**SOURCE CODE**

Github Link : <https://github.com/karthikkoppula/phase2.git>

Elasticsearch configuration

---

## Default Elasticsearch configuration from Elasticsearch base image.

## https://github.com/elastic/elasticsearch/blob/main/distribution/docker/src/docker/config/elasticsearch.yml

#

cluster.name: docker-cluster

network.host: 0.0.0.0

## X-Pack settings

## see https://www.elastic.co/guide/en/elasticsearch/reference/current/security-settings.html

#

xpack.license.self\_generated.type: trial

xpack.security.enabled: true

Docker Ignore

# Ignore Docker build files

Dockerfile

.dockerignore

# Ignore OS artifacts

\*\*/.DS\_Store

Docker File

ARG ELASTIC\_VERSION

# https://www.docker.elastic.co/

FROM docker.elastic.co/elasticsearch/elasticsearch:${ELASTIC\_VERSION}

# Add your elasticsearch plugins setup here

# Example: RUN elasticsearch-plugin install analysis-icu

Curator.yml

# Curator configuration

# https://www.elastic.co/guide/en/elasticsearch/client/curator/current/configfile.html

elasticsearch:

  client:

    hosts: [ http://elasticsearch:9200 ]

  other\_settings:

    username: elastic

    password: ${ELASTIC\_PASSWORD}

logging:

  loglevel: INFO

  logformat: default

Delete Log files curator

actions:

1:

action: delete\_indices

description: >-

Delete indices. Find which to delete by first limiting the list to

logstash- prefixed indices. Then further filter those to prevent deletion

of anything less than the number of days specified by unit\_count.

Ignore the error if the filter does not result in an actionable list of

indices (ignore\_empty\_list) and exit cleanly.

options:

ignore\_empty\_list: True

disable\_action: False

filters:

- filtertype: pattern

kind: prefix

value: logstash-

- filtertype: age

source: creation\_date

direction: older

unit: days

unit\_count: 2

Docker – Compose.yml

version: '3.7'

services:

curator:

build:

context: extensions/curator/

init: true

volumes:

- ./extensions/curator/config/curator.yml:/.curator/curator.yml:ro,Z

- ./extensions/curator/config/delete\_log\_files\_curator.yml:/.curator/delete\_log\_files\_curator.yml:ro,Z

environment:

ELASTIC\_PASSWORD: ${ELASTIC\_PASSWORD:-}

networks:

- elk

depends\_on:

- elasticsearch

Extensions>Curator>DockerFile

FROM untergeek/curator:8.0.2

USER root

RUN >>/var/spool/cron/crontabs/nobody \

echo '\* \* \* \* \* /curator/curator /.curator/delete\_log\_files\_curator.yml'

ENTRYPOINT ["crond"]

CMD ["-f", "-d8"]

Enterprise search.yml

---

## Enterprise Search core configuration

## https://www.elastic.co/guide/en/enterprise-search/current/configuration.html

#

## --------------------- REQUIRED ---------------------

# Encryption keys to protect application secrets.

secret\_management.encryption\_keys:

  # example:

  #- 680f94e568c90364bedf927b2f0f49609702d3eab9098688585a375b14274546

## ----------------------------------------------------

# IP address Enterprise Search listens on

ent\_search.listen\_host: 0.0.0.0

# URL at which users reach Enterprise Search / Kibana

ent\_search.external\_url: http://localhost:3002

kibana.host: http://localhost:5601

# Elasticsearch URL and credentials

elasticsearch.host: http://elasticsearch:9200

elasticsearch.username: elastic

elasticsearch.password: ${ELASTIC\_PASSWORD}

# Allow Enterprise Search to modify Elasticsearch settings. Used to enable auto-creation of Elasticsearch indexes.

allow\_es\_settings\_modification: true

Enterprise Search- Compose.yml

version: '3.7'

services:

enterprise-search:

build:

context: extensions/enterprise-search/

args:

ELASTIC\_VERSION: ${ELASTIC\_VERSION}

volumes:

- ./extensions/enterprise-search/config/enterprise-search.yml:/usr/share/enterprise-search/config/enterprise-search.yml:ro,Z

environment:

JAVA\_OPTS: -Xms2g -Xmx2g

ENT\_SEARCH\_DEFAULT\_PASSWORD: 'changeme'

ELASTIC\_PASSWORD: ${ELASTIC\_PASSWORD:-}

ports:

- 3002:3002

networks:

- elk

depends\_on:

- elasticsearch

Filebeat.yml

## Filebeat configuration

## https://github.com/elastic/beats/blob/main/deploy/docker/filebeat.docker.yml

#

name: filebeat

filebeat.config:

modules:

path: ${path.config}/modules.d/\*.yml

reload.enabled: false

filebeat.autodiscover:

providers:

# The Docker autodiscover provider automatically retrieves logs from Docker

# containers as they start and stop.

- type: docker

hints.enabled: true

processors:

- add\_cloud\_metadata: ~

monitoring:

enabled: true

elasticsearch:

username: beats\_system

password: ${BEATS\_SYSTEM\_PASSWORD}

output.elasticsearch:

hosts: [ http://elasticsearch:9200 ]

username: filebeat\_internal

password: ${FILEBEAT\_INTERNAL\_PASSWORD}

## HTTP endpoint for health checking

## https://www.elastic.co/guide/en/beats/filebeat/current/http-endpoint.html

#

http:

enabled: true

host: 0.0.0.0

filebeat- compose.yml

version: '3.7'

services:

  filebeat:

    build:

      context: extensions/filebeat/

      args:

        ELASTIC\_VERSION: ${ELASTIC\_VERSION}

    # Run as 'root' instead of 'filebeat' (uid 1000) to allow reading

    # 'docker.sock' and the host's filesystem.

    user: root

    command:

      # Log to stderr.

      - -e

      # Disable config file permissions checks. Allows mounting

      # 'config/filebeat.yml' even if it's not owned by root.

      # see: https://www.elastic.co/guide/en/beats/libbeat/current/config-file-permissions.html

      - --strict.perms=false

    volumes:

      - ./extensions/filebeat/config/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro,Z

      - type: bind

        source: /var/lib/docker/containers

        target: /var/lib/docker/containers

        read\_only: true

      - type: bind

        source: /var/run/docker.sock

        target: /var/run/docker.sock

        read\_only: true

    environment:

      FILEBEAT\_INTERNAL\_PASSWORD: ${FILEBEAT\_INTERNAL\_PASSWORD:-}

      BEATS\_SYSTEM\_PASSWORD: ${BEATS\_SYSTEM\_PASSWORD:-}

    networks:

      - elk

    depends\_on:

      - elasticsearch

AgentApmserver- compose.yml

version: '3.7'

# Example of Fleet-enrolled Elastic Agent pre-configured with an agent policy

# for running the APM Server integration (see kibana.yml).

#

# Run with

#   docker-compose \

#     -f docker-compose.yml \

#     -f extensions/fleet/fleet-compose.yml \

#     -f extensions/fleet/agent-apmserver-compose.yml \

#     up

services:

  apm-server:

    build:

      context: extensions/fleet/

      args:

        ELASTIC\_VERSION: ${ELASTIC\_VERSION}

    volumes:

      - apm-server:/usr/share/elastic-agent/state:Z

    environment:

      FLEET\_ENROLL: '1'

      FLEET\_TOKEN\_POLICY\_NAME: Agent Policy APM Server

      FLEET\_INSECURE: '1'

      FLEET\_URL: http://fleet-server:8220

      # Enrollment.

      # (a) Auto-enroll using basic authentication

      ELASTICSEARCH\_USERNAME: elastic

      ELASTICSEARCH\_PASSWORD: ${ELASTIC\_PASSWORD:-}

      # (b) Enroll using a pre-generated enrollment token

      #FLEET\_ENROLLMENT\_TOKEN: <enrollment\_token>

    ports:

      - 8200:8200

    hostname: apm-server

    # Elastic Agent does not retry failed connections to Kibana upon the initial enrollment phase.

    restart: on-failure

    networks:

      - elk

    depends\_on:

      - elasticsearch

      - kibana

      - fleet-server

volumes:

  apm-server:

fleet-compose. Yml

version: '3.7'

services:

  fleet-server:

    build:

      context: extensions/fleet/

      args:

        ELASTIC\_VERSION: ${ELASTIC\_VERSION}

    volumes:

      - fleet-server:/usr/share/elastic-agent/state:Z

    environment:

      FLEET\_SERVER\_ENABLE: '1'

      FLEET\_SERVER\_INSECURE\_HTTP: '1'

      FLEET\_SERVER\_HOST: 0.0.0.0

      FLEET\_SERVER\_POLICY\_ID: fleet-server-policy

      # Fleet plugin in Kibana

      KIBANA\_FLEET\_SETUP: '1'

      # Enrollment.

      # (a) Auto-enroll using basic authentication

      ELASTICSEARCH\_USERNAME: elastic

      ELASTICSEARCH\_PASSWORD: ${ELASTIC\_PASSWORD:-}

      # (b) Enroll using a pre-generated service token

      #FLEET\_SERVER\_SERVICE\_TOKEN: <service\_token>

    ports:

      - 8220:8220

    hostname: fleet-server

    # Elastic Agent does not retry failed connections to Kibana upon the initial enrollment phase.

    restart: on-failure

    networks:

      - elk

    depends\_on:

      - elasticsearch

      - kibana

volumes:

  fleet-server:

heartbeat.yml

## Heartbeat configuration

## https://github.com/elastic/beats/blob/main/deploy/docker/heartbeat.docker.yml

#

name: heartbeat

heartbeat.monitors:

- type: http

  schedule: '@every 5s'

  urls:

    - http://elasticsearch:9200

  username: heartbeat\_internal

  password: ${HEARTBEAT\_INTERNAL\_PASSWORD}

- type: icmp

  schedule: '@every 5s'

  hosts:

    - elasticsearch

processors:

- add\_cloud\_metadata: ~

monitoring:

  enabled: true

  elasticsearch:

    username: beats\_system

    password: ${BEATS\_SYSTEM\_PASSWORD}

output.elasticsearch:

  hosts: [ http://elasticsearch:9200 ]

  username: heartbeat\_internal

  password: ${HEARTBEAT\_INTERNAL\_PASSWORD}

## HTTP endpoint for health checking

## https://www.elastic.co/guide/en/beats/heartbeat/current/http-endpoint.html

#

http:

  enabled: true

  host: 0.0.0.0

heartbeat-compose.yml

version: '3.7'

services:

  heartbeat:

    build:

      context: extensions/heartbeat/

      args:

        ELASTIC\_VERSION: ${ELASTIC\_VERSION}

    command:

      # Log to stderr.

      - -e

      # Disable config file permissions checks. Allows mounting

      # 'config/heartbeat.yml' even if it's not owned by root.

      # see: https://www.elastic.co/guide/en/beats/libbeat/current/config-file-permissions.html

      - --strict.perms=false

    volumes:

      - ./extensions/heartbeat/config/heartbeat.yml:/usr/share/heartbeat/heartbeat.yml:ro,Z

    environment:

      HEARTBEAT\_INTERNAL\_PASSWORD: ${HEARTBEAT\_INTERNAL\_PASSWORD:-}

      BEATS\_SYSTEM\_PASSWORD: ${BEATS\_SYSTEM\_PASSWORD:-}

    networks:

      - elk

    depends\_on:

      - elasticsearch

Logspout-compose.yml

version: '3.7'

services:

  logspout:

    build:

      context: extensions/logspout

    volumes:

      - type: bind

        source: /var/run/docker.sock

        target: /var/run/docker.sock

        read\_only: true

    environment:

      ROUTE\_URIS: logstash://logstash:50000

      LOGSTASH\_TAGS: docker-elk

    networks:

      - elk

    depends\_on:

      - logstash

    restart: on-failure

Kibana.yml

---

## Default Kibana configuration from Kibana base image.

## https://github.com/elastic/kibana/blob/main/src/dev/build/tasks/os\_packages/docker\_generator/templates/kibana\_yml.template.ts

#

server.name: kibana

server.host: 0.0.0.0

elasticsearch.hosts: [ http://elasticsearch:9200 ]

monitoring.ui.container.elasticsearch.enabled: true

monitoring.ui.container.logstash.enabled: true

## X-Pack security credentials

#

elasticsearch.username: kibana\_system

elasticsearch.password: ${KIBANA\_SYSTEM\_PASSWORD}

## Fleet

## https://www.elastic.co/guide/en/kibana/current/fleet-settings-kb.html

#

xpack.fleet.agents.fleet\_server.hosts: [ http://fleet-server:8220 ]

xpack.fleet.outputs:

  - id: fleet-default-output

    name: default

    type: elasticsearch

    hosts: [ http://elasticsearch:9200 ]

    is\_default: true

    is\_default\_monitoring: true

xpack.fleet.packages:

  - name: fleet\_server

    version: latest

  - name: system

    version: latest

  - name: elastic\_agent

    version: latest

  - name: apm

    version: latest

xpack.fleet.agentPolicies:

  - name: Fleet Server Policy

    id: fleet-server-policy

    description: Static agent policy for Fleet Server

    monitoring\_enabled:

      - logs

      - metrics

    package\_policies:

      - name: fleet\_server-1

        package:

          name: fleet\_server

      - name: system-1

        package:

          name: system

      - name: elastic\_agent-1

        package:

          name: elastic\_agent

  - name: Agent Policy APM Server

    id: agent-policy-apm-server

    description: Static agent policy for the APM Server integration

    monitoring\_enabled:

      - logs

      - metrics

    package\_policies:

      - name: system-1

        package:

          name: system

      - name: elastic\_agent-1

        package:

          name: elastic\_agent

      - name: apm-1

        package:

          name: apm

        # See the APM package manifest for a list of possible inputs.

        # https://github.com/elastic/apm-server/blob/v8.5.0/apmpackage/apm/manifest.yml#L41-L168

        inputs:

          - type: apm

            vars:

              - name: host

                value: 0.0.0.0:8200

              - name: url

                value: http://apm-server:8200

logstash.yml

---

## Default Logstash configuration from Logstash base image.

## https://github.com/elastic/logstash/blob/main/docker/data/logstash/config/logstash-full.yml

#

http.host: 0.0.0.0

node.name: logstash