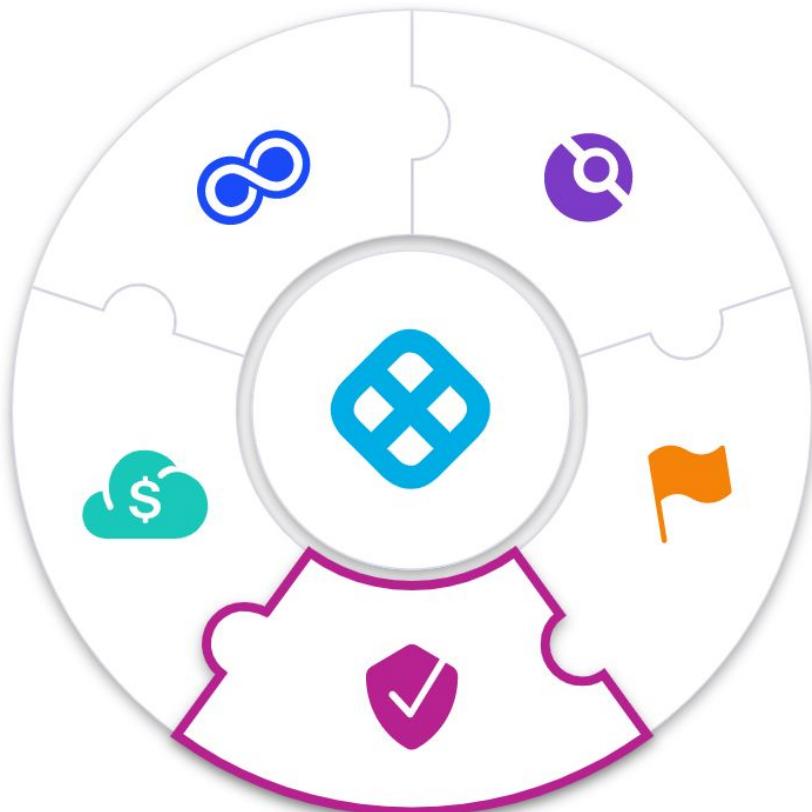
Popular Integrations

Analyzes data from most popular monitoring and observability solutions.

Demo

or [Learn More >](#)

Click the ring to explore



Module

Cloud Cost Management

Analyze and correlate cloud costs by utilized, idle, and unallocated resources, as well as changes like software deployments, config changes, and autoscaling.

Cloud & Container Cost Visibility

Unique cloud cost visibility by application, microservice, cluster, namespace, and more.

Cloud Cost Event Correlation

Correlate software deployments, config changes, and autoscaling with changes cloud cost.

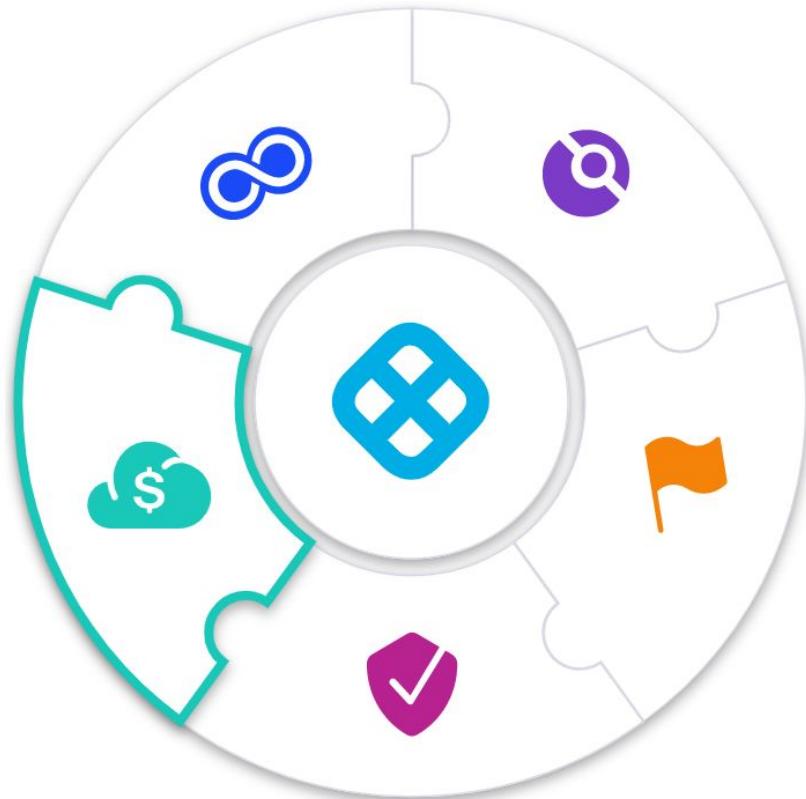
No Tagging Required

Detailed cost analysis without manual tagging - better visibility, less developer toil.

Try Free

or [Learn More >](#)

Click the ring to explore



Harness

Create a free account

<https://harness.io/>

Istio

Simplify observability, traffic management, security, and policy with the leading service mesh.

Istio addresses the challenges developers and operators face with a distributed or microservices architecture.

Whether you're building from scratch or migrating existing applications to cloud native, Istio can help.

Istio

Your app normally has:

- API server, needs securing with TLS
- API client, needs retries
- Load balancer to send traffic to the right place depending on policies
- Authentication and authorization
- Circuit breaking
- Monitoring instrumentation (Prometheus)

Istio

Istio is an open source service mesh that layers transparently onto existing distributed applications. Istio's powerful features provide a uniform and more efficient way to secure, connect, and monitor services. Istio is the path to load balancing, service-to-service authentication, and monitoring – with few or no service code changes. Its powerful control plane brings vital features, including:

- Secure service-to-service communication in a cluster with TLS encryption, strong identity-based authentication and authorization
- Automatic load balancing for HTTP, gRPC, WebSocket, and TCP traffic
- Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection
- A pluggable policy layer and configuration API supporting access controls, rate limits and quotas
- Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress

How it Works

Istio has two components: the data plane and the control plane.

The data plane is the communication between services. Without a service mesh, the network doesn't understand the traffic being sent over, and can't make any decisions based on what type of traffic it is, or who it is from or to.

Service mesh uses a proxy to intercept all your network traffic, allowing a broad set of application-aware features based on configuration you set.

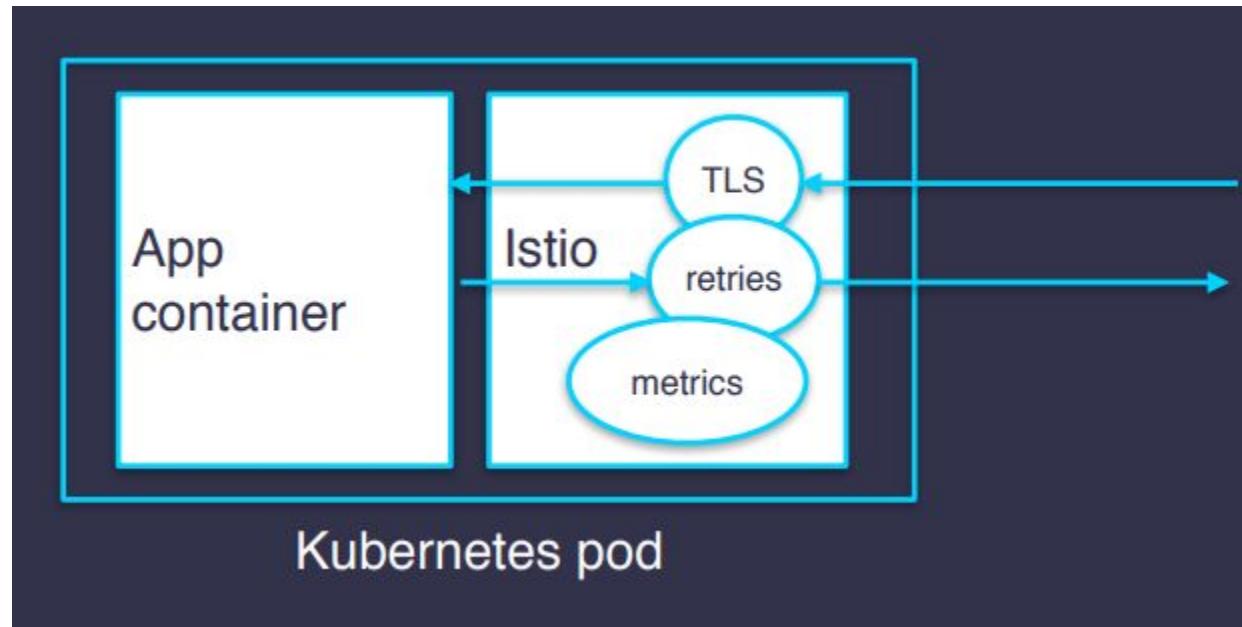
An Envoy proxy is deployed along with each service that you start in your cluster, or runs alongside services running on VMs.

The control plane takes your desired configuration, and its view of the services, and dynamically programs the proxy servers, updating them as the rules or the environment changes.

Service mesh

Service mesh says:

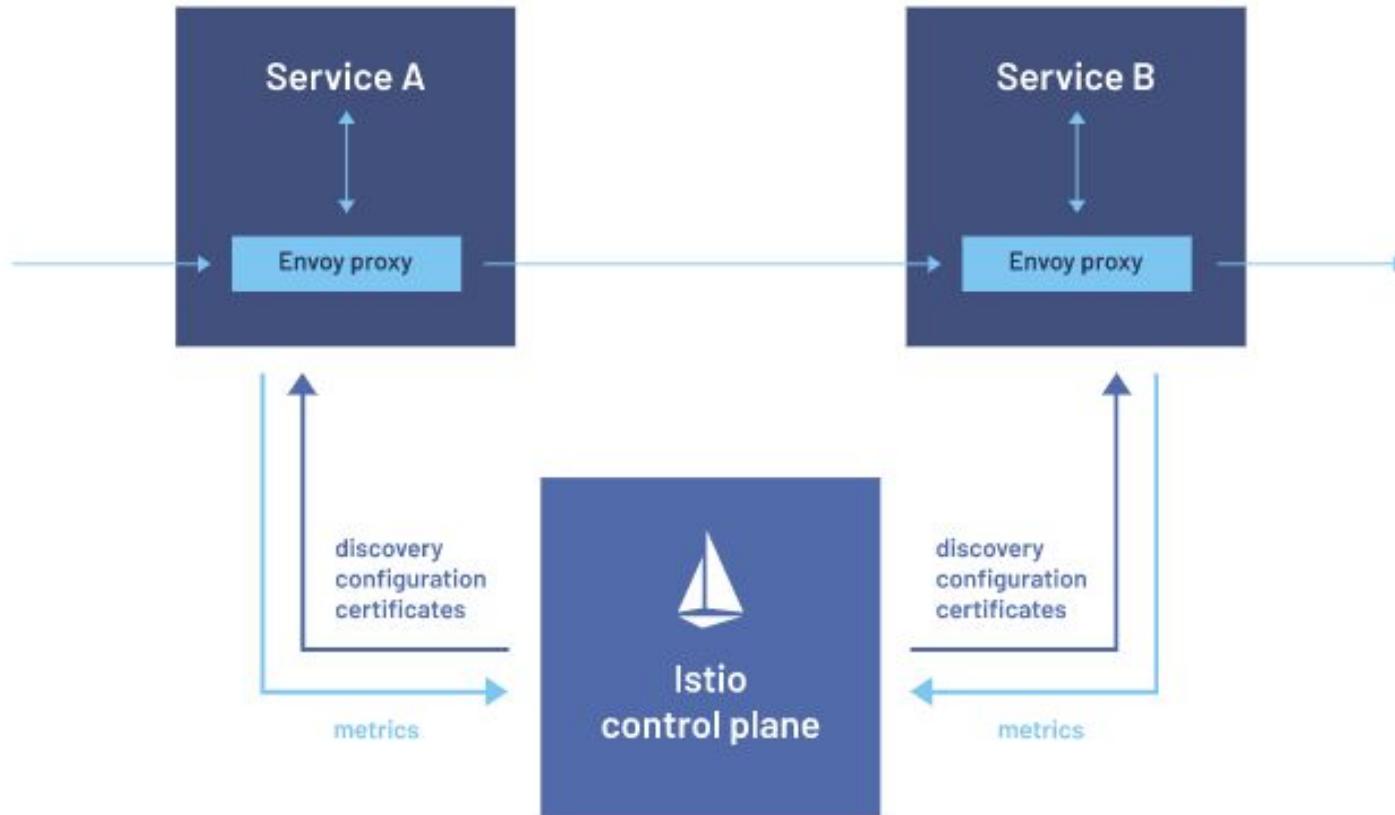
- Move this into a sidecar!

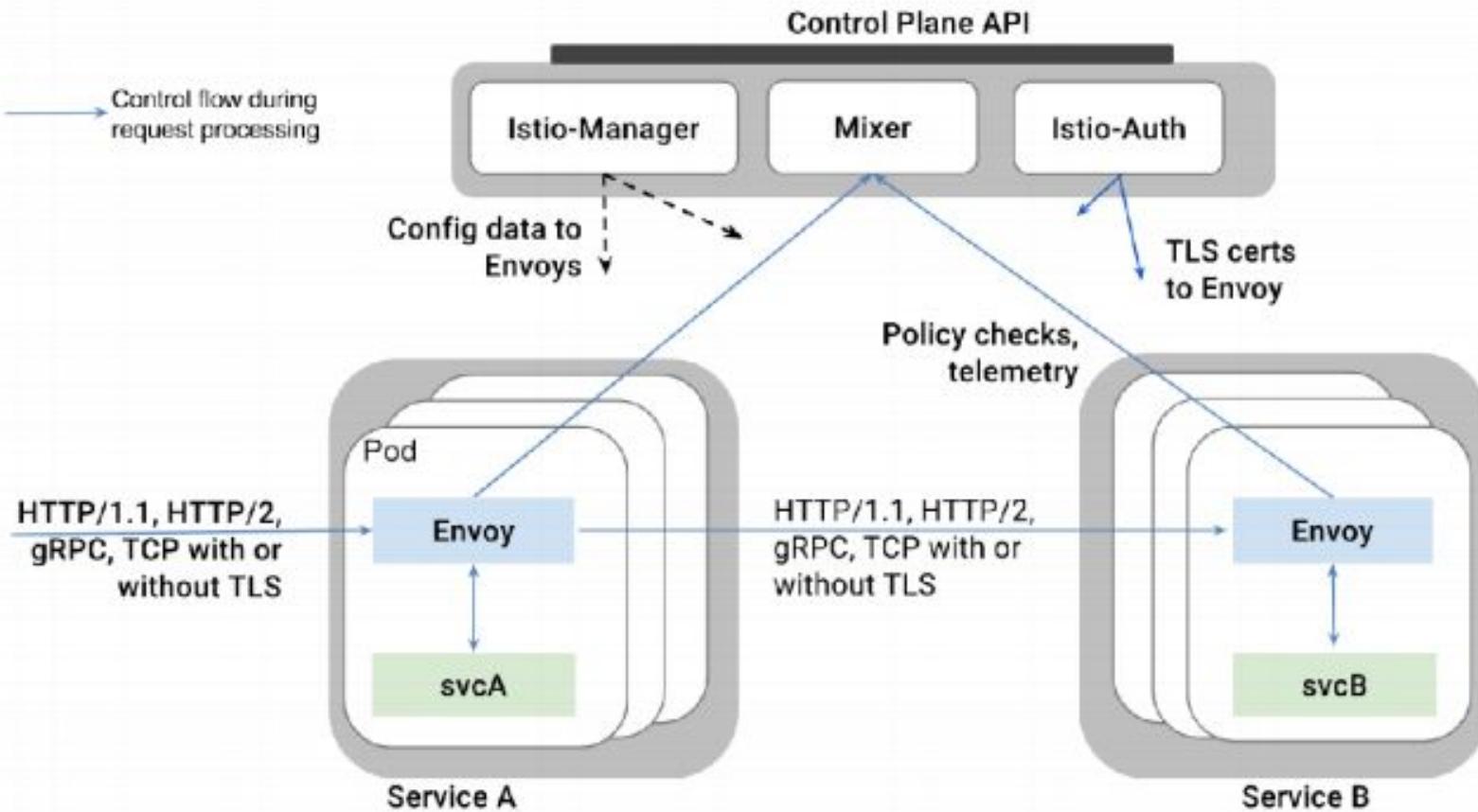


Service mesh

Do it well once, rather than over and over
again

- Save effort!
- Microservices are polyglot
- Don't want to have to rewrite all that logic in N different languages





Istio Architecture

Istio

<https://www.katacoda.com/courses/istio/deploy-istio-on-kubernetes>