



과제 4 해설

🕒 Created

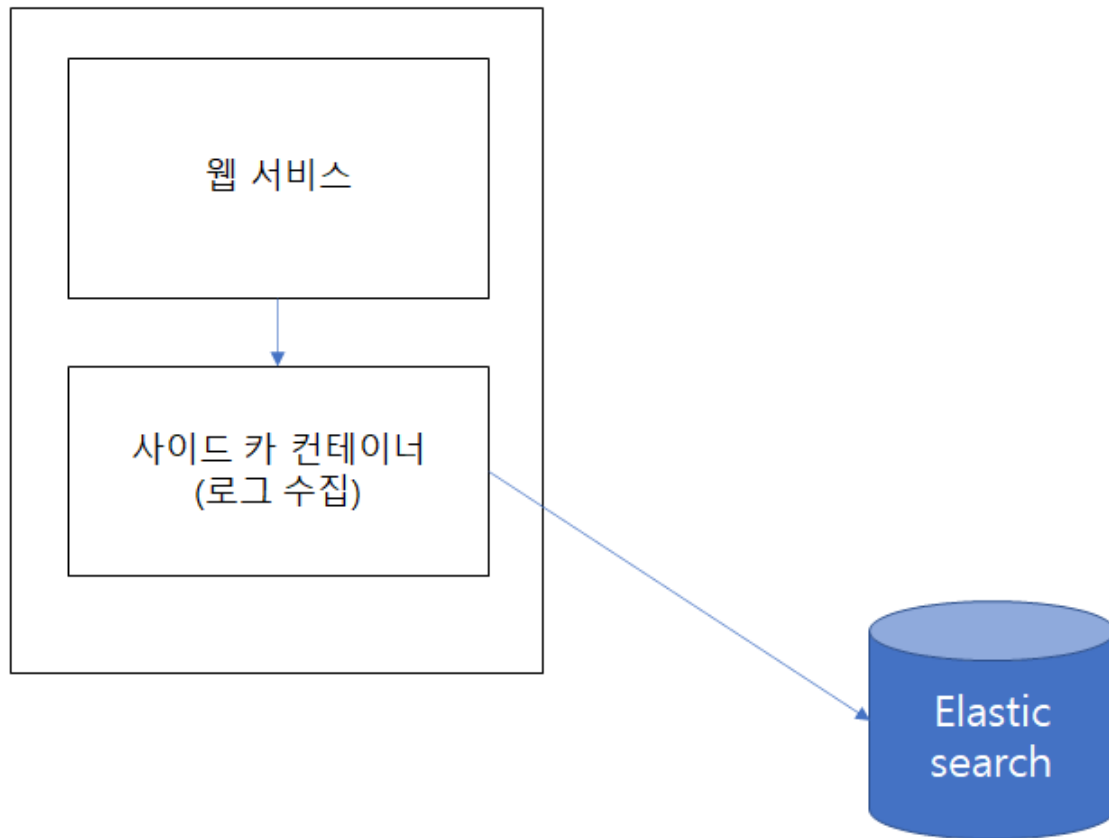
2022년 1월 3일 오후 9:07

🏷️ Tags

비어 있음

과제 수행

- 수행 조건 : 웹 로그 수집을 위한 사이드 컨테이너 및 파이프 라인 구축
- 수행 방법
 - 웹/사이드카 컨테이너 선정의 이유 및 구축 과정에 대한 가이드라인을 상세하게 작성
 - 웹/사이드카 컨테이너는 과제 수행에 적당한 것으로 자유 선택 가능
 - 웹 로그를 저장하는 엘라스틱서치는 데이터가 계속 보존될 수 있도록 해결책을 제시하고, 엘라스틱서치가 새로운 버전으로 업데이트 되어도 데이터가 보존될 수 있도록 처리하는 실습이 반드시 포함되어야 함



엘라스틱서치 및 키바나 서비스 구성

다음과 같이 리소스를 구성한다.

- elastic 네임스페이스
 - 엘라스틱서치
 - 서비스: elasticsearch-svc
 - 디플로이먼트: elasticsearch
 - 퍼시스턴스볼륨: pvc를 활용해 볼륨을 생성하고 ela-data로 연결
 - 권한을 설정하기 위한 init 컨테이너 구성
 - 키바나
 - 서비스: kibana-svc
 - 디플로이먼트: kibana

다음 명령을 사용해 클러스터에 엘라스틱서치를 구성한다.

```
cat <<EOF | kubectl apply -f -
apiVersion: v1 kind: Namespace metadata: name:
elastic --- apiVersion: v1 kind: Service metadata: labels: app: elasticsearch
name: elasticsearch-svc namespace: elastic spec: ports: - name: elasticsearch-
rest port: 9200 protocol: TCP targetPort: 9200 - name: elasticsearch-nodecom
port: 9300 protocol: TCP targetPort: 9300 selector: app: elasticsearch ---
apiVersion: apps/v1 kind: Deployment metadata: name: elasticsearch namespace:
elastic labels: app: elasticsearch spec: replicas: 1 selector: matchLabels:
app: elasticsearch template: metadata: labels: app: elasticsearch spec:
containers: - name: elasticsearch image: elastic/elasticsearch:7.14.1 env: -
name: discovery.type value: single-node ports: - containerPort: 9200 -
containerPort: 9300 volumeMounts: - name: ela-data mountPath:
/usr/share/elasticsearch/data volumes: - name: ela-data persistentVolumeClaim:
claimName: elasticsearch-pvc initContainers: - name: fix-permissions image:
busybox command: ["sh", "-c", "chown -R 1000:1000
/usr/share/elasticsearch/data"] securityContext: privileged: true
volumeMounts: - name: ela-data mountPath: /usr/share/elasticsearch/data ---
apiVersion: v1 kind: PersistentVolumeClaim metadata: namespace: elastic name:
elasticsearch-pvc spec: accessModes: - ReadWriteOnce volumeMode: Filesystem
resources: requests: storage: 10Gi --- apiVersion: apps/v1 kind: Deployment
metadata: name: kibana namespace: elastic labels: app: kibana spec: replicas:
1 selector: matchLabels: app: kibana template: metadata: labels: app: kibana
spec: containers: - name: kibana image: elastic/kibana:7.14.1 env: - name:
SERVER_NAME value: kibana.kubernetes.example.com - name: ELASTICSEARCH_HOSTS
value: http://elasticsearch-svc:9200 ports: - containerPort: 5601 ---
apiVersion: v1 kind: Service metadata: labels: app: kibana name: kibana-svc
namespace: elastic spec: ports: - nodePort: 30080 port: 80 protocol: TCP
targetPort: 5601 selector: app: kibana type: LoadBalancer EOF
```

nginx 디플로이먼트 구성

nginx는 가장 대중적으로 사용되는 웹서비스 중 하나다. 이 nginx를 사용해 주 컨테이너를 구성하고 사이드카 컨테이너를 활용해 데이터를 수집한다. 데이터 공유는 emptydir을 활용한다. filebeat의 설정은 configmap을 사용해 전달한다.

- 파일비트 설정: filebeat-configmap
- 디플로이먼트: nx-deployment
 - 주 컨테이너: nginx
 - 사이드카 컨테이너: filebeat
 - 볼륨: emptydir을 사용해 /var/log/nginx/ 디렉토리 공유

```
cat <<EOF | kubectl apply -f - apiVersion: v1 kind: ConfigMap metadata: name:
filebeat-configmap data: filebeat.yml: | filebeat: config: modules: path:
/usr/share/filebeat/modules.d/*.yaml reload: enabled: true modules: - module:
apache access: var.paths: ["/var/log/nginx/access.log*"] error: var.paths:
["/var/log/nginx/error.log*"] output: elasticsearch: hosts: ["elasticsearch-
svc.elastic:9200"] --- apiVersion: apps/v1 kind: Deployment metadata: name:
nx-deployment labels: app: nx-deployment spec: replicas: 1 selector:
matchLabels: app: nx-pod template: metadata: name: nx-pod labels: app: nx-pod
spec: containers: - name: nx-world image: nginx ports: - containerPort: 80
volumeMounts: - name: nginx-logs mountPath: /var/log/nginx/ - name: filebeat-
sidecar image: docker.elastic.co/beats/filebeat:7.14.1 volumeMounts: - name:
nginx-logs mountPath: var/log/nginx/ - name: filebeat-config mountPath:
/usr/share/filebeat/filebeat.yml subPath: filebeat.yml volumes: - name: nginx-
logs - name: filebeat-config configMap: name: filebeat-configmap items: - key:
filebeat.yml path: filebeat.yml --- apiVersion: v1 kind: Service metadata:
labels: app: nx-deployment name: nx-deployment namespace: default spec: ports:
- nodePort: 30000 port: 80 protocol: TCP targetPort: 80 selector: app: nx-pod
type: LoadBalancer EOF
```

실습을 통해 올라온 모든 서비스를 확인한다.

```
$ kubectl get svc -A NAMESPACE NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
default kubernetes ClusterIP 10.48.0.1 <none> 443/TCP 28s default nx-
deployment LoadBalancer 10.48.1.113 <pending> 80:30000/TCP 11s elastic
elasticsearch-svc ClusterIP 10.48.8.234 <none> 9200/TCP,9300/TCP 26s elastic
kibana-svc LoadBalancer 10.48.12.36 <pending> 80:30080/TCP 21s
```

노드포트 30000으로 접속해 로그를 생성한다.

Welcome to nginx!


If you see this page, the nginx web server is successfully installed and working. Further configuration is required.



For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.




Thank you for using nginx.

키바나에 접속해 새로운 인덱스 패턴을 생성한다. 인덱스 패턴은 filebeat-7.14.1-*로 구성한다.

그리고 discover 메뉴에서 정보가 정확히 파싱되어 들어가는지 확인한다.

 elastic






  Discover 


Options New Save Open Share Inspect

Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our [Privacy Statement](#). To stop collection, disable usage data here.

Dismiss


 Search  Last 15 minutes  Refresh


 + Add filter


filebeat-7.14.1-*


Filter by type 0


Available fields 58


 _id


 _index


 _score


 _type


 @timestamp


 agent.ephemeralId


 agent.hostname


 agent.id


 agent.name


 agent.type


 agent.version


 ecs.version


 error.message


 event.category


 event.created


 event.dataset


 event.ingested

 event.kind

 event.module

 event.original


 event.outcome

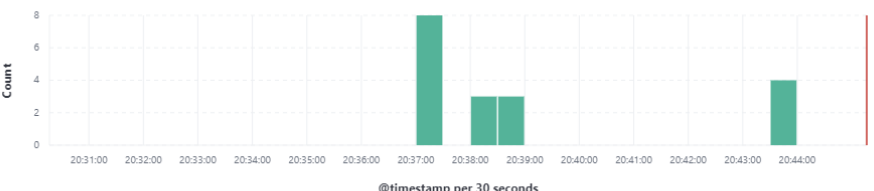
 event.timezone

18 hits

Jan 3, 2022 @ 20:30:16.612 - Jan 3, 2022 @ 20:45:16.612


Auto

 Hide chart




Count


@timestamp per 30 seconds

 event.original


10.44.0.1 - - [03/Jan/2022:11:43:55 +0000] "GET / HTTP/1.1" 304 0 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36 Edg/96.0.1054.62 "-"

 event.outcome


success

 fileset.name


access

 host.name


nx-deployment-549bfbd74-sf5q4

 http.request.method


GET

 http.request.referrer


-

 http.response.body.bytes


0

 http.response.status_code


304

 http.version


1.1

 input.type


log

 log.file.path


/var/log/nginx/access.log

 log.offset


1,498

 service.type

apache

 source.address

10.44.0.1

 source.ip

10.44.0.1