DZ105-Deploying and Monitoring Elastic Stack on GCP

Speaker: David Pilato

Google Cloud

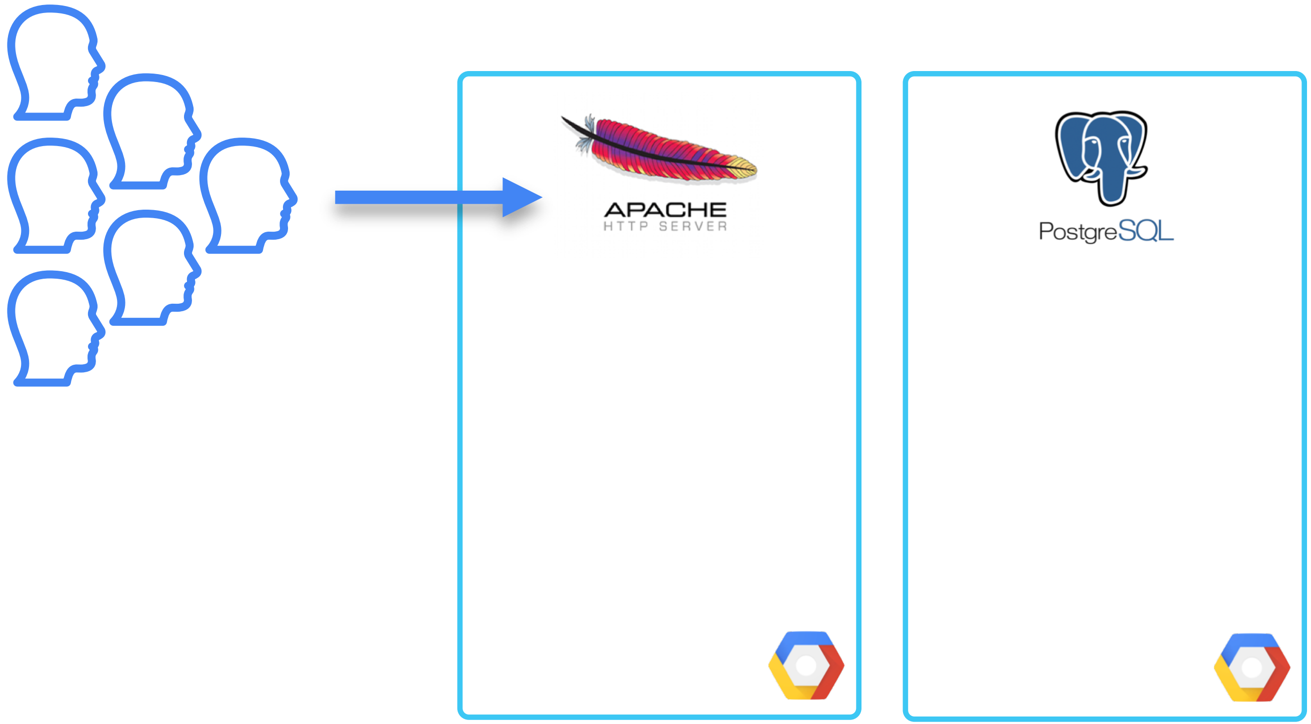
David Pilato

Developer | Evangelist
Elastic
@dadoonet
https://speaker.pilato.fr/

Deploying on Google Cloud Platform

01

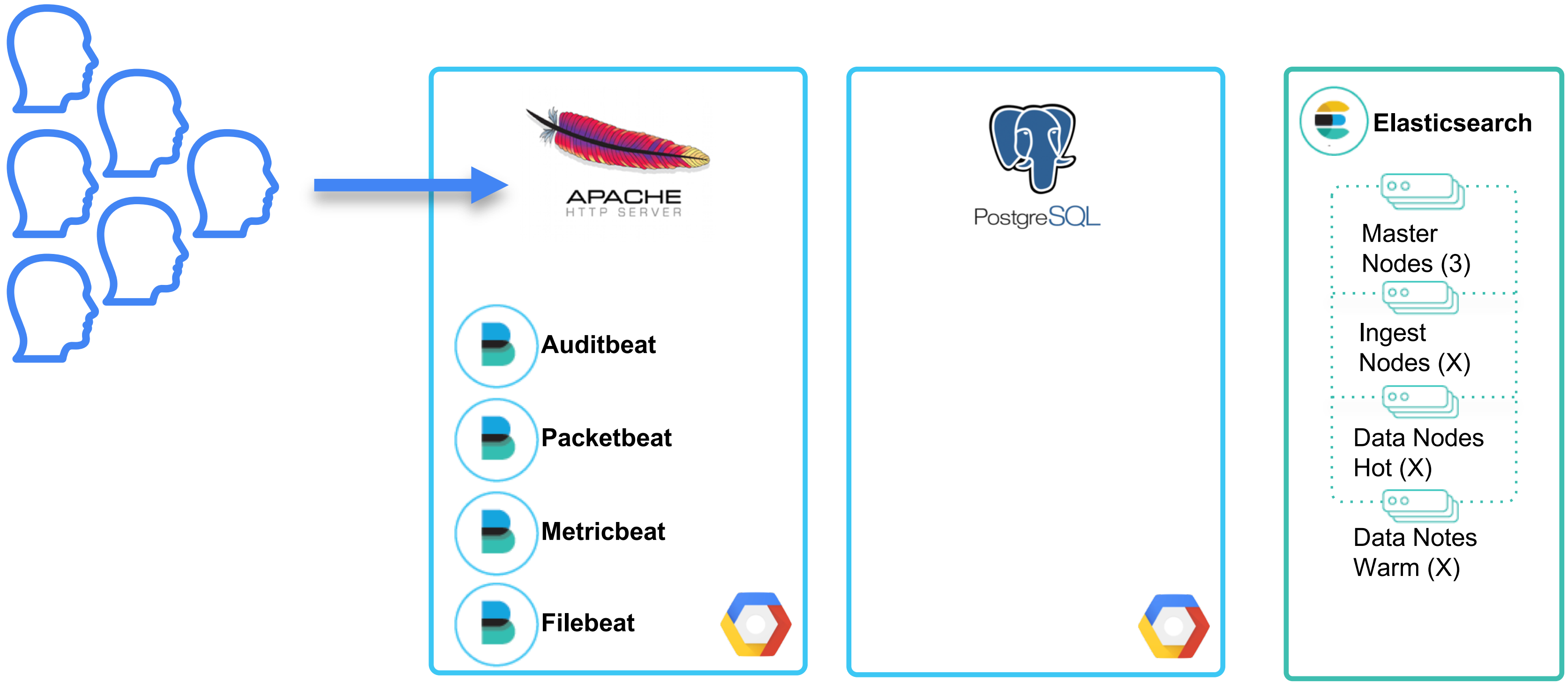
Deploying beats



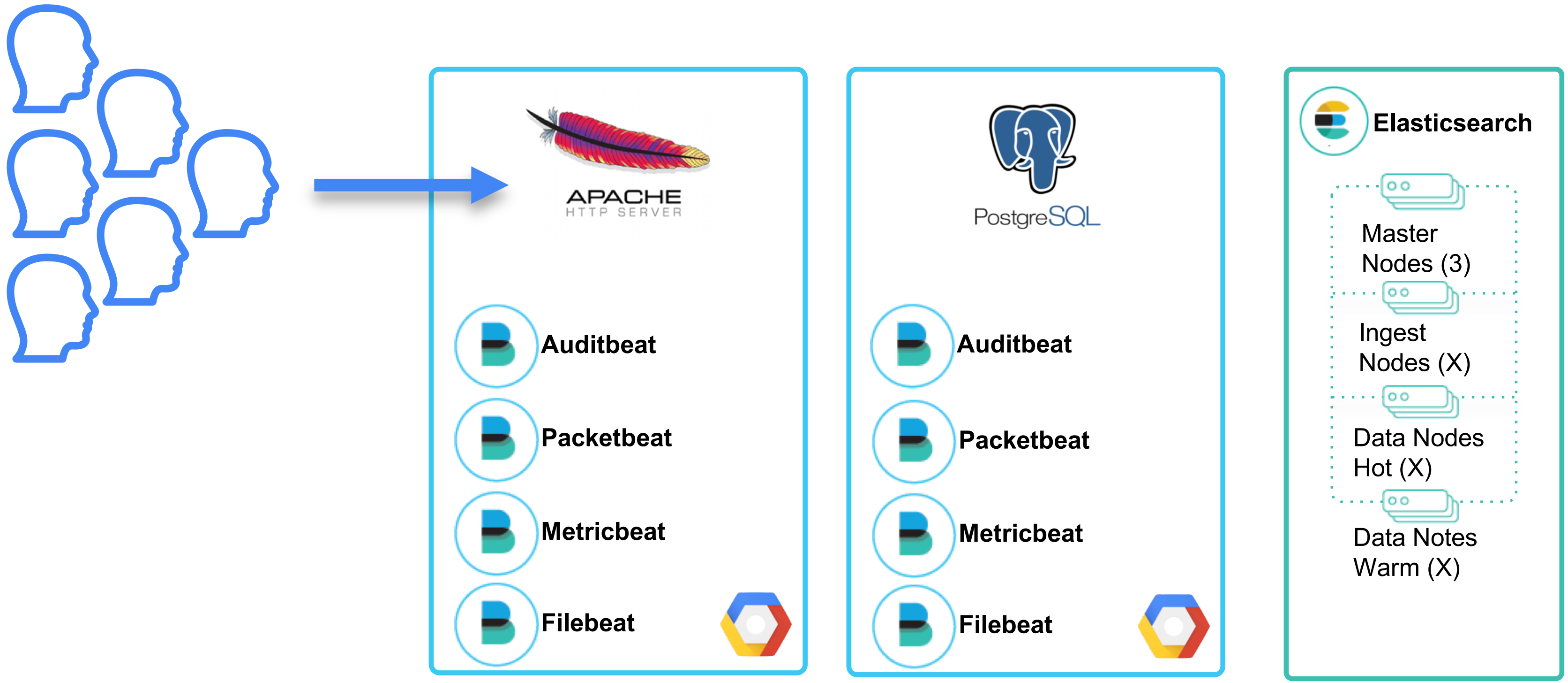
Google Cloud

Deploying beats

Deploying beats

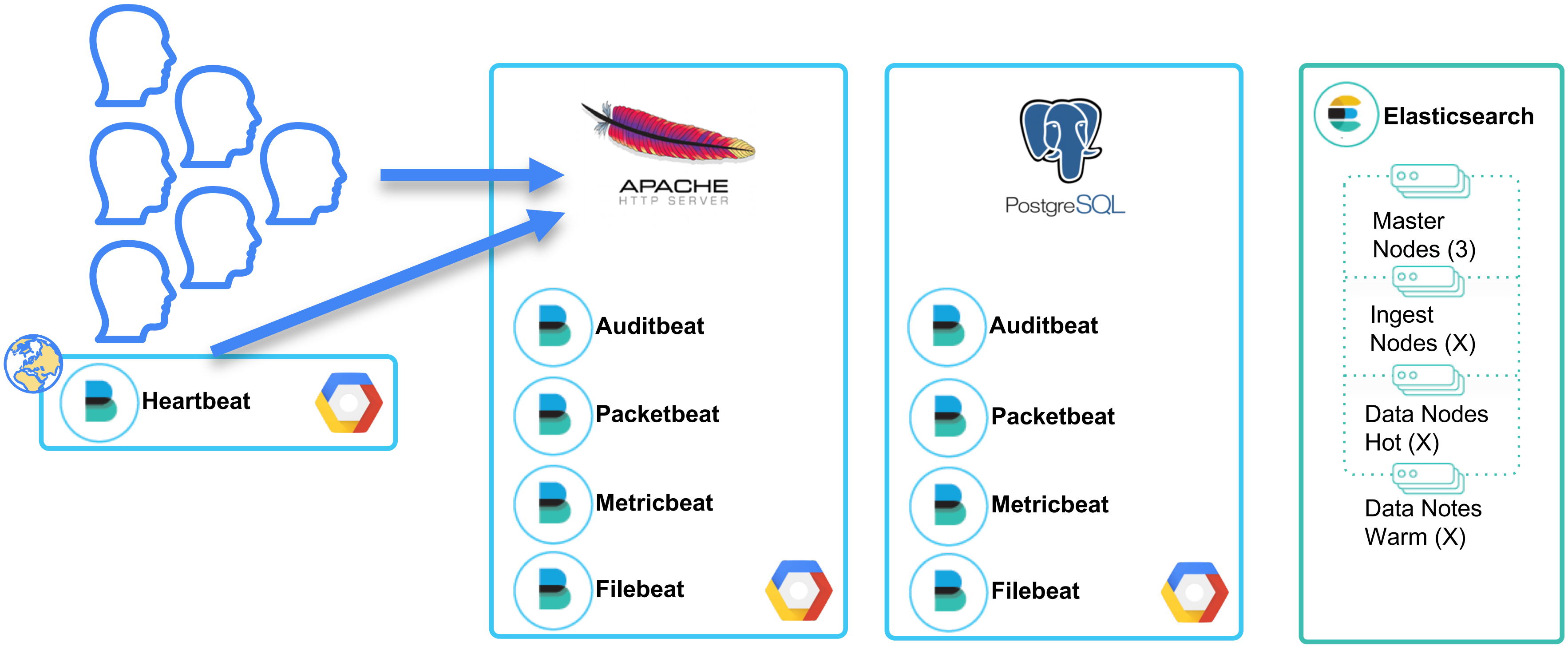


Deploying beats



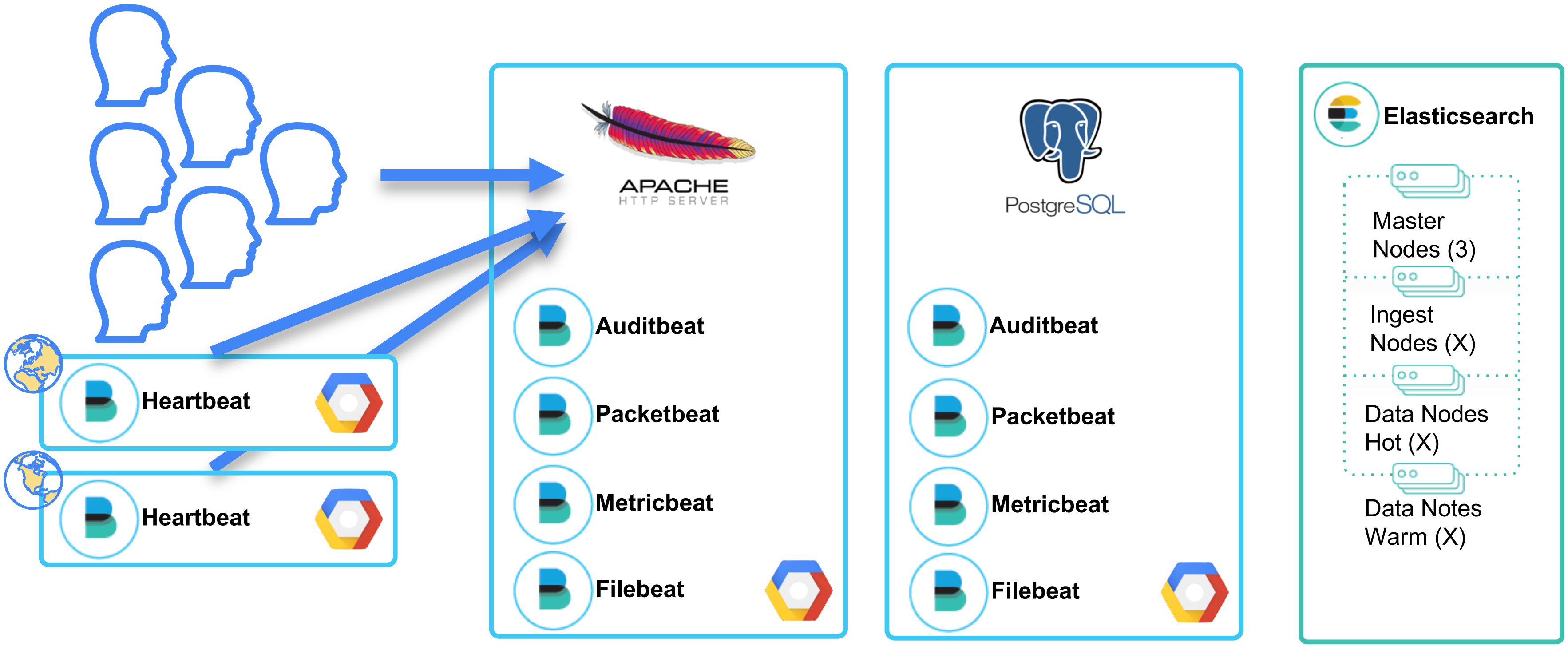
Google Cloud

Deploying beats



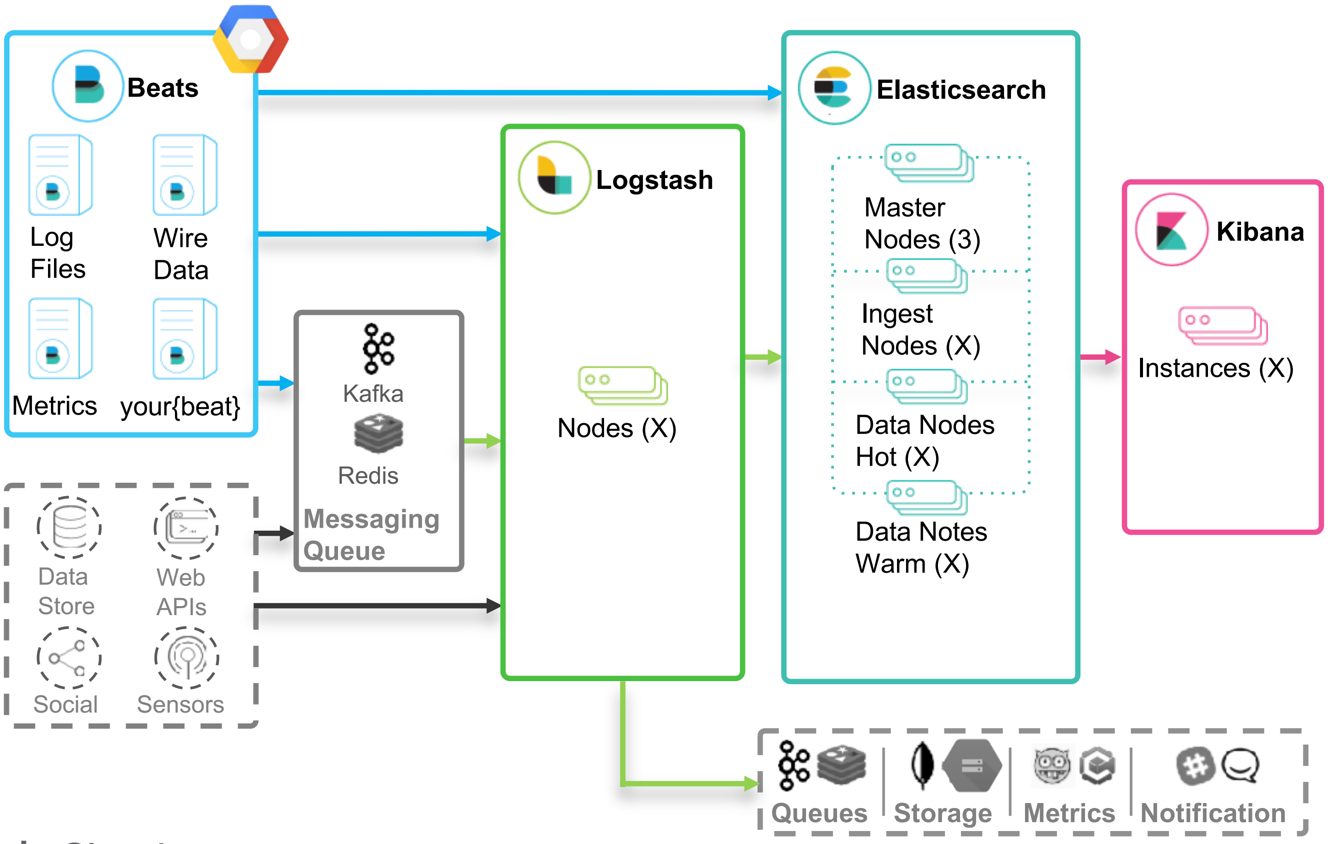
Google Cloud

Deploying beats



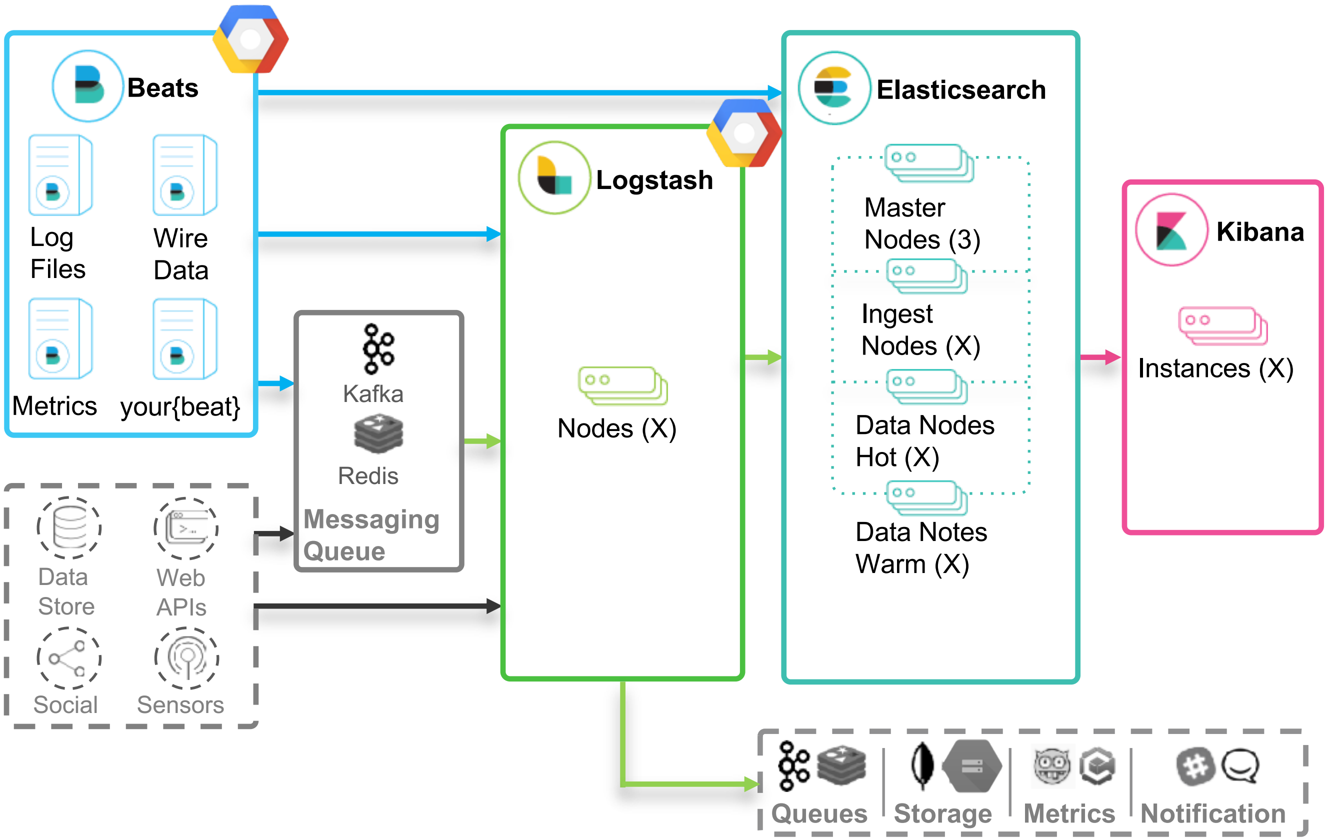
Google Cloud

Elastic Platform



Google Cloud

Elastic Platform



Google Cloud

Elasticsearch manual deployment

02

Manual deployment

Create the first machine

```
$ gcloud beta compute instances create \
gnext-dadoonet-1 \
--machine-type=n1-standard-1 \
--image=debian-9-stretch-v20191115 \
--image-project=debian-cloud \
--boot-disk-size=10GB \
--boot-disk-type=pd-standard \
--boot-disk-device-name=gnext-dadoonet-1
```

Manual deployment

Create the second machine.
For High Avaibility.

```
$ gcloud beta compute instances create \
gnext-dadoonet-2 \
--machine-type=n1-standard-1 \
--image=debian-9-stretch-v20191115 \
--image-project=debian-cloud \
--boot-disk-size=10GB \
--boot-disk-type=pd-standard \
--boot-disk-device-name=gnext-dadoonet-2
```

Manual deployment

Create the third machine.
To avoid split brain issues.

```
$ gcloud beta compute instances create \
gnext-dadoonet-3 \
--machine-type=n1-standard-1 \
--image=debian-9-stretch-v20191115 \
--image-project=debian-cloud \
--boot-disk-size=10GB \
--boot-disk-type=pd-standard \
--boot-disk-device-name=gnext-dadoonet-3
```

Manual deployment

Check that all machines are running

```
$ gcloud beta compute instances list
NAME               INTERNAL_IP   STATUS
gnext-dadoonet-1 10.154.0.2    RUNNING
gnext-dadoonet-2 10.154.0.3    RUNNING
gnext-dadoonet-3 10.154.0.4    RUNNING
```

Manual deployment

Install Elasticsearch

```
# SSH to the instance
$ gcloud beta compute ssh gnext-dadoonet-1

# Install Elasticsearch
$ wget -qO - \
https://artifacts.elastic.co/GPG-KEY-elasticsearch | \
sudo apt-key add -
$ sudo apt-get install apt-transport-https
$ echo \
"deb https://artifacts.elastic.co/packages/7.x/apt stable main"|\
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
$ sudo apt-get update && sudo apt-get install elasticsearch

# Automatic restart
$ sudo /bin/systemctl daemon-reload
$ sudo /bin/systemctl enable elasticsearch.service
```

Manual deployment

On the first node

```
# Generate a certificate and a private key
$ sudo /usr/share/elasticsearch/bin/elasticsearch-certutil ca
$ sudo /usr/share/elasticsearch/bin/elasticsearch-certutil cert \
--ca elastic-stack-ca.p12

# Copy certificates /usr/share/elasticsearch/*.p12 on each node
# In /etc/default/elasticsearch/certs

$ sudo mkdir /etc/elasticsearch/certs
$ sudo cp /usr/share/elasticsearch/*.p12 /etc/elasticsearch/certs
$ sudo chmod g+rw /etc/elasticsearch/certs/elastic-
certificates.p12
```

Manual deployment

Setup elasticsearch

```
# Configure Elasticsearch
$ sudo vi /etc/elasticsearch/elasticsearch.yml
network.host: _site_
discovery.seed_hosts:
  - 10.154.0.2
  - 10.154.0.3
  - 10.154.0.4
xpack.security.enabled: true
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: certs/elastic-
certificates.p12
xpack.security.transport.ssl.truststore.path: certs/elastic-
certificates.p12
# Optionally
# xpack.security.http.ssl.enabled: true
# xpack.security.http.ssl.keystore.path: certs/elastic-
certificates.p12
# xpack.security.http.ssl.truststore.path: certs/elastic-
certificates.p12
```

Manual deployment

Start elasticsearch

```
# Start Elasticsearch
$ sudo systemctl start elasticsearch.service

# Check logs
$ sudo tail -f /var/log/elasticsearch/elasticsearch.log

[INFO ][o.e.n.Node                 ] [gnext-dadoonet-1] started
```

Repeat on the 2 other nodes

Manual deployment

Generate stack users

```
$ sudo /usr/share/elasticsearch/bin/elasticsearch-setup-passwords
interactive
```

Manual deployment

Check the cluster

```
# Check the cluster
$ curl "10.154.0.2:9200/_cat/nodes?v&h=ip,node.role,master,name"\
-u elastic
Enter host password for user 'elastic':
ip          node.role master name
10.154.0.3 dilm      -      gnext-dadoonet-2
10.154.0.4 dilm      -      gnext-dadoonet-3
10.154.0.2 dilm      *      gnext-dadoonet-1
```

Use discovery-gce plugin

03

With discovery-gce plugin

Install the plugin

```
# Install GCE Discovery plugin
$ sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install
discovery-gce

# Change elasticsearch settings
$ sudo vi /etc/elasticsearch/elasticsearch.yml
# Remove the following lines
# discovery.seed_hosts:
#   - 10.154.0.2
#   - 10.154.0.3
#   - 10.154.0.4
# And add
cloud.gce.project_id: elastic-community
cloud.gce.zone: europe-west2-c
discovery.seed_providers: gce
```

With discovery-gce plugin

Stop the VMs and add compute-rw rights

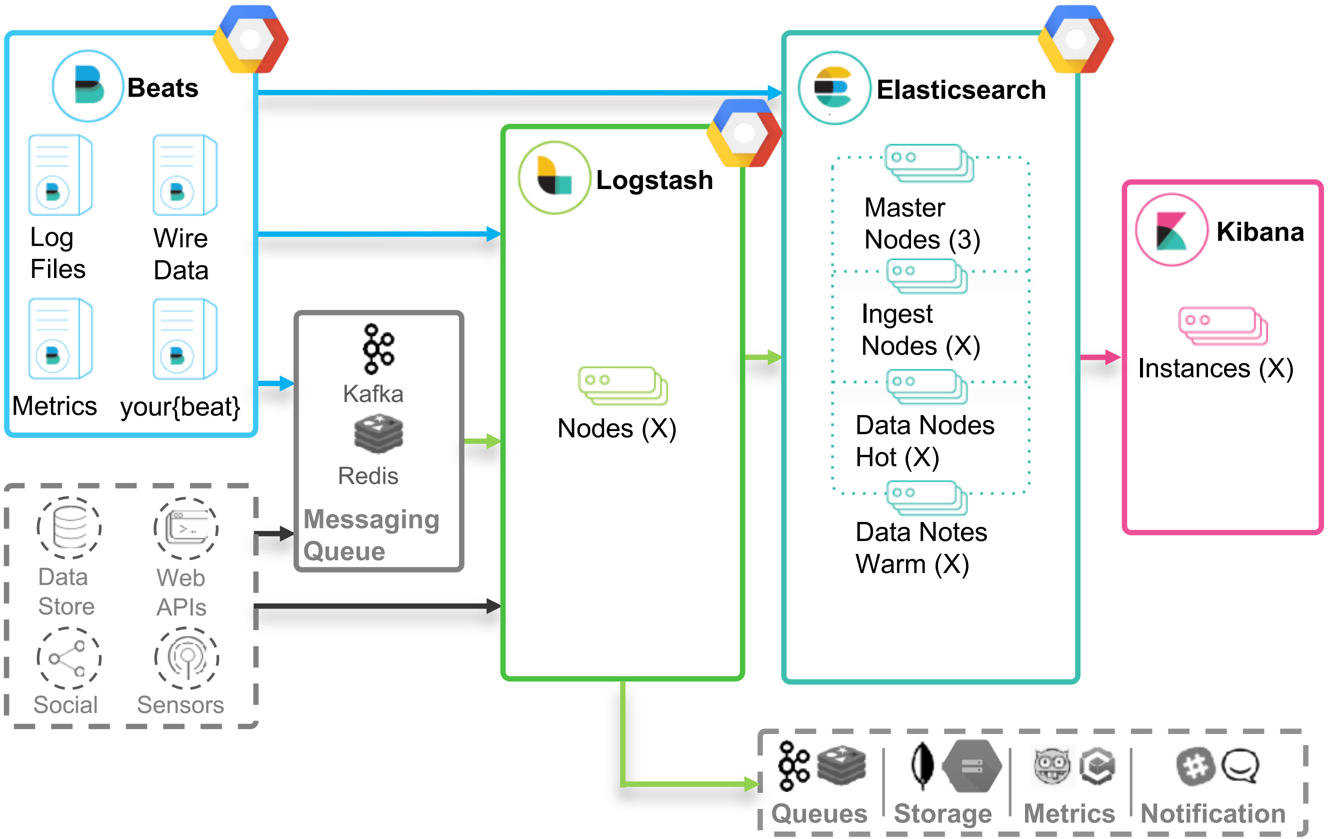
```
# Add compute-rw scope for each machine
$ gcloud beta compute instances stop gnext-dadoonet-1
$ gcloud beta compute instances set-scopes gnext-dadoonet-1 \
--scopes=default,compute-rw
$ gcloud beta compute instances start gnext-dadoonet-1

$ gcloud beta compute instances stop gnext-dadoonet-2
$ gcloud beta compute instances set-scopes gnext-dadoonet-2 \
--scopes=default,compute-rw
$ gcloud beta compute instances start gnext-dadoonet-2

$ gcloud beta compute instances stop gnext-dadoonet-3
$ gcloud beta compute instances set-scopes gnext-dadoonet-3 \
--scopes=default,compute-rw
$ gcloud beta compute instances start gnext-dadoonet-3
```

With discovery-gce plugin

Check the cluster

```
# Check the cluster
$ curl "10.154.0.2:9200/_cat/nodes?v&h=ip,node.role,master,name"\
-u elastic
Enter host password for user 'elastic':
ip         node.role master name
10.154.0.3 dilm      -      gnext-dadoonet-2
10.154.0.4 dilm      -      gnext-dadoonet-3
10.154.0.2 dilm      *      gnext-dadoonet-1
```

Elastic Platform



Google Cloud

Kibana manual deployment

04

Kibana deployment

Create Kibana VM

```
# Create firewall rules
$ gcloud beta compute firewall-rules create kibana \
--direction=INGRESS --priority=1000 \
--network=default --action=ALLOW --rules=tcp:5601
--source-ranges=0.0.0.0/0 --target-tags=kibana

# Create Kibana GCE Instance
$ gcloud beta compute instances create \
gnext-dadoonet-k \
--machine-type=n1-standard-1 \
--tags kibana \
--image=debian-9-stretch-v20191115 \
--image-project=debian-cloud \
--boot-disk-size=10GB \
--boot-disk-type=pd-standard \
--boot-disk-device-name=gnext-dadoonet-k
```

Manual deployment

Install Kibana

```
# SSH to the instance
$ gcloud beta compute ssh gnext-dadoonet-k

# Install Kibana
$ wget -qO - \
https://artifacts.elastic.co/GPG-KEY-elasticsearch | \
sudo apt-key add -
$ sudo apt-get install apt-transport-https
$ echo \
"deb https://artifacts.elastic.co/packages/7.x/apt stable main"|\
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
$ sudo apt-get update && sudo apt-get install kibana

# Automatic restart
$ sudo /bin/systemctl daemon-reload
$ sudo /bin/systemctl enable kibana.service
```

Manual deployment

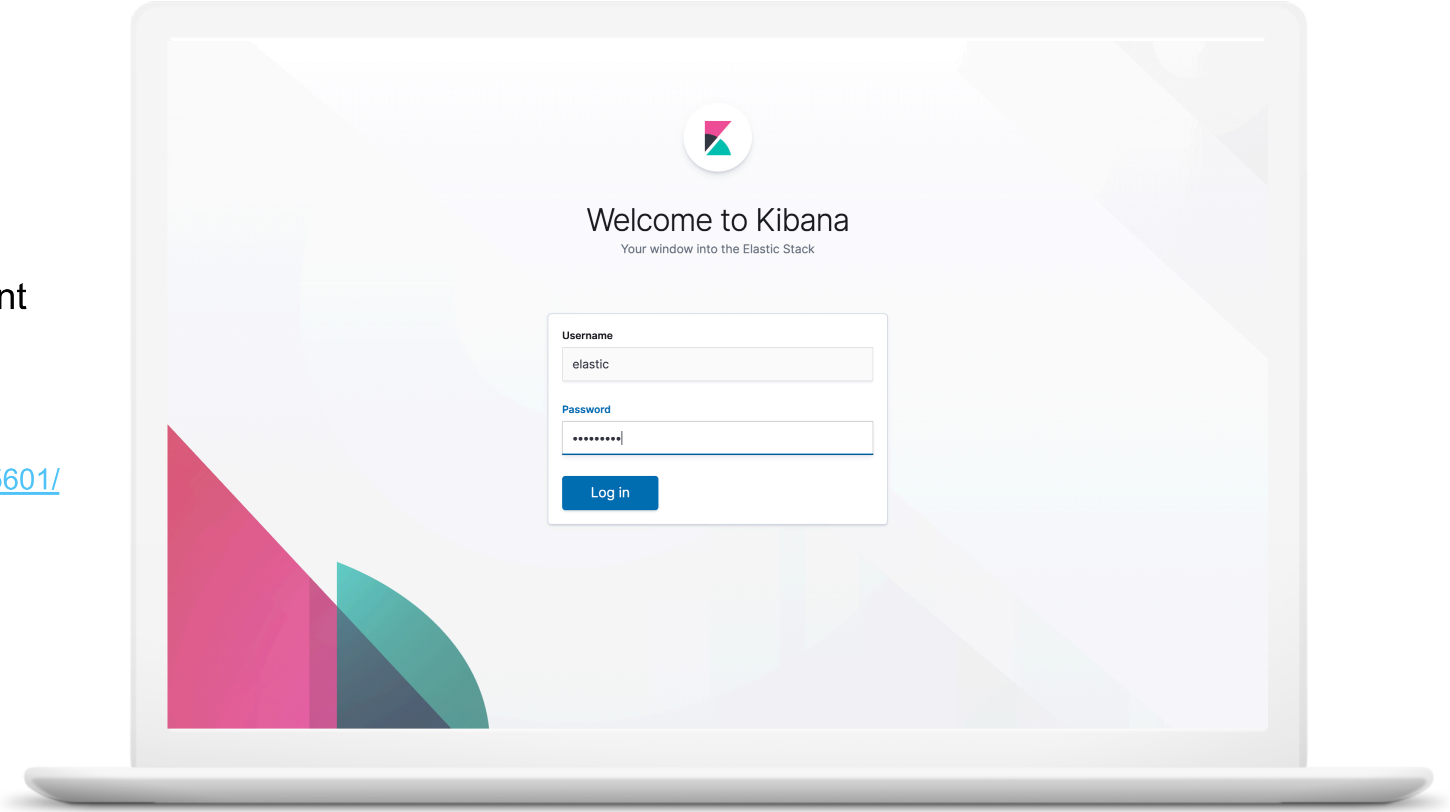
Setup and Start Kibana

```
# Configure Kibana
$ sudo vi /etc/kibana/kibana.yml
server.host: "10.154.0.5"
elasticsearch.hosts: ["http://10.154.0.2:9200",
"http://10.154.0.3:9200", "http://10.154.0.4:9200"]
xpack.security.encryptionKey:
"something_at_least_32_characters_gnext-dadoonet-k_gnext-
dadoonet-k"
elasticsearch.username: "kibana"
elasticsearch.password: "my_incredible_password"

# Start Kibana
sudo systemctl start kibana.service

# Check logs
sudo journalctl --unit kibana -f
```

Manual deployment

Open

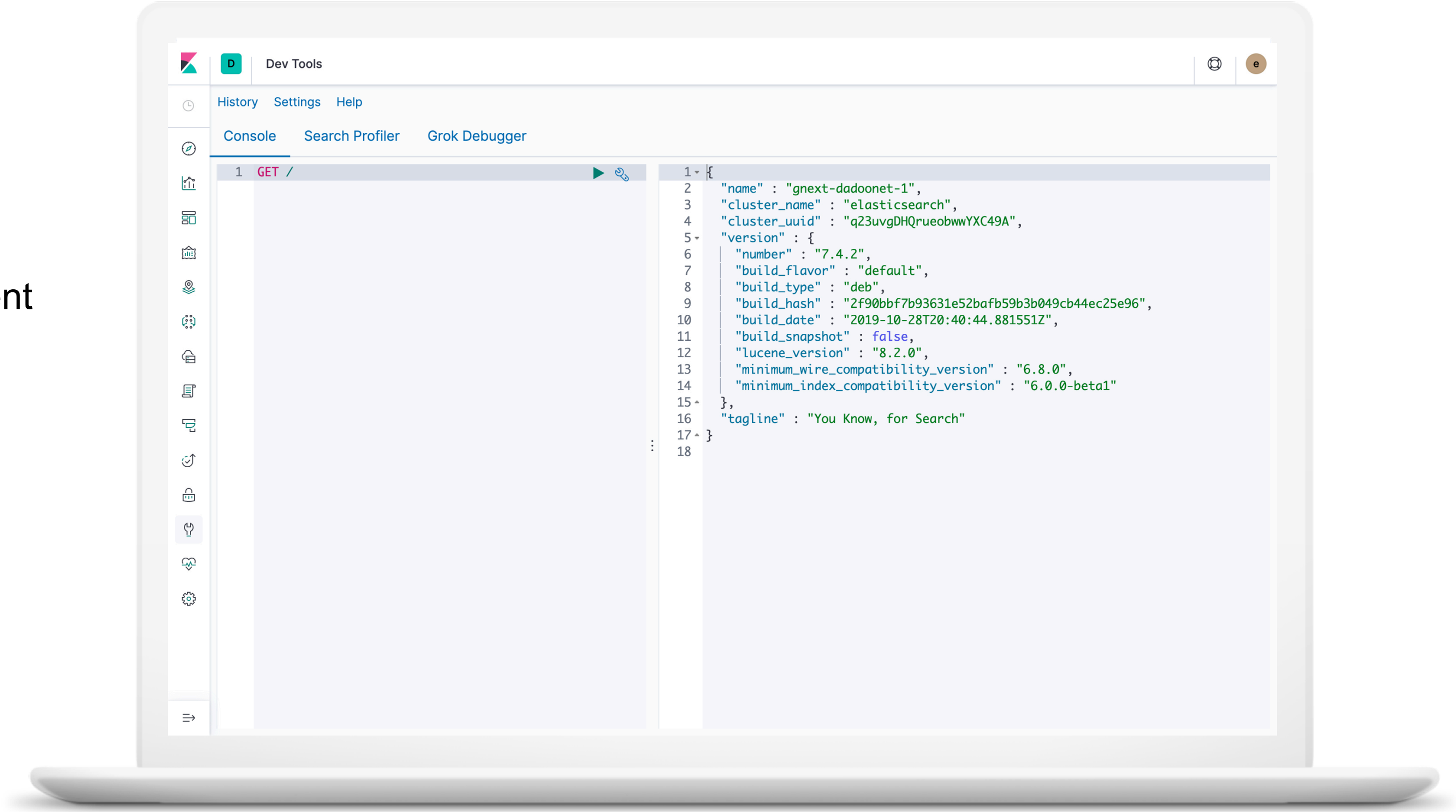
http://35.197.195.120:5601/



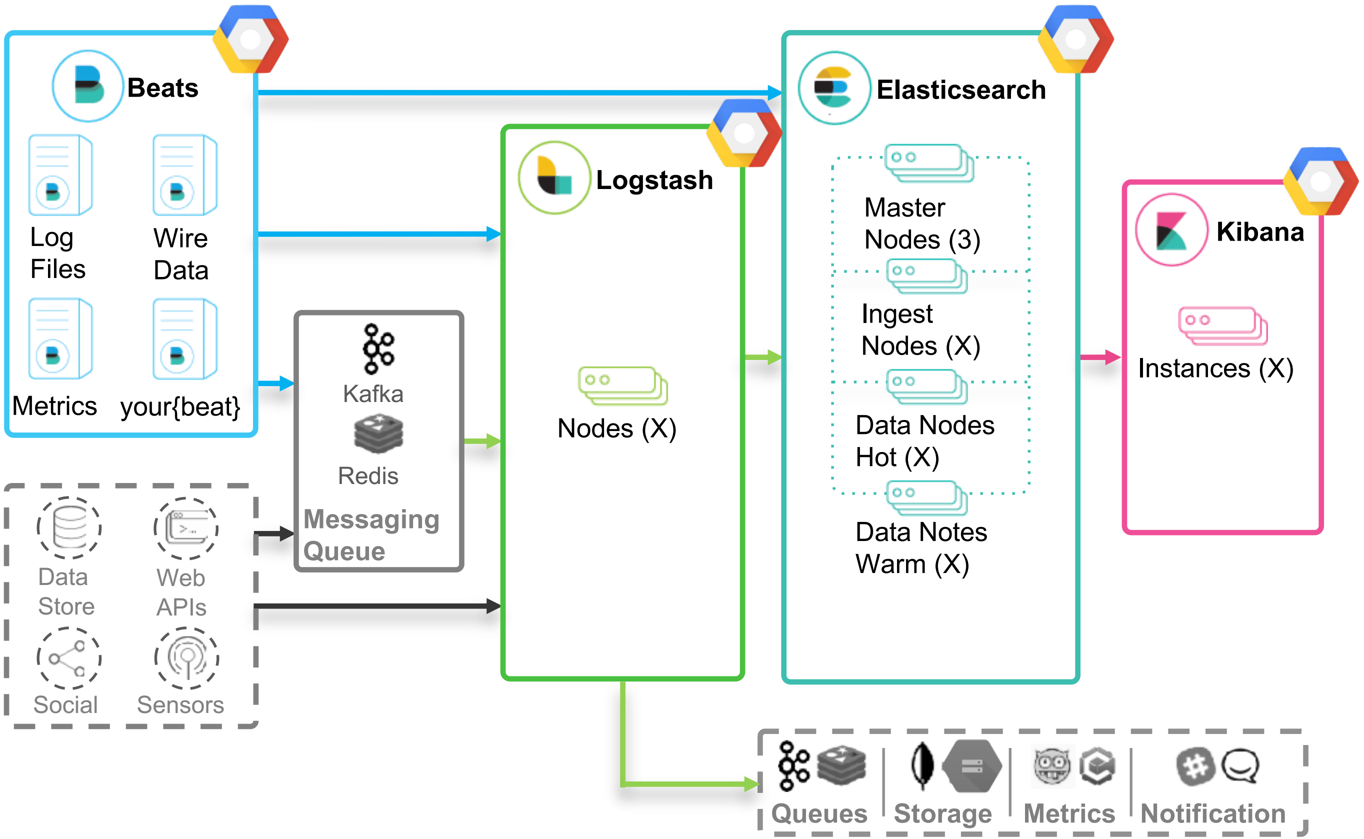
Manual deployment

Check with Dev Tools

GET /



Elastic Platform



Google Cloud

Using ECK

05

Kubernetes deployment

Create Kubernetes cluster

```
$ gcloud beta container clusters create "gnext-dadoonet" \
--zone "europe-west2-c" --machine-type "n1-standard-4"


$ gcloud beta container clusters get-credentials gnext-dadoonet \
--zone europe-west2-c
```

Kubernetes deployment

Deploy ECK in Kubernetes

```
# Install the operator
$ kubectl apply -f
https://download.elastic.co/downloads/eck/1.0.0-beta1/all-in-
one.yaml

# Monitor the operator logs
$ kubectl -n elastic-system logs -f statefulset.apps/elastic-
operator
```

Kubernetes deployment

Deploy an Elasticsearch cluster

```
# Deploy 3 elasticsearch nodes
$ cat <<EOF | kubectl apply -f -
apiVersion: elasticsearch.k8s.elastic.co/v1beta1
kind: Elasticsearch
metadata:
  name: elasticsearch-gnext
spec:
  version: 7.4.2
  nodeSets:
  - name: default
    count: 3
    config:
      node.store.allow_mmap: false
EOF

# Check if it's running and ready
$ kubectl get elasticsearch
NAME                  HEALTH    NODES     VERSION    PHASE    AGE
elasticsearch-gnext   green     3         7.4.2      Ready    1m
# Check Elasticsearch logs from 1st node
$ kubectl logs -f elasticsearch-gnext-es-default-0
```

Kubernetes deployment

Connect to an Elasticsearch cluster

```
# Get the credentials
$ PASSWORD=$(kubectl get secret elasticsearch-gnext-es-elastic-user -o=jsonpath='{.data.elastic}' | base64 --decode)

# Connect to elasticsearch from your workstation
$ kubectl port-forward service/elasticsearch-gnext-es-http 9200
$ curl -u "elastic:$PASSWORD" -k "https://elasticsearch-gnext-es-http:9200"

{
  "name" : "elasticsearch-gnext-es-default-0",
  "cluster_name" : "elasticsearch-gnext",
  "cluster_uuid" : "e4nry98BTjCsLSZASIcrRg",
  "version" : { ... },
  "tagline" : "You Know, for Search"
}
```

Kubernetes deployment

Deploy a Kibana instance

```
# Deploy Kibana
$ cat <<EOF | kubectl apply -f -
apiVersion: kibana.k8s.elastic.co/v1beta1
kind: Kibana
metadata:
  name: kibana-gnext
spec:
  version: 7.4.2
  count: 1
  elasticsearchRef:
    name: elasticsearch-gnext
EOF

# Check until it's running
$ kubectl get kibana
```

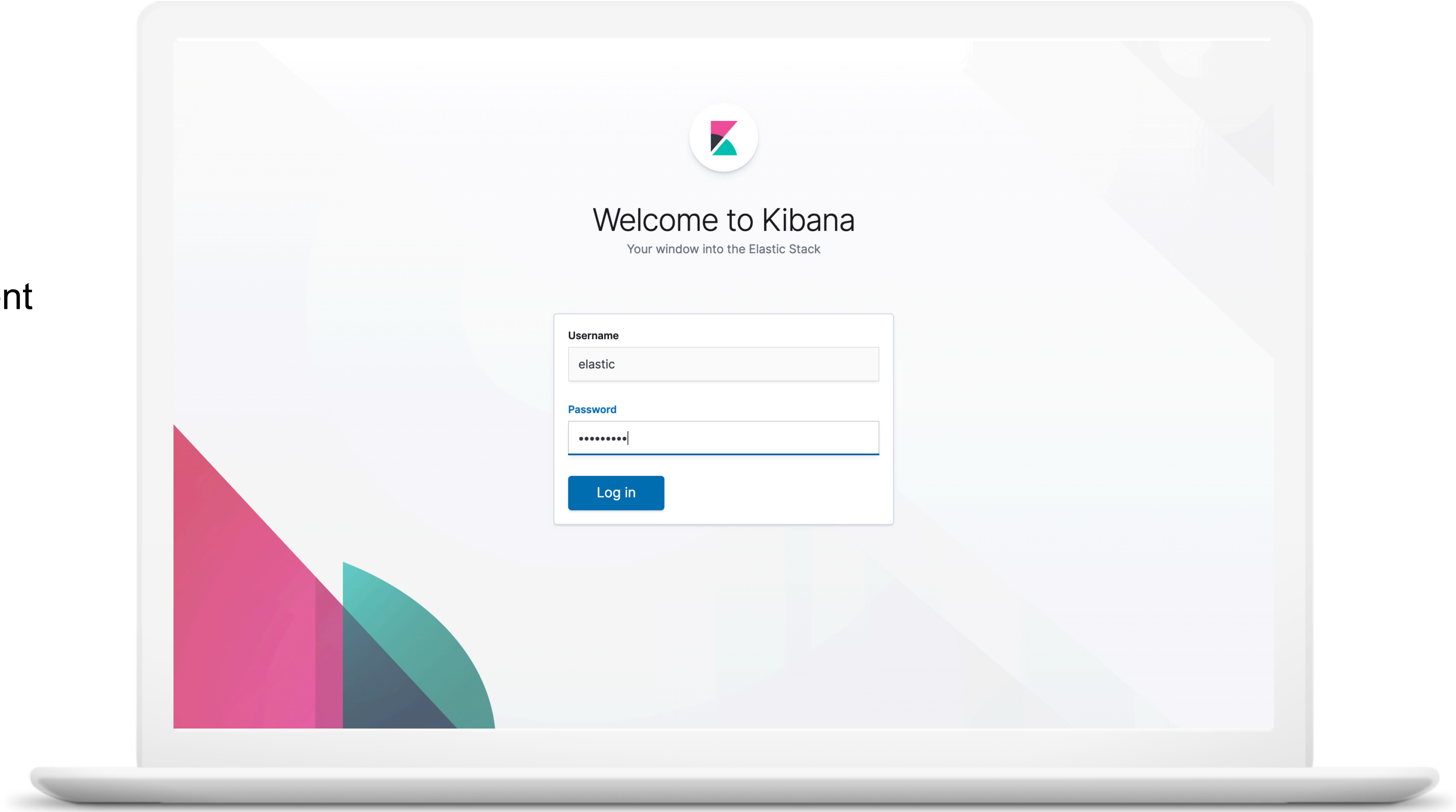
Kubernetes deployment

Connect to Kibana

```
# Get the elastic password
$ kubectl get secret elasticsearch-gnext-es-elastic-user -
o=jsonpath='{.data.elastic}' | base64 --decode; echo

# Connect to Kibana from your workstation
$ kubectl port-forward service/kibana-gnext-kb-http 5601
```

Manual deployment

Open
https://localhost:5601

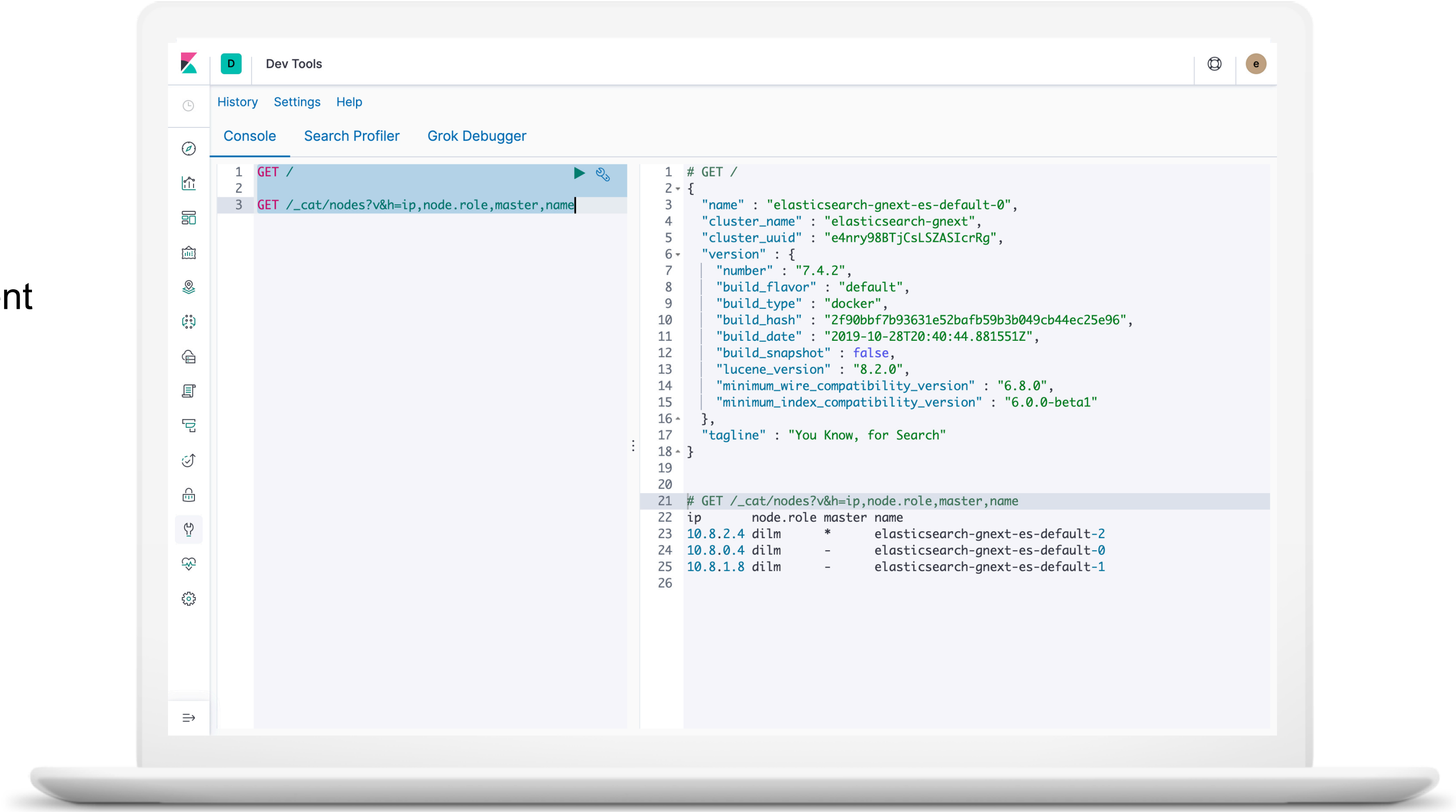


Manual deployment

Check with Dev Tools

GET /

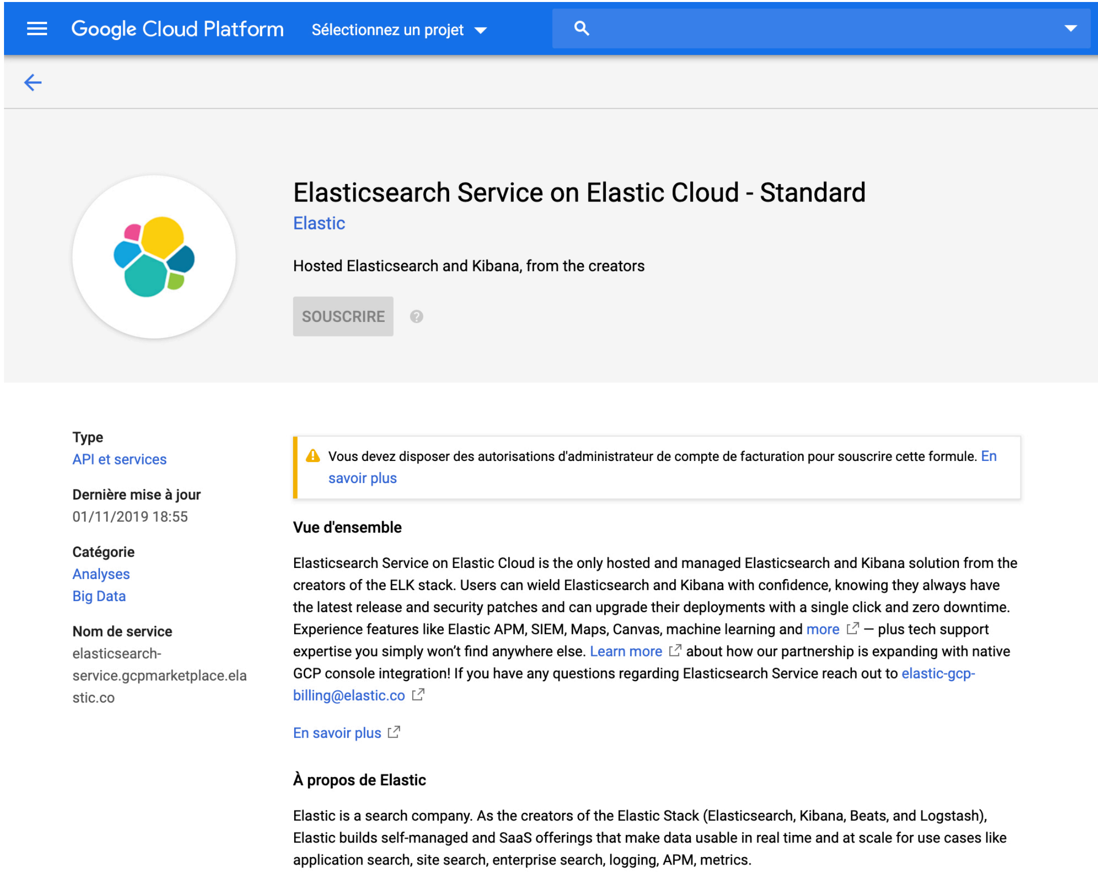
Check nodes



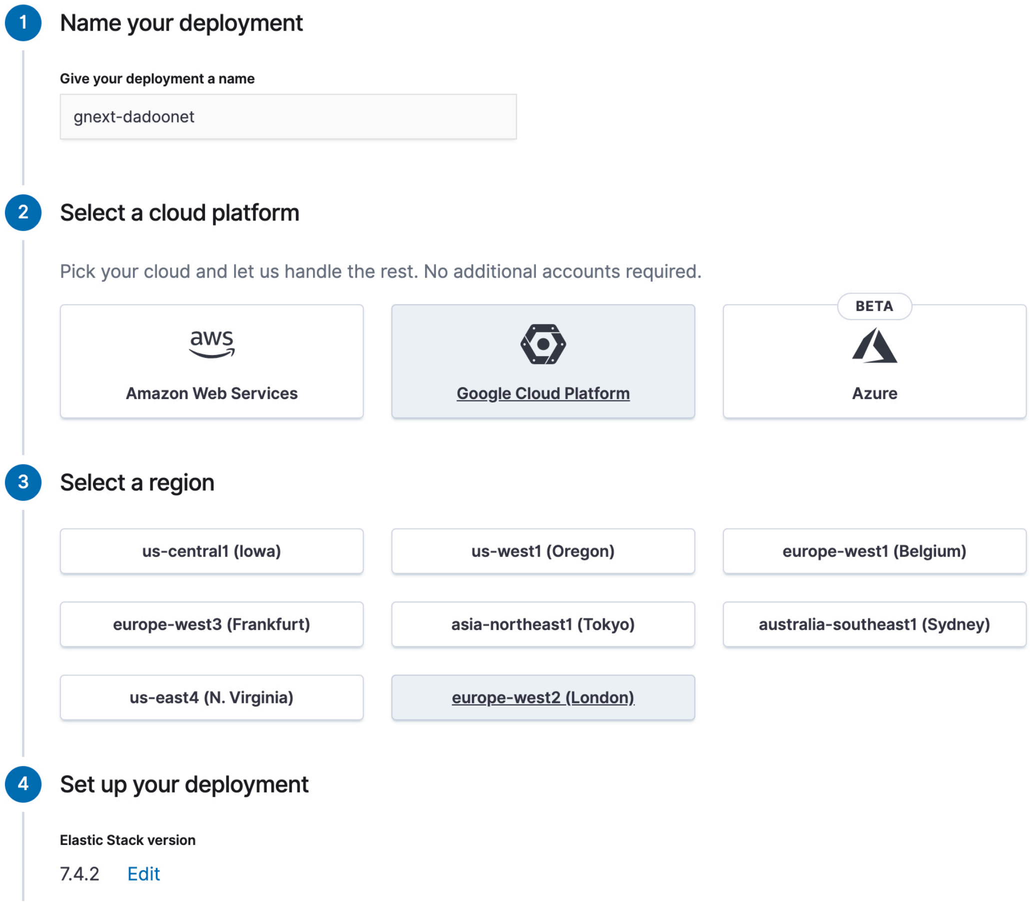
One click install

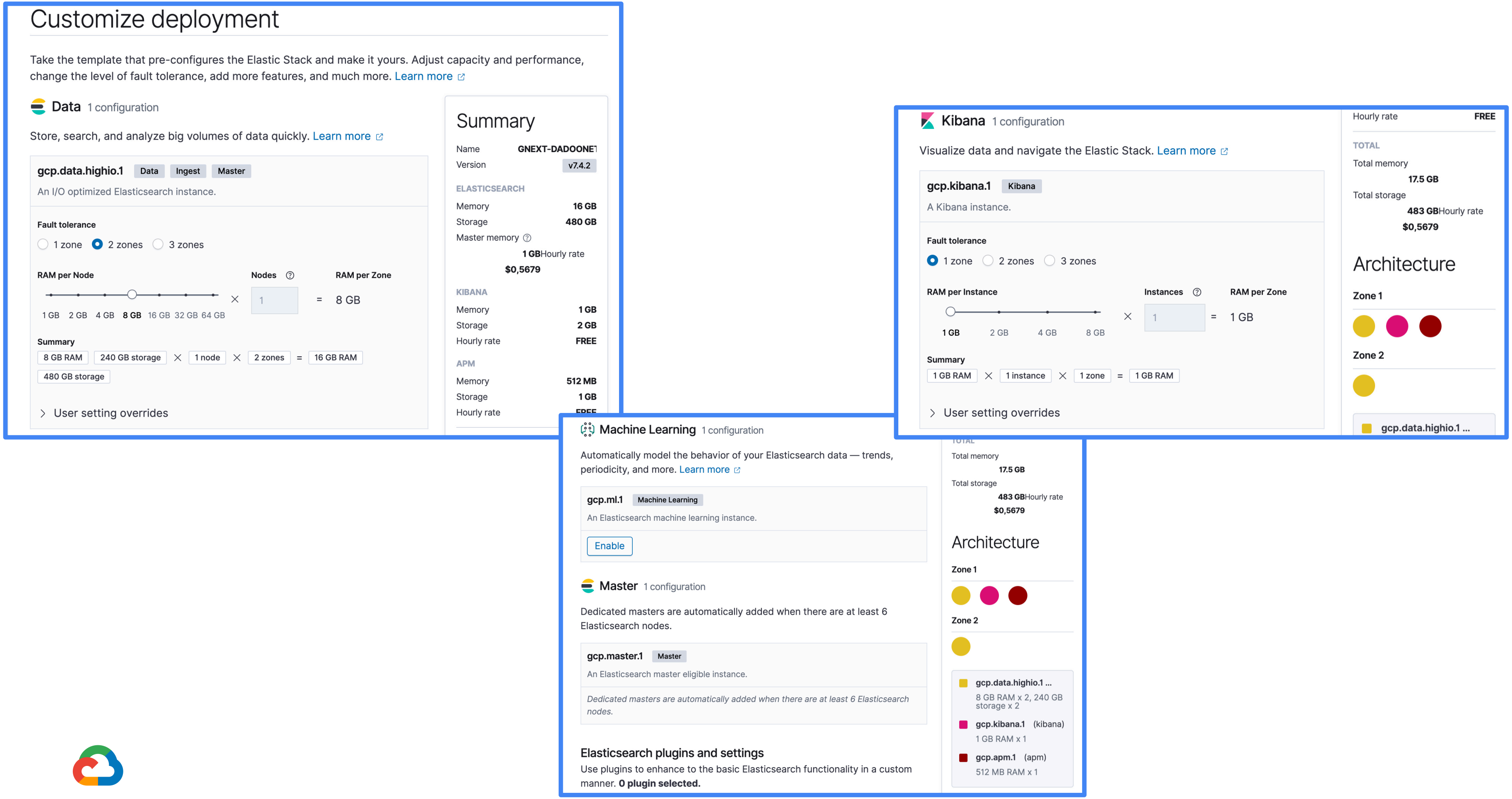
06

From Google Cloud Marketplace



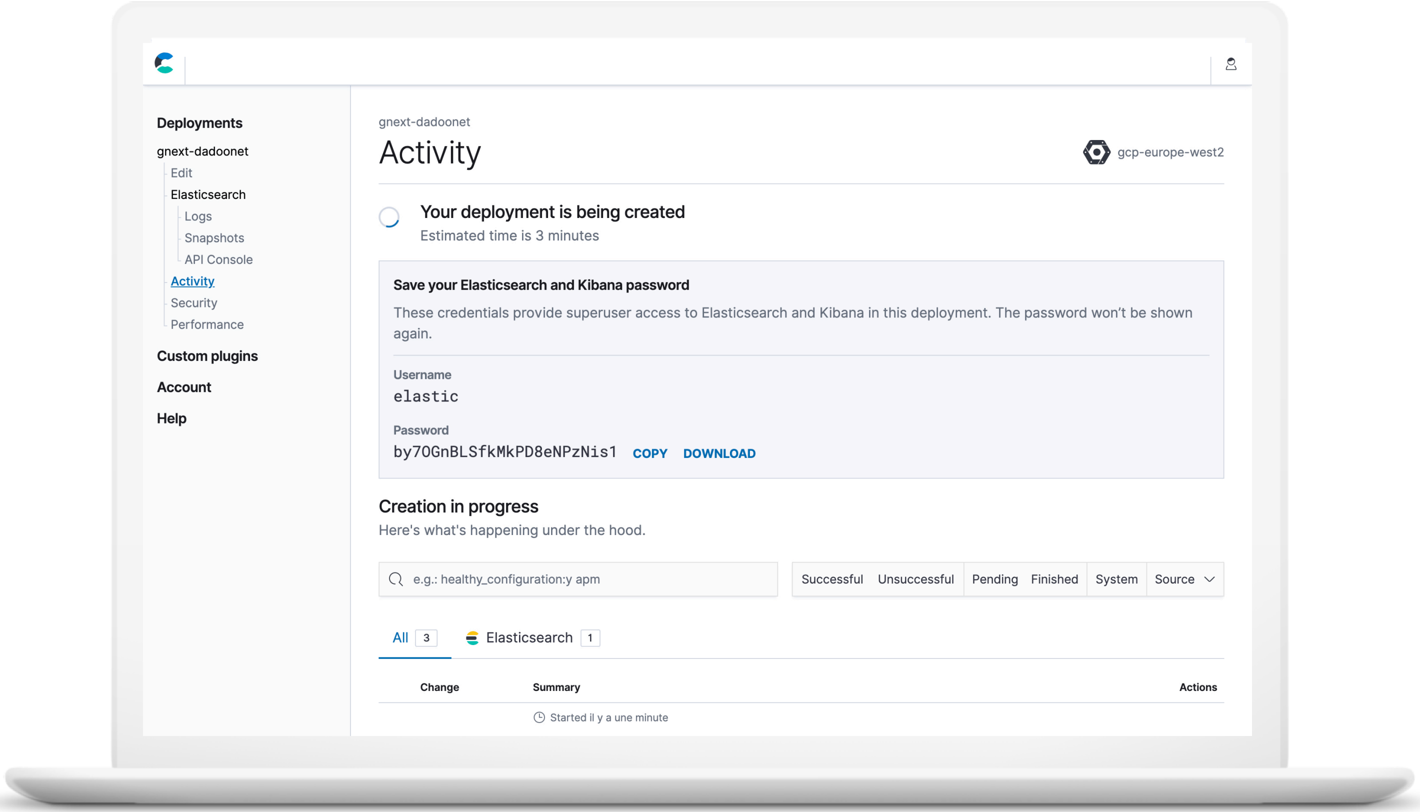
Or https://cloud.elastic.co/



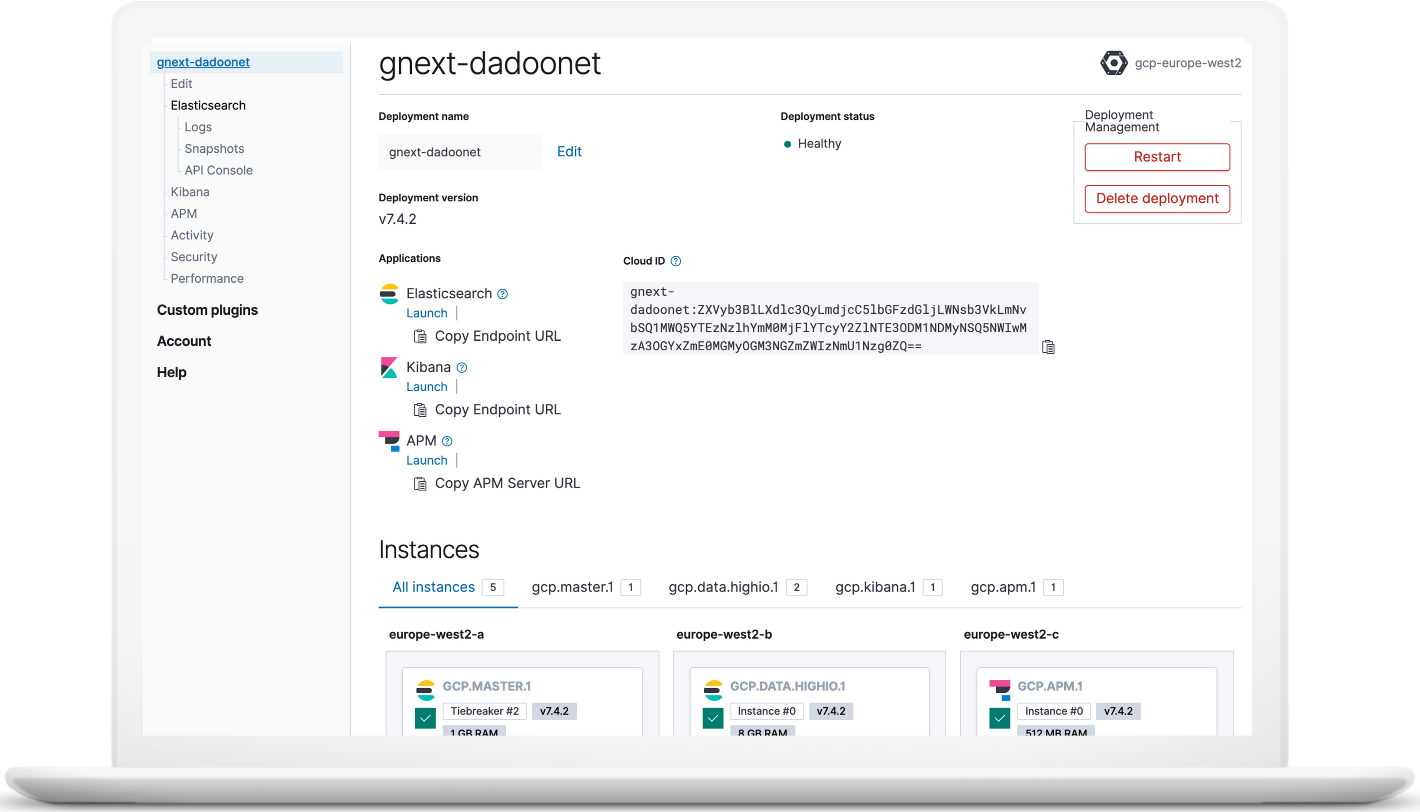


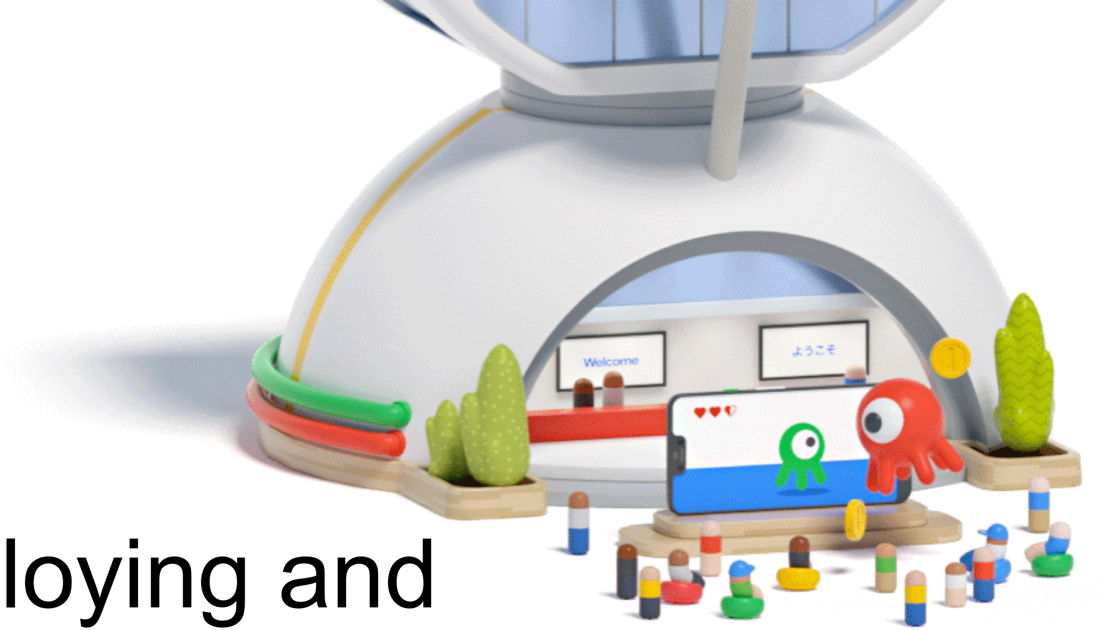
Customize deployment
Take the template that pre-configures the Elastic Stack and make it yours. Adjust capacity and performance, change the level of fault tolerance, add more features, and much more. Learn more
Data 1 configuration
Store, search, and analyze big volumes of data quickly. Learn more
gcp.data.highio.1 Data Ingest Master
An I/O optimized Elasticsearch instance.
Fault tolerance
1 zone 2 zones 3 zones
RAM per Node
Nodes
RAM per Zone
1 GB 2 GB 4 GB 8 GB 16 GB 32 GB 64 GB
1
= 8 GB
Summary
8 GB RAM 240 GB storage × 1 node × 2 zones = 16 GB RAM
480 GB storage
User setting overrides
Summary
Name GNEXT-DADOONE
Version v7.4.2
ELASTICSEARCH
Memory 16 GB
Storage 480 GB
Master memory
1 GB Hourly rate
$0,5679
KIBANA
Memory 1 GB
Storage 2 GB
Hourly rate FREE
APM
Memory 512 MB
Storage 1 GB
Hourly rate
Machine Learning 1 configuration
Automatically model the behavior of your Elasticsearch data — trends, periodicity, and more. Learn more
gcp.ml.1 Machine Learning
An Elasticsearch machine learning instance.
Enable
Master 1 configuration
Dedicated masters are automatically added when there are at least 6 Elasticsearch nodes.
gcp.master.1 Master
An Elasticsearch master eligible instance.
Dedicated masters are automatically added when there are at least 6 Elasticsearch nodes.
Elasticsearch plugins and settings
Use plugins to enhance to the basic Elasticsearch functionality in a custom manner. 0 plugin selected.
Total memory
17.5 GB
Total storage
483 GB Hourly rate
$0,5679
Architecture
Zone 1
Zone 2
gcp.data.highio.1 ...
8 GB RAM x 2, 240 GB storage x 2
gcp.kibana.1 (kibana)
1 GB RAM x 1
gcp.apm.1 (apm)
512 MB RAM x 1
Kibana 1 configuration
Visualize data and navigate the Elastic Stack. Learn more
gcp.kibana.1 Kibana
A Kibana instance.
Fault tolerance
1 zone 2 zones 3 zones
RAM per Instance
Instances
RAM per Zone
1 GB 2 GB 4 GB 8 GB
1
= 1 GB
Summary
1 GB RAM × 1 instance × 1 zone = 1 GB RAM
User setting overrides
Hourly rate FREE
TOTAL
Total memory
17.5 GB
Total storage
483 GB Hourly rate
$0,5679
Architecture
Zone 1
Zone 2
gcp.data.highio.1 ...

One click deploy



One click deploy





DZ105-Deploying and Monitoring Elastic Stack on GCP

Speaker: David Pilato