Elastic QDM

question-driven-meetup

Tatjana Frank
Solutions Architect @ Elastic
17.10.2019

SHOW ME YOUR ELASTIC!

1. Project + Discovery → Data → WHAT?
2. Sizing + Hot/Warm → Index → WHERE?
3. Value + Solutions → Use Case → WHY?

elastic

Elastic Download Statistics

We are in using the products already!

>600M

Total Downloads

Elasticsearch
+ Kibana
+ Logstash
+ Beats

>20k

SaaS Instances

Managed by us

Across > 200TB RAM

3

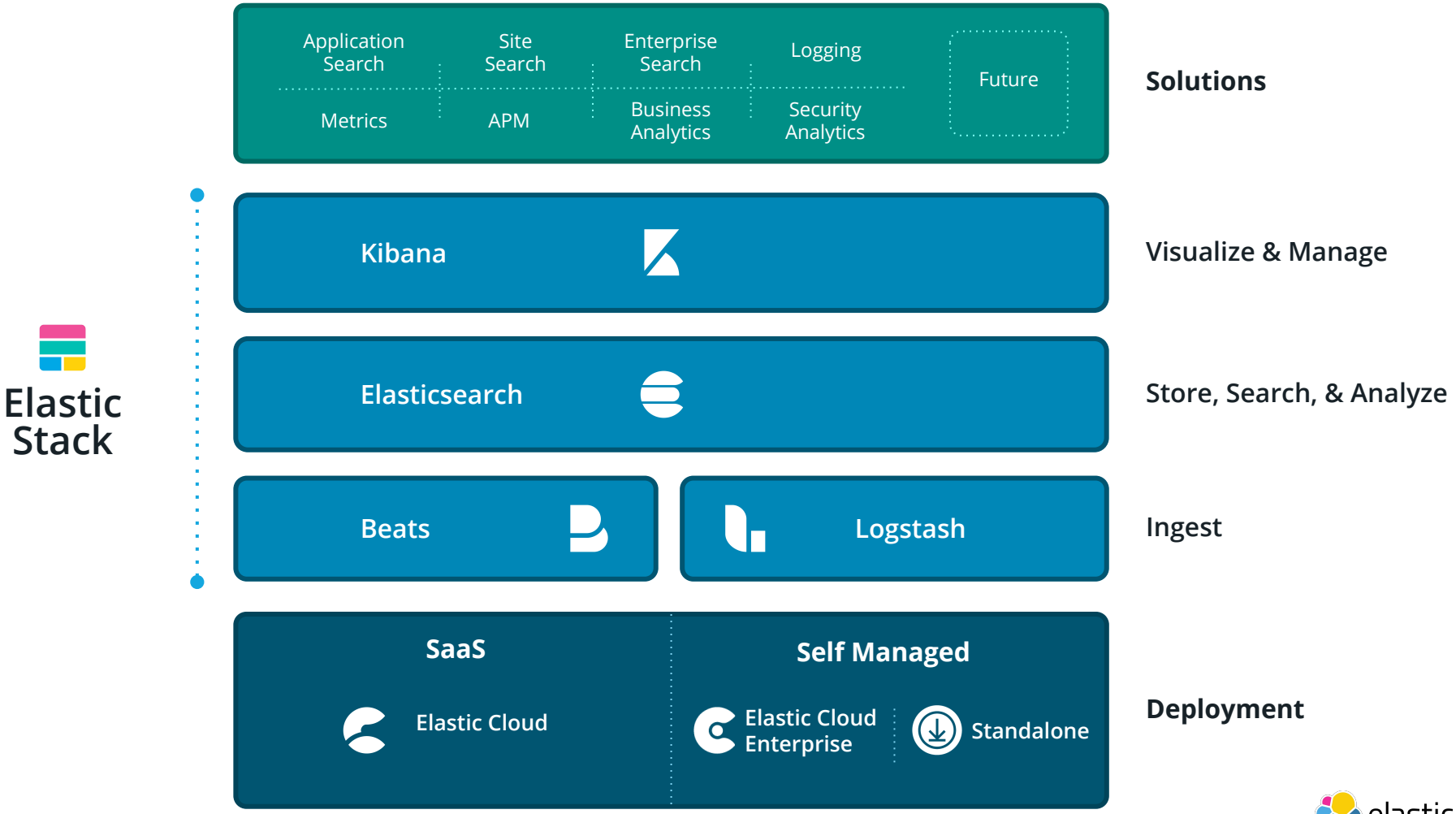
Ways to Consume

On-prem downloads
(RPM, ZIP, Docker, Kubernetes, ...)
SaaS *(AWS, GCP, Azure)*
ECE/ECK

7.4

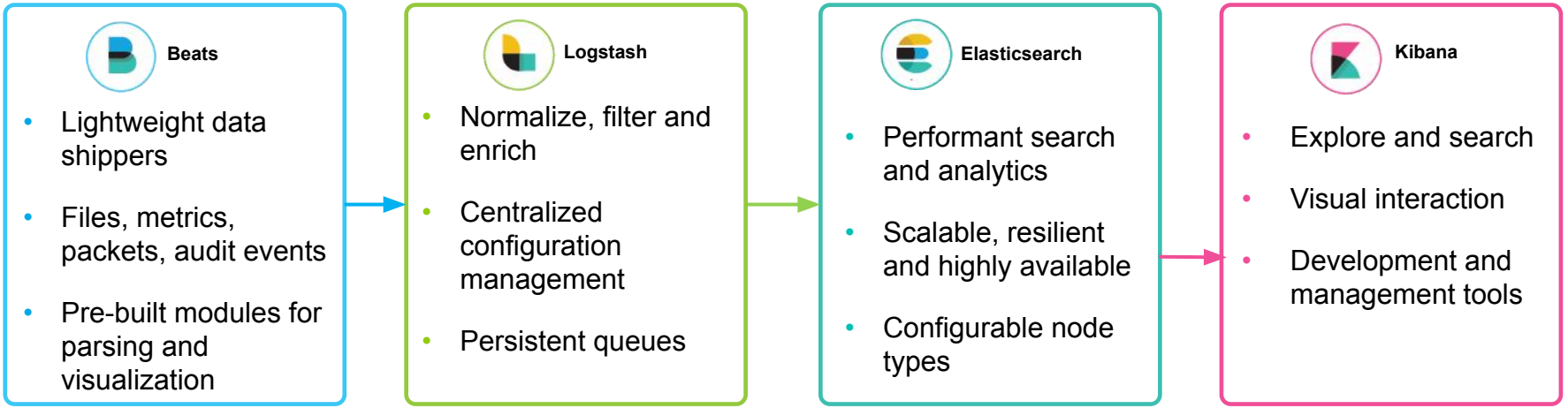
Current Version

Unified versioning
across all of the stack


Application Search
Site Search
Enterprise Search
Logging
Future
Metrics
APM
Business Analytics
Security Analytics
Solutions
Kibana
Visualize & Manage
Elasticsearch
Store, Search, & Analyze
Beats
Logstash
Ingest
Elastic Stack
SaaS
Elastic Cloud
Self Managed
Elastic Cloud Enterprise
Standalone
Deployment

elastic

Logical Processing Pipeline



WHAT?

WHERE?

WHY?

Project

How to get on the same page?

What challenges you are trying to solve?

What are the major capabilities you are looking for?

What type of data will be stored in Elasticsearch?

Does the data contain Personally Identifiable Information or other sensitive information?

What does the (planned) technology landscape look like, apart from Elastic Stack?

What types of logs and/metrics do you plan to ingest into Elasticsearch? (Application logs, Security logs, System metrics, Sensor logs)

Discovery

How to understand the size?

Can you share the size of your envisioned solution in terms of… (rough estimates are fine)

Gigabytes per day: → **daily ingest**

Average Events Per Second:

Peak Events Per Second:

What is your desired retention period of data before it is deleted or moved away from Elasticsearch? (e.g. 3 months, 1 year) → **data retention**

What percentage growth do you expect from your data volumes over the next 12-24 months?

Describe any additional requirements you believe to be unique or uncommon in your project

WHAT?

WHERE?

WHY?

Hot, Warm, Frozen

Elasticsearch can use ***shard allocation awareness*** to allocate shards on specific hardware.
Index heavy use cases often use this to store indices on ***Hot***, ***Warm***, and ***Frozen*** tiers of hardware,
and then schedule the migration of those indices from hot to warm to frozen to deleted or archived.
This is an economical way to store lots of data while optimizing performance for more recent data.
During capacity planning, each tier must be sized independently and then combined.

Tier	Goal	Example	Storage Ratio
Hot	Optimize for speed	SSD Class SAN/DAS (> 200Gb /s)	1:30 (Ram/Disk)
Warm	Optimize for storage	HDD Class SAN/DAS (~ 100Gb /s)	1:100 (Ram/Disk)
Frozen	Optimize for archives	Cheapest SAN/DAS (< 100Gb /s)	1:500 (Ram/Disk)
		Beware of recovery failures with this much data per node	

elastic

Hot

In this phase, you are actively querying and writing to your index.

Warm

You are still querying your index, but it is read-only. You can allocate shards to less performant hardware. For faster searches, you can reduce the number of shards and force merge segments.

Cold

You are querying your index less frequently, so you can allocate shards on significantly less performant hardware. Because your queries are slower, you can reduce the number of replicas.

Frozen

A frozen index has little overhead on the cluster and is blocked for write operations. You can search a frozen index, but expect queries to be slower.

Backup

Using Snapshot and Restore API, the data will be moved out of an Elasticsearch cluster and archived on a defined storage. It can be restored back into an Elasticsearch cluster.

Scenario A

We want to migrate our data from Arcsight to Elasticsearch. We have two years of logs in Arcsight, which is 10TB after being compressed at a 10:1 ratio. We tend to look at just the last 30 days. We want those searches to be really fast. Sometimes we look as far back as a year. Regulations require us to retain our logs for seven years, and we'd prefer to keep it all in Elasticsearch to keep things simple. Those searches can be as slow as necessary to keep costs low. The largest sized disk we can put on any node is 24TB HDD. What options do we have to balance performance with costs? What will the architecture look like?

Example Solution

Volume Sizing

Raw data /day	137GB	(10TB * 10 Uncompressed / 365 Days / 2 Years)
Total storage /day	559GB	(137GB * 1.7 Net expansion * 2 Replication * 1.2 Reserved)

Tier	Retention	Storage	RAM	Ratio	Nodes
Hot	30 Days	16.8TB SSD	64GB	1:30	9
Warm	335 Days	187.3TB HDD	128GB	1:100	15
Frozen	2,190 Days	612.1TB HDD**	128GB	1:188	26

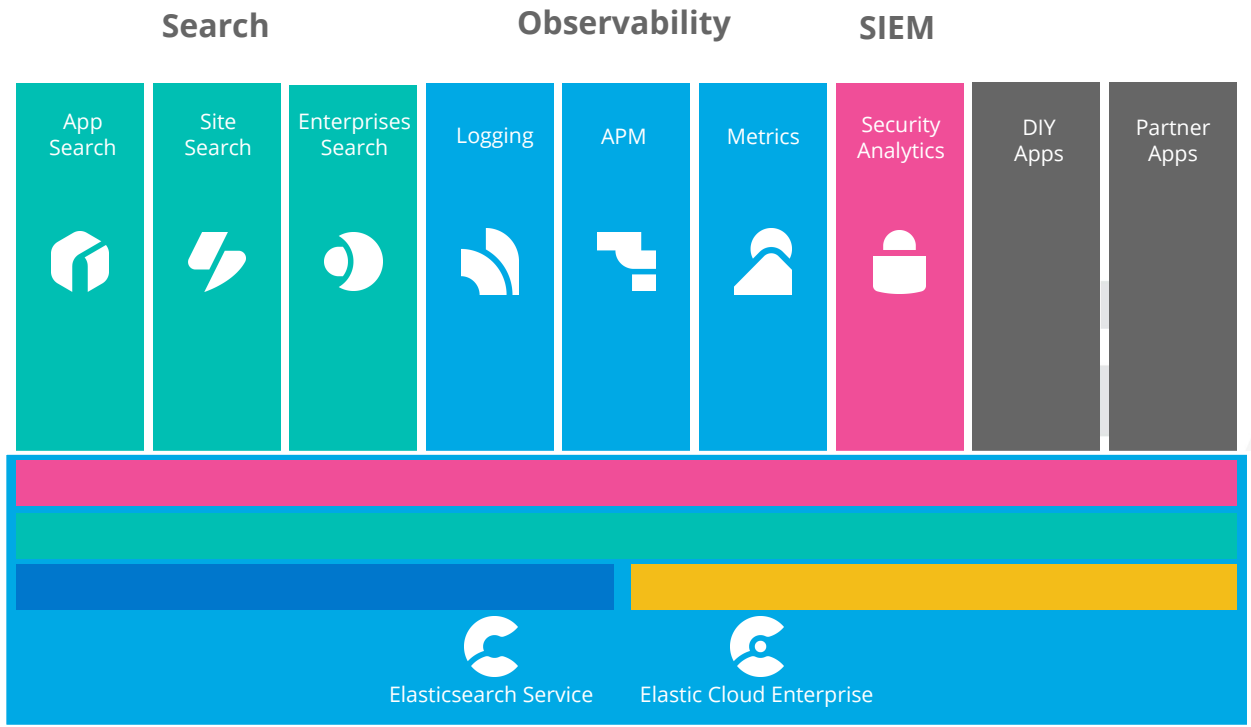
***Disable replication, restore from snapshots as needed*

Data Nodes 50 (9 Hot, 15 Warm, 26 Frozen)

elastic

WHAT? WHERE? **WHY?**

Elastic Helps Get You There Faster



elastic

IMAGINE… THINK… TALK…

WHAT? WHERE? WHY?

PRESENT!