Don’t want to use https and user:password

Kumar_Saurabh_Srivas Kumar Saurabh Srivastava **Oct 2019**

Hi
I am deploying ECK on GKE in a private Kubernetes cluster. That cluster has the only service which will talk to Elasticsearch. So I don't need to have any https or user:password authentication. All I want is a simple clusterIP service which can be directly accessed by the service within the kubernetes cluster.
Please let me know how to do that.

Thibault_Richard Thibault Elastic Team Member **Oct 2019**

Hello,

TLS can be disabled as explained here: **https://www.elastic.co/guide/en/cloud-on-k8s/0.9/k8s-accessing-elastic-services.html#k8s-disable-tls** .

```
spec:
  http:
    tls:
      selfSignedCertificate:
        disabled: true
```

Basic authentication can be bypassed by enabling anonymous access: **https://www.elastic.co/guide/en/elastic-stack-overview/7.4/anonymous-access.html** .

```
spec:
  nodes:
  - nodeCount: 1
    config:
      xpack.security.authc:
          anonymous:
            username: anonymous
            roles: superuser
            authz_exception: false
```

Even in a private cluster, we don't recommend to turn off these security layers.
Obviously, it is up to you to make the decision 🙂

Skip to main content

Recommended for you

- **Deploy a Kibana instance | Elastic Cloud on Kubernetes**
- **Introduction | Elasticsearch .NET Clients**
- **Installation | Elasticsearch .NET Clients**
- **Install Elasticsearch from archive on Linux or MacOS | Elasticsearch**
- **Install Elasticsearch with Debian Package | Elasticsearch**

megatrond Oct 2019

How does this apply to Kibana? I've tried applying the settings above to my elasticsearch deployment, as well as the following on my Kibana deployment:

```
config:
  xpack.security.enabled: false
```

This causes the Kibana pods to get stuck on startup "Optimizing and caching bundled". Re-enabling xpack security causes them to be able to start up again.

Kumar_Saurabh_Srivas Kumar Saurabh Srivastava Oct 2019

> Thibault_Richard:
>
> xpack.security.authc: anonymous: username: anonymous roles: superuser authz_exception: false

Tried disabling TLS. It says:

```
Error from server (TLS cannot be disabled for Elasticsearch currently): error
```

Thibault_Richard Thibault Elastic Team Member Oct 2019

> megatrond:
>
> How does this apply to Kibana? I've tried applying the settings above to my elasticsearch deployment, as well as the following on my Kibana deployment:
>
> ```
> config:
>   xpack.security.enabled: false
> ```
>
> This causes the Kibana pods to get stuck on startup "Optimizing and caching bundled". Re-enabling xpack security causes them to be able to start up again.

Disabling xpack on Kibana forces replaying the optimization process responsible for generating JS bundles for all of the installed plugins.

This optimization process is very CPU/memory intensive and can take up to several minutes to complete depending on the underlying hardware.

I tested on my side and with the default Kibana resources, the Kibana pod is OOMKilled.

You can give more memory to your Kibana instance(s) to speed up this process. This is documented here: **https://www.elastic.co/guide/en/cloud-on-k8s/master/k8s-managing-compute-resources.html#k8s-compute-resources-kibana-and-apm** .

By increasing the memory limit to 4Gi, it took 83s for me.

```
apiVersion: kibana.k8s.elastic.co/v1beta1
kind: Kibana
  name: quickstart
```

Skip to main content

```
spec:
  version: 7.4.0
  count: 1
  elasticsearchRef:
    name: quickstart
  config:
    xpack.security.enabled: false
  podTemplate:
    spec:
      containers:
      - name: kibana
        resources:
          limits:
            memory: 4Gi
```

KapitanPlaneta Łukasz Konieczny **Jan 2020**

Provided solution does not work.
Kibana pod does not get past thru readiness probes:

> ```
> Events:
>   Type     Reason     Age                 From
>   ----     ------     ----                ----
>   Normal   Scheduled  4m37s               default-scheduler
>   Normal   Pulled     4m36s               kubelet, ip-10-51-189-46.ec2.i
>   Normal   Created    4m36s               kubelet, ip-10-51-189-46.ec2.i
>   Normal   Started    4m36s               kubelet, ip-10-51-189-46.ec2.i
>   Warning  Unhealthy  62s (x4 over 92s)   kubelet, ip-10-51-189-46.ec2.i
>   Warning  Unhealthy  55s (x17 over 4m14s)  kubelet, ip-10-51-189-46.ec2.i
>   Warning  Unhealthy  47s                 kubelet, ip-10-51-189-46.ec2.i
> ```

---

huseinzol05 HUSEIN ZOLKEPLI **Feb 2020**

Overwrite readiness,

```
apiVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
  name: cluster
spec:
  version: 7.6.0
  http:
    tls:
      selfSignedCertificate:
        disabled: true
  config:
    xpack.security.enabled: false
  podTemplate:
    spec:
      containers:
        - name: kibana
          resources:
```

Skip to main content

```
              memory: 4Gi
          readinessProbe:
            failureThreshold: 3
            httpGet:
              path: /
              port: 5601
              scheme: HTTP
            initialDelaySeconds: 10
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 5
  count: 1
  elasticsearchRef:
    name: cluster
```

strowi Roman **Aug 2020**

Hi,

can somebody confirm that this should stil work with the latest operator?
I have the following, which results in the readiness probe still returning a 401

```
---
apiVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
  name: test
spec:
  version: 7.9.0
  count: 1
  elasticsearchRef:
    name: ci

  config:
    xpack.security.enabled: false
  http:
    tls:
      selfSignedCertificate:
        disabled: true
  podTemplate:
    metadata:
      labels:
        team: "sys"
      annotations:
        app.gitlab.com/env: ci
        app.gitlab.com/app: sys-logging-elk
    spec:
      containers:
        - name: kibana
          resources:
            limits:
              memory: 4Gi
          readinessProbe:
            failureThreshold: 3
            httpGet:
              path: /
                    5601
              scheme: HTTP
```

Skip to main content

```
            initialDelaySeconds: 10
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 5
```

regards,
strowi

charith-elastic Charith Ellawala Elastic Team Member **Aug 2020**

Accessing `/` results in a request to Elasticsearch to locate the default space and requires authentication to succeed. This is why the default readiness probe uses `/login`. You can try using the `/api/features` path but the amount of data it returns could exceed the maximum payload size of the Kubernetes readiness probe and that could result in random failures down the line. You could try switching to a TCP probe but that wouldn't necessarily give you full confidence that Kibana is actually handling requests correctly.

strowi Roman **Aug 2020**

Thx for the reply, but even disabling the readiness-probe, the pod starts, but i get a login-prompt from kibana returning:

```
{"statusCode":401,"error":"Unauthorized","message":"[security_exception] miss
```

Nexonus Moritz **Oct 2020**

Any news on this? I've got the same problem.

weydersantos **Jan 13**

After turn off X-Pack Security my Kibana stucks

```
2021-01-13T15:27:39.309011487Z {"type":"response","@timestamp":"2021-01-13T15:
```

michael.morello Elastic Team Member **Jan 13**

It is not possible to disable security, you can consider **using anonymous** access if don't need authentication.

CLOSED ON JAN 13

I'm closing this topic, to sum up:

- It is not possible to disable security.
- You can consider **using anonymous** access if you don't need authentication.
- **TLS can be disabled on the HTTP layer**

Feel free to open a new topic if you have any trouble with these settings.