Control de cambios

Versión	Cambios
V1.2	Añadida los requisitos **advstats** y **ca** En el requisito **basic** se incluye que no se use mas de 10Gb en el home
V1.3	Detallados los requisitos **rt** y **dashboard**
V1.4	Detallado el requisito **export** y añadido un anexo de manejo de certificados
V1.5	Añadidas las comprobaciones de algunas de las operaciones
V1.6	Recalculo de las puntuaciones
V1.7	Aclaraciones de la instalación

Tabla de requisitos para la práctica y su puntación

Key	Feature (el rojo las obligatorias)	Puntos
basic	Mantener la instancia de elasticsearch operativa Cargar en real time los datos de los últimos 5 minutos Mantener correctamente cargados los datos de los 7 días anteriores No ocupar más de 10 GB en el home del usuario	+4.0
https	Configurar ES para que se acceda con protocolo HTTPS	+0.5
secured	Configurar ES para que se acceda con usuario y contraseña	+0.5
purge	Borrar los datos que llevan más de dos semanas en ES	+0.5
rt	Cargar los datos en Real time calculando indicadores con los datos anteriores	+2.0
dashboard	Crear un cuadro de mandos de Kibana con las visualizaciones indicadas	+1.0
cluster	Añadir una segunda instancia de ES	+2.0
batch	Añadir varias consultas que se ejecutan sobre los datos del día anterior de la instancia origen y se almacenan en la instancia propia	+1.0
export	Configurar un export diario de los datos del índice index-data	+0.5
backup	Configurar un backup diario de los datos del índice index-data	+0.5
audit	Configurar la auditoria para controlar las consultas que se realizan contra la instancia de ES	+1.0
alert-data	Configurar un mecanismo de alerta para que mande un correo de aviso en caso de que se superen ciertos umbrales en los datos de entrada	+1.0
advstats	Proceso de estadísticas avanzadas	+1.5
ca	Proveer servicio de firmado de certificados	+1.0

basic: hay que cargar los datos desde el cluster worker01, worker02 con usando el usuario consultas / icai4ever. Se cogen desde el índice ***index-data*** y se tienen que cargar en el índice ***index-data***. Hay que tener cargados correctamente los 7 días anteriores al día actual. NO se deben usar más de 10 GB de datos en el home del usuario correspondiente (/home/alumno/ahXX) (Este es un requisito fundamental, hay logs que pueden crecer enormente y llegar a llenar el sistema de ficheros y colapsar el servidor)

https: configurar ES para activar el protocolo de acceso https

https://www.elastic.co/es/blog/configuring-ssl-tls-and-https-to-secure-elasticsearch-kibana-beats-and-logstash

secured: crear el usuario consultas / icai4ever que solo tendrá permisos para leer el índice de datos

https://www.elastic.co/guide/en/elasticsearch/reference/current/secure-cluster.html

https://www.elastic.co/guide/en/elasticsearch/reference/current/configuring-stack-security.html

https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-reindex.html

purge: hay que borrar los datos anteriores a dos semanas, es decir si hoy es día 17, tiene que haber datos desde el día 3 en adelante, no anteriores.

https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-delete-by-query.html

rt: Crear un proceso que realice la carga en real time de los datos. Este proceso además de cargar los datos actuales almacenados en el índice index-data realizará los siguientes cálculos:

- Campo delta_rx_bytes: para cada host se tomará el dato de rx_bytes de ese instante y el dato del mismo host para el periodo de 5minutos anterior, y se calculará la diferencia entre el primero y el segundo. En caso de no encontrarse el valor anterior se almacenará un cero
- Campo delta_tx_bytes: idéntico al anterior con el campo tx_bytes.

dashboard: Crear un cuadro de mandos de ES con los siguientes elementos:

1. Una gráfica con la media de rx_bytes y tx_bytes para cada host
2. Una gráfica con la media de uptime por hora para cada host
3. Una gráfica con el máximo de uptime por día para cada host
4. Una gráfica con el valor máximo y mínimo de free_mem_kb, por hora para cada host
5. Una tabla con la media de java_procs para cada host
6. Una gráfica con la desviación estandar de data1 y de data2 por host y día

cluster: configurar otra instancia adicional para que se monten como un cluster.

https://www.elastic.co/guide/en/elasticsearch/reference/current/important-settings.html

batch: crear las siguientes consultas que se realizarán o bien contra el ES propio o contra el remoto y se almacenarán en un nuevo índice dentro del ES local

Consulta1:
Índice destino: agg-hour-yyyy.mm.dd (un índice con los datos de cada día)
Agrupando por el campo host y por hora:
Máximo, media y mínimo de los campos:
free_mem_kb, free_root_kb, java_procs, rx_bytes, tx_bytes
(Los campos se llamarán con el mismo nombre y terminadolos en _max, _min, _avg)

Consulta2: Agrupando por el campo host y por día:
Índice destino: agg-day-yyyy.mm.dd (un índice con los datos de cada día)
Máximo, media y mínimo de los campos:
free_mem_kb, free_root_kb, java_procs, rx_bytes, tx_bytes
(Los campos se llamarán con el mismo nombre y terminadolos en _max, _min, _avg)

Consulta 3:
Índice destino: max-uptime-yyyy.mm.dd ((un índice con los datos de cada día)

Para cara día generar un listado donde para cada hora se obtengan los 3 hosts (solo esos 3 hosts) que tienen el mayor valor de uptime en esa hora.

export: se programará un export diario de los datos del día anterior, el export de los mismo se hará con una llamada java (java -jar /home/shared/export.jar). Este tool necesita como parámetros de entrada:

```
-c,--credentials <arg>   Credentials to connect to elastic user:password
-f,--field <arg>         Timestamp field in the index
-from,--datefrom <arg>   timestamp from (yyyy-mm-ddThh:mi:ss)
-i,--index <arg>         index in elasticsearch
-k,--keystore <arg>      file for java keystore
-o,--output <arg>        output file
-s,--keypass <arg>       password for keystore
-to,--dateto <arg>       timestamp to (yyyy-mm-ddThh:mi:ss)
-u,--url <arg>           Elasticsearch url (http://host:port)
```

Se generará un fichero con el nombre export-yyyymmdd.txt.gz (comprimido) donde se almacenarán los datos correspondientes al día indicado en el fichero

backup: realizar un export del índice index-data empleando los mecanismos propios de Elasticsearch, se creará un repositorio llamado my_repo y los snapshots solo tendrán como nombre, backup_{fecha en formato YYYY.MM.DD} y solo contendrán el índice indicado.

https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-take-snapshot.html

https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-register-repository.html

audit: configurar la auditoria de ES para que se almacenen las consultas que se realizan contra la el propio ES, de esta manera se puede comprobar que usuario ha realizado qué consultas.

https://www.elastic.co/guide/en/elasticsearch/reference/current/enable-audit-logging.html

alert-data: crear un script que cada hora realice una consulta contra el ES local y que compruebe si valor uptime de alguno de los nodos ha superado un umbral (que será un parámetro de fácil configuración) y en caso de que así suceda se envíe un correo con la incidencia. La lista de correos destino será fácilmente modificable como parámetro. (os proveeré de una script que permitirá enviar los correos de manera sencilla)

advstats: en el campo data1 se espera una distribución uniforme de los datos a lo largo de cada día para cada uno de los hosts. Se debe implementar un mecanismo que compruebe que para el día anterior y para cada host se cumple este supuesto. En caso de que no sea así generar una alarma para el día en cuestión (os proveeré de una script que permitirá enviar los correos de manera sencilla)

ca: Proveer de un servicio de Autoridad Certificadora. Para ello es necesario generar los certificados de clave pública y privada de la CA, recibir las peticiones de ficheros de certificados con la solicitud de firma, firmarlos y devolverlos. Estos ficheros firmados tienen que funcionar como ficheros certificados para ES sin ningún problema, tanto para transport como para http.

Instalación de ES

Ejecutar el script `/home/shared/install-local.sh`

Aparecerán los puertos usados y los comandos para arrancar el ElasticSearch y el Kibana

```
puerto usado para elasticsearch: 9200
puerto usado para kibana: 10200
Arranque de elasticsearch:
/home/alumnos//ahxx/elasticsearch-7.16.1/bin/elasticsearch
Arranque de kibana:
/home/alumnos//ahxx/kibana-7.16.2-linux-x86_64/bin/kibana
```

Cada grupo tiene que usar su correspondiente puerto 92XX y 102XX, siendo XX los dos dígitos de su usuario.

IMPORTANTE: ES escucha en la IP: 127.0.0.1 y Kibana escucha en la IP: 192.168.80.38

Ejecutando esos dos comandos se arranca en primer plano las dos aplicaciones, será necesario un script adicional para que se arranque y se quede arrancado una vez cerrado el terminal (por lo menos el ES), para esto se puede usar el comando nohup. (El Kibana es opcional para facilitar la operativa)

Adicionalmente hay que configurar el directorio de ficheros temporales para la JMV, lo mas conveniente es modificar el fichero:

`/home/alumnos/ahXX/elasticsearch-7.16.1/config/jvm.options`

Y cambiar la línea:

`-Djava.io.tmpdir=${ES_TMPDIR}`

Por la línea:

`-Djava.io.tmpdir=/home/alumnos/ahXX/elasticsearch-7.16.1/tmp`

Comprobamos que ES está correctamente arrancado con el comando:

`curl -XGET localhost:9200/_cat/health`

El resultado tiene que ser algo como:

`1642872760 17:32:40 cluster-ah00 green 1 1 8 8 0 0 0 0 - 100.0%`

Podemos ver los índices que existen

`curl localhost:9200/_cat/indices`

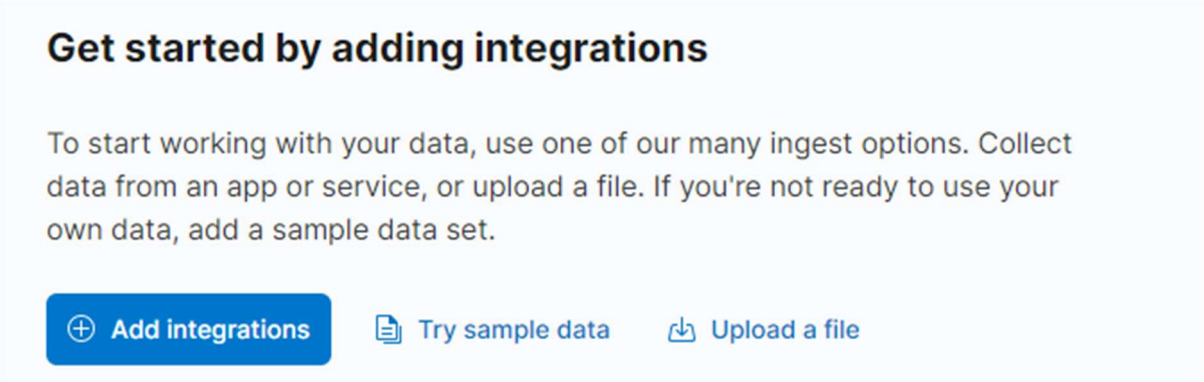
Vamos a copiar todos los datos del índice index-data en el ES local

```
curl -XPOST localhost:<port>/_reindex?pretty -H 'Content-Type: application/json' -d'{
  "source": {
    "remote": {
      "host": "http://worker01:9200",
      "username": "consultas",
      "password": "icai4ever"
    },
   "index": "index-data"
  },
  "dest": {
    "index": "index-data"
  }
}'
```

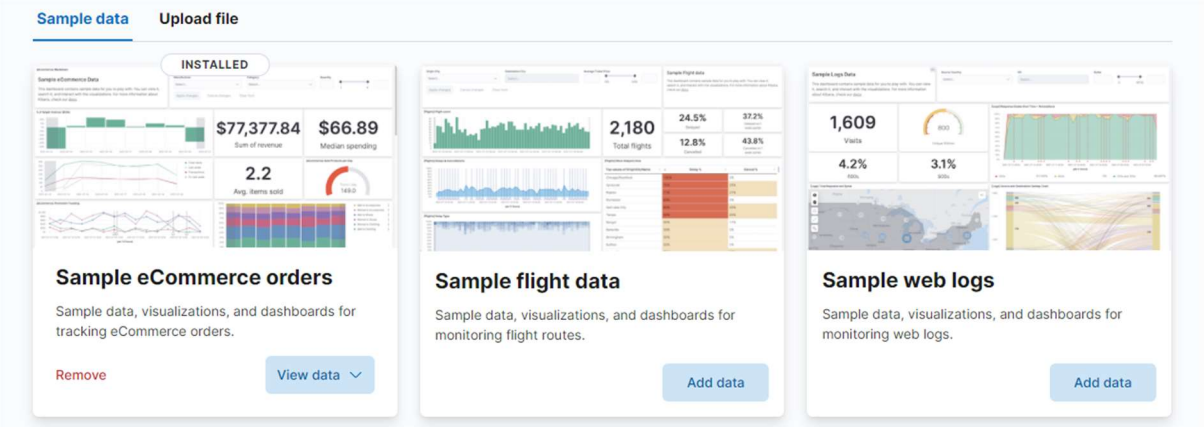
Configuración de Kibana

Se puede acceder a Kibana a través del navegador y la url: `192.168.80.30:<puerto>`

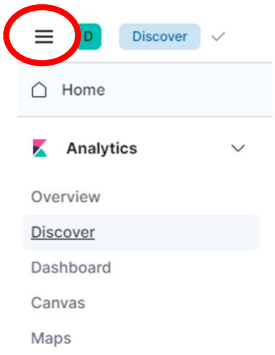
Pinchamos en Try sample data



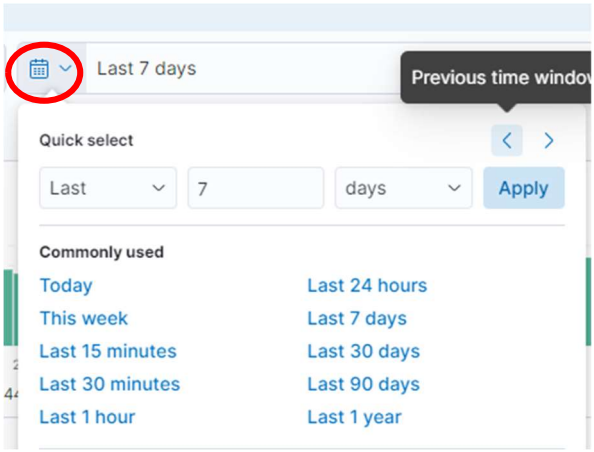
Y añadimos los 3 sets de pruebas



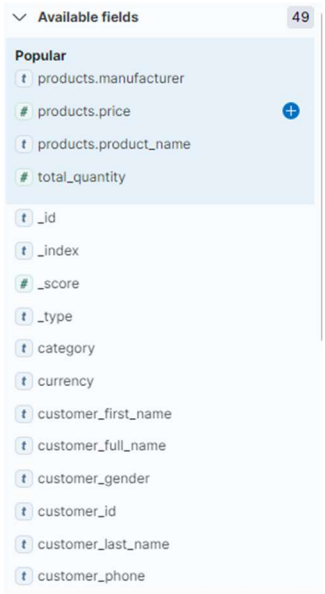
Pinchamos en menú Discover para poder explorar los datos



Hay que jugar con las opciones de filtros de fecha para poder ver datos



En los campos de la izquierda se pueden añadir filtros de una manera sencilla



Vamos a crear un índice para poder ver los datos que hemos importado

Pinchamos en Menú y la última opción Stack Management -> index patterns -> Create index pattern

El nombre será index-data

Y el campo de fechas @timestamp

Y volvemos a la opción de discover para poder explorar estos datos nuevos

Importacion de un certificado en un keystore de java

Para sacar el certificado de un fichero p12 (http.p12) a formato pem (OJO que mete más cosas que el certificado):

```
openssl pkcs12 -in http.p12 -out http.pem -clcerts -nokeys
```

Si tenemos un certificado en formato pem (http.pem), para importarlo en un keystore de java:

```
keytool -import -alias http -file http.pem \
     -keystore mykeystore.jks \
     -storepass icai4ever
```

Para listar todos los objectos que hay en un fichero .p12 (http.p12) :

```
openssl pkcs12 -info -in http.p12 -nodes
```

El formato PEM tiene este aspecto:

```
-----BEGIN CERTIFICATE-----
MIIDLDCCAhSgAwIBAgIVAMD9UOYGwbV14lCcAdVMbG9OvOuyMA0GCSqGSIb3DQEB
CwUAMCAxHjAcBgNVBAMTFUVsYXN0aWNzZWFyY2ggSFRUUCBDQTAeFw0yMjAxMjkx
OTIzNTNaFw0yNzAxMjkxOTIzNTNaMBExDzANBgNVBAMTBmVkZ2UwMjCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAMirChzXhDTvhxkiammicJGVNwCD5GNS
1D/F+gpOKm3UMPGzWB1kgIM9ELxGb1DMRIQ1XQJUB/J/3Rx3j09tPRImwWTmSuJI
txtpK96bUiTR9UDDNFLED5FeGItIbIDH8E5N1cGiGmxwRfS/ntYQZLeJJjZzeCHr
hREgVRGRmhNdN9BW1NcpOZw9br5Sk9OC1X8FpTOUr1Z67JvU+kv5nbKDth89x1Kz
y9oAzFQMhSuGxOLgOyvscWm6hUO+NypVrli3rGr3uZJuIo4I2zQkt4Sa6+RV7jws
+M9l/lzWgEuEB6lcpXubFDCcxGOWqfMDemGyAY6ccjWCPZbdvdtZ5KMCAwEAAaNs
MGowHQYDVR0OBBYEFDtyVAZNoMEatFCfXKdeUJymYY+vMB8GA1UdIwQYMBaAFBfp
nSgZ2uOwICcVZnHW5VXNqMBZMB0GA1UdEQQWMBSHBMCoUCaHBH8AAAGCBmVkZ2Uw
MjAJBgNVHRMEAjAAMA0GCSqGSIb3DQEBCwUAA4IBAQA/K9xFkmKq/hm3HDydS6nS
mMuROkeX3D9GIlECVtH1WZQ2UOWyaMF7wKlcPT8MH/gEHZhrBhnWxIcfFh6Ucr6t
Hor+OAsB+RJyGxa31lqjfjSAAidAV69ynwYI1DIGEED5+WWWpnJjs8FeYsmyyIPO
Jg4gEams8xE8i/mos1lpY6M4Qr1HYnOZj31BpXYJ3vJDcqwzqstOxsUSMztLAW3Q
Upm4rj3PZm9QzUDjtQEZaT+NV2Q5tjRAUmMz1HNbFH/NUtH9VSiJ9po+islxsNjM
UKld7OirSmyOx8IVCunvEpqMd+arTvXcGzPRPCNXkJJLjDoxcj4vRCgBdpnN9nWv
-----END CERTIFICATE-----
```

Comprobaciones

Key	Feature
basic	

purge	`curl -u consultas:icai4ever -k -s -X GET "https://localhost:92XX/index-data/_search?pretty" -H 'Content-Type: application/json' -d'`
	`{ "size" : 0,`
	`  "query": {`
	`  "range": {`
	`  "@timestamp": {`
	`  "lt": "now-14d/d"`
	`  }`
	`  }`
	`  }`
	`}`
	`'`

rt	Consulta para obtener los datos de los últimos 10 minutos:
	`curl -u consultas:icai4ever -k -s -X GET "https://localhost:92xx/index-data*/_search?pretty" -H 'Content-Type: application/json' -d'`
	`{ "size" : 0,`
	`  "query": {`
	`  "range": {`
	`  "@timestamp": {`
	`  "gte": "now-10m"`
	`  }`
	`  }`
	`  }`
	`}'`

dashboard	
https	

secured	La primera consulta NO debe funcionar y la segunda SÍ
	`curl -s -u consultas:icai4ever -k -X GET https://localhost:9204/_cat/indices?pretty -H 'Content-Type: application/json'`
	`curl -s -u consultas:icai4ever -k -X GET https://localhost:9204/index-data/_search?pretty -H 'Content-Type: application/json'`

cluster	
batch	
export	
backup	`curl -k -u elastic:icai4ever "https://localhost:9200/_snapshot/my_repo/backup_yyyy.mm.dd"?pretty`
alert-service	
audit	Propiedades en el fichero elasticserch.yml
alert-data	
advstats	
ca	