6. DATA MANAGEMENT & DATABASES

NEXT OnAir 2020
베스핀글로벌 세션브리핑

Connecting to Cloud SQL from Kubernetes

쿠버네티스 엔진에서 클라우드 SQL 연결하기

Kurtis Van Gent
Developer Programs Engineer, Google Cloud

#Cloud SQL #Kubernetes #Database Modernization

BESPIN GLOBAL

Week6.Data Management & Databases

Connecting to Cloud SQL from Kubernetes

구글 쿠버네티스 엔진에서 클라우드 SQL 연결하기

이번 Google Cloud Next Onair를 시작하기 앞서 클라우드 SQL, 컨테이너, 쿠버네티스에 대한 기본적인 지식은 필수입니다.

클라우드 SQL 연결하기

SQL에 연결하기 위해서는 2가지를 고려해야합니다.

✔ 경로(Path)

Public IP

- 공용 인터넷에서 액세스 할 수 있는 IPv4 주소
- 외부에서 접근을 제한하기 위해 인증 필수
- GCP 외부에서도 연결할 수 있다는 장점
- 레이턴시가 높을 수 있다는 단점

Private IP

- 프로젝트 내 Virtual Private Cloud(VPC)로 피어링 된 IPv4 주소
- 인증이 필요없지만 VPC 접근 권한은 있어야 함
- 인터넷으로 연결하는게 아니기 때문에 레이턴시가 낮을 수 있다는 장점
- VPC 권한이 있어야된다는 단점

✔ 인증방법

Public IP

- IAM credentials를 사용하여 권한 부여 및 강력한 암호화를 제공하는 프록시

Authorized Networks

- 연결권한이 있는 IP 주소 범위를 지정
- Private IP 경우, VPC firewall 규칙을 대신 사용

자체 관리 SSL

- 연결 암호화를 위해 SSL 자체 형성

✔ 구글의 추천 옵션:

- Public IP + Cloud SQL Proxy
- Public IP + Authorized Networks + 자체 관리 SSL
- Private IP + Cloud SQL Proxy

쿠버네티스에서 클라우드 SQL 연결하기

3가지 옵션이 있습니다.

✔ Private IP를 이용한 다이렉트 연결

- Private IP는 인증 절차가 필요 없기 때문에 바로 연결 가능
- Private IP 접속을 위해서는 VPC 네이티브 구글 쿠버네티스 엔진(GKE) 클러스터 필수
- 보안에 취약함

✔ Proxy as a Service

- Private IP와 달리 비교적 복잡한 초기 설정
- 강력한 암호화 및 IAM 권한부여
- 발신 연결 암호화 제공하지만 수신 연결에 대한 암호화 부재
- 클러스터 권한 있는 모든 사람들 접근 가능(보안문제)

✔ Proxy as a sidecar(구글의 추천 옵션)

- 프록시를 바탕으로 강력한 암호화 및 IAM 권한 설정 가능 (클라이언트 사이드 포함)
- SQL 트래픽 로컬 노출 방지(발신, 수신 연결에 대한 암호화)
- 데이터베이스에 대한 각 애플리케이션의 액세스는 다른 애플리케이션에 대해 독립적이므로 복원력 우수
- 단일 장애점 방지
- 애플리케이션별로 IAM 권한 사용 가능
- 사용량에 비례하여 리소스를 소비하는 프록시의 리소스 요청 범위를 보다 정확하게 지정 가능

Proxy as a sidecar를 사용하기 위해서는 Cloud SQL 프록시에 서비스 계정을 제공하도록 GKE를 구성해야합니다. (2 options)

Options1. 워크로드 아이덴티티

- Google Kubernetes Engine을 사용하는 경우 GKE의 워크로드 아이덴티티 기능을 사용하는 것이 좋습니다. 이 방법을
- 사용하면 Kubernetes 서비스 계정(KSA)을 Google 서비스 계정(GSA)에 결합할 수 있습니다.
- Secret management가 없다는 장점
- GKE 클러스터 사용 필수

Options1. 워크로드 아이덴티티 설정 방법

1. 클러스터에 워크로드 아이덴티티 사용 설정
2. 애플리케이션 kubectl apply -f service-account.yaml의 KSA 형성

```
apiVersion: v1
kind: ServiceAccount
metadata:
    name: <YOUR-KSA-NAME> # TODO(developer): replace these values
```

3. YOUR-GSA-NAME과 YOUR-KSA-NAME 간에 IAM binding 사용 설정

```
gcloud iam service-accounts add-iam-policy-binding \
  --role roles/iam.workloadIdentityUser \
  --member
"serviceAccount:<YOUR-GCP-PROJECT>.svc.id.goog[<YOUR-K8S-NAMESPACE>/<YOUR-KSA-NAME>]" \
  <YOUR-GSA-NAME>@<YOUR-GCP-PROJECT>.iam.gserviceaccount.com
```

4. YOUR-KSA-NAME에 주석을 추가하여 binding 완료 작업

```
kubectl annotate serviceaccount \
  <YOUR-KSA-NAME> \
  iam.gke.io/gcp-service-account=<YOUR-GSA-NAME>@<YOUR-GCP-PROJECT>.iam.gserviceaccount.com
```

5. 마지막으로 k8s 객체의 서비스 계정 지정

Options2. 서비스 계정 키 파일

- GKE 클러스터를 사용하지 않거나 워크로드 아이덴티티를 사용할 수 없는 경우 권장됩니다.
- Cloud SQL 프록시 Pod에 마운트
- Json 키 파일 지속적인 관리가 필요하다는 단점
- 스탠다드 k8s에서 실행 할 수 있다는 장점

Options2. 서비스 계정 키 파일 설정 방법

1. 서비스 계정 키의 사용자 인증 정보 파일 형성

```
gcloud iam service-accounts keys create ~/key.json \
  --iam-account <YOUR-SA-NAME>@project-id.iam.gserviceaccount.com
```

2. 서비스 계정 키를 k8s 보안 비밀로 변경

```
kubectl create secret generic <YOUR-SA-SECRET> \
--from-file=service_account.json=~/key.json
```

3. k8s 객체의 spec : 아래에 보안 비밀을 볼륨으로 마운트

```
volumes:
- name: <YOUR-SA-SECRET-VOLUME>
  secret:
    secretName: <YOUR-SA-SECRET>
```

4. 프록시 pod의 볼륨에 액세스

추가 리소스

Cloud SQL proxy on GitHub: https://github.com/GoogleCloudPlatform/cloudsql-proxy

Cloud SQL JDBC Socket Factory: https://github.com/GoogleCloudPlatform/cloud-sql-jdbc-socket-factory

베스핀글로벌 인사이트

온프레미스에서 가상 인프라가 등장했고 이제 추세는 클라우드입니다. 데이터에 가장 critical하고 중요한 사항은 보안이라고 할 수 있습니다. 데이터를 암호화하고 유출 방지를 위해 클라우드 기업들은 보안에 투자하며 급속도로 성장하고 있습니다.
클라우드 인프라에는 확장에 자동 대응하며 보안을 지원하기 위한 지능을 갖춘 자체 인프라 계층이 필요합니다.

쿠버네티스는 쿠버네티스 서비스를 구성하여 로드밸런싱을 정의하는 보다 편리하고 실속있는 방법을 제공합니다. 각자 상황을 고려해서 보안에 비중을 둘지 편리함에 비중을 둘지 판단하여 그에 맞는 경로를 선택해야 됩니다. 구글에서 추천해주는 Proxy as a sidecar은 데이터의 가장 중요한 보안도 뛰어나며 유연하게 설계할 수 있는 충분히 고려할만한 옵션이라고 판단됩니다.

전문적인 판단과 클라우드 도입을위해 클라우드 파트너와 함께 세밀한 로드맵을 세워 진행하는 것을 추천드립니다.