

## TP15- Grupo Nro2

### Crear Stack ELK en cluster minikube [↗](#)

#### Paso 1 [↗](#)

Creamos los YAML para cada uno de los servicios

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: kibana
5  spec:
6    selector:
7      app: kibana
8    ports:
9      - protocol: TCP
10      port: 5601
11      targetPort: 5601
12  ---
13  apiVersion: apps/v1
14  kind: Deployment
15  metadata:
16    name: kibana
17  spec:
18    replicas: 1
19    selector:
20      matchLabels:
21        app: kibana
22    template:
23      metadata:
24        labels:
25          app: kibana
26      spec:
27        containers:
28          - name: kibana
29            image: docker.elastic.co/kibana/kibana:7.10.0
30            ports:
31              - containerPort: 5601
32            resources:
33              limits:
34                cpu: "500m"
35                memory: "1Gi"
36              requests:
37                cpu: "500m"
38                memory: "512Mi"
39            env:
40              - name: ELASTICSEARCH_URL
41                value: "http://elasticsearch:9200"
42
```

ElasticSearch [↗](#)

1 DS-35  
2 DS-5  
3 DS-4  
4 DS-14  
5 DS-15  
6 DS-24  
7 DS-25  
8 DS-34  
9  
10 DS-35  
11  
12 DS-38  
13  
14  
15 Crear Stack ELK en cluster minikube  
16  
17 Adjuntar  
18  
19 Vincular incidencia  
20  
21 Descripción  
22  
23 Editar descripción  
24 Archivos adjuntos (6)  
25  
26  
27  
28 logstash.yaml  
29 17 ene 2024, 07:28 pm  
30  
31 kibana.yaml  
32 17 ene 2024, 07:28 pm  
33  
34 image-20240117-223023.png  
35 17 ene 2024, 07:30 pm  
36  
37 image-20240117-223125.png  
38 17 ene 2024, 07:31 pm  
39  
40 image-20240117-223212.png  
41 17 ene 2024, 07:32 pm  
42  
43 elasticsearch.yaml  
44 17 ene 2024, 07:28 pm  
45 Actividad  
46 Mostrar:  
47  
48 Todo  
49  
50 Comentarios  
51  
52 Historial  
53  
54 Más recientes primero  
55 Imagen de perfil de franco lizarraga  
56 Añadir un comentario...  
57 Consejo de expertos: pulsa  
58 M

```
59 para comentar
60
61 lea.apagro
62 hace 7 minutos
63
64 Generamos los siguientes .yaml:
65
66
67
68 kibana.yaml
69 17 ene 2024, 07:28 pm
70
71 elasticsearch.yaml
72 17 ene 2024, 07:28 pm
73
74 logstash.yaml
75 17 ene 2024, 07:28 pm
76 Una vez generados, aplicamos los cambios con:
77
78
79
80
81 kubectl apply -f .
82 Luego verificamos que estén levantados los servicios:
83
84
85
86
87 kubectl get pods
88 Adjunto captura de la salida:
89
90
91 Dicho esto hacemos un port-forward al servicio de kibana:
92
93
94
95
96 kubectl port-forward svc/kibana 5601:5601
97
98
99 Listo! Ahora podremos acceder a Kibana desde nuestro navegador:
100
101
102
103 Editar
104
105 Eliminar
106
107
108 Finalizada
109 Listo
110
111 Acciones
112 Detalles
113 Responsable, Etiquetas, Sprint, Story point estimate, Informador
114 Creado anteayer
115 Actualizado hace 7 minutos
116 Resuelto hace 7 minutos
```

```

117 Configurar
118
119 elasticsearch.yaml
120 yaml · 860 B
121
122 apiVersion: v1
123 kind: Service
124 metadata:
125   name: elasticsearch
126 spec:
127   selector:
128     app: elasticsearch
129   ports:
130     - protocol: TCP
131       port: 9200
132       targetPort: 9200
133 ---
134 apiVersion: apps/v1
135 kind: Deployment
136 metadata:
137   name: elasticsearch
138 spec:
139   replicas: 1
140   selector:
141     matchLabels:
142       app: elasticsearch
143   template:
144     metadata:
145       labels:
146         app: elasticsearch
147     spec:
148       containers:
149         - name: elasticsearch
150           image: docker.elastic.co/elasticsearch/elasticsearch:7.10.0
151           ports:
152             - containerPort: 9200
153             - containerPort: 9300
154           resources:
155             limits:
156               cpu: "500m"
157               memory: "512Gi"
158             requests:
159               cpu: "500m"
160               memory: "512Mi"
161           env:
162             - name: discovery.type
163               value: single-node

```

## Logstash [↗](#)

```

1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: logstash-config
5 data:
6   logstash.conf: |

```

```

7   input {
8     beats {
9       port => 5044
10    }
11  }
12  output {
13    elasticsearch {
14      hosts => ["elasticsearch:9200"]
15    }
16  }
17  ---
18  apiVersion: apps/v1
19  kind: Deployment
20  metadata:
21    name: logstash
22  spec:
23    replicas: 1
24    selector:
25      matchLabels:
26        app: logstash
27    template:
28      metadata:
29        labels:
30          app: logstash
31      spec:
32        containers:
33          - name: logstash
34            image: docker.elastic.co/logstash/logstash:7.10.0
35            ports:
36              - containerPort: 5044
37            resources:
38              limits:
39                cpu: "500m"
40                memory: "1Gi"
41              requests:
42                cpu: "500m"
43                memory: "512Mi"
44            volumeMounts:
45              - name: logstash-config
46                mountPath: /usr/share/logstash/pipeline
47        volumes:
48          - name: logstash-config
49            configMap:
50              name: logstash-config

```

Una vez generados, aplicamos los cambios con:

```
1 kubectl apply -f .
```

Luego verificamos que estén levantados los servicios:

```
1 kubectl get pods
```

deberíamos ver en estado running los 3 servicios que levantamos

```

PS D:\Educacion IT\DevOps\ELK> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
elasticsearch-648889587c-s5vkr      1/1     Running   0           36s
kibana-5f56b68487-fwngk             1/1     Running   0           5m20s
logstash-67794c88b7-7lv8w          1/1     Running   0           5m20s

```

ahora hacemos un port-forward al servicio de kibana con el si. comando

```
1 kubectl port-forward svc/kibana 5601:5601
```

```

PS D:\Educacion IT\DevOps\ELK> kubectl get svc
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
elasticsearch   ClusterIP   10.99.199.119   <none>        9200/TCP   8m38s
kibana          ClusterIP   10.99.255.198   <none>        5601/TCP   8m38s
kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP    45h

```

Listo! Ahora podremos acceder a Kibana desde nuestro navegador