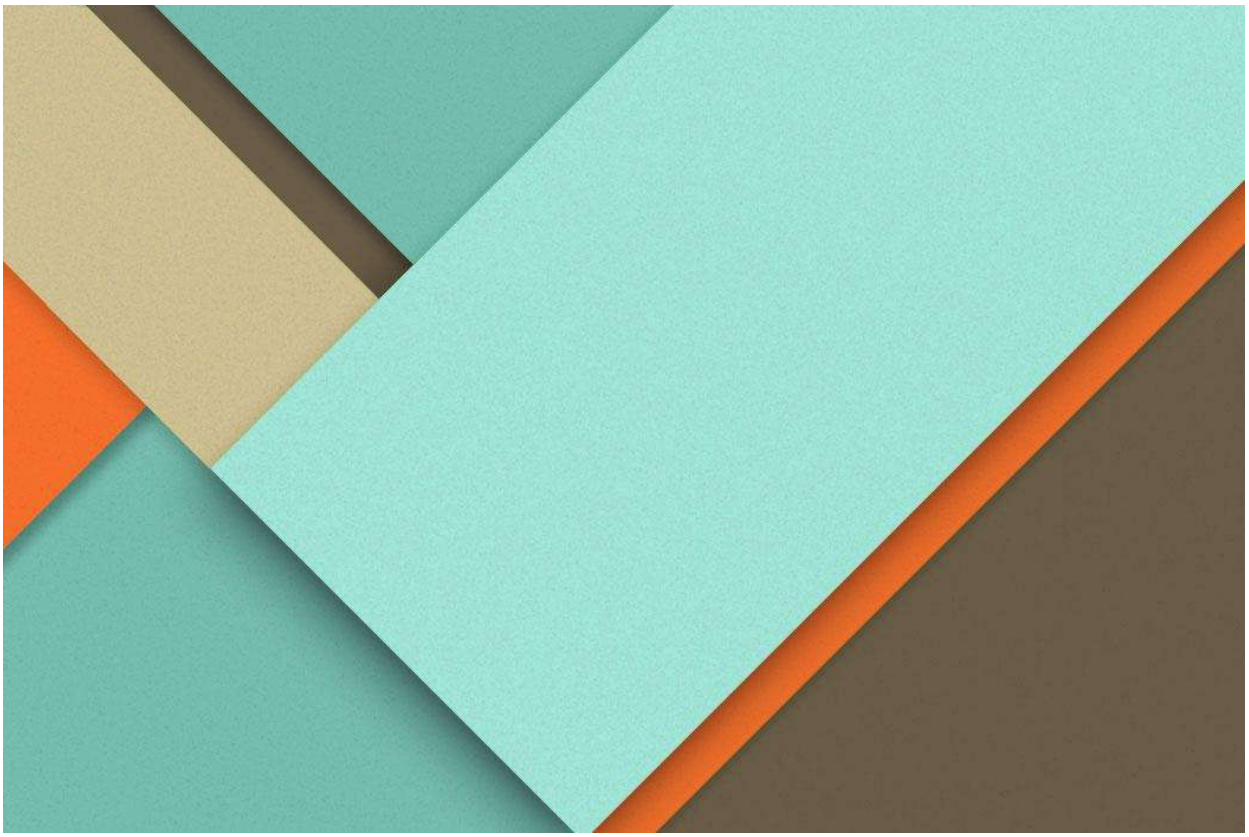


AWS EKS Implementation Guide Part -3

ELK Stack

01.01.2020

Singaravelan Palani

Minfy Technologies Pvt Ltd.

Table of Contents

Overview: Helm

What is Helm ?

Helm is a tool for managing Charts. Charts are packages of pre-configured Kubernetes resources.

Use Helm to:

- Find and use popular software packaged as Helm Charts to run in Kubernetes
- Share your own applications as Helm Charts
- Create reproducible builds of your Kubernetes applications
- Intelligently manage your Kubernetes manifest files
- Manage releases of Helm packages

Helm is a tool that streamlines installing and managing Kubernetes applications. Think of it like apt/yum/homebrew for Kubernetes.

- Helm renders your templates and communicates with the Kubernetes API
- Helm runs on your laptop, CI/CD, or wherever you want it to run.
- Charts are Helm packages that contain at least two things:
    - A description of the package (Chart.yaml)
    - One or more templates, which contain Kubernetes manifest files
- Charts can be stored on disk, or fetched from remote chart repositories (like Debian or RedHat packages)

Helm Installation in Kops Server

- Install the helm using following commands

```
$ curl https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get > install-helm.sh
$ chmod +x install-helm.sh
$ ./install-helm.sh
$ helm init
$ helm repo update
```

```
[ec2-user@ip-12-0-14-210 ~]$ curl https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get > install-helm.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7164  100  7164    0     0  25225      0 --:--:-- --:--:-- --:--:-- 25136
[ec2-user@ip-12-0-14-210 ~]$ chmod +x install-helm.sh
[ec2-user@ip-12-0-14-210 ~]$ ./install-helm.sh
Helm v2.16.1 is already latest
Run 'helm init' to configure helm.
[ec2-user@ip-12-0-14-210 ~]$
```

```
[ec2-user@ip-12-0-14-210 ~]$ helm init
Creating /home/ec2-user/.helm
Creating /home/ec2-user/.helm/repository
Creating /home/ec2-user/.helm/repository/cache
Creating /home/ec2-user/.helm/repository/local
Creating /home/ec2-user/.helm/plugins
Creating /home/ec2-user/.helm/starters
Creating /home/ec2-user/.helm/cache/archive
Creating /home/ec2-user/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$HELM_HOME has been configured at /home/ec2-user/.helm.

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users' policy.
To prevent this, run `helm init` with the --tiller-tls-verify flag.
For more information on securing your installation see: https://docs.helm.sh/using_helm/#securing-your-helm-installation
[ec2-user@ip-12-0-14-210 ~]$
```

- Helm version check

```
[ec2-user@ip-12-0-14-210 ~]$ helm  version
Client: &version.Version{SemVer:"v2.16.1", GitCommit:"bbdfe5e7803a12bbdf97e94cd847859890cf4050", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.16.1", GitCommit:"bbdfe5e7803a12bbdf97e94cd847859890cf4050", GitTreeState:"clean"}
[ec2-user@ip-12-0-14-210 ~]$
```

Sample MySQL Application Deploy using Helm Kubernetes

- MySQL Application Deploy using Helm Kubernetes following commands

```
$ helm install stable/mysql --name my-mysql-install --set mysqlPassword=cloudredhat
```

```
[ec2-user@ip-12-0-14-210 ~]$
[ec2-user@ip-12-0-14-210 ~]$ helm install stable/mysql --name my-mysql-install --set mysqlPassword=cloudredhat
Error: release my-mysql-install failed: namespaces "default" is forbidden: User "system:serviceaccount:kube-system:default" cannot get res
ource "namespaces" in API group "" in the namespace "default"
[ec2-user@ip-12-0-14-210 ~]$
```

- Here we got the error, because of we don't have a privilege access –
- We have to get the privilege access use following commands

```
$ kubectl create serviceaccount --namespace kube-system tiller
$ kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller
$ kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
```

```
[ec2-user@ip-12-0-14-210 ~]$ kubectl create serviceaccount --namespace kube-system tiller
serviceaccount/tiller created
[ec2-user@ip-12-0-14-210 ~]$ kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-syste
m:tiller
clusterrolebinding.rbac.authorization.k8s.io/tiller-cluster-rule created
[ec2-user@ip-12-0-14-210 ~]$ kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":
"tiller"}}}}'
deployment.extensions/tiller-deploy patched
[ec2-user@ip-12-0-14-210 ~]$ helm ls
Error: configmaps is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "configmaps" in API group "" in the
namespace "kube-system"
```

- Now try to install MySQL using helm

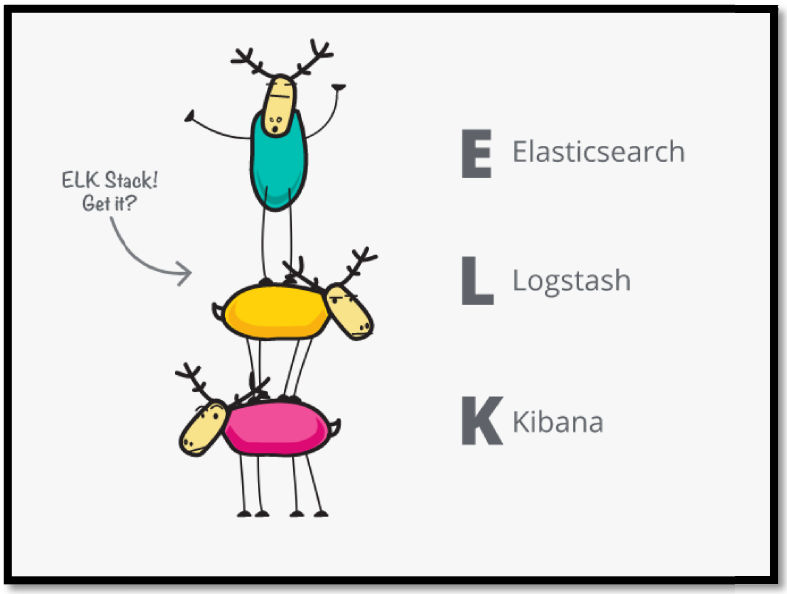
```
[ec2-user@ip-12-0-14-210 ~]$ helm install stable/mysql --name my-mysql-install --set mysqlPassword=cloudredhat
NAME:   my-mysql-install
LAST DEPLOYED: Fri Jan  3 05:11:42 2020
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME                   AGE
my-mysql-install-test  0s

==> v1/Deployment
NAME              AGE
my-mysql-install  0s

==> v1/PersistentVolumeClaim
NAME              AGE
my-mysql-install  0s

==> v1/Pod(related)
NAME                                AGE
my-mysql-install-6f896cfb8f-n7ssv   0s

==> v1/Secret
NAME              AGE
my-mysql-install  0s

==> v1/Service
NAME              AGE
```

```
To get your root password run:

    MYSQL_ROOT_PASSWORD=$(kubectl get secret --namespace default my-mysql-install -o jsonpath="{.data.mysql-root-password}" | base64 --dec
ode; echo)

To connect to your database:

1. Run an Ubuntu pod that you can use as a client:

    kubectl run -i --tty ubuntu --image=ubuntu:16.04 --restart=Never -- bash -il

2. Install the mysql client:

    $ apt-get update && apt-get install mysql-client -y

3. Connect using the mysql cli, then provide your password:
    $ mysql -h my-mysql-install -p

To connect to your database directly from outside the K8s cluster:
    MYSQL_HOST=127.0.0.1
    MYSQL_PORT=3306

    # Execute the following command to route the connection:
    kubectl port-forward svc/my-mysql-install 3306

    mysql -h ${MYSQL_HOST} -P${MYSQL_PORT} -u root -p${MYSQL_ROOT_PASSWORD}


[ec2-user@ip-12-0-14-210 ~]$
```

MySQL Application Deploy using Helm Kubernetes

```
[ec2-user@ip-12-0-14-210 ~]$ helm ls
NAME                    REVISION        UPDATED                         STATUS          CHART                                   A
PP VERSION      NAMESPACE
my-mysql-install        1               Fri Jan  3 05:11:42 2020        DEPLOYED        mysql-1.6.2                             5
.7.28           default
[ec2-user@ip-12-0-14-210 ~]$
```

Overview Kubernetes-Monitoring-System-ELK



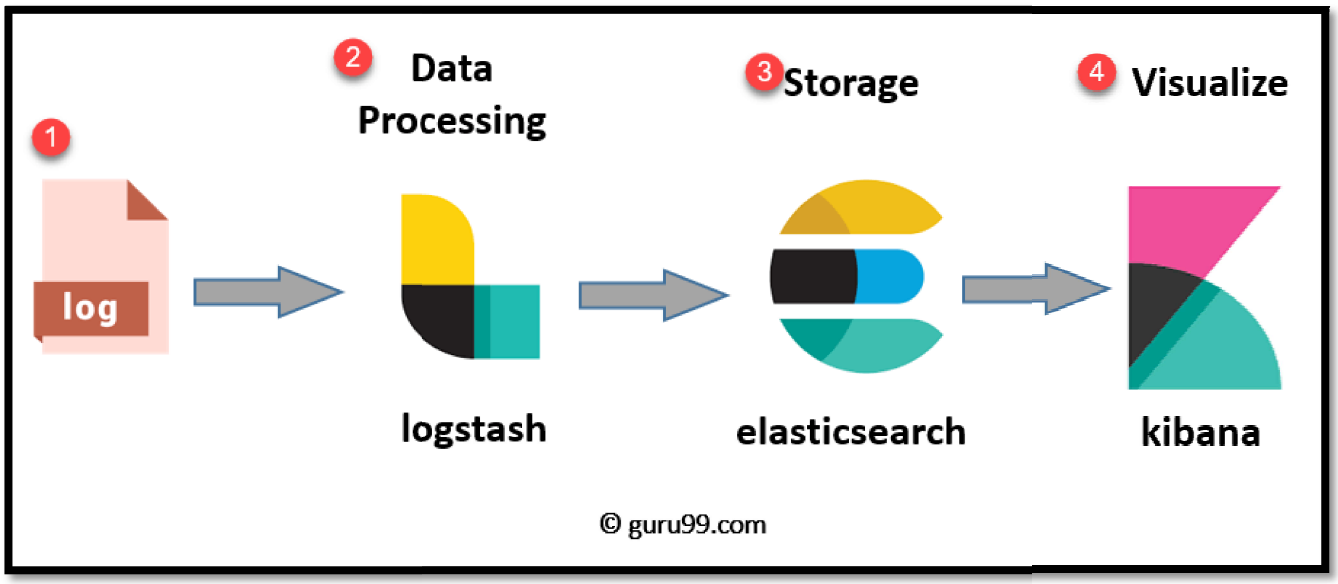
- Monitoring a Kubernetes (K8s) cluster is critical in production and there are many great tools out there like Prometheus and Grafana, but I wanted to 'play' with getting K8s metrics into the Elasticsearch, Logstash, and Kibana (ELK) stack,

What is the ELK Stack?

- The ELK Stack is a collection of three open-source products - ElasticSearch, Log stash, and Kibana. They are all developed, managed, and maintained by the company Elastic.
  1. **E** stands for ElasticSearch: used for storing logs
  2. **L** stands for LogStash : used for both shipping as well as processing and storing logs
  3. **K** stands for Kibana: is a visutalization tool (a web interface) which is hosted through Nginx or Apache
- ELK Stack is designed to allow users to take to data from any source, in any format, and to search, analyze, and visualize that data in real time.
- ELK provides centralized logging that be useful when attempting to identify problems with servers or applications. It allows you to search all your logs in a single place. It also helps to find issues that occur in multiple servers by connecting their logs during a specific time frame.

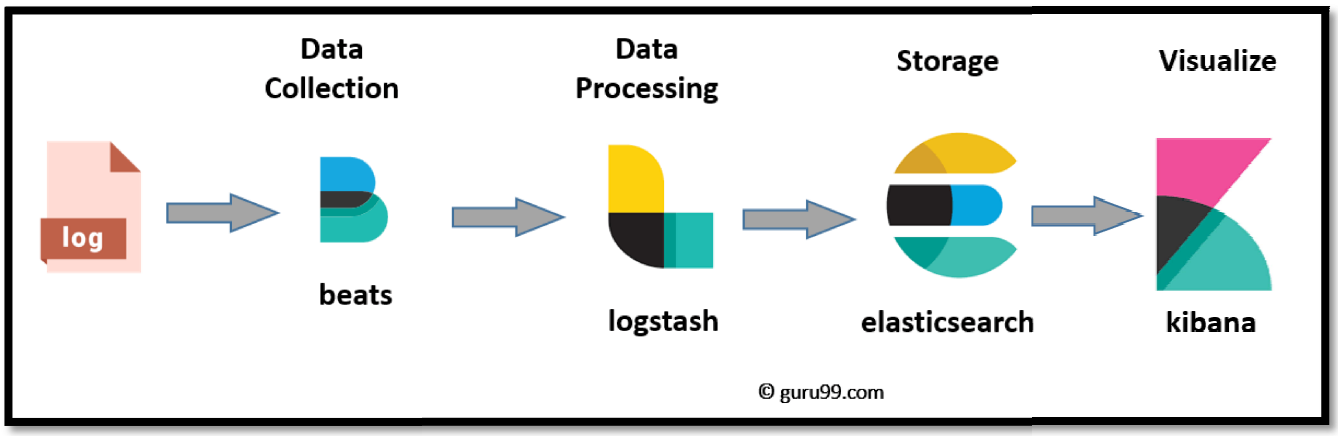
ELK Stack Architecture?

Here is the simple architecture of ELK stack

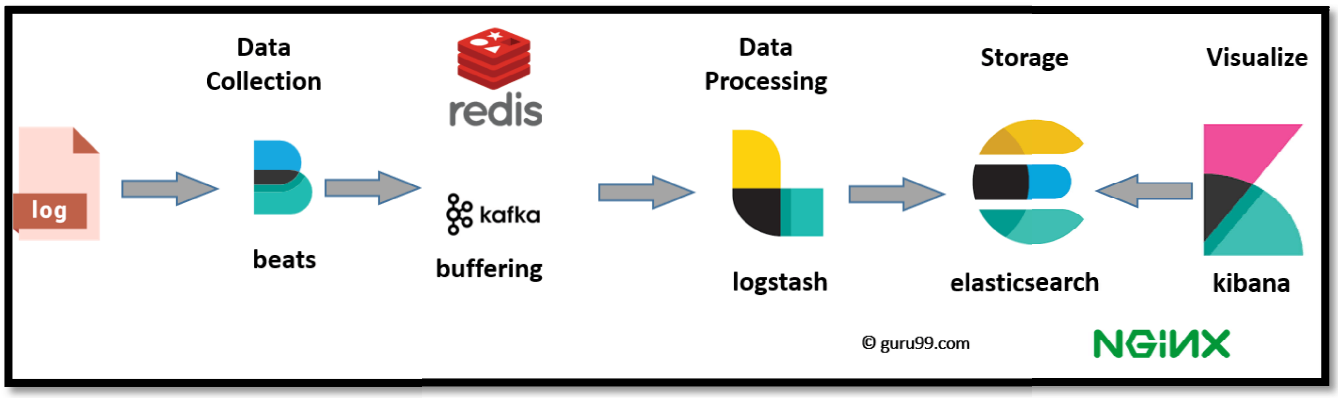


- **Logs**: Server logs that need to be analyzed are identified
- **Logstash**: Collect logs and events data. It even parses and transforms data
- **ElasticSearch**: The transformed data from Logstash is Store, Search, and indexed.
- **Kibana**: Kibana uses ElasticSearch DB to Explore, Visualize, and Share

However, one more component is needed or Data collection called Beats. This led Elastic to rename ELK as the Elastic Stack.



While dealing with very large amounts of data, you may need Kafka, RabbitMQ for buffering and resilience. For security, nginx can be used.



Overview - What is ElasticSearch?

What is ElasticSearch?

- Elasticsearch is a NoSQL database. It is based on Lucene search engine, and it is built with RESTful APIS. It offers simple deployment, maximum reliability, and easy management. It also offers advanced queries to perform detail analysis and stores all the data centrally. It is helpful for executing a quick search of the documents.
- Elasticsearch also allows you to store, search and analyze big volume of data. It is mostly used as the underlying engine to powers applications that completed search requirements. It has been adopted in search engine platforms for modern web and mobile applications. Apart from a quick search, the tool also offers complex analytics and many advanced features.

Features of Elastic search

- Open source search server is written using Java
- Used to index any kind of heterogeneous data
- Has REST API web-interface with JSON output
- Full-Text Search
- Near Real Time (NRT) search
- Sharded, replicated searchable, JSON document store
- Schema-free, REST & JSON based distributed document store
- Multi-language & Geolocation support

Advantages of ElasticSearch

- Store schema-less data and also creates a schema for your data
- Manipulate your data record by record with the help of Multi-document APIs
- Perform filtering and querying your data for insights
- Based on Apache Lucene and provides RESTful API
- Provides horizontal scalability, reliability, and multitenant capability for real time use of indexing to make it faster search
- Helps you to scale vertically and horizontally
-

Important Terms used in Elastic Search

Term	Usage
Cluster	A cluster is a collection of nodes which together holds data and provides joined indexing and search capabilities.
Node	A node is an Elasticsearch Instance. It is created when an Elasticsearch instance begins.
Index	An index is a collection of documents which has similar characteristics. e.g., customer data, product catalog. It is very useful while performing indexing, search, update, and delete operations. It allows you to define as many indexes in one single cluster.
Document	It is the basic unit of information which can be indexed. It is expressed in JSON (key: value) pair. '{"user": "nullcon"}'. Every single Document is associated with a type and a unique id.
Shard	Every index can be split into several shards to be able to distribute data. The shard is the atomic part of an index, which can be distributed over the cluster if you want to add more nodes.

Overview - What is Logstash?

- Logstash is the data collection pipeline tool. It collects data inputs and feeds into the Elasticsearch. It gathers all types of data from the different source and makes it available for further use.
- Logstash can unify data from disparate sources and normalize the data into your desired destinations. It allows you to cleanse and democratize all your data for analytics and visualization of use cases.
- It consists of three components:
  - ✓ Input: passing logs to process them into machine understandable format
  - ✓ Filters: It is a set of conditions to perform a particular action or event
  - ✓ Output: Decision maker for processed event or log

Features of Logstash

- ✓ Events are passed through each phase using internal queues
- ✓ Allows different inputs for your logs
- ✓ Filtering/parsing for your logs

Advantage of Logstash

- ✓ Offers centralize the data processing
- ✓ It analyzes a large variety of structured/unstructured data and events
- ✓ Offers plugins to connect with various types of input sources and platforms

Overview - What is Kibana?

- ✓ Kibana is a data visualization which completes the ELK stack. This tool is used for visualizing the Elasticsearch documents and helps developers to have a quick insight into it. Kibana dashboard offers various interactive diagrams, geospatial data, and graphs to visualize complex quires.
- ✓ It can be used for search, view, and interact with data stored in Elasticsearch directories. Kibana helps you to perform advanced data analysis and visualize your data in a variety of tables, charts, and maps.
- ✓ In Kibana there are different methods for performing searches on your data.

Here are the most common search types:

Search Type	Usage
Free text searches	It is used for searching a specific string
Field-level searches	It is used for searching for a string within a specific field
Logical statements	It is used to combine searches into a logical statement.
Proximity searches	It is used for searching terms within specific character proximity.

Features of Kibana

- Powerful front-end dashboard which is capable of visualizing indexed information from the elastic cluster
- Enables real-time search of indexed information
- You can search, View, and interact with data stored in Elasticsearch
- Execute queries on data & visualize results in charts, tables, and maps
- Configurable dashboard to slice and dice logstash logs in elasticsearch
- Capable of providing historical data in the form of graphs, charts, etc.
- Real-time dashboards which is easily configurable
- Enables real-time search of indexed information

Advantages and Disadvantages of Kibana

- Easy visualizing
- Fully integrated with ElasticSearch
- Visualization tool
- Offers real-time analysis, charting, summarization, and debugging capabilities
- Provides instinctive and user-friendly interface
- Allows sharing of snapshots of the logs searched through
- Permits saving the dashboard and managing multiple dashboards

Deploy the ELK Stack

- Helm provides excellent package management of K8s and is always my first choice for deployment so I started there for the ELK stack, and sure enough, there's a Helm chart for that (elastic-stack).
  Ref : https://github.com/helm/charts/tree/master/stable/elastic-stack
- You can easily override/merge the default values of a chart with your own values file so I created one that enabled filebeat and pointed it to Logstash and the Elasticsearch client and pointed Logstash to the Elasticsearch client:

elastic-stack.yaml

```
logstash:
  enabled: true
  elasticsearch:
    host: elastic-stack-elasticsearch-client

filebeat:
  enabled: true
  config:
    output.file.enabled: false
    output.logstash:
      hosts: ["elastic-stack-logstash:5044"]
  indexTemplateLoad:
    - elastic-stack-elasticsearch-client:9200
```

Then deployed with those values:

```
$ helm install --name elastic-stack stable/elastic-stack -f elastic-stack.yaml
```

You'll see some output detailing what you've deployed and you can check the status of the pods with kubectl

```
$ kubectl get pods -l "release=elastic-stack"
```

```
[ec2-user@ip-12-0-14-210 ELK]$ kubectl get pods -l "release=elastic-stack"
NAME                                            READY   STATUS    RESTARTS   AGE
elastic-stack-elasticsearch-client-5567cc4fbf-2hk9r   1/1     Running   0          74m
elastic-stack-elasticsearch-client-5567cc4fbf-kcxm7   1/1     Running   0          74m
elastic-stack-elasticsearch-data-0                    1/1     Running   0          74m
elastic-stack-elasticsearch-data-1                    1/1     Running   0          71m
elastic-stack-elasticsearch-master-0                  1/1     Running   0          74m
elastic-stack-elasticsearch-master-1                  1/1     Running   0          73m
elastic-stack-elasticsearch-master-2                  1/1     Running   0          72m
elastic-stack-filebeat-6qw26                          2/2     Running   0          74m
elastic-stack-filebeat-dz67k                          2/2     Running   0          74m
elastic-stack-filebeat-kh5v8                          2/2     Running   0          74m
elastic-stack-kibana-68cc66b686-gxczh                 1/1     Running   0          74m
elastic-stack-logstash-0                              1/1     Running   0          74m
[ec2-user@ip-12-0-14-210 ELK]$
```

Edit the elastic-stack-kibana

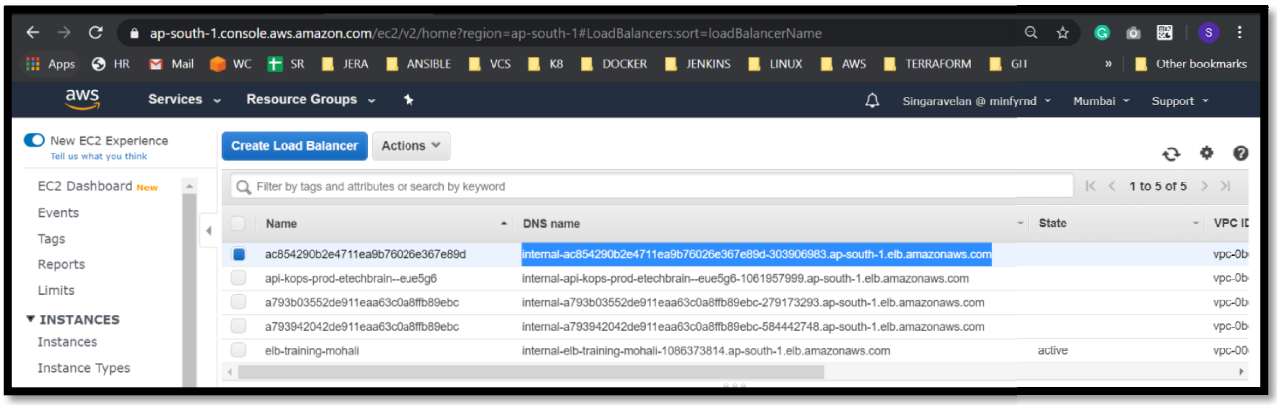
Here change the Type ClusterIP to LoadBalncer

```
$ kubectl edit svc elastic-stack-kibana
```

We got the LoadBalncer URL

```
NAME                                    TYPE           CLUSTER-IP       EXTERNAL-IP
                        PORT(S)          AGE
service/elastic-stack-elasticsearch-client      ClusterIP      100.64.195.25    <none>
                        9200/TCP         65m
service/elastic-stack-elasticsearch-discovery   ClusterIP      None             <none>
                        9300/TCP         65m
service/elastic-stack-filebeat-metrics          ClusterIP      100.68.162.41    <none>
                        9479/TCP         65m
service/elastic-stack-kibana                    LoadBalancer   100.69.223.7     internal-ac854290b2e4711ea9b76026e367e89d-303906983.ap-sou
th-1.elb.amazonaws.com   5601:31537/TCP   65m
service/elastic-stack-logstash                  ClusterIP      100.64.116.53    <none>
                        5044/TCP         65m
service/kube-state-metrics                      ClusterIP      100.64.226.122   <none>
                        8080/TCP         34m
service/kubernetes                              ClusterIP      100.64.0.1       <none>
                        443/TCP          12h

NAME                                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/elastic-metricbeat    6         6         6       6            3           <none>          30m
daemonset.apps/elastic-stack-filebeat   3         3         3       3            1           <none>          65m
```

In Console view



Deploy Metric beat

- Metricbeat collects (easy guess) metrics and, through its Kubenetes modules, can get data from K8s nodes and the API. Sure enough, (you know where this is going…) there's a chart for that (metricbeat).

In this case we have to override values to:

- ✓ Use the open source version of the metricbeat Docker image (metricbeat-oss)
- ✓ Point the daemonset Kubernetes module to the proper K8s SSL metrics host (and disable SSL verification)
- ✓ Point the deployment to the elasticsearch client and kibana in the cluster

elastic-metricbeat.yaml

```
image:
 repository: docker.elastic.co/beats/metricbeat-oss
daemonset:
 config:
  output.file:
  output.elasticsearch:
   hosts: ["elastic-stack-elasticsearch-client:9200"]
 modules:
  kubernetes:
   config:
    - module: kubernetes
      metricsets:
       - node
       - system
       - pod
       - container
       - volume
      period: 10s
      host: ${NODE_NAME}
      hosts: ["https://${HOSTNAME}:10250"]
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
```

```
        ssl.verification_mode: "none"
    deployment:
     config:
      output.file:
      output.elasticsearch:
       hosts: ["elastic-stack-elasticsearch-client:9200"]
      setup.kibana:
       host: "elastic-stack-kibana:5601"
      setup.dashboards.enabled: true
```

```
$ helm $ helm install --name kube-state-metrics stable/kube-state-metrics
elastic-metricbeat.yaml
```

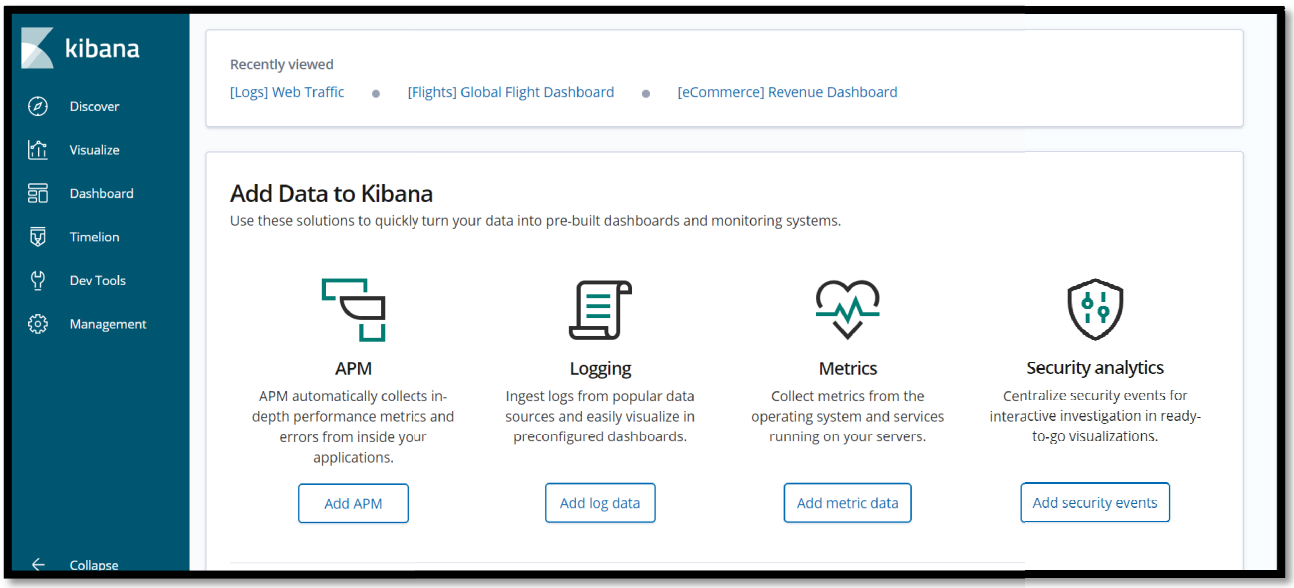
Kubectl get all

```
[ec2-user@ip-12-0-14-210 ELK]$ kubectl get all
NAME                                                READY   STATUS    RESTARTS   AGE
pod/elastic-metricbeat-84b64f64fb-zn5vj             1/1     Running   0          30m
pod/elastic-metricbeat-b92dg                        1/1     Running   0          30m
pod/elastic-metricbeat-bltfw                        1/1     Running   0          30m
pod/elastic-metricbeat-dzfdf                        1/1     Running   0          30m
pod/elastic-metricbeat-fcn42                        1/1     Running   0          30m
pod/elastic-metricbeat-gxmr5                        1/1     Running   0          30m
pod/elastic-metricbeat-v9fjd                        1/1     Running   0          30m
pod/elastic-stack-elasticsearch-client-5567cc4fbf-2hk9r   1/1     Running   0          65m
pod/elastic-stack-elasticsearch-client-5567cc4fbf-kcxm7   1/1     Running   0          65m
pod/elastic-stack-elasticsearch-data-0              1/1     Running   0          65m
pod/elastic-stack-elasticsearch-data-1              1/1     Running   0          62m
pod/elastic-stack-elasticsearch-master-0            1/1     Running   0          65m
pod/elastic-stack-elasticsearch-master-1            1/1     Running   0          63m
pod/elastic-stack-elasticsearch-master-2            1/1     Running   0          63m
pod/elastic-stack-filebeat-6qw26                    2/2     Running   0          65m
pod/elastic-stack-filebeat-dz67k                    2/2     Running   0          65m
pod/elastic-stack-filebeat-kh5v8                    2/2     Running   0          65m
pod/elastic-stack-kibana-68cc66b686-gxczh           1/1     Running   0          65m
pod/elastic-stack-logstash-0                        1/1     Running   0          65m
pod/kube-state-metrics-586c44db96-86z96             1/1     Running   0          34m
```

```
NAME                                        TYPE           CLUSTER-IP       EXTERNAL-IP
                        PORT(S)          AGE
service/elastic-stack-elasticsearch-client      ClusterIP      100.64.195.25    <none>
                        9200/TCP         65m
service/elastic-stack-elasticsearch-discovery   ClusterIP      None             <none>
                        9300/TCP         65m
service/elastic-stack-filebeat-metrics          ClusterIP      100.68.162.41    <none>
                        9479/TCP         65m
service/elastic-stack-kibana                    LoadBalancer   100.69.223.7     internal-ac854290b2e4711ea9b76026e367e89d-303906983.ap-sou
th-1.elb.amazonaws.com   5601:31537/TCP   65m
service/elastic-stack-logstash                  ClusterIP      100.64.116.53    <none>
                        5044/TCP         65m
service/kube-state-metrics                      ClusterIP      100.64.226.122   <none>
                        8080/TCP         34m
service/kubernetes                              ClusterIP      100.64.0.1       <none>
                        443/TCP          12h

NAME                                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/elastic-metricbeat     6         6         6       6            3           <none>          30m
daemonset.apps/elastic-stack-filebeat 3         3         3       3            1           <none>          65m
```

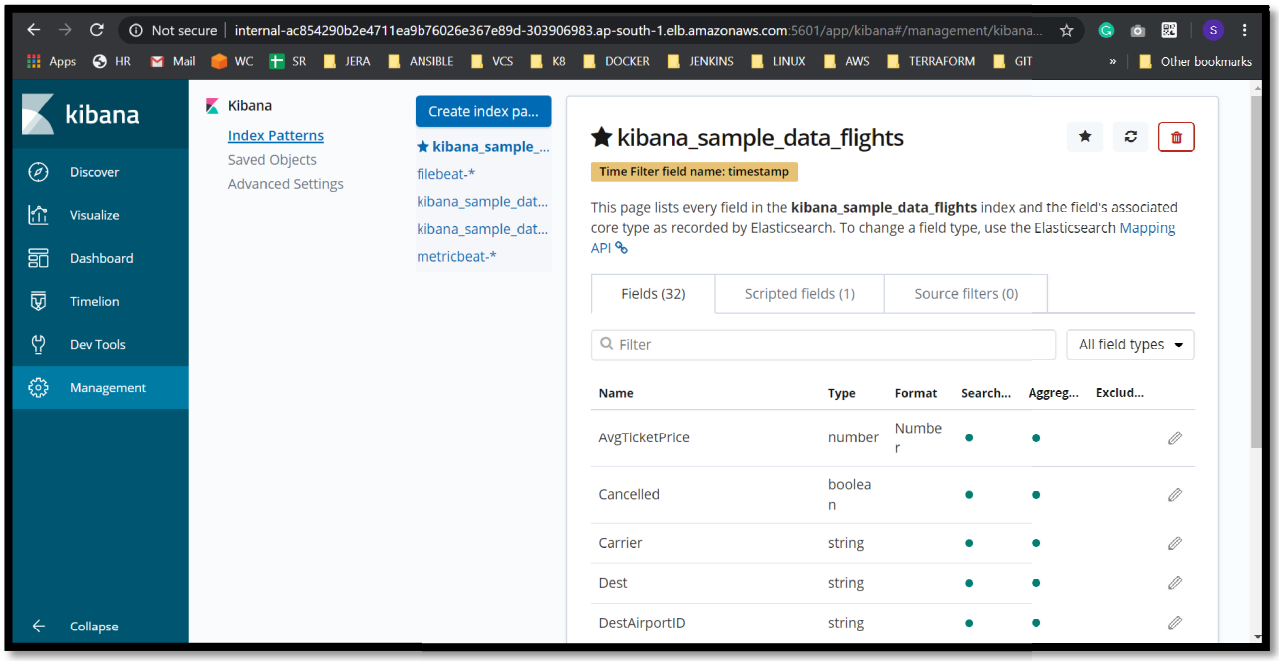
```
NAME                                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/elastic-metricbeat     6         6         6       6            3           <none>          30m
daemonset.apps/elastic-stack-filebeat 3         3         3       3            1           <none>          65m

NAME                                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/elastic-metricbeat               1/1     1            1           30m
deployment.apps/elastic-stack-elasticsearch-client   2/2     2            2           65m
deployment.apps/elastic-stack-kibana             1/1     1            1           65m
deployment.apps/kube-state-metrics               1/1     1            1           34m

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/elastic-metricbeat-84b64f64fb          1         1         1       30m
replicaset.apps/elastic-stack-elasticsearch-client-5567cc4fbf   2         2         2       65m
replicaset.apps/elastic-stack-kibana-68cc66b686        1         1         1       65m
replicaset.apps/kube-state-metrics-586c44db96          1         1         1       34m

NAME                                          READY   AGE
statefulset.apps/elastic-stack-elasticsearch-data     2/2     65m
statefulset.apps/elastic-stack-elasticsearch-master   3/3     65m
statefulset.apps/elastic-stack-logstash               1/1     65m
[ec2-user@ip-12-0-14-210 ELK]$
```

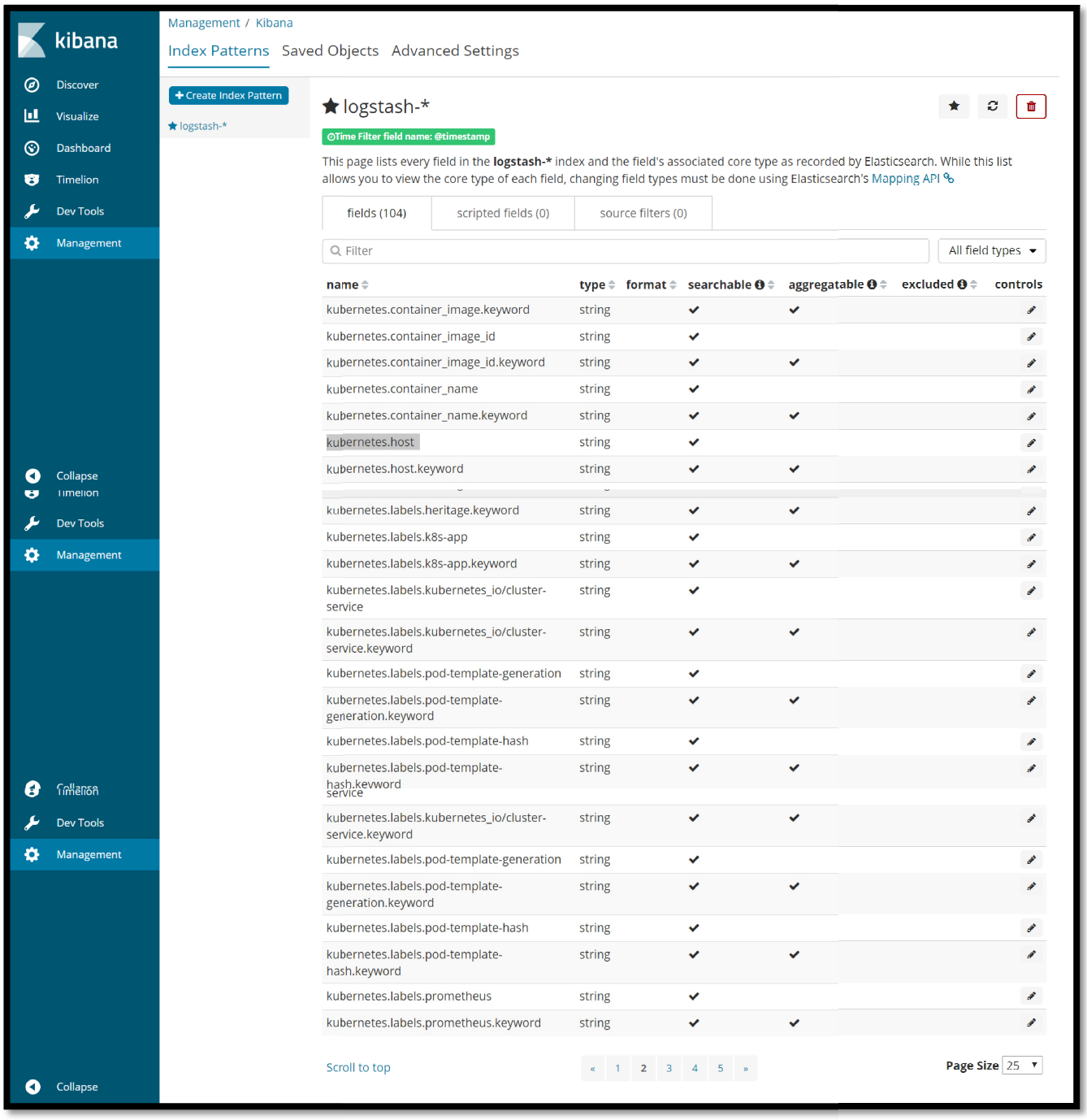
Access the kibana load balancer URL



Sample Log web traffic - dashboard for kibana.

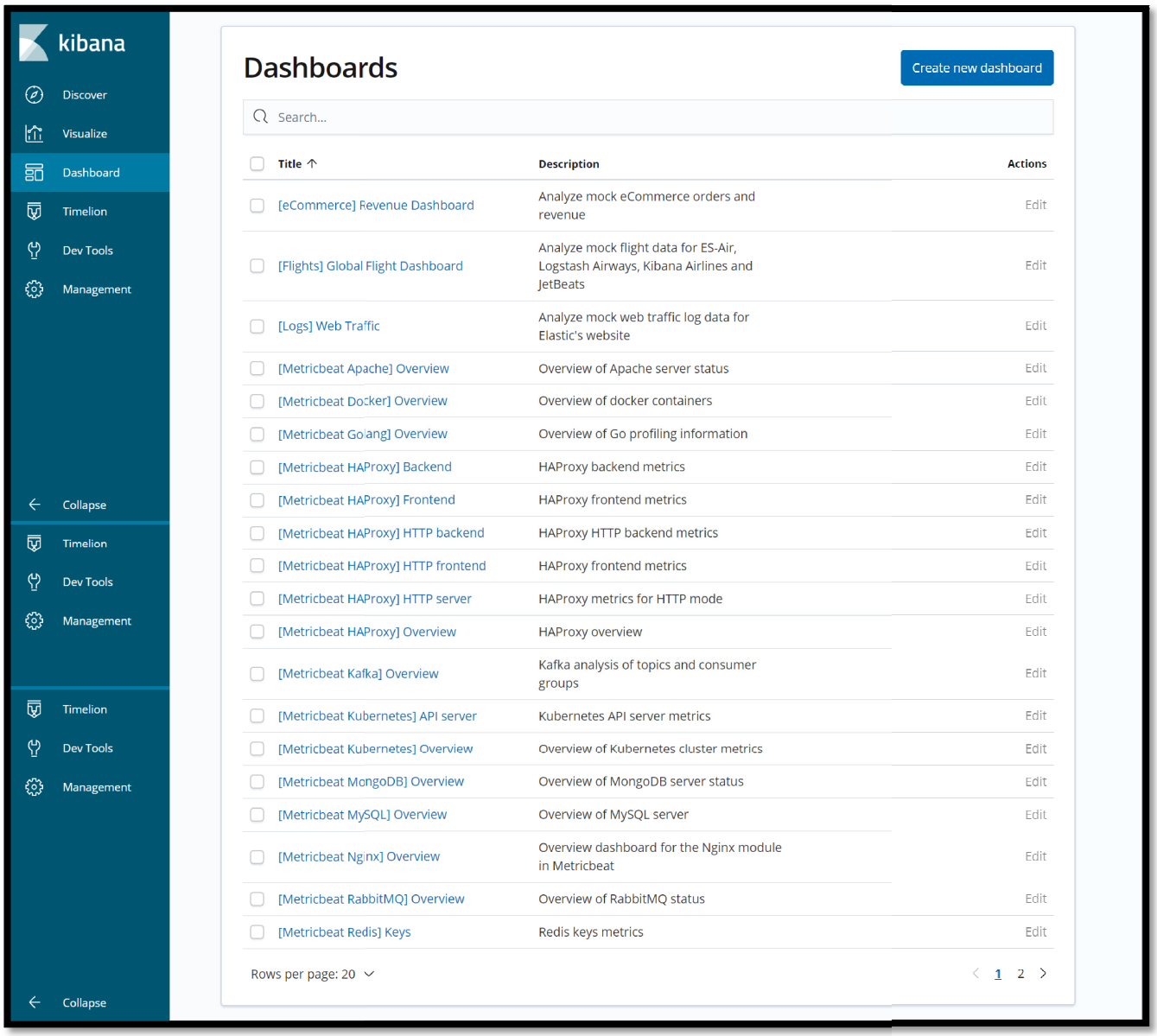
You can then of course create an index pattern for filebeat and browse those entries:



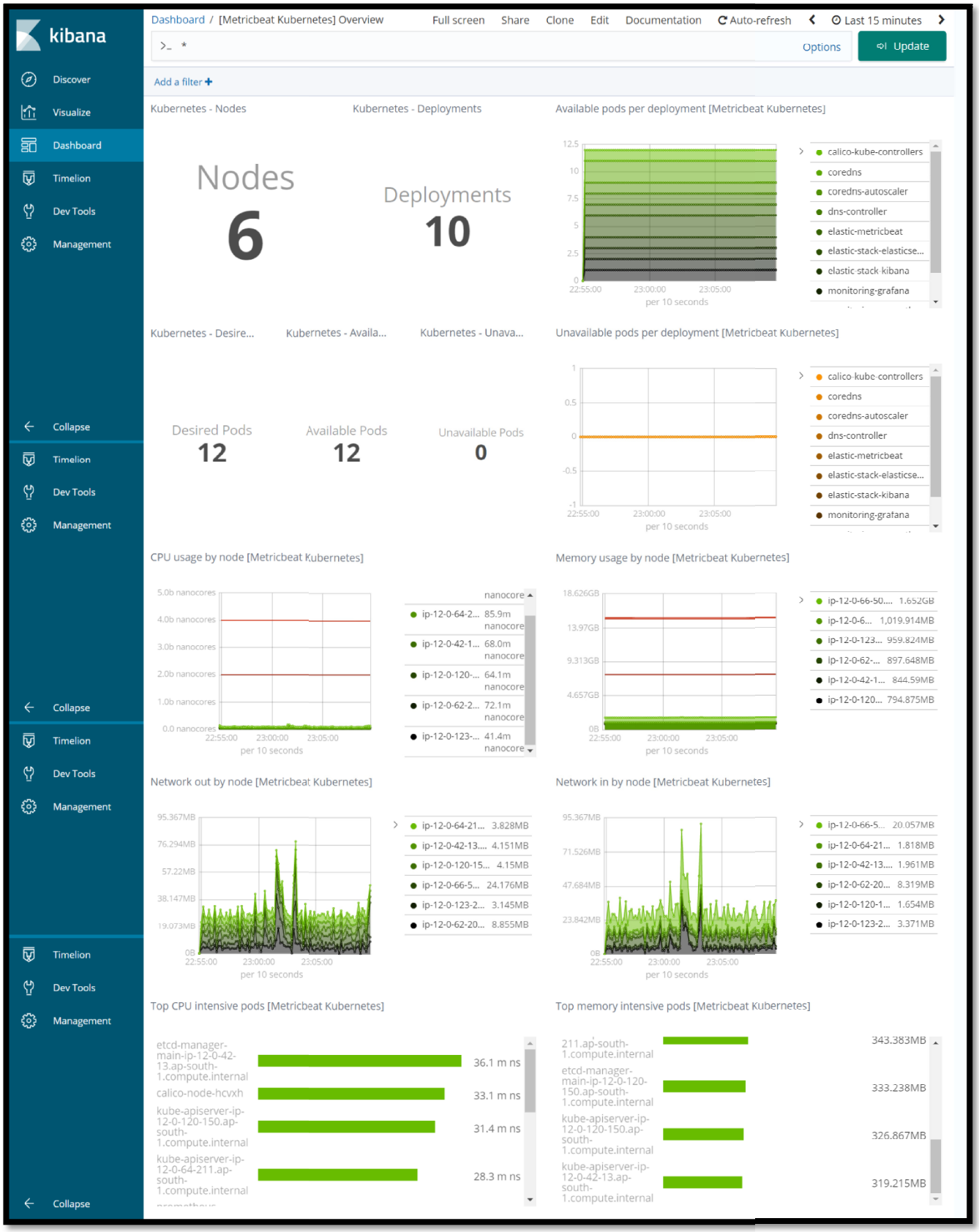


Setup Metric beat Index and K8s Dashboard

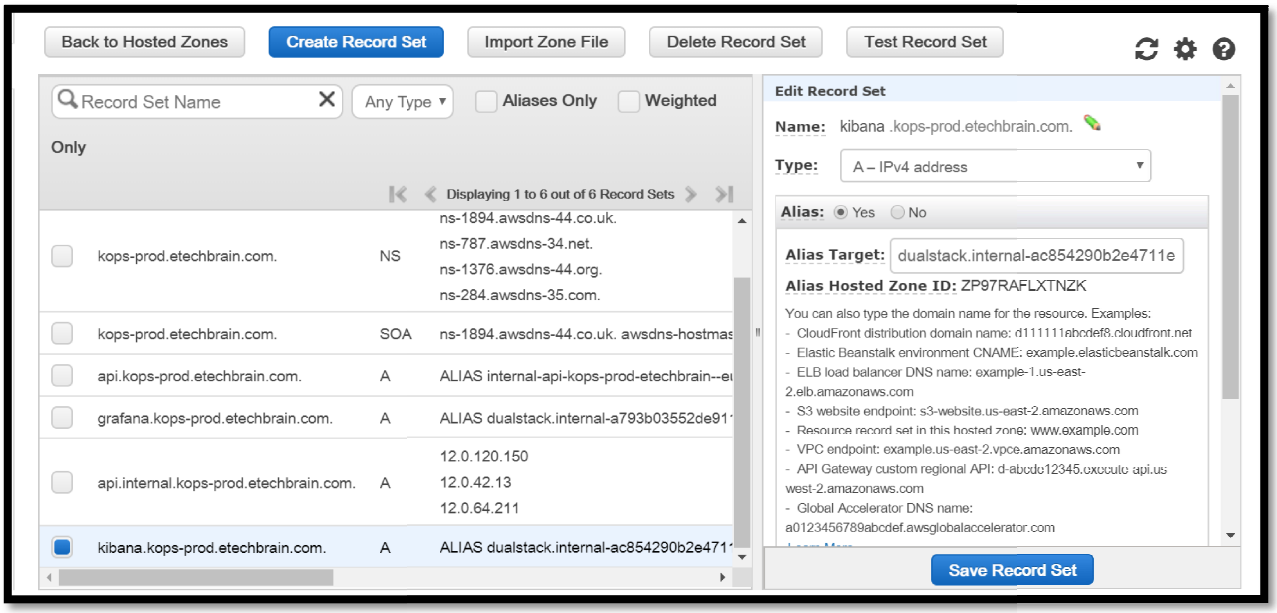
- ✓ Metricbeat will start feeding the K8s metrics to Elasticsearch and should setup another index pattern and load several dashboards into Kibana.
- ✓ You can then open the [Metricbeat Kubernetes] Overview dashboard:


kibana
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
Collapse
Timelion
Dev Tools
Management
Timelion
Dev Tools
Management
Collapse
Dashboards
Create new dashboard
Search...
Title
Description
Actions
[eCommerce] Revenue Dashboard
Analyze mock eCommerce orders and revenue
Edit
[Flights] Global Flight Dashboard
Analyze mock flight data for ES-Air, Logstash Airways, Kibana Airlines and JetBeats
Edit
[Logs] Web Traffic
Analyze mock web traffic log data for Elastic's website
Edit
[Metricbeat Apache] Overview
Overview of Apache server status
Edit
[Metricbeat Docker] Overview
Overview of docker containers
Edit
[Metricbeat Golang] Overview
Overview of Go profiling information
Edit
[Metricbeat HAProxy] Backend
HAProxy backend metrics
Edit
[Metricbeat HAProxy] Frontend
HAProxy frontend metrics
Edit
[Metricbeat HAProxy] HTTP backend
HAProxy HTTP backend metrics
Edit
[Metricbeat HAProxy] HTTP frontend
HAProxy frontend metrics
Edit
[Metricbeat HAProxy] HTTP server
HAProxy metrics for HTTP mode
Edit
[Metricbeat HAProxy] Overview
HAProxy overview
Edit
[Metricbeat Kafka] Overview
Kafka analysis of topics and consumer groups
Edit
[Metricbeat Kubernetes] API server
Kubernetes API server metrics
Edit
[Metricbeat Kubernetes] Overview
Overview of Kubernetes cluster metrics
Edit
[Metricbeat MongoDB] Overview
Overview of MongoDB server status
Edit
[Metricbeat MySQL] Overview
Overview of MySQL server
Edit
[Metricbeat Nginx] Overview
Overview dashboard for the Nginx module in Metricbeat
Edit
[Metricbeat RabbitMQ] Overview
Overview of RabbitMQ status
Edit
[Metricbeat Redis] Keys
Redis keys metrics
Edit
Rows per page: 20
1
2

Kubernetes cluster dashboard



Add a loadbalncer in Roure53 Domain name



http://kinana.kpos-prod.etechbrain.com:5601