OBSERVABILIDADE

Tags: #observabilidade #elasticsearch #kibana #monitoramento #log

O que é ?

Observabilidade é uma ferramenta para verificar os estados de cada parte de um software.

- Métricas
  Realiza uma análise em tempo real dos dados afim de prever e evitar determinadas ações.
  - Negócio
    Analísa questões relacionada as regras de negócio da aplicação, como: quantos usuários logados, quantas vendas foram realizadas, quantas aulas foram assistidas ...
  - Técnicas
    As métricas técnicas analisa questões relacionada a infraestrutura, hardware, software ... Como: Quanto de CPU está sendo usado, qual o gasto de memória ram ...
- Logs
  Monitoramento de algo que já aconteceu para correção ou melhoria.

ElasticSearch

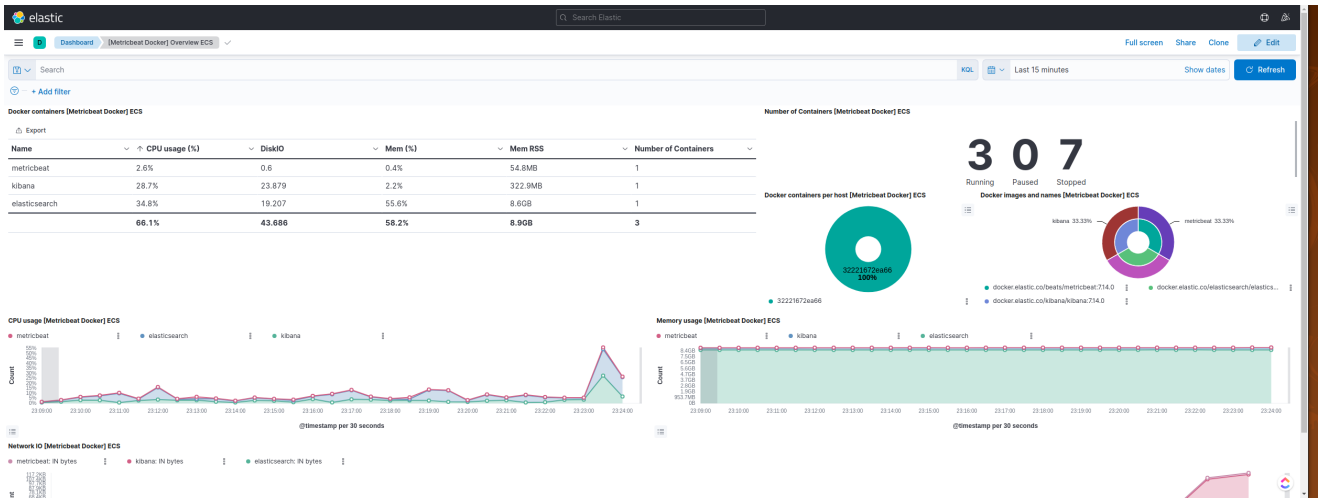
Ferramenta para armazenar, buscar e analisar grandes quantidades de dados.

Vantagens

1. Armazenamento e Indexação
   Ele tem uma alta capacidade para armazenar quantidades gigantes de dados e com indices para tornar a pesquisa mais eficiente
2. Pesquisa
   Ele te permite realizar pesquisa aos dados extremamente personalizadas como consultas por: texto, campo, data ... Tudo isso utilizando indices para a pesquisa ser extremamente eficiente.
3. Escala
   Permite a escala horizontal adicionando novos servidores para lidar com quantidades gigantescas de dados.

Kibana

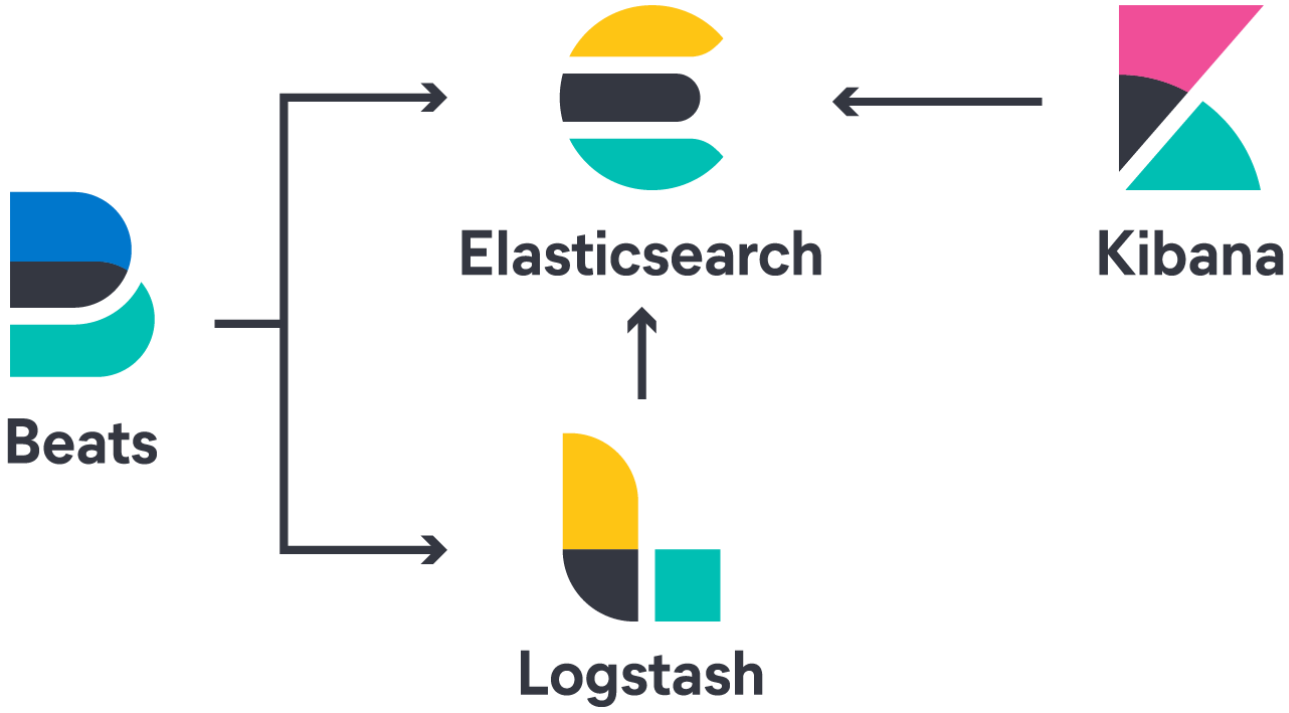
Plugin para visualização de dados indexados no ElasticSearch.



- Logs
- Métricas
- Filtros de dados
- Dashboards estáticos e integrativos
- Machine Learning e Inteligência artificial

Beats

Plataforma responsavel por coletar dados de centenas ou milhares de aplicações para e enviar para o ElasticSearch.



- MetricBeat
  Coleta as métricas como: CPU, RAM ... de uma ou milhares aplicações/servidores e enviam para o ES
- HeartBeat
  Analisa a disponibilidade e o tempo de resposta de um serviço

Inicializar ElascticSearch e Kibana

Crie o docker-compose.yaml

```
version: '3'
services:
  elasticsearch:
    image:
docker.elastic.co/elasticsearch/elasticsearch:7.14.0
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
    ports:
```

```
      - 9200:9200
    volumes:
      - esdata:/usr/share/elasticsearch/data
    networks:
      - elastic


  kibana:
    image: docker.elastic.co/kibana/kibana:7.14.0
    container_name: kibana
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch
    networks:
      - elastic

networks:
  elastic:
    driver: bridge

volumes:
  esdata:
    driver: local
```

Inicialize com:

```
docker compose up
```

Metricbeat

adicione ao docker-compose.yaml

```
version: '3'
services:
  elasticsearch:
```

```
    image:
docker.elastic.co/elasticsearch/elasticsearch:7.14.0
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
    ports:
      - 9200:9200
    volumes:
      - esdata:/usr/share/elasticsearch/data
    networks:
      - elastic


  kibana:
    image: docker.elastic.co/kibana/kibana:7.14.0
    container_name: kibana
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch
    networks:
      - elastic


  metricbeat:
    image: docker.elastic.co/beats/metricbeat:7.14.0
    container_name: metricbeat
    user: root
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro
      - /proc:/hostfs/proc:ro
      - /:/hostfs:ro
      -
./metricbeat.yml:/usr/share/metricbeat/metricbeat.yml
    command: metricbeat -e -system.hostfs=/hostfs
    depends_on:
      - kibana
```

```
    networks:
      - elastic


networks:
  elastic:
    driver: bridge

volumes:
  esdata:
    driver: local
```

Crie o arquivo de configuração

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

- module: elasticsearch
  metricsets: ["node", "node_stats", "cluster_stats",
"index"]
  period: 10s
  hosts: ["elasticsearch:9200"]


output.elasticsearch:
  hosts: ["elasticsearch:9200"]

setup.kibana:
  host: "kibana:5601"

setup.dashboards.enabled: true
```

De permissão ao arquivo de configuração

```
sudo chown root metricbeat.yml
sudo chmod a+r metricbeat.yml
```

Adicione:

```
restart: on-failure
```

Heartbeat

Crie o heartbeat.yml

```
heartbeat.monitors:
- type: http
  schedule: '@every 5s'
  urls:
    - https://plataforma.pythonando.com.br
    - http://elasticsearch:9200
    - http://kibana:5601

processors:
- add_cloud_metadata: ~
- add_docker_metadata: ~

output.elasticsearch:
  hosts: ["elasticsearch:9200"]

setup.kibana:
  host: "kibana:5601"
```

Adicione o service no docker compose

```
 heartbeat:
    image: docker.elastic.co/beats/heartbeat:7.13.0
    container_name: heartbeat
```

```
    volumes:
    - ./heartbeat.yml:/usr/share/heartbeat/heartbeat.yml
    environment:
    - setup.kibana.host=kibana:5601
    networks:
      - elastic
```

De permissão ao arquivo de configuração

```
sudo chmod 644 heartbeat.yml
```

Inicialize o container!

APM

Crie o arquivo de configuração apm-server.yml

```
########################## APM Server Configuration
#########################

################################### APM Server
###################################

apm-server:
  # Defines the host and port the server is listening
on. Use "unix:/path/to.sock" to listen on a unix domain
socket.
  host: "0.0.0.0:8200"

  # Maximum permitted size in bytes of a request's
header accepted by the server to be processed.
  #max_header_size: 1048576

  # Maximum amount of time to wait for the next incoming
request before underlying connection is closed.
  #idle_timeout: 45s
```

```
  # Maximum permitted duration for reading an entire
request.
  #read_timeout: 30s

  # Maximum permitted duration for writing a response.
  #write_timeout: 30s

  # Maximum duration before releasing resources when
shutting down the server.
  #shutdown_timeout: 5s

  # Maximum permitted size in bytes of an event accepted
by the server to be processed.
  #max_event_size: 307200

  # Maximum number of new connections to accept
simultaneously (0 means unlimited).
  #max_connections: 0

  # Custom HTTP headers to add to all HTTP responses,
e.g. for security policy compliance.
  #response_headers:
  #  X-My-Header: Contents of the header

  # If true (default), APM Server captures the IP of the
instrumented service
  # or the IP and User Agent of the real user (RUM
requests).
  #capture_personal_data: true

  # If specified, APM Server will record this value in
events which have no service environment
  # defined, and add it to agent configuration queries
to Kibana when none is specified in the
  # request from the agent.
  #default_service_environment:

  # Enable APM Server Golang expvar support
```

```
(https://golang.org/pkg/expvar/).
  #expvar:
    #enabled: false

    # Url to expose expvar.
    #url: "/debug/vars"

  # A pipeline is a definition of processors applied to
documents when ingesting them to Elasticsearch.
  # Using pipelines involves two steps:
  # (1) registering a pipeline
  # (2) applying a pipeline during data ingestion (see
`output.elasticsearch.pipeline`)
  #
  # You can manually register a pipeline, or use this
configuration option to ensure
  # the pipeline is loaded and registered at the
configured Elasticsearch instances.
  # Find the default pipeline configuration at
`ingest/pipeline/definition.json`.
  # Automatic pipeline registration requires the
`output.elasticsearch` to be enabled and configured.
  #register.ingest.pipeline:
    # Registers APM pipeline definition in Elasticsearch
on APM Server startup. Defaults to true.
    #enabled: true
    # Overwrites existing APM pipeline definition in
Elasticsearch. Defaults to false.
    #overwrite: false


  #---------------------------- APM Server - Secure
Communication with Agents ----------------------------

  # Enable secure communication between APM agents and
the server. By default ssl is disabled.
  #ssl:
    #enabled: false
```

```
    # Path to file containing the certificate for server
authentication.
    # Needs to be configured when ssl is enabled.
    #certificate: ''

    # Path to file containing server certificate key.
    # Needs to be configured when ssl is enabled.
    #key: ''

    # Optional configuration options for ssl
communication.

    # Passphrase for decrypting the Certificate Key.
    # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
    #key_passphrase: ''

    # List of supported/valid protocol versions. By
default TLS versions 1.1 up to 1.3 are enabled.
    #supported_protocols: [TLSv1.1, TLSv1.2, TLSv1.3]

    # Configure cipher suites to be used for SSL
connections.
    # Note that cipher suites are not configurable for
TLS 1.3.
    #cipher_suites: []

    # Configure curve types for ECDHE based cipher
suites.
    #curve_types: []

  # The APM Server endpoints can be secured by
configuring a secret token or enabling the usage of API
keys. Both
  # options can be enabled in parallel, allowing Elastic
APM agents to chose whichever mechanism they support.
  # As soon as one of the options is enabled, requests
```

```
without a valid token are denied by the server. An
exception
  # to this are requests to any enabled RUM endpoint.
RUM endpoints are generally not secured by any token.
  #
  # Configure authorization via a common `secret_token`.
By default it is disabled.
  # Agents include the token in the following format:
Authorization: Bearer <secret-token>.
  # It is recommended to use an authorization token in
combination with SSL enabled,
  # and save the token in the apm-server keystore.
  #secret_token:

  # Enable API key authorization by setting enabled to
true. By default API key support is disabled.
  # Agents include a valid API key in the following
format: Authorization: ApiKey <token>.
  # The key must be the base64 encoded representation of
the API key's "id:key".
  #api_key:
    #enabled: false

    # Restrict how many unique API keys are allowed per
minute. Should be set to at least the amount of
different
    # API keys configured in your monitored services.
Every unique API key triggers one request to
Elasticsearch.
    #limit: 100

    # API keys need to be fetched from Elasticsearch. If
nothing is configured, configuration settings from the
    # output section will be reused.
    # Note that configuration needs to point to a
secured Elasticsearch cluster that is able to serve API
key requests.
    #elasticsearch:
```

```
      #hosts: ["localhost:9200"]

      #protocol: "http"

      # Username and password are only needed for the
apm-server apikey sub-command, and they are ignored
otherwise
      # See `apm-server apikey --help` for details.
      #username: "elastic"
      #password: "changeme"

      # Optional HTTP Path.
      #path: ""

      # Proxy server url.
      #proxy_url: ""
      #proxy_disable: false

      # Configure http request timeout before failing an
request to Elasticsearch.
      #timeout: 5s

      # Enable custom SSL settings. Set to false to
ignore custom SSL settings for secure communication.
      #ssl.enabled: true

      # Optional SSL configuration options. SSL is off
by default, change the `protocol` option if you want to
enable `https`.
      #
      # Control the verification of Elasticsearch
certificates. Valid values are:
      # * full, which verifies that the provided
certificate is signed by a trusted
      # authority (CA) and also verifies that the
server's hostname (or IP address)
      # matches the names identified within the
certificate.
```

```
      # * strict, which verifies that the provided
certificate is signed by a trusted
      # authority (CA) and also verifies that the
server's hostname (or IP address)
      # matches the names identified within the
certificate. If the Subject Alternative
      # Name is empty, it returns an error.
      # * certificate, which verifies that the provided
certificate is signed by a
      # trusted authority (CA), but does not perform any
hostname verification.
      #  * none, which performs no verification of the
server's certificate. This
      # mode disables many of the security benefits of
SSL/TLS and should only be used
      # after very careful consideration. It is
primarily intended as a temporary
      # diagnostic mechanism when attempting to resolve
TLS errors; its use in
      # production environments is strongly discouraged.
      #ssl.verification_mode: full

      # List of supported/valid TLS versions. By default
all TLS versions 1.0 up to
      # 1.2 are enabled.
      #ssl.supported_protocols: [TLSv1.0, TLSv1.1,
TLSv1.2]

      # List of root certificates for HTTPS server
verifications.
      #ssl.certificate_authorities:
["/etc/pki/root/ca.pem"]

      # Certificate for SSL client authentication.
      #ssl.certificate: "/etc/pki/client/cert.pem"

      # Client Certificate Key
      #ssl.key: "/etc/pki/client/cert.key"
```

```
      # Optional passphrase for decrypting the
Certificate Key.
      # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
      #ssl.key_passphrase: ''

      # Configure cipher suites to be used for SSL
connections.
      #ssl.cipher_suites: []

      # Configure curve types for ECDHE based cipher
suites.
      #ssl.curve_types: []

      # Configure what types of renegotiation are
supported. Valid options are
      # never, once, and freely. Default is never.
      #ssl.renegotiation: never


  #----------------------------- APM Server - RUM Real
User Monitoring -----------------------------

  # Enable Real User Monitoring (RUM) Support. By
default RUM is disabled.
  # RUM does not support token based authorization.
Enabled RUM endpoints will not require any authorization
  # token configured for other endpoints.
  rum:
    enabled: true

    #event_rate:

      # Defines the maximum amount of events allowed to
be sent to the APM Server RUM
      # endpoint per IP per second. Defaults to 300.
      #limit: 300
```

```
      # An LRU cache is used to keep a rate limit per IP
for the most recently seen IPs.
      # This setting defines the number of unique IPs
that can be tracked in the cache.
      # Sites with many concurrent clients should
consider increasing this limit. Defaults to 1000.
      #lru_size: 1000

    #-- General RUM settings

    # A list of service names to allow, to limit
service-specific indices and data streams
    # created for unauthenticated RUM events.
    # If the list is empty, any service name is allowed.
    #allow_service_names: []

    # A list of permitted origins for real user
monitoring.
    # User-agents will send an origin header that will
be validated against this list.
    # An origin is made of a protocol scheme, host and
port, without the url path.
    # Allowed origins in this setting can have * to
match anything (eg.: http://*.example.com)
    # If an item in the list is a single '*', everything
will be allowed.
    allow_origins: ['*']

    # A list of Access-Control-Allow-Headers to allow
RUM requests, in addition to "Content-Type",
    # "Content-Encoding", and "Accept"
    #allow_headers: []

    # Custom HTTP headers to add to RUM responses, e.g.
for security policy compliance.
    #response_headers:
    #  X-My-Header: Contents of the header
```

```
    # Regexp to be matched against a stacktrace frame's
`file_name` and `abs_path` attributes.
    # If the regexp matches, the stacktrace frame is
considered to be a library frame.
    library_pattern: "node_modules|bower_components|~"

    # Regexp to be matched against a stacktrace frame's
`file_name`.
    # If the regexp matches, the stacktrace frame is not
used for calculating error groups.
    # The default pattern excludes stacktrace frames
that have a filename starting with '/webpack'
    exclude_from_grouping: "^/webpack"

    # If a source map has previously been uploaded,
source mapping is automatically applied.
    # to all error and transaction documents sent to the
RUM endpoint.
    source_mapping:

      # Sourcemapping is enabled by default.
      enabled: true

      # Source maps are always fetched from
Elasticsearch, by default using the output.elasticsearch
configuration.
      # A different instance must be configured when
using any other output.
      # This setting only affects sourcemap reads - the
output determines where sourcemaps are written.
      elasticsearch:
        # Array of hosts to connect to.
        # Scheme and port can be left out and will be
set to the default (`http` and `9200`).
        # In case you specify and additional path, the
scheme is required: `http://localhost:9200/path`.
        # IPv6 addresses should always be defined as:
```

```
`https://[2001:db8::1]:9200`.
        hosts: ["localhost:9200"]

        # Protocol - either `http` (default) or `https`.
        #protocol: "https"

        # Authentication credentials - either API key or
username/password.
        #api_key: "id:api_key"
        username: "elastic"
        password: "changeme"

      # The `cache.expiration` determines how long a
source map should be cached before fetching it again
from Elasticsearch.
      # Note that values configured without a time unit
will be interpreted as seconds.
      #cache:
        expiration: 5m

      # Source maps are stored in a separate index.
      # If the default index pattern for source maps at
'outputs.elasticsearch.indices'
      # is changed, a matching index pattern needs to be
specified here.
      index_pattern: "apm-*-sourcemap*"

  #----------------------------- APM Server - Agent
Configuration -----------------------------
  # Directly specify agent configuration. If
`agent_config` is set, agent
  # configuration under `kibana` will be ignored.
  # An agent's incoming configuration request will be
matched to an
  # agent_config with the following precedence:
  # - service.name and service.environment match an
agent_config
  # - service.name matches an agent_config,
```

```
service.environment == ""
  # - service.environment matches an agent_config,
service.name == ""
  # - an agent_config without a name or environment set
  # An empty result is is returned if no matching result
is found.
  # agent_config:
  # - service.name: ten_percent
  #   service.environment: production
  #   config:
  #     capture_body: off
  #     capture_body: true
  #     log_level: info
  #     recording: true
  #     transaction_sample_rate: 0.1
  # - service.name: frontend
  #   agent.name: rum-js
  #   config:
  #     transaction_sample_rate: 0.1

  # When using APM agent configuration, information
fetched from Kibana will be cached in memory for some
time.
  # Specify cache key expiration via this setting.
Default is 30 seconds.
  #agent.config.cache.expiration: 30s

  kibana:
    # For APM Agent configuration in Kibana, enabled
must be true.
    enabled: true

    # Scheme and port can be left out and will be set to
the default (`http` and `5601`).
    # In case you specify an additional path, the scheme
is required: `http://localhost:5601/path`.
    # IPv6 addresses should always be defined as:
`https://[2001:db8::1]:5601`.
```

```
    host: "kibana:5601"

    # Optional protocol and basic auth credentials.
    #protocol: "https"
    #username: "elastic"
    #password: "changeme"

    # Optional HTTP path.
    #path: ""

    # Enable custom SSL settings. Set to false to ignore
custom SSL settings for secure communication.
    #ssl.enabled: true

    # Optional SSL configuration options. SSL is off by
default, change the `protocol` option if you want to
enable `https`.
    #
    # Control the verification of Kibana certificates.
Valid values are:
    # * full, which verifies that the provided
certificate is signed by a trusted
    # authority (CA) and also verifies that the server's
hostname (or IP address)
    # matches the names identified within the
certificate.
    # * strict, which verifies that the provided
certificate is signed by a trusted
    # authority (CA) and also verifies that the server's
hostname (or IP address)
    # matches the names identified within the
certificate. If the Subject Alternative
    # Name is empty, it returns an error.
    # * certificate, which verifies that the provided
certificate is signed by a
    # trusted authority (CA), but does not perform any
hostname verification.
    #  * none, which performs no verification of the
```

```
server's certificate. This
    # mode disables many of the security benefits of
SSL/TLS and should only be used
    # after very careful consideration. It is primarily
intended as a temporary
    # diagnostic mechanism when attempting to resolve
TLS errors; its use in
    # production environments is strongly discouraged.
    #ssl.verification_mode: full

    # List of supported/valid TLS versions. By default
all TLS versions 1.0 up to
    # 1.2 are enabled.
    #ssl.supported_protocols: [TLSv1.0, TLSv1.1,
TLSv1.2]

    # List of root certificates for HTTPS server
verifications.
    #ssl.certificate_authorities:
["/etc/pki/root/ca.pem"]

    # Certificate for SSL client authentication.
    #ssl.certificate: "/etc/pki/client/cert.pem"

    # Client Certificate Key
    #ssl.key: "/etc/pki/client/cert.key"

    # Optional passphrase for decrypting the Certificate
Key.
    # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
    #ssl.key_passphrase: ''

    # Configure cipher suites to be used for SSL
connections.
    #ssl.cipher_suites: []

    # Configure curve types for ECDHE based cipher
```

```
suites.
    #ssl.curve_types: []

  #---------------------------- APM Server - ILM Index
Lifecycle Management ----------------------------

  #ilm:
    # Supported values are `auto`, `true` and `false`.
    # `true`: Make use of Elasticsearch's Index
Lifecycle Management (ILM) for APM indices. If no
Elasticsearch output is
    # configured or the configured instance does not
support ILM, APM Server cannot apply ILM and must create
    # unmanaged indices instead.
    # `false`: APM Server does not make use of ILM in
Elasticsearch.
    # `auto`: If an Elasticsearch output is configured
with default index and indices settings, and the
configured
    # Elasticsearch instance supports ILM, `auto` will
resolve to `true`. Otherwise `auto` will resolve to
`false`.
    # Default value is `auto`.
    #enabled: "auto"

    #setup:
      # Only disable setup if you want to set up
everything related to ILM on your own.
      # When setup is enabled, the APM Server creates:
      # - aliases and ILM policies if `apm-
server.ilm.enabled` resolves to `true`.
      # - An ILM specific template per event type. This
is required to map ILM aliases and policies to indices.
In case
      # ILM is disabled, the templates will be created
without any ILM settings.
      # Be aware that if you turn off setup, you need to
manually manage event type specific templates on your
```

```
own.
      # If you simply want to disable ILM, use the above
setting, `apm-server.ilm.enabled`, instead.
      # Defaults to true.
      #enabled: true

      # Configure whether or not existing policies and
ILM related templates should be updated. This needs to
be
      # set to true when customizing your policies.
      # Defaults to false.
      #overwrite: false

      # Set `require_policy` to `false` when policies
are set up outside of APM Server but referenced here.
      # Default value is `true`.
      #require_policy: true

      # Customized mappings will be merged with the
default setup, so you only need to configure mappings
for the
      # event types, policies, and index suffixes that
you want to customize.
      # Indices are named in this way: `apm-%
{[observer.version]}-%{[event.type]}-{index_suffix}`,
      # e.g., apm-7.9.0-span-custom*. The `index_suffix`
is optional.
      # NOTE: When configuring an `index_suffix`, ensure
that no previously set up templates conflict with the
      #       newly configured ones. If an index matches
multiple templates with the same order, the settings of
      #       the templates will override each other.
Any conflicts need to be cleaned up manually.
      # NOTE: When customizing `setup.template.name` and
`setup.template.pattern`, ensure they still match the
indices.
      #mapping:
        #- event_type: "error"
```

```
        #  policy_name: "apm-rollover-30-days"
        #  index_suffix: ""
        #- event_type: "span"
        #  policy_name: "apm-rollover-30-days"
        #  index_suffix: ""
        #- event_type: "transaction"
        #  policy_name: "apm-rollover-30-days"
        #  index_suffix: ""
        #- event_type: "metric"
        #  policy_name: "apm-rollover-30-days"
        #  index_suffix: ""

      # Configured policies are added to pre-defined
default policies.
      # If a policy with the same name as a default
policy is configured, the configured policy overwrites
the default policy.
      #policies:
        #- name: "apm-rollover-30-days"
          #policy:
            #phases:
              #hot:
                #actions:
                  #rollover:
                    #max_size: "50gb"
                    #max_age: "30d"
                  #set_priority:
                    #priority: 100
              #warm:
                #min_age: "30d"
                #actions:
                  #set_priority:
                    #priority: 50
                  #readonly: {}



  #---------------------------- APM Server -
```

```
Experimental Jaeger integration ------------------------
----

  # When enabling Jaeger integration, APM Server acts as
Jaeger collector. It supports jaeger.thrift over HTTP
  # and gRPC. This is an experimental feature, use with
care.
  #
  # WARNING: This configuration is deprecated, and will
be removed in the 8.0 release.
  #
  # Jaeger gRPC is now served on the same port as
Elastic APM agents, defined by the
  # "apm-server.host" configuration; it is implicitly
enabled, and an agent tag called
  # "elastic-apm-auth" is required when auth is enabled.
  #jaeger:
    #grpc:
      # Set to true to enable the Jaeger gRPC collector
service.
      #enabled: false

      # Defines the gRPC host and port the server is
listening on.
      # Defaults to the standard Jaeger gRPC collector
port 14250.
      #host: "0.0.0.0:14250"

      # Set to the name of a process tag to use for
authorizing
      # Jaeger agents.
      #
      # The tag value should have the same format as an
HTTP
      # Authorization header, i.e. "Bearer
<secret_token>" or
      # "ApiKey <base64(id:key)>".
      #
```

```
      # By default (if the auth_tag value is empty),
authorization
      # does not apply to Jaeger agents.
      #auth_tag: ""

    #http:
      # Set to true to enable the Jaeger HTTP collector
endpoint.
      #enabled: false

      # Defines the HTTP host and port the server is
listening on.
      # Defaults to the standard Jaeger HTTP collector
port 14268.
      #host: "0.0.0.0:14268"

#================================== General
==================================

# Data is buffered in a memory queue before it is
published to the configured output.
# The memory queue will present all available events (up
to the outputs
# bulk_max_size) to the output, the moment the output is
ready to serve
# another batch of events.
#queue:
  # Queue type by name (default 'mem').
  #mem:
    # Max number of events the queue can buffer.
    #events: 4096

    # Hints the minimum number of events stored in the
queue,
    # before providing a batch of events to the outputs.
    # The default value is set to 2048.
    # A value of 0 ensures events are immediately
available
```

```
    # to be sent to the outputs.
    #flush.min_events: 2048

    # Maximum duration after which events are available
to the outputs,
    # if the number of events stored in the queue is <
`flush.min_events`.
    #flush.timeout: 1s

# Sets the maximum number of CPUs that can be executing
simultaneously. The
# default is the number of logical CPUs available in the
system.
#max_procs:

#================================== Template
==================================

# A template is used to set the mapping in
Elasticsearch.
# By default template loading is enabled and the
template is loaded.
# These settings can be adjusted to load your own
template or overwrite existing ones.

# Set to false to disable template loading.
#setup.template.enabled: true

# Template name. By default the template name is "apm-%
{[observer.version]}"
# The template name and pattern has to be set in case
the elasticsearch index pattern is modified.
#setup.template.name: "apm-%{[observer.version]}"

# Template pattern. By default the template pattern is
"apm-%{[observer.version]}-*" to apply to the default
index settings.
# The first part is the version of apm-server and then -
```

```
* is used to match all daily indices.
# The template name and pattern has to be set in case
the elasticsearch index pattern is modified.
#setup.template.pattern: "apm-%{[observer.version]}-*"

# Path to fields.yml file to generate the template.
#setup.template.fields: "${path.config}/fields.yml"

# Overwrite existing template.
#setup.template.overwrite: false

# Elasticsearch template settings.
#setup.template.settings:

  # A dictionary of settings to place into the
settings.index dictionary
  # of the Elasticsearch template. For more details,
please check
  #
https://www.elastic.co/guide/en/elasticsearch/reference/
current/mapping.html
  #index:
    #number_of_shards: 1
    #codec: best_compression
    #number_of_routing_shards: 30
    #mapping.total_fields.limit: 2000

#============================== Elastic Cloud
==============================

# These settings simplify using APM Server with the
Elastic Cloud (https://cloud.elastic.co/).

# The cloud.id setting overwrites the
`output.elasticsearch.hosts` option.
# You can find the `cloud.id` in the Elastic Cloud web
UI.
#cloud.id:
```

```
# The cloud.auth setting overwrites the
`output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format
is `<user>:<pass>`.
#cloud.auth:

#================================== Outputs
==================================

# Configure the output to use when sending the data
collected by apm-server.

#-------------------------- Elasticsearch output -------
------------------
output.elasticsearch:
  # Array of hosts to connect to.
  # Scheme and port can be left out and will be set to
the default (`http` and `9200`).
  # In case you specify and additional path, the scheme
is required: `http://localhost:9200/path`.
  # IPv6 addresses should always be defined as:
`https://[2001:db8::1]:9200`.
  hosts: ["elasticsearch:9200"]

  # Boolean flag to enable or disable the output module.
  #enabled: true

  # Set gzip compression level.
  #compression_level: 0

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or
username/password.
  #api_key: "id:api_key"
  #username: "elastic"
```

```
  #password: "changeme"

  # Dictionary of HTTP parameters to pass within the url
with index operations.
  #parameters:
    #param1: value1
    #param2: value2

  # Number of workers per Elasticsearch host.
  #worker: 1

  # By using the configuration below, APM documents are
stored to separate indices,
  # depending on their `processor.event`:
  # - error
  # - transaction
  # - span
  # - sourcemap
  #
  # The indices are all prefixed with `apm-%
{[observer.version]}`.
  # To allow managing indices based on their age, all
indices (except for sourcemaps)
  # end with the information of the day they got
indexed.
  # e.g. "apm-7.3.0-transaction-2019.07.20"
  #
  # Be aware that you can only specify one Elasticsearch
template.
  # If you modify the index patterns you must also
update these configurations accordingly,
  # as they need to be aligned:
  # * `setup.template.name`
  # * `setup.template.pattern`
  #index: "apm-%{[observer.version]}-%{+yyyy.MM.dd}"
  #indices:
  #  - index: "apm-%{[observer.version]}-sourcemap"
  #    when.contains:
```

```
  #      processor.event: "sourcemap"
  #
  #  - index: "apm-%{[observer.version]}-error-%
{+yyyy.MM.dd}"
  #    when.contains:
  #      processor.event: "error"
  #
  #  - index: "apm-%{[observer.version]}-transaction-%
{+yyyy.MM.dd}"
  #    when.contains:
  #      processor.event: "transaction"
  #
  #  - index: "apm-%{[observer.version]}-span-%
{+yyyy.MM.dd}"
  #    when.contains:
  #      processor.event: "span"
  #
  #  - index: "apm-%{[observer.version]}-metric-%
{+yyyy.MM.dd}"
  #    when.contains:
  #      processor.event: "metric"
  #
  #  - index: "apm-%{[observer.version]}-onboarding-%
{+yyyy.MM.dd}"
  #    when.contains:
  #      processor.event: "onboarding"

  # A pipeline is a definition of processors applied to
documents when ingesting them to Elasticsearch.
  # APM Server comes with a default pipeline definition,
located at `ingest/pipeline/definition.json`, which is
  # loaded to Elasticsearch by default (see `apm-
server.register.ingest.pipeline`).
  # APM pipeline is enabled by default. To disable it,
set `pipeline: _none`.
  #pipeline: "apm"

  # Optional HTTP Path.
```

```
  #path: "/elasticsearch"

  # Custom HTTP headers to add to each request.
  #headers:
  #  X-My-Header: Contents of the header

  # Proxy server url.
  #proxy_url: http://proxy:3128

  # The number of times a particular Elasticsearch index
operation is attempted. If
  # the indexing operation doesn't succeed after this
many retries, the events are
  # dropped. The default is 3.
  #max_retries: 3

  # The maximum number of events to bulk in a single
Elasticsearch bulk API index request.
  # The default is 50.
  #bulk_max_size: 50

  # The number of seconds to wait before trying to
reconnect to Elasticsearch
  # after a network error. After waiting backoff.init
seconds, apm-server
  # tries to reconnect. If the attempt fails, the
backoff timer is increased
  # exponentially up to backoff.max. After a successful
connection, the backoff
  # timer is reset. The default is 1s.
  #backoff.init: 1s

  # The maximum number of seconds to wait before
attempting to connect to
  # Elasticsearch after a network error. The default is
60s.
  #backoff.max: 60s
```

```
  # Configure http request timeout before failing an
request to Elasticsearch.
  #timeout: 90

  # Enable custom SSL settings. Set to false to ignore
custom SSL settings for secure communication.
  #ssl.enabled: true

  # Optional SSL configuration options. SSL is off by
default, change the `protocol` option if you want to
enable `https`.
  #
  # Control the verification of Elasticsearch
certificates. Valid values are:
  # * full, which verifies that the provided certificate
is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
  # * strict, which verifies that the provided
certificate is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
If the Subject Alternative
  # Name is empty, it returns an error.
  # * certificate, which verifies that the provided
certificate is signed by a
  # trusted authority (CA), but does not perform any
hostname verification.
  #  * none, which performs no verification of the
server's certificate. This
  # mode disables many of the security benefits of
SSL/TLS and should only be used
  # after very careful consideration. It is primarily
intended as a temporary
  # diagnostic mechanism when attempting to resolve TLS
errors; its use in
```

```
  # production environments is strongly discouraged.
  #ssl.verification_mode: full

  # List of supported/valid TLS versions. By default all
TLS versions 1.0 up to
  # 1.2 are enabled.
  #ssl.supported_protocols: [TLSv1.0, TLSv1.1, TLSv1.2]

  # List of root certificates for HTTPS server
verifications.
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication.
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

  # Optional passphrase for decrypting the Certificate
Key.
  # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
  #ssl.key_passphrase: ''

  # Configure cipher suites to be used for SSL
connections.
  #ssl.cipher_suites: []

  # Configure curve types for ECDHE based cipher suites.
  #ssl.curve_types: []

  # Configure what types of renegotiation are supported.
Valid options are
  # never, once, and freely. Default is never.
  #ssl.renegotiation: never

  # Enable Kerberos support. Kerberos is automatically
enabled if any Kerberos setting is set.
```

```
  #kerberos.enabled: true

  # Authentication type to use with Kerberos. Available
options: keytab, password.
  #kerberos.auth_type: password

  # Path to the keytab file. It is used when auth_type
is set to keytab.
  #kerberos.keytab: /etc/elastic.keytab

  # Path to the Kerberos configuration.
  #kerberos.config_path: /etc/krb5.conf

  # Name of the Kerberos user.
  #kerberos.username: elastic

  # Password of the Kerberos user. It is used when
auth_type is set to password.
  #kerberos.password: changeme

  # Kerberos realm.
  #kerberos.realm: ELASTIC



#----------------------------- Console output ----------
-------------------
#output.console:
  # Boolean flag to enable or disable the output module.
  #enabled: false

  # Configure JSON encoding.
  #codec.json:
    # Pretty-print JSON event.
    #pretty: false

    # Configure escaping HTML symbols in strings.
    #escape_html: false
```

```
#---------------------------- Logstash output ----------
------------------
#output.logstash:
  # Boolean flag to enable or disable the output module.
  #enabled: false

  # The Logstash hosts.
  #hosts: ["localhost:5044"]

  # Number of workers per Logstash host.
  #worker: 1

  # Set gzip compression level.
  #compression_level: 3

  # Configure escaping html symbols in strings.
  #escape_html: true

  # Optional maximum time to live for a connection to
Logstash, after which the
  # connection will be re-established.  A value of `0s`
(the default) will
  # disable this feature.
  #
  # Not yet supported for async connections (i.e. with
the "pipelining" option set).
  #ttl: 30s

  # Optional load balance the events between the
Logstash hosts. Default is false.
  #loadbalance: false

  # Number of batches to be sent asynchronously to
Logstash while processing
  # new batches.
  #pipelining: 2

  # If enabled only a subset of events in a batch of
```

```
events is transferred per
  # group.  The number of events to be sent increases up
to `bulk_max_size`
  # if no error is encountered.
  #slow_start: false

  # The number of seconds to wait before trying to
reconnect to Logstash
  # after a network error. After waiting backoff.init
seconds, apm-server
  # tries to reconnect. If the attempt fails, the
backoff timer is increased
  # exponentially up to backoff.max. After a successful
connection, the backoff
  # timer is reset. The default is 1s.
  #backoff.init: 1s

  # The maximum number of seconds to wait before
attempting to connect to
  # Logstash after a network error. The default is 60s.
  #backoff.max: 60s

  # Optional index name. The default index name is set
to apm
  # in all lowercase.
  #index: 'apm'

  # SOCKS5 proxy server URL
  #proxy_url: socks5://user:password@socks5-server:2233

  # Resolve names locally when using a proxy server.
Defaults to false.
  #proxy_use_local_resolver: false

  # Enable SSL support. SSL is automatically enabled if
any SSL setting is set.
  #ssl.enabled: false
```

```
  # Optional SSL configuration options. SSL is off by
default.
  #
  # Control the verification of Logstash certificates.
Valid values are:
  # * full, which verifies that the provided certificate
is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
  # * strict, which verifies that the provided
certificate is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
If the Subject Alternative
  # Name is empty, it returns an error.
  # * certificate, which verifies that the provided
certificate is signed by a
  # trusted authority (CA), but does not perform any
hostname verification.
  #  * none, which performs no verification of the
server's certificate. This
  # mode disables many of the security benefits of
SSL/TLS and should only be used
  # after very careful consideration. It is primarily
intended as a temporary
  # diagnostic mechanism when attempting to resolve TLS
errors; its use in
  # production environments is strongly discouraged.
  #ssl.verification_mode: full

  # List of supported/valid TLS versions. By default all
TLS versions 1.0 up to
  # 1.2 are enabled.
  #ssl.supported_protocols: [TLSv1.0, TLSv1.1, TLSv1.2]

  # List of root certificates for HTTPS server
```

```
verifications.
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication.
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

  # Optional passphrase for decrypting the Certificate
Key.
  # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
  #ssl.key_passphrase: ''

  # Configure cipher suites to be used for SSL
connections.
  #ssl.cipher_suites: []

  # Configure curve types for ECDHE based cipher suites.
  #ssl.curve_types: []

  # Configure what types of renegotiation are supported.
Valid options are
  # never, once, and freely. Default is never.
  #ssl.renegotiation: never

#------------------------------ Kafka output ----------
-------------------
#output.kafka:
  # Boolean flag to enable or disable the output module.
  #enabled: false

  # The list of Kafka broker addresses from where to
fetch the cluster metadata.
  # The cluster metadata contain the actual Kafka
brokers events are published
  # to.
```

```
  #hosts: ["localhost:9092"]

  # The Kafka topic used for produced events. The
setting can be a format string
  # using any event field. To set the topic from
document type use `%{[type]}`.
  #topic: beats

  # The Kafka event key setting. Use format string to
create unique event key.
  # By default no event key will be generated.
  #key: ''

  # The Kafka event partitioning strategy. Default
hashing strategy is `hash`
  # using the `output.kafka.key` setting or randomly
distributes events if
  # `output.kafka.key` is not configured.
  #partition.hash:
    # If enabled, events will only be published to
partitions with reachable
    # leaders. Default is false.
    #reachable_only: false

    # Configure alternative event field names used to
compute the hash value.
    # If empty `output.kafka.key` setting will be used.
    # Default value is empty list.
    #hash: []

  # Authentication details. Password is required if
username is set.
  #username: ''
  #password: ''

  # Kafka version libbeat is assumed to run against.
Defaults to the "1.0.0".
  #version: '1.0.0'
```

```
  # Configure JSON encoding.
  #codec.json:
    # Pretty print json event
    #pretty: false

    # Configure escaping html symbols in strings.
    #escape_html: true

  # Metadata update configuration. Metadata do contain
leader information
  # deciding which broker to use when publishing.
  #metadata:
    # Max metadata request retry attempts when cluster
is in middle of leader
    # election. Defaults to 3 retries.
    #retry.max: 3

    # Waiting time between retries during leader
elections. Default is 250ms.
    #retry.backoff: 250ms

    # Refresh metadata interval. Defaults to every 10
minutes.
    #refresh_frequency: 10m

  # The number of concurrent load-balanced Kafka output
workers.
  #worker: 1

  # The number of times to retry publishing an event
after a publishing failure.
  # After the specified number of retries, the events
are typically dropped.
  # Set max_retries to a value less than 0 to retry
  # until all events are published. The default is 3.
  #max_retries: 3
```

```
  # The maximum number of events to bulk in a single
Kafka request. The default
  # is 2048.
  #bulk_max_size: 2048

  # The number of seconds to wait for responses from the
Kafka brokers before
  # timing out. The default is 30s.
  #timeout: 30s

  # The maximum duration a broker will wait for number
of required ACKs. The
  # default is 10s.
  #broker_timeout: 10s

  # The number of messages buffered for each Kafka
broker. The default is 256.
  #channel_buffer_size: 256

  # The keep-alive period for an active network
connection. If 0s, keep-alives
  # are disabled. The default is 0 seconds.
  #keep_alive: 0

  # Sets the output compression codec. Must be one of
none, snappy and gzip. The
  # default is gzip.
  #compression: gzip

  # Set the compression level. Currently only gzip
provides a compression level
  # between 0 and 9. The default value is chosen by the
compression algorithm.
  #compression_level: 4

  # The maximum permitted size of JSON-encoded messages.
Bigger messages will be
  # dropped. The default value is 1000000 (bytes). This
```

```
value should be equal to
  # or less than the broker's message.max.bytes.
  #max_message_bytes: 1000000

  # The ACK reliability level required from broker. 0=no
response, 1=wait for
  # local commit, -1=wait for all replicas to commit.
The default is 1.  Note:
  # If set to 0, no ACKs are returned by Kafka. Messages
might be lost silently
  # on error.
  #required_acks: 1

  # The configurable ClientID used for logging,
debugging, and auditing
  # purposes.  The default is "beats".
  #client_id: beats

  # Enable SSL support. SSL is automatically enabled if
any SSL setting is set.
  #ssl.enabled: false

  # Optional SSL configuration options. SSL is off by
default.
  #
  # Control the verification of Kafka certificates.
Valid values are:
  # * full, which verifies that the provided certificate
is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
  # * strict, which verifies that the provided
certificate is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
If the Subject Alternative
```

```
  # Name is empty, it returns an error.
  # * certificate, which verifies that the provided
certificate is signed by a
  # trusted authority (CA), but does not perform any
hostname verification.
  #  * none, which performs no verification of the
server's certificate. This
  # mode disables many of the security benefits of
SSL/TLS and should only be used
  # after very careful consideration. It is primarily
intended as a temporary
  # diagnostic mechanism when attempting to resolve TLS
errors; its use in
  # production environments is strongly discouraged.
  #ssl.verification_mode: full

  # List of supported/valid TLS versions. By default all
TLS versions 1.0 up to
  # 1.2 are enabled.
  #ssl.supported_protocols: [TLSv1.0, TLSv1.1, TLSv1.2]

  # List of root certificates for HTTPS server
verifications.
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication.
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

  # Optional passphrase for decrypting the Certificate
Key.
  # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
  #ssl.key_passphrase: ''

  # Configure cipher suites to be used for SSL
```

```
connections.
  #ssl.cipher_suites: []

  # Configure curve types for ECDHE based cipher suites.
  #ssl.curve_types: []

  # Configure what types of renegotiation are supported.
Valid options are
  # never, once, and freely. Default is never.
  #ssl.renegotiation: never

  # Authentication type to use with Kerberos. Available
options: keytab, password.
  #kerberos.auth_type: password

  # Path to the keytab file. It is used when auth_type
is set to keytab.
  #kerberos.keytab: /etc/krb5kdc/kafka.keytab

  # Path to the Kerberos configuration.
  #kerberos.config_path: /etc/path/config

  # The service principal name.
  #kerberos.service_name: HTTP/my-service@realm

  # Name of the Kerberos user. It is used when auth_type
is set to password.
  #kerberos.username: elastic

  # Password of the Kerberos user. It is used when
auth_type is set to password.
  #kerberos.password: changeme

  # Kerberos realm.
  #kerberos.realm: ELASTIC

#============================== Instrumentation
===============================
```

```
# Instrumentation support for the server's HTTP
endpoints and event publisher.
#instrumentation:

  # Set to true to enable instrumentation of the APM
Server itself.
  #enabled: false

  # Environment in which the APM Server is running on
(eg: staging, production, etc.)
  #environment: ""

  # Hosts to report instrumentation results to.
  # For reporting to itself, leave this field commented
  #hosts:
  #  - http://remote-apm-server:8200

  # API Key for the remote APM Server(s).
  # If api_key is set then secret_token will be ignored.
  #api_key:

  # Secret token for the remote APM Server(s).
  #secret_token:

  # Enable profiling of the server, recording profile
samples as events.
  #
  # This feature is experimental.
  #profiling:
    #cpu:
      # Set to true to enable CPU profiling.
      #enabled: false
      #interval: 60s
      #duration: 10s
    #heap:
      # Set to true to enable heap profiling.
      #enabled: false
```

```
      #interval: 60s

#================================== Paths
==================================

# The home path for the apm-server installation. This is
the default base path
# for all other path settings and for miscellaneous
files that come with the
# distribution.
# If not set by a CLI flag or in the configuration file,
the default for the
# home path is the location of the binary.
#path.home:

# The configuration path for the apm-server
installation. This is the default
# base path for configuration files, including the main
YAML configuration file
# and the Elasticsearch template file. If not set by a
CLI flag or in the
# configuration file, the default for the configuration
path is the home path.
#path.config: ${path.home}

# The data path for the apm-server installation. This is
the default base path
# for all the files in which apm-server needs to store
its data. If not set by a
# CLI flag or in the configuration file, the default for
the data path is a data
# subdirectory inside the home path.
#path.data: ${path.home}/data

# The logs path for an apm-server installation. If not
set by a CLI flag or in the
# configuration file, the default is a logs subdirectory
inside the home path.
```

```
#path.logs: ${path.home}/logs

#============================== Logging
==============================

# There are three options for the log output: syslog,
file, and stderr.
# Windows systems default to file output. All other
systems default to syslog.

# Sets the minimum log level. The default log level is
info.
# Available log levels are: error, warning, info, or
debug.
#logging.level: info

# Enable debug output for selected components. To enable
all selectors use ["*"].
# Other available selectors are "beat", "publish", or
"service".
# Multiple selectors can be chained.
#logging.selectors: [ ]

# Send all logging output to syslog. The default is
false.
#logging.to_syslog: true

# If enabled, apm-server periodically logs its internal
metrics that have changed
# in the last period. For each metric that changed, the
delta from the value at
# the beginning of the period is logged. Also, the total
values for
# all non-zero internal metrics are logged on shutdown.
The default is false.
#logging.metrics.enabled: false

# The period after which to log the internal metrics.
```

```
The default is 30s.
#logging.metrics.period: 30s

# Logging to rotating files. When true, writes all
logging output to files.
# The log files are automatically rotated when the log
file size limit is reached.
#logging.to_files: true
#logging.files:
  # Configure the path where the logs are written. The
default is the logs directory
  # under the home path (the binary location).
  #path: /var/log/apm-server

  # The name of the files where the logs are written to.
  #name: apm-server

  # Configure log file size limit. If limit is reached,
log file will be
  # automatically rotated.
  #rotateeverybytes: 10485760 # = 10MB

  # Number of rotated log files to keep. Oldest files
will be deleted first.
  #keepfiles: 7

  # The permissions mask to apply when rotating log
files. The default value is 0600.
  # Must be a valid Unix-style file permissions mask
expressed in octal notation.
  #permissions: 0600

  # Enable log file rotation on time intervals in
addition to size-based rotation.
  # Intervals must be at least 1s. Values of 1m, 1h,
24h, 7*24h, 30*24h, and 365*24h
  # are boundary-aligned with minutes, hours, days,
weeks, months, and years as
```

```
  # reported by the local system clock. All other
intervals are calculated from the
  # Unix epoch. Defaults to disabled.
  #interval: 0

# Set to true to log messages in json format.
#logging.json: true

# Set to true, to log messages with minimal required
Elastic Common Schema (ECS)
# information. Recommended to use in combination with
`logging.json=true`.
#logging.ecs: true

#============================== HTTP Endpoint
==============================

# apm-server can expose internal metrics through a HTTP
endpoint. For security
# reasons the endpoint is disabled by default. This
feature is currently experimental.
# Stats can be access through
http://localhost:5066/stats. For pretty JSON output
# append ?pretty to the URL.

# Defines if the HTTP endpoint is enabled.
#http.enabled: false

# The HTTP endpoint will bind to this hostname or IP
address. It is recommended to use only localhost.
#http.host: localhost

# Port on which the HTTP endpoint will bind. Default is
5066.
#http.port: 5066

#============================== X-pack Monitoring
==============================
```

```
# APM server can export internal metrics to a central
Elasticsearch monitoring
# cluster.  This requires x-pack monitoring to be
enabled in Elasticsearch.  The
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
#monitoring.enabled: false

# Most settings from the Elasticsearch output are
accepted here as well.
# Note that these settings should be configured to point
to your Elasticsearch *monitoring* cluster.
# Any setting that is not set is automatically inherited
from the Elasticsearch
# output configuration. This means that if you have the
Elasticsearch output configured,
# you can simply uncomment the following line.
#monitoring.elasticsearch:

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or
username/password.
  #api_key: "id:api_key"
  #username: "elastic"
  #password: "changeme"

  # Array of hosts to connect to.
  # Scheme and port can be left out and will be set to
the default (`http` and `9200`).
  # In case you specify and additional path, the scheme
is required: `http://localhost:9200/path`.
  # IPv6 addresses should always be defined as:
`https://[2001:db8::1]:9200`.
  #hosts: ["localhost:9200"]
```

```
  # Set gzip compression level.
  #compression_level: 0

  # Dictionary of HTTP parameters to pass within the URL
with index operations.
  #parameters:
    #param1: value1
    #param2: value2

  # Custom HTTP headers to add to each request.
  #headers:
  #  X-My-Header: Contents of the header

  # Proxy server url.
  #proxy_url: http://proxy:3128

  # The number of times a particular Elasticsearch index
operation is attempted. If
  # the indexing operation doesn't succeed after this
many retries, the events are
  # dropped. The default is 3.
  #max_retries: 3

  # The maximum number of events to bulk in a single
Elasticsearch bulk API index request.
  # The default is 50.
  #bulk_max_size: 50

  # The number of seconds to wait before trying to
reconnect to Elasticsearch
  # after a network error. After waiting backoff.init
seconds, apm-server
  # tries to reconnect. If the attempt fails, the
backoff timer is increased
  # exponentially up to backoff.max. After a successful
connection, the backoff
  # timer is reset. The default is 1s.
```

```
  #backoff.init: 1s

  # The maximum number of seconds to wait before
attempting to connect to
  # Elasticsearch after a network error. The default is
60s.
  #backoff.max: 60s

  # Configure HTTP request timeout before failing an
request to Elasticsearch.
  #timeout: 90

  # Enable custom SSL settings. Set to false to ignore
custom SSL settings for secure communication.
  #ssl.enabled: true

  # Optional SSL configuration options. SSL is off by
default, change the `protocol` option if you want to
enable `https`.
  #
  # Control the verification of Elasticsearch
certificates. Valid values are:
  # * full, which verifies that the provided certificate
is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
  # * strict, which verifies that the provided
certificate is signed by a trusted
  # authority (CA) and also verifies that the server's
hostname (or IP address)
  # matches the names identified within the certificate.
If the Subject Alternative
  # Name is empty, it returns an error.
  # * certificate, which verifies that the provided
certificate is signed by a
  # trusted authority (CA), but does not perform any
hostname verification.
```

```
  #  * none, which performs no verification of the
server's certificate. This
  # mode disables many of the security benefits of
SSL/TLS and should only be used
  # after very careful consideration. It is primarily
intended as a temporary
  # diagnostic mechanism when attempting to resolve TLS
errors; its use in
  # production environments is strongly discouraged.
  #ssl.verification_mode: full

  # List of supported/valid TLS versions. By default all
TLS versions 1.0 up to
  # 1.2 are enabled.
  #ssl.supported_protocols: [TLSv1.0, TLSv1.1, TLSv1.2]

  # List of root certificates for HTTPS server
verifications.
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication.
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

  # Optional passphrase for decrypting the Certificate
Key.
  # It is recommended to use the provided keystore
instead of entering the passphrase in plain text.
  #ssl.key_passphrase: ''

  # Configure cipher suites to be used for SSL
connections.
  #ssl.cipher_suites: []

  # Configure curve types for ECDHE based cipher suites.
  #ssl.curve_types: []
```

```
  # Configure what types of renegotiation are supported.
Valid options are
  # never, once, and freely. Default is never.
  #ssl.renegotiation: never

  # Enable Kerberos support. Kerberos is automatically
enabled if any Kerberos setting is set.
  #kerberos.enabled: true

  # Authentication type to use with Kerberos. Available
options: keytab, password.
  #kerberos.auth_type: password

  # Path to the keytab file. It is used when auth_type
is set to keytab.
  #kerberos.keytab: /etc/elastic.keytab

  # Path to the Kerberos configuration.
  #kerberos.config_path: /etc/krb5.conf

  # Name of the Kerberos user.
  #kerberos.username: elastic

  # Password of the Kerberos user. It is used when
auth_type is set to password.
  #kerberos.password: changeme

  # Kerberos realm.
  #kerberos.realm: ELASTIC

  #metrics.period: 10s
  #state.period: 1m
```

Adicione o serviço ao docker compose

```
  apm:
    image: docker.elastic.co/apm/apm-server-oss:7.13.0
    container_name: apm
    volumes:
    - ./apm-server.yml:/usr/share/apm-server/apm-
server.yml
    ports:
      - "8200:8200"
    restart: on-failure
    networks:
      - elastic
```

Instale a lib

```
pip install elastic-apm
```

Adicione aos apps instalados

```
'elasticapm.contrib.django',
```

Crie a constante

```
ELASTIC_APM = {
  'SERVICE_NAME': 'djangoproject',
  'DEBUG': True,
  'SERVER_URL': 'http://localhost:8200',
  'ENVIRONMENT': 'production',
}
```

Adicione o Middleware

```
'elasticapm.contrib.django.middleware.TracingMiddleware'
,
```

Adicione o context_processors de templates

```
'elasticapm.contrib.django.context_processors.rum_tracing',
```