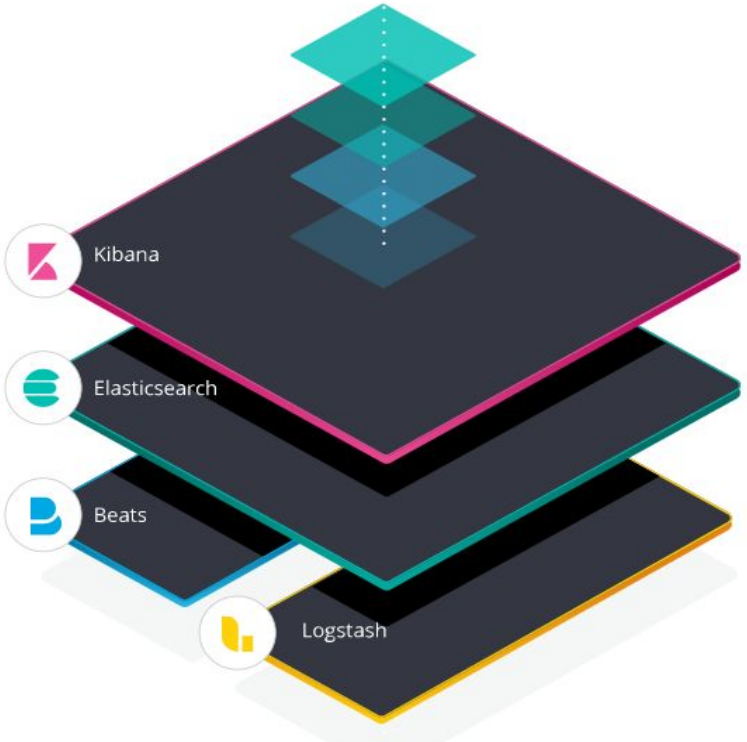
Introducción a Elastic Stack

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

¿Que es Elastic Stack?

- Conjunto de Herramientas de código abierto
- Conformado por:
  - ElasticSearch: motor de búsqueda y análisis
  - Logstash: canal de procesamiento
  - Kibana: herramienta de visualización gráfica
  - Beats: herramienta de recolección de datos



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

¿Que es Elastic Stack?

Casos de uso: Mercado Libre

- Permitir que los vendedores accedan a los datos críticos del producto
- Optimizar la administración del producto
- Ofrecer disponibilidad las 24 horas, los 7 días de la semana
- Prestar servicio a 4 millones de vendedores
- Manejar un crecimiento de 12 millones a 20 millones de listados de productos
- Agregar servidores en segundos según sea necesario

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Casos de uso: Telefónica

- Usa el motor elasticsearch para almacenar su tráfico de CDR (Call Detail Record -> Registro de Llamada Detallada).
- Es una característica del sistema que toma los detalles de llamadas, como tipo, tiempo, duración, origen y destino.
- Estos datos son usados para el control de la red, contabilidad y propósitos de facturación.

- Actualmente en un índice para 1 **solo día** existen más de **2 mil millones de registros** sobre un clúster de 50 instancias ElasticSearch almacenando hasta 18 meses de histórico y, en HDFS, el histórico total (hasta 10 años)

Casos de uso: Telefónica (cont.)

- Usa Elastic Stack para procesar los logs de su plataforma de video a nivel global.
- Obtener información sobre el consumo realizado por los clientes y el rendimiento del servicio.
- Examinar los canales que ven los clientes, así como los datos sobre latencia y las estadísticas de tasas de transmisión asociadas.

Telefónica

Número de clústeres	1
Número de nodos	10
Número total de documentos	30,176,007,552
Tamaño de datos total	27TB
Índice de indexación diaria	Aprox. 1-1,5 TB por día

Elasticsearch

- ElasticSearch es un motor de búsqueda y análisis de código abierto altamente escalable.
- Permite almacenar, realizar búsquedas y analizar un gran volumen de información de manera rápida y, casi, en tiempo real.

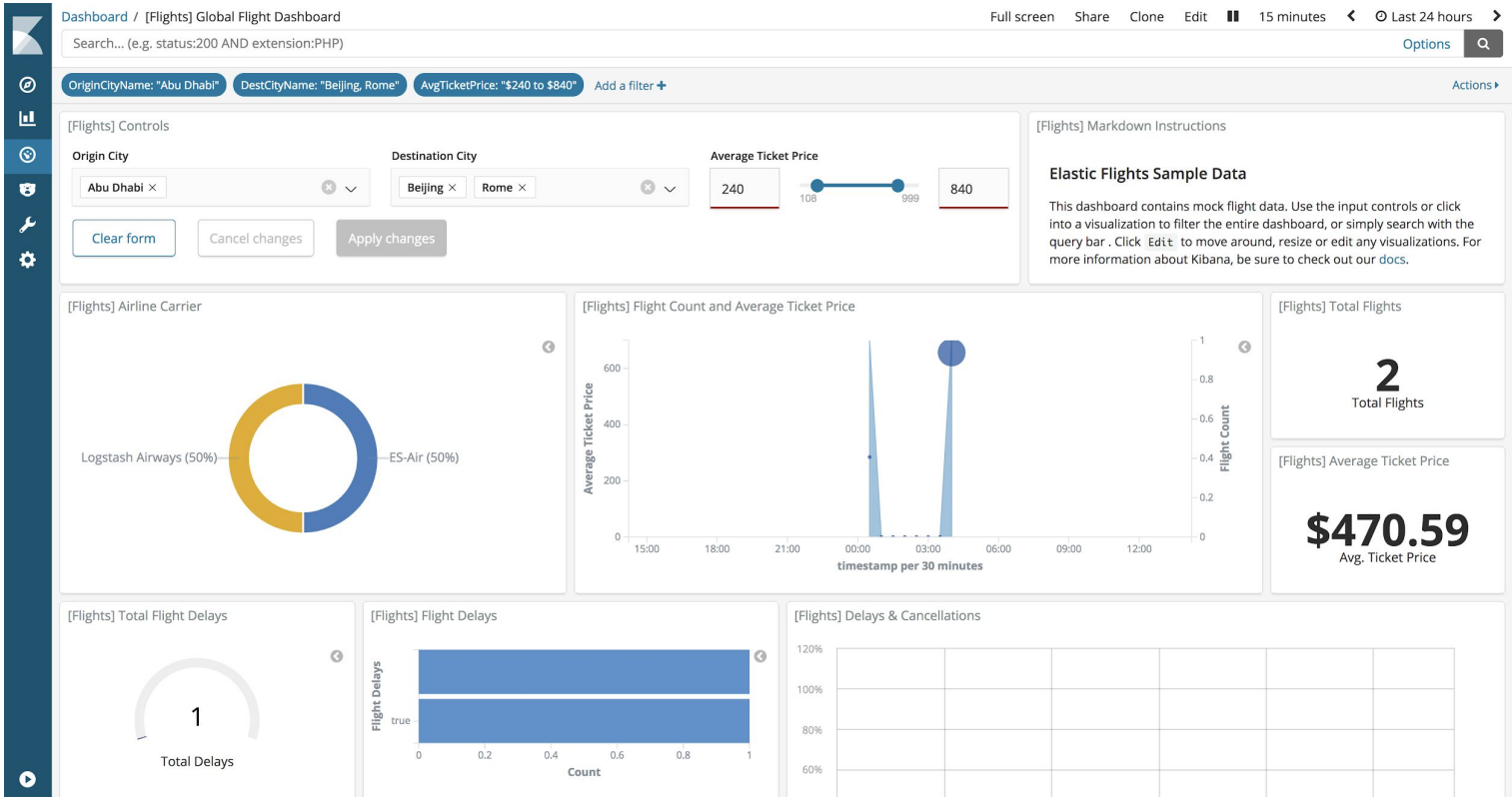
Elasticsearch (cont.)

- Tiene una muy buena performance en búsquedas de texto completo
- Gran flexibilidad de filtros por búsqueda y paginación
- Soporta gran volumen de indexación (cantidad de documentos / segundo).
- Basado en Apache Lucene.

Kibana

- Herramienta de búsqueda, visualización y explotación de datos de ElasticSearch
- Consultar los datos en ElasticSearch
- Navegar por todo el Elastic Stack
- Monitoreo de rendimiento del clúster de ElasticSearch

Kibana



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

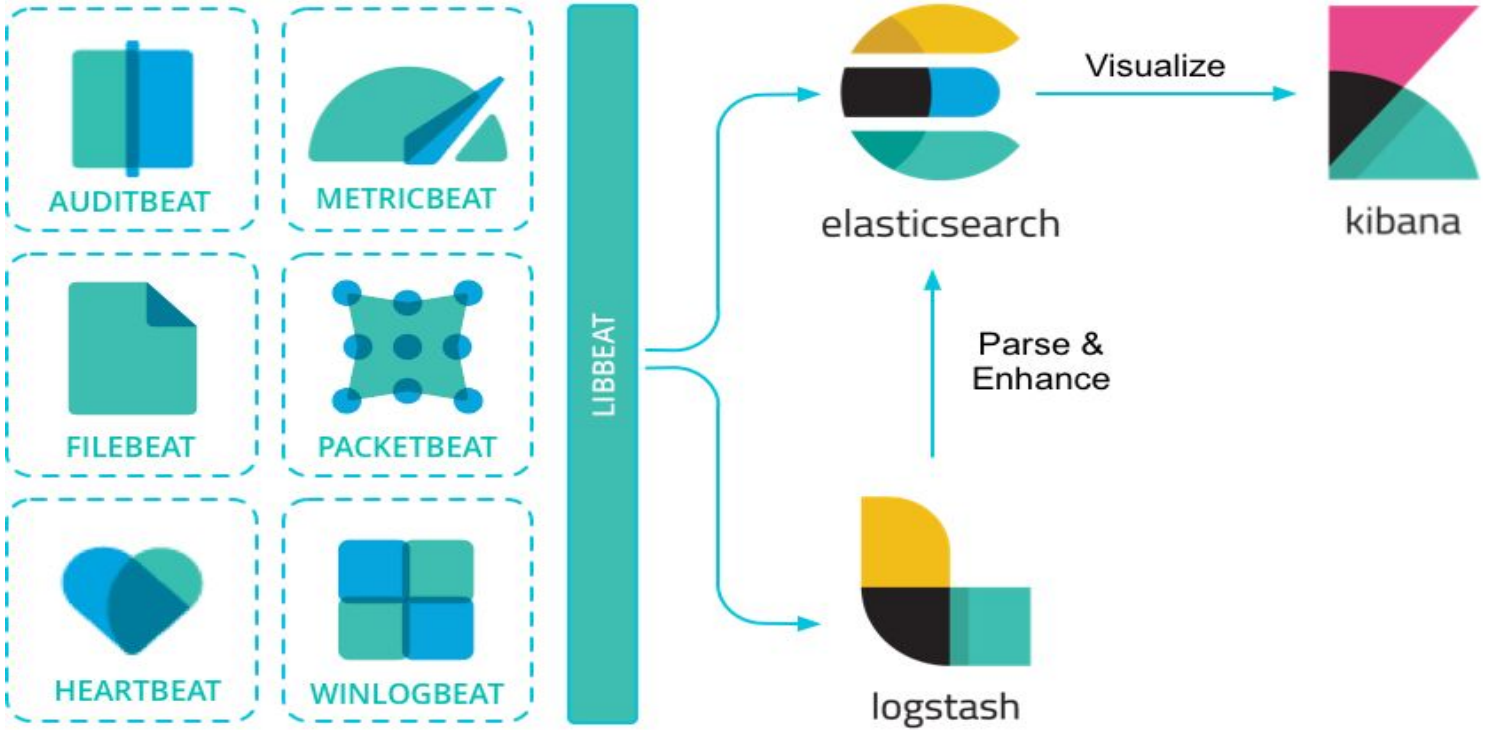
Kibana

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Beats

- Herramienta de recolección de datos
- Múltiples variantes dependiendo el tipo de datos
- Ocupa el mínimo espacio en memoria para optimizar el tráfico de la información

Beats



Logstash

- Herramienta de procesamiento de logs
- Recolectar, Parsear y Guardar logs para futuras búsquedas
- Basado en Jruby, funciona sobre la JVM

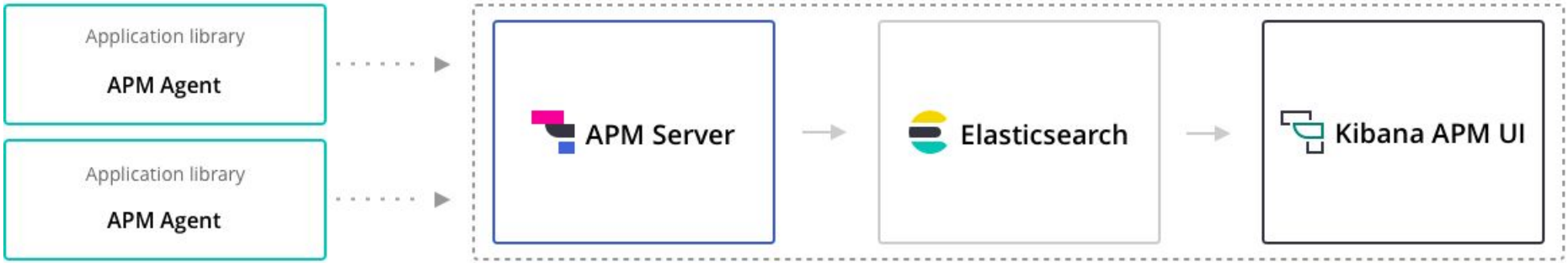
Elastic APM

Elastic APM: introducción

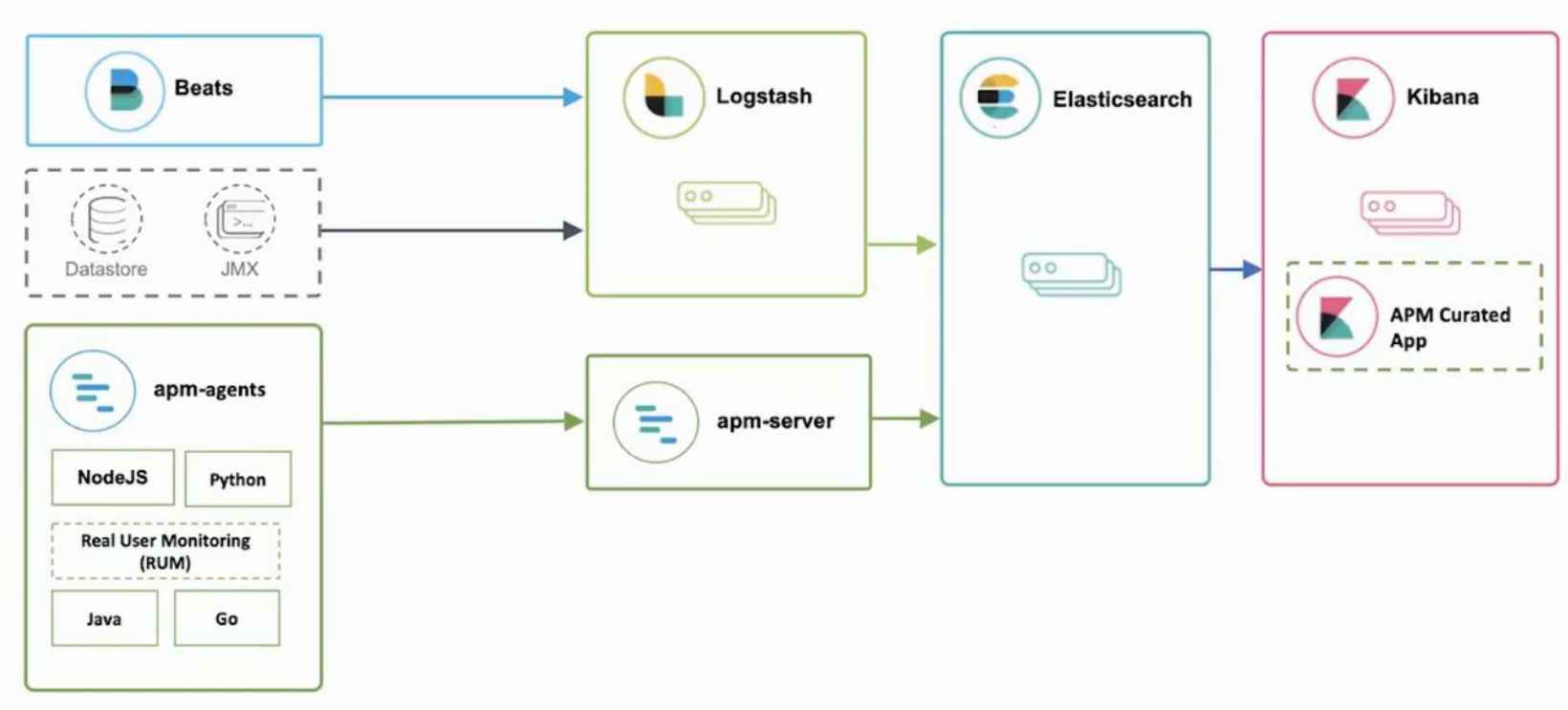
A partir de la versión 6.6 de ElasticSearch se integraron 2 componentes muy útiles para Application Performance Management:

- **APM server**
- **APM agents**

Elastic APM: arquitectura

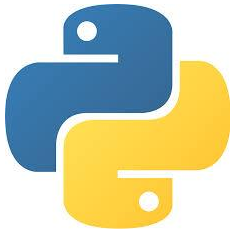


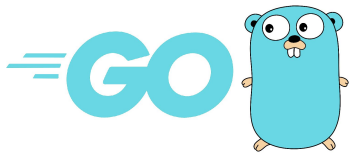
Elastic APM: arquitectura completa



Elastic APM: agent

- Los **APM agents** son bibliotecas de ***código abierto*** escritas en el mismo lenguaje de la aplicación y recolecta información de la performance y los errores en tiempo de ejecución.
- Almacenan la información en un buffer por un periodo corto de tiempo y la envían al **APM server**.
- Actualmente, existen agentes para los siguientes lenguajes: Python, Java, NodeJS, JS(RUM), Go, .NET, Ruby.



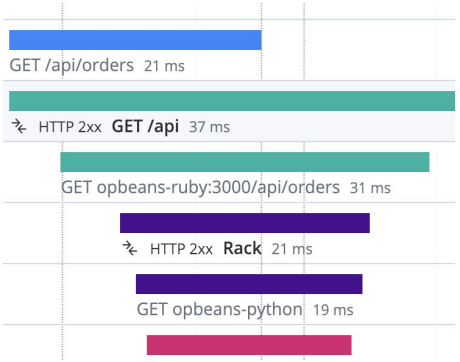



Elastic APM: agent (cont.)

- Para recolectar datos el **APM agent** se integra con el framework utilizado por cada lenguaje (p ej: para Python el agente soporta Django, Flask y WSGI). La integración del mismo requiere mínimos cambios en el código de la aplicación.
- El **APM agent** tiene integración con los conectores de base de datos o bibliotecas HTTP en las cuales no hará falta ningún cambio en el código de la aplicación.
- Adicionalmente, el agente recolecta métricas del sistema.

Elastic APM: JS agent

- **Real User Monitoring** (RUM): el **APM agent** para JS recolecta información sobre la interacción de un cliente con el código javascript.
- Provee métricas detalladas y “error tracking” de las aplicaciones web.
- Soporta los frameworks y plataformas populares y posee una API para casos de uso puntuales.
- El RUM agent brinda datos de manera automática de:
  - Métricas de carga de cada página
  - Tiempo de carga del contenido estático
  - API requests (XMLHttpRequest y Fetch)



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Elastic APM: server

- El **APM server** recibe datos de los **APM agents** y los transforma en documentos hacia Elasticsearch.
- Disponibiliza un endpoint HTTP en el cual los **APM agents** envían la información que recolectan.
- Envía toda esta información hacia índices que residen en **Elasticsearch** para luego ser consultados y/o analizados.

Tuning ElasticSearch / APM Server

- Hay parámetros que se pueden optimizar y ajustar para optimizar el funcionamiento de la herramienta tanto sea para acelerar la velocidad de indexación como para la mejor utilización de espacio en disco.
- Por un lado se podrá ajustar la configuración del **APM Server**, y por el otro, la configuración de los índices dentro de **ElasticSearch**

Tuning Elastic APM Server

- Cantidad de workers en paralelo que el servidor destinará a la indexación: **output.elasticsearch.worker**. Este valor se calcula realizando pruebas incrementando dicha cantidad hasta que el I/O del disco o el uso de CPU del clúster se sature.
- Tamaño máximo de chunks de indexación en una operación bulk: **output.elasticsearch.bulk_max_size**. Este valor se recomienda sea de aprox. 5000 (el default es 50)

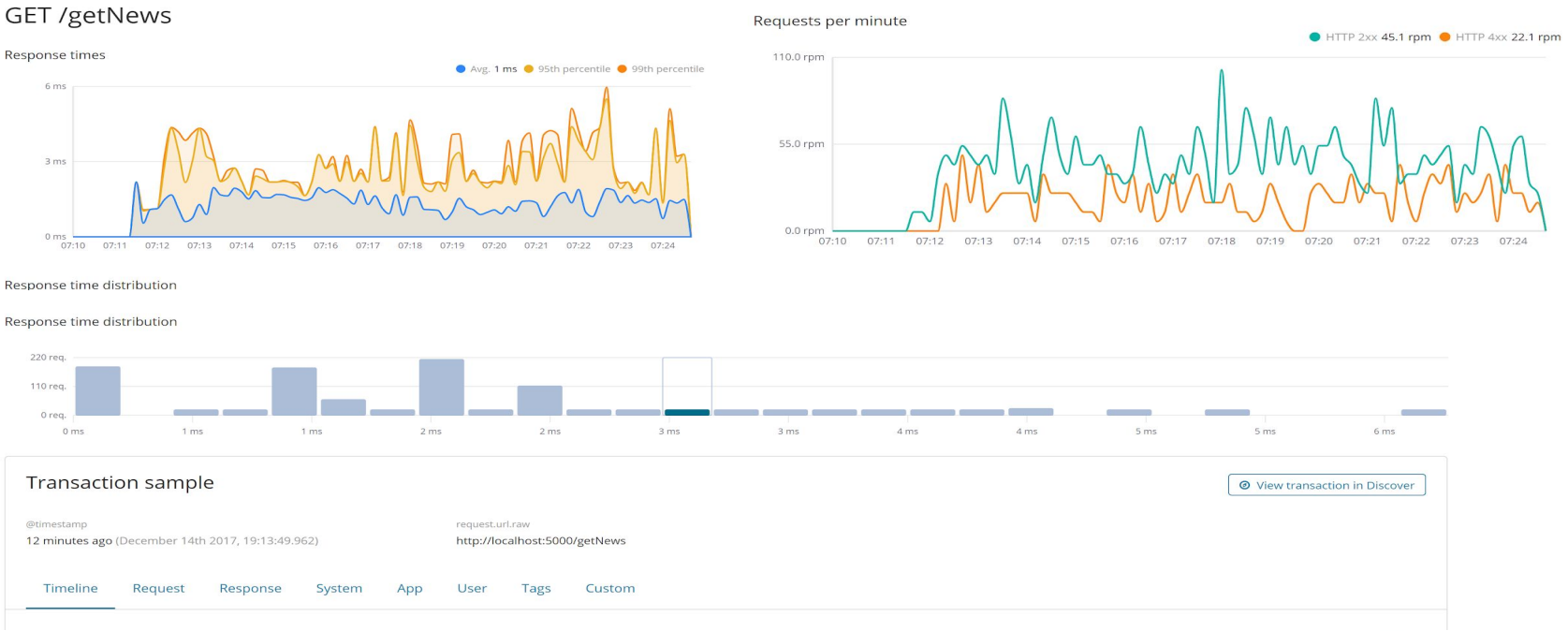
Tuning Elastic APM Server

- Si al server no le diera el tiempo para procesar la información se observarán “request timeouts”. Esto se puede resolver al agregar más instancias **APM server**. Lo que proveerá **disponibilidad** y al mismo tiempo aumentará la capacidad de procesamiento.
- Ajustar el tamaño de la cola interna del server. Esto posibilitará mantener más eventos en memoria aunque ElasticSearch no esté disponible por largos períodos. Asimismo, podría mitigar errores ocurridos a partir de picos de tráfico muy altos.
- Grandes Payloads pueden resultar en “request timeouts”. Para lo cual, el reducirlo causará que los agentes envíen menor cantidad de información pero de manera más frecuente.

Tuning ElasticSearch / APM Server

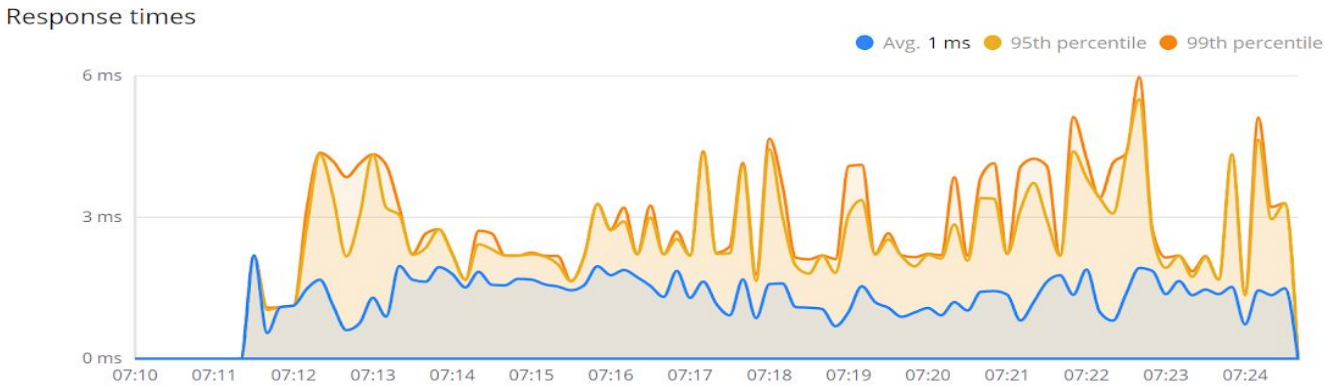
- Se puede tunear desde el lado de ElasticSearch para optimizar:
  - Velocidad de indexación: no setear el “refresh interval” o en un valor alto, deshabilitar swapping, optimizar la caché del filesystem, configurar el tamaño del buffer por índice (se recomienda 512MB por shard, por default es el 10% del java heap)
  - Uso de almacenamiento: deshabilitar funcionalidades que no se utilizarán, ajustar tamaño del shard, reducir cantidad de shards por índice

Elastic: APM ejemplo

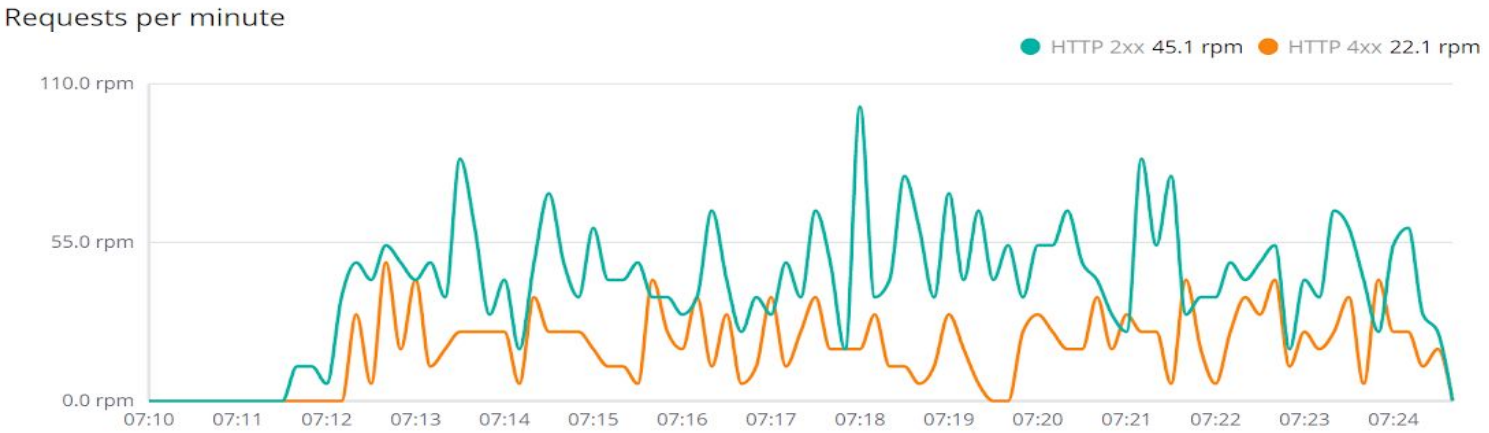


UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Elastic: APM ejemplo



Elastic: APM ejemplo

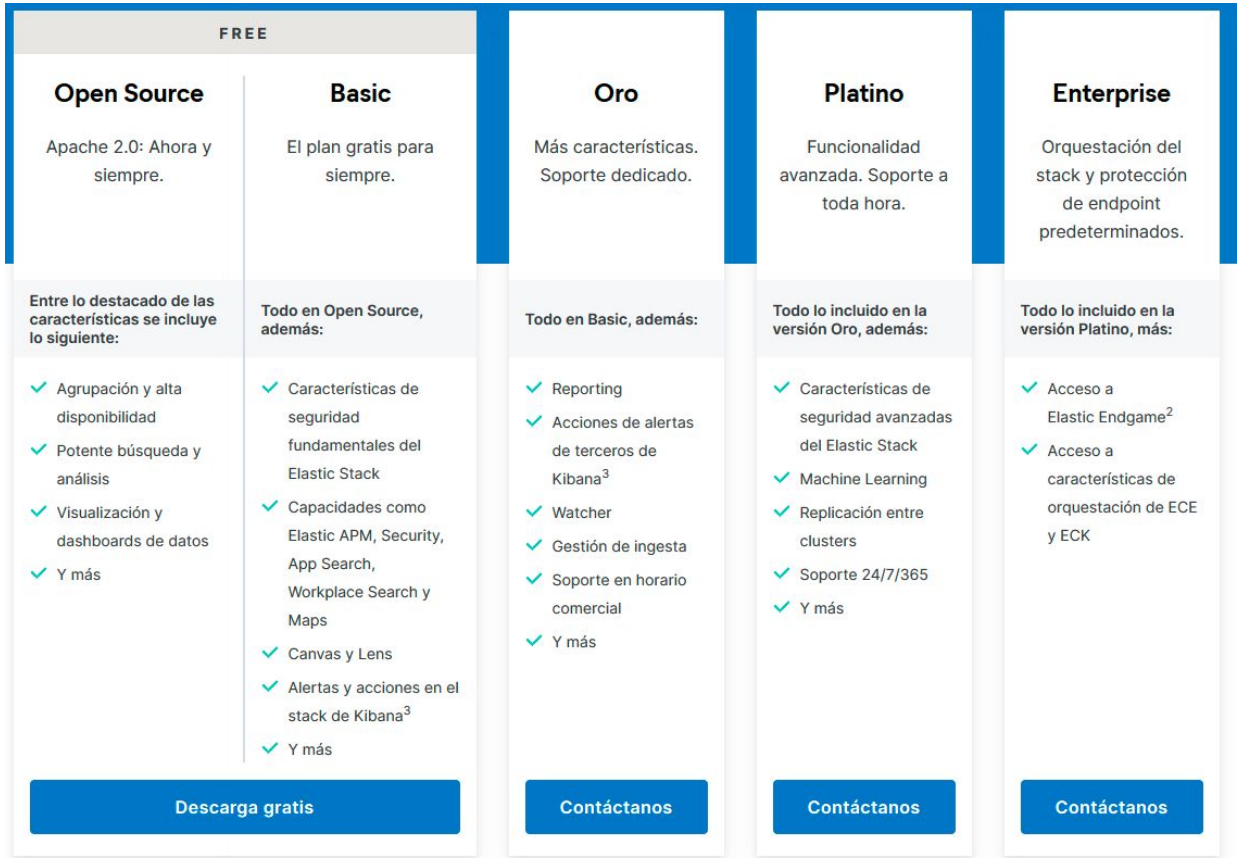


Elastic: APM ejemplo

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Suscripciones

ElasticSearch: suscripciones



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

ElasticSearch Basic

ElasticSearch Basic

- Storage types
  - Inverted index (for search)
  - Document store (for unstructured)
  - Columnar store (for analytics)
  - Frozen indices (for long term storage)

ElasticSearch Basic

- Clients
  - Clients
  - REST APIs
  - Language clients
  - Query DSL
  - Console
  - ES-Hadoop
  - Elasticsearch SQL APIs & CLI

ElasticSearch Basic

- Data Management
  - Data management
  - Snapshot/restore
  - Minimal snapshots
  - Snapshot lifecycle management
  - Index management
  - Index lifecycle management

ElasticSearch Basic

- Stack Management
  - Stack management
  - Data import tutorials
  - Grok Debugger
  - License management
- Machine learning
  - Data Visualizer

ElasticSearch Basic

- Scalability & resiliency
  - Clustering & high availability
  - Automatic data rebalancing
  - Cross-cluster search
  - Voting-only master nodes

ElasticSearch Basic

- Ingest products & features
  - Filebeat, Metricbeat, Winlogbeat, Packetbeat, Heartbeat, Auditbeat
  - Functionbeat
  - Logstash
  - ES-Hadoop
  - File import wizard

ElasticSearch Basic

- Share & collaborate
  - Embeddable dashboards
  - Object export UI & APIs
  - CSV exports
  - Saved queries

ElasticSearch Enterprise

ElasticSearch Enterprise

- Clients
  - JDBC Client
  - ODBC Client
- Scalability & resiliency
  - Cross-cluster replication*
- Machine learning
  - Root cause indication
  - Alerting on anomalies

ElasticSearch Enterprise

- Security
  - Audit logging
  - IP filtering
  - LDAP, PKI*, Active Directory authentication
  - Elasticsearch Token Service
  - Single sign-on (SAML, OpenID Connect, Kerberos)
  - Field- and document-level security
  - Custom authentication & authorization realms

ElasticSearch Enterprise

- Stack Monitoring
  - Security
  - Stack monitoring
  - Configurable retention policy
- Alerting
  - Highly available, scalable alerting
  - Notifications via email, Slack, Pagerduty, Jira, or webhooks
  - Alerting UI

ElasticSearch Enterprise

- Ingest products & features
  - Elastic Endpoint Security
- Share & collaborate
  - PDF and PNG reports

ElasticSearch Enterprise: Elastic Endpoint Security

- Endgame Platform
  - Role-based access control
  - LDAP authentication
  - Single sign-on (SAML 2.0)
  - Mutual authentication between the platform and endpoint
  - RESTful API
  - Policy-based management

Endgame está pensado para automatizar y procesar auditorías

Primeros Pasos

Instalación de Elastic Stack: ElasticSearch

- Ingresar a: https://www.elastic.co/es/downloads/elasticsearch
- Descargar la versión correspondiente al sistema operativo
- Ingresar desde el cmd (o consola) a la carpeta correspondiente y ejecutar **“./elasticsearch”**

Instalación de Elastic Stack: Kibana

- Ingresar a: https://www.elastic.co/es/downloads/kibana
- Descargar la versión correspondiente al sistema operativo
- Ingresar desde el cmd (o consola) a la carpeta correspondiente y ejecutar **“./kibana.bat”**
- Ingresar a: http://localhost:5601/app/kibana y verificar el funcionamiento

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Instalación de Elastic Stack: Beats

- Ingresar a: https://www.elastic.co/es/downloads/beats
- Seleccionar el beat a descargar
- Descargar la versión correspondiente al sistema operativo
- Ingresar desde el cmd (o consola) a la carpeta correspondiente y ejecutar, ej: **“./filebeat”**

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Instalación de Elastic Stack: Logstash

- Ingresar a: https://www.elastic.co/es/downloads/logstash
- Descargar la versión correspondiente al sistema operativo.

Instalación de Elastic Stack: Logstash

- Crear un archivo de configuración example.conf con el contenido de la imagen.
- Ingresar desde el cmd (o consola) a la carpeta correspondiente y ejecutar **“./logstash -f example.conf”**

```
#Estructura Example.conf

    input {
        stdin {
            codec => json
        }
    }


    output {
        stdout {
            codec => json_lines
        }
    }
```