

Tencent Kubernetes Engine

FAQ

제품 문서



Copyright Notice

©2013-2023 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

목록:

FAQ

- 유지보수

 - 로그 수집

- 클러스터

 - 오토 스케일링 관련

 - 클러스터

 - Containerd 및 Docker를 선택하는 방법

- 서비스

- TKE Serverless 클러스터

 - TKE Serverless 클러스터

 - 로드 밸런서 FAQ

- 이미지 레지스트리

- 원격 터미널

FAQ

유지보수 로그 수집

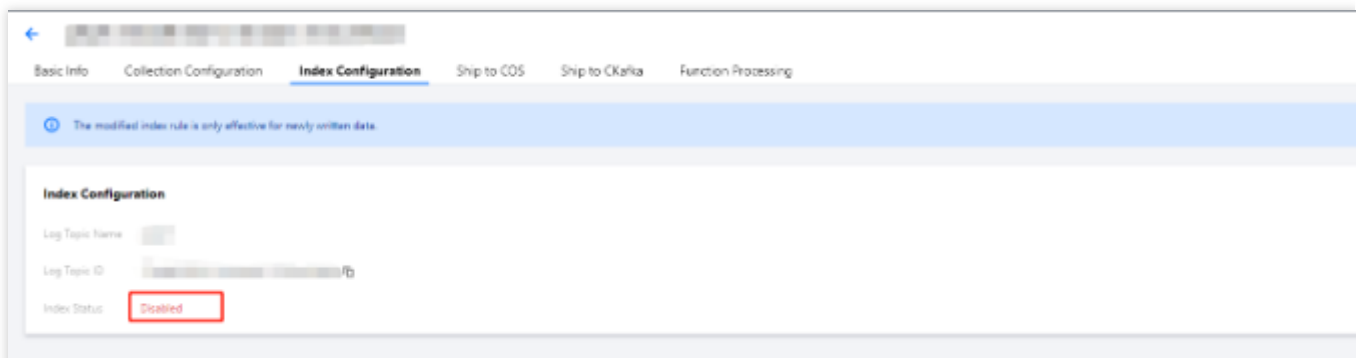
최종 업데이트 날짜: : 2023-04-26 19:23:11

로그 수집 FAQ

클러스터에 로그 수집을 구성한 후 CLS 콘솔에서 로그를 볼 수 없는 이유는 무엇입니까?

로그를 볼 수 없거나 누락된 경우 다음 문제가 있는지 확인하십시오.

선택한 로그 topic에 대한 인덱스를 활성화했는지 확인하십시오. CLS 로그 검색 및 분석을 위해서는 인덱스 구성이 필요합니다. 활성화되어 있지 않으면 로그를 볼 수 없습니다. 인덱스 구성 방법은 [CLS 인덱스 구성](#)을 참고하십시오.



로그, 감사 및 이벤트가 동일한 topic을 사용하는지 여부입니다. 동일한 topic을 사용하는 경우 로그를 덮어쓰기 하여 로그가 누락됩니다.

선택한 topic이 두 가지 추출 모드를 사용하는지 여부입니다. 새 추출 모드는 기존 추출 모드를 덮어씁니다.

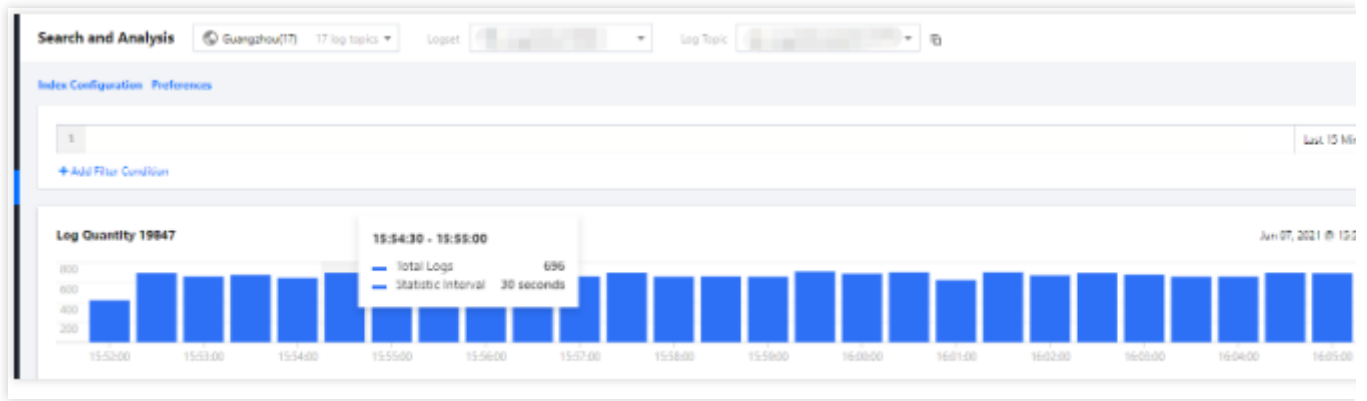
소프트 링크가 있는지 여부입니다. 컬렉션 유형을 '컨테이너 파일 경로'로 선택한 경우 해당 경로는 소프트 링크가 될 수 없습니다. 그렇지 않으면 소프트 링크의 실제 경로가 수집기 컨테이너에 존재하지 않아 로그 수집이 실패합니다.

환경 변수를 사용하여 TKE Serverless 로그 수집을 활성화하고 권한 부여 방법으로 역할을 선택한 경우 **역할 생성 시 역할 엔터티로 CVM을 선택해야 합니다.** TKE를 선택하면 인증을 완료할 수 없습니다.

문제가 지속되면 [티켓 제출](#)하여 문의해주시요.

로그 규칙을 구성한 후 로그는 어디에서 볼 수 있습니까?

1. [CLS 콘솔](#)에 로그인하고 왼쪽 사이드바에서 **검색 및 분석**을 선택합니다.
2. '검색 및 분석' 페이지에서 로그를 조회할 리전, 로그셋, 로그 테마를 선택하고 아래와 같이 전체 텍스트 인덱스를 활성화하여 로그를 검색 및 분석합니다.



환경 변수를 사용하여 TKE Serverless 로그 수집 활성화

Java 애플리케이션은 다중 라인 로그 병합을 어떻게 구현합니까?

애플리케이션의 로그 데이터가 다중 라인을 차지하는 경우(예시: Java 프로그램 로그) 줄 바꿈 `\n` 을 사용하여 로그의 끝을 표시할 수 없습니다. CLS가 각 로그를 명확하게 구분할 수 있도록 하려면 첫 라인 정규식을 사용하여 `configmap`을 구성해야 합니다. 라인의 로그가 미리 설정된 정규식과 일치하면 로그의 시작으로 간주되며 일치하는 다음 라인은 로그의 끝 표시가 됩니다. 자세한 내용은 [TKE Serverless 로그 수집을 위한 다중 라인 로그 병합 구현](#)을 참고하십시오.

다양한 로그 출력 속도에 맞게 로그 수집 구성을 조정하는 방법은 무엇입니까?

환경 변수를 사용하여 TKE Serverless 로그 수집을 활성화하면 TKE Serverless는 pod sandbox에서 로그 수집 컴포넌트를 시작하여 로그를 수집하고 보고합니다. TKE Serverless는 로그 수집 컴포넌트의 메모리 사용량을 제한하므로 로그 출력 속도가 너무 높으면 로그 수집 컴포넌트가 메모리 부족(OOM) 상태가 될 수 있습니다. 이때 필요에 따라 로그 수집 구성을 조정할 수 있습니다. 구체적인 방법은 메모리 사용량을 줄이기 위해 로그 수집 컴포넌트에서 사용하는 메모리 버퍼를 줄이기 위해 다음 pod annotation을 수동으로 수정하는 것입니다.



```
internal.eks.tke.cloud.tencent.com/tail-buffer-chunk-size: "2M"
internal.eks.tke.cloud.tencent.com/tail-buffer-max-size: "2M"
```

annotation에 대한 설명은 다음 표에 나와 있습니다. 자세한 내용은 [Fluent Bit](#)를 참고하십시오.

매개변수	설명	구분
Buffer_Chunk_Size	컨테이너 파일 로그를 수집하기 위한 초기 버퍼 크기를 설정하는 데 사용됩니다. 버퍼 크기를 늘리는 데에도 사용할 수 있습니다.	21

Buffer_Max_Size	모니터링되는 각 파일의 버퍼 크기 제한을 설정하는 데 사용됩니다. 버퍼를 늘려야 하는 경우(예시: 긴 로그) 이 값은 버퍼 증가량을 제한할 수 있습니다. 읽은 파일이 이 제한을 초과하면 파일이 모니터링 목록에서 삭제됩니다.	21
-----------------	--	----

Serverless 클러스터의 컨테이너에서 애플리케이션의 로그를 출력하는 기준은 무엇입니까?

애플리케이션이 로그를 출력하면 로그가 stdout으로 출력되어야 합니다. 프로그램 로그가 컨테이너 파일로 출력되는 경우 로그 파일을 정기적으로 정리하거나 영구 저장을 위해 volume을 마운트해야 합니다. 그렇지 않으면 20G 임시 저장소가 모두 사용됩니다. 자세한 내용은 [Pod 임시 스토리지](#)를 참고하십시오.

로그 파일을 정리하는 방법을 모르는 경우 다음 방법을 권장합니다.

영구 스토리지로 volume을 마운트합니다. 자세한 내용은 [스토리지 관리](#)를 참고하십시오.

TKE Serverless 로그 수집을 활성화합니다. 자세한 내용은 [TKE 로그 수집](#)을 참고하십시오.

클러스터

오토 스케일링 관련

최종 업데이트 날짜: : 2022-03-31 15:04:02

Cluster Autoscaler와 모니터링 메트릭을 기반으로 하는 노드 오토 스케일링의 차이점은 무엇입니까?

Cluster Autoscaler는 클러스터의 모든 Pod가 실제 부하와 상관없이 스케줄링될 수 있도록 하는 반면 모니터링 메트릭을 기반으로 하는 노드 오토 스케일링은 오토 스케일링 동안 Pod를 고려하지 않습니다. 따라서 Pod가 없는 노드가 추가되거나 kube-dns와 같은 시스템에 중요한 Pod가 있는 노드가 오토 스케일링 중에 삭제될 수 있습니다.

Kubernetes는 후자의 오토 스케일링 메커니즘을 권장하지 않습니다. 결론적으로 이 두 모드는 충돌하므로 동시에 활성화하면 안 됩니다.

CA는 오토 스케일링 그룹과 어떻게 작동합니까?

CA 지원 클러스터는 선택한 노드의 구성에 따라 시작 구성을 생성하고 여기에 오토 스케일링 그룹을 바인딩합니다. 그러면 클러스터는 이 바인딩된 오토 스케일링 그룹에서 scale-in/out을 수행합니다. 확장된 CVM 인스턴스는 클러스터에 자동으로 추가됩니다. 자동으로 확장/축소되는 노드는 사용한 만큼만 요금이 청구됩니다. 오토 스케일링 그룹에 대한 자세한 내용은 [Auto Scaling](#)을 참고하십시오.

TKE 콘솔에 수동으로 추가된 노드를 CA에서 축소할 수 있습니까?

아니요. CA는 오토 스케일링 그룹 내의 노드만 확장합니다. [TKE 콘솔](#)에 추가된 노드는 오토 스케일링 그룹에 추가되지 않습니다.

오토 스케일링 콘솔에서 CVM 인스턴스를 추가하거나 제거할 수 있습니까?

아니요. [오토 스케일링 콘솔](#)을 수정하지 않는 것이 좋습니다.

확장하는 동안 선택한 노드의 어떤 구성이 상속됩니까?

오토 스케일링 그룹을 생성할 때 클러스터의 노드를 참고로 선택하여 [시작 구성](#)을 생성해야 합니다. 참고용 노드 구성에는 다음이 포함됩니다.

- vCPU
- 메모리
- 시스템 디스크 크기
- 데이터 디스크 크기
- 디스크 유형
- 대역폭
- 대역폭 과금 방식
- 공용 IP 할당 여부
- 보안 그룹

- Virtual Private Cloud
- 서브넷

여러 오토 스케일링 그룹을 사용하려면 어떻게 해야 하나요?

서비스의 레벨과 종류에 따라 여러 개의 오토 스케일링 그룹을 생성한 후 각각 다른 label을 설정하고, 오토 스케일링 그룹에서 스케일 아웃된 노드의 label을 지정하여 서비스를 분류할 수 있습니다.

스케일링 최대 할당량은 얼마입니까?

각 Tencent Cloud 사용자에게는 각 가용존에서 종량제 CVM 인스턴스 30개의 할당량이 제공됩니다. 오토 스케일링 그룹에 더 많은 인스턴스를 신청하기 위해 [티켓 제출](#)할 수 있습니다.

할당량에 대한 자세한 내용은 현재 가용존에 대한 [CVM 인스턴스 수량 및 할당량](#)을 참고하십시오. 또한 오토 스케일링의 인스턴스는 최대 200개로 제한됩니다. [티켓 제출](#)하여 더 높은 할당량을 신청할 수 있습니다.

스케일 다운이 클러스터에 안전한가요?

노드가 축소되면 Pod가 다시 예약되므로 서비스가 일정 변경 및 단기 중단을 허용할 수 있는 경우에만 축소를 수행할 수 있습니다. [PDB](#) 사용을 권장합니다. PDB는 항상 사용 가능한 상태로 유지되는 Pod 세트에 대한 복제본의 최소 수/백분율을 지정할 수 있습니다. PodDisruptionBudget을 사용하면 애플리케이션 배포자가 Pod를 적극적으로 제거하는 클러스터 작업이 한 번에 너무 많은 Pod를 종료하지 않도록 하여 데이터 손실, 서비스 중단 또는 허용할 수 없는 서비스 저하를 방지할 수 있습니다.

노드의 어떤 유형의 Pod가 축소되지 않습니까?

- Pod에 대해 엄격한 PodDisruptionBudget을 설정한 경우 PDB가 충족되지 않으면 축소되지 않습니다.
- Kube-system의 Pod.
- deployment, replica set, job 또는 stateful set과 같은 컨트롤러에서 생성되지 않은 노드의 Pod.
- 로컬 스토리지가 있는 Pod.
- 다른 노드에 예약할 수 없는 Pod입니다.

조건이 충족된 후 축소를 트리거하는 데 얼마나 걸립니까?

10분.

노드가 Not Ready로 표시될 때 축소를 트리거하는 데 얼마나 걸립니까?

20분.

스케일링을 위해 노드를 얼마나 자주 스캔합니까?

10초.

CVM 인스턴스를 확장하는 데 얼마나 걸립니까?

일반적으로 10분 미만이 소요됩니다. 자세한 내용은 [오토 스케일링](#)을 참고하십시오.

Unschedulable Pod가 있는 노드가 확장되지 않는 이유는 무엇입니까?

다음을 확인하십시오.

- Pod의 요청된 리소스가 너무 크지 않은지.
- node selector 설정 여부.
- 오토 스케일링 그룹의 최대 노드 수 도달 여부.
- 계정 잔액 충분 여부(계정 잔액이 부족한 경우 수평 확장을 트리거할 수 없음), 할당량 부족 여부. 자세한 내용은 [Troubleshooting](#)을 참고하십시오.

Cluster Autoscaler가 특정 노드에서 확장되지 않도록 하려면 어떻게 해야 합니까?

```
# 노드의 annotations에서 다음 정보를 설정할 수 있습니다.
kubectl annotate node <nodename> cluster-autoscaler.kubernetes.io/scale-down-disabled=true
```

스케일링 이벤트에 대한 세부 정보는 어디에서 찾을 수 있나요?

오토 스케일링 그룹의 스케일링 이벤트를 조회하고 오토 스케일링 콘솔에서 K8s 이벤트를 볼 수 있습니다. 이벤트는 다음 세 가지 리소스에서 찾을 수 있습니다.

- kube-system/cluster-autoscaler-status config map
 - **ScaledUpGroup** - CA가 스케일 업을 트리거합니다.
 - **ScaleDownEmpty** - CA는 실행 중인 Pod가 없는 노드를 삭제합니다.
 - **ScaleDown** - CA가 스케일 다운을 트리거합니다.
- node
 - **ScaleDown** - CA가 스케일 다운을 트리거합니다.
 - **ScaleDownFailed** - CA가 스케일 다운을 트리거하지 못했습니다.
- pod
 - **TriggeredScaleUp** - CA는 이 Pod로 인해 스케일 업을 트리거합니다.
 - **NotTriggerScaleUp** - CA가 이 Pod를 스케줄링하기 위해 사용 가능한 오토 스케일링 그룹을 찾을 수 없습니다.
 - **ScaleDown** - CA는 노드를 스케일 업하기 위해 이 Pod를 트레이닝하려고 합니다.

클러스터

최종 업데이트 날짜: : 2022-03-31 15:04:02

클러스터 생성 FAQ

클러스터를 생성할 때 CVM에 대한 공용 IP 주소를 설정하지 않도록 선택할 수 있습니까?

네. 공용 IP 주소가 없는 CVM은 이미지 레지스트리의 내 이미지에서만 이미지를 가져올 수 있지만 dockerhub 이미지 또는 3rd party 이미지는 가져올 수 없습니다.

공용 IP 주소는 없지만 인터넷 대역폭이 있는 CVM은 EIP를 바인딩하여 인터넷에 액세스할 수 있습니다.

클러스터를 생성할 때 네트워크를 선택해야 하는 이유는 무엇입니까?

선택한 네트워크 및 서브넷은 CVM이 상주하는 위치입니다. 교차 가용존 재해 복구를 위해 서로 다른 가용존의 서브넷에 서로 다른 CVM을 추가할 수 있습니다.

클러스터 생성 시 지원되는 CVM 모델은 무엇입니까?

클라우드 디스크를 시스템 디스크로 사용하는 모든 사용량 과금 모델이 지원됩니다.

TKE 호스트에 지원되는 운영 체제는 무엇입니까?

현재 Ubuntu 16.04 및 CentOS 7.2가 지원됩니다. 다른 운영 체제가 필요한 경우 [티켓 제출](#)하여 당사에 문의하십시오.

CVM 추가 FAQ

클러스터에 CVM을 추가할 때의 제한 사항은 무엇입니까?

클러스터와 동일한 지역에 있는 CVM만 선택할 수 있습니다. 그러나 클러스터의 교차 가용존 배포를 구현하기 위해 다른 가용존을 선택할 수 있습니다.

CVM 수에 제한이 있습니까?

네. 사용량 과금 CVM의 수는 현재 계정의 사용 할당량을 초과할 수 없습니다. 자세한 내용은 [CVM 개요](#)를 참고하십시오.

CVM 종료 FAQ

CVM이 종료된 후 CVM에 배포된 컨테이너는 어떻게 됩니까?

CVM 인스턴스가 종료되면 컨테이너와 같은 포함된 리소스도 종료됩니다. 서비스의 컨테이너 수가 예상 실행 컨테이너 수 아래로 떨어지면 클러스터는 원하는 수에 도달할 때까지 다른 CVM 인스턴스에서 더 많은 컨테이너를 시작합니다.

Containerd 및 Docker를 선택하는 방법

최종 업데이트 날짜: : 2022-03-31 15:06:15

컨테이너 런타임은 어떻게 선택합니까?

Kubernetes(K8S)의 가장 중요한 컴포넌트 중 하나인 컨테이너 런타임(Container Runtime)은 이미지 및 컨테이너의 라이프사이클을 관리합니다. Kubelet은 Container Runtime Interface (CRI) 를 통해 컨테이너 런타임과 상호 작용하여 이미지와 컨테이너를 관리합니다.

TKE는 컨테이너 런타임으로 containerd 및 docker를 지원합니다.

- Containerd는 호출 체인이 더 짧고 컴포넌트가 적으며 안정성이 더 높고 노드 리소스 소비가 더 적습니다. containerd 선택을 권장합니다.
- 다음을 사용해야 하는 경우 docker를 런타임 컴포넌트로 사용해야 합니다.
- docker in docker.
- docker build/push/save/load와 같은 명령.
- docker API.
- docker compose 또는 docker swarm.

Containerd 및 Docker에서 일반적으로 사용되는 명령은 무엇입니까?

Containerd는 docker API 또는 docker CLI를 지원하지 않습니다. 그러나 cri-tool 명령으로 이러한 기능을 구현할 수 있습니다.

이미지 관련 기능	Docker	Containerd
로컬 이미지 목록 표시	docker images	crictl images
이미지 다운로드	docker pull	crictl pull
이미지 업로드	docker push	없음
로컬 이미지 삭제	docker rmi	crictl rmi
이미지 세부정보 보기	docker inspect IMAGE-ID	crictl inspecti IMAGE-ID

컨테이너 관련 기능	Docker	Containerd
컨테이너 목록 표시	docker ps	crictl ps
컨테이너 생성	docker create	crictl create
컨테이너 시작	docker start	crictl start

컨테이너 관련 기능	Docker	Containerd
컨테이너 중지	docker stop	crictl stop
컨테이너 삭제	docker rm	crictl rm
컨테이너 세부정보 보기	docker inspect	crictl inspect
attach	docker attach	crictl attach
exec	docker exec	crictl exec
logs	docker logs	crictl logs
stats	docker stats	crictl stats

POD 관련 기능	Docker	Containerd
POD 목록 표시	없음	crictl pods
POD 세부정보 보기	없음	crictl inspectp
POD 실행	없음	crictl runp
POD 중지	없음	crictl stopp

Containerd와 Docker의 호출 체인의 차이점은 무엇입니까?

- Docker가 K8S 컨테이너 런타임으로 사용되는 경우 호출 체인은 다음과 같습니다.

```
kubelet --> docker shim (kubelet 프로세스에서) --> dockerd --> containerd
```

- Containerd가 K8S 컨테이너 런타임으로 사용되는 경우 호출 체인은 다음과 같습니다.

```
kubelet --> cri plugin(containerd 프로세스에서) --> containerd
```

dockerd는 swarm cluster, docker build 및 docker API와 같은 더 많은 기능을 제공하지만 일부 bug가 발생할 수 있으며 containerd보다 호출 단계가 하나 더 필요합니다.

Stream 서비스

Kubectl exec/logs 등과 같은 명령을 사용하려면 apiserver와 컨테이너 런타임 사이에 스트림 전달 채널을 설정해야 합니다.

Containerd에서 Stream 서비스는 어떻게 사용 및 구성됩니까?

Docker API 자체가 stream 서비스를 제공하고 Kubelet 내부의 docker-shim은 docker API를 통해 스트림을 전달합니다.

Containerd의 stream 서비스는 별도로 구성해야 합니다.

```
[plugins.cri]
stream_server_address = "127.0.0.1"
stream_server_port = "0"
enable_tls_streaming = false
```

K8S 1.11 이전 버전과 이후 버전의 차이점은 무엇입니까?

Containerd의 stream 서비스는 K8S 버전에 따라 구성이 다릅니다.

- K8S 1.11 이전:

Kubelet은 리디렉션을 수행하지만 stream proxy는 수행하지 않습니다. 즉, Kubelet은 containerd에 의해 열린 stream server 주소를 apiserver로 보낸 다음 containerd의 stream 서비스에 직접 액세스합니다. 보안을 위해 stream 서비스 전달자를 인증해야 합니다.
- K8S 1.11 이후:

K8S1.11은 kubelet stream proxy를 도입하여 containerd의 stream 서비스는 로컬 주소만 수신하면 됩니다.

기타 차이점

컨테이너 로그 및 관련 매개변수

항목	Docker	Containerd
스토리지 경로	Docker는 컨테이너 로그를 <code>/var/lib/docker/containers/\$CONTAINERID</code> 와 같은 디렉터리에 저장합니다. Kubelet은 <code>/var/log/pods</code> 및 <code>/var/log/containers</code> 아래에 <code>/var/lib/docker/containers/\$CONTAINERID</code> 디렉터리의 컨테이너 로그 파일을 가리키는 소프트 링크를 생성합니다.	Kubelet은 컨테이너 로그를 <code>/var/log/pods/\$CONTAINER_NAME</code> 디렉터리에 저장하고 로그 파일을 가리키는 <code>/var/log/containers</code> 아래에 소프트 링크를 생성합니다.
매개변수 설정	docker 구성 파일에서 다음을 지정합니다. <pre>"log-driver": "json-file", "log-opts": {"max-size": "100m", "max-file": "5"}</pre>	<ul style="list-style-type: none"> • 방법1: kubelet 매개변수에 다음을 지정: <pre>--container-log-max-files=5 --container-log-max-size="100Mi"</pre> • 방법2: KubeletConfiguration에서 다음을 지정:

		<pre>"containerLogMaxSize": "100Mi", "containerLogMaxFiles": 5,</pre>
<p>데이터 디스크에 컨테이너 로그 저장</p>	<p>데이터 디스크를 'data-root'(기본적으로 <code>/var/lib/docker</code>)에 마운트합니다.</p>	<p>데이터 디스크 탑재 지점 아래의 디렉터리를 가리키는 소프트 링크 <code>/var/log/pods</code> 를 만듭니다. TKE에서 '데이터 디스크에 컨테이너 및 이미지 저장'을 선택하면 소프트 링크 <code>/var/log/pods</code> 가 자동으로 생성됩니다.</p>

CNI 네트워크

항목	Docker	Containerd
CNI 호출을 담당하는 구성 요소	Kubelet 내부의 docker-shim	Containerd의 내장 cri-plugin(containerd 1.1 이상)
CNI를 구성하는 방법	Kubelet 매개변수 <code>--cni-bin-dir</code> 및 <code>--cni-conf-dir</code>	Containerd 구성 파일(toml): <pre>[plugins.cri.cni] bin_dir = "/opt/cni/bin" conf_dir = "/etc/cni/net.d"</pre>

서비스

최종 업데이트 날짜: : 2022-03-31 15:04:02

서비스 생성 관련 FAQ

서비스 이름이 고유해야 하는 이유는 무엇입니까?

서비스 이름은 클러스터에서 서비스를 고유하게 식별합니다. 서비스는 서비스 이름+액세스 포트를 사용하여 서로 액세스할 수 있습니다.

Tencent Cloud 또는 dockerhub 이미지 대신 3rd party 이미지를 사용하여 서비스를 생성할 수 있습니까?

네. CVM에 로그인하고 docker login 명령을 실행하여 3rd party 이미지 레지스트리에 로그인하고 이미지를 풀링할 수 있습니다.

인터넷 서비스를 이용하기 위한 전제 조건은 무엇입니까?

클러스터의 CVM 인스턴스는 인터넷 대역폭으로 구성되어야 합니다. 그렇지 않으면 인터넷 서비스가 생성되지 않습니다.

메모리 및 CPU 제한은 어떻게 구성합니까?

자세한 내용은 [위크로드의 리소스 제한 설정](#)을 참고하십시오.

서비스를 생성할 때 혜택 옵션은 무엇을 의미합니까?

이 옵션이 활성화되면 컨테이너의 애플리케이션은 진정한 root 권한을 갖게 됩니다. 예를 들어 NFS 서버를 구축할 때와 같이 컨테이너의 애플리케이션에 대해 높은 수준의 시스템 작업을 수행해야 하는 경우 이 옵션을 활성화하는 것이 좋습니다.

CLB 생성 시 보안 그룹을 지정할 수 있습니까?

네. 서비스에서 CLB를 사용할 때 다음 솔루션 중 하나를 사용하여 보안 그룹을 지정할 수 있습니다.

- 기존 CLB를 사용합니다. 먼저 CLB를 생성하고 보안 그룹을 구성한 다음 서비스에 탑재합니다. 자세한 내용은 기존 CLB 사용 [Service](#)를 참고하십시오.
- 서비스에서 `TkeServiceConfig` 를 사용하여 보안 그룹을 설정합니다. CLB가 생성되면 CLB는 구성에 따라 해당 보안 그룹을 사용합니다. 이 기능을 사용하려면 [티켓 제출](#)하십시오.

접근 실패를 방지하기 위해 CLB IP 주소를 사용하여 클러스터의 서비스에 접근하지 마십시오.

일반적으로 Layer4 CLB는 여러 Node를 real server(rs)로 바인딩합니다. client와 rs가 동일한 CVM에 있지 않

은지 확인합니다. 그렇지 않으면 루프백으로 인해 패킷이 전송되지 않을 수 있습니다.

Pod가 CLB에 액세스할 때 Pod IP 주소는 소스 IP 주소입니다. 패킷이 사설망으로 라우팅될 때 CLB는 snat를 사용하여 소스 IP 주소를 Node IP 주소로 변환하지 않습니다. 결과적으로 CLB는 패킷의 소스 Node를 식별할 수 없고 CLB의 루프백 방지 정책이 적용되지 않으며 패킷은 임의의 rs로 전달될 수 있습니다. client가 있는 Node로 패킷을 전달할 때 CLB는 응답을 수신할 수 없으므로 액세스 실패가 발생합니다.

서비스 컨테이너 수 업데이트에 대한 FAQ

컨테이너 수를 업데이트할 때 주의해야 할 사항은 무엇입니까?

CPU 및 메모리 리소스가 충분한지 확인합니다. 그렇지 않으면 컨테이너 생성이 실패합니다.

컨테이너 수를 0으로 설정할 수 있습니까?

네. 컨테이너 수를 0으로 설정하면 서비스 설정이 유지되고 리소스가 해제됩니다.

서비스 설정 업데이트에 대한 FAQ

롤링 업데이트가 지원됩니까?

네. 롤링 업데이트와 빠른 업데이트가 모두 지원됩니다.

공개 CLB를 비공개 CLB로 변경할 수 있습니까?

네. 현재 서비스 액세스 모드를 인터넷을 통해에서 VPC를 통해로 또는 VPC를 통해에서 인터넷을 통해로 변경하거나 VPC 서브넷을 전환할 수 있습니다. 자세한 내용은 Service 라이프사이클 관리를 참고하십시오.

- 서비스가 CLB 라이프사이클을 관리하는 경우 CLB 및 해당 공용 IP 주소가 해제됩니다.
- 인터넷에서 VPC로 전환하는 데 시간이 걸립니다. 즉, public CLB가 오프라인 상태가 된 후 private CLB가 서비스를 제공하기 시작할 때까지 시간이 걸립니다. 먼저 클러스터에서 프라이빗 서비스를 구성하고 테스트하는 것이 좋습니다. 그런 다음 트래픽이 새 서비스로 전환된 후 원래 공공 서비스를 삭제합니다.

서비스 삭제 관련 FAQ

서비스가 생성한 CLB는 서비스 삭제 후 폐기되나요?

서비스를 생성할 때 CLB가 자동으로 생성된 경우 서비스 삭제 시 CLB도 함께 삭제됩니다. 서비스 생성 시 기존 CLB를 선택하면 해당 CLB는 삭제되지 않습니다.

서비스가 삭제되면 서비스 데이터가 영향을 받습니까?

아니요. 서비스를 삭제해도 서비스 컨테이너는 삭제되지 않으며 데이터도 영향을 받지 않습니다. 서비스를 삭제하기 전에 데이터를 백업할 필요가 없습니다.

서비스 운영에 관한 FAQ

컨테이너 시스템 시간을 베이징 시간으로 설정하려면 어떻게 합니까?

컨테이너의 기본 시스템 시간은 협정 세계시(UTC)입니다. 컨테이너 사용 시 컨테이너 시스템 시간이 현지 시간대와 다른 경우 `dockerfile`에 시간대 파일을 생성하여 이 문제를 해결할 수 있습니다. 자세한 내용은 [컨테이너에서 일관되지 않은 표준 시간대 문제 해결](#)을 참고하십시오.

ubuntu, php, busybox와 같은 일부 DockerHub 이미지에 TKE 예외가 발생하면 어떻게 해야 합니까?

시작 명령이 설정되지 않았거나 기본 시작 명령이 `bash`인 경우 시작 절차가 완료된 후 컨테이너가 종료됩니다. 컨테이너를 계속 실행하려면 PID가 1로 설정된 컨테이너의 프로세스가 영구 프로세스여야 합니다. 그렇지 않으면 이 프로세스가 종료될 때 컨테이너가 종료됩니다. `centos`와 같은 일부 이미지의 경우 `/bin/bash`를 실행 명령으로 사용하고 `-c sleep 800000`을 실행 매개변수로 사용하여 서비스를 생성할 수 있습니다. `-c` 및 `sleep 800000`은 콘솔의 다른 행에 입력해야 합니다.

현재 기본 매개변수를 사용하여 시작할 수 없는 이미지에는 `clearlinux`, `ros`, `mageia`, `amazonlinux`, `ubuntu`, `clojure`, `crux`, `gcc`, `photon`, `java`, `debian`, `oraclelinux`, `mono`, `bash`, `buildpack-deps`, `golang`, `sourcemage`, `swift`, `openjdk`, `centos`, `busybox`, `docker`, `alpine`, `ibmjava`, `php` 및 `python`이 있습니다.

TKE Serverless 클러스터

TKE Serverless 클러스터

최종 업데이트 날짜: : 2023-04-26 19:23:11

본문은 TKE Serverless 클러스터에 대한 FAQ, 원인 및 솔루션을 소개합니다.

Pod 사양이 입력된 Request/Limit 값과 일치하지 않는 이유는 무엇인가요?

Pod에 리소스를 할당할 때 TKE Serverless는 설정된 Request 및 Limit 값에 따라 리소스를 할당하는 대신 워크로드에 의해 설정된 Request 및 Limit를 계산하고 Pod를 실행하는 데 필요한 리소스의 양을 자동으로 결정합니다. 자세한 내용은 [CPU specifications calculation methods for pods](#) 및 [GPU specification calculation methods for pods](#)를 참고하십시오. Request, Limit를 통해 지정된 리소스 사양을 자동으로 계산하는 방법에 대해 자세히 알아보십시오.

TKE Serverless 클러스터의 컨테이너 네트워크를 생성 또는 수정하려면 어떻게 해야 하나요?

클러스터를 생성할 때 VPC를 클러스터 네트워크로 선택하고 서브넷을 컨테이너 네트워크로 지정해야 합니다. 자세한 내용은 [컨테이너 네트워크 설명](#)을 참고하십시오. Serverless의 Pod는 컨테이너 네트워크 서브넷의 IP 주소를 직접 점유합니다. 클러스터를 사용할 때 슈퍼 노드 생성 또는 삭제를 통해 컨테이너 네트워크를 생성하거나 수정할 수 있습니다. 자세한 지침은 아래와 같습니다.

1단계: 컨테이너 네트워크를 추가하기 위해 슈퍼 노드 생성

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 [Cluster](#)를 선택합니다.
2. 컨테이너 네트워크 수정이 필요한 클러스터 ID를 클릭하여 클러스터 기본 정보 페이지로 이동합니다.
3. 왼쪽에서 [슈퍼 노드](#)를 선택하고 [슈퍼 노드](#) 페이지에서 [생성](#)을 클릭합니다.
4. [슈퍼 노드 생성](#) 페이지에서 IP 주소가 충분한 컨테이너 네트워크를 선택하고 [확인](#)을 클릭합니다.

Basic information

Region: [Redacted] (ju)

Cluster ID: [Redacted]

Kubernetes version: 1. [Redacted]

Cluster network: [Redacted] 7) [↗](#)

Super node

Node pool: Auto-create a node pool Add to an existing node pool

Node pool name:

The name cannot exceed 25 characters. It only supports Chinese characters, English letters, numbers, underscores, hyphens ("-") and dots.

OS type: Linux ▼

When Windows OS is selected, Windows containers are supported in this subnet. Linux OS is selected by default.

⚠ From January 31, 2023 (UTC +8), discounts on the official site of Tencent Cloud that apply to pay-as-you-go Pods of small sizes on a s

Super node configuration

Availability zone: Guangzhou Zone 3 Guangzhou Zone 4 Guangzhou Zone 6 Guangzhou Zone 7 Guangz

Billing mode: Pay-as-you-go Monthly subscription

For more information, see [Super Node Pricing](#) [↗](#)

Container network

<input type="checkbox"/>	Subnet ID	Subnet name	Availability zo...	Remaining IPs
<input type="checkbox"/>	[Redacted]	[Redacted]	Guangz	[Redacted]

The Pod will occupy the IPs of selected subnets. Please select subnets with sufficient IPs and not conflict with other serv suitable, please go to the console to [create subnet](#) [↗](#).

Node name: Auto-generated [✎](#)

Confirm Cancel

[Add node](#)

Create super node Cancel

2단계: 슈퍼 노드를 삭제하여 컨테이너 네트워크 삭제

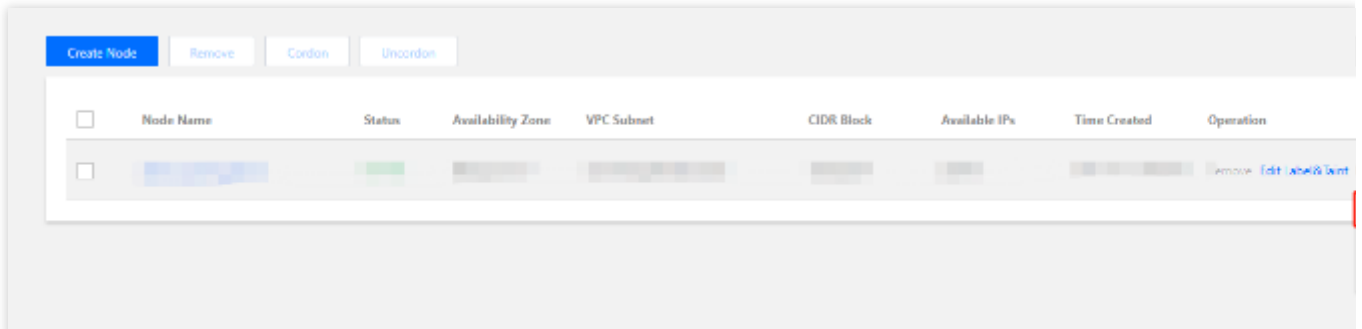
참고

삭제 후 Serverless 클러스터에 하나 이상의 슈퍼 노드가 남아 있는지 확인하십시오. 슈퍼 노드가 하나만 있으면 삭제할 수 없습니다.

슈퍼 노드를 삭제하기 전에 거기에 있는 모든 Pod(DaemonSet에서 관리하는 Pod 제외)를 다른 슈퍼 노드로 드레이닝해야 합니다. 드레이닝이 완료되면 슈퍼 노드를 삭제할 수 있습니다. 그렇지 않으면 삭제에 실패합니다. 자세한 안내는 아래와 같습니다.

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 **Cluster**를 선택합니다.
2. 컨테이너 네트워크 수정이 필요한 클러스터 ID를 클릭하여 클러스터 기본 정보 페이지로 이동합니다.

3. 왼쪽의 **슈퍼 노드**를 선택하고 **슈퍼 노드** 페이지에서 노드 이름 오른쪽의 **더보기 > 드레이닝**을 아래 이미지와 같이 선택합니다.



4. **드레인 노드** 페이지에서 노드 정보를 확인하고 **확인**을 클릭합니다. 드레이닝 후 '차단됨' 상태가 되며 더 이상 Pod를 스케줄링할 수 없습니다.

참고

노드가 드레이닝되면 Pod가 다시 빌드됩니다.

5. **슈퍼 노드** 페이지에서 노드 이름 오른쪽에 있는 **삭제**를 선택합니다.

6. **노드 삭제** 페이지에서 **확인**을 클릭하면 노드 삭제 작업이 완료됩니다.

서브넷 IP 주소가 부족하여 Pod 스케줄링에 실패하면 어떻게 해야 하나요?

서브넷 IP 주소가 부족하여 Pod 스케줄링에 실패하면 노드 로그에서 두 개의 이벤트를 찾을 수 있습니다.

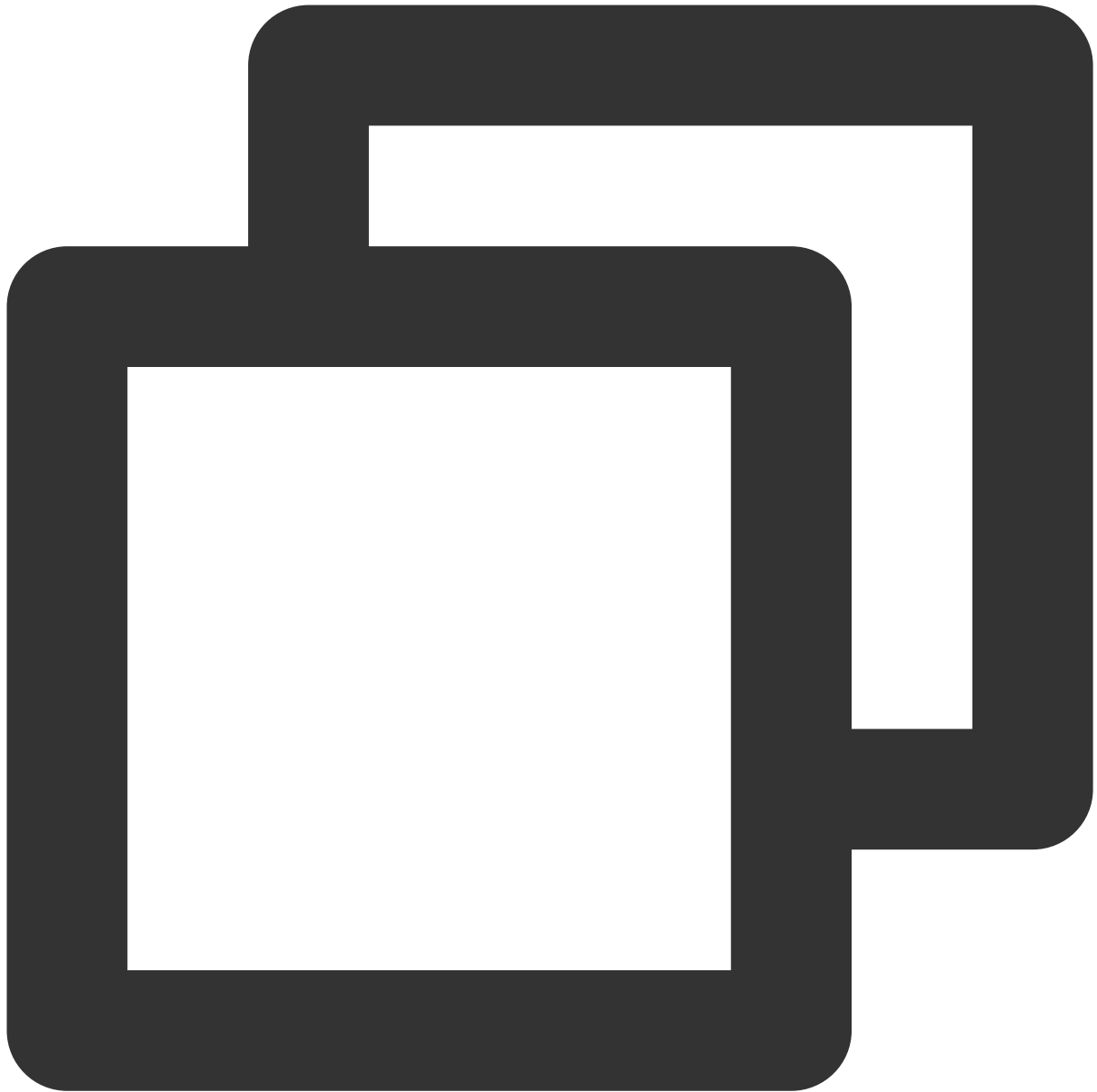
이벤트1:

2021-04-20 15:16:01	2021-04-20 15:35:17	Warning	Pod		FailedScheduling
2021-04-20 15:19:11	2021-04-20 15:35:17	Warning	Pod		FailedScheduling

이벤트2:

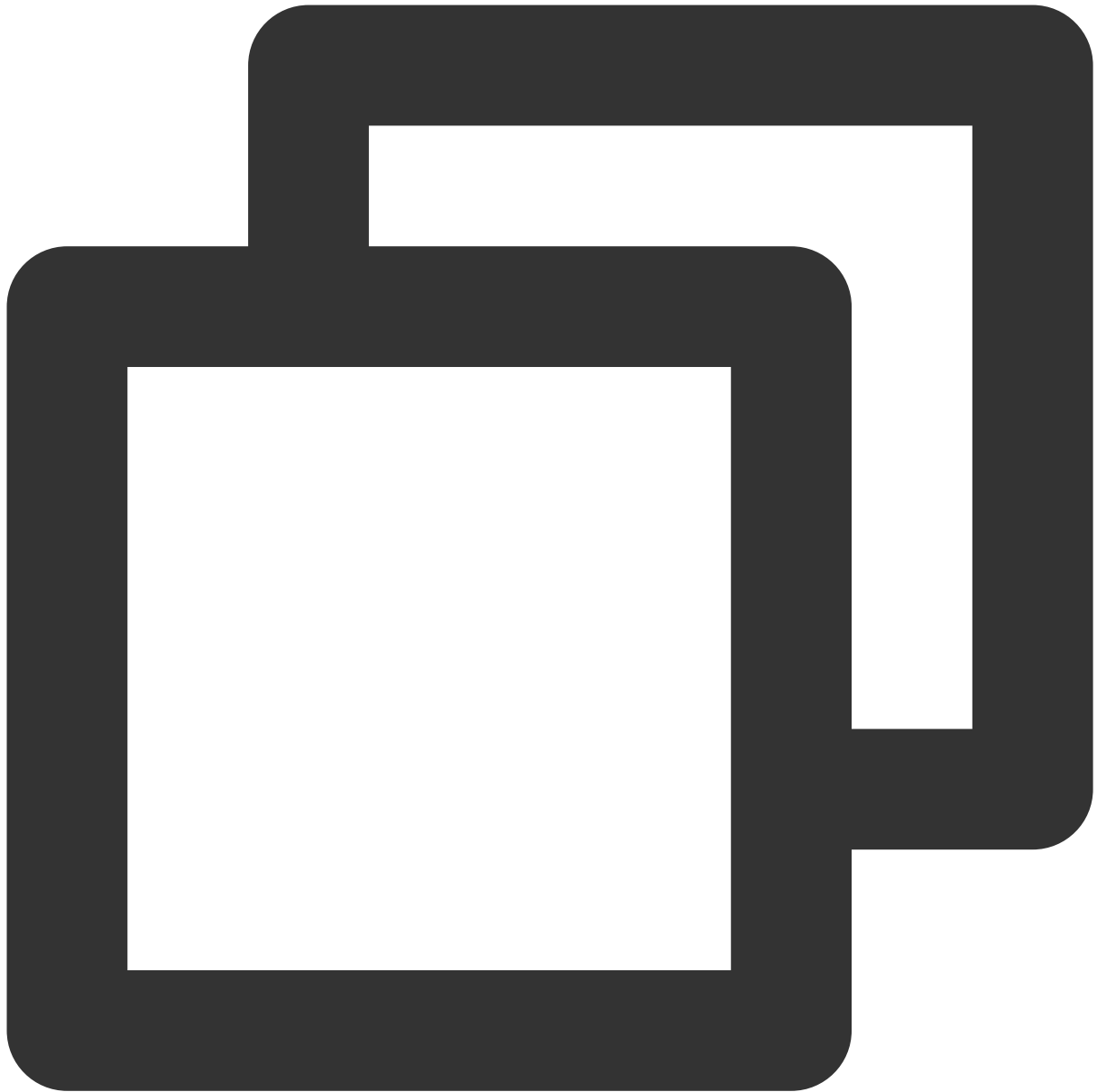
2021-04-20 15:00:10	2021-04-20 15:00:10	Warning	Pod		FailedCreatePodSandB
2021-04-20 15:00:07	2021-04-20 15:00:07	Warning	Pod		FailedCreatePodSanc

TKE 콘솔 또는 명령 라인 툴에서 다음 명령을 실행하여 슈퍼 노드의 YAML을 쿼리할 수 있습니다.

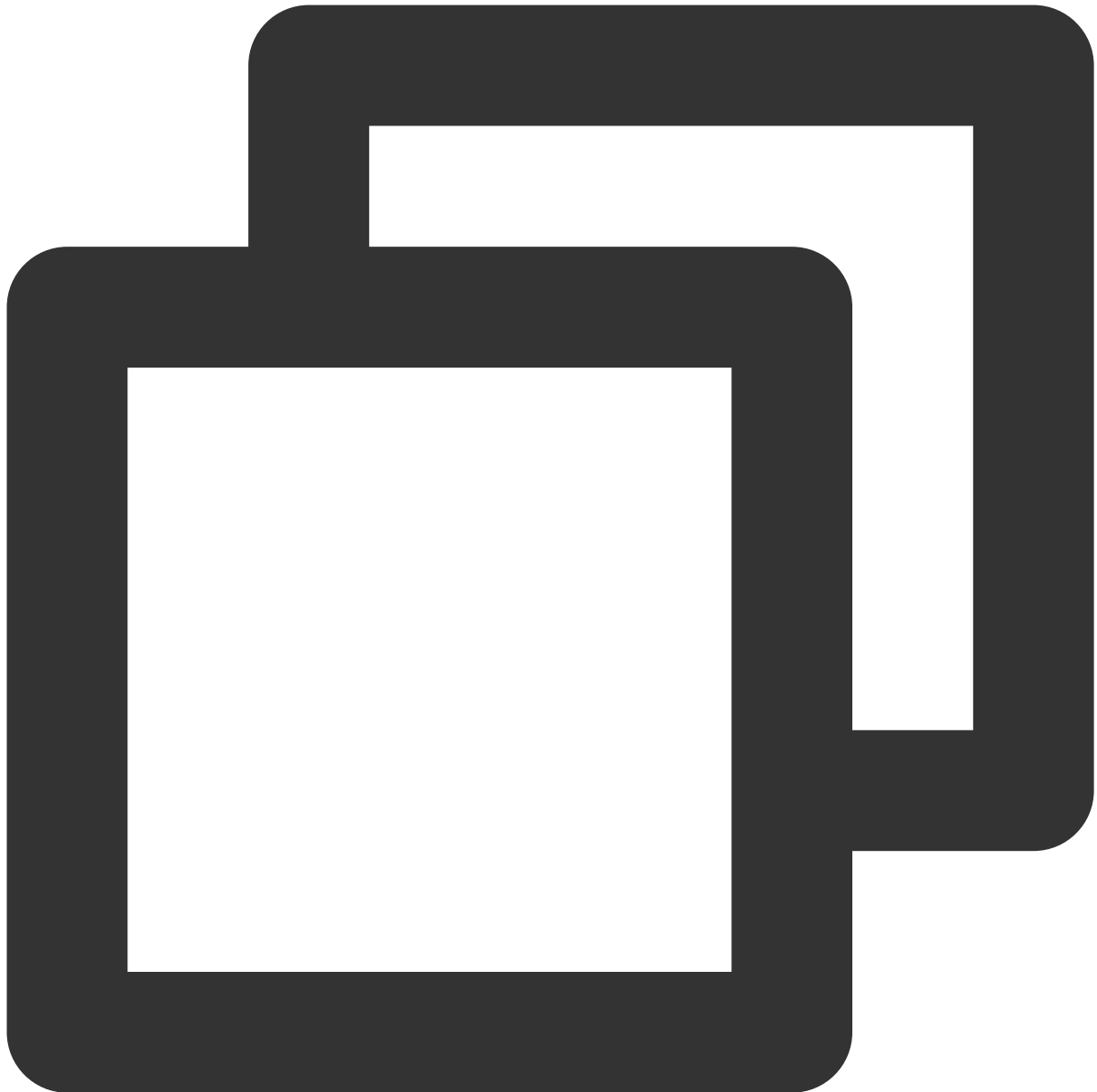


```
kubectl get nodes -oyaml
```

반환 결과는 다음과 같습니다.



```
spec:
  taints:
  - effect: NoSchedule
    key: node.kubernetes.io/network-unavailable
    timeAdded: "2021-04-20T07:00:16Z"
```

```
- lastHeartbeatTime: "2021-04-20T07:55:28Z"  
  lastTransitionTime: "2021-04-20T07:00:16Z"  
  message: eklet node has insufficient IP available of subnet subnet-bok73g4c  
  reason: EKLetHasInsufficientSubnetIP  
  status: "True"  
  type: NetworkUnavailable
```

컨테이너 네트워크의 서브넷 IP 주소가 부족하여 Pod 스케줄링에 실패했음을 보여줍니다. 이 경우 서브넷과 사용 가능한 IP 범위를 추가하기 위해 슈퍼 노드를 생성해야 합니다. 슈퍼 노드 생성 방법은 [슈퍼 노드 생성](#)을 참고하십시오.

TKE Serverless 보안 그룹을 어떻게 사용합니까?

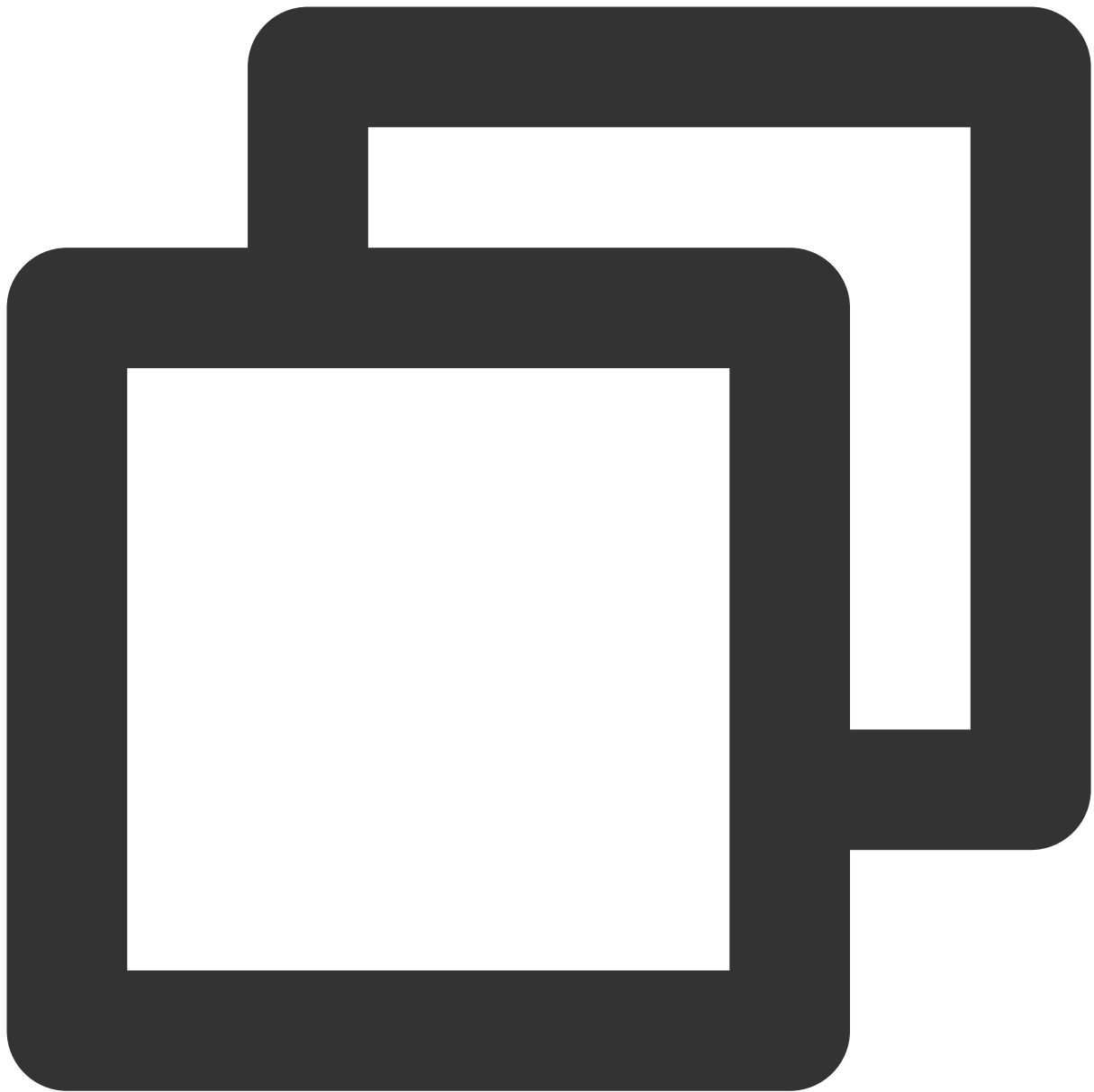
Serverless 클러스터 Pod 생성 시 보안 그룹을 지정하지 않으면 default 보안 그룹이 사용됩니다. Annotation `eks.tke.cloud.tencent.com/security-group-id` : 보안 그룹 ID 를 통해서도 Pod에 대한 보안 그룹을 지정할 수 있습니다. 워크로드가 있는 리전에 보안 그룹 ID가 이미 있는지 확인하십시오. 이 Annotation에 대한 자세한 내용은 [Annotation](#)을 참고하십시오.

컨테이너 종료 메시지는 어떻게 설정합니까?

Kubernetes는 `terminationMessagePath`를 통해 컨테이너 종료의 메시지 소스를 설정할 수 있습니다. 즉, 컨테이너가 종료되면 Kubernetes는 컨테이너의 `terminationMessagePath` 필드에 지정된 종료 메시지 파일에서 종료 메시지를 검색하고 메시지를 사용하여 컨테이너 종료 메시지를 채웁니다. 메시지의 기본값은 `/dev/termination-log`입니다.

또한 컨테이너의 `terminationMessagePolicy` 필드를 설정하여 종료 메시지를 사용자 지정할 수 있습니다. 이 필드의 기본값은 `File`이며, 이는 종료 메시지가 종료 메시지 파일에서만 검색됨을 의미합니다. 필요에 따라 `FallbackToLogsOnError`로 설정할 수 있습니다. 이는 컨테이너가 오류와 함께 종료되고 종료 메시지 파일이 비어 있는 경우 컨테이너 로그 출력의 마지막 청크가 종료 메시지로 사용됨을 의미합니다.

예시 코드는 다음과 같습니다.



```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: nginx
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
  resources:
    limits:
```

```
cpu: 500m
memory: 1Gi
requests:
  cpu: 250m
  memory: 256Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: FallbackToLogsOnError
```

상기 구성에서 컨테이너가 오류와 함께 종료되고 종료 메시지 파일이 비어 있는 경우 Get Pod는 stderr의 출력이 containerStatuses에 표시되는 것을 찾습니다.

Host 매개변수는 어떻게 사용합니까?

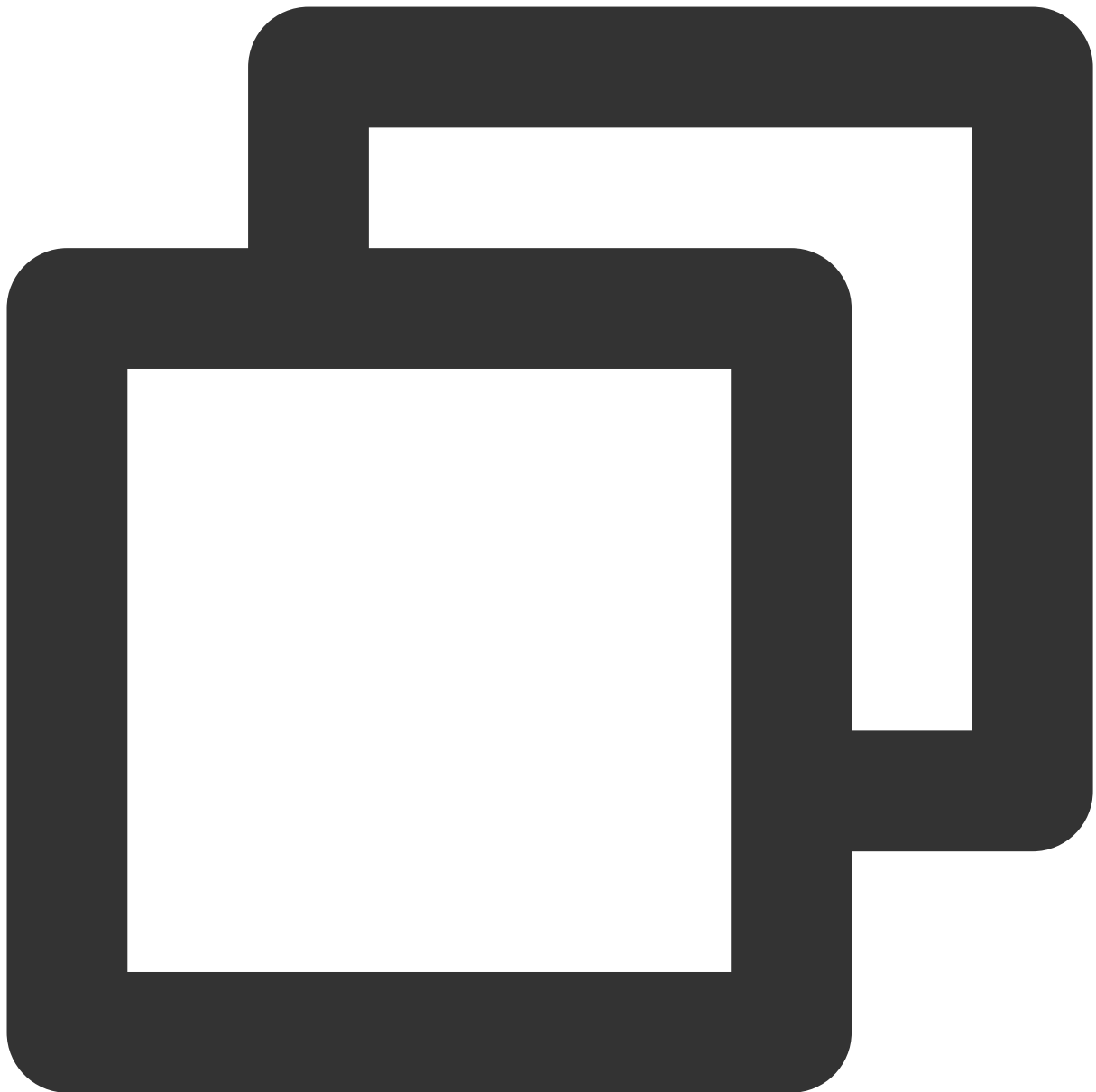
Serverless 클러스터를 사용할 때 다음 사항에 유의하십시오.

Serverless 클러스터에는 노드가 없지만 Hostpath, Hostnetwork: true 및 DnsPolicy: ClusterFirstWithHostNet과 같은 Host 매개변수와 호환됩니다. 노드가 없기 때문에 이러한 매개변수는 k8s의 전체 기능을 제공할 수 없습니다.

예를 들어, Hostpath를 사용하여 데이터를 공유할 수 있지만 동일한 슈퍼 노드에 예약된 두 Pod는 서로 다른 호스트의 Hostpath를 보게 됩니다. 또한 Pod가 재구축되면 Hostpath 파일도 동시에 삭제됩니다.

CFS/NFS를 어떻게 마운트합니까?

Serverless 클러스터에서는 Tencent Cloud의 [Cloud File Storage\(CFS\)](#)를 사용하거나 외부 NFS를 Volume으로 Pod에 마운트하여 데이터를 영구 저장할 수 있습니다. Pod에 CFS/NFS를 마운트하는 YAML 예시는 다음과 같습니다.



```
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /cache
      name: cache-volume
```

```
volumes:  
- name: nfs  
  nfs:  
    path: /dir  
    server: 127.0.0.1  
---
```

`spec.volumes`: 볼륨의 이름, 유형 및 매개변수를 설정합니다.

`spec.volumes.nfs`: NFS/CFS 디스크를 설정합니다.

`spec.containers.volumeMounts`: Pod에서 볼륨의 마운트 포인트를 설정합니다.

Pod에 Volume을 마운트하는 자세한 방법은 [스토리지 볼륨 사용 설명](#)을 참고하십시오.

이미지 재사용으로 컨테이너 실행 속도를 높이려면 어떻게 해야 하나요?

TKE Serverless는 컨테이너 이미지 캐시를 지원하여 동일한 이미지로 컨테이너의 다음 실행 속도를 높입니다.

재사용 조건:

1. 워크로드가 동일한 Pod의 경우 캐시 시간 내에 동일한 가용존(Zone)에서 Pod가 생성되고 종료되면 새 Pod는 기본적으로 동일한 이미지를 풀링하지 않습니다.
2. 다른 워크로드(Deployment, Statefulset 및 Job 포함)가 있는 Pod의 이미지를 재사용하려면 다음 annotation을 사용하십시오.



```
eks.tke.cloud.tencent.com/cbs-reuse-key
```

동일한 사용자 계정에서 동일한 annotation value를 가진 Pod의 경우, 시작 이미지는 캐시 시간 내에서 가능한 한 재사용됩니다. annotation value의 이미지 이름을 입력하는 것이 좋습니다: `eks.tke.cloud.tencent.com/cbs-reuse-key: "image-name"` .

캐시 시간: 2시간.

예외적인 이미지 재사용 문제를 해결하는 방법은 무엇입니까?

이미지 재사용 기능이 활성화되고 Pod가 생성되면 `$kubect1 describe pod` 에서 다음 오류가 발생할 수 있습니다.

```
no space left on device: unknown
```

```
Warning FreeDiskSpaceFailed 26m eklet, eklet-subnet-xxx failed to garbage collect
required amount of images. Wanted to free 4220828057 bytes, but freed 3889267064
bytes
```

솔루션:

작업이 필요하지 않습니다. 몇 분 정도 기다리면 Pod가 자동으로 running됩니다.

원인:

```
no space left on device: unknown
```

Pod가 기본적으로 시스템 디스크를 재사용하는 경우, 시스템 디스크의 원본 이미지가 모든 공간을 차지하고 디스크에 새 이미지를 다운로드할 공간이 부족하여 'no space left on device: unknown' 오류가 보고됩니다. TKE Serverless는 전체 공간이 점유되면 시스템 디스크의 기존 중복 이미지를 자동으로 삭제하여 현재 디스크를 확보하는 일반적인 이미지 회수 메커니즘을 지원합니다.(몇 분 소요됨)

```
Warning FreeDiskSpaceFailed 26m eklet, eklet-subnet-xxx failed to garbage collect
required amount of images. Wanted to free 4220828057 bytes, but freed 3889267064
bytes
```

이 로그는 현재 Pod가 이미지를 다운로드하는 데 4220828057바이트가 필요하지만 현재는 3889267064바이트만 사용할 수 있음을 보여줍니다. 이 event 원인은 디스크에 여러 이미지가 있고 일부 이미지만 해제되었기 때문입니다. TKE Serverless의 일반 이미지 회수 메커니즘은 새 이미지를 성공적으로 가져올 수 있을 때까지 계속 해제됩니다.

외부 nfs를 마운트할 때 Operation not permitted가 보고되면 어떻게 해야 하나요?

영구 스토리지에 외부 nfs를 사용하는 경우 연결이 설정되면 Operation not permitted 이벤트가 보고됩니다. nfs의 /etc/exports 파일을 수정하고 /<path><ip-range>(rw,insecure) 매개변수를 추가해야 합니다. 아래 예시를 참고하십시오.



```
/data/ 10.0.0.0/16(rw,insecure)
```

Pod 디스크가 가득 찬 경우(ImageGCFailed) 어떻게 해야 하나요?

TKE Serverless의 Pod는 기본적으로 20GB의 여유 시스템 디스크 공간을 제공합니다. 디스크가 꽉 찬 경우 다음과 같은 방법으로 디스크를 비울 수 있습니다.

1. 사용하지 않는 컨테이너 이미지 정리

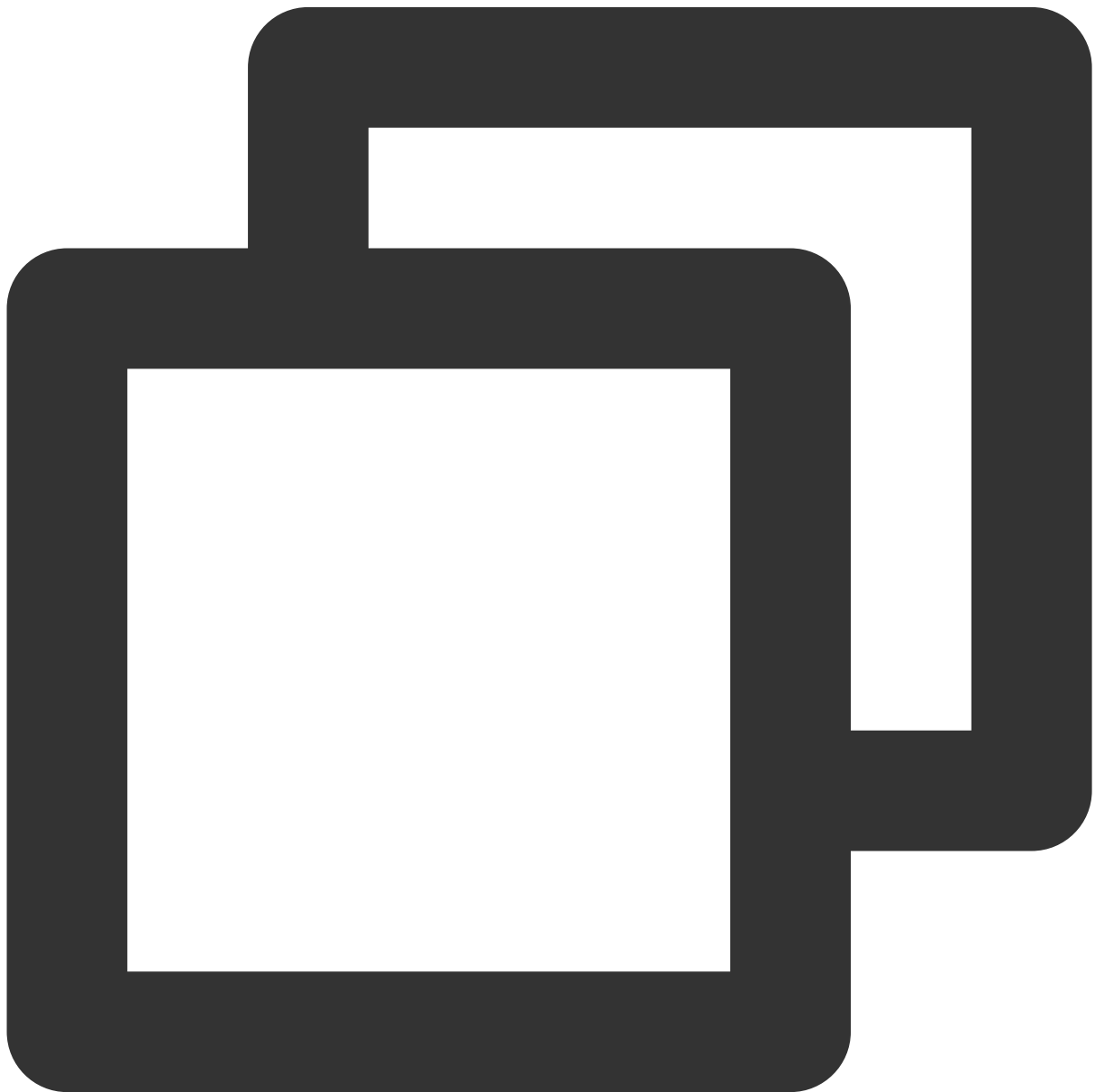
공간의 80%가 사용된 경우 TKE Serverless 백엔드는 컨테이너 이미지 재소유 프로세스를 트리거하여 사용하지 않는 이미지를 복구하고 공간을 확보합니다. 이것이 실패하면 `ImageGCFailed: failed to garbage collect required amount of images`가 보고되어 디스크 공간이 부족함을 알려줍니다.

디스크 공간 부족의 일반적인 원인은 다음과 같습니다.

비즈니스에는 많은 임시 출력이 있습니다. `du` 명령으로 이를 확인할 수 있습니다.

비즈니스는 삭제된 파일 설명자를 보유하므로 디스크 공간이 확보되지 않습니다. `lsdf` 명령으로 이를 확인할 수 있습니다.

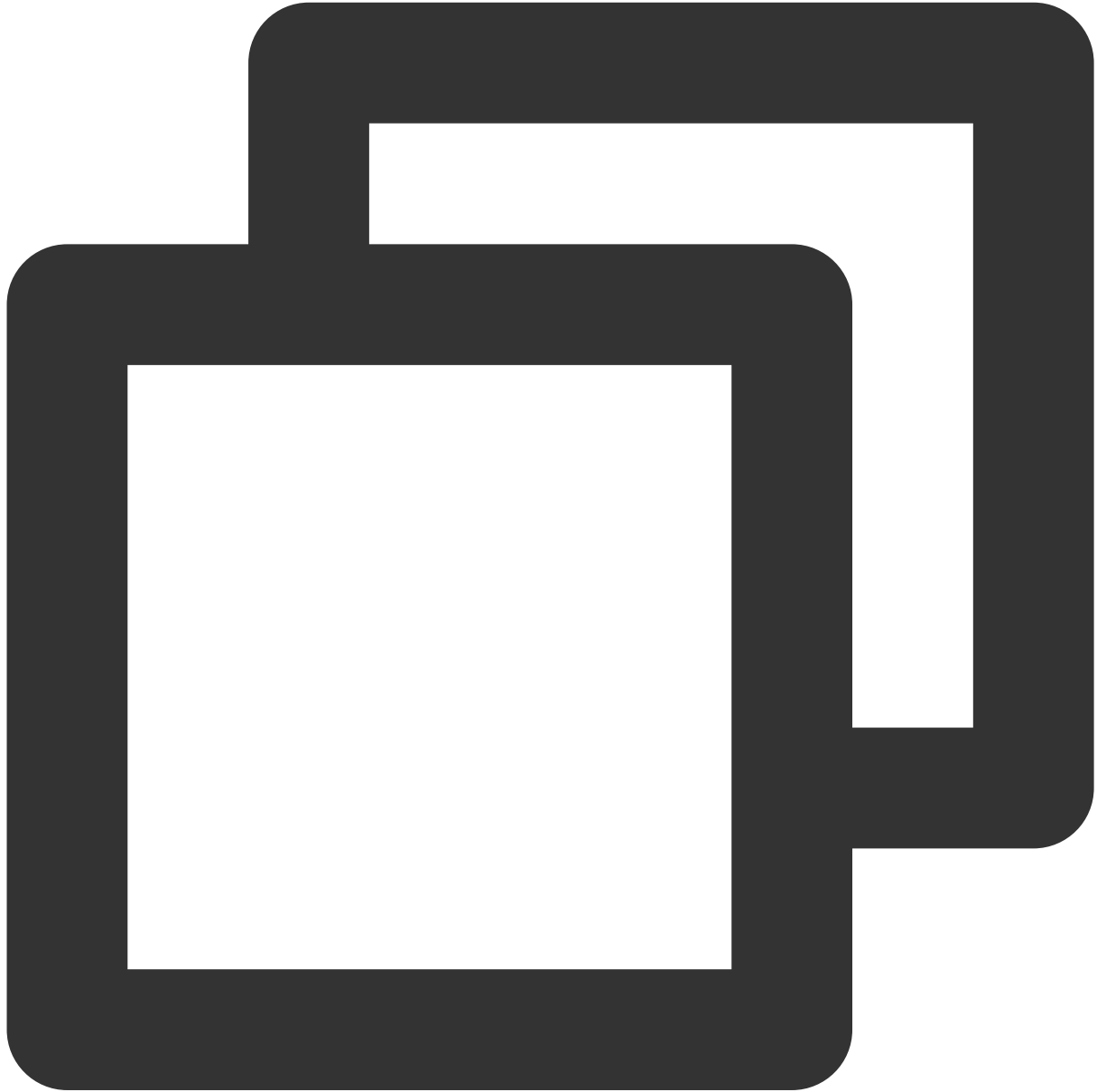
컨테이너 이미지 회수에 대한 임계값을 조정하려면 다음 `annotation`을 설정하십시오.



```
eks.tke.cloud.tencent.com/image-gc-high-threshold: "80"
```

2. 종료된 컨테이너 정리

비즈니스가 인플레이스 업그레이드되었거나 컨테이너가 비정상적으로 종료된 경우 디스크 사용률이 85%에 도달할 때까지 종료된 컨테이너가 유지됩니다. 정리 임계값은 다음 Annotation으로 조정할 수 있습니다.



```
eks.tke.cloud.tencent.com/container-gc-threshold: "85"
```

종료된 컨테이너를 자동으로 정리하지 않으려면(예시: 추가 문제 해결을 위해 종료 정보가 필요함) 다음 Annotation을 사용하여 자동 정리를 비활성화할 수 있습니다. 그러나 이 경우 디스크 공간을 자동으로 확보할 수 없습니다.



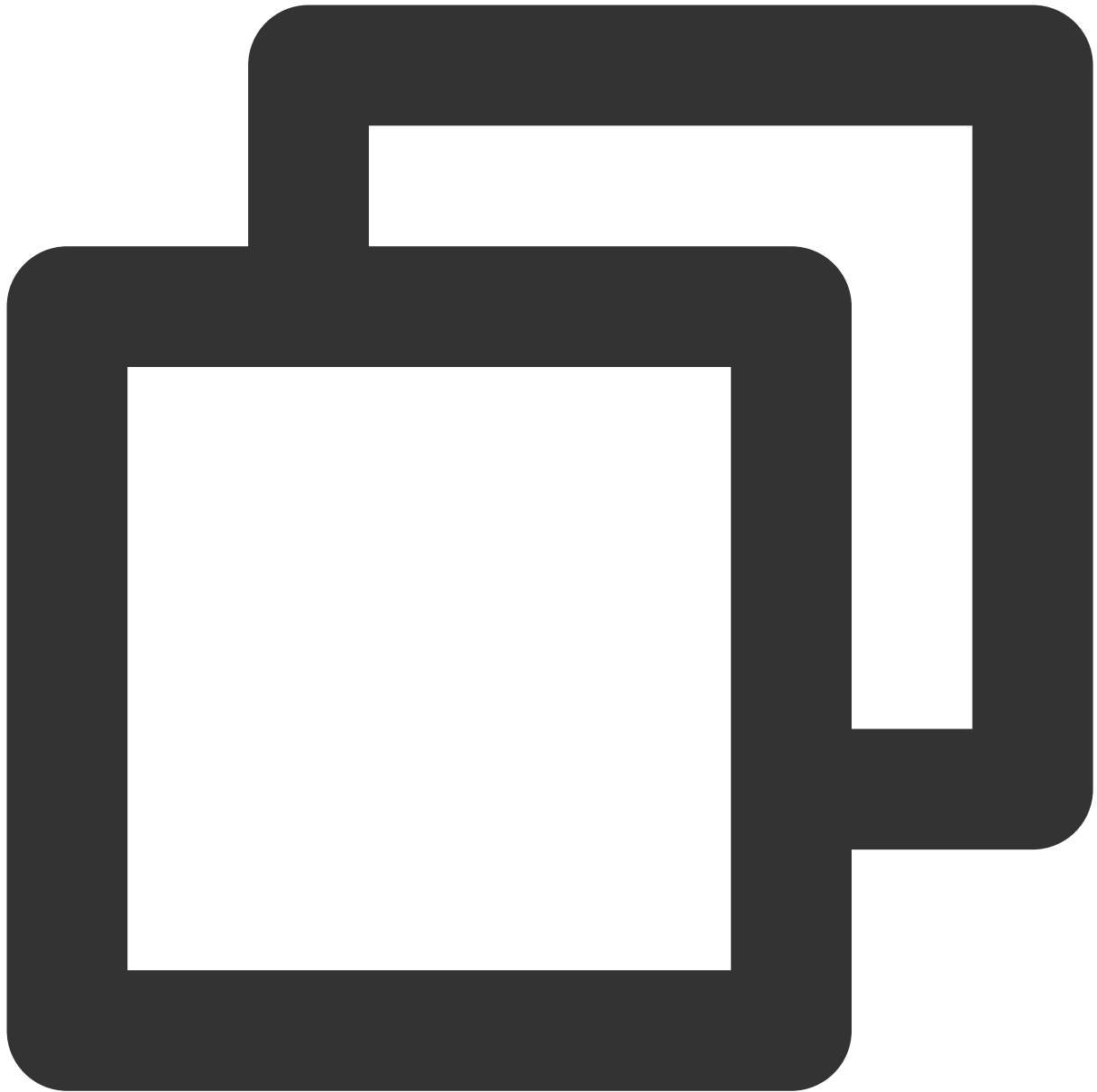
```
eks.tke.cloud.tencent.com/must-keep-last-container: "true"
```

설명

이 기능은 2021-09-15(UTC+8)에 출시되었으며 이전에 생성된 Pod에서는 사용할 수 없습니다.

3. 디스크 사용량이 많은 Pod 재시작

컨테이너의 시스템 디스크 사용량이 특정 비율을 초과한 후에는 다음 Annotation으로 Pod를 재시작해야 합니다.



```
eks.tke.cloud.tencent.com/pod-eviction-threshold: "85"
```

Pod만 재시작되며 호스트는 다시 빌드되지 않습니다. 종료 및 실행에 대해 일반 `gracestop`, `prestop` 및 상태 확인이 수행됩니다.

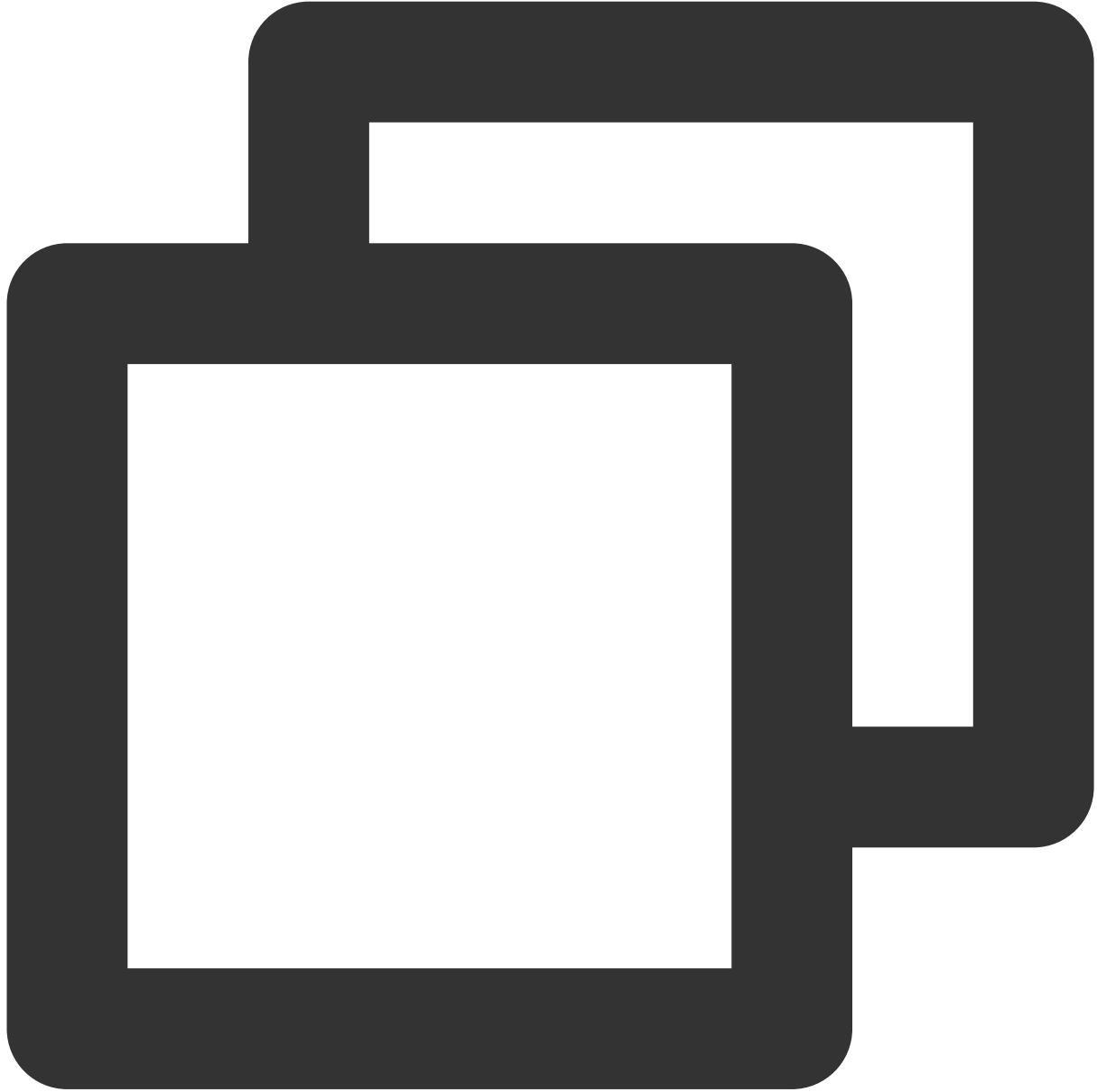
설명

이 기능은 2022-04-27(UTC+8)에 출시되었으며 이전에 생성된 Pod는 다시 빌드된 후에만 활성화할 수 있습니다.

9100 포트 문제

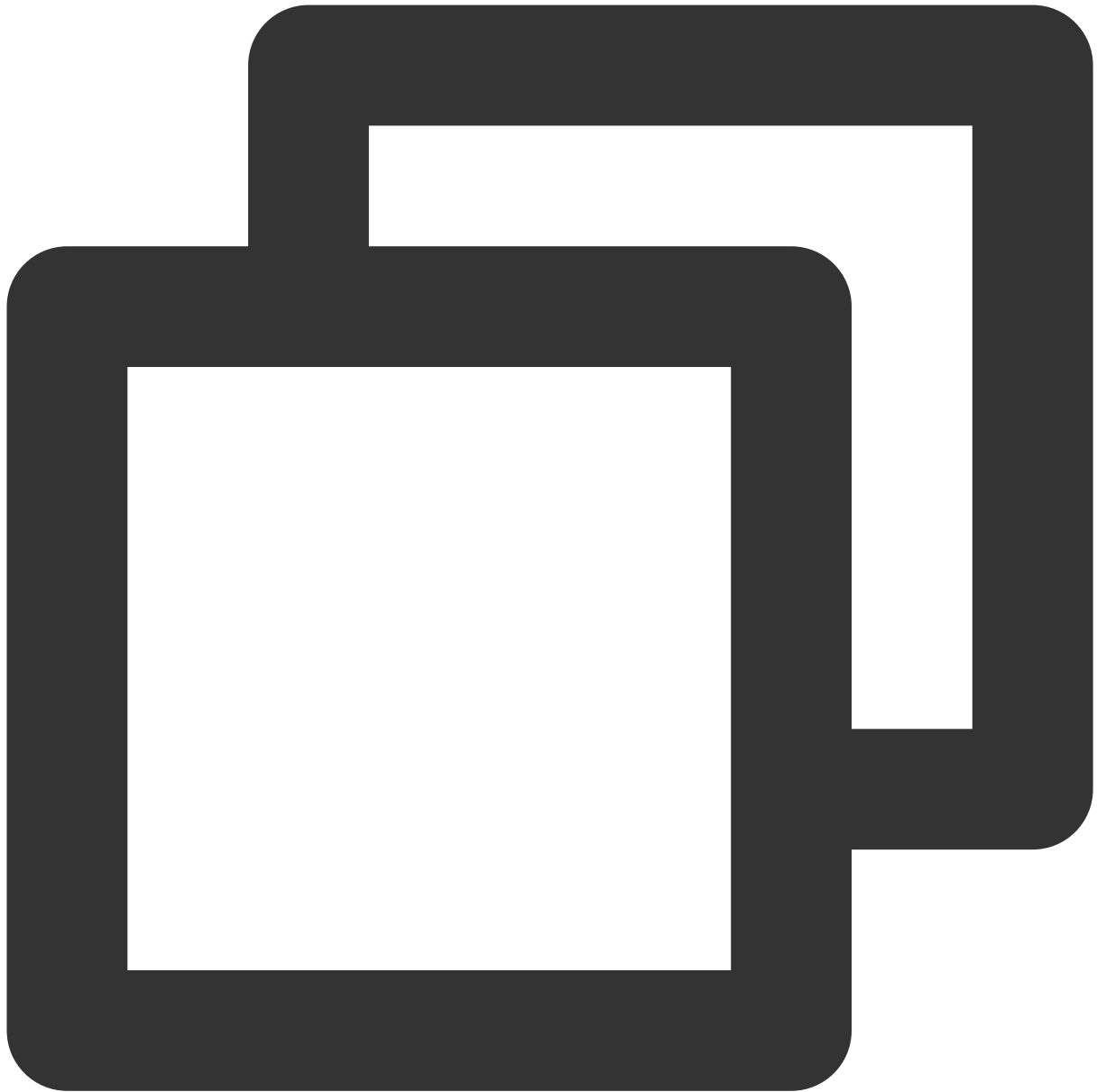
TKE Serverless의 Pod는 기본적으로 포트 9100을 통해 모니터링 데이터를 노출하며 다음 명령을 실행하여 9100/metrics에 액세스하여 데이터를 가져올 수 있습니다.

모든 메트릭 가져오기:



```
curl -g "http://<pod-ip>:9100/metrics"
```

대규모 클러스터의 경우 ipvs 메트릭을 삭제하는 것이 좋습니다.



```
curl -g "http://<pod-ip>:9100/metrics?collect[]=ipvs"
```

비즈니스 필요에 의해 포트 9100에서 수신 대기해야 하는 경우, Pod 생성 시 모니터링 데이터를 수집하기 위해 다른 9100 이외의 포트를 사용하여 9100 포트 충돌을 방지할 수 있습니다. 구성은 아래와 같습니다.



```
eks.tke.cloud.tencent.com/metrics-port: "9110"
```

데이터 노출 모니터링을 위한 포트가 변경되지 않고 비즈니스가 포트 9100에서 직접 수신 대기하는 경우 새 TKE Serverless 네트워크 구성표에 오류가 보고되어 포트 9100이 이미 사용 중임을 나타냅니다.



```
listen() to 0.0.0.0:9100, backlog 511 failed (1: Operation not permitted)
```

이 오류가 보고되면 Annotation: `metrics-port` 를 Pod에 추가하여 모니터링 포트를 변경한 다음 Pod를 다시 빌드해야 합니다.

참고

포드에 공중망 eip가 있는 경우 보안 그룹을 설정해야 합니다. 포트 9100에 주의하고 필요한 포트를 엽니다.

로드 밸런서 FAQ

최종 업데이트 날짜: : 2023-04-26 19:33:50

본문은 CLB의 FAQ와 Service/Ingress CLB의 다양한 FAQ에 대한 원인 및 해결 방법에 대해 설명합니다.

전제 조건:

Pod, Workload, Service, Ingress 등과 같은 K8S [기본 개념](#)에 대해 잘 알고 있습니다.

[TKE 콘솔](#)에서 TKE Serverless 클러스터의 일반 작업에 대해 잘 알고 있습니다.

kubectl 명령 라인 툴을 사용하여 K8S 클러스터의 리소스를 관리할 수 있습니다.

주의사항:

다양한 방법으로 K8S 클러스터 리소스를 관리할 수 있습니다. 본문은 Tencent Cloud 콘솔 및 kubectl 명령 라인 툴을 통해 K8S 클러스터 리소스를 관리하는 방법을 설명합니다.

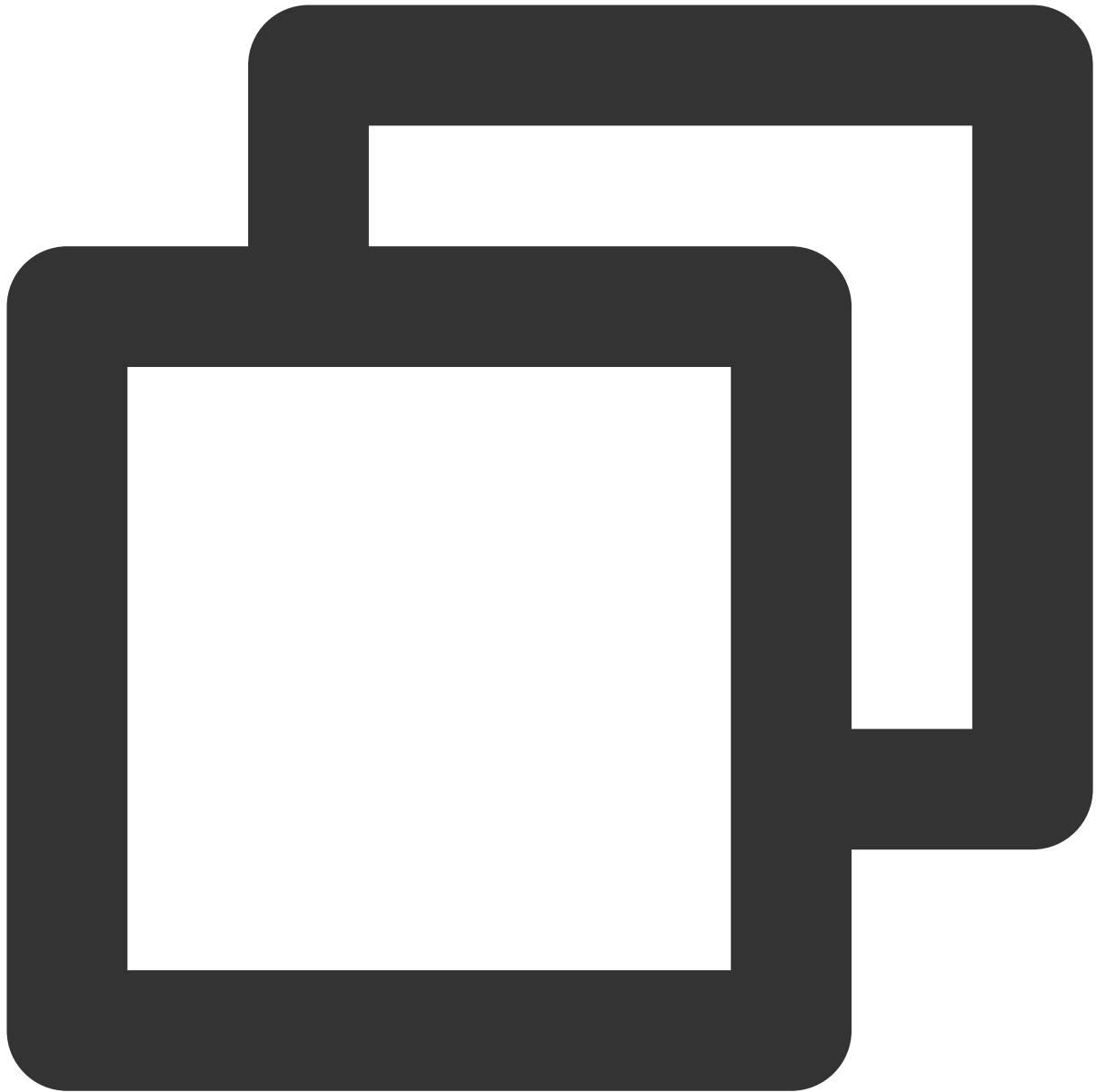
TKE Serverless가 CLB 인스턴스를 생성할 수 있는 Ingress는 무엇입니까?

TKE Serverless는 다음 조건을 충족하는 Ingress용 CLB 인스턴스를 생성합니다.

Ingress 리소스 요구사항	기타 설명
annotations에는 다음 키-값 쌍이 포함됩니다. kubernetes.io/ingress.class: qcloud	TKE Serverless가 Ingress용 CLB 인스턴스를 생성하지 않도록 하려면(예를 들어 Nginx-Ingress를 사용하려는 경우) 키-값 쌍이 annotations에 포함되지 않는지 확인하기만 하면 됩니다.

Ingress용 TKE Serverless에서 생성한 CLB 인스턴스를 보려면 어떻게 해야 합니까?

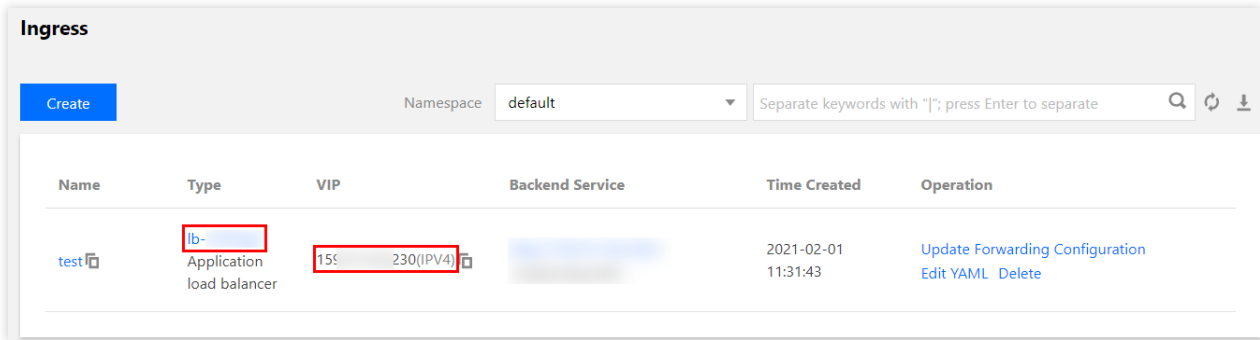
TKE Serverless가 Ingress용 CLB 인스턴스를 성공적으로 생성한 경우 CLB 인스턴스의 VIP를 Ingress 리소스의 `status.loadBalancer.ingress`에 입력하고 다음 키-값 쌍을 annotations에 입력합니다.



```
kubernetes.io/ingress.qcloud-loadbalance-id: CLB 인스턴스 ID
```

Ingress용 TKE Serverless에서 생성한 CLB 인스턴스를 보려면 다음을 수행하십시오.

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 **Cluster**를 선택합니다.
2. 클러스터 목록 페이지에서 대상 클러스터의 ID를 클릭하여 클러스터 관리 페이지로 이동합니다.
3. 클러스터 관리 페이지의 왼쪽 사이드바에서 **서비스 및 라우팅 > Ingress**를 선택합니다.
4. 'Ingress' 페이지에서 CLB 인스턴스 ID와 해당 VIP를 찾을 수 있습니다.



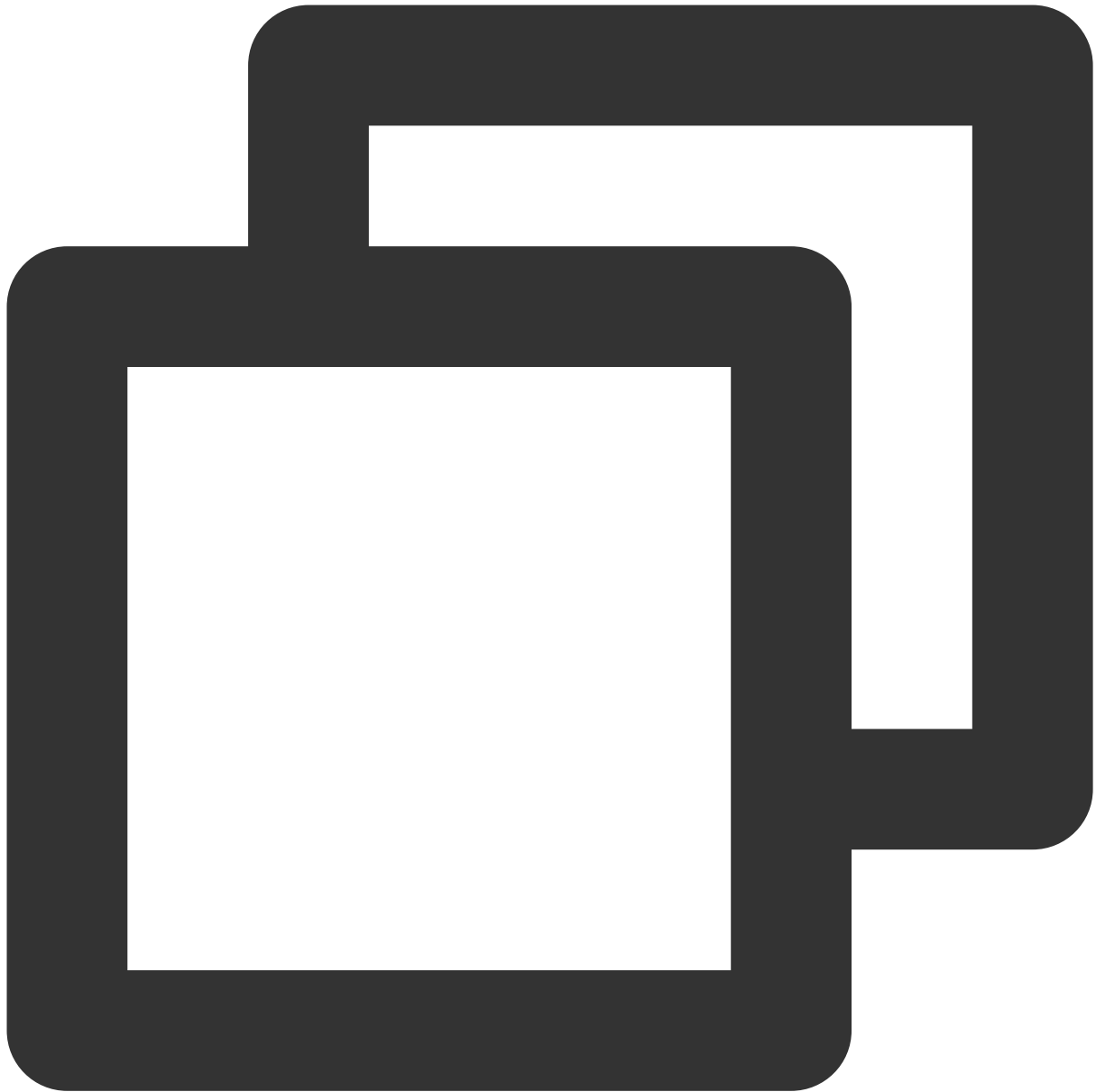
TKE Serverless가 CLB 인스턴스를 생성할 수 있는 Service는 무엇입니까?

TKE Serverless는 다음 조건을 충족하는 Service에 대한 CLB 인스턴스를 생성합니다.

K8S 버전	Service 리소스에 대한 요구 사항
TKE Serverless에서 지원하는 모든 K8S 버전	spec.type은 LoadBalancer입니다
K8S의 수정된 버전(kubectl version에서 반환된 Server GitVersion에는 "eks." 또는 "tke." 접미사가 있음)	spec.type은 ClusterIP이고, spec.clusterIP의 값은 None이 아니며 Headless가 아닌 ClusterIP Service를 나타냅니다
K8S의 수정되지 않은 버전 (kubectl version에서 반환된 Server GitVersion에는 "eks." 또는 "tke." 접미사가 없음)	spec.type은 ClusterIP이고 spec.clusterIP는 빈 문자열 ("")로 지정됩니다

주의사항:

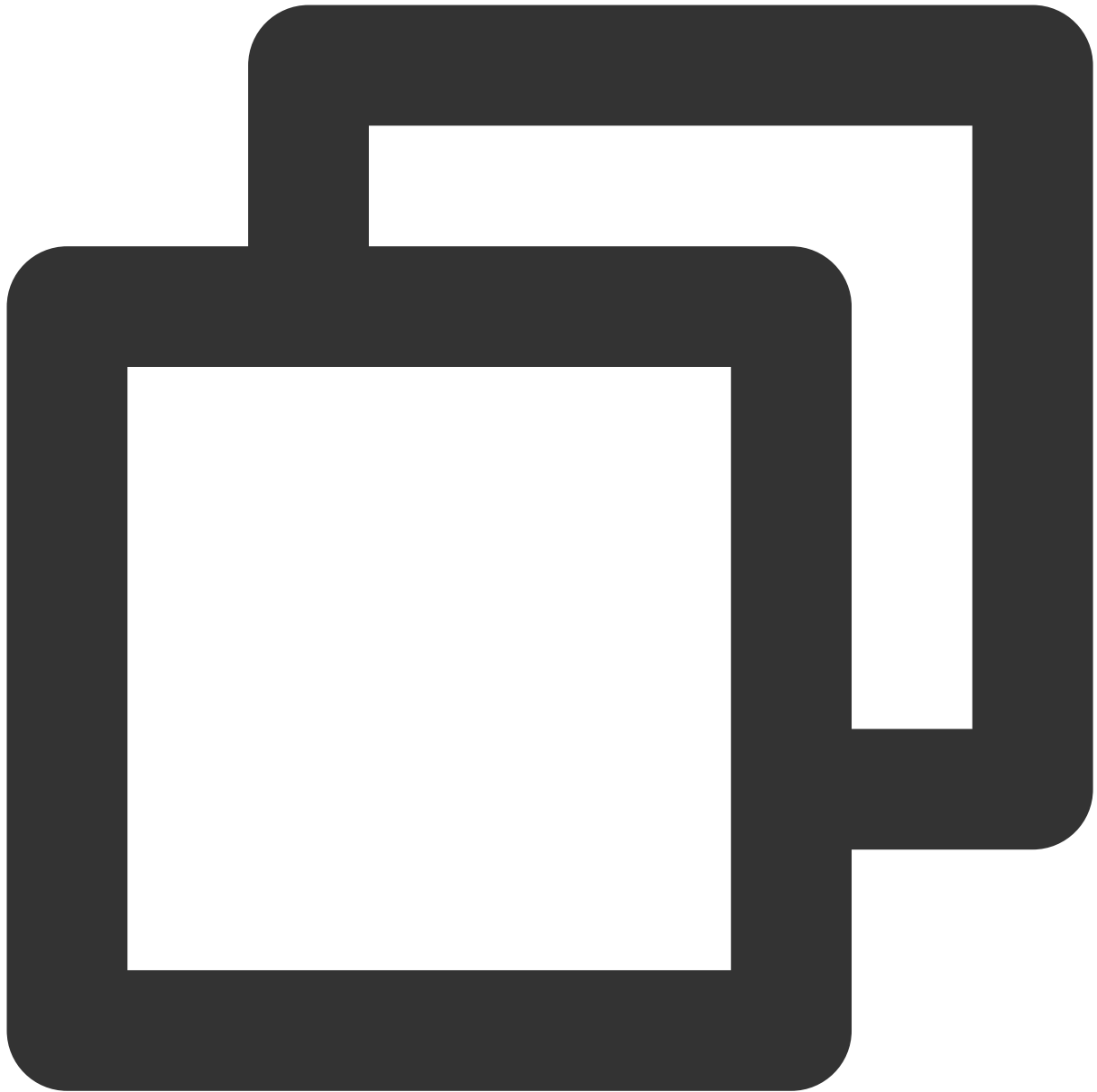
CLB 인스턴스가 성공적으로 생성되면 TKE Serverless는 다음 키-값 쌍을 Service annotations에 입력합니다.



```
service.kubernetes.io/loadbalance-id: CLB 인스턴스 ID
```

Service용 TKE Serverless에서 생성한 CLB 인스턴스를 보려면 어떻게 해야 하나요?

TKE Serverless가 Service용 CLB 인스턴스를 성공적으로 생성한 경우 CLB 인스턴스의 VIP를 Service 리소스의 `status.loadBalancer.ingress` 에 입력하고 다음 키-값 쌍을 annotations에 입력합니다.



```
kubernetes.io/ingress.qcloud-loadbalance-id: CLB 인스턴스 ID
```

TKE Serverless for Service에서 생성한 CLB 인스턴스를 보려면 다음을 수행하십시오.

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 **Cluster**를 선택합니다.
2. 클러스터 목록 페이지에서 대상 클러스터의 ID를 클릭하여 클러스터 관리 페이지로 이동합니다.
3. 클러스터 관리 페이지의 왼쪽 사이드바에서 **서비스 및 라우팅 > Service**를 선택합니다.
4. **Service** 페이지에서 CLB 인스턴스 ID와 해당 VIP를 찾을 수 있습니다.

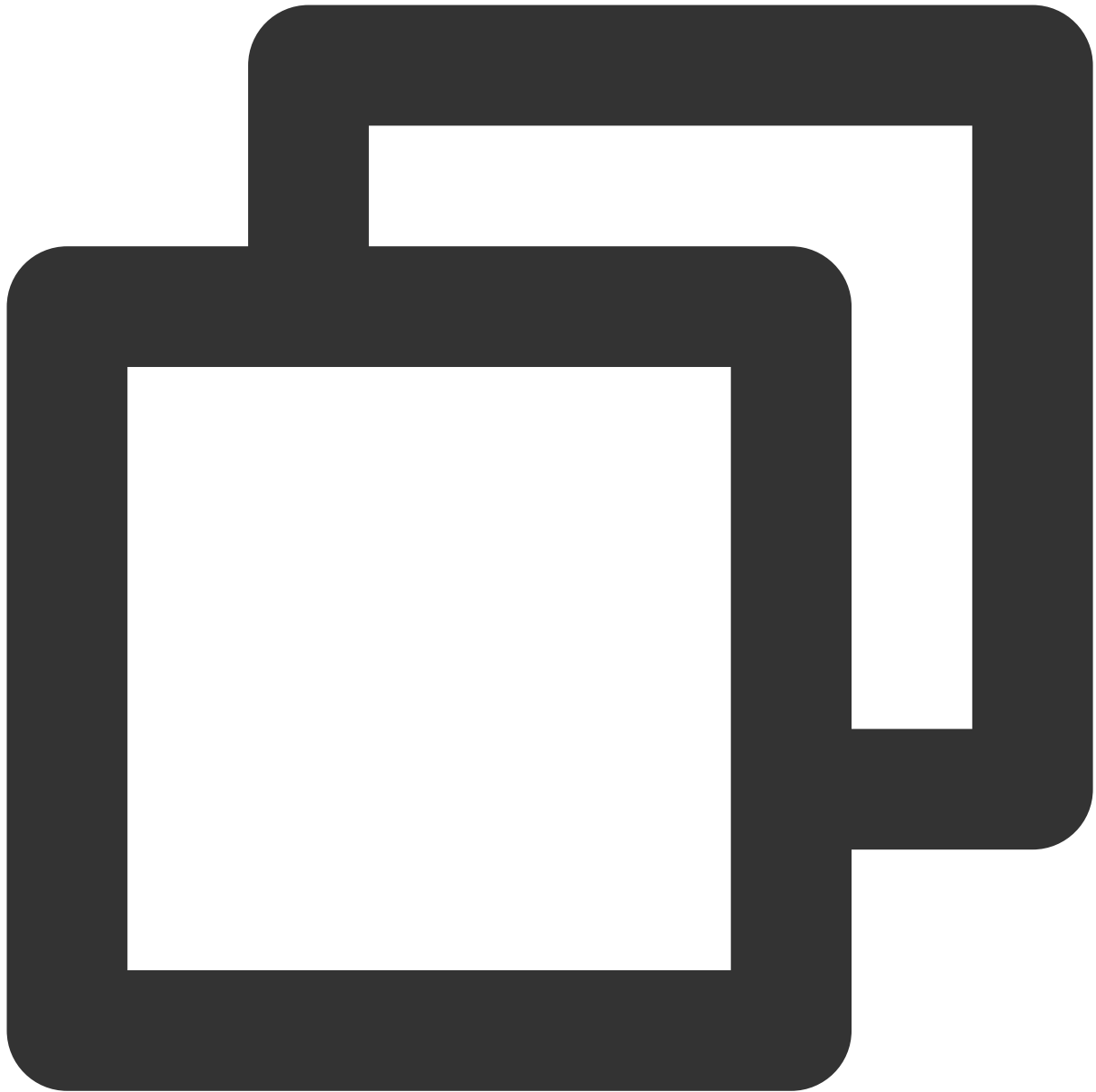
Service

Create Namespace default Separate keywords with "; press Enter to separate

Name	Type	Selector	IP Address ⓘ	Time Created	Operation
kubernetes	ClusterIP	N/A	-(Service IP)		Update access method Edit YAML Delete
test	lb- Load Balancer	k8s-app:test, qclo...	42.18(IPV4)		Update access method Edit YAML Delete

Service의 ClusterIP가 유효하지 않거나(정상적으로 액세스할 수 없음) ClusterIP가 없는 이유는 무엇인가요?

spec.type이 LoadBalancer인 Service의 경우 현재 TKE Serverless는 기본적으로 ClusterIP를 할당하지 않거나 할당된 ClusterIP가 유효하지 않습니다(정상적으로 액세스할 수 없음). 사용자가 ClusterIP를 사용하여 Service에 액세스해야 하는 경우 annotations에 다음 키-값 쌍을 추가하여 TKE Serverless가 사설망 CLB를 기반으로 ClusterIP를 구현함을 나타낼 수 있습니다.



```
service.kubernetes.io/qcloud-clusterip-loadbalancer-subnetid: Service CIDR 서브넷 ID
```

클러스터를 생성할 때 지정하는 Service CIDR 서브넷 ID는 `subnet-*****` 형식의 문자열입니다. CLB 기본 정보 페이지에서 서브넷 ID를 확인할 수 있습니다.

주의사항:

수정된 버전의 K8S(kubectl version에서 반환된 Server GitVersion에는 "eks." 또는 "tke." 접미사가 있음)를 사용하는 TKE Serverless 클러스터만 이 기능을 지원합니다. 수정되지 않은 K8S(kubectl version에서 반환된 Server GitVersion에는 "eks." 또는 "tke." 접미사가 없음)버전을 사용하는 이전에 생성된 TKE Serverless 클러스터의 경우 이 기능을 사용하려면 K8S 버전을 업그레이드해야 합니다.

CLB 인스턴스 유형(공중망 또는 사설망)을 어떻게 지정합니까?

TKE 콘솔 또는 Kubectl 명령 라인 툴을 통해 CLB 인스턴스 유형을 지정할 수 있습니다.

TKE 콘솔을 통해 지정

kubectl 명령 라인 툴을 통해 지정

Ingress의 경우 '네트워크 유형'에서 '공중망' 또는 '사설망'를 선택하여 CLB 인스턴스 유형을 지정합니다.

Service의 경우 '서비스 액세스'를 설정하여 CLB 인스턴스 유형을 지정합니다. 'VPC를 통해' 사설망 CLB 인스턴스를 의미합니다.

기본적으로 생성되는 CLB 인스턴스는 '공중망' 유형입니다.

'사설망' 유형의 CLB 인스턴스를 생성하려면 해당 annotation을 Service 또는 Ingress에 추가해야 합니다.

리소스 유형	annotations에 추가할 키-값 쌍
Service	service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: 서브넷 ID
Ingress	kubernetes.io/ingress.subnetId: 서브넷 ID

주의사항:

서브넷 ID는 subnet-***** 형식의 문자열이며, 이 서브넷은 클러스터를 생성 시 '클러스터 네트워크'에 지정된 VPC에 있어야 하며, 이 VPC 정보는 Tencent Cloud 콘솔 클러스터 '기본 정보'에서 확인할 수 있습니다.

기존 CLB 인스턴스를 어떻게 지정합니까?

TKE 콘솔 또는 Kubectl 명령 라인 툴을 통해 기존 CLB 인스턴스를 지정할 수 있습니다.

TKE 콘솔을 통해 지정

Kubectl 명령 라인 툴을 통해 지정

Service 또는 Ingress를 생성할 때 기존 CLB 인스턴스를 사용하려면 '기존 사용'을 선택할 수 있습니다. Service는 Service 생성 후 '액세스 방식 업데이트'를 통해 기존 CLB 인스턴스를 사용하기 위해 '기존 사용'으로 전환할 수 있습니다.

Service/Ingress를 생성하거나 Service를 수정할 때 Service 또는 Ingress에 해당하는 annotation을 추가하면 됩니다.

리소스 유형	annotations에 추가할 키-값 쌍
Service	service.kubernetes.io/tke-existed-lbid: CLB 인스턴스 ID
Ingress	kubernetes.io/ingress.existed-lbid: CLB 인스턴스 ID

주의사항:

'기존 CLB 인스턴스'는 'TKE Serverless for Service 또는 Ingress에 의해 생성된 CLB 인스턴스'일 수 없으며, TKE Serverless는 동일한 기존 CLB 인스턴스를 공유하기 위해 여러 Service/Ingress를 지원하지 않습니다.

CLB 인스턴스의 액세스 로그는 어떻게 확인하나요?

레이어 7 CLB 인스턴스만 액세스 로그 구성을 지원하지만 TKE Serverless for Ingress에서 생성한 레이어 7 CLB 인스턴스의 액세스 로그는 기본적으로 활성화되어 있지 않습니다. 아래와 같이 CLB 인스턴스의 세부 정보 페이지에서 CLB 인스턴스의 액세스 로그를 활성화할 수 있습니다.

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 **Cluster**를 선택합니다.
2. 클러스터 목록 페이지에서 대상 클러스터의 ID를 클릭하여 클러스터 관리 페이지로 이동합니다.
3. 클러스터 관리 페이지의 왼쪽 사이드바에서 **서비스 및 라우팅 > Ingress**를 선택합니다.
4. 'Ingress' 페이지에서 CLB 인스턴스 ID를 클릭하면 CLB 기본 정보 페이지로 이동합니다.


Name	Type	VIP	Backend Service	Time Created	Operation
test	lb- Application load balancer				Update Forwarding Configuration Edit YAML Delete

5. CLB 기본 정보 페이지의 '액세스 로그(레이어 7)' 섹션에서

을(를) 클릭하여 다음과 같이 CLB 액세스 로그를 활성화합니다.

Access Log (Layer-7)

Access logs can only be configured for layer-7 (HTTP/HTTPS) listeners but not for layer-4 (TCP/UDP) listeners.

Cloud Log Service ⓘ Not Enabled 

TKE Serverless가 Ingress 또는 Service용 CLB 인스턴스를 생성하지 않은 이유는 무엇입니까?

TKE Serverless가 CLB 인스턴스를 생성할 수 있는 Ingress는 무엇입니까? 및 TKE Serverless가 CLB 인스턴스를 생성할 수 있는 Service는 무엇입니까?를 참고하여 해당 리소스가 CLB 인스턴스를 생성할 수 있는 조건을 가지고 있는지 확인합니다. 조건이 충족되었지만 CLB 인스턴스가 성공적으로 생성되지 않은 경우 `kubectl describe` 명령을 사용하여 '리소스'의 관련 이벤트를 볼 수 있습니다.

일반적으로 TKE Serverless는 관련 Warning 이벤트를 출력합니다. 다음 예시에서 출력 이벤트는 서브넷에 사용 가능한 IP 리소스가 없으므로 CLB 인스턴스를 성공적으로 생성할 수 없음을 나타냅니다.

```
[flyma@VM_227_154_centos ~]$ kubectl describe svc -n demo normal-clusterip-nginx
Name: normal-clusterip-nginx
Namespace: demo
Labels: <none>
Annotations: service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: subnet-j55z6ay7
Selector: k8s-app=nginx,qcloud-app=nginx
Type: ClusterIP
IP:
Port: tcp-80-80 80/TCP
TargetPort: 80/TCP
Endpoints: 10.0.24.70:80,10.0.24.81:80,10.1.0.174:80
Session Affinity: None
Events:
  Type Reason Age From Message
  ---
Warning CreatingLoadBalancerFailed 46m clb-service-controller Failed to synclbWithService (will retry): failed to cer for 'service:demo/normal-clusterip-nginx', [TencentCloudSDKError] Code=ResourceInsufficent, Message=The number of IP in subnet subne
RequestId=3c6f9d95-650d-4e98-8171-006fbb0d0f88
Warning CreatingLoadBalancerFailed 29m clb-service-controller Failed to synclbWithService (will retry): failed to cer for 'service:demo/normal-clusterip-nginx', [TencentCloudSDKError] Code=ResourceInsufficent, Message=The number of IP in subnet subne
RequestId=8567bab4-3f57-4f3d-9e99-f34d8c28d5e8
```

동일한 CLB를 여러 Service에서 어떻게 사용하나요?

TKE Serverless 클러스터의 경우 여러 Service가 기본적으로 동일한 CLB 인스턴스를 공유할 수 없습니다. Service가 다른 Service가 차지하는 CLB를 사용하기를 원하는 경우 이 annotation을 추가하고 value을 "true"로 지정하십시오.

```
service.kubernetes.io/qcloud-share-existed-lb: true
```

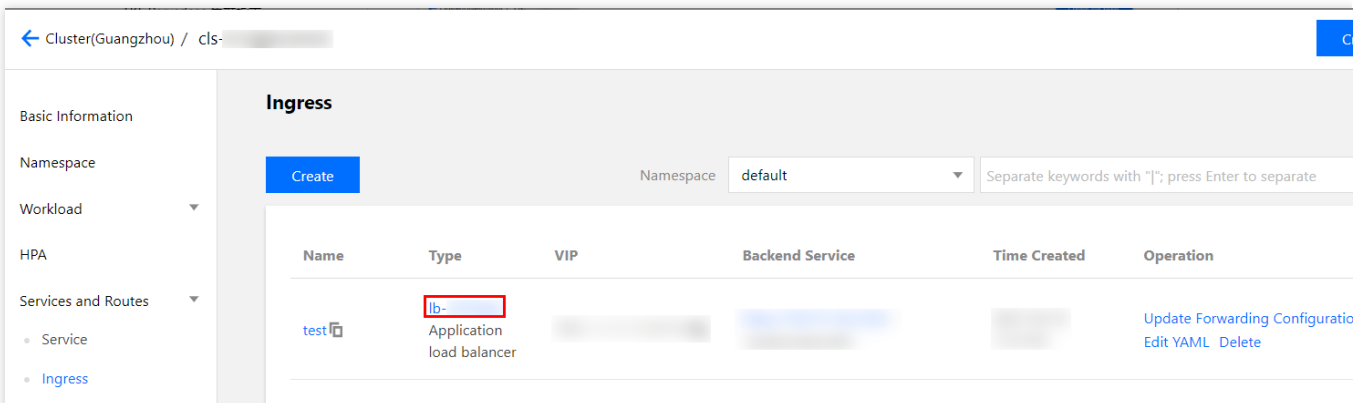
이 annotation에 대한 자세한 내용은 [Annotation](#)을 참고하십시오.

왜 CLB VIP 액세스에 실패하나요?

분석하려면 아래 단계를 따르십시오.

CLB 인스턴스 유형 보기

1. 'Service' 또는 'Ingress' 페이지에서 CLB 인스턴스 ID를 클릭하면 CLB 기본 정보 페이지로 이동합니다.



2. CLB 기본 정보 페이지에서 상기 CLB 인스턴스의 '인스턴스 유형'을 확인할 수 있습니다.

CLB VIP 액세스 환경 정상 여부 확인

CLB 인스턴스의 '인스턴스 유형'이 사설망인 경우 해당 VIP는 자신이 속한 VPC에서만 액세스할 수 있습니다. TKE Serverless 클러스터에 있는 Pods의 IP는 VPC에 있는 ENI IP이므로 Pods에서 클러스터에 있는 Service 또는 Ingress의 CLB 인스턴스 VIP에 액세스할 수 있습니다.

참고

LoadBalancer 시스템에는 항상 루프백 문제(예시: [AzureLoad Balancer 문제 해결](#))가 있습니다. 이 워크로드가 속한 Pods에서 자체적으로(Service 또는 Ingress를 통해) 열린 VIP를 통해 워크로드가 제공하는 서비스에 액세스하지 마십시오. 즉, Pods는 VIP를 통해 자체적으로 제공되는 서비스('사설망' 및 '공중망' 포함)에 액세스해서는 안 됩니다. 그렇지 않으면 VIP에 해당하는 규칙에 따라 RS/Pod가 하나만 있는 경우 액세스 지연이 증가하거나 액세스가 차단될 수 있습니다.

CLB 인스턴스의 '인스턴스 유형'이 공중망인 경우 공중망 액세스가 활성화된 환경에서 VIP에 액세스할 수 있습니다.

클러스터에서 공중망 VIP에 액세스하려면 NAT 게이트웨이 또는 기타 방법을 구성하여 클러스터에 대한 공개 네트워크 액세스가 활성화되었는지 확인하십시오.

예상되는 Pods의 IP + 포트를 포함하여(및 포함) CLB에서 RS 보기

CLB 관리 페이지에서 **리스너 관리** 탭을 선택하여 포워딩 규칙(레이어 7 프로토콜) 및 바인딩된 백엔드 서비스(레이어 4 프로토콜)를 봅니다. IP 주소는 각 Pod의 IP로 예상됩니다. Ingress에 대해 TKE Serverless에서 생성한 CLB의 예시는 다음과 같습니다.

The screenshot shows the 'Listener Management' tab for a CLB instance. A note at the top states: "Note: When custom redirection policies are configured, the original forwarding rules are modified, the redirection policies will be removed automatically. You need to configure it again. See details". Below this, the 'HTTP/HTTPS Listener' section is visible, with a 'Create' button. The listener is named 'TKE-DEDICATED-LISTENER(HTTP:80)' and has a 'Default Access' rule. The 'Forwarding Rules' section is expanded, showing a 'Bound Real Server' with a 'Bind' button and 'Modify Port', 'Modify Weight', and 'Unbind' options. A table below lists the bound real servers:

<input type="checkbox"/>	CVM ID/Name	Port Health Status	IP Address	Port	Weight	Oper...
<input type="checkbox"/>	ins- as-tke-np-azvii3n8	 	[Redacted]	80	10	Unbind

해당 Endpoints 정상 여부 확인

워크로드(Workload)의 레이블(Labels) 및 Service 리소스의 선택기(Selector)를 올바르게 설정한 경우, 워크로드의 Pods가 성공적으로 실행되면 K8S에 의해 Pods가 Service에 해당하는 Endpoints의 준비된 IP 목록으로 추가되는 것을 `kubectl get endpoints` 명령을 통해 확인할 수 있습니다. 예시는 다음과 같습니다.

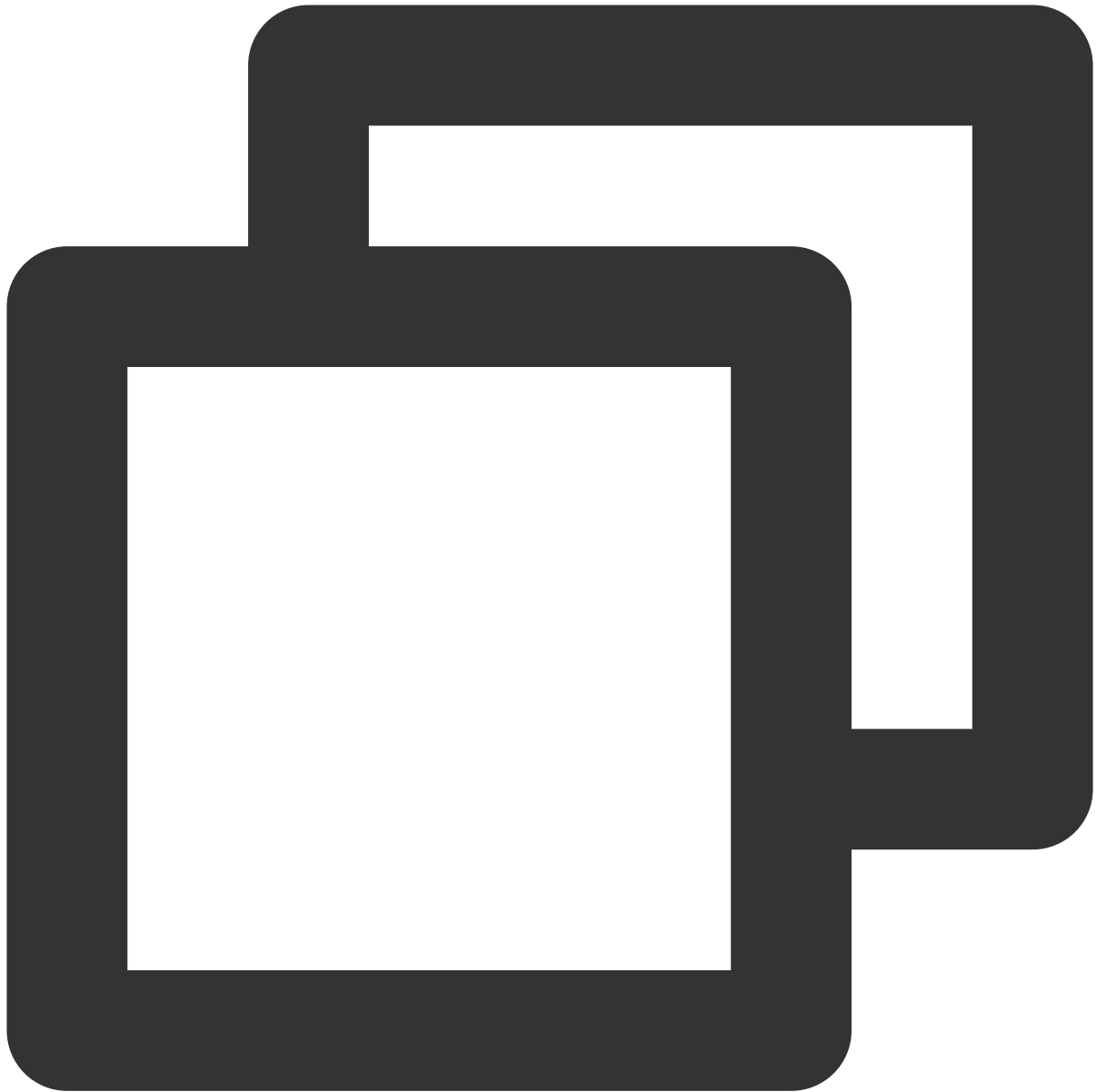
```
[flyma@VM_227_154_centos ~]$ kubectl get pod -n demo -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP              NODE                                                    NOMINATED NODE   READINESS GATES
nginx-7c7c647ff7-4b8n5  1/1    Running   0           18h   10.0.24.70     cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0        <none>            <none>
nginx-7c7c647ff7-b95zc  1/1    Running   0           18h   10.0.24.81     cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0        <none>            <none>
nginx-7c7c647ff7-ckhb9  1/1    Running   0           18h   10.1.0.174     cls-4plj5ho0-virtual-kubelet-subnet-l9dkwcyx-0        <none>            <none>
[flyma@VM_227_154_centos ~]$ kubectl get deployment -n demo nginx
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx         3/3     3             3           18h
[flyma@VM_227_154_centos ~]$ kubectl get svc -n demo internal-clb-nginx
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
internal-clb-nginx  LoadBalancer  10.0.15.28   10.0.15.28   80:30082/TCP    18h
[flyma@VM_227_154_centos ~]$ kubectl get ep -n demo internal-clb-nginx
NAME          ENDPOINTS
internal-clb-nginx  10.0.24.70:80,10.0.24.81:80,10.1.0.174:80 18h
[flyma@VM_227_154_centos ~]$
```

생성되었지만 비정상 상태인 Pods는 Service에 해당하는 Endpoints의 준비되지 않은 IP 목록에 K8S에 의해 추가됩니다. 예는 다음과 같습니다.

```
[flyma@VM_227_154_centos ~]$ kubectl get pod -n demo -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP              NODE                                                    NOMINATED NODE   READINESS GATES
nginx-7c7c647ff7-4b8n5  0/1    ImagePullBackOff  0           18h   10.0.24.70     cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0        <none>            <none>
nginx-7c7c647ff7-b95zc  1/1    Running    0           18h   10.0.24.81     cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0        <none>            <none>
nginx-7c7c647ff7-ckhb9  1/1    Running    0           18h   10.1.0.174     cls-4plj5ho0-virtual-kubelet-subnet-l9dkwcyx-0        <none>            <none>
[flyma@VM_227_154_centos ~]$ kubectl get ep -n demo internal-clb-nginx -o yaml
apiVersion: v1
kind: Endpoints
metadata:
  annotations:
    endpoints.kubernetes.io/last-change-trigger-time: "2020-12-22T11:41:25+08:00"
  creationTimestamp: "2020-12-21T08:58:22Z"
  name: internal-clb-nginx
  namespace: demo
  resourceVersion: "2482053203"
  selfLink: /api/v1/namespaces/demo/endpoints/internal-clb-nginx
  uid: 7893ab4c-f8b4-45d9-a4df-21daabb1c42e
subsets:
- addresses:
  - ip: 10.0.24.81
    nodeName: cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0
    targetRef:
      kind: Pod
      name: nginx-7c7c647ff7-b95zc
      namespace: demo
      resourceVersion: "2475611832"
      uid: 38c40d5f-9ab6-4647-9e75-642b8203268e
  - ip: 10.1.0.174
    nodeName: cls-4plj5ho0-virtual-kubelet-subnet-l9dkwcyx-0
    targetRef:
      kind: Pod
      name: nginx-7c7c647ff7-ckhb9
      namespace: demo
      resourceVersion: "2475610295"
      uid: c839e449-aa5b-4cf7-a5ff-a35df7262811
  notReadyAddresses:
  - ip: 10.0.24.70
    nodeName: cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0
    targetRef:
      kind: Pod
      name: nginx-7c7c647ff7-4b8n5
      namespace: demo
      resourceVersion: "2482053202"
      uid: f5d2a2d3-676d-4323-98dd-4aaf168e6e22
ports:
- name: tcp-80-80
  port: 80
  protocol: TCP
[flyma@VM_227_154_centos ~]$
```

주의사항:

비정상적인 Pods의 원인을 볼 수 있도록 kubectl describe 명령을 실행할 수 있습니다. 명령은 다음과 같습니다.



```
kubectl describe pod nginx-7c7c647ff7-4b8n5 -n demo
```

Pods가 정상적으로 서비스를 제공할 수 있는지 확인

Running 상태의 Pods조차도 일부 예외로 인해 일반적으로 서비스를 제공하지 못할 수 있습니다. 예를 들어, 지정된 프로토콜+포트는 미수신, Pods의 내부 로직 오류, 프로세스 차단 등이 있습니다. `kubectl exec` 명령을 실행하여 Pod에 로그인하고 `telnet/wget/curl` 명령을 실행하거나 사용자 지정 클라이언트 툴을 사용하여 Pod IP+포트에 직접 액세스 할 수 있습니다. Pod에서 직접 액세스가 실패하면 Pod가 정상적으로 서비스를 제공할 수 없는 이유를 추가로 분석해야 합니다.

Pod에 바인딩 된 보안 그룹이 Pods가 제공하는 서비스의 프로토콜과 포트를 허용하는지 확인

보안 그룹은 Linux 서버의 IPTables 규칙과 마찬가지로 Pods의 네트워크 액세스 정책을 제어합니다. 실제 상황에 따라 확인하십시오.

TKE 콘솔을 통해 워크로드 생성

kubectl 명령을 실행하여 워크로드 생성

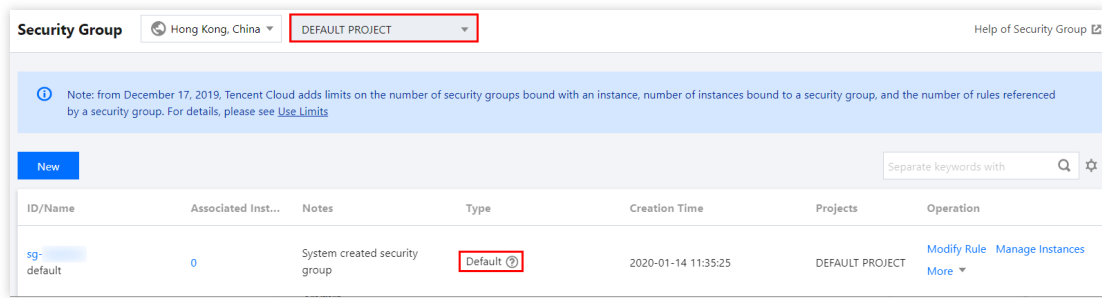
인터랙티브 프로세스는 보안 그룹을 지정해야 하며 TKE Serverless는 이 보안 그룹을 사용하여 Pods의 네트워크 액세스 정책을 제어합니다. 지정된 보안 그룹은 워크로드의 `spec.template.metadata.annotations` 에 저장되고 마지막으로 Pods의 annotations에 추가됩니다. 예시는 다음과 같습니다.

```
[flyma@VM_227_154_centos ~]$ kubectl get pod -n demo nginx-7c7c647ff7-ckhb9 -o yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    eks.tke.cloud.tencent.com/cpu-type: intel
    eks.tke.cloud.tencent.com/security-group-id: sg-7wsmhh5d
  creationTimestamp: "2020-12-21T08:55:17Z"
  generateName: nginx-7c7c647ff7-
  labels:
    k8s-app: nginx
    pod-template-hash: 7c7c647ff7
    qcloud-app: nginx
  name: nginx-7c7c647ff7-ckhb9
  namespace: demo
  ownerReferences:
  - apiVersion: apps/v1
    blockOwnerDeletion: true
    controller: true
    kind: ReplicaSet
    name: nginx-7c7c647ff7
    uid: dc06824c-425c-4229-929d-46d9afe393b6
  resourceVersion: "2475610295"
  selfLink: /api/v1/namespaces/demo/pods/nginx-7c7c647ff7-ckhb9
  uid: c839e449-aa5b-4cf7-a5ff-a35df7262811
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
  resources:
```

kubectl 명령을 통해 워크로드를 생성하고 Pods에 대한 보안 그룹을 지정하지 않는 경우(annotations 추가를 통해)

TKE Serverless는 계정 아래 리전 내에 있는 기본 프로젝트의 default 보안 그룹을 사용합니다. 단계는 다음과 같습니다.

1. VPC 콘솔에 로그인하고 왼쪽 사이드바에서 [보안 그룹](#)을 클릭하십시오.
2. '보안 그룹' 페이지 상단에서 리전 내 기본 프로젝트를 선택합니다.
3. 목록에서 default 보안 그룹을 볼 수 있으며 [규칙 수정](#)을 클릭하여 세부 정보를 볼 수 있습니다.



문의하기

문제가 지속되면 [티켓 제출](#)하여 당사 TKE 팀에 문의하십시오.

이미지 레지스트리

최종 업데이트 날짜: : 2022-03-31 15:04:02

이미지 레지스트리 활성화 FAQ

네임스페이스는 무엇에 사용됩니까?

네임스페이스는 사용자의 개인 이미지를 식별하는 주소 접두사입니다.

이미지 레지스트리 계정이란 무엇입니까?

기본적으로 이 계정은 Tencent Cloud 계정(QQ 계정)입니다.

활성화 중에 설정한 비밀번호를 잊어버린 경우 어떻게 해야 합니까?

콘솔에서 비밀번호를 재설정할 수 있습니다.

클러스터 생성 FAQ

생성된 이미지 수에 할당량 제한이 있습니까?

네. 기본 할당량은 리전당 500개의 이미지 레지스트리와 이미지 레지스트리당 100개의 이미지 태그입니다. 더 많은 할당량이 필요한 경우 [티켓 제출](#)하시기 바랍니다.

내가 만든 이미지를 다른 사용자와 공유할 수 있습니까?

아니요, 이 기능은 현재 지원되지 않습니다.

생성된 이미지는 어떻게 사용하나요?

먼저 사용 가능한 이미지 태그를 업로드한 다음 특정 이미지 태그를 사용하여 서비스를 생성합니다.

이미지 삭제 FAQ

이미지의 특정 태그를 어떻게 삭제합니까?

콘솔에서 직접 특정 태그를 지정하고 삭제할 수 있습니다.

이미지 목록에서 이미지를 삭제하면 이미지의 태그도 모두 삭제되나요?

네. 데이터를 미리 백업해 두십시오.

이미지 구축 FAQ

소스 코드를 사용하여 이미지를 빌드할 때 Dockerfile 경로와 빌드 디렉터리를 어떻게 지정합니까?

- 경로를 입력하지 않으면 시스템은 다음 기본값을 사용합니다.
- 기본 Dockerfile 경로: 리포지토리의 루트 디렉터리 아래에 있는 Dockerfile(`Dockerfile`).
- 기본 빌드 디렉터리: 리포지토리의 루트 디렉터리(`./`).
- 경로를 지정하려면 다음 그림과 같이 루트 경로로 프로젝트 경로를 사용하여 상대 경로를 입력합니다.

←

Build Config

Image Address ccr.ccs.tencentyun.com/test-yunx/helloworld

Code source

Github

Gitlab

Image Tag Naming Rules
 Branch/label - Update Time - Commit No.

Please enter the naming rule - Custom prefix, supports variables in the format of \$(Foo)

Overwrite the image tag Please enter the tag

The generated image also contains the tag

Dockerfile path qcloud/Dockerfile

Path of the Dockerfile in the code source

Building Directory qcloud

The working directory for building, should be a relative path

Building Parameters [Add a variable](#)

Complete

소스 코드 빌드 기능은 Dockerfile 경로와 빌드 디렉터리를 어떻게 사용합니까?

사용자 지정 리포지토리를 clone한 다음 해당 branch 또는 tag로 전환하고 마지막으로 코드 리포지토리의 루트 디렉터리에서 `docker build -f $DOCKERFILE_PATH $WORKDIR` 명령을 실행하여 컨테이너 이미지를 빌드합니다.

Dockerfile에서 소스 경로를 어떻게 지정합니까?

`COPY` , `ADD` 및 소스 경로와 관련된 기타 명령의 경우 소스 경로를 빌딩 디렉터리에 대한 상대 경로로 지정하십시오.

원격 터미널

최종 업데이트 날짜: : 2022-03-31 15:04:02

컨테이너에 bash가 없으면 어떻게 해야 하나요?

컨테이너에 bash가 없으면 명령 라인에 실행하려는 명령을 입력합니다. 그러면 명령의 결과가 화면에 표시됩니다. 명령 라인은 자동 완성과 같은 특정 기능이 없는 단순한 bash로 간주할 수 있습니다. 계속 진행하기 전에 bash 설치 명령을 실행하는 것이 좋습니다.

apt-get을 사용한 소프트웨어 설치가 왜 이렇게 느린가요?

apt-get 을 사용하여 소프트웨어를 설치하는 데 너무 많은 시간이 걸리는 경우 중국 외부의 소프트웨어 소스에 대한 서버 액세스가 느리기 때문일 수 있습니다. 다양한 운영 체제의 컨테이너에 대한 가속화 솔루션을 제공합니다. 필요한 경우 솔루션을 선택할 수 있습니다.

주의사항

가속 솔루션을 선택하기 전에 아래 지침에 따라 컨테이너의 운영 체제를 확인하십시오.

- Ubuntu: `cat /etc/lsb-release` 를 실행하여 `/etc/lsb-release` 파일이 있는지 확인합니다.
- CentOS: `cat /etc/redhat-release` 를 실행하여 `/etc/redhat-release` 파일이 있는지 확인합니다.
- Debian: `cat /etc/debian_version` 을 실행하여 `/etc/debian_version` 파일이 있는지 확인합니다.

솔루션

Ubuntu 16.04

Ubuntu 16.04를 실행하는 컨테이너의 경우 터미널에서 다음 명령을 실행하여 apt 소스를 Tencent Cloud 소스 서버로 설정합니다.

```
cat << EOF > /etc/apt/sources.list
deb http://mirrors.tencentyun.com/ubuntu/ xenial main restricted universe multiverse
deb http://mirrors.tencentyun.com/ubuntu/ xenial-security main restricted universe multiverse
deb http://mirrors.tencentyun.com/ubuntu/ xenial-updates main restricted universe multiverse
deb-src http://mirrors.tencentyun.com/ubuntu/ xenial main restricted universe multiverse
deb-src http://mirrors.tencentyun.com/ubuntu/ xenial-security main restricted universe multiverse
deb-src http://mirrors.tencentyun.com/ubuntu/ xenial-updates main restricted universe multiverse
EOF
```

CentOS 7

CentOS 7을 실행하는 컨테이너의 경우 아래 지침에 따라 소스 주소를 직접 변경하여 설치 속도를 높입니다.

1. 컨테이너에서 다음 코드를 복사하고 실행합니다.

```
cat << EOF > /etc/yum.repos.d/CentOS-Base.repo
[os]
name=Qcloud centos os - \${basearch}
baseurl=http://mirrors.tencentyun.com/centos1/\${releasever}/os/\${basearch}/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
[updates]
name=Qcloud centos updates - \${basearch}
baseurl=http://mirrors.tencentyun.com/centos1/\${releasever}/updates/\${basearch}/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
#[centosplus]
#name=Qcloud centosplus - \${basearch}
#baseurl=http://mirrors.tencentyun.com/centos1/\${releasever}/centosplus/\${basear
ch/
#enabled=1
#gpgcheck=1
#gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
#[cloud]
#name=Qcloud centos contrib - \${basearch}
#baseurl=http://mirrors.tencentyun.com/centos1/\${releasever}/cloud/\${basearch}/ope
nstack-kilo/
#enabled=1
#gpgcheck=1
#gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
#[cr]
#name=Qcloud centos cr - \${basearch}
#baseurl=http://mirrors.tencentyun.com/centos1/\${releasever}/cr/\${basearch}/
#enabled=1
#gpgcheck=1
#gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
[extras]
name=Qcloud centos extras - \${basearch}
baseurl=http://mirrors.tencentyun.com/centos1/\${releasever}/extras/\${basearch}/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
#[fasttrack]
#name=Qcloud centos fasttrack - \basearch
```

```
#baseurl=http://mirrors.tencentyun.com/centos1/\$releasever/fasttrack/\$basearch/
#enabled=1
#gpgcheck=1
#gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
ENDOF
```

2. 다음 명령을 실행하여 YUM 캐시를 지우고 다시 작성하십시오.

```
yum clean all && yum clean metadata && yum clean dbcache && yum makecache
```

Debian

Debian을 실행하는 컨테이너의 경우 터미널에서 다음 명령을 실행하여 apt 소스를 Tencent Cloud 소스 서버로 설정합니다.

```
cat << ENDOF > /etc/apt/sources.list
deb http://mirrors.tencentyun.com/debian stretch main contrib non-free
deb http://mirrors.tencentyun.com/debian stretch-updates main contrib non-free
deb http://mirrors.tencentyun.com/debian-security stretch/updates main
deb-src http://mirrors.tencentyun.com/debian stretch main contrib non-free
deb-src http://mirrors.tencentyun.com/debian stretch-updates main contrib non-free
e
deb-src http://mirrors.tencentyun.com/debian-security stretch/updates main
ENDOF
```

결론

컨테이너에서 소스 주소를 직접 변경하는 것은 임시 솔루션입니다. 컨테이너 스케줄링이 변경되면 변경 사항이 무효화됩니다. 따라서 이미지 생성 시 다음과 같은 방법으로 이 문제를 해결할 것을 권장합니다.

컨테이너 이미지 생성을 위해 Dockerfile 파일의 RUN 필드에 해당 운영 체제용 [솔루션](#)에서 제공하는 소스 주소를 추가합니다. 예를 들어 Ubuntu 운영 체제를 실행하는 이미지의 Dockerfile 파일에 다음 코드를 추가합니다.

```
RUN cat << ENDOF > /etc/apt/sources.list
deb http://mirrors.tencentyun.com/ubuntu/ xenial main restricted universe multiverse
deb http://mirrors.tencentyun.com/ubuntu/ xenial-security main restricted universe multiverse
deb http://mirrors.tencentyun.com/ubuntu/ xenial-updates main restricted universe multiverse
#deb http://mirrors.tencentyun.com/ubuntu/ xenial-proposed main restricted universe multiverse
#deb http://mirrors.tencentyun.com/ubuntu/ xenial-backports main restricted universe multiverse
ENDOF
```

```

rse multiverse
deb-src http://mirrors.tencentyun.com/ubuntu/ xenial main restricted universe mul
tiverse
deb-src http://mirrors.tencentyun.com/ubuntu/ xenial-security main restricted uni
verse multiverse
deb-src http://mirrors.tencentyun.com/ubuntu/ xenial-updates main restricted univ
erse multiverse
#deb-src http://mirrors.tencentyun.com/ubuntu/ xenial-proposed main restricted un
iverse multiverse
#deb-src http://mirrors.tencentyun.com/ubuntu/ xenial-backports main restricted u
niverse multiverse
ENDOF

```

컨테이너에 로그인한 후 vim 및 netstat와 같은 도구를 찾을 수 없으면 어떻게 해야 하나요?

apt-get install vim 및 apt-get install net-tools(CentOS에서 yum install vim 실행)와 같은 명령을 실행하여 필요한 도구를 다운로드합니다.

apt-get install 명령을 실행할 때 도구를 찾을 수 없으면 어떻게 해야 하나요?

다음과 같이 소프트웨어 프로그램을 설치합니다.

1. 다음 명령을 실행하여 소프트웨어 목록을 업데이트합니다.

```
apt-get update
```

2. 다음 명령을 실행하여 소프트웨어 프로그램을 설치합니다(CentOS에서 yum updateinfo 실행).

```
apt-get install
```

컨테이너에서 사내(in-house) 툴을 어떻게 사용하나요?

원격 터미널 페이지로 이동하여 오른쪽 하단 모서리에 있는 파일 도우미를 클릭하고 도구를 업로드합니다.

로컬 분석을 위해 dump 또는 로그와 같은 라이브 파일을 어떻게 업로드하나요?

원격 터미널 페이지로 이동하여 오른쪽 하단 모서리에 있는 파일 도우미를 클릭하고 파일을 업로드합니다.

파일을 컨테이너에 업로드하거나 로컬 시스템에 파일을 다운로드할 수 없는 이유는 무엇입니까?

이 문제는 tar 프로그램이 컨테이너 이미지에 포함되어 있지 않은 경우에 발생합니다. 이를 수정하려면 apt-get install vim 또는 apt-get install net-tools(CentOS에서 yum install vim 실행)를 실행하여 tar 프로그램을 설치하고 다시 시도하십시오.

이전에 설치한 도구를 찾을 수 없는 이유는 무엇입니까?

Kubernetes가 컨테이너를 다시 스케줄링하면 이미지를 가져와서 새 컨테이너를 만들고 이미지에 없는 도구는 새 컨테이너에 설치되지 않습니다. 따라서 이미지를 만들 때 몇 가지 일반적인 문제 해결 도구를 설치하는 것이 좋습니다.

콘솔에서 텍스트를 복사하려면 어떻게 합니까?

강조 표시하고 복사하여 텍스트를 복사할 수 있습니다.

복사한 텍스트를 어떻게 붙여넣나요?

Shift + Insert를 눌러 복사한 텍스트를 붙여넣습니다.

연결이 끊어진 이유는 무엇입니까?

이는 컨테이너 상태를 수정하는 다른 페이지의 컨테이너 또는 CVM에서 일부 작업을 수행하거나 현재 페이지가 3분 이상 유휴 상태로 유지될 때 발생합니다. 두 경우 모두 서버는 연결을 끊습니다.

top과 같은 명령을 실행할 때 TERM environment variable not set 오류가 발생하면 어떻게 해야 합니까?

export TERM linux를 실행합니다.

긴 절대 경로가 있는 디렉터리를 입력할 때 bash 프롬프트에 '<'와 경로의 일부만 표시되는 이유는 무엇입니까?

bash 프롬프트는 기본적으로 'username@hostname 현재 디렉터리'를 표시하도록 설정되어 있습니다. 현재 경로가 너무 길면 bash는 기본적으로 '<'와 경로의 마지막 부분만 표시합니다.