

이슈 사항 보고서

> 이슈 + 장애 + 조치방안

1. 체크리스트 항목

진단에 사용될 체크리스트는 총 5 개의 영역(EKS(2 개 항목), 애플리케이션(1 개 항목) , DB (3 개 항목), 모니터링(2 개 항목), 자동배포(1 개 항목)) 으로 구성하였습니다. 각 영역별 이슈사항을 파악하고 장애대응 조치 후 해당 시스템이 정상적으로 동작하는 지 확인할 수 있도록 총 9 개의 방법으로 구성 하였습니다.

[표] 1. 이슈사항 점검표

영역	항목코드	항목명	장애 내역	조치 사항
EKS	E.1.1	EKS Node 접근	접근 불가	권한 확인
	E.1.2	Node 장애 및 리소스 부족	Node 상태 이상 서버 동작 이상	Cluster 상태 확인 AutoScaling 설정
애플리케이션	E.2.1	웹 서버 동작 확인	Web 접근 불가 DB 연결 이상	LoadBalancer 확인 DB 연결 확인
DB	E.3.1	DB State Pending	Pod 상태 Pending	StorageClass 확인 Statefulset 확인
	E3.2	DB Status Init	Pod 상태 Init	NameSpace 확인, ConfigMap 확인
	E.3.3	Read Replica Sync	싱크 장애	바이너리 로그 및 MasterHOST 확인
모니터링	E.4.1	로그 라우팅 실패	로그 라우팅 불가로 S3 에서 로그 수집 불가	권한 및 Fluent Bit ConfigMap 확인
	E.4.2	프로메테우스 쿼리 전달 실패	프로메테우스 서버가 쿼리 전달 불가	프로메테우스 관련 환경 변수 확인
자동 배포	E.5.1	자동배포 동작 확인	자동화 불가	Rollout 상태 확인

2. 상황별 이슈 사항

2.1. EKS

2.1.1. EKS Node 접근

영역	EKS	항목코드	E.1.1																													
항목명	EKS Node 접근																															
세부 설명	EKS 클러스터 구성 후 Node 에 접근이 불가하여 추가적인 작업이 불가능한 상황입니다.																															
조치 방안	가. 권한																															
	1) IAM 권한 확인 및 부여																															
	<div><div><div>권한 정책 (7)</div><div>사용자에게 직접 연결된 정책을 통해 또는 그룹을 통해 권한을 정의합니다.</div><div><div>필터링 기준 유형</div><div>모든 유형</div><div>< 1 ></div></div><table><tr><th><input type="checkbox"/> 정책 이름</th><th>유형</th><th>연결 방식</th></tr><tr><td><input type="checkbox"/> AdministratorAccess</td><td>AWS 관리형 - 직무</td><td>직접</td></tr><tr><td><input type="checkbox"/> AmazonEC2ContainerRegistryFullA...</td><td>AWS 관리형</td><td>직접</td></tr><tr><td><input type="checkbox"/> AmazonEC2ContainerRegistryPowe...</td><td>AWS 관리형</td><td>직접</td></tr><tr><td><input type="checkbox"/> AmazonEKS_CNI_Policy</td><td>AWS 관리형</td><td>직접</td></tr><tr><td><input type="checkbox"/> AmazonEKSEKSWorkerNodePolicy</td><td>AWS 관리형</td><td>직접</td></tr><tr><td><input type="checkbox"/> AWSElementalMediaConvertFullAcc...</td><td>AWS 관리형</td><td>직접</td></tr><tr><td><input type="checkbox"/> IAMUserChangePassword</td><td>AWS 관리형</td><td>직접</td></tr></table></div><div><div>AWSServiceRoleForAmazonEKSNod... 정보</div><div>This policy allows Amazon EKS to create and manage Nodegroups</div><div><div>요약</div><div>편집</div></div><div><div>생성 날짜</div><div>September 19, 2023, 11:40 (UTC+09:00)</div><div>ARN</div><div>arn:aws:iam::621917999036:role/aws-service-role/eks-nodegroup.amazonaws.com/AWSServiceRoleForAmazonEKSNod...</div><div>마지막 활동</div><div>2시간 전</div><div>최대 세션 지속 시간</div><div>1시간</div></div><div><div>권한</div><div>신뢰 관계</div><div>태그</div><div>엑세스 관리자</div></div><div><div>권한 정책 (1) 정보</div><div>필터링 기준 유형</div><div>모든 유형</div><div>< 1 ></div></div><table><tr><th><input type="checkbox"/> 정책 이름</th><th>유형</th><th>연결된 엔티티</th></tr><tr><td><input type="checkbox"/> AWSServiceRoleForAmazonEKSNod...</td><td>AWS 관리형</td><td>1</td></tr></table></div></div>			<input type="checkbox"/> 정책 이름	유형	연결 방식	<input type="checkbox"/> AdministratorAccess	AWS 관리형 - 직무	직접	<input type="checkbox"/> AmazonEC2ContainerRegistryFullA...	AWS 관리형	직접	<input type="checkbox"/> AmazonEC2ContainerRegistryPowe...	AWS 관리형	직접	<input type="checkbox"/> AmazonEKS_CNI_Policy	AWS 관리형	직접	<input type="checkbox"/> AmazonEKSEKSWorkerNodePolicy	AWS 관리형	직접	<input type="checkbox"/> AWSElementalMediaConvertFullAcc...	AWS 관리형	직접	<input type="checkbox"/> IAMUserChangePassword	AWS 관리형	직접	<input type="checkbox"/> 정책 이름	유형	연결된 엔티티	<input type="checkbox"/> AWSServiceRoleForAmazonEKSNod...	AWS 관리형
<input type="checkbox"/> 정책 이름	유형	연결 방식																														
<input type="checkbox"/> AdministratorAccess	AWS 관리형 - 직무	직접																														
<input type="checkbox"/> AmazonEC2ContainerRegistryFullA...	AWS 관리형	직접																														
<input type="checkbox"/> AmazonEC2ContainerRegistryPowe...	AWS 관리형	직접																														
<input type="checkbox"/> AmazonEKS_CNI_Policy	AWS 관리형	직접																														
<input type="checkbox"/> AmazonEKSEKSWorkerNodePolicy	AWS 관리형	직접																														
<input type="checkbox"/> AWSElementalMediaConvertFullAcc...	AWS 관리형	직접																														
<input type="checkbox"/> IAMUserChangePassword	AWS 관리형	직접																														
<input type="checkbox"/> 정책 이름	유형	연결된 엔티티																														
<input type="checkbox"/> AWSServiceRoleForAmazonEKSNod...	AWS 관리형	1																														
2) Node 연결 상태 확인																																

	<pre> NAME STATUS ROLES AGE VERSION ip-192-168-11-113.ap-northeast-2.compute.internal Ready <none> 19m v1.26.10-eks-4f4795d ip-192-168-13-162.ap-northeast-2.compute.internal Ready <none> 19m v1.26.10-eks-4f4795d </pre>
결과	IAM 정책을 추가하여 NodeGroup 에 접근이 가능해졌고, 작업을 이어나갈 수 있었다.
비고	

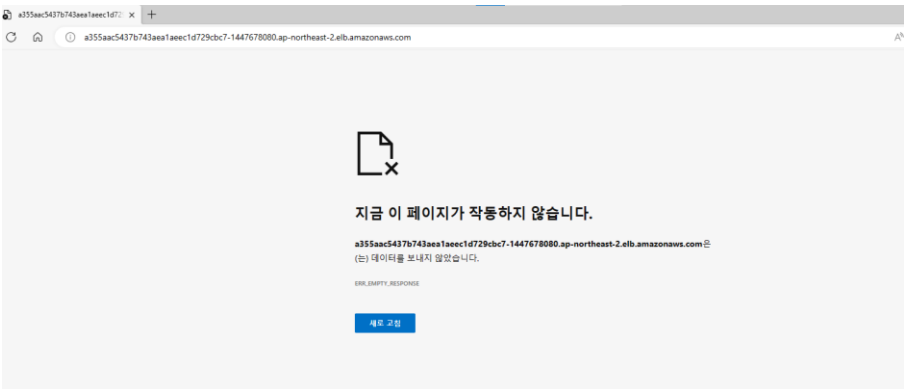
2.1.2. Node 장애 및 리소스 부족

영역	EKS	항목코드	E.1.2
항목명	Node 장애 및 리소스 부족		
세부 설명	EKS 클러스터 사용 중 Node 에 장애가 발생하여 운영 중인 Pod 가 삭제되는 이슈가 발생하였습니다. 또한 Node 의 리소스 부족으로 인해 Pod 가 pending 상태로 배포되지 않아 서비스 유지에 어려움이 있습니다.		
조치 방안	<p>가. Node 상태 확인</p> <p>1) Node 의 상태 확인</p> <pre> Every 2.0s: kubectl get nodes NAME STATUS ROLES AGE VERSION ip-192-168-11-113.ap-northeast-2.compute.internal Ready <none> 19m v1.26.10-eks-4f4795d ip-192-168-13-162.ap-northeast-2.compute.internal Ready <none> 19m v1.26.10-eks-4f4795d ip-192-168-13-181.ap-northeast-2.compute.internal NotReady <none> 0s v1.26.10-eks-4f4795d </pre> <p>2) Karpenter 설정 및 배포</p> <pre> (bonobono@myeks:N/A) [root@myeks-bastion-EC2 ~]# k get pod -n karpenter NAME READY STATUS RESTARTS AGE karpenter-65688557b8-42nt9 1/1 Running 0 50s karpenter-65688557b8-6ms7j 1/1 Running 0 50s </pre> <p>3) Node 장애로 Node 다운시 pod 도 사라짐</p> <pre> Every 2.0s: kubectl get pod,deployment,svc,ingress,hpa NAME READY STATUS RESTARTS AGE pod/admin-wordpress-58567d684c-59nt6 1/1 Running 0 8m24s pod/user-wordpress-7b86b58fd5-2hgmw 1/1 Terminating 0 112s pod/user-wordpress-7b86b58fd5-64js7 1/1 Terminating 0 112s pod/user-wordpress-7b86b58fd5-6wh2s 1/1 Terminating 0 112s pod/user-wordpress-7b86b58fd5-74skq 1/1 Terminating 0 112s </pre> <p>4) Karpent 에 의해 자동으로 Node 생성 및 pod 재생성</p> <pre> Every 2.0s: kubectl get nodes NAME STATUS ROLES AGE VERSION ip-192-168-11-113.ap-northeast-2.compute.internal Ready <none> 15m v1.26.10-eks-4f4795d ip-192-168-11-233.ap-northeast-2.compute.internal Ready <none> 14s v1.26.10-eks-4f4795d ip-192-168-13-162.ap-northeast-2.compute.internal Ready <none> 15m v1.26.10-eks-4f4795d </pre>		

	<pre> pod/user-wordpress-7b86b58fd5-2hgmw 1/1 Running 0 81s pod/user-wordpress-7b86b58fd5-64js7 1/1 Running 0 81s pod/user-wordpress-7b86b58fd5-6wh2s 1/1 Running 0 81s pod/user-wordpress-7b86b58fd5-74skq 1/1 Running 0 81s pod/user-wordpress-7b86b58fd5-7bbf6 1/1 Running 0 81s pod/user-wordpress-7b86b58fd5-7djcs 1/1 Running 0 81s pod/user-wordpress-7b86b58fd5-7qhnx 1/1 Running 0 80s pod/user-wordpress-7b86b58fd5-7w4d4 1/1 Running 0 80s </pre>
결과	조치 결과 Node 장애 발생 시에도 Karpenter 에 의해 빠르게 생성되며 pending 중인 pod 들 역시 빠른 시간내에 새로운 node 에 올라가 서비스의 안정성을 확보했다
비고	

2.2. 애플리케이션

2.2.1. 웹 서버 동작 확인

영역	애플리케이션	항목코드	E.2.1
항목명	웹 서버 동작 확인		
세부 설명	<p>웹 서버와 연결한 loadbalancer 로 접근하여 웹 접근 확인중 연결 되지 않는 문제가 발생하였습니다. loadbalancer 는 정상 동작중이더라도 웹에 접근이 되지 않아 서비스가 중지된 상황입니다.</p> <p>DB 와 연결되지 않아 웹에 접근이 되더라도 서비스를 이용할 수 없는 장애 상황입니다</p>		
조치 방안	<p>가. 웹 서버 접근 불가</p> <p>1) 웹 접근 불가</p>  <p>2) Loadbalancer 연결 확인(정상 동작중)</p>		

로트 밸런스: a355aac5437b743aea1aec1d729cbc7

세부 정보 | 레스나 | 네트워크 채팅 | 보안 | 상태 검사 | **대상 인스턴스** | 모니터링 | 속성 | 태그

대상 인스턴스 (2)

대상 인스턴스 필터링

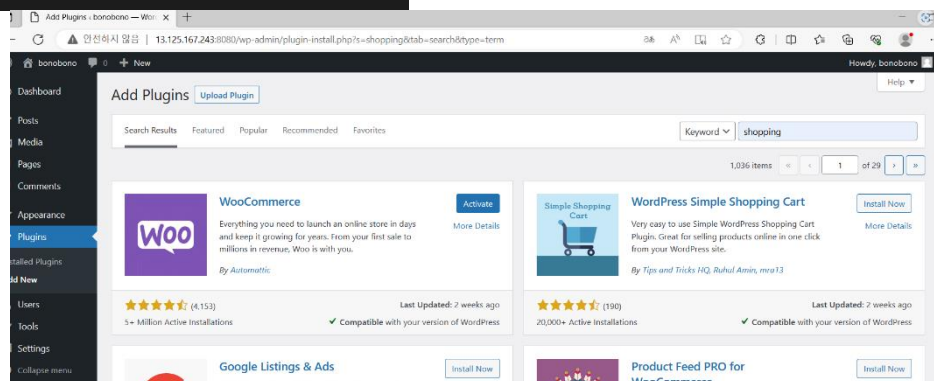
<input type="checkbox"/>	인스턴스 ID	이름	상태 확인	상태 확인 설명	보안 그룹	영역	퍼블릭 IP
<input type="checkbox"/>	i-0b09b2b9932f5472b	myeks-ng1-Node	서비스 중	확인되지 않음	eks-cluster-sg-myeks-104368993, eks...	ap-northeast-2a	-
<input type="checkbox"/>	i-033b8d752c245d436	myeks-ng1-Node	서비스 중	확인되지 않음	eks-cluster-sg-myeks-104368993, eks...	ap-northeast-2c	-

3) Pod 상태 확인

```
(bonobono@myeks:N/A) [root@myeks-bastion-EC2 ~]# k get pod
NAME                                READY   STATUS    RESTARTS   AGE
user-wordpress-567b94b84d-sfjp8    1/1     Terminating    0           6m3s
user-wordpress-598dbcdf7-cwg24     0/1     ErrImagePull    0           4s
```

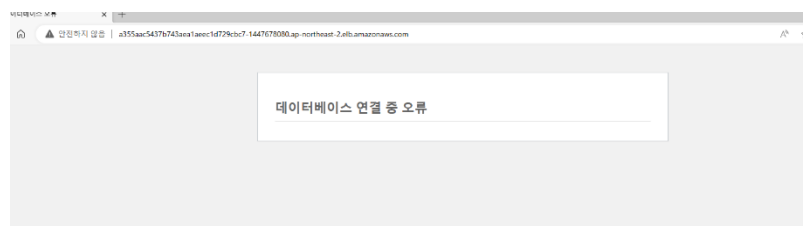
4) Deployment 수정 및 접근 확인

```
spec:
  containers:
    - image: kth3819/wp:3.2
      name: user-wordpress
      ports:
        - containerPort: 80
          name: user-wordpress
      resources:
        requests:
          memory: "512Mi"
          cpu: "0.5"
        limits:
          memory: "512Mi"
          cpu: "0.5"
```



나. DB 연결 장애

1) 웹 접근



2) DB SG 수정

인바운드 규칙 편집

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

인바운드 규칙 정보

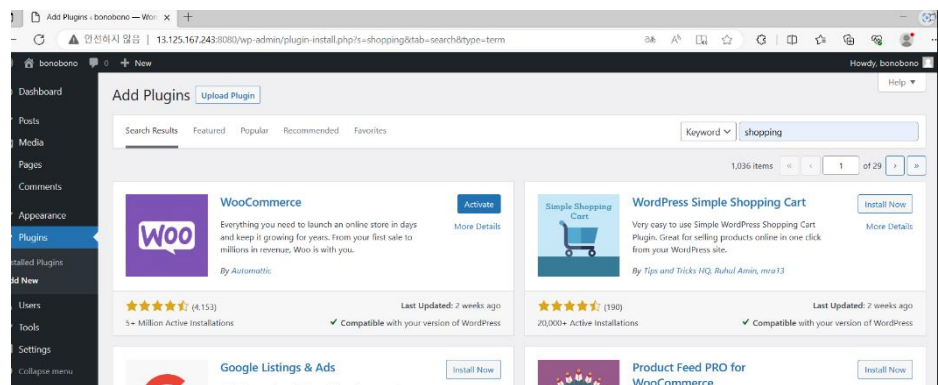
보안 그룹 규칙 ID 유형 정보 프로토콜 정보 포트 범위 정보 소스 정보 설정 - 선택 사항 정보

sg-03f8bab90c5cff6ce 모든 트래픽 전체 전체 사용... Q sg-0501eab17a642593a 삭제

- 모든 트래픽 전체 전체 Anyw... Q 0.0.0.0/0 삭제

규칙 추가

3) 접근 확인



결과

진단 기준

: LB 접근 이 가능한가
웹 서비스가 정상적으로 운영중인가
DB 연결이 되어 있는가

비고

2.3. DB

2.3.1. DB State Pending

영역	DB	항목코드	3.1
항목명	DB State Pending		
세부 설명	DB 가 Pending 상태인 경우		

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#  
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k apply -f mariadb-Server.yaml  
statefulset.apps/mariadb-sts created  
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#  
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod  
NAME          READY   STATUS    RESTARTS   AGE  
mariadb-sts-0  0/1     Pending   0           4s  
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#  
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod  
NAME          READY   STATUS    RESTARTS   AGE  
mariadb-sts-0  0/1     Pending   0           43s  
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
```

가. PVC 예러

1) kubectl describe pod 명령어로 Pending 원인 찾기

```
QoS Class:           BestEffort
Node-Selectors:      <none>
Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason              Age   From          Message
  ----    -
  Warning FailedScheduling    97s   default-scheduler    0/2 nodes are available: pod has unbound immediate PersistentVolumeClaims. preemption: 0/2 nodes are a
  Preemption is not helpful for scheduling.
  (ns=alexeys:mariaadb) [root@myeks-bastion-EC2 ~]#
```

2) 1) kubectl get pvc 명령어로 현재 PVC 상태 확인

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pvc
NAME                                STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    AGE
datadir-mariadb-sts-0              Pending    
                                     ebs-sc         4m23s
```

3) kubectl get pv 명령어로 현재 PV 상태 확인

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pvc
NAME                                STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
data-tr-mariadb-sts-0              Pending                1Gi         RWO          ebs-sc        4m23s
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pv
No resources found
```

```
(nseat@myeks.mai.tu) [root@myeks-Bastion-EC2 ~]#
```

조치 방안	4) kubectl get StorageClass 명령어로 현재 StorageClass 상태 확인
-------	--

```
No resources found
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get StorageClass
NAME                PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
gp2 (default)       kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer false                  24m
```

5) evs-sc StorageClass 생성

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# cat <<'SSTC' > StorageClass.yaml
> apiVersion: storage.k8s.io/v1
> kind: StorageClass
> metadata:
provisioner: ebs.csi.aws.com
parameters:
  type: gp2 # 혹은 원하는 다른 EBS 볼륨 타입
volumeBindingMode: WaitForFirstConsumer
allowVolumeExpansion: true
SSTC

k apply -f StorageClass.yaml>   name: ebs-sc
> namespace: mariadb
> provisioner: ebs.csi.aws.com
> parameters:
>   type: gp2 # 혹은 원하는 다른 EBS 볼륨 타입
> volumeBindingMode: WaitForFirstConsumer
> allowVolumeExpansion: true
> SSTC

(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k apply -f StorageClass.yaml
storageclass.storage.k8s.io/ebs-sc created
```

	<p>6) StorageClass 확인</p> <pre>(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# kubectl get StorageClass NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE ebs-sc ebs.csi.aws.com Delete WaitForFirstConsumer true 54s gp2 (default) kubernetes.io/aws-ebs Delete WaitForFirstConsumer false 27m</pre> <p>7) Pending pod 재시작</p> <pre>(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod NAME READY STATUS RESTARTS AGE mariadb-sts-0 0/1 Pending 0 9m9s (nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k delete pod mariadb-sts-0 -n mariadb pod "mariadb-sts-0" deleted</pre> <p>8) PV, PVC 할당 확인</p> <pre>(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pv NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE pvc-f9fe9d6-86ea-445c-b30a-3ecb82ef4fa9 10Gi RWO Delete Bound mariadb/datadir-mariadb-sts-0 ebs-sc 47s (nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pvc NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE datadir-mariadb-sts-0 Bound pvc-f9fe9d6-86ea-445c-b30a-3ecb82ef4fa9 10Gi RWO ebs-sc 11m</pre> <p>9) Pod 상태 확인</p> <pre>(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod NAME READY STATUS RESTARTS AGE mariadb-sts-0 1/1 Running 0 15s</pre>
결과	<p>진단 기준</p> <p>: StorageClass 존재 확인</p> <p>: Statefulset 볼륨 확인</p>
비고	

2.3.2. DB Status Init

영역	DB	항목코드	E.3.2
항목명	DB Status Init		
세부 설명	<p>현재 DB Pod 를 생성하면 Init0/1 에서 장기간 멈출 때</p> <pre>(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod NAME READY STATUS RESTARTS AGE mariadb-sts-0 0/1 Init:0/1 0 3m22s</pre>		
조치 방안	<p>가. ConfigMap 에러</p> <p>1) kubectl describe pod mariadb-sts-0 명령어를 통해 현재 Pod 의 상태 상세확인</p>		


```
Optional: false
mariadb-config:
  Type: EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
  SizeLimit: <unset>
initdb:
  Type: EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
  SizeLimit: <unset>
kube-api-access-sjv47:
  Type: Projected (a volume that contains injected data from multiple sources)
  TokenExpirationSeconds: 3607
  ConfigMapName: kube-root-ca.crt
  ConfigMapOptional: <nil>
  DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type Reason Age From Message
  ----
Normal Scheduled 2m52s default-scheduler Successfully assigned mariadb/mariadb-sts-0 to ip-192-168-155-212.ap-northeast-2.compute-1.amazonaws.com
Warning FailedMount 49s kubelet AttachVolume.Attach succeeded for volume "pvc-f9fe0d0-88ea-445c-b30a-3acb2ef4fa0"
Warning FailedMount 44s (x9 over 2m51s) kubelet Unable to attach or mount volumes: unmounted volumes=[mariadb-config-map], unattached volumes=[initdb kube-api-access-sjv47 datadir mariadb-config-map mariadb-config]: timed out waiting for the condition
Warning FailedMount 44s (x9 over 2m51s) kubelet MountVolume.Setup failed for volume "mariadb-config-map" : configmap "mariadb-config-map" not found
```

2) configmap 확인

```
kube-root-ca.crt 1 22m
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# kubectl get configmap -n mariadb
NAME DATA AGE
kube-root-ca.crt 1 23m
```

3) configmap 생성 및 확인

```
> GRANT ALL PRIVILEGES ON *.* TO 'repluser'@'%' WITH GRANT OPTION;
> CREATE DATABASE wordpress;

> secondary.sql: |
> # We have to know name of sts ('mariadb-sts') and
> # service 'mariadb-service' in advance as an FQDN.
> # No need to use master_port
> CHANGE MASTER TO
> MASTER_HOST='mariadb-sts-0.mariadb-service.mariadb.svc.cluster.local',
> MASTER_USER='repluser',
> MASTER_PASSWORD='replsecret',
> MASTER_CONNECT_RETRY=10;
> SSTC

(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k apply -f mariadb-ConfigMap.yaml
configmap/mariadb-configmap created
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# kubectl get configmap -n mariadb
NAME DATA AGE
kube-root-ca.crt 1 23m
mariadb-configmap 4 8s
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
```

4) DB 재시작

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod
NAME READY STATUS RESTARTS AGE
mariadb-sts-0 0/1 Init:0/1 0 6m58s
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k delete pod mariadb-sts-0
pod "mariadb-sts-0" deleted
```

5) Pod 상태 확인

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod
NAME READY STATUS RESTARTS AGE
mariadb-sts-0 1/1 Running 0 15s
```

결과

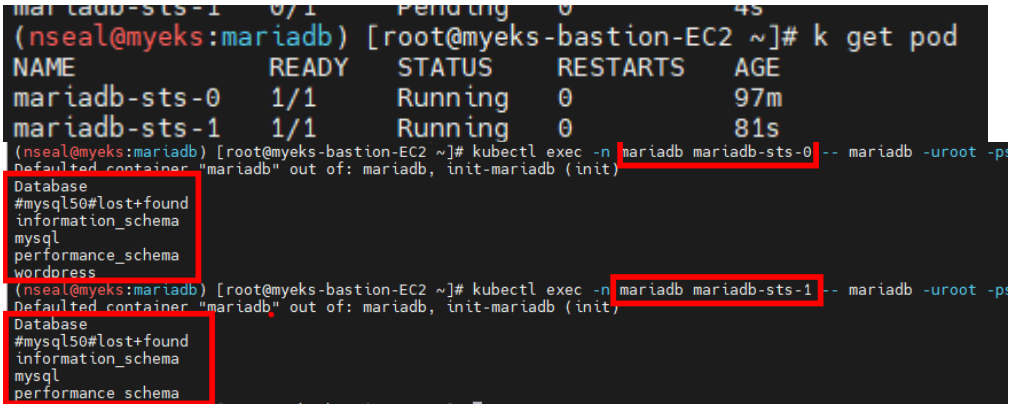
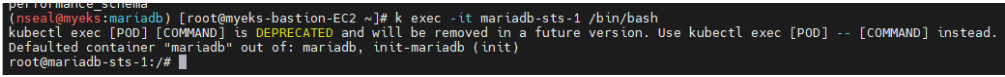
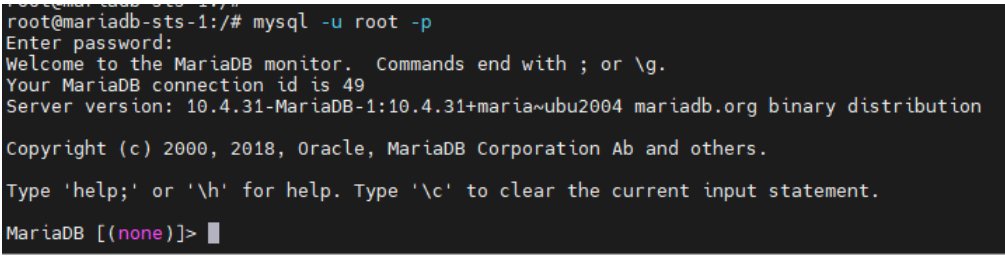
진단 기준

: ConfigMap 존재 확인

: ConpigMap, Statefulset 네임스페이스 확인

비고

2.3.3. Read Replica Sync

영역	DB	항목코드	E.3.3
항목명	Read Replica Sync		
세부 설명	쿼리 요청 시 Master / Slave 의 데이터 값이 다르게 나오는 경우		
조치 방안	<p>가. Master/Slave 가 Running 이지만 데이터 싱크가 되지 않는 경우</p> 		
	<p>1) Slave Pod 접속</p> 		
	<p>2) MariaDB 접속</p> 		
	<p>3) SHOW SLAVE STATUSWG 명령어로 설정한 Master 의 정보(HOST, User, Prot), Error 메시지 확인</p>		

```
Master_Host: mariadb-sfs-0.mariadb-service.default.svc.cluster.local
Master_User: repluser
Master_Port: 3306
```

```
Slave_SQL_Running: Yes
```

현재 DB 의 네임스페이스 경로가 잘못 설정되어있다.

- MASTER_HOST='mariadb-sts-0.mariadb-service.default.svc.cluster.local' 를
MASTER_HOST='mariadb-sts-0.mariadb-service.DB의 네임 스페이스.svc.cluster.local' 로 변경 후 적용

5) Configmap 적용 후 Pod, Pvc(Slave)삭제, 재시작

```
> # We have to know name of sts ( mariadb-sts ) and
> # service 'mariadb-service' in advance as an FQDN.
> # No need to use master_port
> CHANCE MASTER TO
>
> MASTER_HOST='mariadb-sts-0.mariadb-service.mariadb.svc.cluster.local',
> MASTER_USER='repluser',
> MASTER_PASSWORD='replsecret',
> MASTER_CONNECT_RETRY=10;
>
> SSTC

(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k apply -f mariadb-ConfigMap.yaml
configmap/mariadb-configmap configured
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k delete statefulset mariadb-sts
statefulset.apps "mariadb-sts" deleted
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k delete pvc datadir-mariadb-sts-1
persistentvolumeclaim "datadir-mariadb-sts-1" deleted
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k apply -f mariadb-Server.yaml
statefulset.apps/mariadb-sts created
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]#
```

6) Slave pod 접속, SHOW SLAVE STATUS\G 명령어로 현재 상태 확인

```
(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k get pod
NAME          READY   STATUS    RESTARTS   AGE
mariadb-sts-0  1/1     Running   0           70s
mariadb-sts-1  1/1     Running   0           57s

(nseal@myeks:mariadb) [root@myeks-bastion-EC2 ~]# k exec -it mariadb-sts-1 /bin/bash
Kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
Defaulted container "mariadb" out of: mariadb, init-mariadb (init)
root@mariadb-sts-1:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.4.31-MariaDB-1:10.4.31+maria~ubu2004 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW SLAVE STATUS\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: mariadb-sts-0.mariadb-service.mariadb.svc.cluster.local
Master_User: repluser
Master_Port: 3306
Connect_Retry: 10
Master_Log_File: my-mariadb-bin.000003
Read_Master_Log_Pos: 347
Relay_Log_File: my-mariadb-relay-bin.000005
Relay_Log_Pos: 651
Relay_Master_Log_File: my-mariadb-bin.000003
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
```

	<pre> Skip_Counter: 0 Exec_Master_Log_Pos: 347 Relay_Log_Space: 886338 Until_Condition: None Until_Log_File: Until_Log_Pos: 0 Master_SSL_Allowed: No Master_SSL_CA_File: Master_SSL_CA_Path: Master_SSL_Cert: Master_SSL_Cipher: Master_SSL_Key: Seconds_Behind_Master: 0 Master_SSL_Verify_Server_Cert: No Last_IO_Errno: 0 Last_IO_Error: Last_SQL_Errno: 0 Last_SQL_Error: Replicate_Ignore_Server_Ids: Master_Server_Id: 3000 Master_SSL_Crl: Master_SSL_Crlpath: Using_Gtid: No Gtid_IO_Pos: Replicate_Do_Domain_Ids: Replicate_Ignore_Domain_Ids: Parallel_Mode: conservative SQL_Delay: 0 SQL_Remaining_Delay: NULL Slave_SQL_Running_State: Slave has read all relay log; waiting for the slave I/O thread to update it Slave_DDL_Groups: 204 Slave_Non_Transactional_Groups: 0 Slave_Transactional_Groups: 27 1 row in set (0.000 sec) MariaDB [(none)]> </pre>
	<p>7) 쿼리 출력 확인</p> <pre> MariaDB [(none)]> use wordpress; Reading table information for completion of table and column names You can turn off this feature to get a quicker startup with -A Database changed MariaDB [wordpress]> SELECT * FROM wp_users; +-----+-----+-----+-----+-----+-----+-----+-----+ ID user_login user_pass user_nicename user_email user_url user_registered user_activation_key user_status +-----+-----+-----+-----+-----+-----+-----+-----+ 1 sstc \$p\$ha67U7ZYacYiZHf9.Gt5PM50XvhjP. sstc sstc@gmail.com http://localhost 2023-11-06 14:21:46 0 +-----+-----+-----+-----+-----+-----+-----+-----+ 1 row in set (0.001 sec) </pre>
결과	<p>진단 기준</p> <ul style="list-style-type: none"> : 동일 쿼리 실행 시 동일 결과값 출력 확인 : 바이너리 로그 파일 이름, 액세스 지점 일치 확인 : 에러 메시지 분석 : Configmap 의 네임스페이스 확인 : Configmap 적용 후 PVC, Pod 삭제
비고	

2.4. 모니터링

2.4.1. S3 Bucket 으로 로그 전송 불가

영역	모니터링	항목코드	E.4.1
항목명	S3 Bucket 으로 로그 전송 불가		
세부 설명	EKS 내부 환경에서 FluentBit 를 이용한 로그 S3 Bucket 으로 전송되지 않는 문제 발생했습니다.		

가. FluentBit 구성 확인

1) FluentBit ConfigMap 확인

#명령어로 ConfigMap 구성 진입

kubectl edit configmap fluent-bit-config -n amazon-cloudwatch

```
Name                tail
Tag                 host.secure
Path                /var/log/secure
Parser              syslog
DB                  /var/fluent-bit/state/flb_secure.db
Mem_Buf_Limit       5MB
Skip_Long_Lines     On
Refresh_Interval    10
Read_from_Head      ${READ_FROM_HEAD}

[FILTER]
  Name              aws
  Match             host.*
  imds_version      v2

[OUTPUT]
  Name              cloudwatch_logs
  Match             host.*
  region            ${AWS_REGION}
  log_group_name    /aws/containerinsights/${CLUSTER_NAME}/host
  log_stream_prefix ${HOST_NAME}.
  auto_create_group true
  extra_user_agent  container-insights

[OUTPUT]
  Name              s3
  Match             *
  bucket            eks-log
  s3_key_format     /$TAG[2]/$TAG[0]/%Y/%m/%d/$UUID.gz
  s3_key_format_tag_delimiters .-
```

조치 방안

ConfigMap 구성 안에서 OUTPUT 으로 S3 설정과 bucket 이름 일치 확인

2) Daemonset 재시작

#Fluent Bit Daemonset 재시작하여 구성 초기화

kubectl rollout restart daemonset fluent-bit -n amazon-cloudwatch

```
(t)rcxs12@nyeks:default) [root@nyeks-bastion-EC2 ~]# kubectl edit configmap fluent-bit-config -n amazon-cloudwatch
configmap/fluent-bit-config edited
(1)rcxs12@nyeks:default) [root@nyeks-bastion-EC2 ~]# kubectl rollout restart daemonset fluent-bit -n amazon-cloudwatch
daemonset.apps/fluent-bit restarted
(1)rcxs12@nyeks:default) [root@nyeks-bastion-EC2 ~]# kubectl get po -n amazon-cloudwatch
NAME                                READY   STATUS    RESTARTS   AGE
loudwatch-agent-48s97              1/1     Running   0           5m21s
loudwatch-agent-splqz              1/1     Running   0           5m21s
luent-bit-dqsfz                    1/1     Running   0           4s
luent-bit-rs496                    0/1     ContainerCreating 0           1s
(1)rcxs12@nyeks:default) [root@nyeks-bastion-EC2 ~]#
```

나. IAM 정책 확인

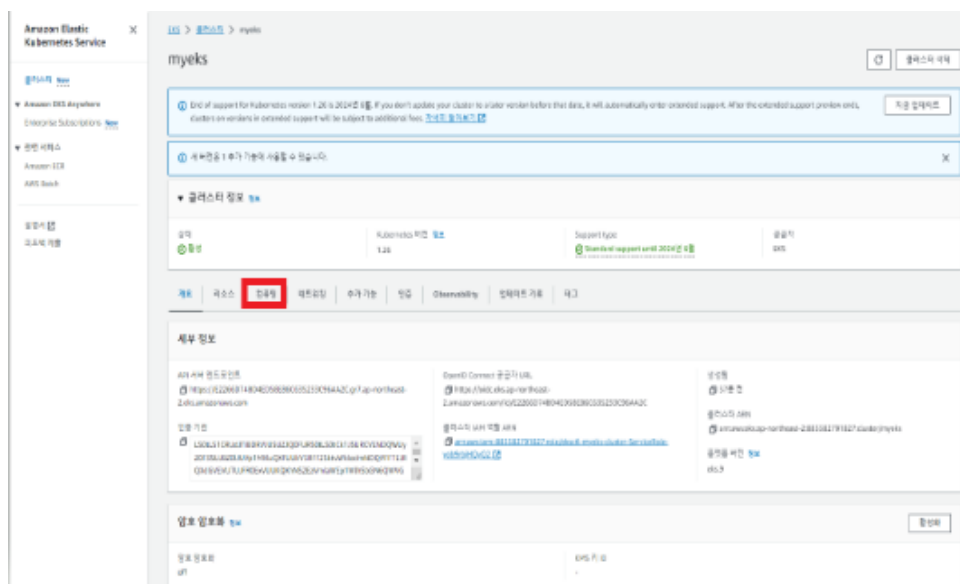
1) S3 에 대한 Policy 구성



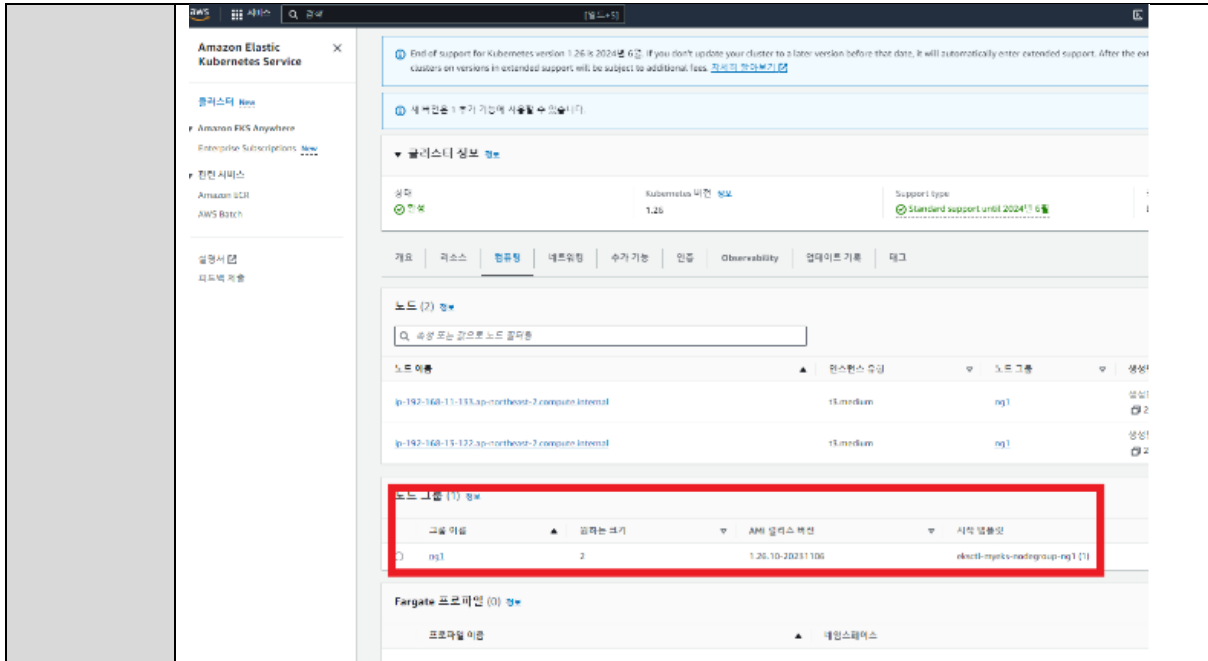
위 사진과 같이 S3 에 대해 읽기 , 쓰기 , 나열 가능한 정책을 생성 합니다

2) Node Group 에 IAM Policy 부여

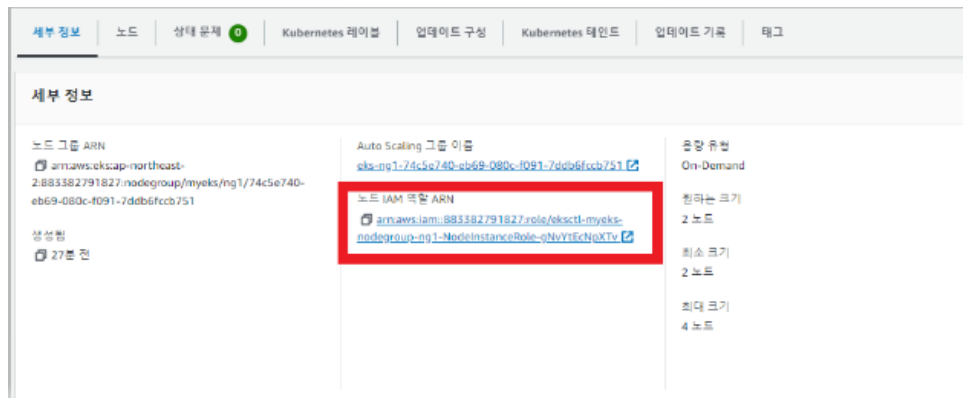
1-1. EKS 콘솔로 접근해 표시 된 컴퓨팅 메뉴 진입



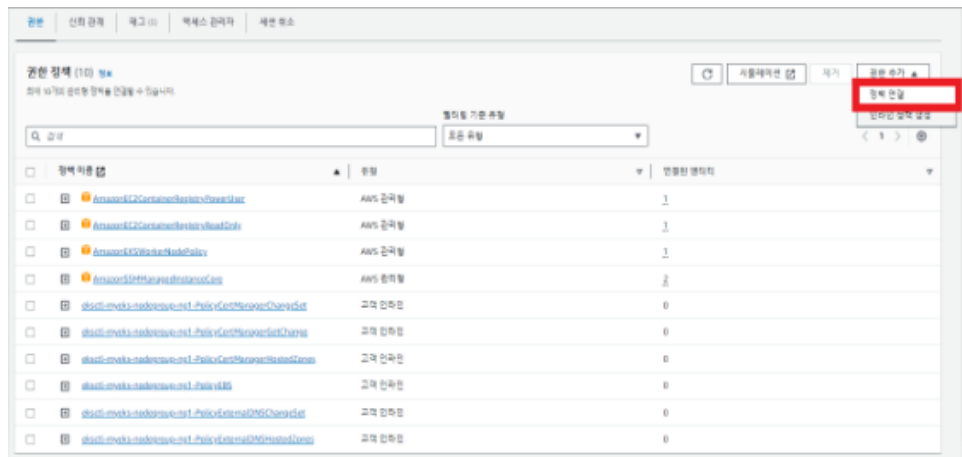
1-2. 해당 워커 노드 그룹 선택해서 설정 콘솔로 진입



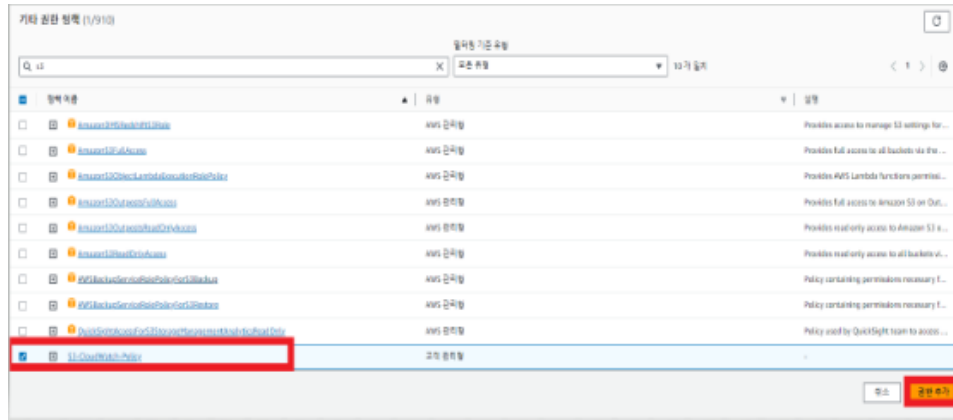
1-3. 해당 노드 그룹 IAM 역할 구성 콘솔 진입



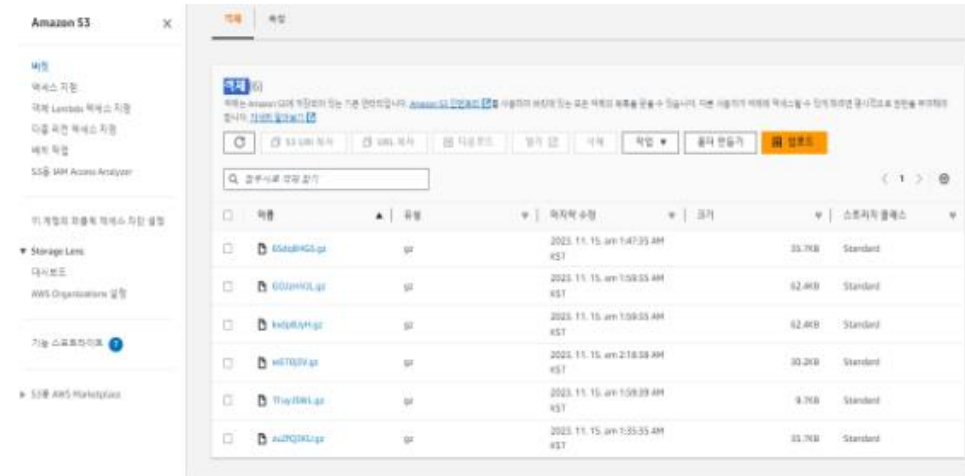
1-4. 정책 연결 콘솔 진입



1-5. 위에서 구성했던 S3 Policy 연결



정상적으로 S3 Bucket 으로 로그 전송 된 모습 확인



결과

Fluent Bit ConfigMap 구성을 통해서 S3 로 로그 전송 가능해지고 관련 S3 Policy 를 연결하여 정상적으로 S3 로 로그 전송되는 모습 확인

비고

2.4.2. 프로메테우스 쿼리 실패

영역	모니터링	항목코드	E.4.1
항목명	프로메테우스 쿼리 AMP 전달 실패		
세부 설명			
조치 방안	<p>가. AMP 환경 변수 확인</p> <p>1) AMP 환경 변수 구성</p> <p>export REGION="사용하는 리전"</p>		

```
export WORKSPACE_ID="AMP 워크 스페이스 "  
export AMP_QUERY_ENDPOINT= "AMP 엔드 포인트 URL"
```

#awscurl 을 통해서 AMP 를 통해서 Query 출력 확인
awscurl -X POST --region \$REGION --service aps
"\$AMP_QUERY_ENDPOINT?query=up"

```
[root@ip-10-10-10-10 ~]# export REGION=northwest-2  
[root@ip-10-10-10-10 ~]# export WORKSPACE_ID="us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2"  
[root@ip-10-10-10-10 ~]# export AMP_QUERY_ENDPOINT="https://aps-workspaces.ap-northwest-2.amazonaws.com/workspaces/us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2/aps/v1/query"  
[root@ip-10-10-10-10 ~]# export WORKSPACE_ID="us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2"  
[root@ip-10-10-10-10 ~]# awscurl --region $REGION --service aps "$AMP_QUERY_ENDPOINT?query=up"  
{"status": "success", "data": {"resultType": "vector", "result": [{"]]}}
```

1) AMP 메트릭 쿼리 조회

#환경 변수를 이용한 명령어를 통해 AMP 메트릭 쿼리 조회

awscurl -X POST --region \$REGION --service aps

"\$AMP_QUERY_ENDPOINT?query=container_cpu_usage_seconds_total" | jq

```
{  
  "metric": {  
    "name": "container_cpu_usage_seconds_total",  
    "type": "cpu",  
    "endpoint": "aps-workspaces",  
    "id": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "image": "awscli/awscli",  
    "instance": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "job": "awscli",  
    "metrics_path": "metrics/containers",  
    "name": "container_cpu_usage_seconds_total",  
    "namespace": "prometheus",  
    "pod": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "prometheus_replica": "prometheus-kube-prometheus-stack-prometheus-0",  
    "service": "kubernetes-stack-kubelet",  
    "value": {  
      "value": 1000000000.0,  
      "unit": "seconds" }  
    }  
  },  
  "metric": {  
    "name": "container_cpu_usage_seconds_total",  
    "type": "cpu",  
    "endpoint": "aps-workspaces",  
    "id": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "image": "awscli/awscli",  
    "instance": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "job": "awscli",  
    "metrics_path": "metrics/containers",  
    "name": "container_cpu_usage_seconds_total",  
    "namespace": "prometheus",  
    "pod": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "prometheus_replica": "prometheus-kube-prometheus-stack-prometheus-0",  
    "service": "kubernetes-stack-kubelet",  
    "value": {  
      "value": 1000000000.0,  
      "unit": "seconds" }  
    }  
  },  
  "metric": {  
    "name": "container_cpu_usage_seconds_total",  
    "type": "cpu",  
    "endpoint": "aps-workspaces",  
    "id": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "image": "awscli/awscli",  
    "instance": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "job": "awscli",  
    "metrics_path": "metrics/containers",  
    "name": "container_cpu_usage_seconds_total",  
    "namespace": "prometheus",  
    "pod": "us_c5f0bc3d-0abf-4b95-b197-ed00a0c5b4a2",  
    "prometheus_replica": "prometheus-kube-prometheus-stack-prometheus-0",  
    "service": "kubernetes-stack-kubelet",  
    "value": {  
      "value": 1000000000.0,  
      "unit": "seconds" }  
    }  
}
```

결과

AMP 의 엔드포인트 혹은 워크 스페이스 ID 를 잘못 입력하는 경우 발생하는 문제입니다. 한번 더 확인하고 다시 넣고 해당 명령어로 구성 확인하는 경우 AMP 에서 메트릭 Query 제대로 전달되는지 확인합니다.

비고

2.5. 자동배포

2.5.1. 자동배포

영역	자동배포	항목코드	5.1
항목명	자동배포 동작 확인		
세부 설명	<ol style="list-style-type: none"> Blue/Green 배포 시 Paused 상태에서 배포가 진행되지 않은 상태 확인 <pre>(test12@myeks:N/A) [root@myeks-bastion-EC2 ~]# kubectl argo rollouts list rollout -n argocd NAME STRATEGY STATUS STEP SET-WEIGHT READY DESIRED UP-TO-DATE AVAILABLE rollout-bluegreen BlueGreen Paused - - 2/4 2 2 2</pre> 이미 배포가 되어 있는 Pod 들의 상태 확인 <pre>rollout-bluegreen-84745f6c67-76sh7 1/1 Running 0 6m20s rollout-bluegreen-84745f6c67-qxkmb 1/1 Running 0 16m</pre> Github 의 Rollout 파일 확인 후 배포 자동화 안되어 있는 상태 확인 <pre>strategy: blueGreen: #activeService는 이전의 배포된 Blue 서비스 activeService: rollout-bluegreen-active #previewService는 새롭게 배포될 Green 서비스 previewService: rollout-bluegreen-preview autoPromotionEnabled: false</pre> Promote 명령어를 통해 배포 시작 <pre>(test12@myeks:N/A) [root@myeks-bastion-EC2 ~]# kubectl argo rollouts promote rollout-bluegreen -n argocd rollout 'rollout-bluegreen' promoted</pre> 		
조치 방안	AutoPromotionEnabled 를 활성화하여 배포까지 자동화하게 구성하였습니다.		
결과	운영자의 기술 습득 부족으로 생긴 단순한 해프닝으로 직원 교육을 통해 해당 내용 교육		
비고			