Elastic Suite Configuration Details

Version 1.18-SNAPSHOT

2018-09-17

Table of Contents

Table 1. History

Date	Author	Detail
2018-08-29	bcouetil	Asciidoc HTML look & feel changes
2018-08-24	bcouetil	Icones added for download + favicon added for webpage
2018-08-23	bcouetil	Initial commit

1. Elastic servers configuration

1.1. Initialize VM

- Adding a user

```
$ adduser devops
```

- Granting him root privileges

```
$ visudo
```

```
devops ALL=(ALL:ALL) ALL
```

- Checking FS size

```
$ parted
$ print free
```

- Example

```
Number  Start   End     Size    Type      File system  Flags
        32.3kB  1049kB  1016kB             Free Space
1       1049kB  500MB   499MB   primary   ext2         boot
2       500MB   53.7GB  53.2GB  primary                lvm
        53.7GB  53.7GB  1049kB             Free Space
```

⚠️ Below instructions are for Ubuntu only. You can check your Linux distribution with this command : cat /etc/*-release

- Add some server for apt-get

```
$ sudo vi /etc/apt/sources.list
```

```
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-security main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-updates main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-proposed main restricted universe multiverse
deb [arch=amd64] http://archive.ubuntu.com/ubuntu/ trusty-backports main restricted universe multiverse
```

1.2. Install Docker

> ⚠ Below instructions are for Ubuntu 14 only. You can check your Linux distribution with this command : cat /etc/*-release

```
$ apt-get install apt-transport-https ca-certificates curl software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable"
$ apt-get update
$ apt-cache search docker-ce
$ apt-get install docker-ce
```

- May work on Jenkins slave

> ⚠ don't use on managed PL, we don't have enough rights

```
$ sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo "$ID") \
    $(lsb_release -cs) \
    stable"
$ sudo apt-get update
$ sudo apt-cache search docker-ce
$ sudo apt-get install --assume-yes docker-ce
$ sudo dockerd
```

- Allow Docker remote API

Solution found here https://forums.docker.com/t/enable-remote-api-on-docker-hosts-in-ubuntu-14/11583/2

```
$ vi /etc/default/docker
```

```
$ DOCKER_OPTS="-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock"
```

- Start Docker Daemon

```
$ sudo dockerd
```

- To restart (as root)

> ⚠ Don't forget the docker.sock chmod if you use metricbeat

```
$ service docker restart
```

- To check FS size

```
root@frpardge:/var/lib/docker
$ du -sh -- * .*
92K     aufs
44K     containers
116K    image
52K     network
20K     plugins
4.0K    swarm
4.0K    tmp
4.0K    trust
28K     volumes
4.0K    .
61M     ..
```

- Get rid of sudo for devops user

```
$ sudo groupadd docker
$ sudo gpasswd -a devops docker
$ newgrp docker
$ docker run hello-world
```

- Install **Portainer** to ease administration

```
$ sudo docker pull portainer/portainer

$ sudo docker run -d --name portainer --restart=always -p 19000:9000 -v /var/run/docker.sock:/var/run/docker.sock
portainer/portainer
```

- To use, go to http://frpardge.corp.nvx.com:19000
  - login/password = admin / **
    - Install docker-compose

```
curl -L https://github.com/docker/compose/releases/download/1.19.0/docker-compose-`uname -s`-`uname -m` -o
/usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

docker-compose --version
```

1.2.1. Define Nexus3 as the Docker registry

- Raise a ticket in INSERE to ask a port opening for Nexus3 as a Docker registry
  - They will provide this kind of response, which indicates how to login before 'docker push' :

```
$ docker login docker-registry-bpmfactory.s2-eu.nvx.com
 User name: docker
 User Password: dockerPWdbpmfactory
```

- Use the information to add the registry in docker configuration

```
$ vi /etc/docker/daemon.json
```

```
{
  "storage-driver": "devicemapper",
  "insecure-registries": [
    "docker-registry-bpmfactory.s2-eu.nvx.com"
  ],
  "disable-legacy-registry": true
}
```

- be carefull not to have INSECURE_REGISTRY here, it would not start :

```
$ vim /etc/sysconfig/docker
```

```
#INSECURE_REGISTRY='--insecure-registry userbxxy05.socle:8444'
```

- Redémarrer docker

```
$ service docker restart
```

1.3. Setup a dockerized Oracle12c database

Database found here : https://hub.docker.com/r/sath89/oracle-12c/

```
$ docker pull sath89/oracle-12c

$ docker run --restart=always --name dbdev -d -p 18080:8080 -p 1521:1521 sath89/oracle-12c

$ docker logs -f feef20144fdc124d7b19d22aaf7bd63cbb837df667cc9764e7bdb5bcafa1af46

 Database not initialized. Initializing database.
 Starting tnslsnr
 Copying database files
 1% complete
 3% complete
 Import finished
 Database ready to use. Enjoy! ;)
```

Connect to Oracle Application Express web management console with following settings :

- host = http://frpardge:18080/apex

- workspace = **INTERNAL**

- user = **ADMIN**

- password **0raclE!**

1.4. Install Elastic items

Configuration files are given in next associated sections below. For some of them, some chmod change is needed :

```
$ cd ~/elastic
$ chmod go-w ./*.yml
```

1.4.1. Migration prerequisites

If you are upgrading from a previous version of Elastic, you have to do this before anything :

- Close data senders using Portainer for containers
  - Shutdown the IS, or just disable CgElastic & WmMediator packages
  - Stop Heartbeat, Filebeat, Metricbeat containers
  - No need to stop Logstash if Filebeat is closed
- Check that nothing is coming in Elasticsearch with Kibana, then stop Kibana container
- Stop Elasticsearch container

For now, no data migration has been tried, so no support on it. This will be a fresh new Elasticsearch, and a Kibana with imported dashboards (hoping they still work).

Rename all stopped container, to be able to get the initial name on new containers.

1.4.2. Elasticsearch

> **ℹ** If you are new to the Elastic Stack, learn with the excellent official Kibana tutorial :
> https://www.elastic.co/guide/en/kibana/current/getting-started.html

- Install with docker without x-pack

```
$ docker pull docker.elastic.co/elasticsearch/elasticsearch-oss:6.0.0
```

To start it

```
$ docker run --restart=always -d --name elastic -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node"
docker.elastic.co/elasticsearch/elasticsearch-oss:6.0.0
```

- if elastic stops directly after start with this error

```
max virtual memory areas vm.max_map_count [65530] likely too low, increase to at least [262144]
```

- Then type before retry

```
$ sudo sysctl -w vm.max_map_count=262144
```

1.5. Kibana

- Install with docker without x-pack

```
$ docker pull docker.elastic.co/kibana/kibana-oss:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/kibana.yml
```

- Start the container

```
$ docker run --restart=always -d --name kibana -p 5601:5601 -v ~/elastic/kibana.yml:/usr/share/kibana/config/kibana.yml
docker.elastic.co/kibana/kibana-oss:6.0.0
```

Check that it is up and running : http://frpardge:5601/

Once every application is up, you will be able to declare patterns :

- cgwmbeat-*
- heartbeat-*
- jenkins
- logstash-*
- metricbeat-*
- webmethodsmediator

And to apply some Elasticsearch default index configuration :

- the limit of 1000 fields by index is a bit low, updated to 2000
- default is 5 shards per index, too many for dev
- default is 1 replica, for a single node ES it's 0

```
PUT _template/all
{
  "index_patterns" : ["*"],
  "settings": {
    "index.mapping.total_fields.limit": 2000,
    "index.max_docvalue_fields_search": 400,
    "number_of_shards": 1,
    "number_of_replicas": 0
  }
}
```

Here is something to try, inside the "PUT _template/all", someday, to not have keyword (fixed word) + text (searchable) but only keyword :

```
  "dynamic_templates": [
    {
      "match_mapping_type": "string",
      "mapping": {
        "type": "keyword"
      }
    }
  ]
```

For Elasticsearch monitoring :

```
GET /_cat/indices?v
GET _cluster/health
```

~/elastic/kibana.yml

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "0.0.0.0"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy. This only affects
# the URLs generated by Kibana, your proxy is expected to remove the basePath value before forwarding requests
# to Kibana. This setting cannot end in a slash.
#server.basePath: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://frpardge.corp.nvx.com:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "discover"

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
#elasticsearch.username: "user"
#elasticsearch.password: "pass"

# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# Optional settings that provide the paths to the PEM-format SSL certificate and key files.
# These files validate that your Elasticsearch backend uses the same key files.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key
```

```
# Optional setting that enables you to specify a path to the PEM file for the certificate
# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# To disregard the validity of SSL certificates, change this setting's value to 'none'.
#elasticsearch.ssl.verificationMode: full

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500

# Time in milliseconds to wait for responses from the back end or Elasticsearch. This value
# must be a positive integer.
#elasticsearch.requestTimeout: 30000

# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]

# Header names and values that are sent to Elasticsearch. Any custom headers cannot be overwritten
# by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist configuration.
#elasticsearch.customHeaders: {}

# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to disable.
#elasticsearch.shardTimeout: 0

# Time in milliseconds to wait for Elasticsearch at Kibana startup before retrying.
#elasticsearch.startupTimeout: 5000

# Specifies the path where Kibana creates the process ID file.
#pid.file: /var/run/kibana.pid

# Enables you specify a file where Kibana stores log output.
#logging.dest: stdout

# Set the value of this setting to true to suppress all logging output.
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000
```

1.5.1. Troubleshoot

Here is a list of problems and solutions.

Kibana cannot connect to Elasticsearch

If Kibana cannot connect to Elasticsearch with this message :

```
blocked by: [FORBIDDEN/12/index read-only / allow delete (api)];: [cluster_block_exception] blocked by:
[FORBIDDEN/12/index read-only / allow delete (api)];
```

Then apply these settings :

```
PUT _settings
    {
        "index": {
        "blocks": {
        "read_only_allow_delete": "false"
    }
PUT cgwmbeat-2018.02.16/_settings
    {
        "index": {
        "blocks": {
        "read_only_allow_delete": "false"
    }
```

1.6. Curator

```
$ wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

$ sudo vi /etc/apt/sources.list

 deb [arch=amd64] http://packages.elastic.co/curator/5/debian stable main

$ sudo apt-get update && sudo apt-get install elasticsearch-curator
```

To start it

```
$ curator --config ~/elastic/curator.config.yml --dry-run ~/elastic/curator.delete_indices.yml
$ curator --config ~/elastic/curator.config.yml ~/elastic/curator.delete_indices.yml
```

1.6.1. Automation

- Create below script

~/elastic/curator.sh

```
#!/bin/sh

curator --config ~/elastic/curator.config.yml ~/elastic/curator.delete_indices.yml
```

- Open crontab

```
$  crontab -e
```

- Add this line to launch it at 8:00 everyday

```
0 8 * * * ~/elastic/curator.sh
```

- Exit and save with **Ctrl+X**, **Y**, **Enter**

1.6.2. Configuration

```yaml
---
# Remember, leave a key empty if there is no value.  None will be a string,
# not a Python "NoneType"
client:
  hosts:
    - 127.0.0.1
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth:
  timeout: 30
  master_only: False

logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
```

```yaml
---
# Remember, leave a key empty if there is no value.  None will be a string,
# not a Python "NoneType"
#
# Also remember that all examples have 'disable_action' set to True.  If you
# want to use this action as a template, be sure to set this to False after
# copying it.
# # # #
# curator --config ~/elastic/curator.config.yml --dry-run ~/elastic/curator.delete_indices.yml
# curator --config ~/elastic/curator.config.yml ~/elastic/curator.delete_indices.yml
# # # #
actions:
  1:
    action: delete_indices
    description: Delete indices older than 30 days. No error when no actual deletion.
    options:
      ignore_empty_list: True
    filters:
    - filtertype: age
      source: name
      direction: older
      timestring: '%Y.%m.%d'
      unit: days
      unit_count: 30
```

1.7. Heartbeat

- Pull the image

```
$ docker pull docker.elastic.co/beats/heartbeat:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/heartbeat.yml
```

- Start the container

```
$ docker run --name heartbeat -d -v ~/elastic/heartbeat.yml:/usr/share/heartbeat/heartbeat.yml
docker.elastic.co/beats/heartbeat:6.0.0
```

~/elastic/heartbeat.yml

```yaml
#
# wget --user=svc-fr-pldouane --password=Na9Is4Aw0! https://cdsdouane.pl.s2-eu.nvx.com/jenkins/job/DTXE_P1_CodeReview/
#
heartbeat.monitors:
- name: Jenkins
  type: http
  schedule: '@every 30s'
  urls: ["https://bpmfactory.s2-eu.nvx.com/jenkins/job/CNAV-DGE_P1_Review/"]
  username: svc-fr-bpmfact
  password: ****
  check.request.method: GET
  check.response.status: 200
- name: 'Jenkins Douane'
  type: http
  schedule: '@every 30s'
  urls: ["https://cdsdouane.pl.s2-eu.nvx.com/jenkins/job/DTXE_P1_CodeReview/"]
  username: svc-fr-pldouane
  password: ****
  check.request.method: GET
  check.response.status: 200
- name: 'Gerrit home'
  type: http
  schedule: '@every 30s'
  urls: ["https://bpmfactory.s2-eu.nvx.com/gerrit/changes/?n=25&O=81"]
  username: svc-fr-bpmfact
  password: ****
  check.response.status: 200
- name: 'Gerrit viewFile'
  type: http
  schedule: '@every 30s'
  urls: ["https://bpmfactory.s2-
eu.nvx.com/gerrit/changes/421/revisions/5ab9d4c5cab6a087b936748f2df6550666a502dd/files/Jenkinsfile-2-deploy-to-
dev/diff?context=ALL"]
  username: svc-fr-bpmfact
  password: ****
  check.response.status: 200
- name: 'IS Dev'
  type: http
  schedule: '@every 30s'
  urls: ["http://frpardge:5555"]
  username: Administrator
  password: ****
  check.response.status: 200
- name: Kibana
  type: http
  schedule: '@every 30s'
  urls: ["http://frpardge:5601/app/kibana#/management?_g=()"]
  check.response.status: 200
- name: 'UM Dev'
  type: tcp
  schedule: '@every 30s'
  hosts: ["frpardge:9000"]

heartbeat.scheduler:
  limit: 10

output.elasticsearch:
  hosts: ["frpardge.corp.nvx.com:9200"]

dashboards.enabled: true
```

1.8. Logstash

ℹ️ Install this only if you have files to be parsed and sent to Elasticsearch

- Pull the image

```
$ docker pull docker.elastic.co/logstash/logstash-oss:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/logstash-pipelines/logstash.conf
```

- Start the container

```
$ docker run --restart=always --name logstash -d -p 5043:5043 -v ~/elastic/logstash-
pipelines/:/usr/share/logstash/pipeline/ docker.elastic.co/logstash/logstash-oss:6.0.0
```

~/elastic/logstash-pipelines/logstash.conf

```
input {
    beats {
        port => "5043"
    }
}

filter {
  if [fields][log_type] == "perflog" {
    grok {
        match => { "message" => "%{TIMESTAMP_ISO8601:timestamp} INFO  PERFORMANCES - \[%{GREEDYDATA:package}\]
%{WORD:method}\(\) completed successfully in %{NUMBER:duration:int} ms" }
    }
  }
  else {
    grok {
        match => { "message" => "\[%{TIMESTAMP_ISO8601:timestamp}\] \[%{NOTSPACE:wMCode}\] %{GREEDYDATA:textMsg}" }
    }
  }
  date {
    match => [ "timestamp", ISO8601 ]
    timezone => "Europe/Paris"
    target => "@timestamp"
  }
}

output {
    elasticsearch {
        hosts => [ "frpardge.corp.nvx.com:9200" ]
    }
    #stdout { codec => rubydebug }
}
```

1.9. Filebeat

ℹ Install this only if you have files to be parsed and sent to Elasticsearch

- Pull the image

```
$ docker pull docker.elastic.co/beats/filebeat:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/filebeat.yml
```

- Start the container

```
$ docker run --name filebeat -d -v /opt/sagis/IntegrationServer/instances/default/logs/:/islogs/ -v
~/elastic/filebeat.yml:/usr/share/filebeat/filebeat.yml docker.elastic.co/beats/filebeat:6.0.0
```

~/elastic/filebeat.yml

```yaml
filebeat.prospectors:

- type: log
  paths:
    - /islogs/log4j2/perfs.log
  fields: {log_type: perflog}

- type: log
  paths:
    - /islogs/server.log
  multiline.pattern: '^\[20'
  multiline.negate: true
  multiline.match: after
  fields: {log_type: serverlog}

output.logstash:
  hosts: ["frpardge.corp.nvx.com:5043"]
```

1.10. Metricbeat

This chmod has to be done again after each VM reboot before starting Metricbeat :

```
$ sudo chmod 777 /var/run/docker.sock
```

- Pull the image

```
$ docker pull docker.elastic.co/beats/metricbeat:6.0.0
```

- Create the file described at the end of this section

```
~/elastic/metricbeat.yml
```

- Start the container

```
$ docker run --name metricbeat -d -v /var/run/docker.sock:/var/run/docker.sock -v
~/elastic/metricbeat.yml:/usr/share/metricbeat/metricbeat.yml --volume=/proc:/hostfs/proc:ro --volume
=/sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro --volume=/:/hostfs:ro --net=host docker.elastic.co/beats/metricbeat:6.0.0
metricbeat -e -system.hostfs=/hostfs
```

To test you CPU graphs, with the proper handling of the cores, you can use stress application to load one or multiple cores :

```
$ sudo apt-get install stress
$ stress --cpu 2
```

```yaml
metricbeat.modules:
- module: system
  period: 10s
  metricsets:
    - cpu
    #- load
    - memory
    #- network
    - process
    - process_summary
    #- core
    #- diskio
    #- socket
  processes: ['.*']
  process.include_top_n:
    by_cpu: 10      # include top processes by CPU
    by_memory: 10   # include top processes by memory

- module: system
  period: 1m
  metricsets:
    - filesystem
    - fsstat
  processors:
  - drop_event.when.regexp:
      system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|hostfs|run|var)($|/)'

- module: docker
  metricsets:
    #- container
    - cpu
    #- diskio
    #- healthcheck
    #- image
    #- info
    - memory
    #- network
  hosts: ["unix:///var/run/docker.sock"]
  period: 10s

output.elasticsearch:
  hosts: ["frpardge.corp.nvx.com:9200"]

metricbeat.config.modules:
  path: /usr/share/metricbeat/metricbeat.yml
  reload.enabled: true
  reload.period: 60s
```

1.11. Grafana

```
$ wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.4.3_amd64.deb

$ sudo apt-get install -y adduser libfontconfig

$ sudo dpkg -i grafana_4.4.3_amd64.deb
```

To start it

```
$ sudo service grafana-server start
```

To auto start it at boot time

```
$ sudo update-rc.d grafana-server defaults
```

1.12. Jaeger Tracing (OpenZipkin-like)

To start it

```
$ docker run --restart=always --name jaeger -d -p5775:5775/udp -p6831:6831/udp -p5778:5778 -p16686:16686
jaegertracing/all-in-one:latest
```

2. Appendix

2.1. Revision marks

Differences since last tag