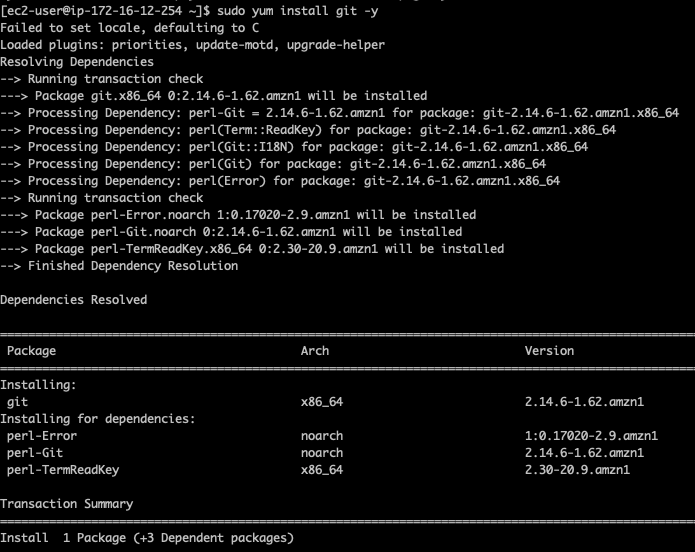
**Microservice on AWS HOL**

0. 해당 실습은 AWS Amazon Linux2 EC2 인스턴스에서 진행한 사항을 기준으로 작성하였습니다.

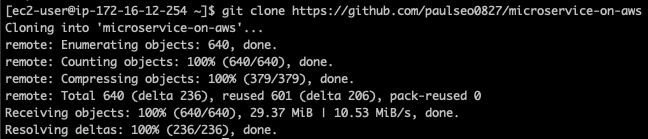
1. git 설치

sudo yum update

sudo yum install git -y



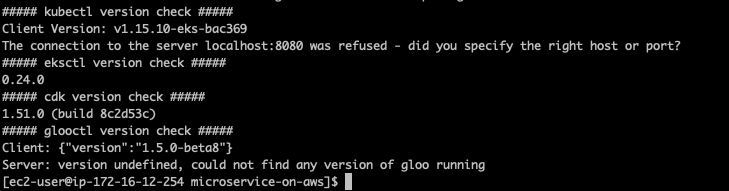
2. 실습에 사용할 소스 repository clone

git clone https://github.com/paulseo0827/microservice-on-aws.git

3. 작업에 필요한 툴 설치 : kubectl, eksctl, cdk, glooctl 설치 정상인지 버전 체크 확인

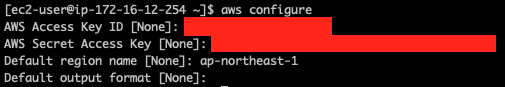
cd microservice-on-aws

./0\_install.sh



4. aws cli 설정 : AdministratorAccess Policy 가 들어있는 IAM User 등록, Default Region 은 사용자가 작업하는 곳으로 설정

aws configure



참고 링크 : <https://docs.aws.amazon.com/ko_kr/cli/latest/userguide/cli-configure-quickstart.html>

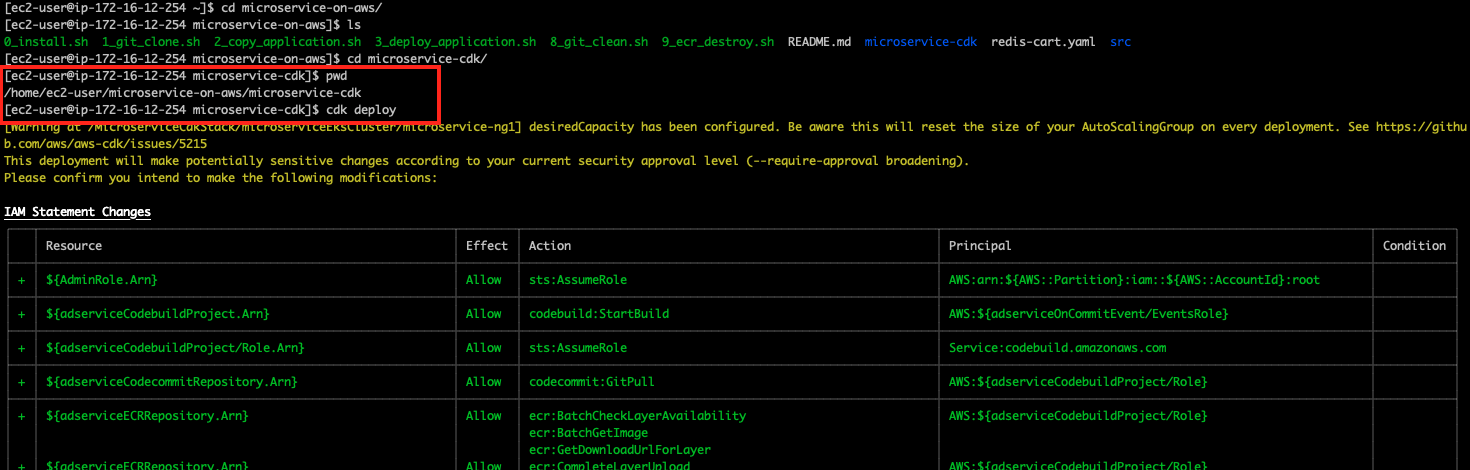
5. cdk 를 이용해서, VPC, EKS, CodeBuild, ECR 구성하기

cd microservice-cdk

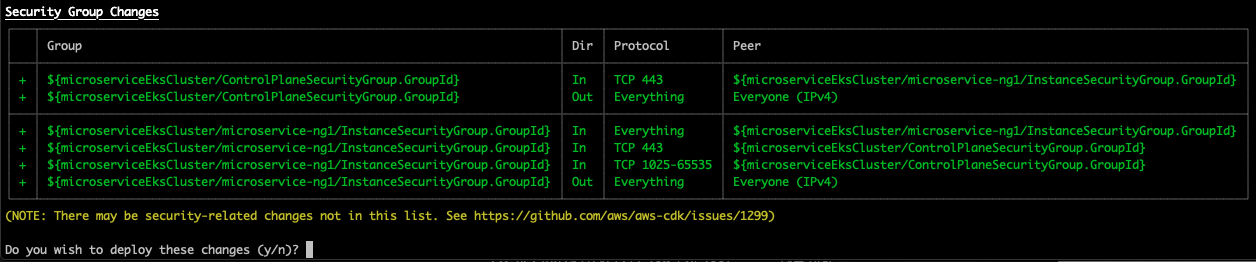
. ~/.bash\_profile

cdk bootstrap

cdk deploy

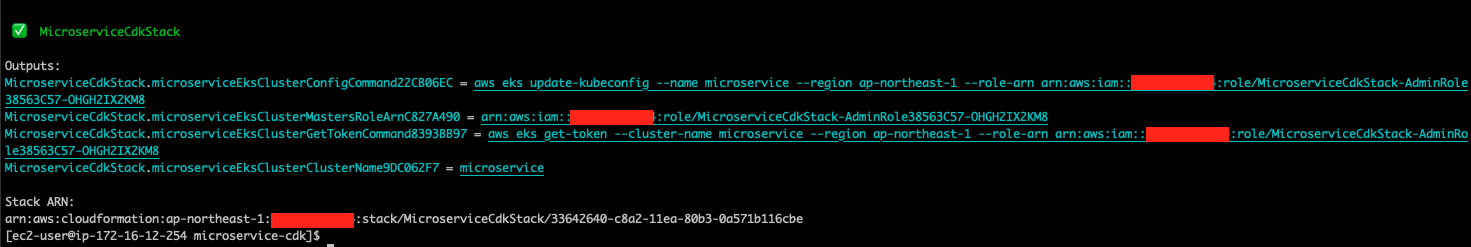


아래와 같이 변경된 사항에 대해서 deploy 할지 물어보면 y 값을 입력



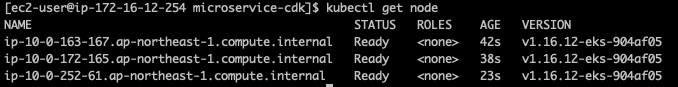
작업 소요시간은 평균 30분 내외 걸림

6. 작업이 완료되면 하단과 같이 Outputs 이 나옴



7. 상단 Output 에 첫번째 줄, “aws eks update-kubeconfig …” 명령어 줄을 복사하여 그대로 수행하고, 아래 명령어로 EKS에 Worker Node 3개 리스트 확인

kubectl get node



8. codecommit 관련 git config 수정, 아래 명령어 수행 후, 하단 박스와 같게 데이터를 입력

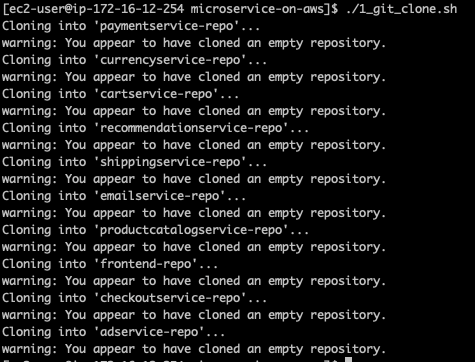
git config --global --edit

|  |
| --- |
| [credential]  helper = !aws codecommit credential-helper $@  Usehttppath = true |

참고 링크 : https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-https-windows.html

9. cdk로 생성한 codecommit repository 10개 clone 작업, “1\_git\_clone.sh” 명령어 수행

./1\_git\_clone.sh



10. src 에 있는 파일들을 codecommit 으로 복사, “2\_copy\_application.sh” 명령어 수행

./2\_copy\_application.sh

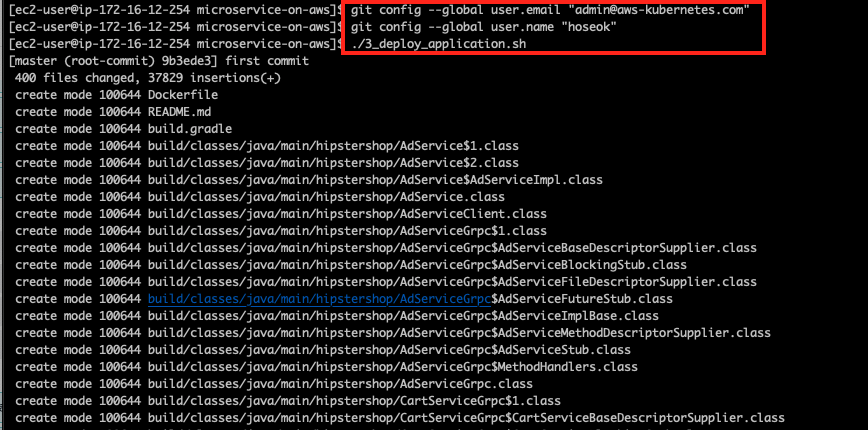


11. codecommit 디렉토리로 복사한 어플리케이션 소스를 commit 함, “3\_deploy\_application.sh” 명령어 수행

git config --global user.email "email address"

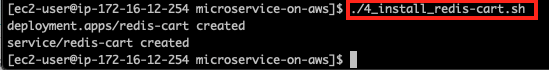
git config --global user.name "Your Name"

./3\_deploy\_application.sh

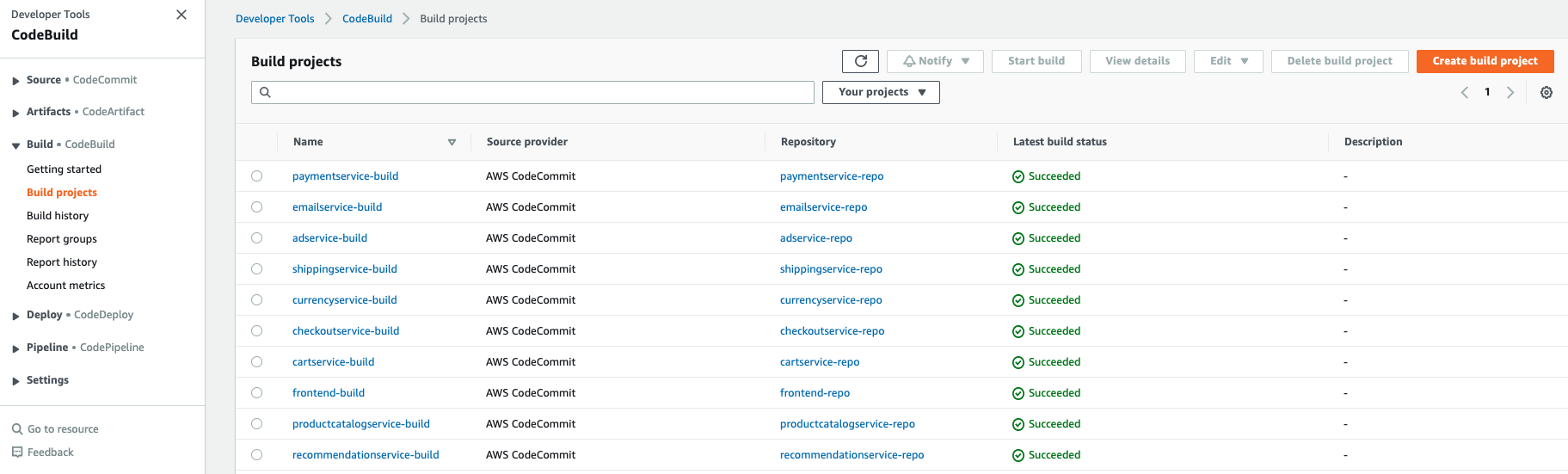


12. redis application EKS microservice 에 설치, “4\_install\_redis-cart.sh” 명령어 수행

./4\_install\_redis-cart.sh



13. 콘솔에 CodeBuild 접속하여 10개 배포 완료 확인



14. kubectl 명령어로 11개 pod가 정상적으로 기동되어 있는지 확인

kubectl -n microservice get pods

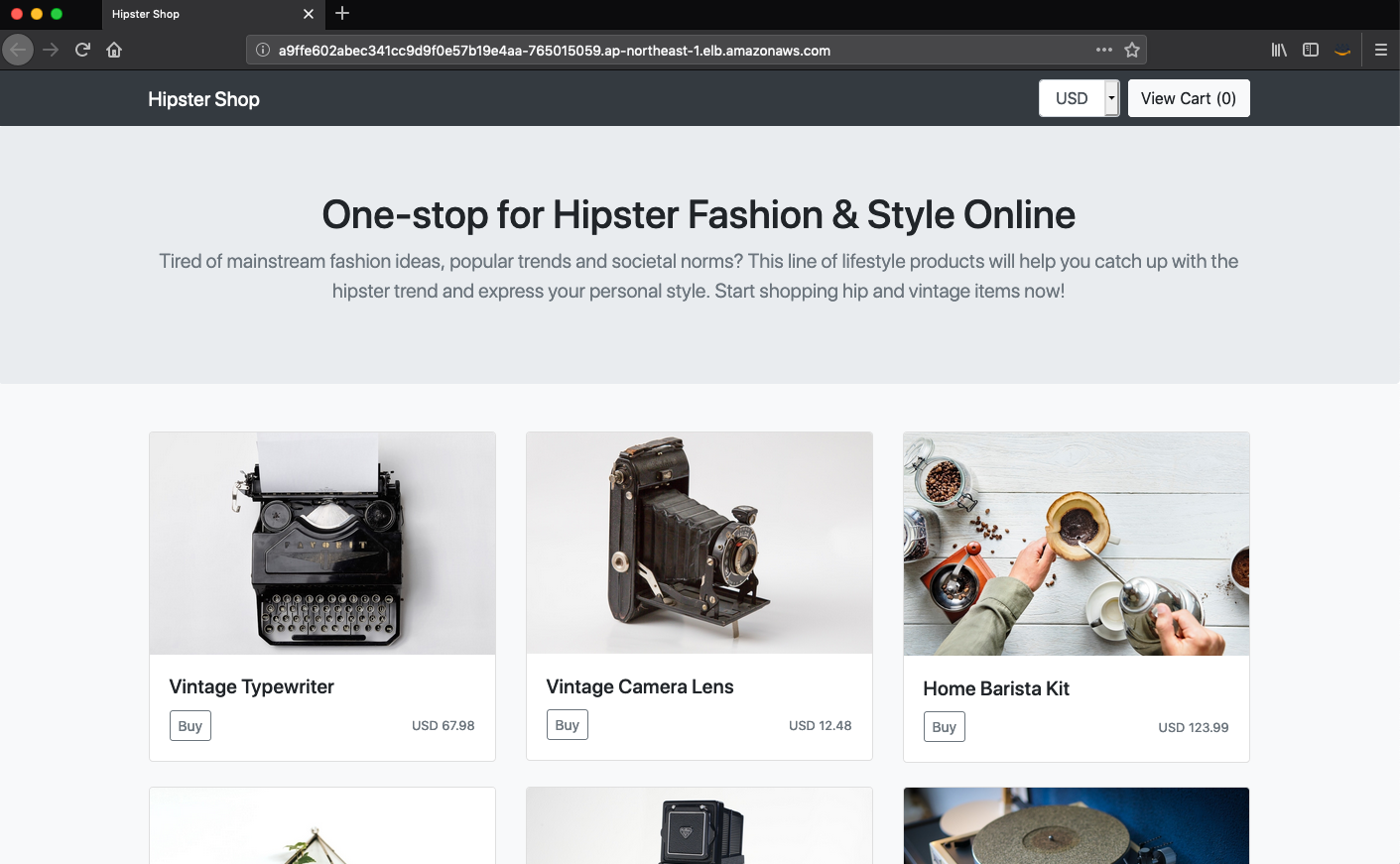


15. gloo gateway 설치, “5\_install\_gloogateway.sh” 명령어 수행

./ 5\_install\_gloogateway.sh

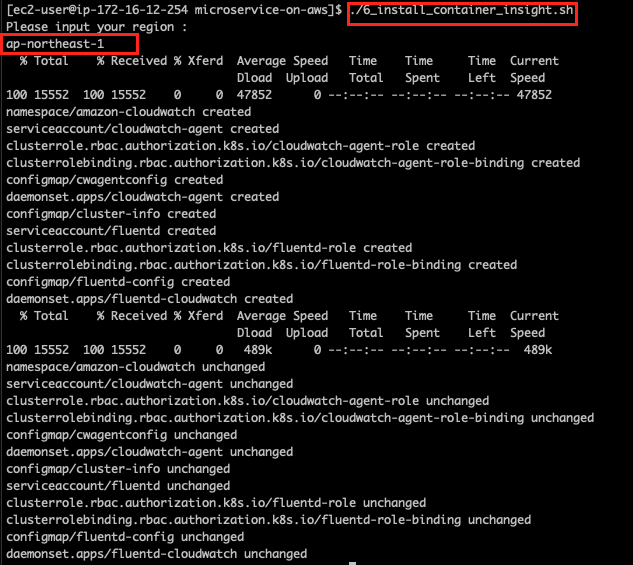


16. 위에 화면에 결과물로 나온 ELB 주소를 웹브라우저에 입력하면 하단처럼 화면을 확인할 수 있음

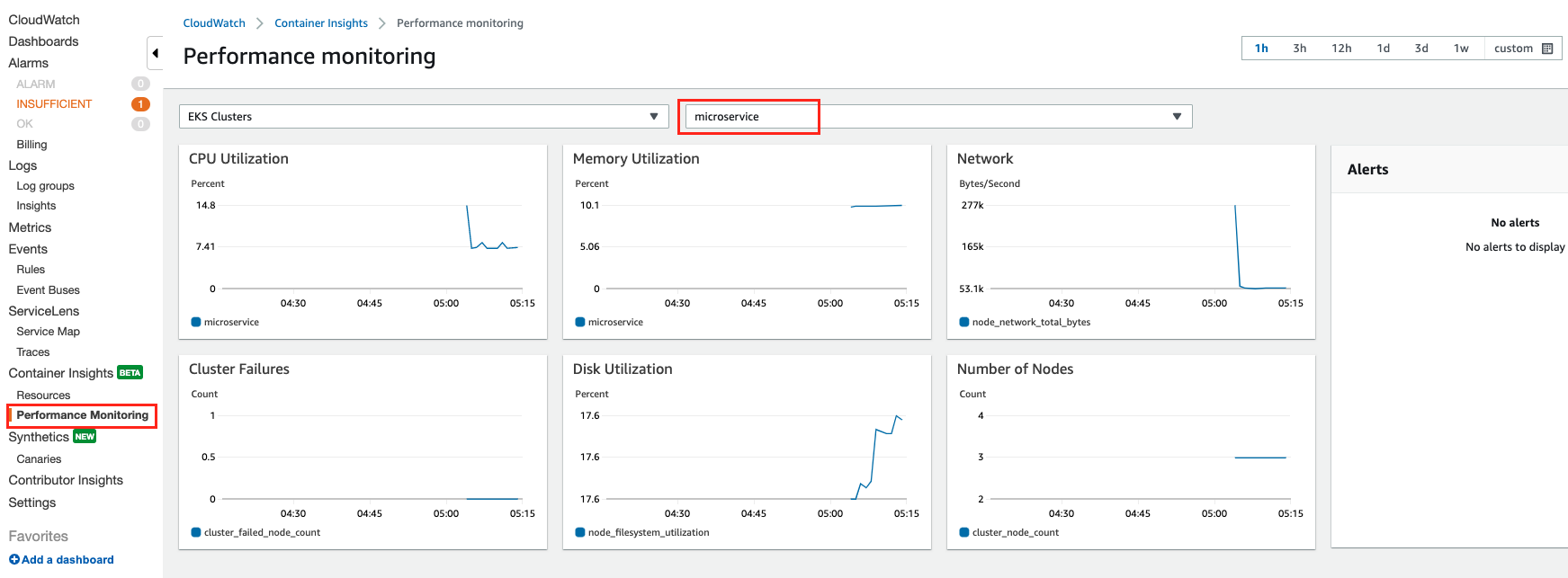


17. Amazon EKS 모니터링을 위해서 Container Insight 설치, “6\_install\_container\_insight.sh” 명령어 수행

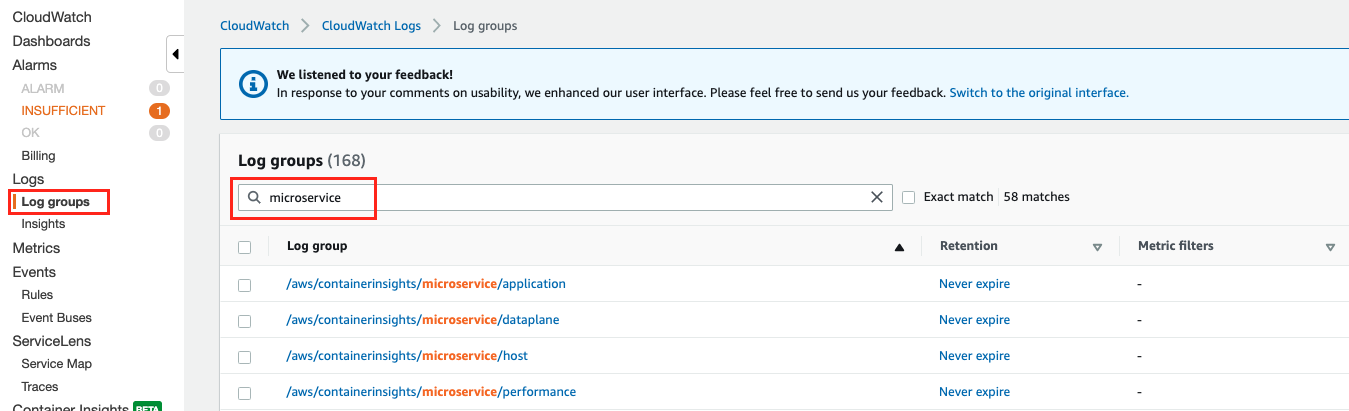
./6\_install\_container\_insight.sh



18. 콘솔에 CloudWatch 서비스에서, 왼쪽에 “Performance Monitoring” 를 선택하여서, microservice Cluster 를 생성하면 Container Insight 그래프를 볼 수 있음



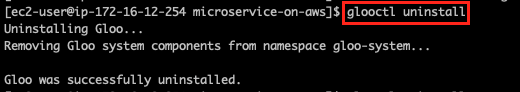
19. 콘솔 CloudWatch 서비스에서, 왼쪽에 “Log groups” 를 선택하여 microservice 를 입력하면 EKS 에서 생성하는 로그와 application에서 생성하는 로그를 확인할 수 있음



**여기서부터는 만들었던 자원을 삭제하는 단계 입니다.**

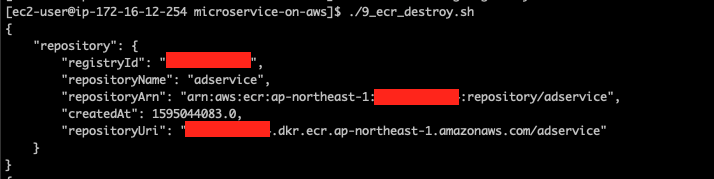
20. gloo gateway 삭제

glooctl uninstall



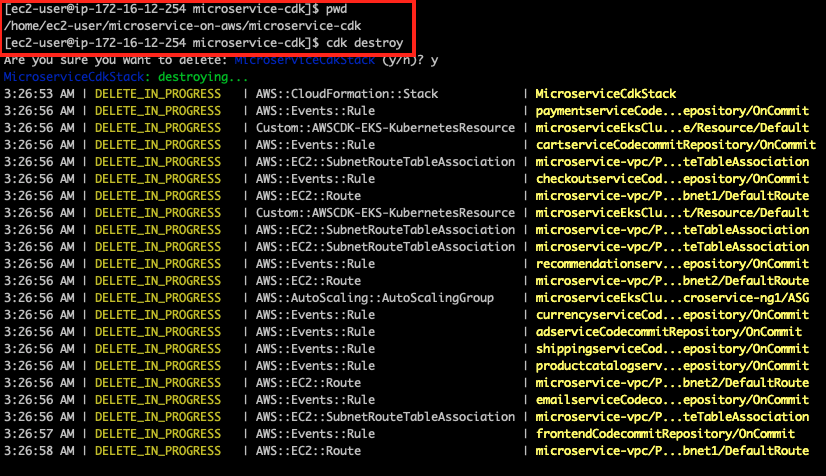
21. Docker Image 가 저장되어있는 ECR 삭제 , “9\_ecr\_destroy.sh” 명령어 수행

./9\_ecr\_destroy.sh



22. “microservice-cdk” 디렉토리에서 “cdk destroy” 명령어로 생성한 자원 삭제

cdk destroy



23. 자원이 정상적으로 삭제되면 하단처럼 메시지가 나오면서 종료됩니다.

