

쿠버네티스를 활용한 멀티클라우드 도입과 운영전략

- AWS, Azure, GCP 비교와 실습

2020년 3월
아인인텔리전스
권재원, Ph.D.

Lab: EKS

AWS - EKS

eksctl 시작하기

https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/getting-started-eksctl.html

```
curl --silent --location
```

```
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

```
sudo mv -v /tmp/eksctl /usr/local/bin
```

```
eksctl version
```

```
eksctl completion bash >> ~/.bash_completion
```

```
. /etc/profile.d/bash_completion.sh
```

```
. ~/.bash_completion
```

환경 설정 - eksctl

Cluster 만들기

* 생성되는데 15분 가량 걸림

```
eksctl create cluster --name guestbook --version 1.14 --region ap-northeast-2 --nodegroup-name standard-workers --node-type t3.small --nodes 3 --nodes-min 1 --nodes-max 4 --managed
```

```
chaiwon@surface-Laptop:~/aws_eks$ eksctl create cluster \  
> --name guestbook \  
> --version 1.14 \  
> --region ap-northeast-2 \  
> --nodegroup-name standard-workers \  
> --node-type t3.small \  
> --nodes 3 \  
> --nodes-min 1 \  
> --nodes-max 4 \  
> --managed  
[■] eksctl version 0.15.0  
[■] using region ap-northeast-2  
[■] setting availability zones to [ap-northeast-2b ap-northeast-2c ap-northeast-2a]  
[■] subnets for ap-northeast-2b - public:192.168.0.0/19 private:192.168.96.0/19  
[■] subnets for ap-northeast-2c - public:192.168.32.0/19 private:192.168.128.0/19  
[■] subnets for ap-northeast-2a - public:192.168.64.0/19 private:192.168.160.0/19  
[■] using Kubernetes version 1.14  
[■] creating EKS cluster "guestbook" in "ap-northeast-2" region with managed nodes  
[■] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup  
[■] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-northeast-2 --cluster=guestbook'  
[■] CloudWatch logging will not be enabled for cluster "guestbook" in "ap-northeast-2"  
[■] you can enable it with 'eksctl utils update-cluster-logging --region=ap-northeast-2 --cluster=guestbook'  
[■] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "guestbook" in "ap-northeast-2"  
[■] 2 sequential tasks: { create cluster control plane "guestbook", create managed nodegroup "standard-workers" }  
[■] building cluster stack "eksctl-guestbook-cluster"  
[■] deploying stack "eksctl-guestbook-cluster"
```

환경 설정 - eksctl

Cluster 만들기

Amazon Container Service

Amazon ECS

클러스터

작업 정의

Amazon EKS

클러스터

Amazon ECR

리포지토리

EKS > 클러스터

새 Kubernetes 버전이 1개의 클러스터에 대해 사용 가능합니다.

클러스터 (1)

이름을 기준으로 클러스터 찾기

	클러스터 이름	Kubernetes 버전	상태
<input type="radio"/>	guestbook	1.14 지금 업데이트	<input checked="" type="checkbox"/> 활성화

참고 - aws-iam-authenticator 설치

https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/install-aws-iam-authenticator.html

- `curl -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/aws-iam-authenticator`
- `chmod +x ./aws-iam-authenticator`
- `mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$PATH:$HOME/bin`
- `echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc`
- `aws-iam-authenticator help`

참고 - Amazon EKS에 대한 kubeconfig 생성

https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/create-kubeconfig.html

- aws eks get-token 명령과 kubectl을 클러스터 인증에 사용
 - aws sts get-caller-identity
- AWS CLI update-kubeconfig 명령을 사용하여 클러스터에 대한 kubeconfig를 생성하거나 업데이트
 - aws eks --region <region-code> update-kubeconfig --name <cluster_name>
- 구성을 테스트
 - kubectl get svc

게스트 북 애플리케이션

https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/eks-guestbook.html

<https://github.com/kubernetes/examples/blob/master/guestbook-go/README.md>

- Redis 마스터 복제 컨트롤러를 생성
 - kubectl apply -f
<https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-controller.json>
- Redis 마스터 서비스를 생성
 - kubectl apply -f
<https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json>
- Redis 슬레이브 복제 컨트롤러를 생성
 - kubectl apply -f
<https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-controller.json>

게스트 북 애플리케이션 (계속)

- Redis 슬레이브 서비스를 생성
 - `kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-service.json`
- 게스트 북 복제 컨트롤러를 생성
 - `kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-controller.json`
- 게스트 북 서비스를 생성
 - `kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-service.json`
- guestbook 서비스의 External IP 열이 채워질 때까지 클러스터에서 서비스를 쿼리 (*IP 주소를 사용할 수 있을 때까지 몇 분이 걸릴 수 있음)
 - `kubectl get services -o wide`

게스트 북 애플리케이션 (계속)

- 외부 IP 주소가 이용 가능한 이후 웹 브라우저를 포트 3000의 해당 주소로 가리켜 게스트 북을 확인
 - 예) `http://a7a95c2b9e69711e7b1a3022fdcfd2e-1985673473.region.elb.amazonaws.com:3000`

Guestbook

`http://a671944b36eba11eab29a0282ce1cd76-1738504911.ap-northeast-2.elb.amazonaws.com:3000/`
[/env](#) [/info](#)

- 게스트 북 애플리케이션을 정리 (모든 서비스 및 복제 컨트롤러를 삭제)
 - `kubectl delete rc/redis-master rc/redis-slave rc/guestbook svc/redis-master svc/redis-slave svc/guestbook`
 - `aws eks delete-nodegroup --cluster-name guestbook --nodegroup-name standard-workers`
 - `aws eks delete-cluster --name guestbook`