Amazon EKS Workshop

201904 김진웅

발표자

김진웅 @ddiiwoong

- Platform Architect @SK C&C
- Interested in Kubernetes and Serverless(FaaS), DevOps, SRE, ML/DL
- ~18년 OSS기반 FaaS 플랫폼 (https://skcloud.github.io/zaction/) 스크럼마스터, SRE
- 19년~ BaaS(BlockChain) 기반 데이터레이크 플랫폼 설계/구축

EKS

- Amazon Elastic Container Service for Kubernetes
  - Fully Managed Kubernetes Services
  - https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/what-is-eks.html
  - 기존 서비스통합
    - Registry : Amazon ECR
    - LoadBalancer : Elastic Load Balancing
    - Auth : 인증용 IAM
    - CNI : Amazon VPC

Kubernetes 기능	OSS, 상용	AWS
Registry	Dockerhub, Harbor	ECR
LoadBalancer	nginx, traefik	ELB(ALB)
Cluster 인증	dex, keycloak	aws-iam-authenticator
CNI(Network Plugin)	Calico, weave, Cilium	VPC

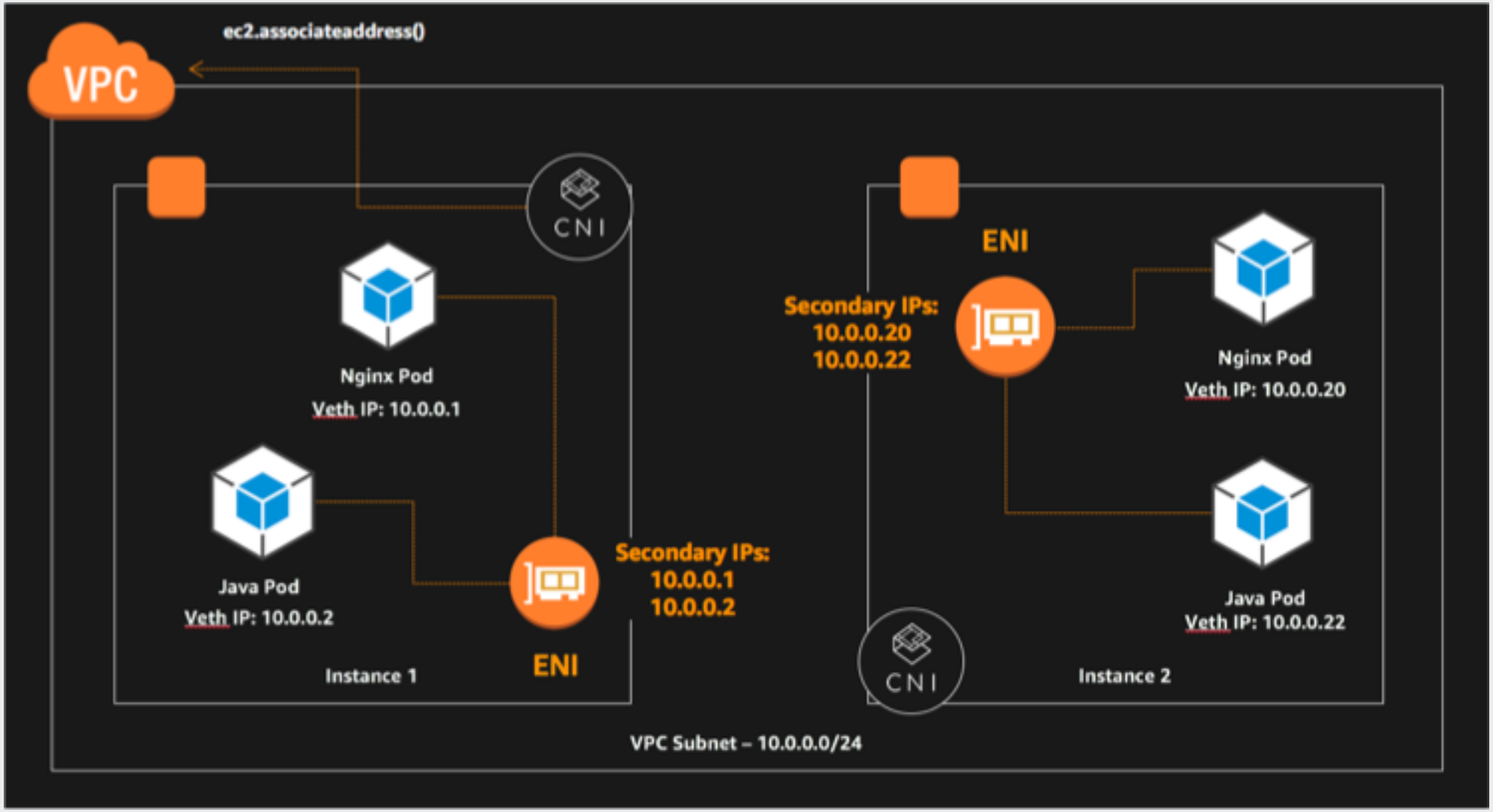
EKS

- **Kops(EC2기반 Kubernetes배포)와 차이 (19년 4월 4일 기준)**

구분	KOPS	EKS
버전	1.14.0	1.12.7
권한관리	RBAC	IAM(aws-iam-authenticator)
관리형태	Master, ETCD, DNS, Scheduler 직접관리	완전관리형 (0.2USD/hr*Cluster)
네트워킹	Flannel(CNI 대부분 지원)	VPC(Calico, Cilium 등)
상태	S3 Bucket	완전관리형
API권한	모든권한	KubeAPI, Kubelet 수정불가
Taint, Labeling	Instance 단위 가능	미지원

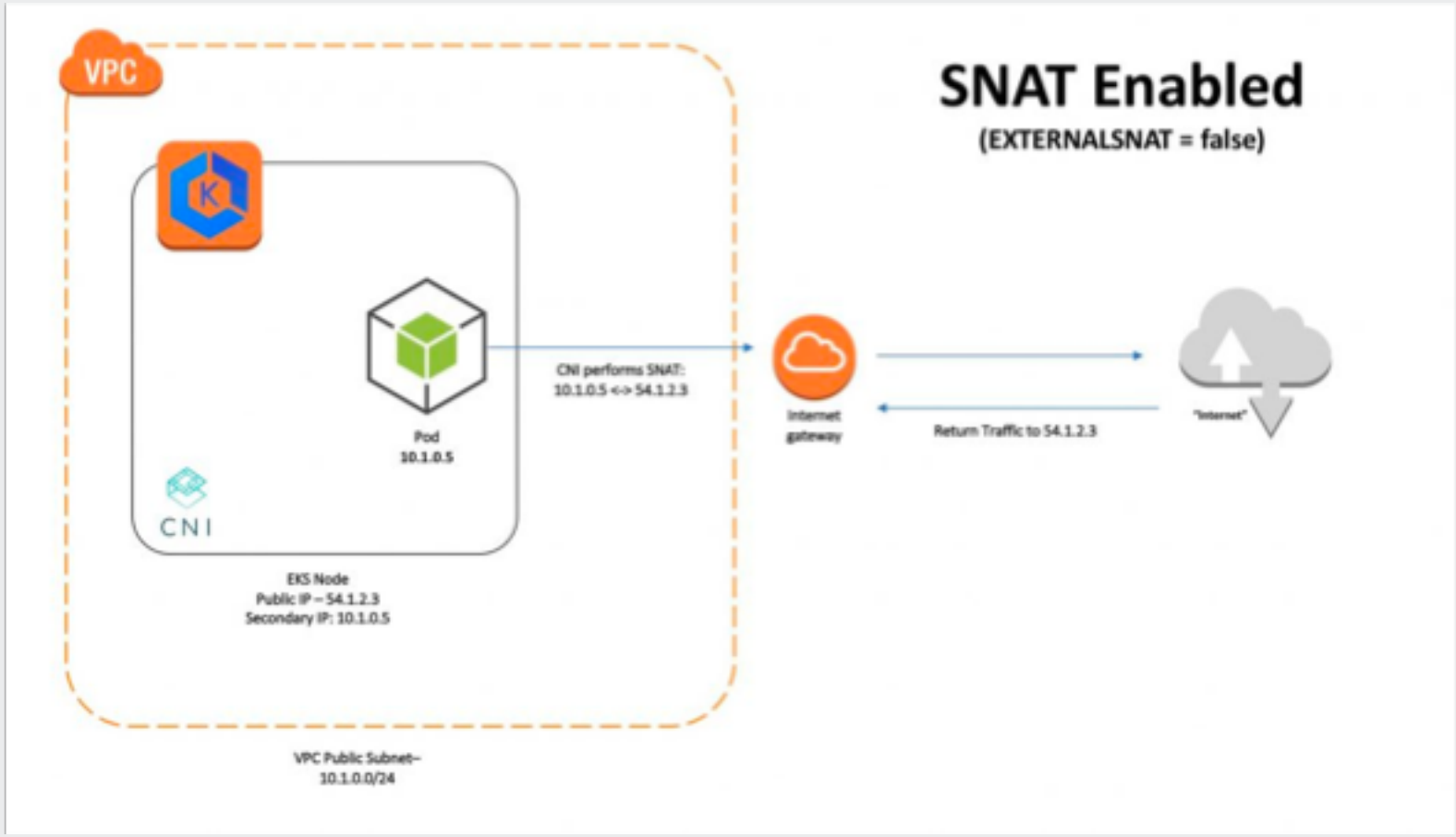
EKS Networking

- CNI (Container Network Interface)



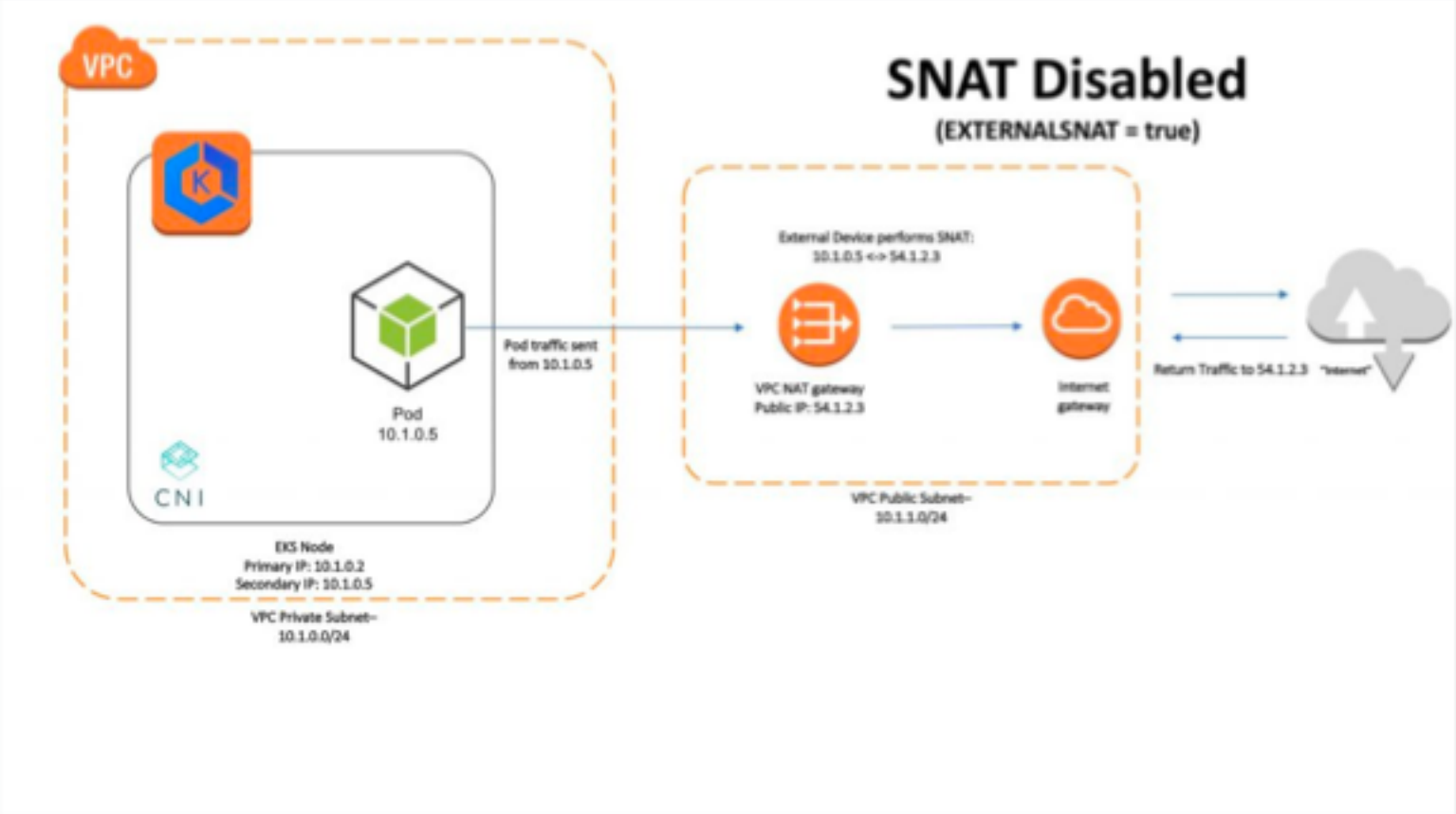
EKS Networking

- SNAT Enabled
  - Default
  - CNI가 SNAT를 수행해서 Public 주소로 외부와 통신



EKS Networking

- SNAT Disabled
  - VPC Peering, Transit VPC, Direct Connect의 경우
  - NAT를 외부에서 처리되도록 함 (https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/external-snat.html)



EKS Networking

- 3rd Party CNI (Container Network Interface)

- Calico (L3 Layer)
  - https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/calico.html
  - 테넌트 별(dev,stage,prod) 네트워크 격리 및 세분화 구현

- Cilium (L7 Layer)
  - http://docs.cilium.io/en/v1.4/gettingstarted/k8s-install-eks/?highlight=eks
  - Kafka, Elasticsearch 보안
  - EKS – Instance 간 SecurityGroup 설정

EKS Workshop

- https://eksworkshop.com/
  Kubernete를 처음접하는 유저를 위한 기본 개념과 아키텍처, 그리고 VPC, ALB를 활용하여 EKS에 대한 설치, 구성, 데모앱 배포 등을 해볼수 있는 튜토리얼 사이트

- AWSKRUG에서 한글화 작업도 진행중
  - https://awskrug.github.io/eks-workshop/deploy/

EKS Workshop 준비사항

- AWS Account
  - Free Tire 불가 (EKS)
  - Credit 필요

- IAM
  - Terraform과 다른 권한 필요
  - https://github.com/ddiiwoong/eksworkshop/blob/master/iam_for_eksworkshop.json

- kubectl, aws-iam-authenticator
  - kubectl : kubernetes CLI
  - aws-iam-authenticator : AWS IAM Authenticator CLI

- eksctl
  - Cloudformation 사용 (15-20분 소요)

EKS Workshop 준비사항

- IAM 설정
  - EKS workshop은 기본적으로 Administrator 권한으로 진행됨
  - 주요 Action : iam, eks(*), ec2, autoscaling
  - User 권한은 아래 링크를 참고
    - https://github.com/ddiiwoong/eksworkshop/blob/master/iam_for_eksworkshop.json
    - 추가적인 확인 필요
  - Terraform eks IAM
    - https://github.com/terraform-aws-modules/terraform-aws-eks/blob/master/examples/eks_test_fixture/README.md

EKS Workshop 준비사항

- Kubectl : Kubernetes CLI
- Linux

```
$ mkdir –p ~/.kube

$ sudo curl --silent --location -o /usr/local/bin/kubectl "https://amazon-eks.s3-us-west-2.amazonaws.com/1.11.5/2018-12-06/bin/linux/amd64/kubectl" $ sudo chmod +x /usr/local/bin/kubectl
```

- Mac

```
$ brew install kubernetes-cli
```

- PowerShell (PATH설정 추가필요)

```
curl -o kubectl.exe https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/windows/amd64/kubectl.exe
```

- 설치 확인

```
$ kubectl version --short --client
```

EKS Workshop 준비사항

- **aws-iam-authenticator**
- **Linux**: https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator
- **MacOS**: https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/darwin/amd64/aws-iam-authenticator
- **Windows**: https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/windows/amd64/aws-iam-authenticator.exe

- **MacOS/Linux 설정**

```
$ curl -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/darwin/amd64/aws-iam-authenticator
$ chmod +x ./aws-iam-authenticator
$ cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH
```

```
$ echo 'export PATH=$HOME/bin:$PATH' >> ~/.bash_profile
$ echo 'export PATH=$HOME/bin:$PATH' >> ~/.zshrc
```

- **설치 확인**

```
$ aws-iam-authenticator help
```

EKS Workshop 준비사항

- Eksctl : simple CLI tool for creating EKS Cluster
- Linux

```
$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/download/latest_release/eksctl_$(uname -s)_amd64.tar.gz"
| tar xz -C /tmp
$ sudo mv -v /tmp/eksctl /usr/local/bin
```

- Mac (homebrew)

```
$ brew tap weaveworks/tap
$ brew install weaveworks/tap/eksctl
```

- Powershell

```
$ chocolatey install eksctl
```

EKS Workshop 준비사항

- CLI 자격증명 구성

- ~/.aws/credentials

```
[default]
aws_access_key_id={EXAMPLE}
aws_secret_access_key={EXAMPLEKEY}
```

- ~/.aws/config

```
[default]
region=ap-northeast-2
output=json
```

EKS 배포 (eksctl)

- **Eksctl 배포**

(ap-northeast-2 기준 약 15-20분이 소요)

```
$ eksctl create cluster --name=eksworkshop-eksctl --nodes=3 --node-ami=auto
```

- 클러스터명은 자동생성, –name 옵션으로 지정가능 (eksworkshop-eksctl)
- CloudFormation : eksctl-{$Cluster_Name}-cluster
- m5.large * 2 instances (EKS Instance Type NodeInstanceType.AllowedValues 참고)
- Default AMI : AWS EKS AMI (custom EKS AMI 가능 - Packer활용 가능)
- Default Region : us-west-2
- dedicated VPC : 192.168.0.0/16
- kubernetes version : 1.11.x (EKS Version 참고)
- StorageClass : gp2 (AWS EBS 참고)
- CNI : Amazon VPC
- Node Autoscaler : –node-min, –node-max Auto Scaling 설정
- nodegroup : worker가 포함되는 group
- kubeconfig : ~/.kube/config 로 통합됨

EKS 배포 (eksctl config)

- Config File 사용
  https://github.com/weaveworks/eksctl/tree/master/examples
- VPC subnet정보 및 AutoScaling, AZ(availabilityZones)설정,
  nodegroup 관리, node Instance에 preBootstrapCommand 등 사전활용 가능

```
$ eksctl create cluster -f example.yaml
```

Deploy K8s Dashboard

- Kubernetes 대시보드 배포

```
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommended/kubernetes-dashboard.yaml
```

- Expose (백그라운드, 모든 인터페이스 접속가능)

```
$ kubectl proxy --port=8080 --address='0.0.0.0' --disable-filter=true &
```

- aws-iam-authenticator를 통해 해당 클러스터 token 확인

```
$ aws-iam-authenticator token -i eksworkshop-eksctl --token-only
```

- 대시보드 접속
  http://localhost:8080/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login

Deploy Microservices

- Git Clone (Sock Shop)

```
$ git clone https://github.com/microservices-demo/microservices-demo.git sockshop
$ cd sockshop/deploy/kubernetes
```

- LoadBalancer 수정

```
$ sed -i 's/NodePort/LoadBalancer/g' complete-demo.yaml
```

- Namespace 생성 및 sock-shop 배포

```
$ kubectl create namespace sock-shop
$ kubectl apply -f complete-demo.yaml
$ kubectl get svc –n sock-shop
```

- front-end EXTERNAL-IP 접속
  - DNS 배포시간으로 약 5분후 접속

EKS 클러스터 및 삭제

- EXTERNAL-IP값과 연결된 모든 서비스를 삭제

```
$ kubectl get svc –n sock-shop
$ kubectl delete svc front-end
```

- ALB삭제
  - EC2 - Load Balancer

- EKS 클러스터 삭제

```
$ eksctl delete cluster --name=eksworkshop-eksctl
```

정리

- eksctl 활용 단순, 명령 한줄로 배포
- 기본 VPC, KUBECONFIG 자동생성, aws-auth 자동설정(ConfigMap)
- Terraform 동시 활용 방안 (고민)
  - aws-auth관리 – eksctl
  - VPC, IAM관리 – terraform
- 서비스 제한 사항
  - 계정, 리전당 최대 EKS Cluster 수 : 50
  - 클러스터당 Control plane 최대 Security그룹 수 : 5
- 다른 Managed Kubernetes와의 비교
  - https://docs.google.com/spreadsheets/d/1U0x4-NQegEPGM7eVTKJemhkPy18LWuHW5vX8uZzqzYo/edit#gid=0
- 추가 Resources
  - Spot Instances 활용 (https://eksworkshop.com/spotworkers/)
  - VPC확장 (https://eksworkshop.com/advanced-networking/)
  - Cilium으로 micro-segmentation security 구현 – Kafka, ELK Stack 등 (http://docs.cilium.io/en/stable/gettingstarted/k8s-install-eks/)

문의

- 페이스북, 트위터 : @ddiiwoong
- 블로그 : ddii.dev
- 이메일 : ddiiwoong@gmail.com