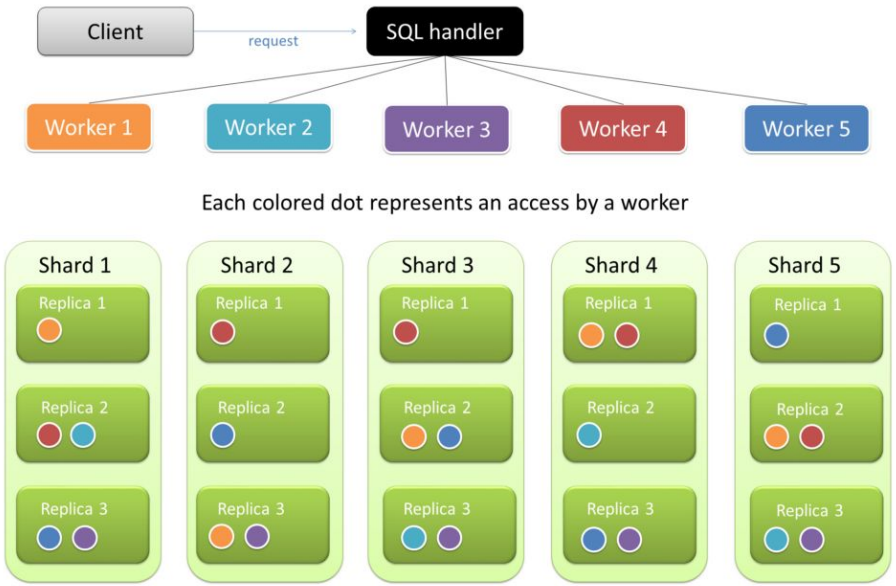
Elastic Stack

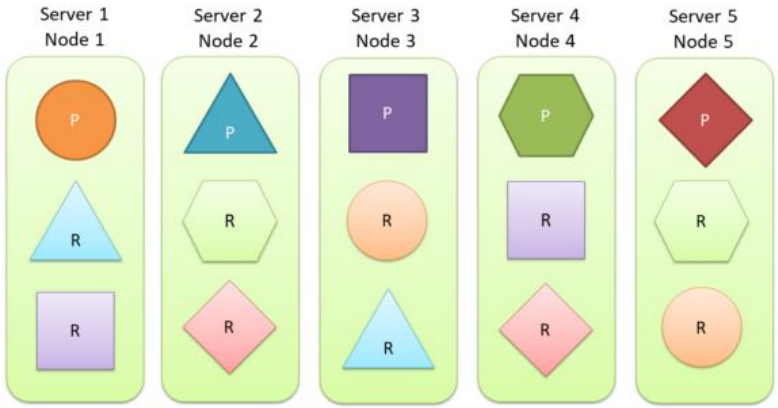
Elasticsearch, Logstash, Kibana, Beats

Elasticsearch

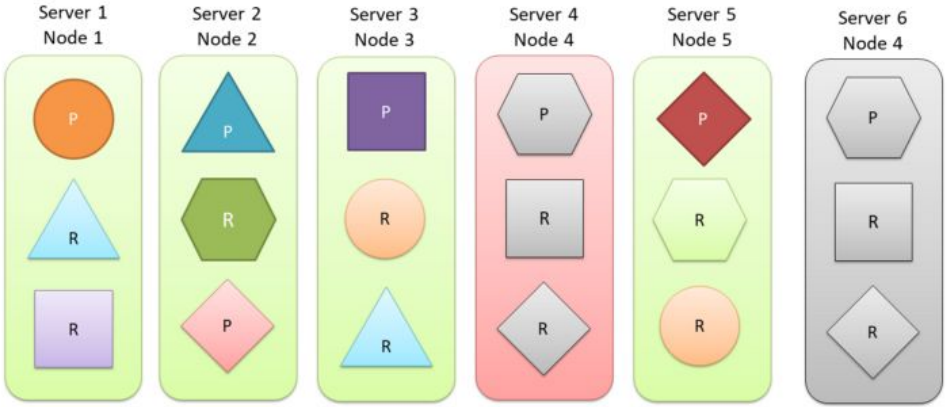
- Search and analytics engine (Built on Apache Lucerne)
- Document-style, schema-free (JSON) storage accessible via REST API
- Sharded horizontally (rows)
- Replicated
- Horizontally scaling
- Resilient/HA
- Distributed
- Multi-shard queries by workers (figure) - coordinated, multishard & replica querying for faster speed



Elasticsearch Resiliency



All servers up and running.
P = 'Primary shard'
R = 'Replica shard'



Server 4 goes down.
Server 2's replica of activated as primary.
Server 6 instantiated and starts cloning data from
node 5 hexagon,
node 1 square,
node 2 diamond.

Elasticsearch uses

- **Logs** - Import tons of log formats out-of-the-box.
- **Metrics** - Track server/docker/k8/application metrics/uptime.
- **APM - Application Performance Monitoring** - Elasticsearch help enable ‘distributed tracing’ which is following a client journey through different parts of the system to identify bottlenecks
- **Site, App, workplace search**
- **Maps** - visually analyzing geospatial data, identify website traffic, geospatial holes in data, outlier cases and anomalies.
- https://www.elastic.co/elasticsearch/#what-exactly-can-i-use-elasticsearch-for%3F

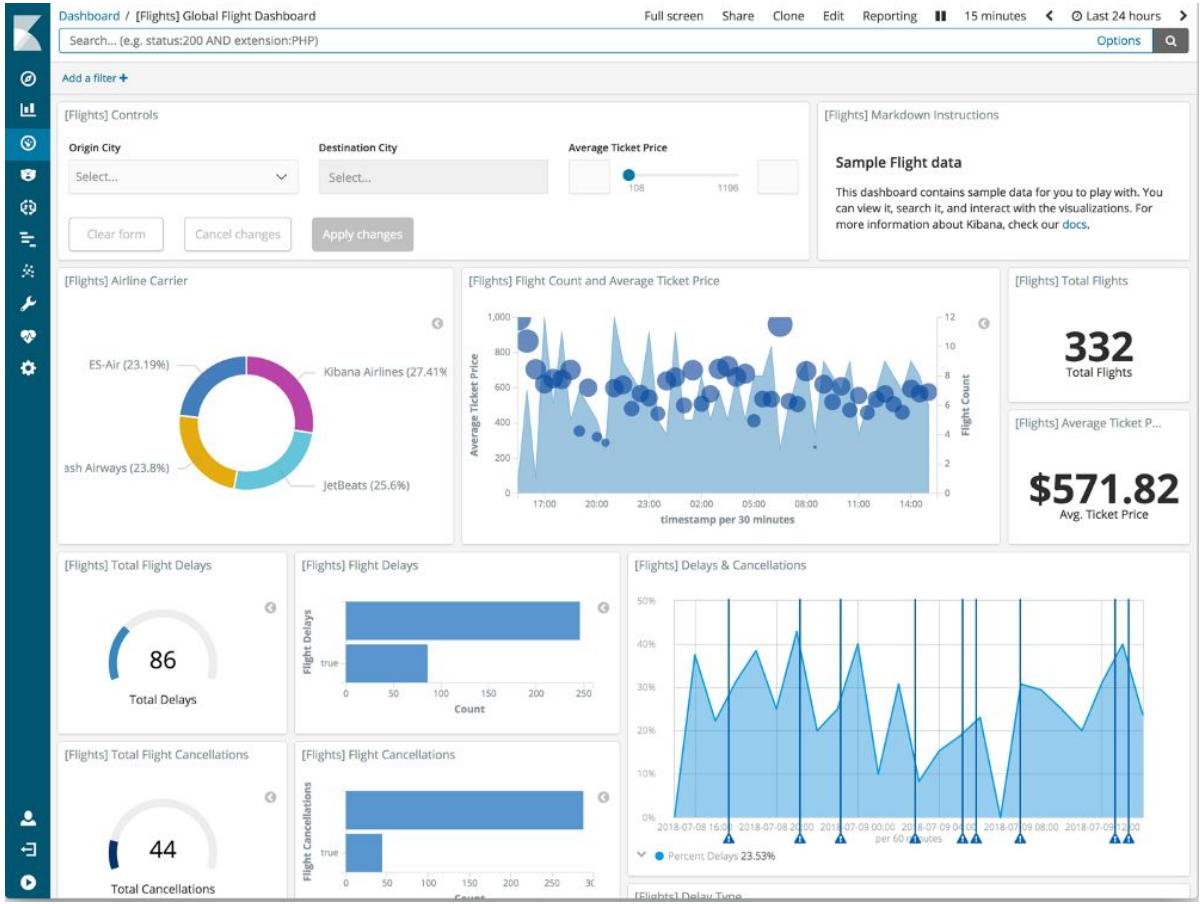
Kibana - Front-end

Provides:

1. **Admin tool** for managing an ES cluster of nodes health and performance
2. **Visualization tool** for data collected and indexed by ES

Full Feature list: https://www.elastic.co/kibana/features

Admin Dashboard



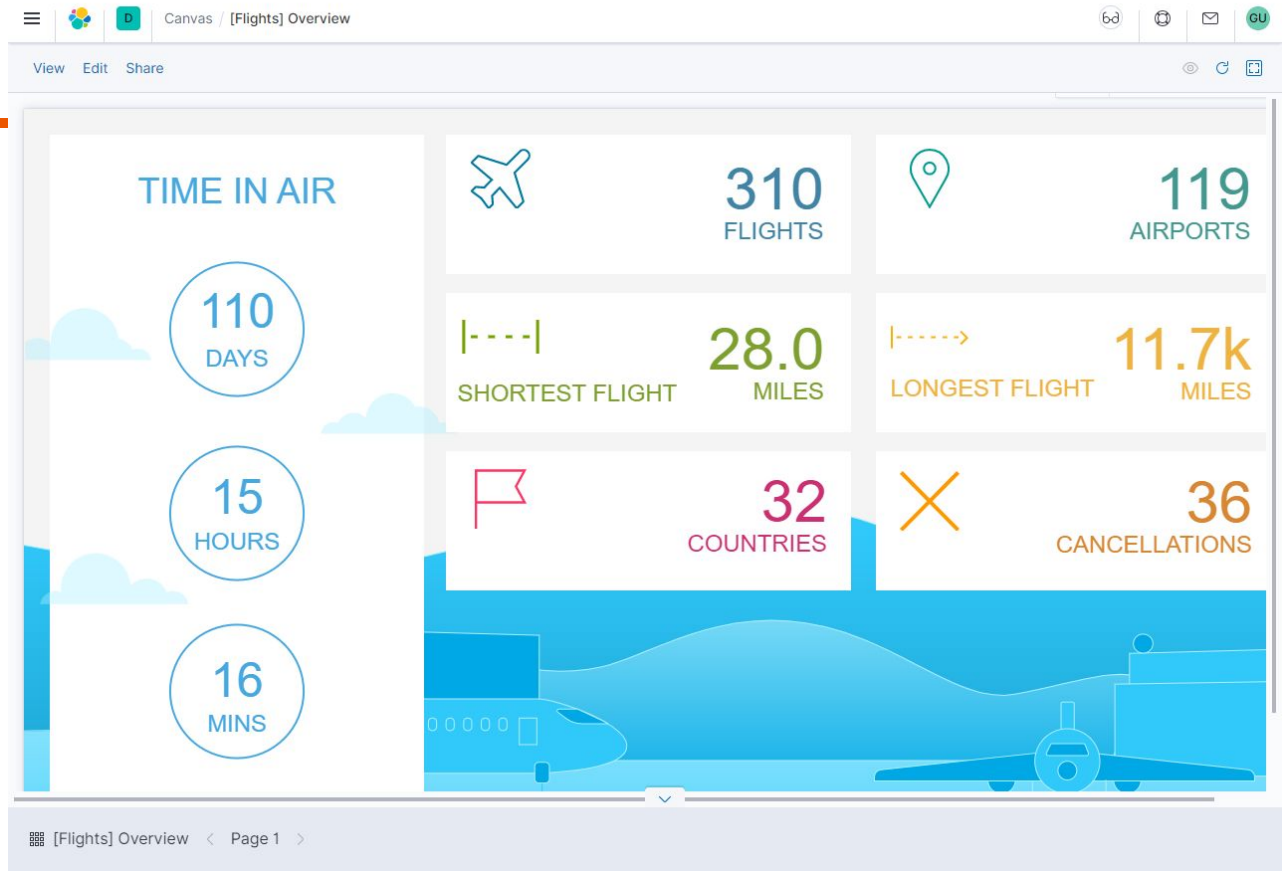
- RBAC (role-based access control)
- Alerting to email, slack, pagerDuty, webhooks
- out-of-box preconfigured for docker, dmbs, webservers
- Custom dashboards for any data

Visualizations available on Kibana

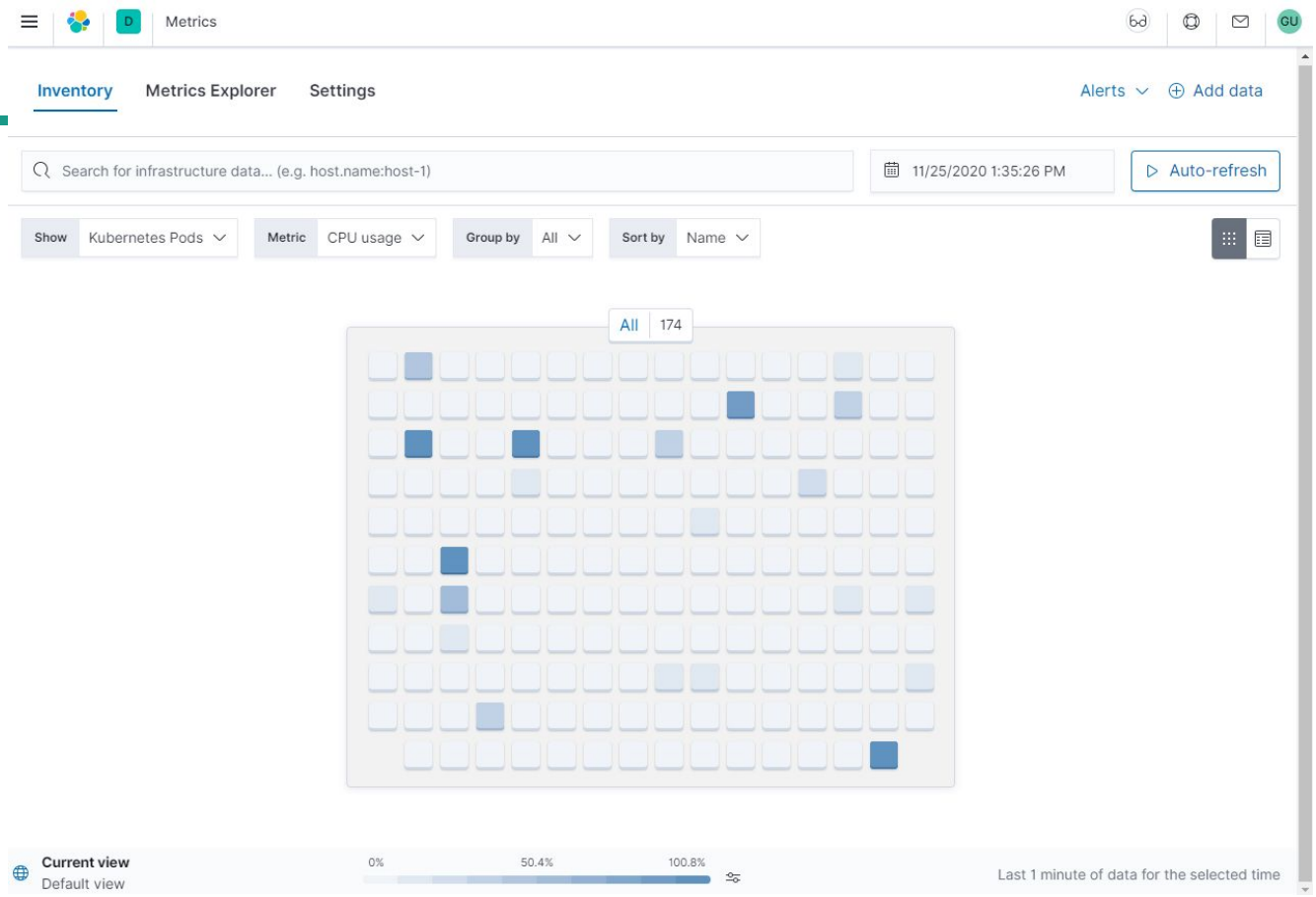
- Data can be ingested from many data sources
  - App, service, sample sets by configuring a beats data shipper
  - Upload data files (csv, json, logs) up to 100mb default (1gb max)
  - Geospatial data (lat/long columns or geoJSON/GDAL)
- Main demo page (some examples in following slides)

Maps demo: [link]

Canvas demo: [link] Flights



Infrastructure dashboard - k8pods - [link]



Beats - Data shippers

Eight beats included: (Also, Community Beats archive for user-created Beats.)

- **Filebeat** - log files (backpressure handler if Logstash gets backed up). Lowers number of beats, until LS catches up.
- **Metricbeat** - Server CPU/mem/disk/network stats
- **Packetbeat** - Network packages (http), traffic, latency, error data
- **Winlogbeat** - Windows event logs, for security/activity analysis
- **Auditbeat** - Data from audit daemon, kernel level access to Linux events
- **Heartbeat** - Monitors uptime and response time for sites
- **Journalbeat** - system daemon journals for linux at all levels
- **Functionbeat** - data streaming pipeline for logs coming from FaaS (function as service aka Lambdas) platforms

Logstash

Elasticstash is the backend data store, Kibana is the front-end reporting tool. Beats can take in data and process each line, but if you want to transform that data, here is Logstash.

Logstash is where built-in inputs, filters, codecs and outputs live.

- Unstructured data => structured data
- IP addresses => geographic locations
- Fields removed, manipulated or recalculated

Many plugins out of the box for all kinds of events (beats, cloudwatch, files, gc storage, irc, kafka, redis, network stuff (tcp, snmp, udp, websocket), twitter)

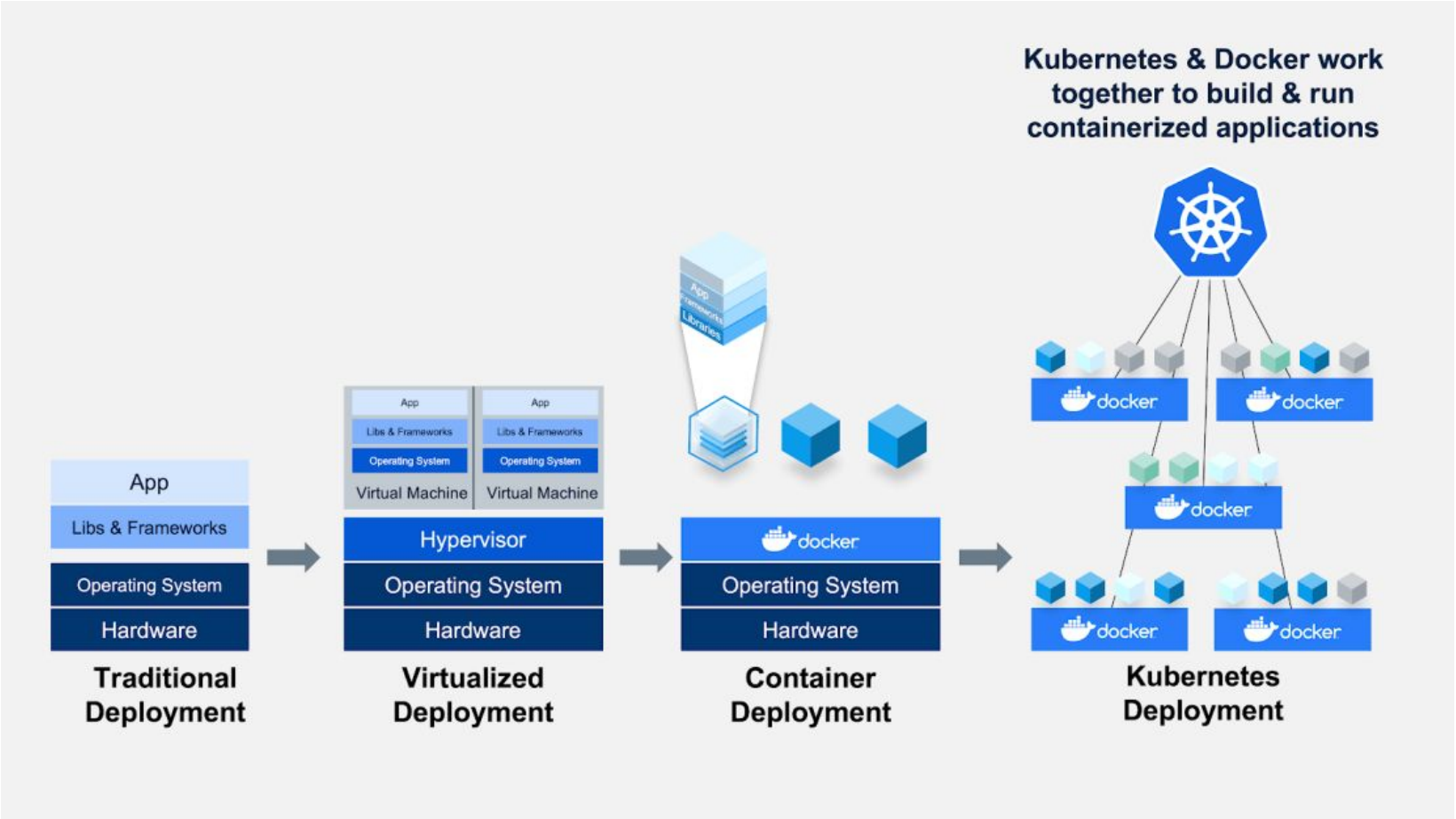
Elasticsearch Modules Related to our Stack

1. Filebeat - Kafka module - The kafka module collects and parses the logs created by Kafka.
2. Metricbeat - Kafka module - default (consumer, partition), also has broker, consumergroup, producer metricsets
3. Metricbeat - ZooKeeper module- default (mntr/server), also connection. Mntr = network, latency, follower, connections
4. Elasticsearch for Apache Hadoop - connector for Hadoop/hive/pig/spark/storm to Elasticsearch. Features.
5. Filebeat - nginx module - parses and ingests access and error logs for nginx http server for kibana
6. Metricbeat - nginx module - periodically fetches metrics from nginx http servers (default stubstatus)
7. Filebeat - docker input - reads logs from docker containers
8. Metricbeat - docker module - fetches metrics from docker containers (default: container, cpu, diskio, healthcheck, info, memory, network.)
9. Metricbeat - k8 event metricset - This is the event metricset of the Kubernetes module.
10. Metricbeat - ec2 metricset - The ec2 metricset of aws module allows you to monitor your AWS EC2 instances, including cpu, network, disk and status. ec2 metricset fetches a set of values from Cloudwatch AWS EC2 Metrics.

Links

- Elastic website is here. Kibana Demo site
- On AWS: the ELK stack
- On Stackshare: Elasticsearch Kibana Logstash
- For the article that a lot of this slideshow came from (somewhat rambling) check here.

Kubernetes (K8), Docker, Helm



Source: https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/

Helm is a package manager and provides ready-to-deploy
Instances (chart) of many popular packages. [link]

Chart = package
Repository = chart storage
Release = chart running in k8

Helm installs charts into Kubernetes, creating a new release for each installation. And to find new charts, you can search Helm chart repositories.

Vocab/Abbrev

- **GA** = General Availability Release (as opposed to Beta, Alpha, etc)
- **GKE** = Google Kubernetes Engine (https://cloud.google.com/kubernetes-engine)
- **ECK** = Elastic Cloud on Kubernetes (https://github.com/elastic/cloud-on-k8s)
- **Operator pattern** = https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
- **K8 concepts**: https://kubernetes.io/docs/concepts/

Links

Videos

- Introduction to Microservice, Docker and Kubernetes by James Quiqley(good)

References

- Helm charts for Elastic @elastic
- All docker images @elastic
- Kubernetes icons on github

Applications for PJ3

Applications for Project 3

Enterprise search

Search processed data, query same data spark is accessing by linking to HDFS/other data store that we're using as a data sink. **Logstash** ingests data (how does it work?) and outputs to **Elasticsearch**.

Observability

Monitor container, EC2, S3 metrics, logs and status (using file, packet and metric beat)

Real-time visualizations

Using **Kibana**

Ways to do stuff with Logstash

Integrations: pull data in through jdbc, kafka, rabbitmq (message-broker software)

Input plugins: ex: beats, file, gCloud storage/pubsub, github wh, graphite, log4j, s3, twitter, http/udp, more

Output plugins: csv, datadog, email, file, pubsub, mongodb.stdout/pipe, redis, s3, webhdfs, more

Filter plugins: aggregate, alter, clone, csv parse, dissect unstructured data, add geo to ip data, parse json/kv/xml, mutate fields, more

Links

Slidedeck: Building Streaming Data Pipelines with Elasticsearch, Apache Kafka and KSQL

Kafka Producers->Brokers/Cluster Architecture