

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?	2
Helm Installation in Kops Server	2
Sample MySQL Application Deploy using Helm Kubernetes	3

Overview kubernetes-Monitoring-System-ELK

What is the ELK Stack?	6
ELK Stack Architecture?	7

Overview - What is ElasticSearch?

What is ElasticSearch?	8
Features of Elastic search	8
Advantages of ElasticSearch	9
Important Terms used in Elastic Search	9

Overview - What is Logstash?

What is Logstash?	10
Features of Logstash	10
Advantage of Logstash	10

Overview - What is Kibana?

What is Kibana?	10
Features of Kibana	11
Advantages and Disadvantages of Kibana	11
Deploy the ELK Stack	12
Deploy Metric beat	13
Setup Metric beat & Logstash Index and K8s Dashboard	19

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     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 resource "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
```

```
To get your root password run:
  MYSQL_ROOT_PASSWORD=$(kubectl get secret --namespace default my-mysql-install -o jsonpath=".data.mysql-root-password" | base64 --decode; 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

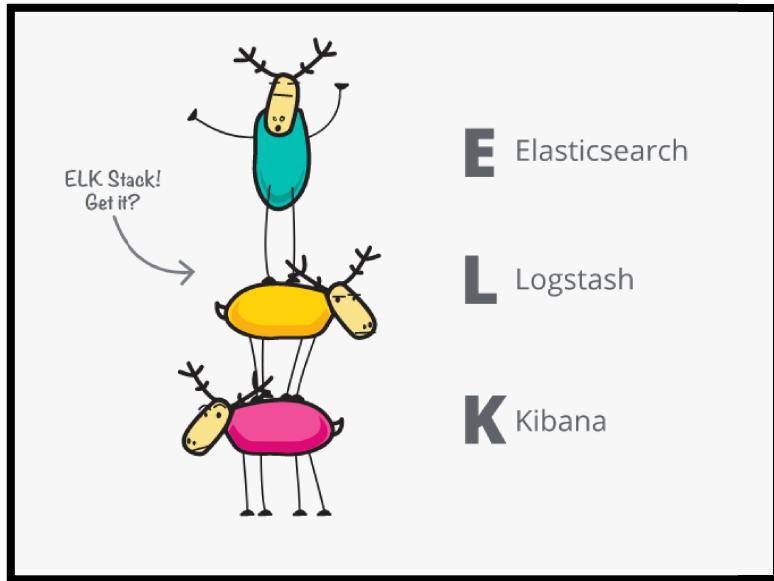
  mysql -h ${MYSQL_HOST} -P${MYSQL_PORT} -u root -p${MYSQL_ROOT_PASSWORD}

[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
PP VERSION    NAMESPACE
my-mysql-install  1          Fri Jan  3 05:11:42 2020  DEPLOYED   mysql-1.6.2
.7.28         default
[ec2-user@ip-12-0-14-210 ~]$
```

Overview Kubernetes-Monitoring-System-ELK



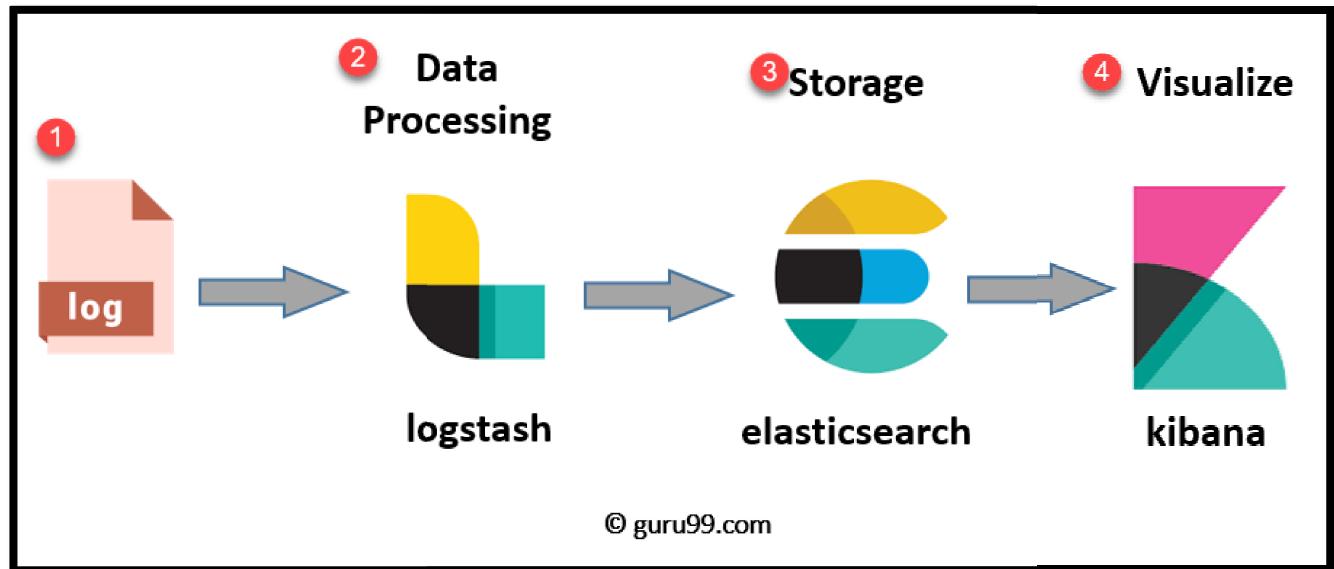
- Monitoring a Kubernetes (K8s) cluster is critical in production and there are many great tools out there like Prometheus and Grafana, but I wanted to 'play' with getting K8s metrics into the Elasticsearch, Logstash, and Kibana (ELK) stack,

What is the ELK Stack?

- The ELK Stack is a collection of three open-source products - Elasticsearch, Logstash, 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 visualization tool (a web interface) which is hosted through Nginx or Apache
- ELK Stack is designed to allow users to take data from any source, in any format, and to search, analyze, and visualize that data in real time.
- ELK provides centralized logging that is 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.

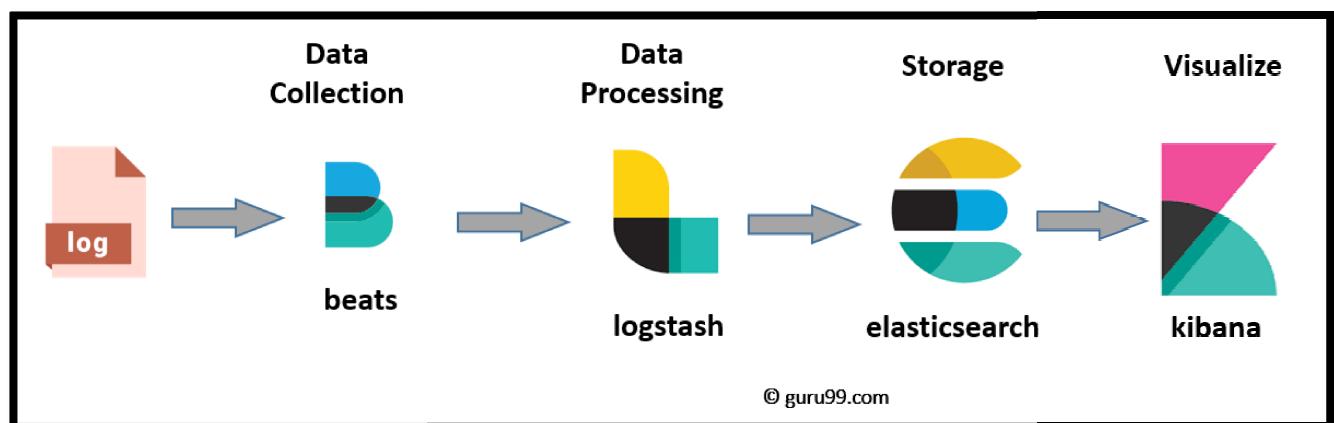
ELK Stack Architecture?

Here is the simple architecture of ELK stack

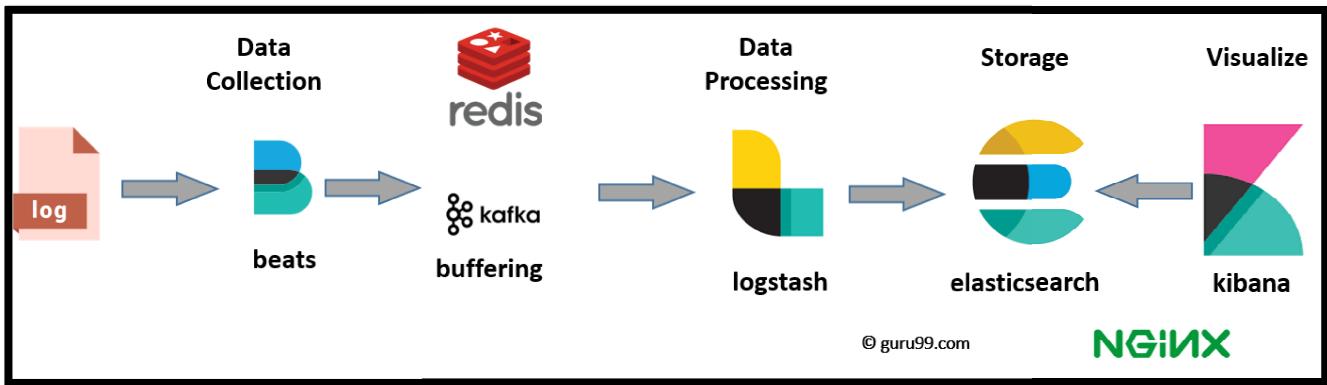


- **Logs:** Server logs that need to be analyzed are identified
- **Logstash:** Collect logs and events data. It even parses and transforms data
- **ElasticSearch:** The transformed data from Logstash is Store, Search, and indexed.
- **Kibana:** Kibana uses ElasticSearch DB to Explore, Visualize, and 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-kc xm7   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	PORT(S)	TYPE	CLUSTER-IP	EXTERNAL-IP
service/elastic-stack-elasticsearch-client	9200/TCP	ClusterIP	100.64.195.25	<none>
service/elastic-stack-elasticsearch-discovery	9300/TCP	ClusterIP	None	<none>
service/elastic-stack-filebeat-metrics	9479/TCP	ClusterIP	100.68.162.41	<none>
service/elastic-stack-kibana	5601:31537/TCP	LoadBalancer	100.69.223.7	internal-ac854290b2e4711ea9b76026e367e89d-303906983.ap-sou
service/elastic-stack-logstash	5044/TCP	ClusterIP	100.64.116.53	<none>
service/kube-state-metrics	8080/TCP	ClusterIP	100.64.226.122	<none>
service/kubernetes	443/TCP	ClusterIP	100.64.0.1	<none>
NAME	DESIRED	CURRENT	READY	UP-TO-DATE
daemonset.apps/elastic-metricbeat	6	6	6	6
daemonset.apps/elastic-stack-filebeat	3	3	3	3
NAME	AGE	STATE	SELECTOR	AGE

In Console view

The screenshot shows the AWS Lambda function configuration page. The top navigation bar includes links for AWS Lambda, Services, Resource Groups, and other AWS services like S3, CloudWatch, and CloudFormation. The main content area is titled 'Create Function' and contains several tabs: General, Code, Runtime, Handler, and Test. Under the General tab, there are fields for 'Function name' (set to 'minfynd'), 'Memory size' (set to 128), 'Timeout' (set to 3), and 'Runtime' (set to 'Node.js 12.x'). The 'Handler' field is set to 'index.handler'. The 'Code' tab shows a 'Upload' button and a 'Lambda@edge' dropdown. The 'Test' tab has a 'Test' button. On the left side, there's a sidebar with links for 'New Lambda Experience', 'Events', 'Tags', 'Reports', 'Limits', and 'INSTANCES' (which is expanded to show 'Instances' and 'Instance Types'). The bottom right corner of the page shows the user's name 'Singaravelan @ minfynd'.

Deploy Metric beat

- Metricbeat collects (easy guess) metrics and, through its Kubernetes 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

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

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

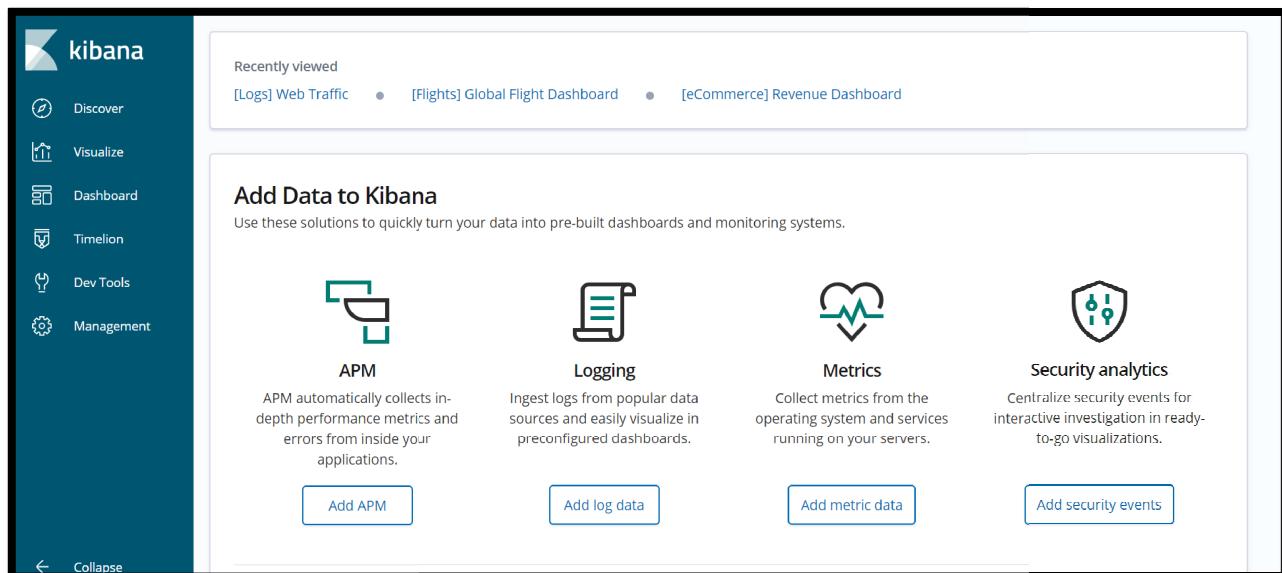
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/elastic-metricbeat	1/1	1	1	30m
deployment.apps/elastic-stack-elasticsearch-client	2/2	2	2	65m
deployment.apps/elastic-stack-kibana	1/1	1	1	65m
deployment.apps/kube-state-metrics	1/1	1	1	34m

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



The screenshot shows the Kibana interface with a sidebar on the left containing links for Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The main area displays a 'Recently viewed' section with three items: [Logs] Web Traffic, [Flights] Global Flight Dashboard, and [eCommerce] Revenue Dashboard. Below this is a 'Add Data to Kibana' section with four categories:

- APM**: Describes APM as automatically collecting in-depth performance metrics and errors from inside your applications. Includes an 'Add APM' button.
- Logging**: Describes Logging as ingest logs from popular data sources and easily visualize in preconfigured dashboards. Includes an 'Add log data' button.
- Metrics**: Describes Metrics as collect metrics from the operating system and services running on your servers. Includes an 'Add metric data' button.
- Security analytics**: Describes Security analytics as centralize security events for interactive investigation in ready-to-go visualizations. Includes an 'Add security events' button.

Sample Log web traffic - dashboard for kibana.



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

The screenshot shows the Kibana Management interface. On the left sidebar, under the 'Management' section, the 'Index Patterns' option is selected. The main content area displays the 'kibana_sample_data_flights' index pattern. At the top, there's a title bar with a star icon, a refresh icon, and a delete icon. Below it, a note says 'Time Filter field name: timestamp'. A table lists fields with their types and searchability status.

Name	Type	Format	Search...	Aggreg...	Exclud...
AvgTicketPrice	number	Number	●	●	✎
Cancelled	boolean		●	●	✎
Carrier	string		●	●	✎
Dest	string		●	●	✎
DestAirportID	string		●	●	✎

The screenshot shows the Kibana Management interface for the 'Index Patterns' section. The left sidebar has a 'Management' tab selected. The main area displays the 'logstash-*' index pattern, which contains 104 fields. The fields are listed in a table with columns for name, type, format, searchable, aggregatable, excluded, and controls. Most fields are of type 'string' or 'keyword'. The table includes rows for container-related fields like 'kubernetes.container_image.keyword' and 'kubernetes.container_name', as well as various Kubernetes labels such as 'kubernetes.labels.k8s-app.keyword'.

name	type	format	searchable	aggregatable	excluded	controls
kubernetes.container_image.keyword	string		✓	✓		edit
kubernetes.container_image_id	string		✓			edit
kubernetes.container_image_id.keyword	string		✓	✓		edit
kubernetes.container_name	string		✓			edit
kubernetes.container_name.keyword	string		✓	✓		edit
kubernetes.host	string		✓			edit
kubernetes.host.keyword	string		✓	✓		edit
kubernetes.labels.heritage.keyword	string		✓	✓		edit
kubernetes.labels.k8s-app	string		✓			edit
kubernetes.labels.k8s-app.keyword	string		✓	✓		edit
kubernetes.labels.kubernetes_io/cluster-service	string		✓			edit
kubernetes.labels.kubernetes_io/cluster-service.keyword	string		✓	✓		edit
kubernetes.labels.pod-template-generation	string		✓			edit
kubernetes.labels.pod-template-generation.keyword	string		✓	✓		edit
kubernetes.labels.pod-template-hash	string		✓			edit
kubernetes.labels.pod-template-hash.keyword	string		✓	✓		edit
kubernetes.labels.prometheus	string		✓			edit
kubernetes.labels.prometheus.keyword	string		✓	✓		edit

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:

The screenshot shows the Kibana interface with a sidebar on the left containing icons for Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The 'Dashboard' item is selected and highlighted in blue. The main content area is titled 'Dashboards' and contains a search bar with placeholder text 'Search...'. Below the search bar is a table listing 20 different dashboard entries. Each entry includes a checkbox, the dashboard title, a brief description, and an 'Edit' button. The titles and descriptions are as follows:

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

At the bottom of the table, there are buttons for 'Rows per page: 20' and a page navigation section with links for < 1 2 >.

Kubernetes cluster dashboard



Add a loadbalancer in Route53 Domain name

The screenshot shows the AWS Route53 'Edit Record Set' interface. In the main pane, there is a table listing various DNS records. One entry for 'kibana.kops-prod.etechbrain.com.' is selected and highlighted with a blue border. The right-hand panel displays the configuration for this specific record set, which is currently an alias pointing to 'dualstack.internal-ac854290b2e4711e'. The 'Save Record Set' button is visible at the bottom right of the panel.

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

The screenshot shows the Kibana dashboard for a Metricbeat Kubernetes overview. The left sidebar has 'Dashboard' selected. The main area displays two large numerical values: 'Nodes' (6) and 'Deployments' (10). Below these are three smaller cards: 'Kubernetes - Nodes', 'Kubernetes - Deployments', and 'Available pods per deployment [Metricbeat Kubernetes]'. This card features a stacked bar chart with green segments representing different pod components. Another chart below it shows 'Unavailable pods per deployment [Metricbeat Kubernetes]' with orange segments for 'calico-kube-controllers' and 'coredns'.