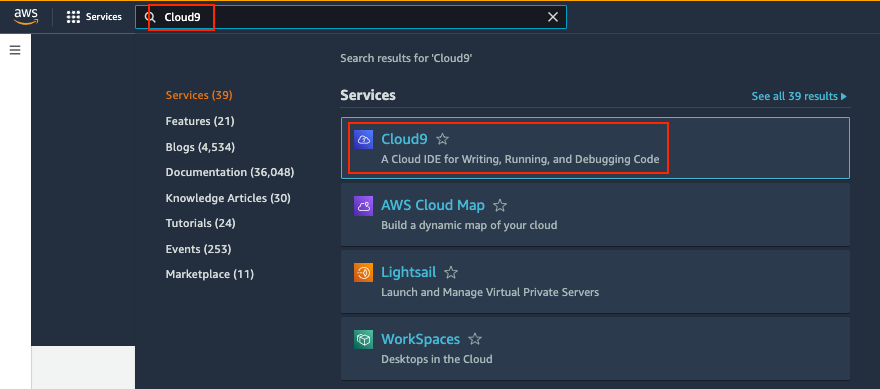
**Microservices on Amazon EKS HOL**

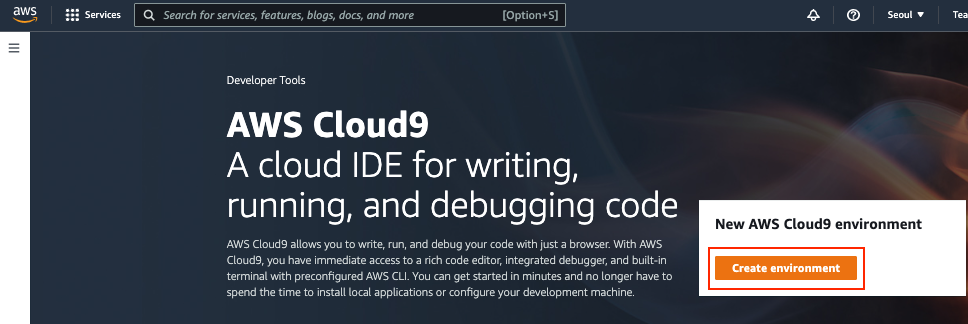
0. 해당 실습은 AWS Cloud9 를 기준으로 테스트한 내용 작성하였습니다.

1. AWS Cloud9 기동

AWS Console 의 Services 에서 Cloud9 를 입력하고, 하단에 Cloud9 를 선택한다.



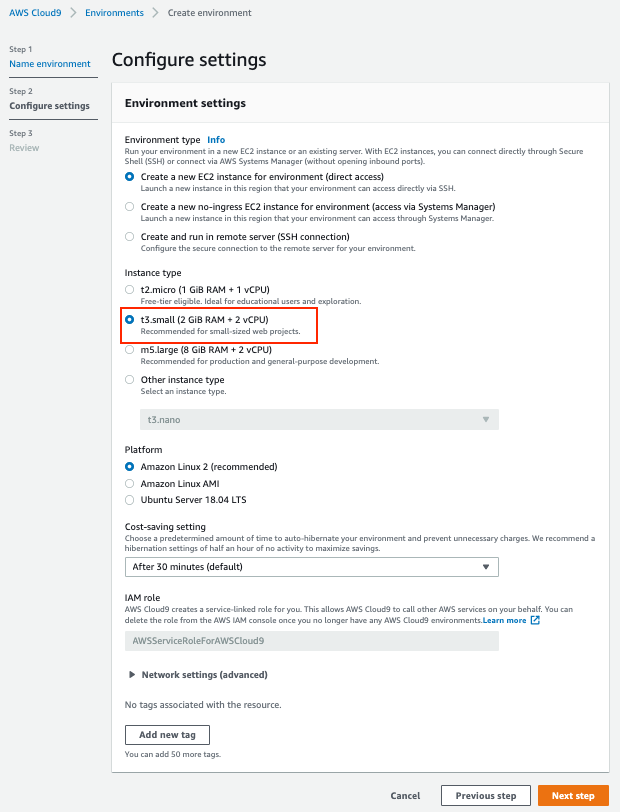
“Create environment” 를 선택한다.



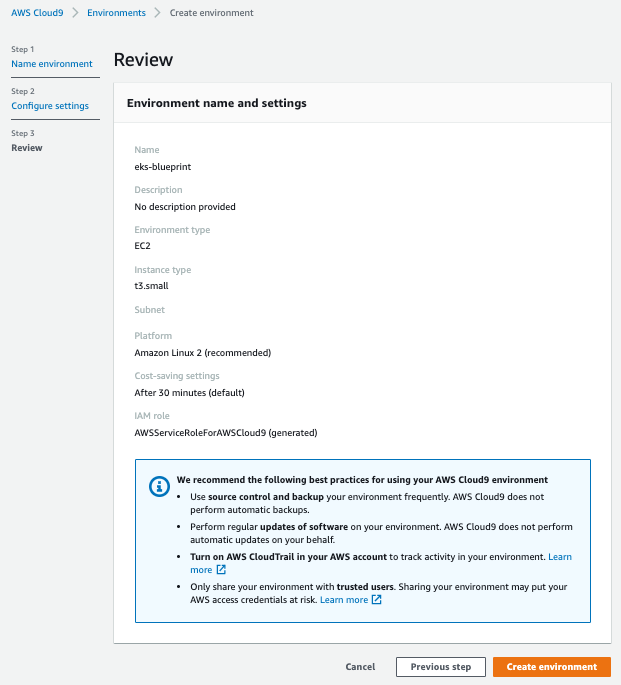
Name에 “eks-blueprint” 를 입력하고 “Next step” 버튼을 선택한다.



Instance type 에 t3.small를 선택하고, “Next step” 버튼을 선택한다.



다음 화면에서는 리뷰를 하고 문제가 없으면 “Create environment”를 선택한다.



2. 실습 환경 설정

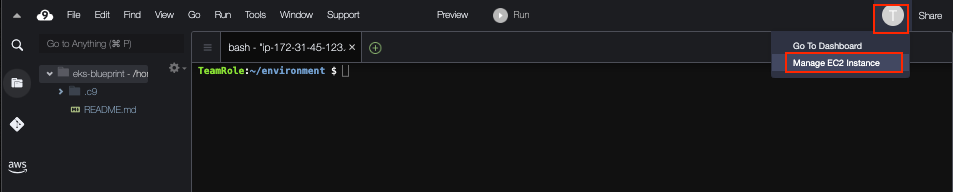
- IAM Role 생성 : 아래 URL 를 클릭하여, IAM Role 를 생성한다.

<https://console.aws.amazon.com/iam/home#/roles$new?step=review&commonUseCase=EC2%2BEC2&selectedUseCase=EC2&policies=arn:aws:iam::aws:policy%2FAdministratorAccess&roleName=eks-blueprints-for-terraform-workshop-admin>

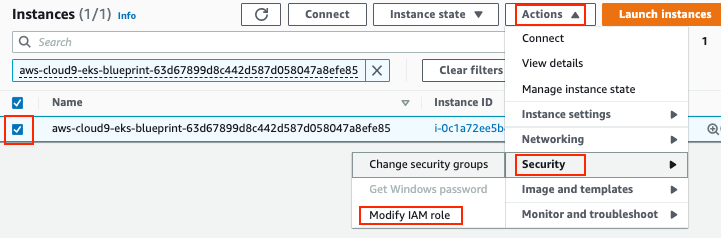
- Cloud9 에 IAM Role 할당하기

Cloud9 에 “T” 로 보이는 버튼을 선택하여 “Managed EC2 Instance” 를 선택한다.

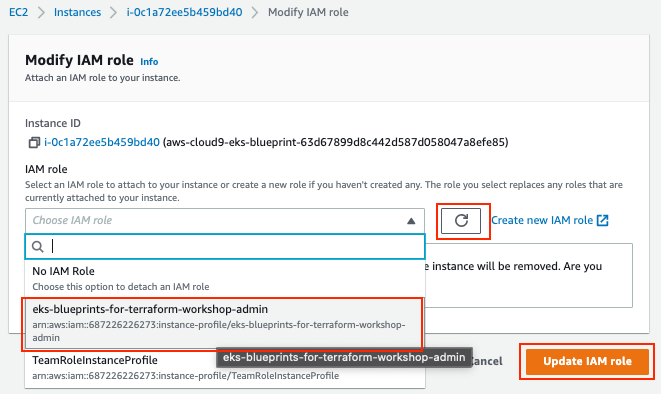
(“T”는 현재 접속하고 있는 IAM Role의 맨 앞자리 입니다.)



EC2 인스턴스를 선택한 상태에서, 위의 Action 버튼을 클릭하고, Security 항목에 “Modify IAM role” 를 선택한다.

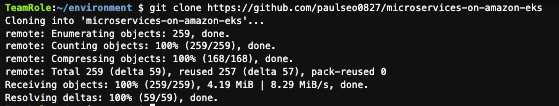


위에서 생성한 “eks-blueprints-for-terraform-workshop-admin” 를 선택하고, “Update IAM role”를 클릭한다.



3. 실습에 사용할 소스 repository clone

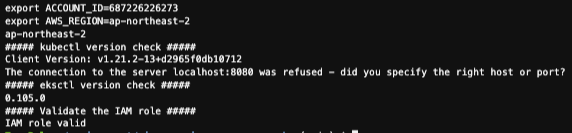
git clone https://github.com/paulseo0827/microservices-on-amazon-eks



4. 작업에 필요한 툴(kubectl, eksctl, aws-cli등) 설치와 필요한 설정 작업들을 한다.

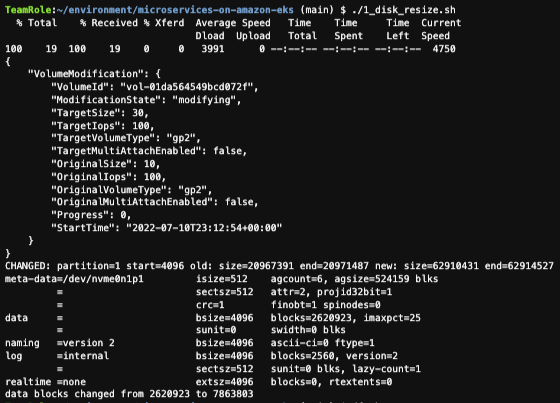
cd microservices-on-amazon-eks

./0\_install.sh



4. Cloud9 의 디스크를 30G 증설한다.

./1\_disk\_resize.sh



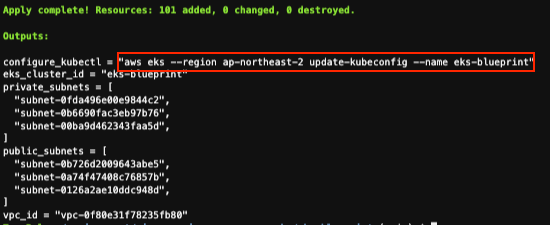
5. Amazon EKS Blueprint for Terraform를 이용하여 Amazon EKS 를 생성한다.

cd eks-blueprint

terraform init

terraform plan

terraform apply --auto-approve

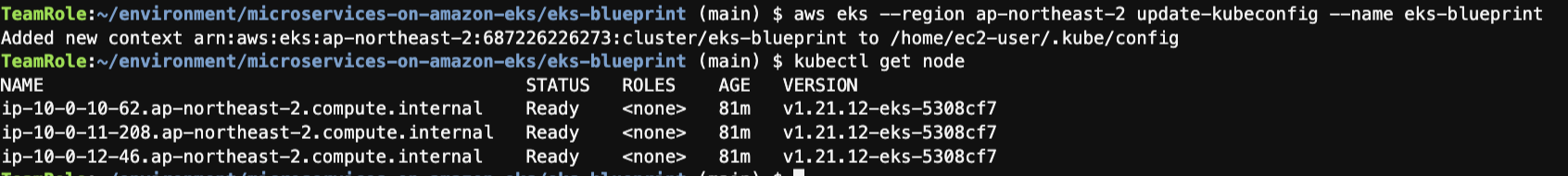


작업 소요시간은 평균 30분 내외 소요됩니다.

위 그림에 빨간색 박스로 표시한 텍스트를 그대로 복사 및 붙여넣기 하여서 명령어를 수행하면 kubectl 툴을 이용해서 Amazon EKS에 접속이 가능합니다.

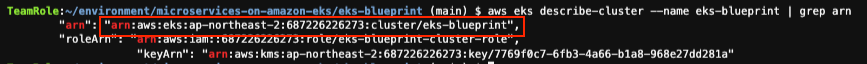
aws eks --region ap-northeast-2 update-kubeconfig --name eks-blueprint

kubectl get node



6. CDK 로 Amazon EKS 에 어플리케이션 배포에 필요한 CodeCommit, CodeBuild, ECR 를 생성할 것입니다. CDK 코드에서 Amazon EKS 의 arn 주소를 변경합니다.

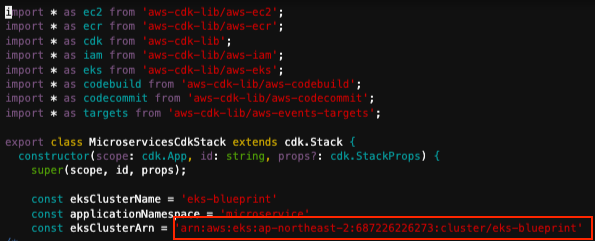
aws eks describe-cluster --name eks-blueprint | grep arn



위 이미지에 빨간색 박스로 표시한 내용을 복사합니다.

cd ~/environment/microservices-on-amazon-eks/microservices-cdk/

vi lib/microservices-cdk-stack.ts

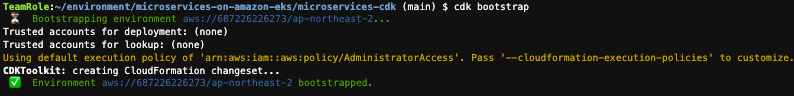


위 이미지에 빨간색 박스로 표시한 내용을 위에서 복사한 값으로 변경하고 저장을 합니다.

7. CDK로 어플리케이션 배포에 필요한 자원을 배포합니다.

npm install

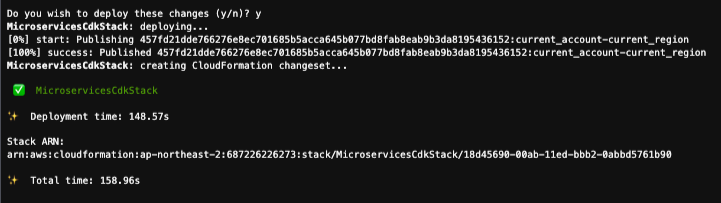
cdk bootstrap



cdk deploy

아래와 같이 메시지가 나오면 y 를 입력

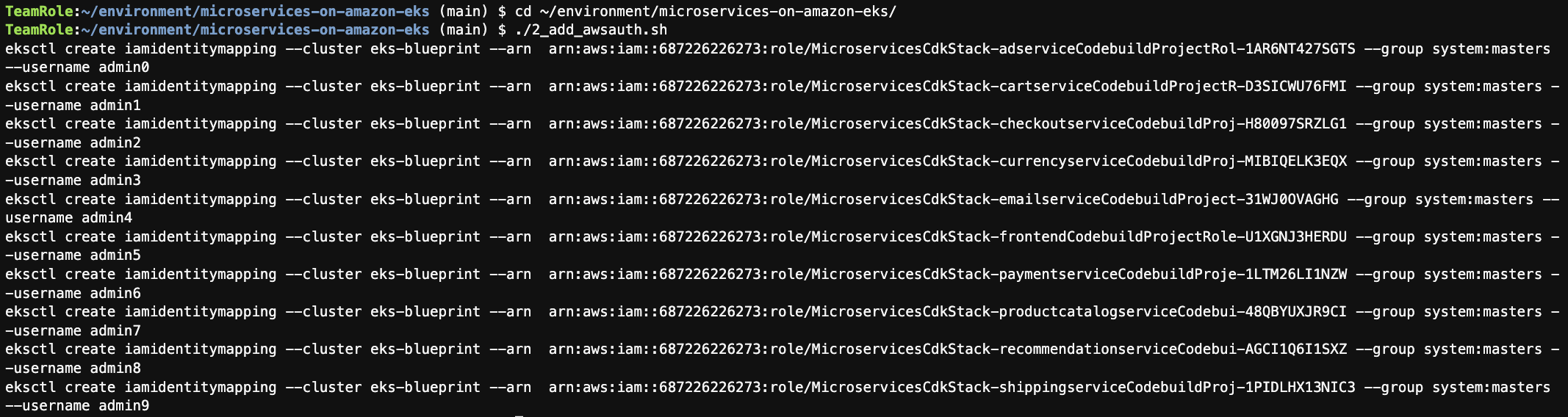


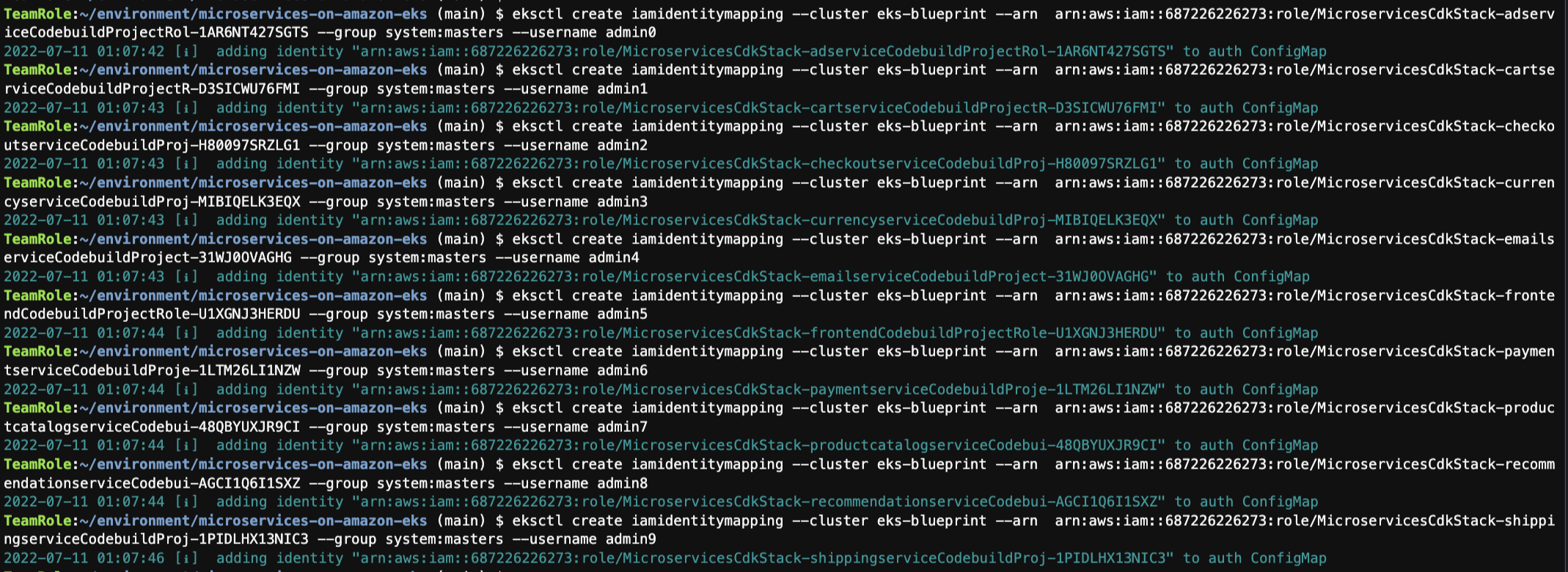


8. CodeBuild 에서 배포를 할 수 있도록 Amazon EKS aws-auth configmap 에 해당 IAM Role 에 권한을 부여합니다. “2\_add\_awsauth.sh” 를 수행하였을 때, 나온 명령어들을 그대로 복사 & 붙여넣기하여 수행합니다.

cd ~/environment/microservices-on-amazon-eks/

./2\_add\_awsauth.sh





9. Amazon EKS 에 어플리케이션 배포를 위한 namespaces 를 생성합니다.

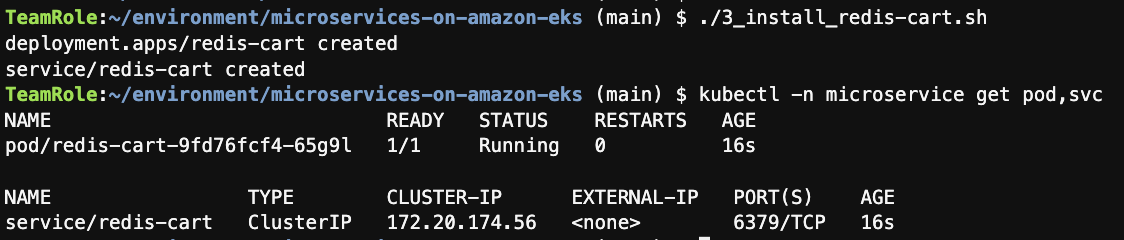
kubectl create ns microservice



10. Redis 를 EKS 에 설치하기 위해서, “3\_install\_redis-cart.sh” 를 수행합니다.

./3\_install\_redis-cart.sh

kubectl -n microservice get pod,svc



11. CodeCommit 관련 git config 수정, 아래 명령어 수행 후, 하단 박스와 같게 데이터를 입력합니다.

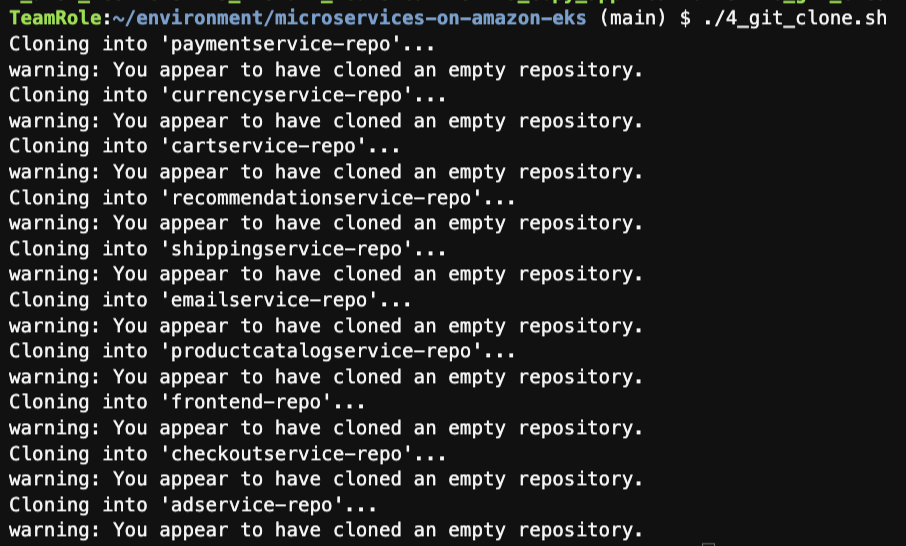
git config --global --edit

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

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

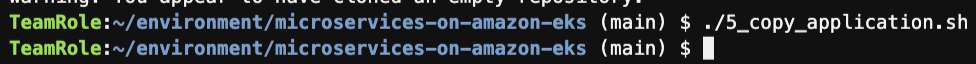
12. CDK로 생성한 CodeCommit Repository 10개에 clone 작업을 위해서, “4\_git\_clone.sh” 명령어를 수행합니다.

./4\_git\_clone.sh



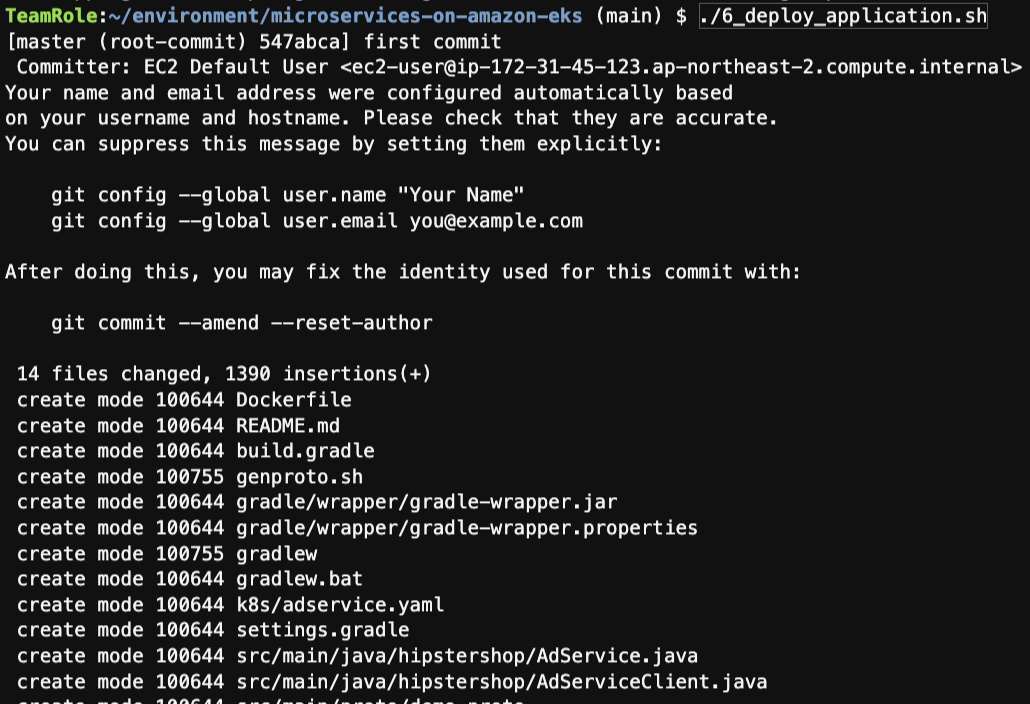
13. src 에 있는 어플리케이션 소스들을 CodeCommit 으로 복사하기 위해서, “5\_copy\_application.sh” 명령어를 수행합니다.

./5\_copy\_application.sh

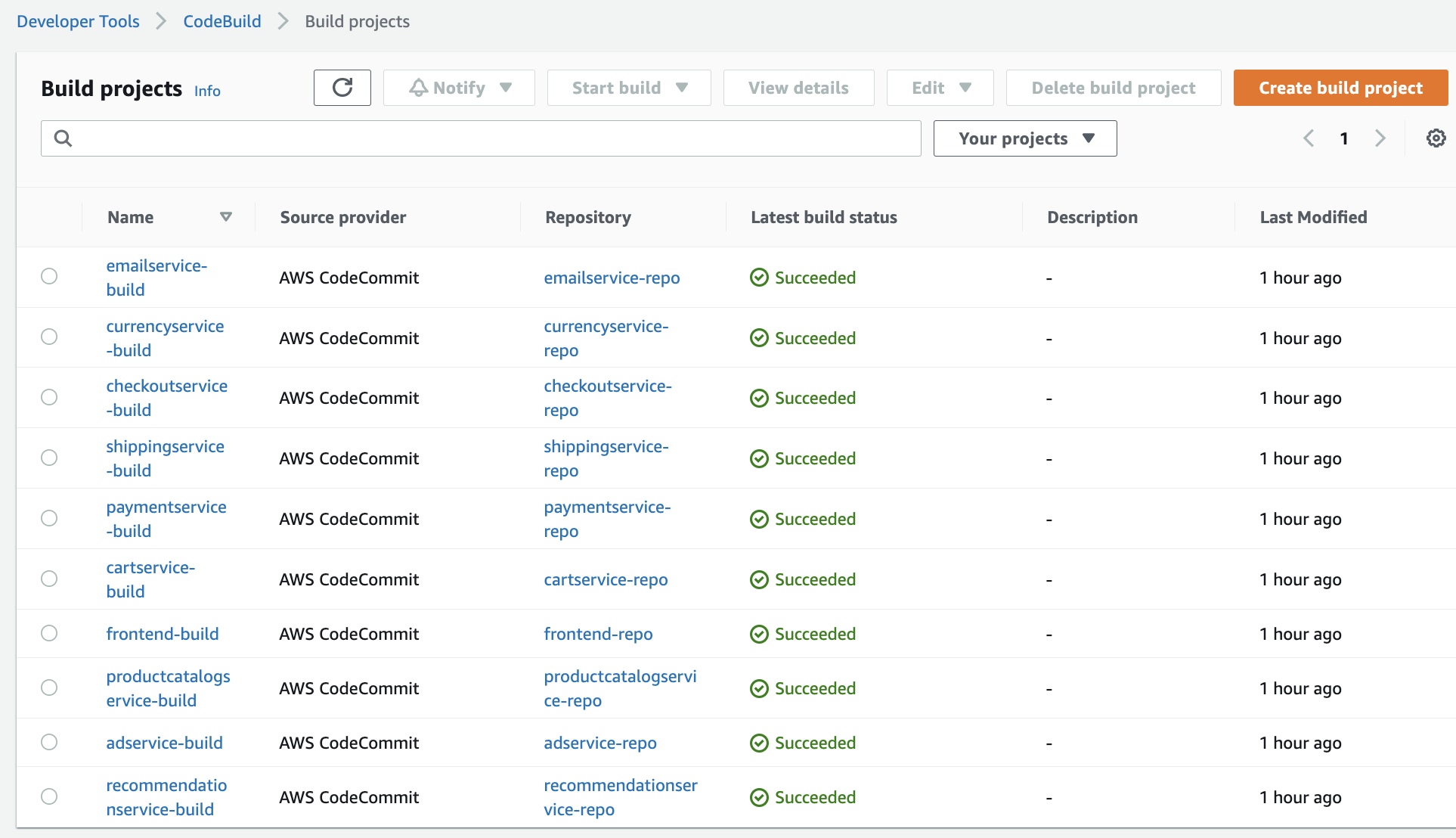


14. CodeCommit 디렉토리로 복사한 어플리케이션 소스를 commit 하기 위해서, “./6\_deploy\_application.sh” 명령어를 수행합니다.

./6\_deploy\_application.sh



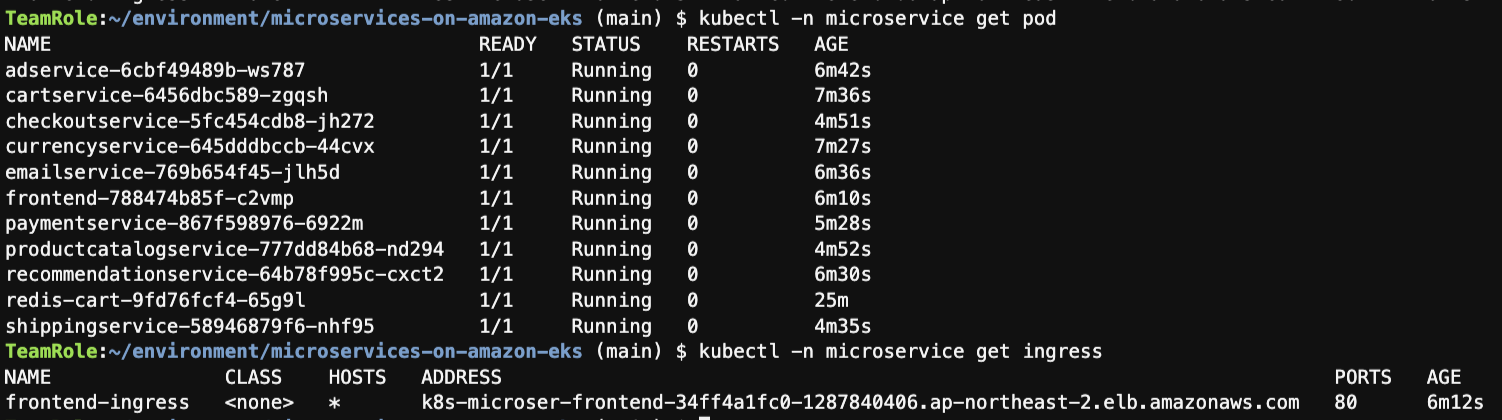
15. CodeBuild Console 에 접속하여 어플리케이션 10개의 배포가 정상적으로 성공했는지 확인을 합니다.



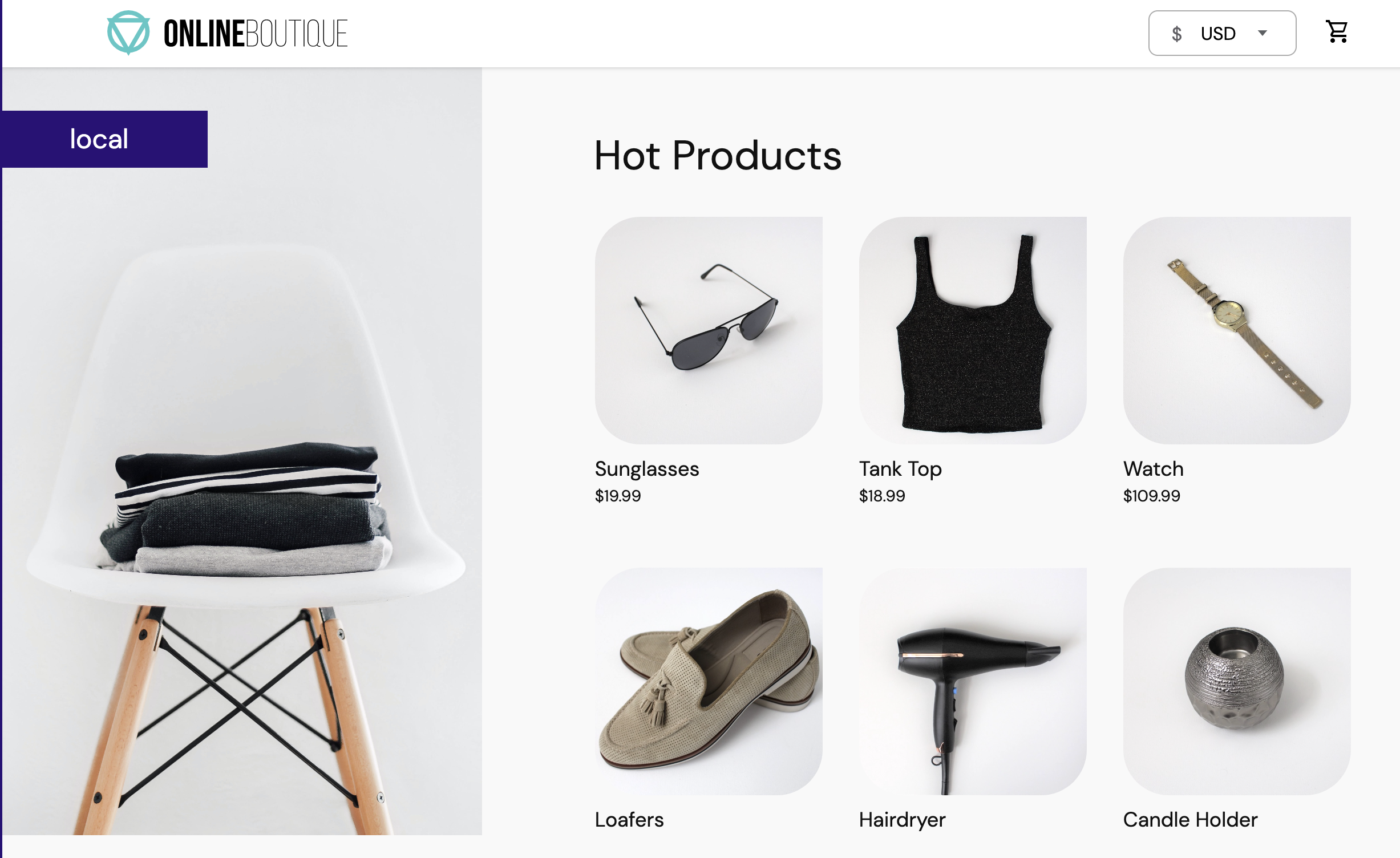
16. kubectl 명령어로 pod 11개가 정상적으로 기동되어 있는지 확인하고, ALB Ingress 도 정상적으로 올라왔는지 확인합니다.

kubectl -n microservice get pod

kubectl -n microservice get ingress

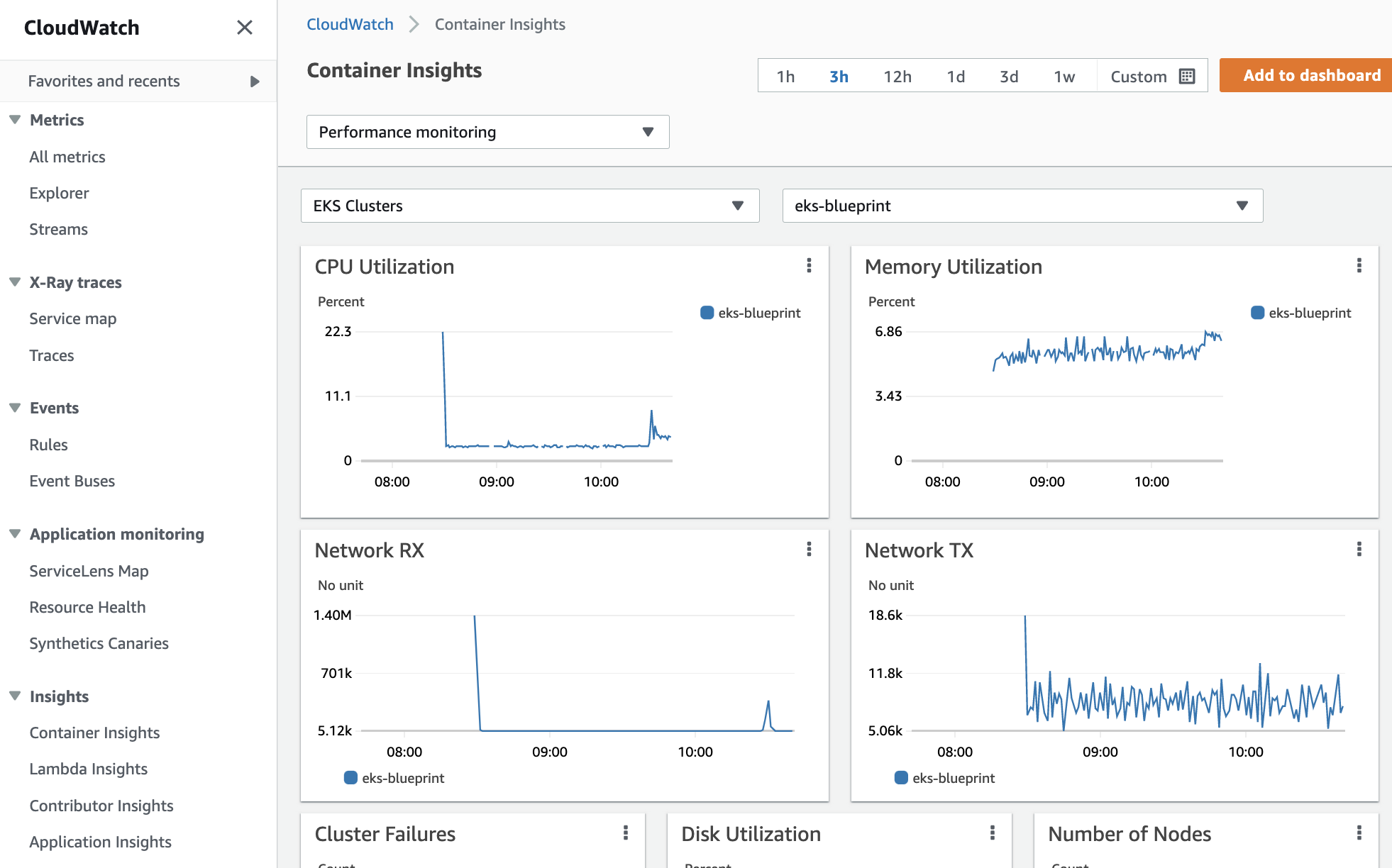


17. 위에 ingress 에서 나온 Address 의 ELB 주소를 웹브라우저에 입력하면 하단처럼 마이크로서비스로 구성된 쇼핑몰을 확인할 수 있습니다.



18. CloudWatch Console 서비스에서, 왼쪽에 “Container Insights” 를 선택한 다음에 “Performance Monitoring” 를 선택하면, 생성한 Amazon EKS Cluster의 모니터링을 할 수 있습니다.





**여기서부터는 만들었던 자원을 삭제하는 단계 입니다.**

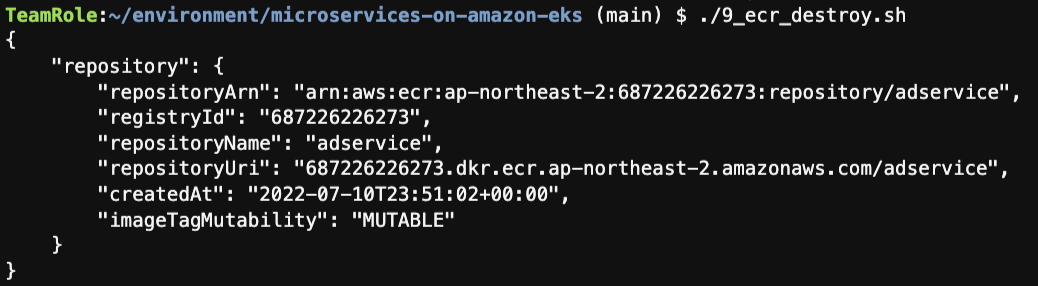
19. ALB Ingress 를 삭제합니다.

kubectl -n microservice delete ingress frontend-ingress



20. Container Image 가 저장 되어있는 Amazon ECR Repository삭제를 위해서 , “9\_ecr\_destroy.sh” 명령어를 수행합니다.

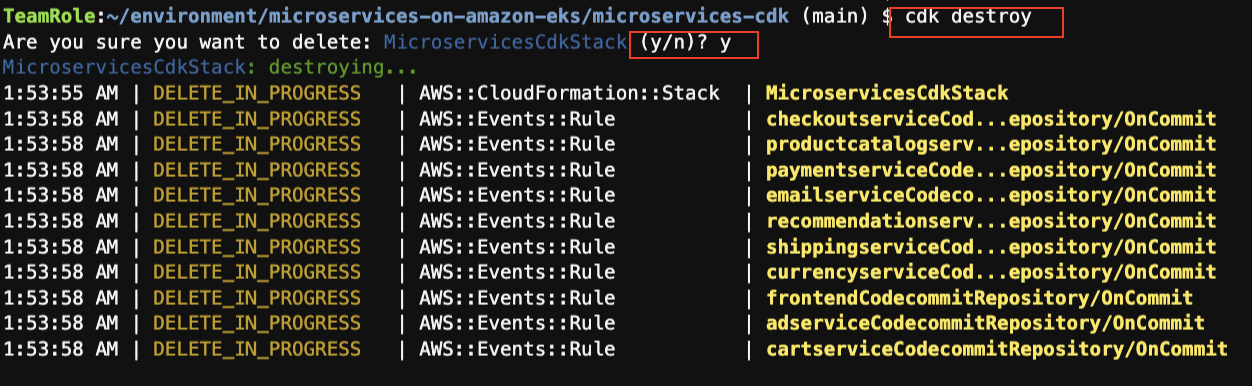
./9\_ecr\_destroy.sh



21. CDK에서 생성한 CodeCommit 과 CodeBuild를 삭제합니다.

cd ~/environment/microservices-on-amazon-eks/microservices-cdk/

cdk destroy

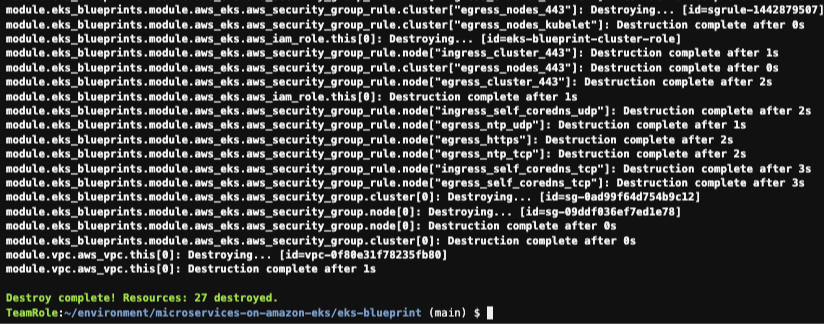


22. Amazon EKS 를 삭제합니다.

cd ~/environment/microservices-on-amazon-eks/eks-blueprint/

terraform destroy

해당 자원이 한번에 삭제가 안될 수 있습니다. 여러 번 “terraform destroy” 명령어를 수행하시면 됩니다.



23. Cloud9 Console 에 접속하셔서, 생성한 Cloud9 를 삭제하시면 됩니다. HOL에 생성한 자원을 모두 삭제 완료하였습니다.

