

Tencent Kubernetes Engine

よくある質問

製品ドキュメント



Tencent Cloud

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.

カタログ：

よくある質問

運用保守系

ログ収集関連

クラスター類

スケールアウトスケールイン関連

クラスター関連

ContainerdとDockerの選択方法

サービス類

TKE Serverlessクラスター

TKE Serverlessクラスター関連

Cloud Load Balancer (CLB) 関連

イメージリポジトリ類

リモート端末類

よくある質問

運用保守系

ログ収集関連

最終更新日：：2023-04-26 19:23:11

ログ収集に関する一般的なご質問

クラスターでログ収集を設定後、CLSコンソールでログを確認できないのはなぜですか。

ログを確認できない、またはログが見つからない場合は、以下の問題がないかをチェックしてください。

選択したログtopicのインデックスが有効になっているかをチェックします。インデックスの設定はCloud Log Service (CLS) を使用して検索と分析を行うための必須条件です。有効になっていない場合はログを確認できません。インデックス設定の詳細な操作については[CLSのインデックス設定](#)をご参照ください。



ログ、審査、イベントが同じtopicを使用していないかをチェックします。同じtopicを使用するとログが上書きされ、削除されます。

選択したtopicが2種類の抽出モードを使用していないかをチェックします。古い抽出モードは新しい抽出モードによって上書きされます。

ソフトリンクが存在しないかをチェックします。選択した収集タイプが「コンテナファイルパス」の場合、対応する「コンテナファイルパス」をソフトリンクにすることはできず、これを行った場合はソフトリンクの実際のパスが収集ツール内のコンテナに存在しなくなり、ログ収集に失敗します。

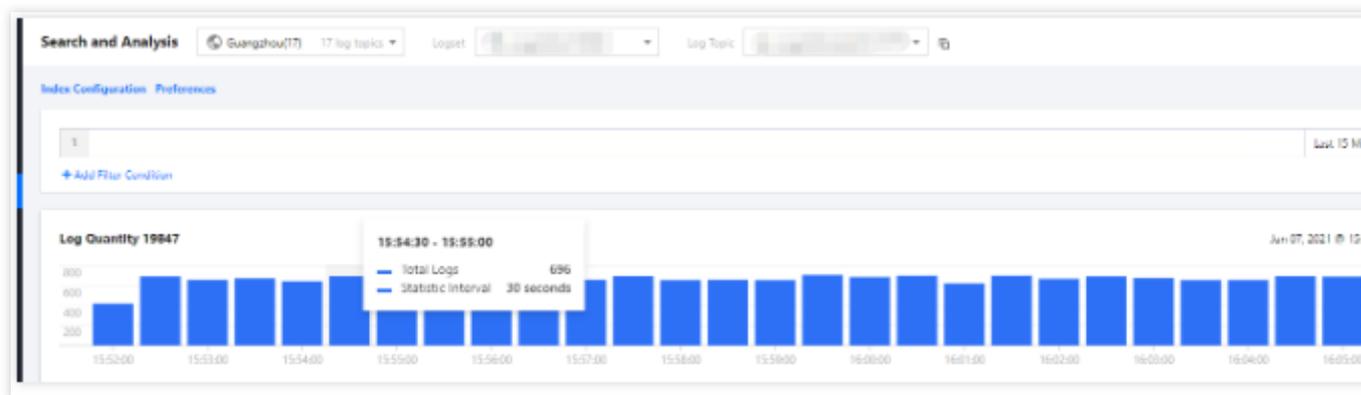
環境変数方式でTKE Serverlessログ収集を有効にし、ロールの権限承認を選択する場合は、**ロールの作成時にCVM キャリアを選択する必要があります**。TKEを選択すると権限承認を完了することができません。

問題がまだ解決されていない場合は、[チケットを提出](#)してご連絡ください。

ログルールを設定後は、どこでログを確認できますか。

1. CLSコンソールにログインし、左側の[検索分析](#)をクリックします。

2. 「検索分析」ページに進み、リージョン、確認したいログのログセットとログトピックを選択し、フルテキストインデックスを有効にすると、ログの検索と分析を行えるようになります。下の図をご覧ください。



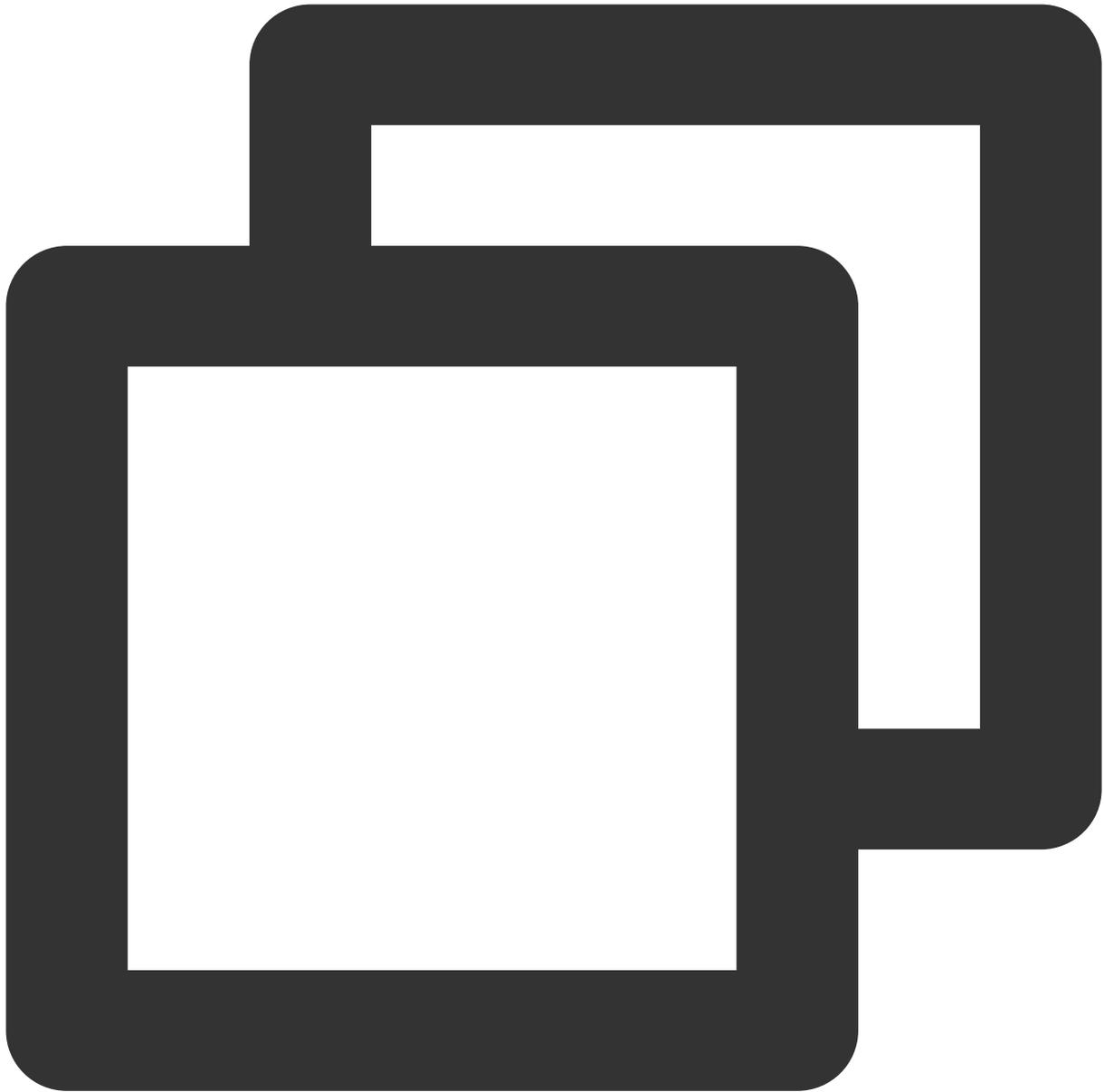
環境変数を使用してTKE Serverlessのログ収集を有効にする場合

Javaクラスアプリケーションで複数行のログをマージするにはどうすればよいですか。

クライアントプログラムのログデータが複数行にわたる場合は（Javaプログラムログなど）、改行コード `\n` をログの終了識別子として使用することはできません。ログシステムが各行のログを明確に区別できるようにするには、最初の行に正規表現を有する `configmap` を設定する必要があります。ある行のログがあらかじめ設定した正規表現と一致する場合、それをログの先頭とみなし、次の行の先頭が発生した時点でそれをこのログの終了識別子とします。詳細については [TKE Serverless ログ収集で複数行のログをマージする](#) をご参照ください。

ログごとの出力速度に対応するようにログ収集設定を調整するにはどうすればよいですか。

環境変数方式を使用してTKE Serverlessのログ収集を有効にする場合、TKE Serverlessはpod sandbox内で追加のログ収集コンポーネントを起動してログを収集し、レポートします。TKE Serverlessはログ収集コンポーネントのメモリ使用量を制限するため、ログの出力速度が速すぎる場合、ログ収集コンポーネントにOOMが発生することがあり、この場合は必要に応じてログ収集の設定を調整することができます。具体的な方法は、以下の `pod annotation` を手動で変更し、ログ収集コンポーネントが使用するメモリバッファを小さくしてメモリ占有率を下げるものです。



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

annotationの意味は次のとおりです。詳細については[Fluent Bit](#)をご参照ください。

パラメータ	意味	テ オ ト
Buffer_Chunk_Size	コンテナファイルログを収集するための初期バッファサイズを設定します。バッファサイズの拡大にも使用できます。	21

Buffer_Max_Size	各監視対象ファイルのバッファサイズの制限を設定します。バッファサイズを拡大したい場合（長いログなど）、この値によってバッファサイズの拡大量を制限することができます。読み取ったファイルがこの制限を超えた場合は、監視リストからこのファイルを削除します。	21
-----------------	--	----

Serverlessクラスターコンテナ内のアプリケーションのログ出力の基準は何ですか。

アプリケーションでログを出力する際、コンテナ内のアプリケーションは可能な限りstdoutに出力されることに注意する必要があります。プログラムログをコンテナのファイルに出力する場合、ログファイルのローテーションとクリーンアップを管理するための手段を備えるか、もしくはvolume永続ストレージをマウントする必要があり、そうしなければ20Gの一時領域が一杯になります。詳細については[Podの一時ストレージ](#)をご参照ください。ログファイルのローテーションとクリーンアップを管理する手段がない場合は、次の操作を推奨します。volume永続ストレージをマウントします。詳細については[ストレージ管理](#)をご参照ください。TKE Serverlessのログ収集を有効にします。詳細については[環境変数を使用したログ収集の設定](#)をご参照ください。

クラスター類

スケールアウトスケールイン関連

最終更新日：：2022-03-31 15:04:02

Cluster Autoscaler と監視指標に基づくノードのAuto Scalingはどのようなちがいがありますか。

Cluster Autoscaler はロードに関わらず、クラスター内のすべてのPodをスケジューリングできます。監視指標に基づくノードのAuto Scaling は、自動でスケールイン・アウトする際にPodを考慮しませんが、Podのないノードを追加したり、システムの中核Podがあるノードを削除したりする場合があります。Kubernetes は、このような自動スケールリングを推奨しません。Cluster Autoscaler と監視指標に基づくAuto Scalingのノードは競合するため、両方ともに有効にしないでください。

CA とスケールリンググループはどのような対応関係にありますか。

CAを有効にしたクラスターは、選択されたノード設定に従って、起動設定を作成し、その起動設定のスケールリンググループをバインディングします。バインディング後、そのスケールリンググループ内でスケールイン・アウトをして、スケールアウト後のCVMは自動的にクラスターに入られます。自動スケールイン・アウトのノードは従量制課金です。スケールリンググループに関しては [Auto Scaling 資料](#) をご参照ください。

Tencent Kubernetes Engine (TKE) コンソールで手動で追加したノードはCAスケールインすることはありませんか。

ありません。CAスケールインのノードはスケールリンググループ内のノードに限られます。 [TKE コンソール](#) で追加されたノードはスケールリンググループに追加されません。

Auto Scaling コンソールで、Cloud Virtual Machine (CVM) の追加や削除ができますか。

できません。 [Auto Scaling コンソール](#) にて、設定を変更する一切の操作は推奨されていません。

スケールイン・アウトは選択したノードのどのような設定を継承しますか。

スケールリンググループ作成時、 [起動設定](#) を作成するために、クラスター内のあるノードを参考として選択する必要があります。参考としたノードの設定は以下の通りです。

- vCPU
- メモリ
- システムディスクのサイズ
- データディスクのサイズ
- ディスクタイプ
- 帯域幅
- 帯域幅課金モデル
- パブリックネットワークIPの割当

- セキュリティグループ
- Virtual Private Cloud
- サブネット

複数のスケーリンググループを使用するには？

サービスの重要度やタイプといった特徴によって、複数のスケーリンググループを作成し、スケーリンググループに異なるラベルを設定することで、スケーリンググループからスケールアウトしたノードのラベルを指定することができます。これにより、サービスを分類します。

スケールイン・アウトの最大値はいくつに設定できますか。

現在、Tencent Cloudユーザーには各アベイラビリティゾーンにつき30の従量課金CVMクォーターが利用可能です。スケーリンググループで30台を超えた従量課金のCVMをご希望の場合、[チケットの提出](#)を通して申請してください。

詳細のクォーターについては現在のアベイラビリティゾーンのCVMの [インスタンス及びクォーター](#)をご参照ください。また、Auto Scalingにも最大値の制限があり、最大値は200となっています。それ以上の値を希望する場合、[チケットの提出](#)から申請してください。

クラスターに対してスケールインを有効にするのは安全ですか。

ノードをスケールインする際にPodをスケジューリングしなおすため、スケジューリングと一時中断後の再度スケールインを許容する必要があります。サービスに [PDB](#) を設定することを推奨します。PDBは、実行中のPodセットのコピーの最小数または最小パーセンテージを任意の時点で指定できます。PodDisruptionBudget を使用すると、アプリケーションデプロイヤーは同一時点で行ったPodを削除するクラスターオペレーションが過剰にPodを廃棄しないことを保証し、データ紛失やサービス中断、許容できないダウングレードを回避することができます。

ノード上にどんなPodがあるとスケールインされなくなりますか。

- 厳密に設定された PodDisruptionBudget の Pod が PDB を満たさない時、スケールインされません。
- Kube-system の Pod。
- ノード上に deployment、replica set、job、stateful set以外のコンソールで作成された Podが存在します。
- Podにローカルストレージがあります。
- Pod は他のノード上にスケジューリングできません。

ノードがスケールインの条件を満たしてからスケールインされるまでどのくらいの時間がかかりますか。

所要時間は10分です。

ノードがNot Ready 状態になってからスケールインされるまでどのくらいの時間がかかりますか。

所要時間は20分です。

どのくらいの間隔で、スケールイン・アウトが必要かどうかをスキャンしますか。

スキャン間隔は10秒です。

CVMにスケールアウトするまでどのくらいの時間がかかりますか。

基本的には10分以内ですが、Auto Scalingに関しては [Auto Scaling](#) をご参照ください。

Unschedulable の Pod があるのに、スケールアウトされないのはなぜですか。

以下の事項を確認してください：

- Podのリクエストリソースが大きすぎる。
- node selector を設定しているか。
- スケーリンググループは最大値に達しているか。
- アカウントの残高（残高不足の場合、Auto Scaling はスケールアウトできません）、または、クォーターなどが不足しているか。詳細については、 [Auto Scaling トラブルシューティング](#) をご参照ください。

Cluster Autoscaler が特定のノードをスケールインするのは、どのように防ぐことができますか。

```
# ノードのannotationsに次のメッセージを設定してください。
kubectl annotate node <nodename> cluster-autoscaler.kubernetes.io/scale-down-disabled=true
```

スケールイン・アウトのイベントは、どのようにフィードバックされますか。

Auto Scaling コンソールにて、スケーリンググループのスケーリング状況を調べたり、k8sイベントを閲覧できます。以下の3つのリソースは対応するイベントがあります：

- kube-system/cluster-autoscaler-status config map
 - **ScaledUpGroup** - CA がスケールアウト。
 - **ScaleDownEmpty** - CA が実行されていないPodのあるノードを1つ削除。
 - **ScaleDown** - CA がスケールイン。
- node
 - **ScaleDown** - CAがスケールイン。
 - **ScaleDownFailed** - CA がスケールインすることに失敗。
- pod
 - **TriggeredScaleUp** - CA がこのPodによってスケールアウト。
 - **NotTriggerScaleUp** - CA はスケールアウトできるスケーリンググループが見つからなかったため、このPodをスケジューリングできません。
 - **ScaleDown** - CA はこのPodを追い出すことで、ノードのスケールインを試みました。

クラスター関連

最終更新日：：2022-03-31 15:04:02

クラスター作成

クラスター作成時に、Cloud Virtual Machine (CVM) でパブリックネットワークIPを選択しなくてもよいですか。

CVMは必ずしもパブリックネットワークIPを選択する必要はありません。パブリックネットワークIPのないCVMは、イメージリポジトリ内のマイイメージのみ取得可能で、dockerhubやサードパーティのイメージを取得できません。

パブリックネットワークIPがないが、パブリックネットワーク帯域幅をもつCVMはElastic IP をバインディングすることでインターネットにアクセスすることが可能です。

クラスター作成時に、所属ネットワークを選択するのはなぜですか。

選択された所属ネットワーク及びサブネットは、クラスター内CVMが属するサブネットであり、それぞれのCVMを異なるアベイラビリティゾーンのサブネットに追加することによりアベイラビリティゾーンを跨った災害復帰を実現できます。

クラスター作成はどんなモデルをサポートしていますか。

従量課金のシステムディスクがクラウドディスクであるすべてのモデルをサポートしています。

現在、Tencent Kubernetes Engine(TKE) ホストはどんなOSをサポートしていますか。

現在、Ubuntu 16.04、CentOS 7.2 をサポートしています。他のOS対応を希望する場合、[チケットの提出](#) からご連絡ください。

CVM拡張

CVMを拡張する際、どのような制限がありますか。

現在のクラスターがあるリージョンのみ選択可能ですが、異なるアベイラビリティゾーンを選択し、アベイラビリティゾーンをまたいでデプロイすることが可能です。

CVM の数に制限はありますか。

あります。従量課金のすべてのCVMはユーザーのクォーターを超過してはいけません、詳細は [CVM 概要](#) をご参照ください。

CVM 廃棄

CVM 廃棄後、当該ホストにデプロイされたコンテナはどうなりますか。

CVM が廃棄されると、当該ホスト配下のコンテナ等のリソースも同時に廃棄されます。サービスのコンテナ数が実行したいコンテナ数より少ない場合、実行したいコンテナ数に達するまで、クラスターは他のホストでコンテナを起動します。

ContainerdとDockerの選択方法

最終更新日：2022-03-31 15:06:15

ランタイムコンポーネントの選択方法は？

コンテナランタイム（Container Runtime）はKubernetes（K8S）の最も重要なコンポーネントの1つであり、イメージとコンテナのライフサイクルを管理します。Kubeletは Container Runtime Interface（CRI）を通して、コンテナランタイムと通信し、イメージとコンテナを管理します。

TKEはcontainerdとdockerをランタイムコンポーネントとして選択することをサポートします：

- Containerd呼び出しチェーンはより短く、コンポーネントが少なく、安定性に優れ、占有するノードリソースがより少ないです。Containerdの選択を推奨します。
- 次の場合、ランタイムコンポーネントにdockerを選択してください：
 - docker in dockerを使用する場合。
 - TKEノードにおいてdocker build/push/save/load等のコマンドを使用する場合。
 - docker APIを呼び出す場合。
 - docker compose または docker swarmが必要な場合。

ContainerdとDockerコンポーネントにおいてよく使われるコマンドはなんですか。

Containerdはdocker APIとdocker CLIをサポートしていませんが、cri-toolコマンドを通してそれに当たる機能を実現できます。

イメージに関する機能	Docker	Containerd
ローカルイメージリストを表示	docker images	crictl images
イメージをダウンロード	docker pull	crictl pull
イメージをアップロード	docker push	None
ローカルイメージを削除	docker rmi	crictl rmi
イメージ詳細を閲覧	docker inspect IMAGE-ID	crictl inspecti IMAGE-ID

コンテナに関する機能	Docker	Containerd
コンテナリストを表示	docker ps	crictl ps
コンテナを作成	docker create	crictl create
コンテナを起動	docker start	crictl start

コンテナに関する機能	Docker	Containerd
コンテナを停止	docker stop	crictl stop
コンテナを削除	docker rm	crictl rm
コンテナ詳細を閲覧	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に関する機能	Docker	Containerd
PODリストを表示	None	crictl pods
PODの詳細を閲覧	None	crictl inspectp
PODを実行	None	crictl runp
PODを停止	None	crictl stopp

呼び出しチェーンにはどんな違いがありますか。

- DockerがK8S コンテナとして動作する際、呼び出し関係は以下のようになります。

```
kubelet --> docker shim (kubeletプロセスに存在) --> dockerd --> containerd
```

- ContainerdがK8S コンテナとして動作する際、呼び出し関係は以下のようになります。

```
kubelet --> cri plugin (containerdプロセスに存在) --> containerd
```

そのうち dockerd は、swarm cluster、docker build、docker API といった機能を追加したものの、バグを作り込みました。なお、containerd と比べて、呼び出しが1つ多くなります。

Streamサービス

Kubectl exec/logs といったコマンドは、apiserver とコンテナランタイムの間にストリーム転送チャンネルを構築する必要があります。

どのようにContainerd でStreamサービスを設定または使用しますか？

Docker API自体はStreamサービスを提供し、kubelet 内の docker-shim は docker API を通してストリームを転送します。

Containerd のStreamサービスに対して、個別に設定する必要があります。

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

K8S 1.11より前のバージョンとそれ以降のバージョンにおける設定の違いは？

Containerd のストリームサービスは K8S のバージョンによって、実行時の設定が異なります。

- K8S 1.11より前のバージョン：

Kubelet は stream proxy をせず、リダイレクトのみをします。すなわち、Kubelet は containerd が公開するストリームサーバーのアドレスを apiserver に送信し、apiserver が containerd のストリームサービスに直接アクセスします。このとき、セキュリティのため、Streamサービスのフォワーダーを認証する必要があります。
- K8S 1.11以降：

K8S1.11 は kubelet stream proxy を導入したため、containerd stream サービスはローカルアドレスの監視のみを行えばよいです。

その他のちがひ

コンテナログと関連パラメータ

比較項目	Docker	Containerd
保存パス	Docker を K8S コンテナとして実行する際、コンテナログの書き込みは docker によって行われ、 <code>/var/lib/docker/containers/\$CONTAINERID</code> のようなディレクトリに保存されます。Kubelet は <code>/var/log/pods</code> と <code>/var/log/containers</code> の下に <code>/var/lib/docker/containers/\$CONTAINERID</code> ディレクトリ下にあるコンテナログファイルを指すソフトリンクを作成します。	Containerd を K8S コンテナとして実行する際、コンテナログの書き込みは 1 によって行われ、 <code>/var/log/pods/\$CONTAINER_</code> ディレクトリに保存されると同時に <code>/var/log/containers</code> ディレクトリにログファイルを指すソフトリンクを作成します。
設定パラメータ	Docker の設定ファイルで次を指定します。 <pre>"log-driver": "json-file", "log-opts": {"max-size": "100m", "max-file": "5"}</pre>	<ul style="list-style-type: none"> • 方法 1：Kubelet パラメータに指定します。 <pre>--container-log-max-file-size="100Mi"</pre>

		<ul style="list-style-type: none"> 方法2: KubeletConfiguration で定めます。 <pre> "containerLogMaxSize": "100Mi", "containerLogMaxFiles": </pre>
<p>コンテナログをデータディスクに保存</p>	<p>データディスクを「data-root」（デフォルトは /var/lib/docker ）へマウントしてください。</p>	<p>データディスクのマウントポイントのあるディレクトリを指すソフトリンク /var/log/pods を作成します。TKEで「コンテナとイメージをディスクに保存」を選択すると、ソフトリンク /var/log/pods されます。</p>

CNIネットワーク

比較項目	Docker	Containerd
CNIの呼出元	Kubelet 内の docker-shim	Containerd 内の cri-plugin (containerd 1.1 以降)
CNI の設定方法	Kubelet パラメータ <code>--cni-bin-dir</code> と <code>--cni-conf-dir</code>	Containerd 設定ファイル (toml) : <pre> [plugins.cri.cni] bin_dir = "/opt/cni/bin" conf_dir = "/etc/cni/net.d" </pre>

サービス類

最終更新日：2022-03-31 15:04:02

サービス作成

同じサービス名を使用できないのはなぜですか。

サービス名はクラスター内でサービスを識別するための唯一のものです。サービス名+アクセスポートによって、サービス同士が互いアクセスすることができます。

サービスの作成はTencent Cloud以外の、dockerhubイメージのようなサードパーティイメージを使用可能ですか。

ホストでdocker loginコマンドを実行し、サードパーティのイメージリポジトリにログインして、イメージを取得できます。

パブリックネットワークの使用のための前提条件はなんですか。

クラスター内のCloud Virtual Machine (CVM) がパブリックネットワーク帯域幅をもつことを確認してください。パブリックネットワーク帯域幅が無ければ、パブリックネットワークサービスの作成に失敗します。

メモリ制限、CPU制限をどのように記入しますか。

詳細については、[Tencent Kubernetes Engine \(TKE\) リソース制限に関する説明](#)をご参照ください。

サービス作成時の特権とはなんですか。

特権を有効化すると、コンテナ内のアプリケーションが本当のroot権限を持つようになります。nfsサーバーの構築など、コンテナ内のアプリケーションでより高度なシステムオペレーションを行う際に有効化を推奨します。

Cloud Load Balancer (CLB) 作成時にセキュリティグループを指定できますか。

現在、以下の2つの方法によって指定できます：

- 既存CLBを使用する。CLBを作成してセキュリティグループを設定した後に、サービスにマウントします。詳細は、サービスが既存CLBを使用する方法をご参照ください。
- サービスの中で `TkeServiceConfig` によってセキュリティグループを設定します。CLB作成時に設定に従って対応するセキュリティグループを使用します。この機能を使用するには、[チケット申請](#)に従って申請してください。

CLBのIPを通して、クラスター内でサービスにアクセスすると、アクセスできない恐れがあります。

一般的に、4層CLBは複数のノードをバインディングして、real server (rs) として使用します。一定の確率

でメッセージループバックが失敗する恐れがあるため、使用時に client と rs は同一のCVM上に置いてはいけません。

PodがCLBにアクセスする際に、PodをソースIPとします。ソースIPがプライベートネットワークに転送された場合も、CLBはsnat処理によってソースIPをノードIPに変換しません。そうしないと、CLBがメッセージを受信する際、どのノードが転送したものか判断することが出来ないため、ループバック回避ポリシーが機能せず、すべてのrsが転送される可能性があります。Client が所在するノードに転送されると、CLBはリターンパケットを受信出来ないため、アクセスに失敗します。

コンテナ数の更新

コンテナ数を更新する際にどのような点に注意すべきですか。

コンテナ作成失敗を防ぐため、CPUとメモリリソースが十分確保されていることをご確認ください。

コンテナ数に0を指定できますか。

できます。コンテナ数に0に指定することで、サービス設定をセーブし、占有したリソースを解放できます。

サービス設定の更新

ローリングアップデートはサポートされていますか。

ローリングアップデート及びクイックアップデートの2つがサポートされています。

パブリックネットワークCLBはプライベートネットワークCLBに切り替え可能ですか。

可能です。現在、パブリックネットワークからVPCプライベートネットワークへの切り替え、VPCプライベートネットワークからパブリックネットワークへの切り替え、及びVPCの異なるサブネット間の切り替えが可能です。詳細はサービスライフサイクル管理をご参照ください。

- サービスに対してCLBリソースのライフサイクルを管理する場合、CLB及びそのパブリックネットワークIPは解放されます。
- パブリックネットワークからプライベートネットワークへの切り替えは即時に実行されるのではなく、パブリックネットワークCLBサービスがオフラインになってから、プライベートネットワークCLBがサービス提供を開始するまで、一定時間を要します。クラスター内でプライベートネットワークサービスリソースを予め設定し、テストを実施することを推奨します。トラフィックの切り替え完了後、パブリックネットワークのサービスリソースを削除してください。

サービス削除

作成されたCLBは、サービス削除後に自動的に廃棄されますか。

サービスが削除されると、サービス作成時に作られたCLBは削除されます。サービス作成時に選択した既存のCLBは影響を受けません。

サービス削除によってデータは影響を受けますか。

サービスを削除してもコンテナは削除されません。データは影響を受けないため、バックアップする必要はありません。

サービス運用

コンテナシステム時間の北京時間への設定方法は？

コンテナはデフォルトでUTC時間を採用しており、コンテナ使用時にコンテナシステム時間と北京時間に8時間の差が生じることがあります。dockerfile にタイムゾーンファイルを作成することで解決できます。詳細は [コンテナ内タイムゾーンの不一致について](#) をご参照ください。

ubuntu や php 、 busybox といったDockerhubの一部のイメージがコンテナサービス内で異常が発生したときの対処方法は？

起動コマンドを設定していない、またはデフォルトの起動コマンドが bash である場合に、コンテナがランチャープログラムを実行してから終了するシステム異常が発生します。コンテナが継続して動作するよう、コンテナ内のPIDが1であるプログラムは必ず常駐プロセスでなければいけません。これは、PIDが1であるプロセスが終了すると、コンテナも終了するためです。centos といった一部のイメージについては、/bin/bash を実行コマンドとして使用し、-c sleep 800000 を実行引数としてサービスを作成できます。コンソール上で実行引数を入力する際は、-c と sleep 800000 を2行にわけて入力する必要があります。

現在確認されているデフォルトパラメータでサービスを起動できないイメージには以下のものがあります：

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和python。

TKE Serverless クラスター

TKE Serverless クラスター 関連

最終更新日：：2023-04-26 19:23:11

ここではTKE Serverlessのクラスターに関するよくあるご質問をまとめ、クラスター関連のよくあるトラブルの原因とその解決方法をご紹介します。

Podの仕様が入力されたRequest/Limitと一致しないのはなぜですか。

Podのリソース量を割り当てる際、TKE Serverlessはワークロードによって設定されたRequestとLimit値に従ってリソース量を割り当てるのではなく、設定されたRequestとLimitについて計算を行い、Podの実行に必要なリソース量を自動的に判断します。Request、Limitの自動計算によってリソース仕様を指定する方法の詳細についてお知りになりたい場合は、[CPU Pod仕様の計算方法](#)および[GPU Pod仕様の計算方法](#)をご参照ください。

TKE Serverless クラスターのコンテナネットワークを追加または変更するにはどうすればよいですか。

クラスターを作成する際は、1つのVPCネットワークをクラスターネットワークとして選択し、同時に1つのサブネットをコンテナネットワークに指定します。詳細については[コンテナネットワークの説明](#)をご参照ください。ServerlessクラスターのPodはコンテナネットワークのサブネットIPを直接占有します。クラスターの使用中にコンテナネットワークを追加または変更したい場合は、スーパーノードの追加/除去操作によって実現できます。具体的な操作については次の手順をご参照ください。

ステップ1：スーパーノードを新規作成してコンテナネットワークを追加する

1. TKEコンソールにログインし、左側ナビゲーションバーの[クラスター](#)を選択します。
2. コンテナネットワークを変更したいクラスターIDをクリックし、クラスター基本情報ページに進みます。
3. 左側のスーパーノードを選択し、スーパーノードページで[新規作成](#)をクリックします。
4. スーパーノードの[新規作成](#)ページで、十分なIPを持つコンテナネットワークを選択し、**OK**をクリックすると作成が完了します。

Basic information

Region
Cluster ID
Kubernetes version 1.
Cluster network [7](#)

Super node

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

Container network

<input type="checkbox"/>	Subnet ID	Subnet name	Availability zo...	Remaining IPs
<input type="checkbox"/>	<input type="text" value="..."/>	<input type="text" value="..._subnet103..."/>	<input type="text" value="Guangz..."/>	<input type="text" value="..."/>

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

[Add node](#)

ステップ2：スーパーノードを除去してコンテナネットワークを削除する

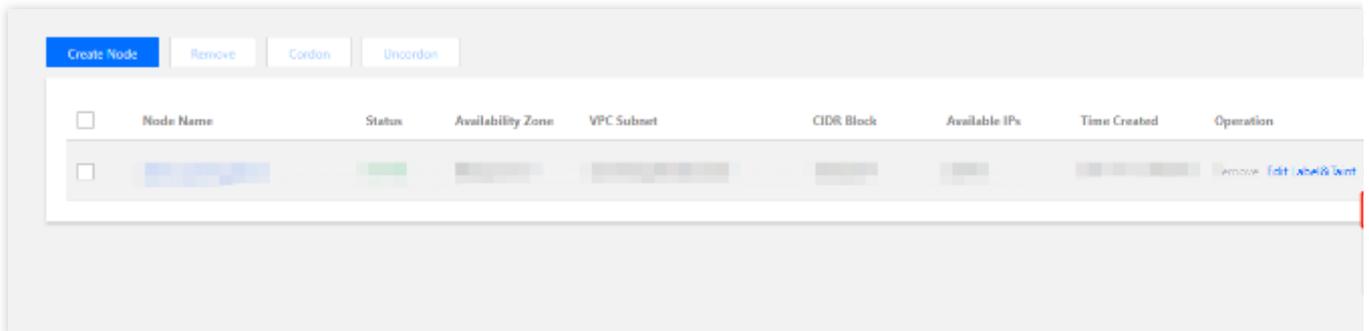
注意

スーパーノードを除去した後に、Serverlessクラスターに少なくとも1つのスーパーノードが存在する必要があります。つまり、この時点でスーパーノードが1つしかない場合は、除去操作は実行できません。

スーパーノードを除去する前に、このスーパーノード上のPodをすべて他のスーパーノード上（DaemonSetの管理するPodを除く）にドレインする必要があります。ドレインが完了してからでなければ除去操作は実行できず、これに従わない場合はノードの除去に失敗します。具体的な操作については次の手順をご参照ください。

1. TKEコンソールにログインし、左側ナビゲーションバーの**クラスター**を選択します。
2. コンテナネットワークを変更したいクラスターIDをクリックし、クラスター基本情報ページに進みます。

3. 左側のスーパーノードを選択し、スーパーノードページでノード名の右側の**その他** > **ドレイン**を選択します。下の図をご覧ください。



4. ノードのドレインページでノード情報を確認し、**OK**をクリックします。ドレイン後、このスーパーノードのステータスは「ロック済み」に変更され、このノード上へはPodをスケジューリングできなくなります。

注意

ドレインを行うとPodが再構築されますので、操作は慎重に行ってください。

5. スーパーノードページでノード名の右側の**除去**を選択します。

6. ノードの削除ページで**OK**をクリックすると、ノードの削除操作が完了します。

サブネットIPの枯渇によってPodのスケジューリングに失敗した場合はどうすればよいですか。

コンテナネットワークのサブネットIPの枯渇によってPodのスケジューリング失敗が起こった場合、ノードのログで下図のような2つのイベントが観察されます。

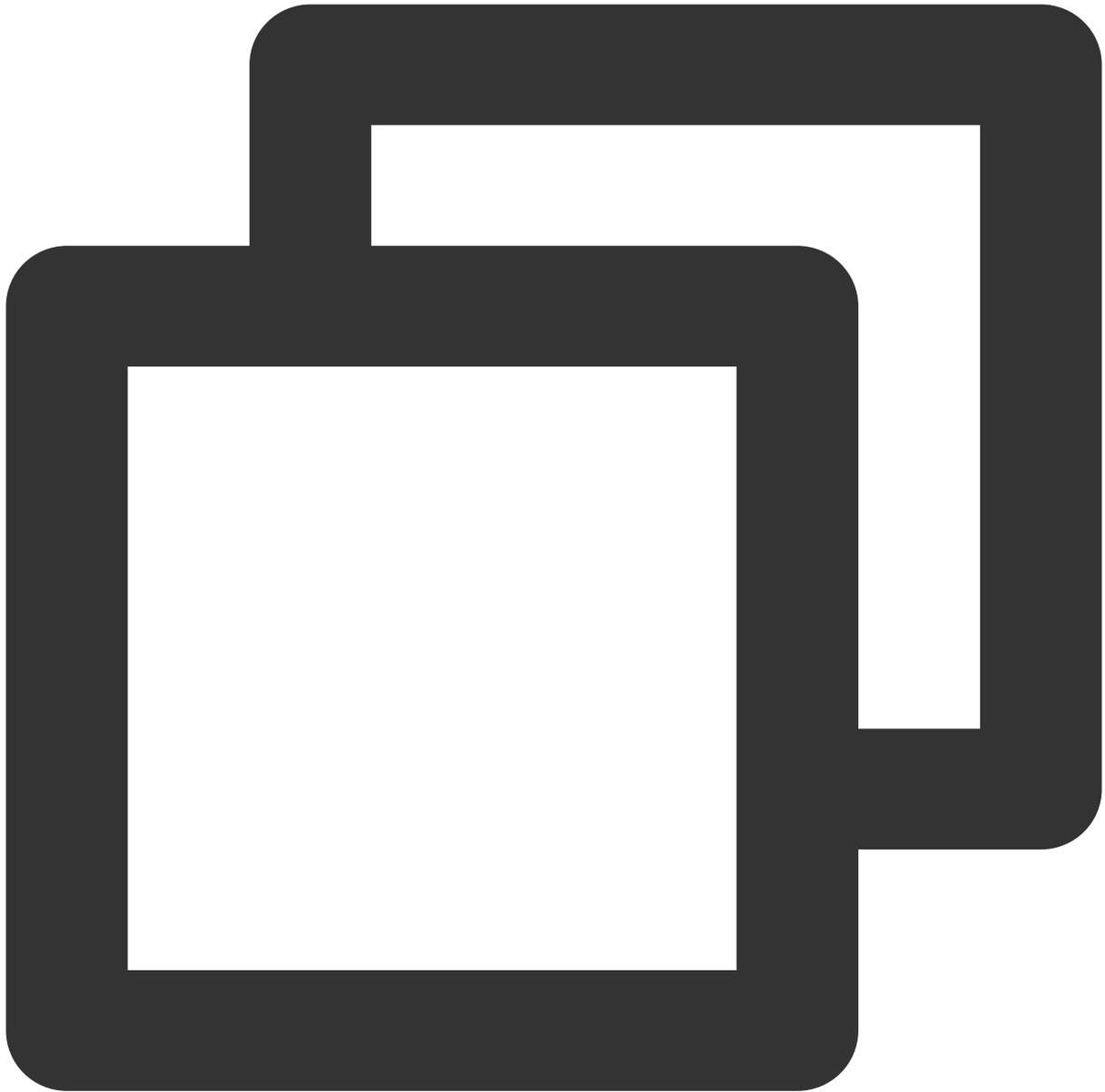
イベント1：

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

イベント2：

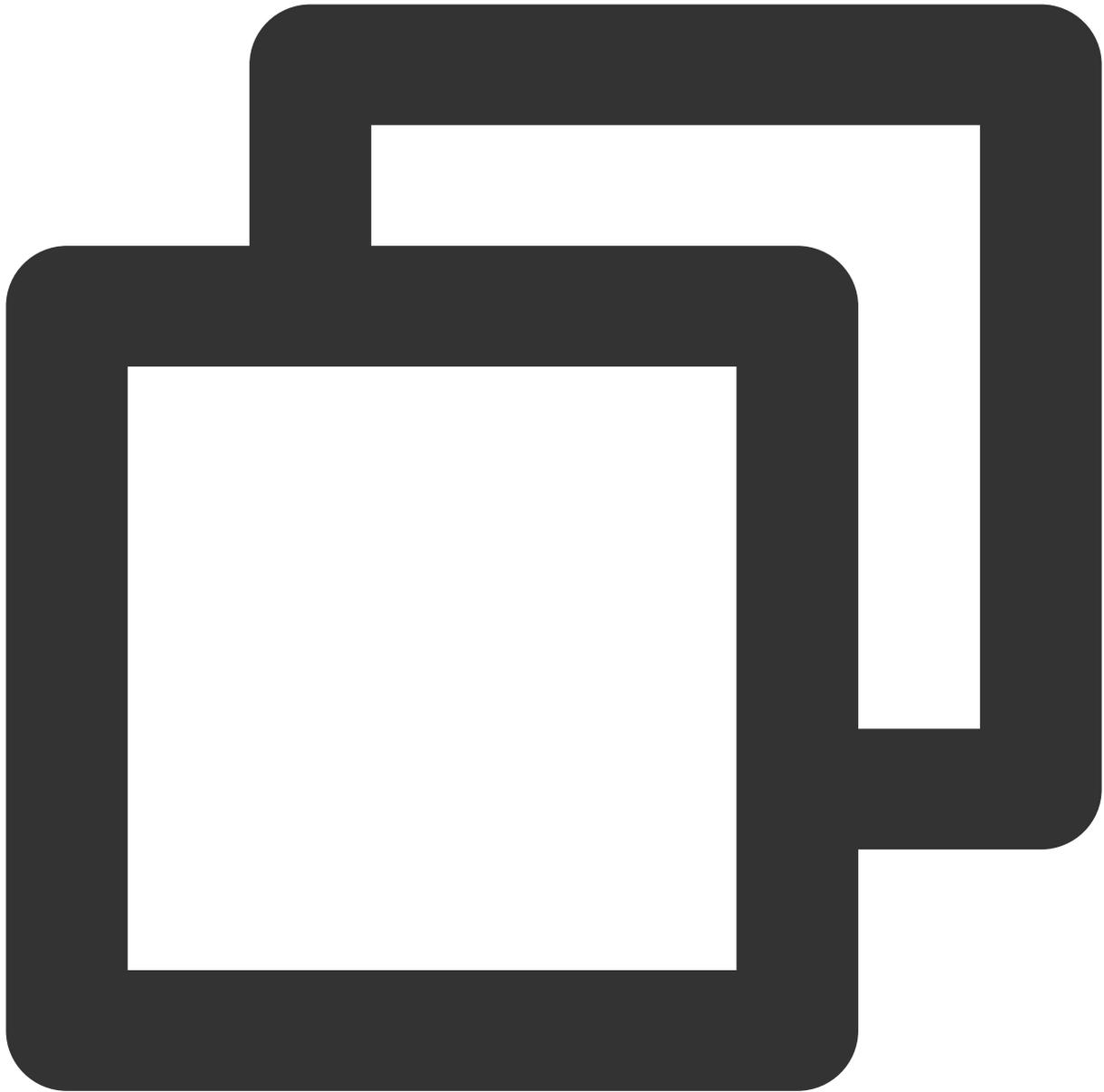
2021-04-20 15:00:10	2021-04-20 15:00:10	Warning	Pod	[Redacted]	FailedCreatePodSandB
2021-04-20 15:00:07	2021-04-20 15:00:07	Warning	Pod	[Redacted]	FailedCreatePodSandC

[TKEコンソール](#)で、またはコマンドラインで次のコマンドを実行することで、スーパーノードのYAMLを照会できます。

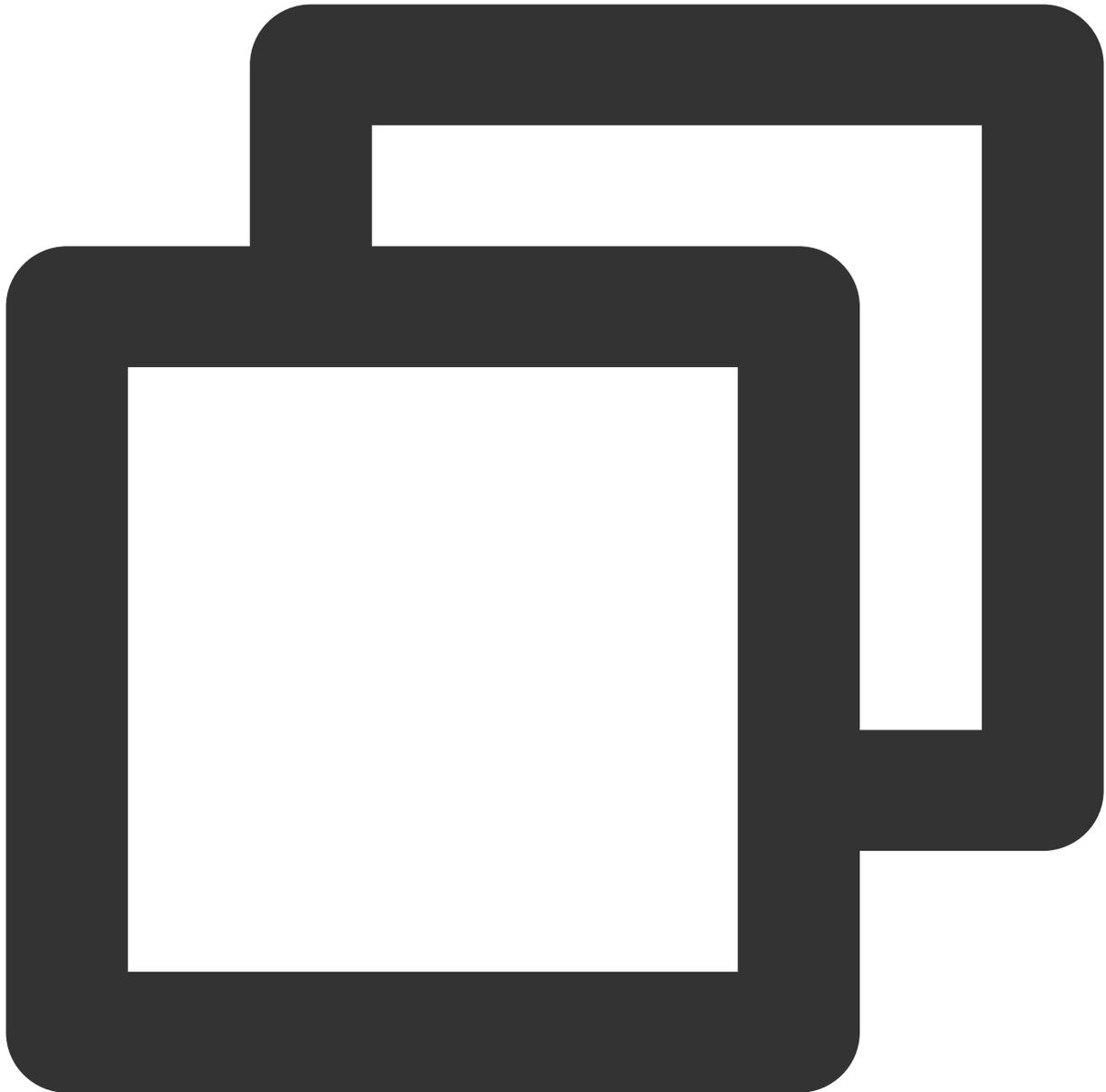


```
kubectl get nodes -oyaml
```

返された結果は次の通りです：



```
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
```

この場合、PodはコンテナネットワークのサブネットIPが枯渇したことによってスケジューリングに失敗したことがわかります。上記の状況が発生した場合は、新しいスーパーノードを作成してサブネットを追加し、それによ

てクラスターPodの利用可能なネットワークセグメントを拡張する必要があります。スーパーノードの新規作成操作については、[スーパーノードの新規作成](#)をご参照ください。

TKE Serverlessのセキュリティグループの使用ガイドおよび説明にはどのようなものがありますか。

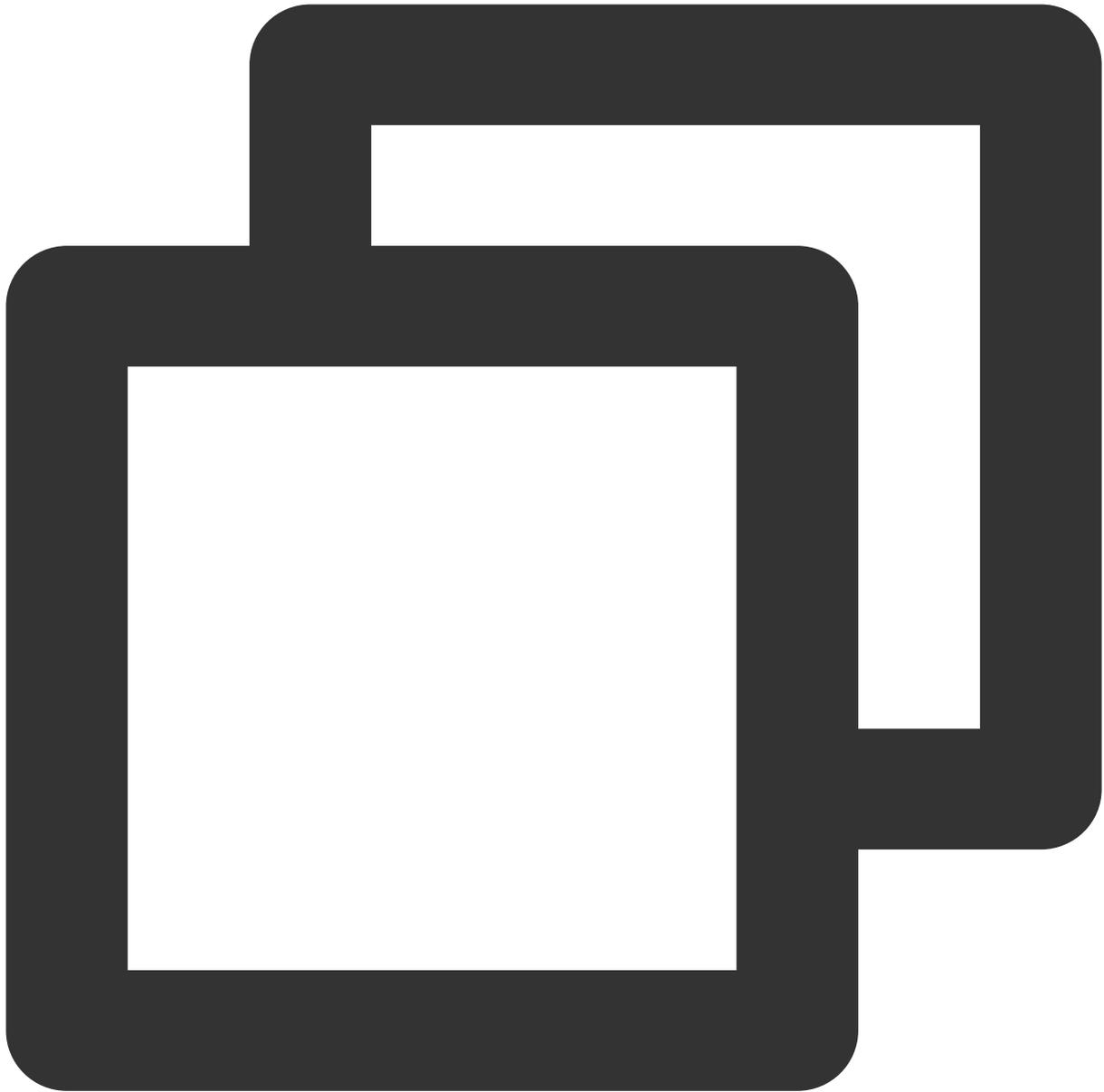
ServerlessクラスターPodを作成する際にセキュリティグループを指定しない場合は、defaultセキュリティグループを使用します。もしくは `Annotation eks.tke.cloud.tencent.com/security-group-id` : セキュリティグループID によってPodにセキュリティグループを指定することもできます。同一のリージョンにこのセキュリティグループIDがすでに存在することを確認してください。このAnnotationの詳細な説明については、[Annotationの説明](#)をご参照ください。

コンテナ終了のメッセージを設定するにはどうすればよいですか。

Kubernetesは`terminationMessagePath`によってコンテナ終了のメッセージソースを設定することができます。コンテナ終了の際、Kubernetesはコンテナの`terminationMessagePath`フィールド内で指定された終了メッセージファイルから終了メッセージを検索し、この内容を使用してコンテナの終了メッセージを入力します。メッセージのデフォルト値は `/dev/termination-log` です。

また、コンテナの`terminationMessagePolicy`フィールドを設定することで、コンテナ終了メッセージをさらにカスタマイズすることもできます。このフィールドのデフォルト値は `File`、すなわち終了メッセージファイルからのみ終了メッセージを検索します。ニーズに応じて`FallbackToLogsOnError`に設定することができ、これはコンテナがエラーによって終了した場合、終了メッセージファイルが空であれば、コンテナログを使用して最後の一部分の内容を終了メッセージとして出力できるものです。

サンプルコードは次のとおりです：



```
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
```

上記の設定により、コンテナがエラーによって終了し、かつメッセージファイルが空の場合、Get Podがstderrの出力を発見するとcontainerStatusesに表示します。

Hostパラメータを使用するにはどうすればよいですか。

Serverlessクラスターを使用する際は次の事項に注意する必要があります。

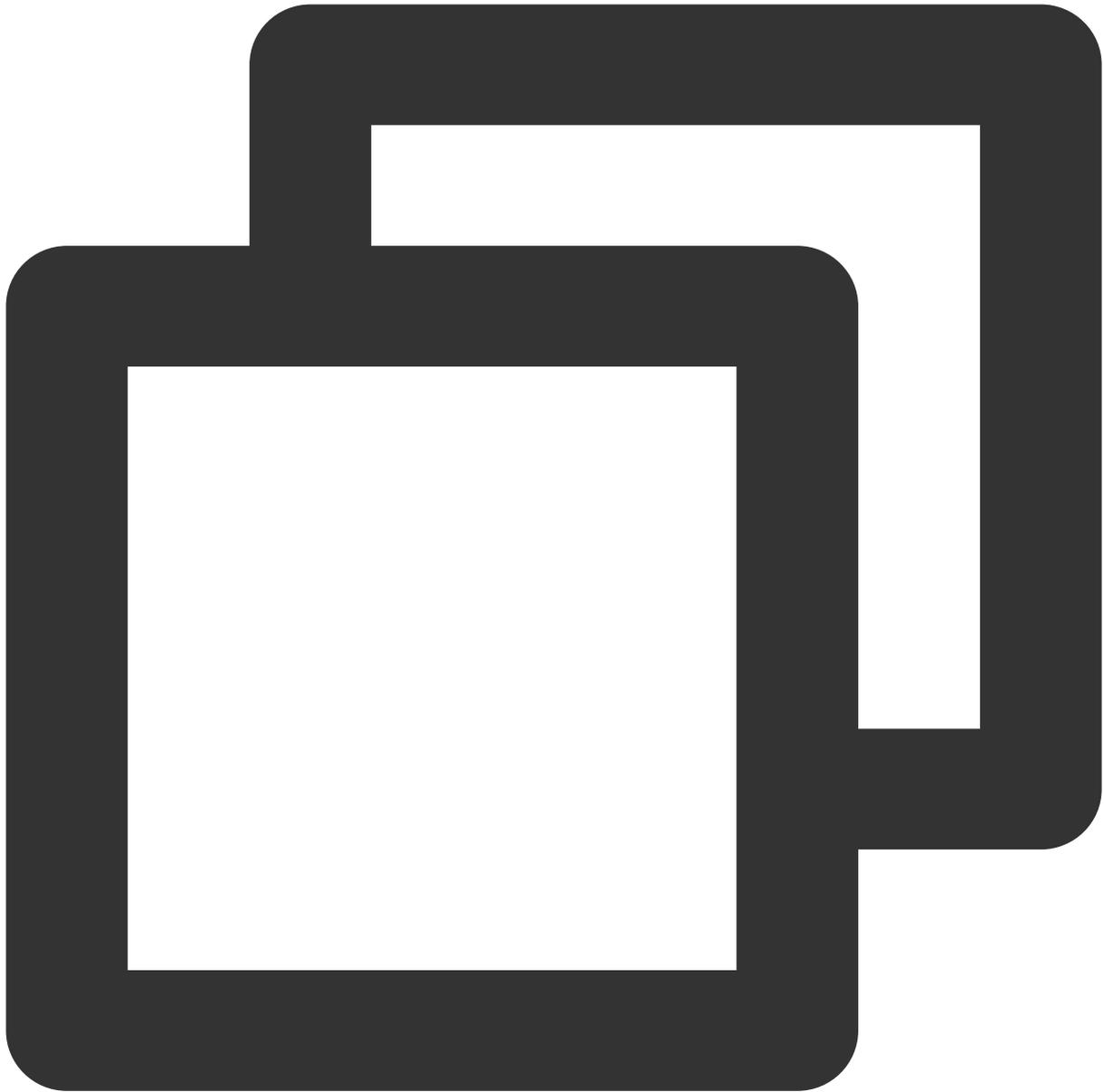
Serverlessクラスターにはノードはありませんが、Hostpath、Hostnetwork: true、DnsPolicy:

ClusterFirstWithHostNetなどのHost関連パラメータとの間に互換性があります。ノードがないため、ご使用の際は、これらのパラメータが提供する機能が標準のk8sと完全には一致しない点にご注意ください。

例えば、Hostpathを使用してデータを共有したい場合でも、同一のスーパーノード上にスケジューリングされた2つのPodには異なるサブマシンのHostpathが表示されます。また、Podの再構築後はHostpathのファイルも同時に削除されます。

CFS/NFSをマウントするにはどうすればよいですか。

Serverlessクラスターでは、Tencent Cloudの[Cloud File Storage \(CFS\)](#)を使用するほか、ご自身で構築したNFSファイルストレージを使用して、Volume形式でPod上にマウントすることで、永続的なデータストレージを実装することが可能です。PodにCFS/NFSをマウントするYAMLの例は次のとおりです。



```
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` : データボリューム名、タイプ、データボリュームのパラメータを設定します。

`spec.volumes.nfs` : NFS/CFSディスクを設定します。

`spec.containers.volumeMounts` : データボリュームのPod上のマウントポイントを設定します。

PodへのVolumeマウントに関する詳細な操作については、[ストレージボリュームの使用説明](#)をご参照ください。

イメージの再利用によってコンテナの起動を高速化するにはどうすればよいですか。

TKE Serverlessは、次回同じイメージを使用してコンテナを起動する際に起動を高速化できるよう、コンテナイメージのキャッシュをサポートしています。

再利用の条件：

1. 同一のワークロードのPodの場合、キャッシュ時間内に同一のアベイラビリティゾーン（Zone）でPodが作成され、かつ破棄された場合、新しく作成されるPodはデフォルトで同じイメージを再度プルしません。
2. 異なるワークロード（Deployment、Statefulset、Jobを含む）のPodでイメージを再利用したい場合は、次のannotationを使用できます。



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

同一ユーザーのアカウントで、同一のannotation valueを持つPodは、キャッシュ時間内において可能な限り起動イメージを再利用します。annotation valueにイメージ名 `eks.tke.cloud.tencent.com/cbs-reuse-key:` `"image-name"` を入力することをお勧めします。

キャッシュ時間：2時間。

イメージの再利用に異常が発生した場合はどうすればよいですか。

イメージ再利用機能を有効にしている場合、Pod作成の際に `$kubectl describe pod` で次のようなエラーが表示される場合があります。

```
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
```

復旧方法：

操作は何も行う必要はありません。数分待つとPodは自動的にrunningとなります。

原因：

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

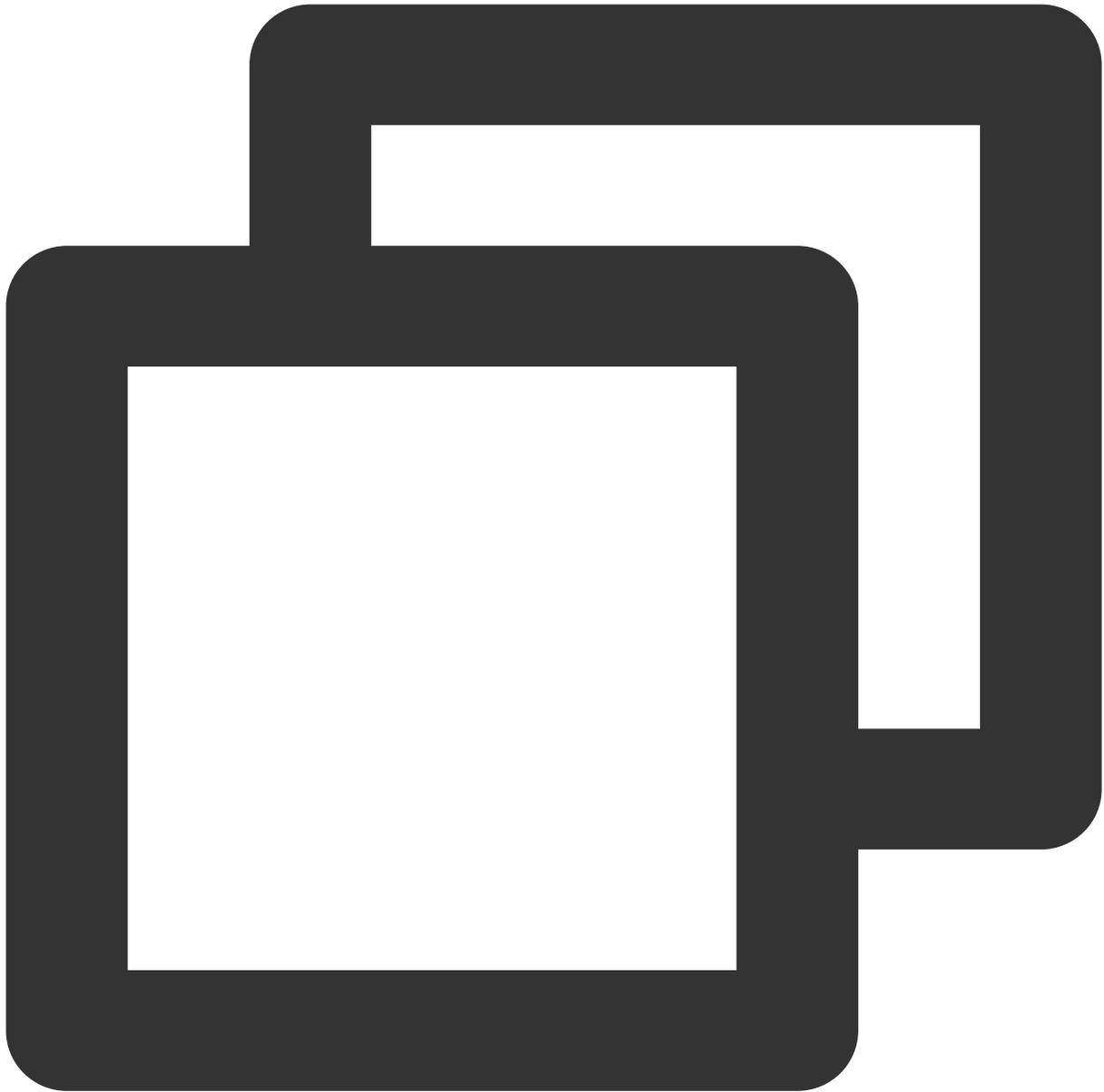
Podがデフォルトでシステムディスクを再利用する際、システムディスク内にもともとあったイメージでディスク容量が一杯の場合、ディスクに新しいイメージをダウンロードする容量がないため、「no space left on device: unknown」というエラーが表示されます。TKE Serverlessはイメージを定期的に回収するメカニズムをサポートしており、ディスク容量が一杯の場合、システムディスク内にある余分なイメージを自動的に削除し、現在のディスクに利用可能な容量を作ります。（数分かかります）

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

このログは、現在のPodにイメージをダウンロードするために4220828057の容量が必要であるのに、現時点で3889267064の容量のデータしかクリアできていないことを表しています。このeventが生成された理由は、ディスク上に複数あるイメージのうち、現時点でその一部しかクリアされていないためです。TKE Serverlessの時間指定イメージ回収メカニズムは、新しいイメージのプルに成功するまでイメージのクリーンアップを継続します。

自作したnfsをマウントした際、イベントOperation not permittedが表示されましたが、どうすればよいですか。

自作したnfsを使用して永続ストレージを実装する場合、接続の際にイベントOperation not permittedが表示されます。自作nfsの/etc/exportsファイルを変更し、/`<path><ip-range>(rw,insecure)`パラメータを追加する必要があります。次に例を示します。



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

Podのディスク容量が一杯になりました (ImageGCFailed) が、どうすればよいですか。

TKE ServerlessのPodは、デフォルトで20Gの利用可能なシステムディスク容量を無料で提供しています。システムディスク容量が一杯になった場合は次の方法で対処できます。

1. 使用していないコンテナイメージをクリーンアップする

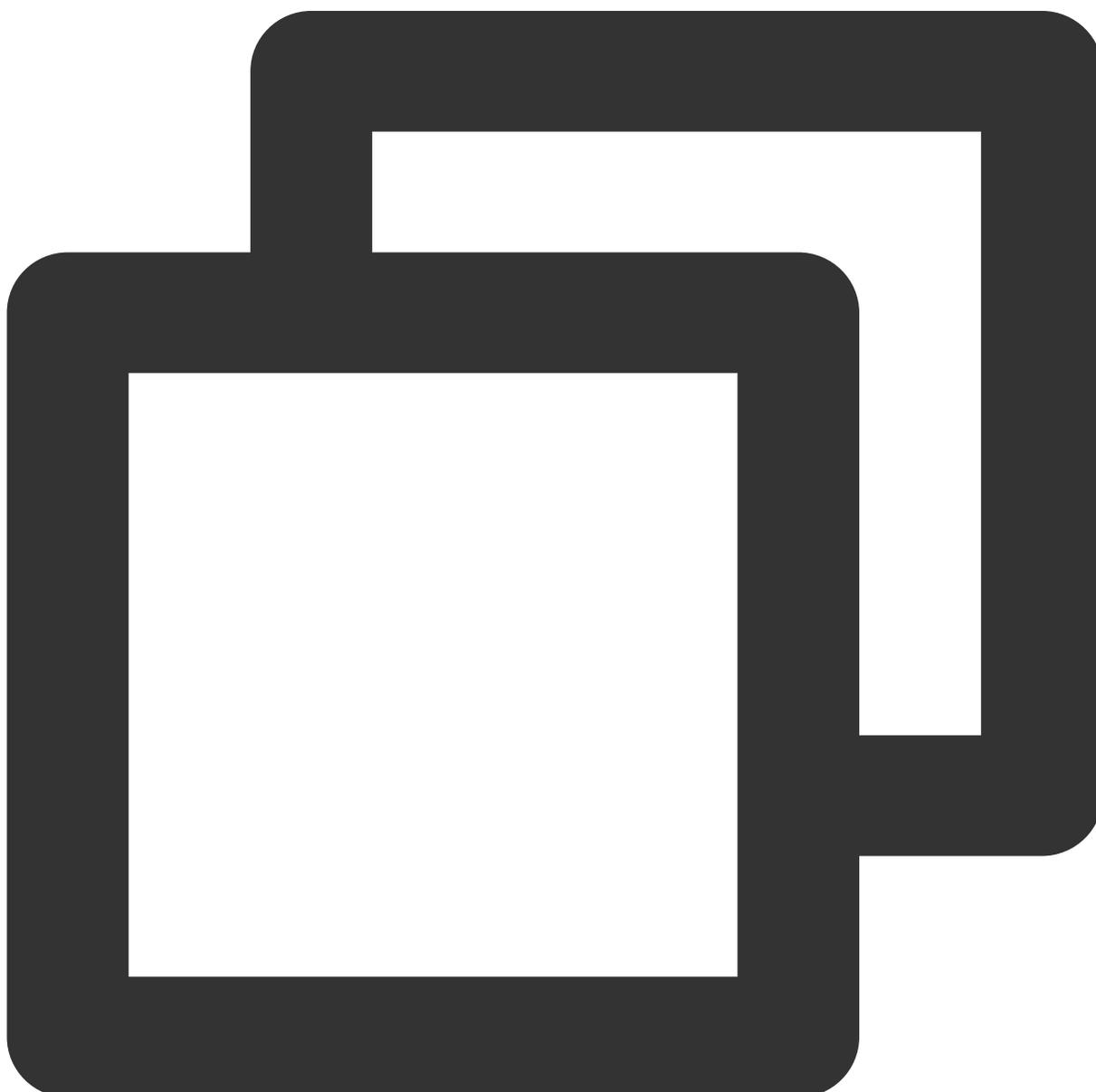
使用容量が80%に達すると、TKE Serverlessバックエンドはコンテナイメージの回収フローをトリガーし、使用していないコンテナイメージを回収してディスク容量を解放しようと試みます。容量をまったく解放できなかった場合は、ImageGCFailed: failed to garbage collect required amount of imagesというイベントが表示され、ディスク容量が不足していることをユーザーに通知します。

ディスク容量不足には一般的に次のような原因があります。

業務に大量の一時的な出力があった。duコマンドによって確認できます。

業務が削除済みのファイルディスクリプタを保有していて、ディスク容量が解放されない。lsfコマンドによって確認できます。

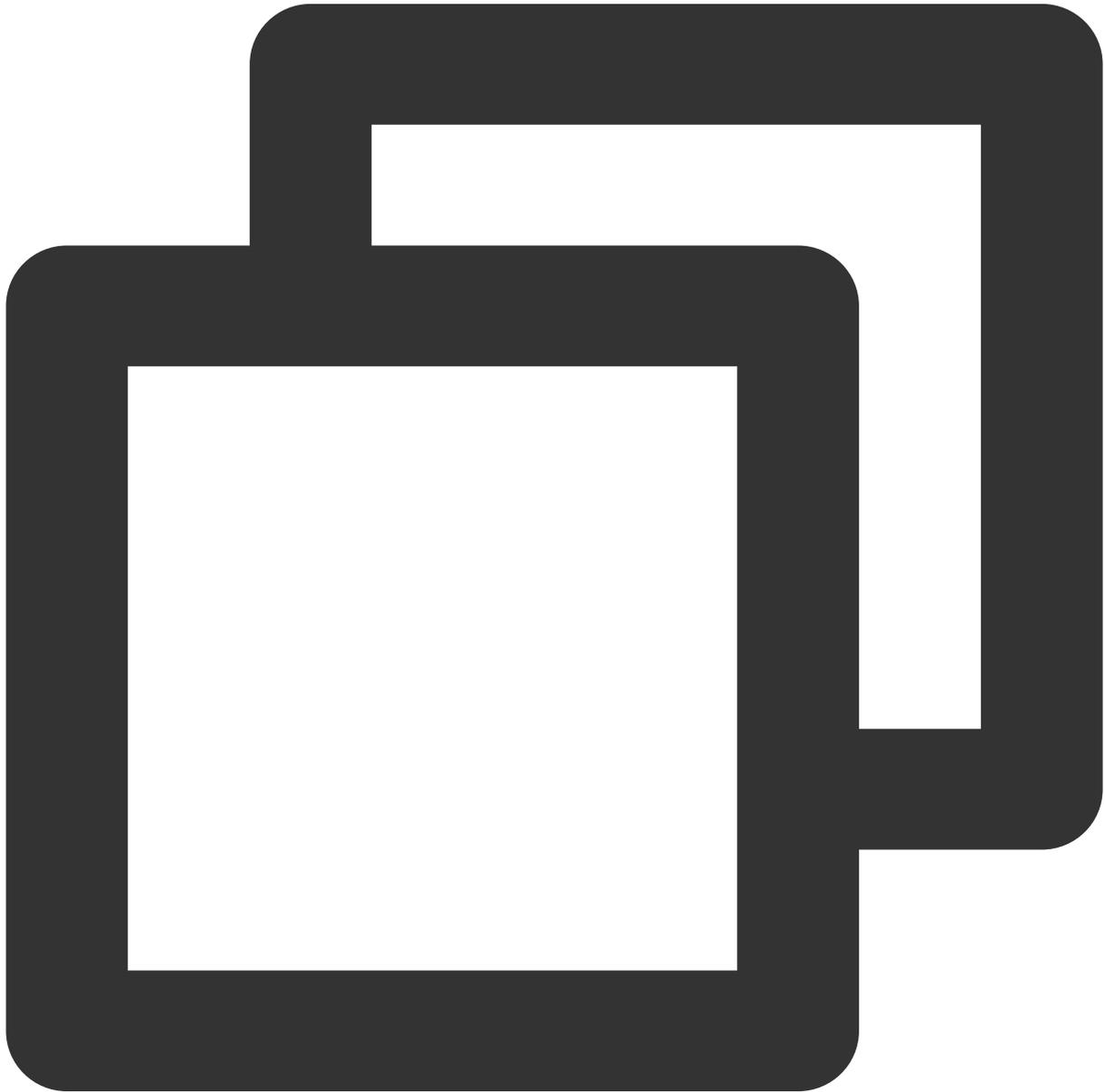
業務上、コンテナイメージ回収の閾値を調整したい場合は、次のannotationを使用できます。



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

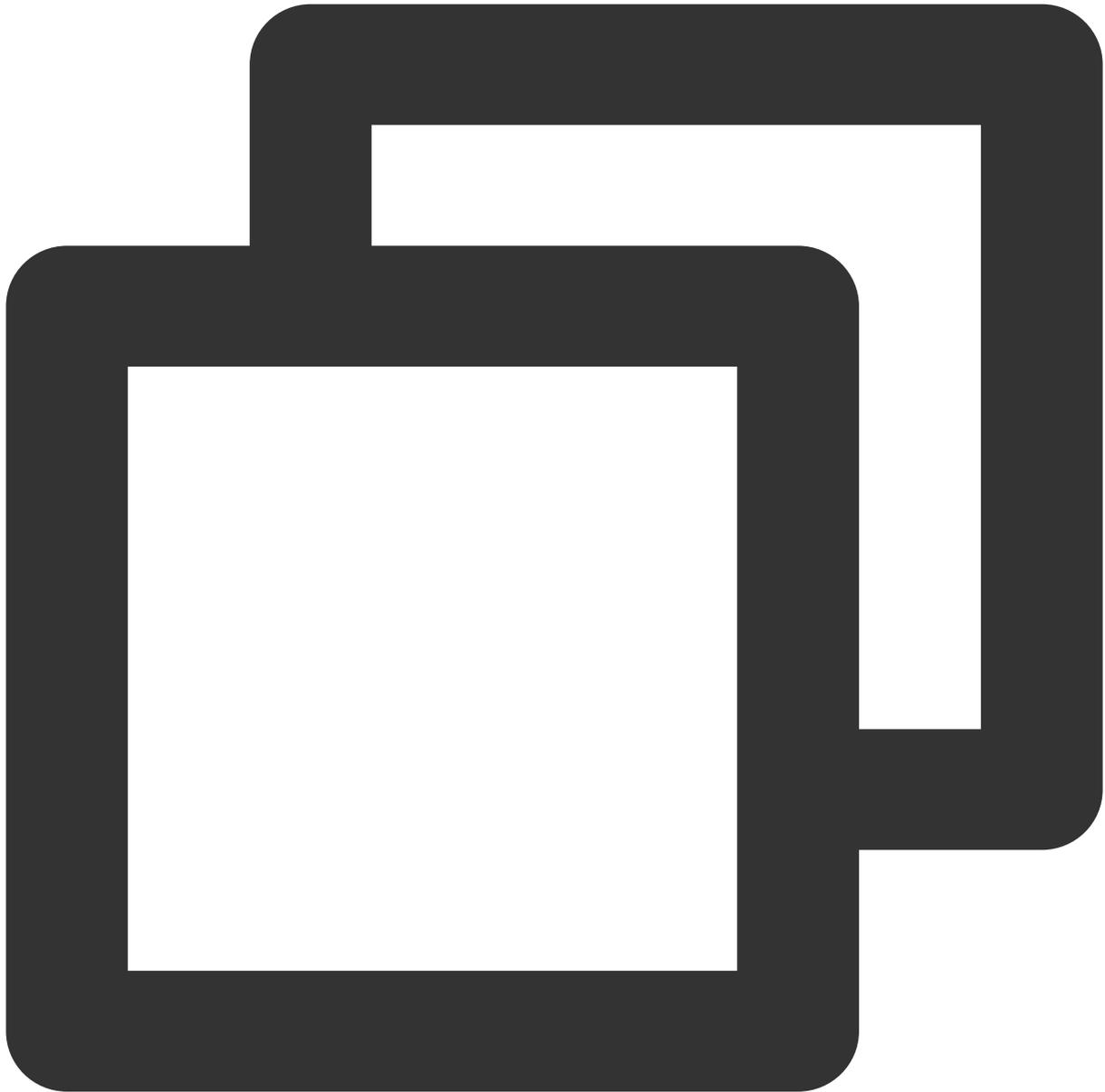
2. 終了したコンテナをクリーンアップする

業務でインプレースアップグレードを実施した場合、またはコンテナが異常終了したことがある場合、終了したコンテナはそのまま維持され、ディスク容量が85%に達しないとクリーンアップされません。クリーンアップの閾値は次のAnnotationを使用して調整できます。



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

終了したコンテナを自動的にクリーンアップしたくない場合（終了情報のさらなるトラブルシューティングが必要な場合など）は、次のAnnotationによってコンテナの自動クリーンアップを無効化することができます。ただし、ディスク容量が自動的に解放されないという副作用があります。



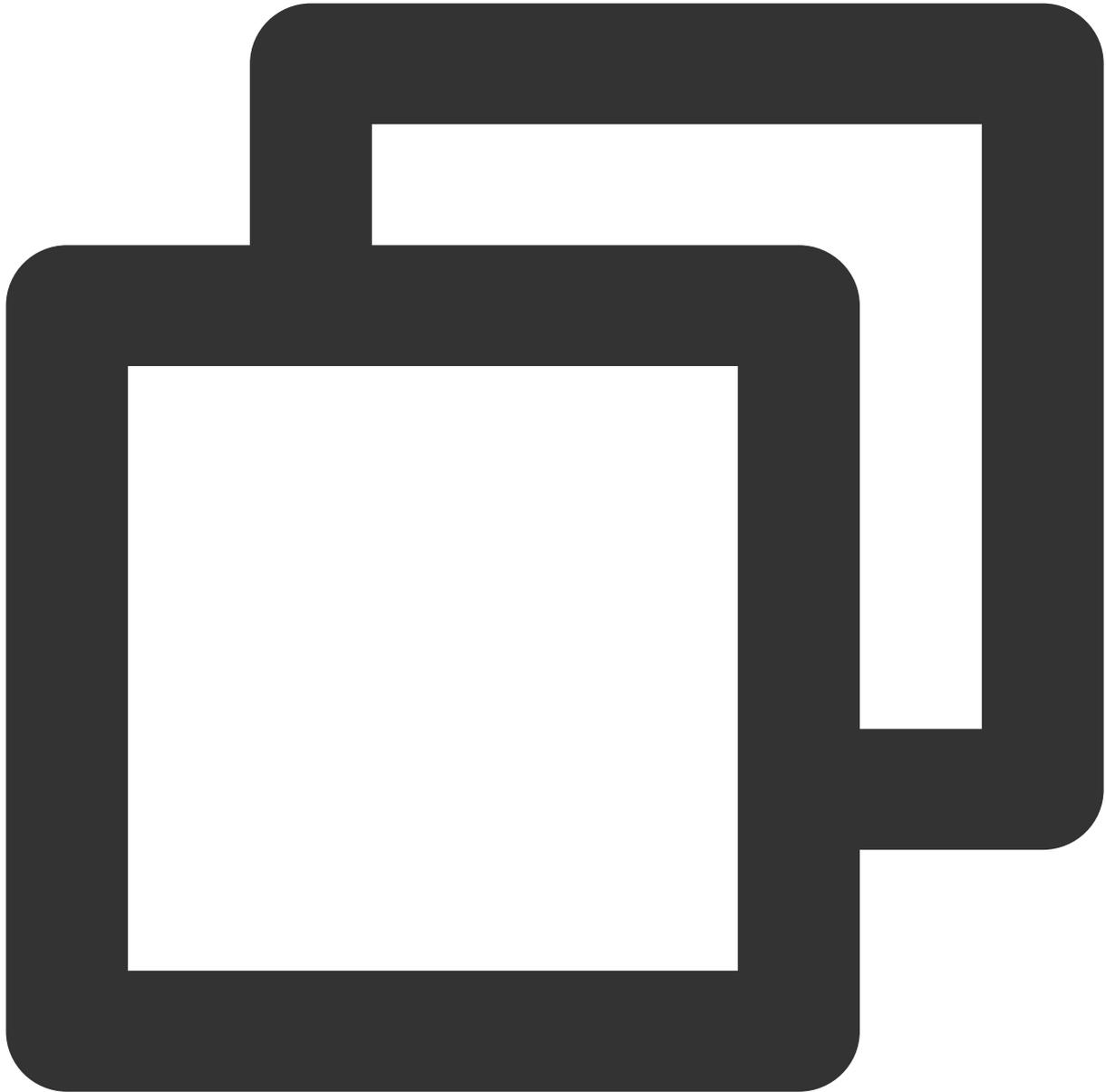
```
eks.tke.cloud.tencent.com/must-keep-last-container: "true"
```

説明

この機能のリリース日は2021-09-15のため、これ以前に作成されたPodにはこの機能はありません。

3. ディスク使用量の大きいPodを再起動する

業務上、コンテナ内のシステムディスク使用量があるパーセンテージを超えるとそのままPodを再起動させたい場合は、次のAnnotationによって設定できます。



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

Podの再起動のみを行い、サブマシンの再構築は行いません。終了と起動の際はどちらも正常なgracestop、prestop、ヘルスチェックを行います。

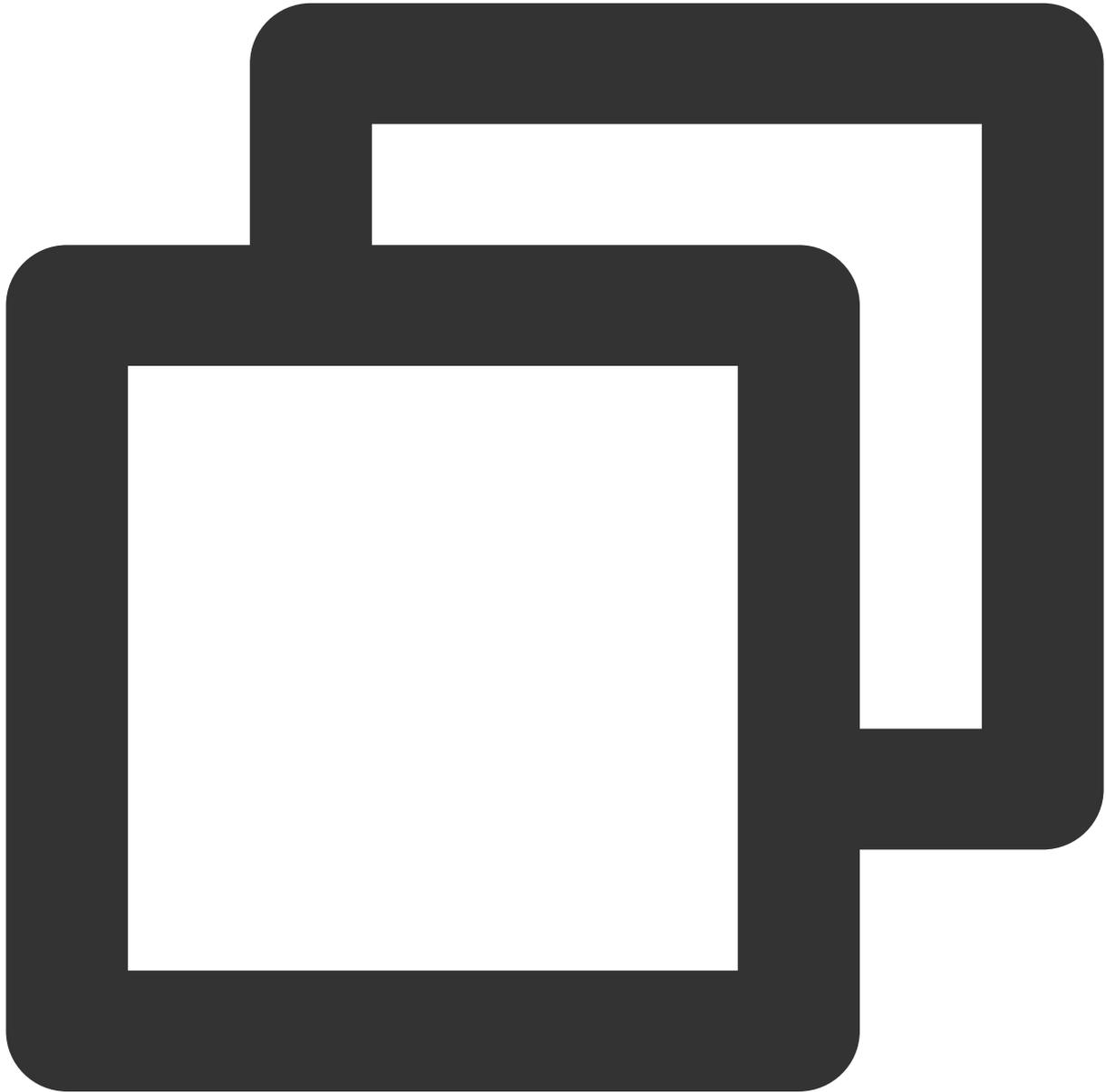
説明

この機能のリリース日は2022-04-27のため、これ以前に作成されたPodでこの機能を有効化する場合は、Podの再構築が必要です。

ポート9100の問題

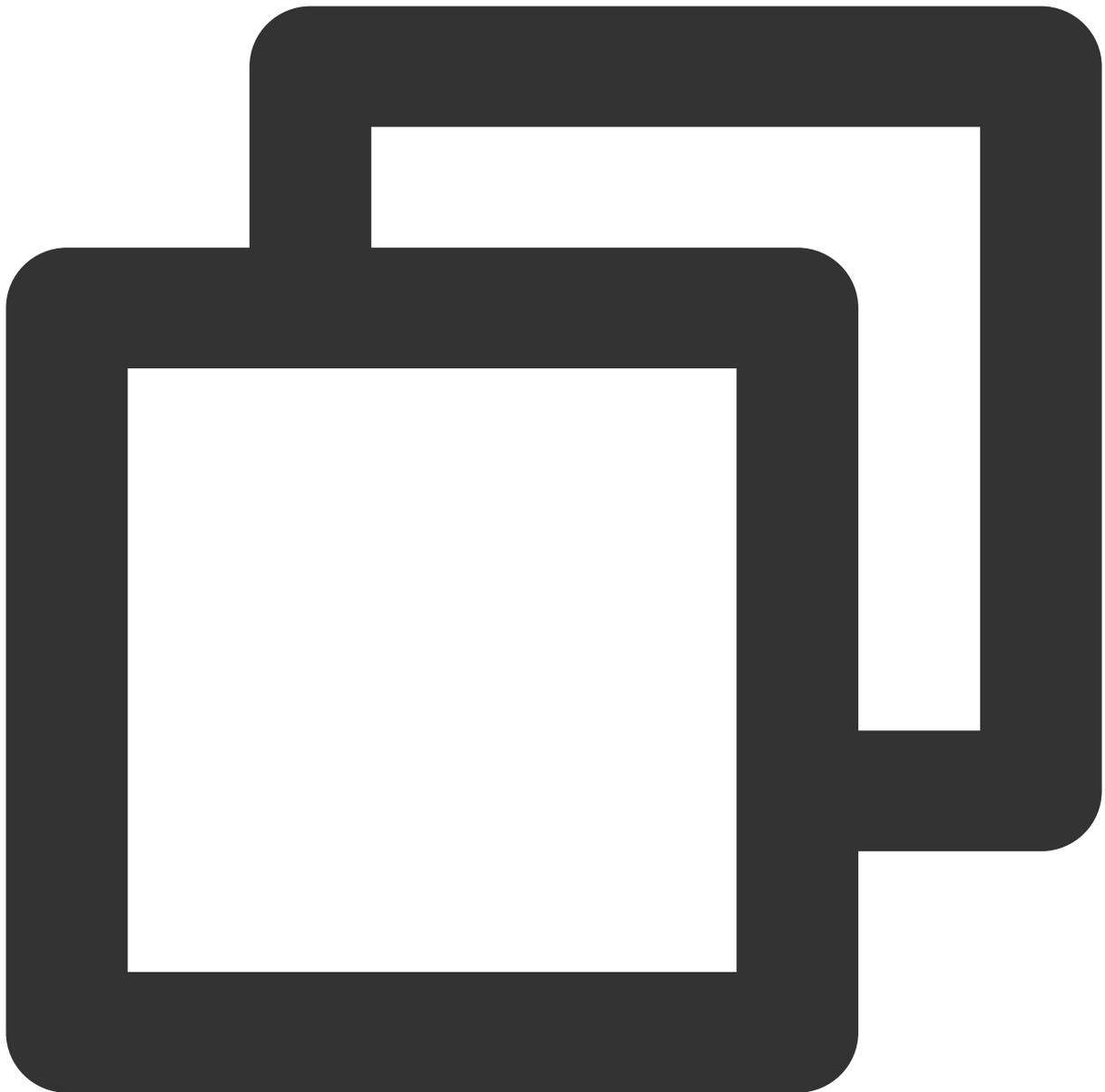
TKE ServerlessのPodはデフォルトでポート9100によってモニタリングデータを外部に公開します。ユーザーは次のコマンドを実行し、9100/metricsにアクセスしてデータを取得できます。

すべてのメトリクスを取得する場合：



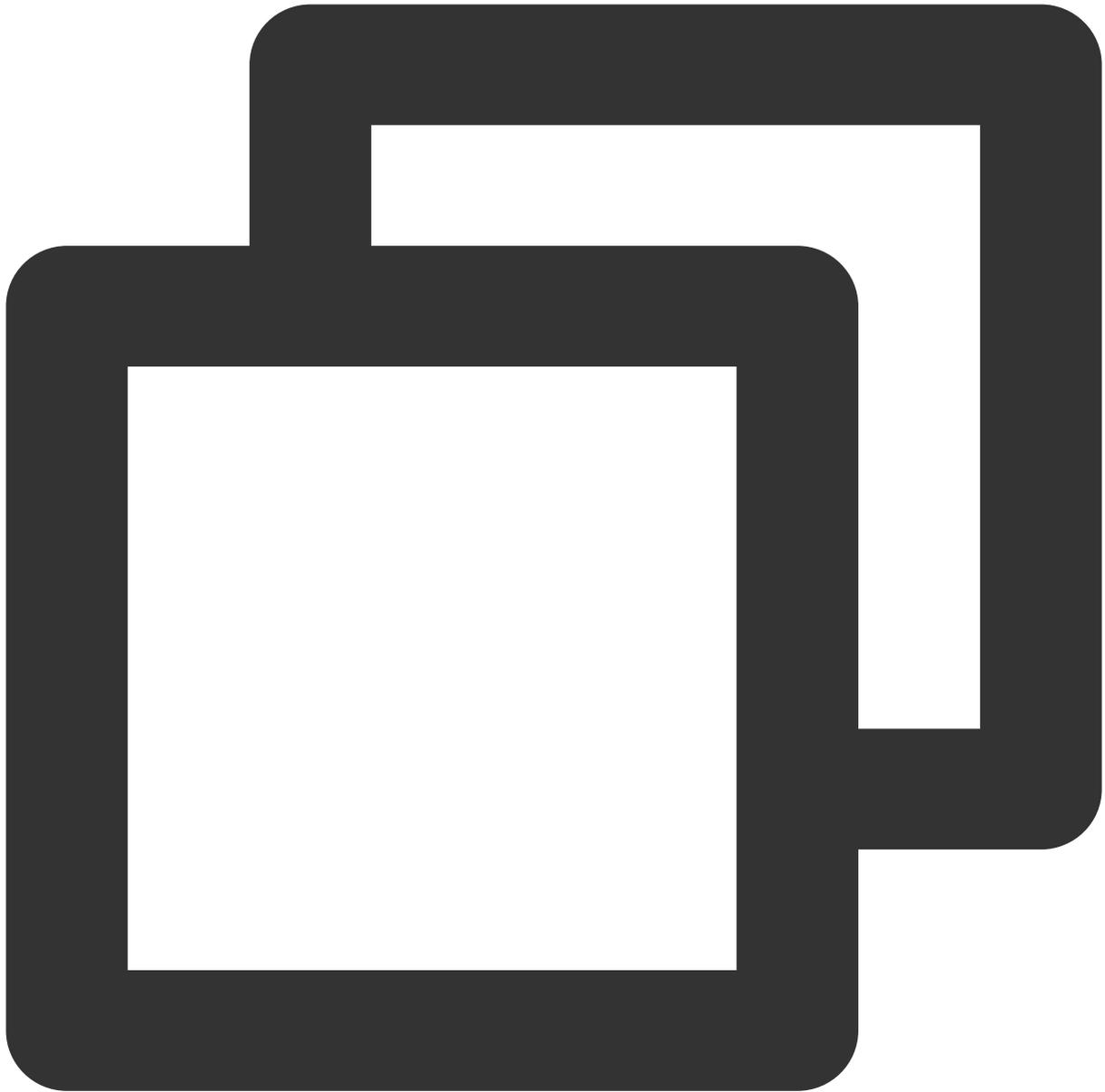
```
curl -g "http://<pod-ip>:9100/metrics"
```

大きなクラスターではipvsメトリクスを除外することをお勧めします。



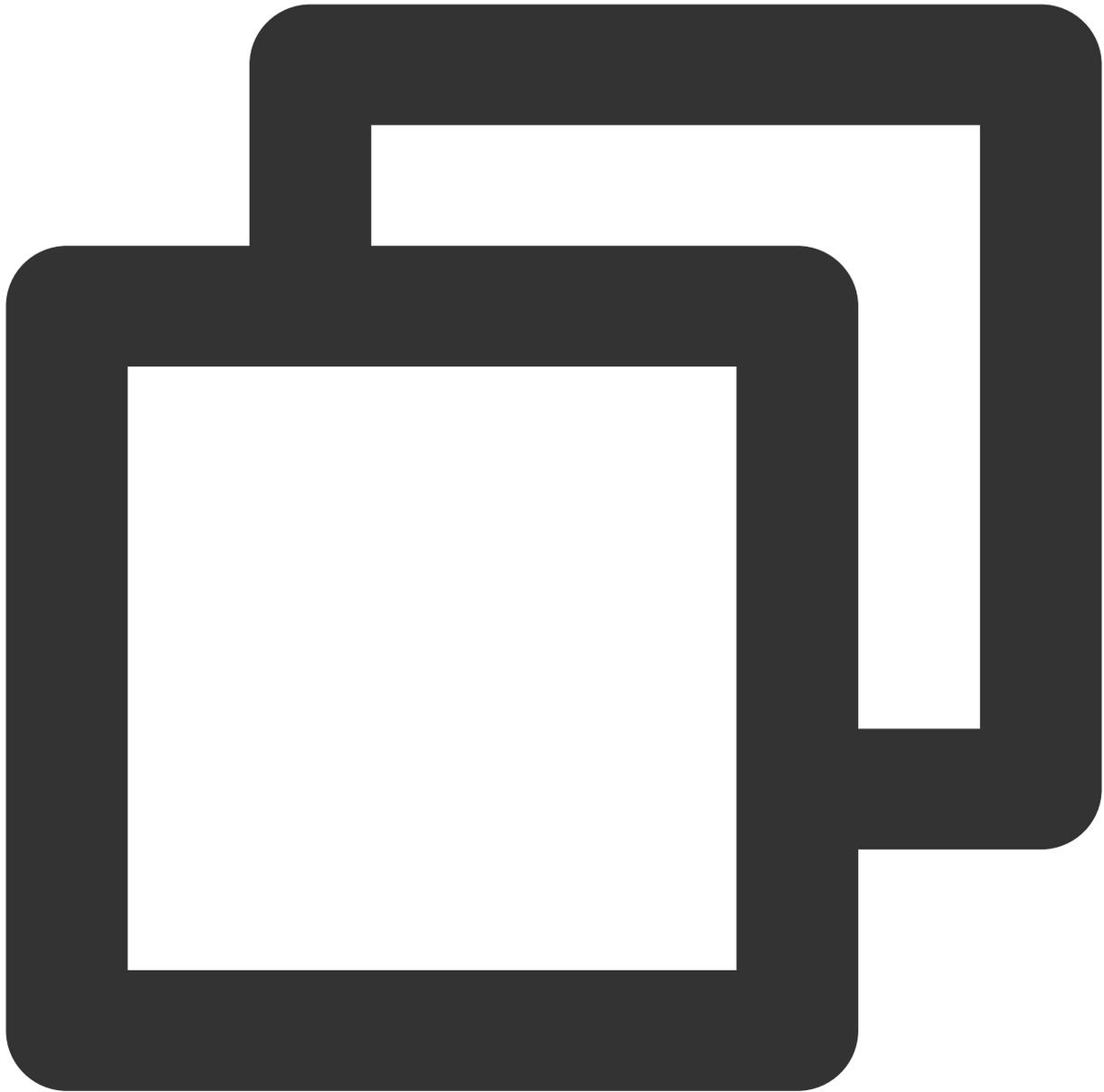
```
curl -g "http://<pod-ip>:9100/metrics?collect[]=ipvs"
```

業務自体の必要性によりポート9100をリッスンする場合は、Podの作成の際に9100以外のポートを使用してモニタリングデータを収集するようにすることで、業務のポート9100との競合を避けることができます。設定方法は次のとおりです。



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

モニタリング用の公開ポートを変更しなかった場合、業務がポート9100を直接リッスンするため、TKE Serverlessの新しいネットワークソリューションではユーザーに対し、ポート9100がすでに使用されているというエラーが通知されます。



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

エラー通知が表示された場合は、PodにAnnotation : `metrics-port` を追加してモニタリングポートを変更し、Podを再構築する必要があります。

注意

Podがパブリックeipを持っている場合はセキュリティグループを設定する必要があります。ポート9100の問題に注意し、必要なポートを許可してください。

Cloud Load Balancer (CLB) 関連

最終更新日： : 2023-04-26 19:30:26

ここではCloud Load Balancer (CLB) に関するよくあるご質問をまとめ、Service/Ingress CLB関連のよくある各種トラブルの原因とその解決方法をご紹介します。

このドキュメントは以下のお客様向けとなります。

Pod、ワークロード/Workload、Service、Ingressなど、K8Sの[基本概念](#)に精通している方。

[Tencent Cloudコンソール](#)のコンテナサービスTKE Serverlessクラスターの一般的な操作に精通している方。

kubectlコマンドラインツールによるK8Sクラスター内のリソース操作に精通している方。

注意：

K8Sクラスター内のリソースの操作はさまざまな方法で行うことができます。このドキュメントでは、Tencent Cloudコンソールによる操作、ならびにkubectlコマンドラインツールによる操作の方法についてご説明します。

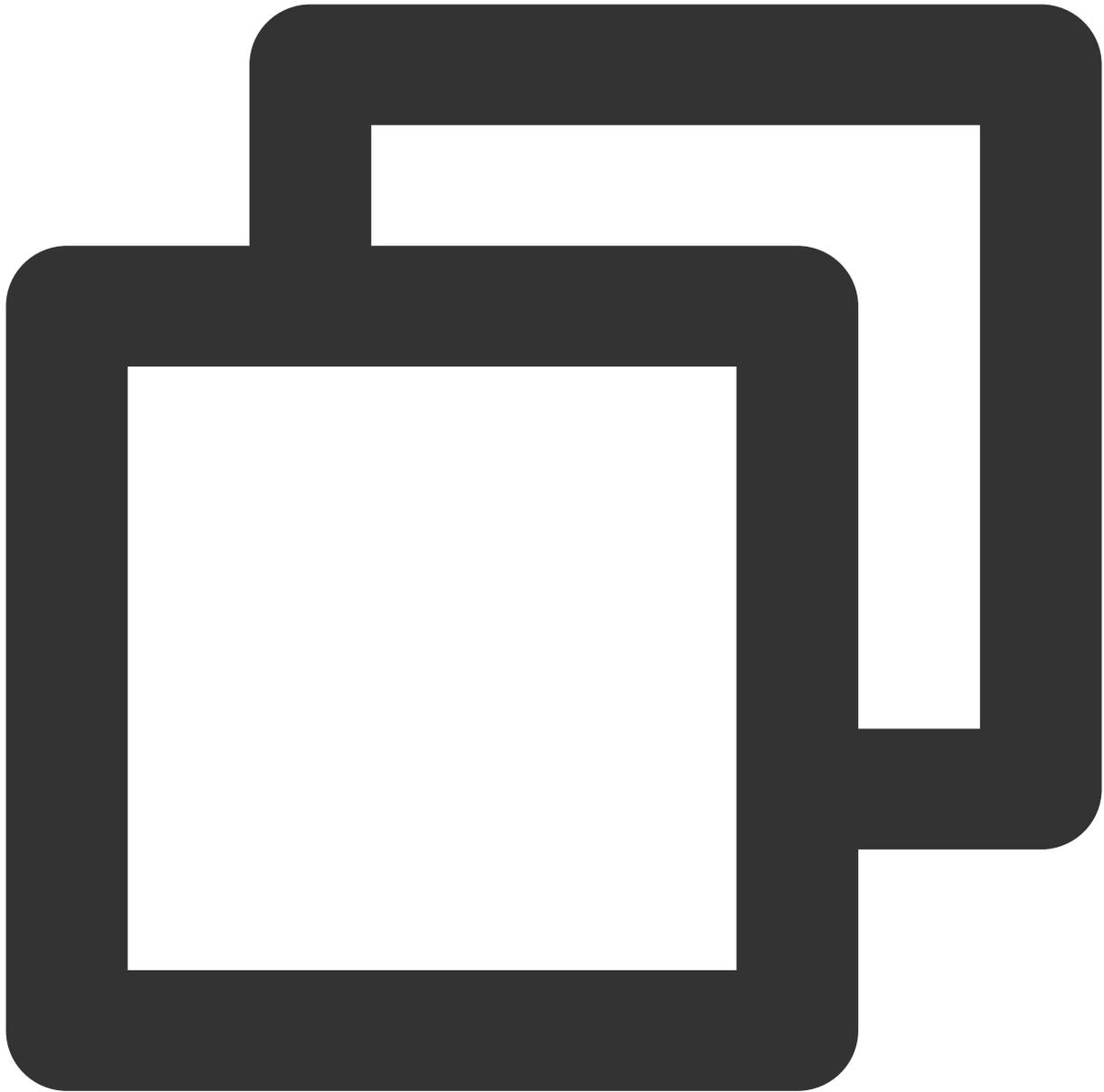
TKE ServerlessはどのIngress用にCLBインスタンスを作成できますか。

TKE Serverlessは次のような条件を満たすIngress用にCLBインスタンスを作成できます。

Ingressリソースに対する要件	その他の説明
annotations内に次のキーバリューペアが含まれること： kubernetes.io/ingress.class: qcloud	TKE ServerlessでIngress用にCLBインスタンスを作成したくない場合、例えばNginx-ingressを使用したい場合などは、annotations内に前述のキーバリューペアを含めないようにします。

TKE ServerlessのIngress用に作成したCLBインスタンスを確認するにはどうすればよいですか。

Ingress用のCLBインスタンスの作成に成功すると、TKE ServerlessはCLBインスタンスのVIPをIngressリソースの `status.loadBalancer.ingress` に書き込み、以下のキーバリューペアをannotationsに書き込みます。



```
kubernetes.io/ingress.qcloud-loadbalance-id: CLBインスタンスID
```

TKE ServerlessのIngress用に作成したCLBインスタンスを確認したい場合の具体的な手順は次のとおりです。

1. TKEコンソールにログインし、左側ナビゲーションバーの**クラスター**を選択します。
2. クラスターリストページで、クラスターIDを選択してクラスター管理ページに進みます。
3. クラスター管理ページで、左側の**サービスとルーティング** > **Ingress**を選択します。
4. 「Ingress」ページで、CLBインスタンスIDとそのVIPを確認します。下の図をご覧ください。

Ingress					
Create		Namespace	default		
Separate keywords with "; press Enter to separate					
Name	Type	VIP	Backend Service	Time Created	Operation
test	lb- Application load balancer	159.230(IPV4)		2021-02-01 11:31:43	Update Forwarding Configur Edit YAML Delete

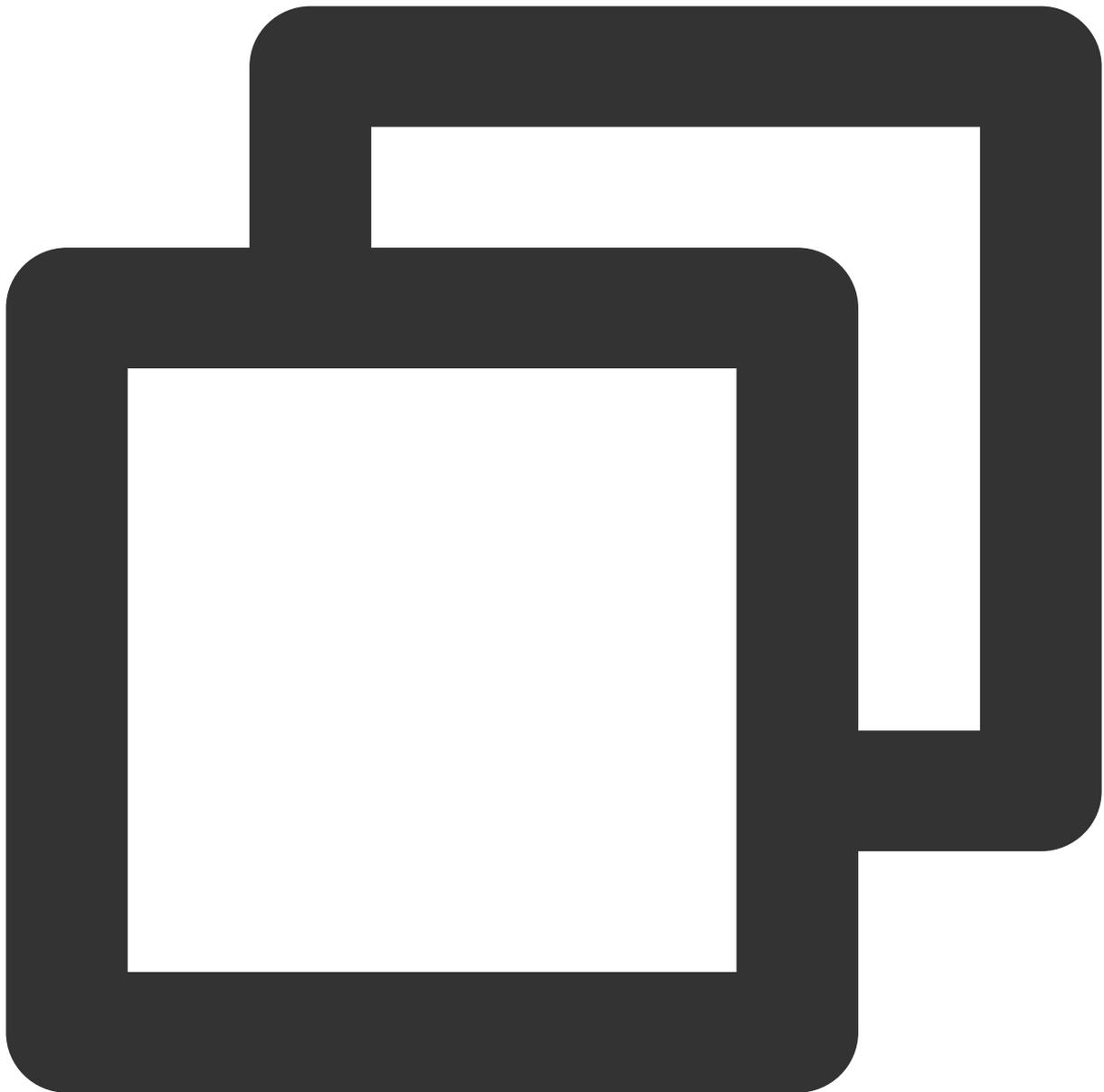
TKE ServerlessはどのService用にCLBインスタンスを作成できますか。

TKE Serverlessは次のような条件を満たすService用にCLBインスタンスを作成できます。

K8Sバージョン	Serviceリソースに対する要件
TKE ServerlessがサポートするすべてのK8Sバージョン	spec.typeがLoadBalancerである
改造版K8S（kubectl versionから返されるServer GitVersionにサフィックス「eks.」または「tke.」がついているもの）	spec.typeがClusterIPであり、かつspec.clusterIPの値がNoneではない（すなわちHeadlessではないClusterIPタイプのService）
非改造版K8S（kubectl versionから返されるServer GitVersionにサフィックス「eks.」または「tke.」がついていないもの）	spec.typeがClusterIPであり、かつspec.clusterIPを空文字列（""）に明確に指定している

注意：

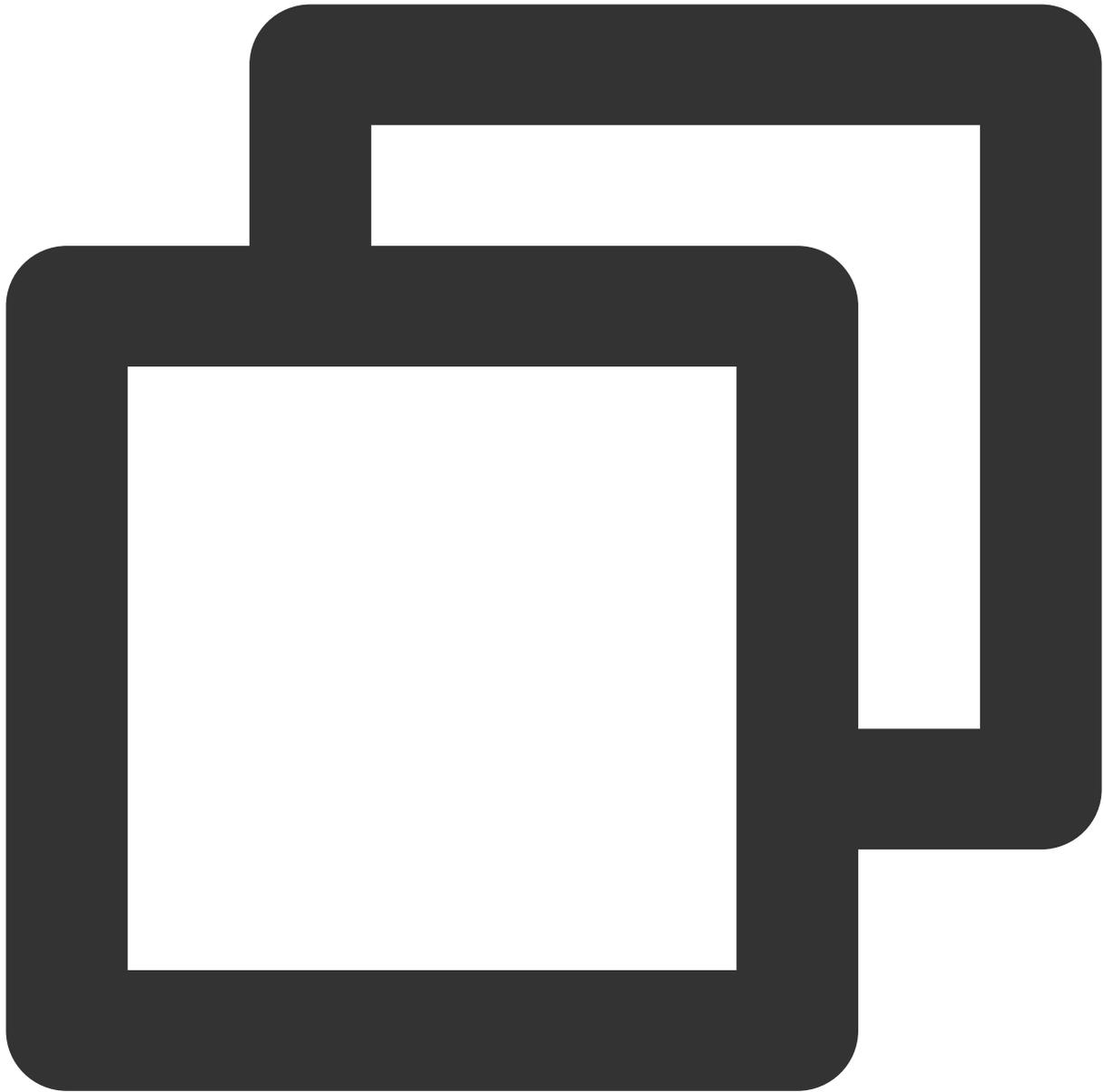
CLBインスタンスの作成に成功すると、TKE Serverlessは以下のキーバリューペアをService annotationsに書き込みます。



```
service.kubernetes.io/loadbalance-id: CLBインスタンスID
```

TKE ServerlessのService用に作成したCLBインスタンスを確認するにはどうすればよいですか。

Service用のCLBインスタンスの作成に成功すると、TKE ServerlessはCLBインスタンスのVIPをServiceリソースの `status.loadBalancer.ingress` に書き込み、以下のキーバリューペアをannotationsに書き込みます。



```
kubernetes.io/ingress.qcloud-loadbalance-id: CLBインスタンスID
```

TKE ServerlessのService用に作成したCLBインスタンスを確認したい場合の具体的な手順は次のとおりです。

1. TKEコンソールにログインし、左側ナビゲーションバーの**クラスター**を選択します。
2. クラスターリストページで、クラスターIDを選択してクラスター管理ページに進みます。
3. クラスター管理ページで、左側の**サービスとルーティング** > **Service**を選択します。
4. **Service**ページで、CLBインスタンスIDとそのVIPを確認します。下の図をご覧ください。

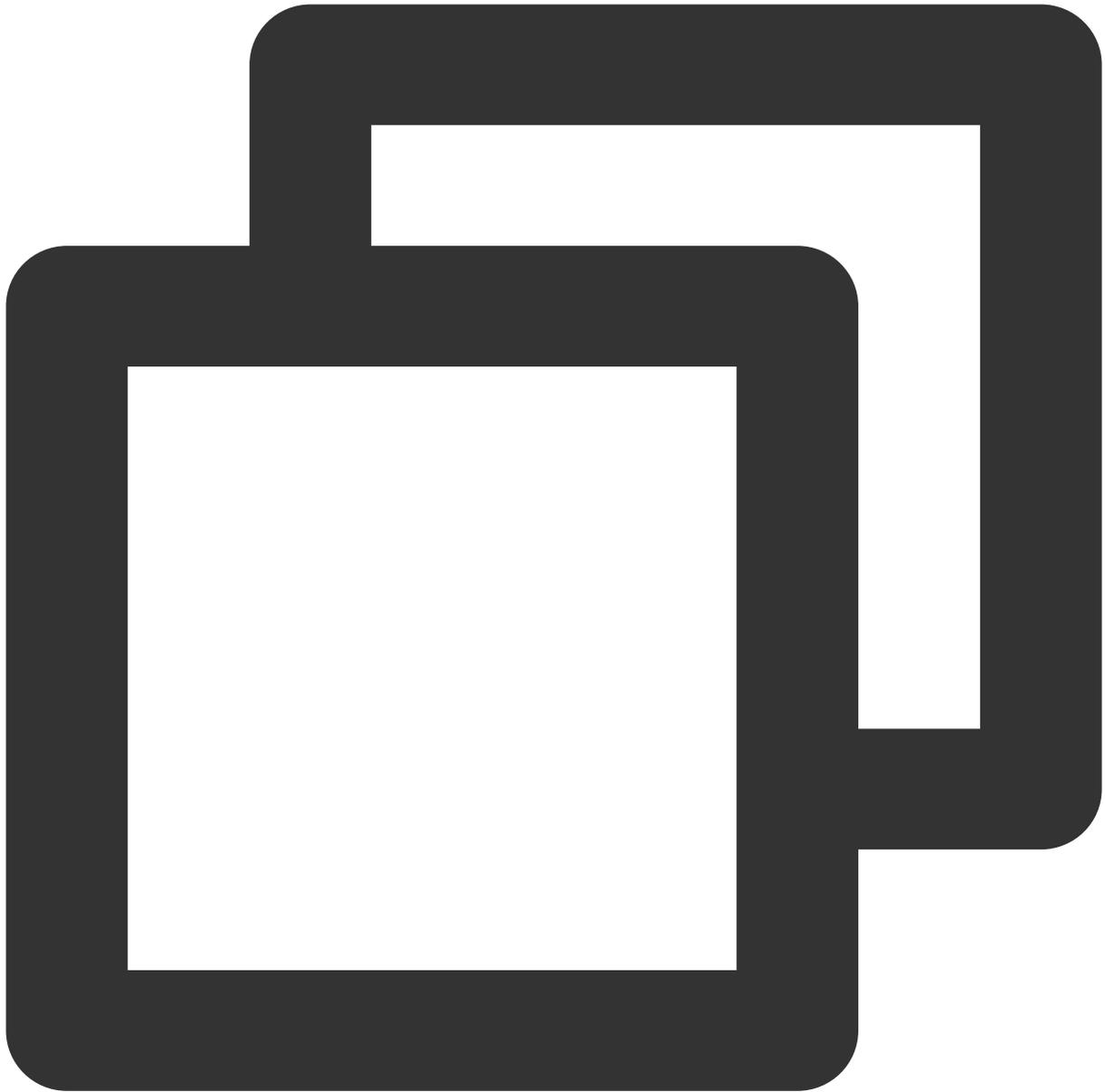
Service

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 Del
test	lb- Load Balancer	k8s-app:test, qclo...	42. ... 18(IPV4)		Update access method Edit YAML Del

ServiceのClusterIPが無効（正常にアクセスできない）またはClusterIPが存在しないのはなぜですか。

現在はspec.typeがLoadBalancerであるServiceに対して、TKE ServerlessがデフォルトでClusterIPを割り当てないか、または割り当てたClusterIPが無効（正常にアクセスできない）となっています。ユーザーが同時にClusterIPを使用してServiceにアクセスしたい場合は、annotationsに以下のキーバリューペアを追加し、TKE ServerlessがプライベートネットワークCLBをベースにClusterIPを実装するよう指示することができます。



```
service.kubernetes.io/qcloud-clusterip-loadbalancer-subnetid: Service CIDR サブネット
```

Service CIDR サブネットIDはクラスターの作成時に指定し、`subnet-*****` の文字列とします。このサブネットIDの情報はCLB基本情報ページで確認することができます。

注意：

この機能をサポートしているのは改造版K8S（`kubectl version`から返されるServer GitVersionにサフィックス「eks.」または「tke.」がついているもの）を使用したTKE Serverlessクラスターのみです。初期に作成された、非改造版K8S（`kubectl version`から返されるServer GitVersionにサフィックス「eks.」または「tke.」がついていな

いもの)を使用したTKE Serverlessクラスターについては、この機能を使用するにはK8Sをバージョンアップする必要があります。

CLBインスタンスタイプ（パブリックネットワークまたはプライベートネットワーク）を指定するにはどうすればよいですか。

CLBインスタンスタイプの指定はTKEコンソールまたはkubectIコマンドラインツールによって行うことができます。

TKEコンソールによる操作

kubectIコマンドラインツールによる操作

Ingressの場合は、「ネットワークタイプ」で「パブリックネットワーク」または「プライベートネットワーク」を選択します。

The screenshot shows the 'Create Ingress' form in the Tencent Cloud console. The form has the following fields and options:

- Ingress Name:** A text input field with a placeholder 'Please enter the Ingress name'. Below it, a note states: 'Up to 63 characters, including lowercase letters, numbers, and hyphens ("-"). It must begin with a lowercase letter, and end with a number or lowercase letter.'
- Description:** A text area with a placeholder 'Up to 1000 characters'.
- Ingress type:** A dropdown menu currently showing 'Application load balancer (supporting HTTP/HTTPS)'.
- Network type:** Two radio buttons: 'Public Network' (selected) and 'Private Network'.

Serviceの場合は「サービスアクセス方式」で制御します。「VPCプライベートネットワークアクセス」がプライベートネットワークCLBインスタンスに対応します。

← CreateService

Basic Information

Service Name
Up to 63 characters, including lowercase letters, numbers, and hyphens ("-"). It must begin with a lowercase letter, and end with a number or lowercase letter.

Description

Namespace

Access Settings (Service)

Service Access Via Internet Intra-cluster Via VPC [How to select](#)

An entry that can be accessed by other resources on the VPC of the cluster is provided. The TCP/UDP protocol is supported. Select VPC intranet access if the service needs to be accessed by other clusters and CVMs on the same VPC.

To forward through internet via HTTP/HTTPS protocol or URL, you can configure Ingress for route forwarding. [View Details](#)

デフォルトで作成したCLBインスタンスは「パブリックネットワーク」タイプです。

「プライベートネットワーク」タイプのCLBインスタンスを作成したい場合は、対応するannotationをServiceまたはIngressに追加する必要があります。

リソースタイプ	annotationsに追加する必要があるキーバリューペア
Service	service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: サブネットID
Ingress	kubernetes.io/ingress.subnetId: サブネットID

注意：

サブネットIDの形式はsubnet-*****の文字列とし、なおかつこのサブネットはクラスター作成時に「クラスターネットワーク」に指定したVPC内になければなりません。このVPCの情報はTencent Cloudコンソールクラスターの「基本情報」で照会できます。

既存のCLBインスタンスの使用を指定するにはどうすればよいですか。

TKEコンソールまたはkubectlコマンドラインツールによって、既存のCLBインスタンスの使用を指定することができます。

TKEコンソールによる操作

kubectlコマンドラインツールによる操作

ServiceまたはIngressを作成する際に、「既存のCLBインスタンスの使用」を選択することができます。Serviceの場合は、Serviceの作成後に、「アクセス方式の更新」によって「既存のCLBインスタンスの使用」に切り替えることも可能です。

Service/Ingressの作成またはServiceの変更の際に、対応するannotationをServiceまたはIngressに追加します。

リソースタイプ	annotationsに追加する必要があるキーバリューペア
---------	-------------------------------

Service	service.kubernetes.io/tke-existed-lbid: CLBインスタンスID
Ingress	kubernetes.io/ingress.existingLbId: CLBインスタンスID

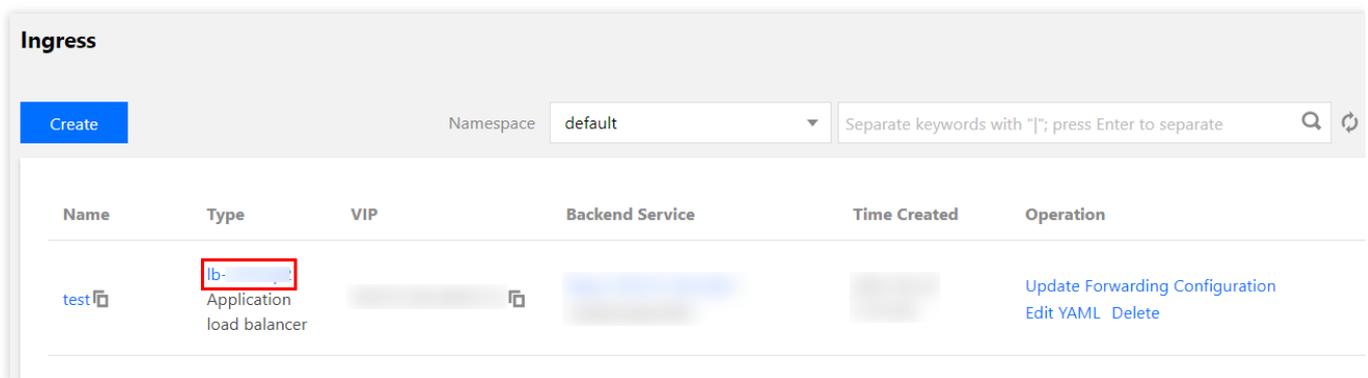
注意：

「既存のCLBインスタンス」を「TKE ServerlessがServiceまたはIngress用に作成するCLBインスタンス」にすることはできず、またTKE Serverlessは複数のService/Ingressによる同一の既存のCLBインスタンスの共用もサポートしていません。

CLBインスタンスのアクセスログを確認するにはどうすればよいですか。

アクセスログはレイヤー7 CLBインスタンスでのみ設定がサポートされていますが、TKE ServerlessがIngress用に作成するレイヤー7 CLBインスタンスでは、アクセスログはデフォルトで有効になっていません。CLBインスタンスのアクセスログを有効化したい場合は、CLBインスタンスの詳細ページで操作を行うことができます。具体的な手順は次のとおりです。

1. TKEコンソールにログインし、左側ナビゲーションバーの**クラスター**を選択します。
2. クラスターリストページで、クラスターIDを選択してクラスター管理ページに進みます。
3. クラスター管理ページで、左側の**サービスとルーティング > Ingress**を選択します。
4. 「Ingress」ページで、CLBインスタンスIDを選択してCLB基本情報ページに進みます。下の図をご覧ください。



5. CLB基本情報ページの「アクセスログ（レイヤー7）」で、

 をクリックし、ポップアップウィンドウで有効化します。下の図をご覧ください。

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 Serverlessで、IngressまたはService用のCLBインスタンスが作成できないのはなぜですか。

TKE ServerlessはどのIngress用にCLBインスタンスを作成できますか、およびTKE ServerlessはどのService用にCLBインスタンスを作成できますかの質問への回答を参照し、対応するリソースが対応する条件を満たしているかどうかを確認してください。条件を満たしているのにCLBインスタンスの作成に成功しない場合は、`kubectl describe` コマンドによって「リソース」関連イベントを確認することができます。

通常、TKE Serverlessは関連のWarningタイプのイベントを出力します。下の図の例では、使用可能なIPリソースがサブネット内にないためにCLBインスタンスの作成に失敗したことが、出力されたイベントによって明らかになっています。

```
[flyma@VM_227_154_centos ~]$
[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=ResourceInsufficient, Message=The number of IP in subnet subnet-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=ResourceInsufficient, Message=The number of IP in subnet subnet-RequestId=8567bab4-3f57-4f3d-9e99-f34d8c28d5e8
```

複数のServiceで同一のCLBを使用するにはどうすればよいですか。

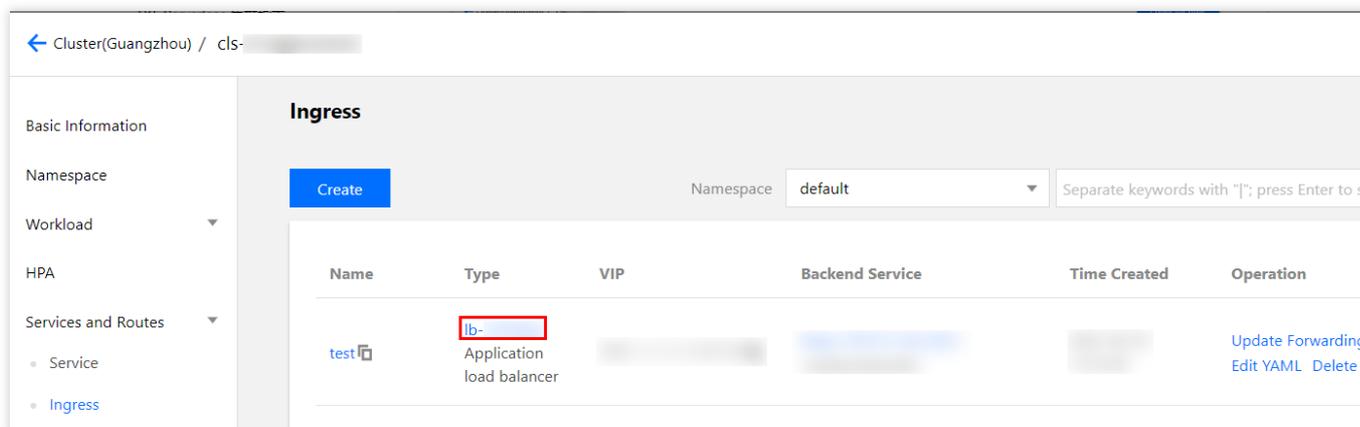
TKE Serverlessクラスターは、デフォルトでは複数のServiceが同一のCLBインスタンスを共有することはできません。他のServiceによって占有されているCLBをServiceに再利用したい場合は、このannotationを追加し、valueを「true」と入力してください。 `service.kubernetes.io/qcloud-share-existed-lb: true`。このannotationに関する詳細な説明については、[Annotationの説明](#)をご参照ください。

CLB VIPにアクセスする際に失敗したのはなぜですか。

次の手順に従って分析を行ってください。

CLBインスタンスタイプを確認する

1. 「Service」または「Ingress」ページで、CLBインスタンスIDを選択してCLB基本情報ページに進みます。下の図をご覧ください。



2. CLB基本情報ページで上記のCLBインスタンスの「インスタンスタイプ」を確認することができます。

CLB VIPへのアクセス環境が正常かどうかを確認する

CLBインスタンスの「インスタンスタイプ」がプライベートネットワークの場合、そのVIPは所属するVPC内にしかアクセスできません。

TKE Serverlessクラスター内のPodsのIPはVPC内のENIのIPであるため、Pods内ではクラスター内のどのServiceまたはIngressのCLBインスタンスのVIPにもアクセスできます。

注意

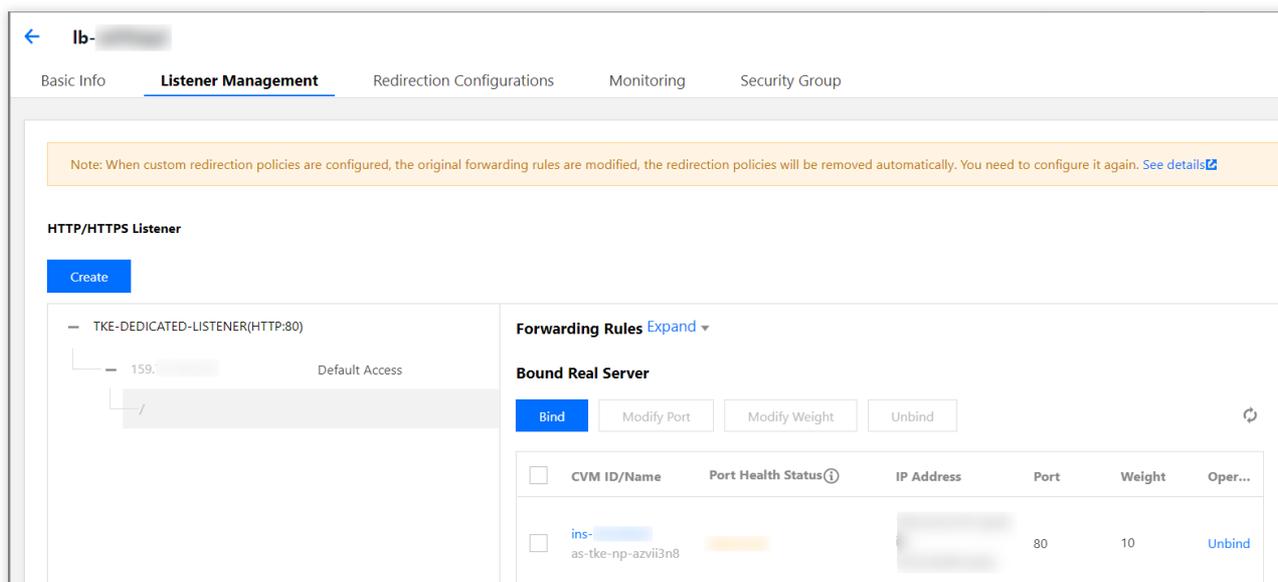
通常、LoadBalancerシステムにはループバックの問題が存在します（例：[Azure Load Balancer トラブルシューティングガイド](#)）。ワークロードが存在するPodsで、このワークロードが（ServiceまたはIngressを介して）外部に公開しているVIPによって、このワークロードが提供するサービスにアクセスすることはおやめください。つまり、PodsがVIP（「プライベートネットワークタイプ」と「パブリックネットワークタイプ」の両方を含む）によって、Pods自体が提供するサービスにアクセスしないようにします。これを行った場合は遅延が増加したり、（VIPに対応するルールにRS/Podが1つしかない場合は）アクセスできなくなったりする場合があります。

CLBインスタンスの「インスタンスタイプ」がパブリックネットワークの場合は、そのVIPはパブリックネットワークアクセス機能を持つ環境でのアクセスが可能です。

クラスター内でパブリックVIPにアクセスしたい場合は、NAT Gatewayまたはその他の方法でクラスターのパブリックネットワークアクセス機能が有効になっていることを確認してください。

CLB下のRSに、想定されるPodsのIP + ポートが含まれる（かつ、それのみが含まれる）ことを確認する

CLB管理ページで、リスナー管理ページを選択し、転送ルール（レイヤー7プロトコルの場合）、バインドされたバックエンドサービス（レイヤー4プロトコルの場合）を確認します。このうちIPアドレスは各PodのIPであることが想定されます。TKE ServerlessがあるIngress用に作成したCLBの例を次に示します。



対応するEndpointsが正常かどうかを確認する

ワークロード (Workload) にラベル (Labels) を正しく設定し、かつServiceのリソースにセレクター (Selectors) を正しく設定している場合、ワークロードのPodsで正常に実行すると、K8SによってServiceが対応するEndpointsの準備完了アドレスリストにリストアップされたPodsを、`kubectl get endpoints` コマンドによって確認できます。下の図の例をご覧ください。

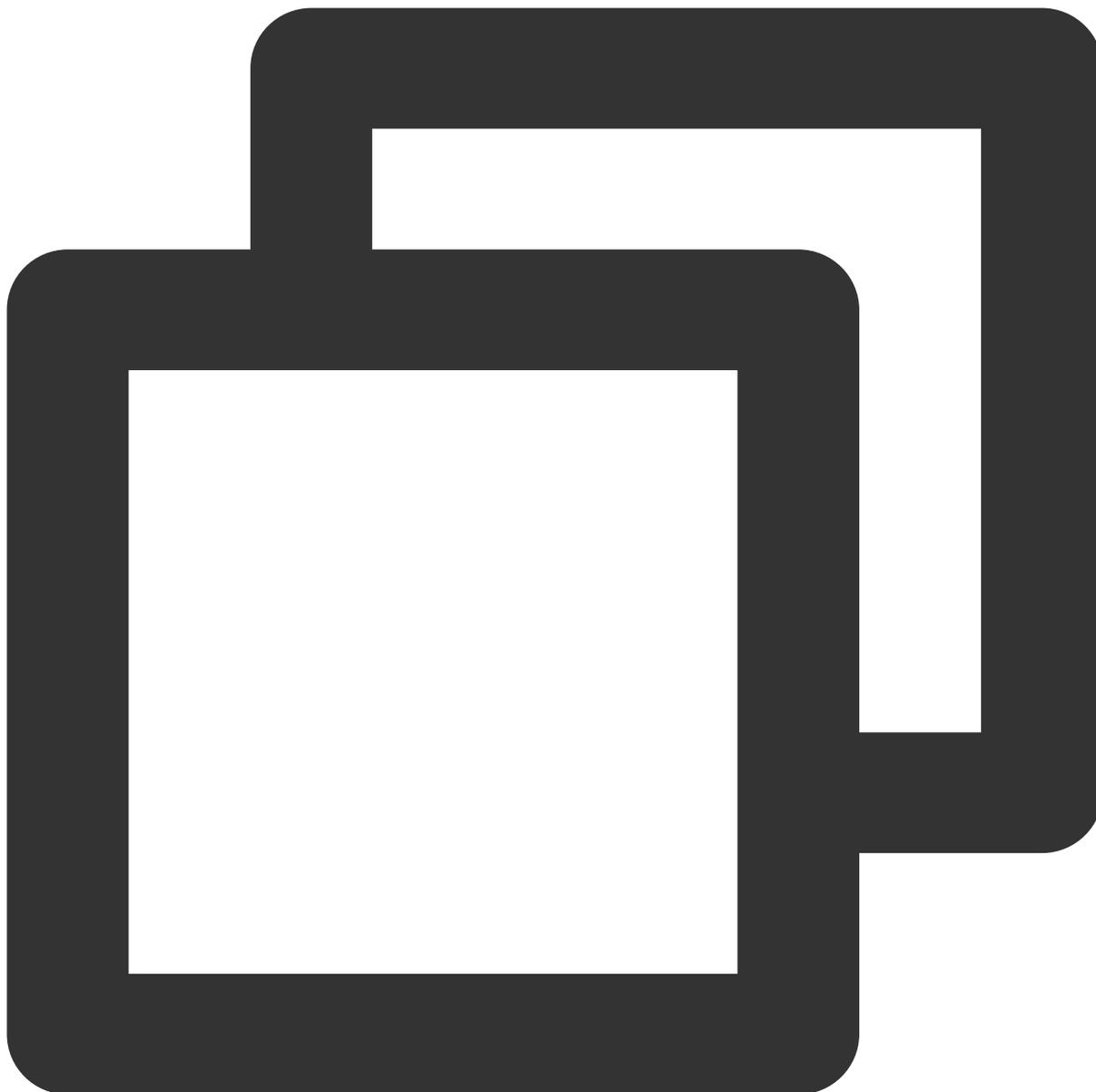
```
[flyma@VM_227_154_centos ~]$
[flyma@VM_227_154_centos ~]$ kubectl get pod -n demo -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE                                     NOMINATED N
nginx-7c7c647ff7-4b8n5  1/1    Running   0           18h   10.0.24.70   cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0   <none>
nginx-7c7c647ff7-b95zc  1/1    Running   0           18h   10.0.24.81   cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0   <none>
nginx-7c7c647ff7-ckhb9  1/1    Running   0           18h   10.1.0.174   cls-4plj5ho0-virtual-kubelet-subnet-l9dkwcyx-0   <none>
[flyma@VM_227_154_centos ~]$
[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 ~]$
[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 ~]$
[flyma@VM_227_154_centos ~]$ kubectl get ep -n demo internal-clb-nginx
NAME          ENDPOINTS          AGE
internal-clb-nginx  10.0.24.70:80,10.0.24.81:80,10.1.0.174:80  18h
[flyma@VM_227_154_centos ~]$
[flyma@VM_227_154_centos ~]$
```

また、作成済みであってもステータスが異常なPodsは、K8SによってServiceが対応するEndpointsの準備未完了アドレスリストにリストアップされます。下の図の例をご覧ください。

```
[flyma@VM_227_154_centos ~]$ kubectl get pod -n demo -o wide
NAME          READY   STATUS             RESTARTS   AGE   IP              NODE                                                                                               NOMINAT
nginx-7c7c647ff7-4b8n5    0/1     ImagePullBackOff   0          18h   10.0.24.70     cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0         <none>
nginx-7c7c647ff7-b95zc    1/1     Running            0          18h   10.0.24.81     cls-4plj5ho0-virtual-kubelet-subnet-e714bobz-0         <none>
nginx-7c7c647ff7-ckhb9    1/1     Running            0          18h   10.1.0.174     cls-4plj5ho0-virtual-kubelet-subnet-l9dkwcyx-0         <none>
[flyma@VM_227_154_centos ~]$
[flyma@VM_227_154_centos ~]$
[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 ~]$
```

注意：

異常なPodsについては、`kubectl describe` コマンドによって異常の原因を確認することができます。コマンドの例は次のとおりです。



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

Podsが正常にサービスを提供可能かどうかを確認する

PodsはステータスがRunningであっても、外部に正常にサービスを提供できない場合があります。例えば、指定のプロトコル+ポートをリッスンしていない、Pods内部の論理エラー、処理プロセスのスタックなどの場合があります。 `kubectl exec` コマンドによってPod内にログインし、 `telnet/wget/curl` コマンドまたはカスタマイズしたクライアントツールを使用してPod IP+ポートに直接アクセスすることができます。Pod内への直接アクセスに失敗する場合は、Podが正常にサービスを提供できない原因をさらに分析する必要があります。

PodにバインドしているセキュリティグループがPodsの提供するサービスのプロトコルおよびポートを許可しているかどうかを確認する

セキュリティグループはLinuxサーバーのIPTablesルールと同じように、Podsのネットワークアクセスポリシーを制御します。実際の状況に応じて確認を行ってください。

TKEコンソールによってワークロードを作成した場合

kubectlコマンドを使用してワークロードを作成した場合

インタラクションのフローで、1つのセキュリティグループを指定することが強制的に要求され、TKE Serverlessはこのセキュリティグループを使用してPodsのネットワークアクセスポリシーを制御します。ユーザーが選択したセキュリティグループはワークロードの `spec.template.metadata.annotations` に保存され、最終的にPodsのannotations内に追加されます。次に例を示します。

```
[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コマンドによってワークロードを作成し、かつ（annotationsによって）Podsにセキュリティグループを指定していない場合、TKE Serverlessはアカウント下の同一リージョンのデフォルトプロジェクトのdefaultセキュリティグループを使用します。確認の手順は次のとおりです。

1. VPCコンソールにログインし、左側ナビゲーションバーの[セキュリティグループ](#)を選択します。
2. 「セキュリティグループ」ページの上方で、同一リージョンのデフォルトプロジェクトを選択します。
3. リストでdefaultセキュリティグループを確認し、[ルールの変更](#)をクリックして詳細を見ることができます。下の図をご覧ください。

Security Group Hong Kong, China DEFAULT PROJECT Help of Security Group

Note: from December 17, 2019, Tencent Cloud adds limits on the number of security groups bound with an instance, number of instances bound to a security group, and the number of rules referenced by a security group. For details, please see [Use Limits](#)

New Separate keywords with Q

ID/Name	Associated Inst...	Notes	Type	Creation Time	Projects	Operation
sg-default	0	System created security group	Default	2020-01-14 11:35:25	DEFAULT PROJECT	Modify Rule Manage Instances More

お問い合わせ

ここままで問題の原因が見つからない場合は、問題解決のため、[チケットを提出](#)してTKEチームにご連絡ください。

イメージリポジトリ類

最終更新日：：2022-03-31 15:04:02

イメージリポジトリの有効化

ネームベースの役割はなんですか。

ネームベースは、ユーザーのプライベートイメージを識別するアドレスプレフィックスです。

イメージリポジトリのアカウントはなんですか。

デフォルトではユーザーのTencent Cloudアカウント（QQアカウント）に設定されています。

アカウント作成時のパスワードを忘れたときはどうしたらいいですか。

コンソールを通してパスワードをリセットできます。

イメージ作成

イメージ作成にクラスターの制限はありますか。

デフォルトのクォーターでは、1リージョンにつき500イメージリポジトリ、1つのイメージリポジトリにつき100イメージバージョンを作成できます。

これ以上のクォーターが必要な場合、[クォーター申請チケット](#)でクォーターを申請してください。

作成されたイメージは他のユーザーに共有できますか。

現在それに該当する機能はありません。

作成されたイメージはどのように使用しますか？

使用可能なイメージバージョンをアップロードし、それを通してサービスを作成します。

イメージ削除

どのようにイメージの指定されたバージョンを削除しますか？

コンソールから直接そのバージョンを指定して削除してください。

イメージリストからイメージを削除した場合、そのイメージのすべてのバージョンを削除したことになりますか？

はい。イメージ削除に伴い、すべてのバージョンが削除されることになるため、あらかじめデータをバックアップしてください。

イメージのビルド

ソースコードによってイメージをビルドする場合、**Dockerfile**パスとビルドディレクトリはどのように記入しますか。

- パスを記入しない場合、システムでは以下のデフォルト値を使用します：
- デフォルトの Dockerfile パス：コードリポジトリのルートディレクトリにある Dockerfile（`Dockerfile`）。
- デフォルトの ビルドディレクトリ：コードリポジトリのルートディレクトリ（`./`）。
- パスを記入する場合、**プロジェクトをルートパスとする相対パス**を記入してください。以下の図をご参照ください：

← 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.
Custom prefix, supports variables in the format of \$(Foo)

Overwrite the image tag
The generated image also contains the tag

Dockerfile path
Path of the Dockerfile in the code source

Building Directory
The working directory for building, should be a relative path

Building Parameters [Add a variable](#)

[Complete](#)

ソースコードのビルド機能で使用される **Dockerfile** パス及びビルドディレクトリの詳細はなんですか？

ソースコードビルド機能はユーザーが指定したリポジトリをクローンして、該当するブランチまたはタグに切り替えます。そして、コードリポジトリのルートディレクトリで `docker build -f $DOCKERFILE_PATH $WORKDIR` コマンドを実行し、コンテナイメージをビルドします。

Dockerfile のソースパスはなんですか。

`COPY`、`ADD` といったソースパスに関わるコマンドは、ビルドディレクトリの相対パスをソースパスとして記入してください。

リモート端末類

最終更新日：2022-03-31 15:04:02

コンテナ内にbashがないときはどうしたらいいですか。

bashがないときは、コマンドラインで実行したいコマンドを入力すると、スクリーンにコマンドの実行結果が表示されます。コマンドラインはオートコンプリート自動補完などの機能を持たない簡易的なbashとみなしてよいです。bashのコマンドを先にインストールして、後続操作を実行することを推奨します。

apt-getのインストールが遅いのはなぜですか。

apt-getのインストールが遅い原因として、サーバーが海外ソフトウェアのソースにアクセスするのが遅いことが挙げられます。コンテナのOSによって、速度向上の案を提供しています。実際の状況に応じて下記の操作をしてください。

注意事項

エラーを防ぐため、コンテナのOSをご確認の上、速度向上の解決方法を選択してください。確認方法は次の通りです。

- Ubuntu： `cat /etc/lsb-release` を実行し、 `/etc/lsb-release` が存在するかを確認。
- CentOS： `cat /etc/redhat-release` を実行し、 `/etc/redhat-release` が存在するかを確認。
- Debian： `cat /etc/debian_version` を実行し、 `/etc/debian_version` が存在するかを確認。

解決方法

Ubuntu 16.04

OSがUbuntu 16.04のコンテナは、端末で次のコマンドを実行し、aptのソースにTencent Cloudのソースサーバーを設定してください。

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

CentOS 7

OSがCentOS 7のコンテナは、端末で次の操作を実行し、ソースアドレスを直接変更することで、インストール速度を改善できます。

1. 以下のコードをコンテナ内にコピーして実行します。

```
cat << ENDOF > /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/\${basearch}
/
#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}/opens
tack-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. YUMキャッシュをクリアして再作成するために、下記のコマンドを実行してください。

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

Debian

OSがDebian のコンテナは、端末で次のコマンドを実行し、apt のソースにTencent Cloudのソースサーバーを設定してください。

```
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
```

まとめ

コンテナ内でソースアドレスを直接変更するのは臨時的な解決方法であり、コンテナがスケジューリングされた場合、その変更は無効になります。イメージ作成の時点でトラブルを解決することを推奨します。具体的な手順は次の通りです。

作成したコンテナイメージの Dockerfile を修正し、Dockerfile の RUN フィールドに [解決方法](#) に従って、対応する OS の情報を追加して、ソースアドレスを変更します。。たとえば、Ubuntuベースのイメージの Dockerfile に次の内容を追加してください。

```
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 univer
se multiverse
#deb http://mirrors.tencentyun.com/ubuntu/ xenial-backports main restricted unive
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
```

コンテナにログイン後、**vim**、**netstat**などのツールがないと気づいたらどうしたらいいですか。

`apt-get install vim`、`net-tools`などのコマンドを実行し、必要なツールをインストールできます（CentOS の場合、`yum install vim`を実行）。

apt-get install を実行したが、ツールが見つからないときはどうしたらいいですか。

以下の操作を実施してインストールしてください。

1. 次のコマンドを実行して、ソフトウェアリストをアップデートします。

```
apt-get update
```

2. 次のコマンドを実行して、ソフトウェアをインストールします（CentOS の場合、`yum updateinfo`を実行）。

```
apt-get install
```

コンテナ内で自作のツールを使用するには、どうしたらいいですか。

リモートターミナルにアクセスし、右下のドキュメントアシスタントをクリックすることで、アップロードとダウンロードが可能になります。

分析のため、**dump**や**ログ**などのライブファイルをローカルにコピーする方法は？

リモートターミナルにアクセスし、右下のドキュメントアシスタントをクリックすることで、アップロードとダウンロードが可能になります。

ドキュメントをコンテナにアップロード、またはローカルシステムにダウンロードできないのはなぜ？

原因として、コンテナイメージに tar プログラムがインストールされていないことが挙げられます。 `apt-get install vim`、`net-tools`などのコマンド（CentOS の場合 `yum install vim`）を実行し、tar プログラムをインストールしてから再試行してください。

以前インストールしたツールが消えてしまいました。

原因として、Kubernetes がコンテナを再スケジューリングする時、新しいコンテナを生成するためにイメージを取得することが挙げられます。イメージの中に以前インストールしたツールが含まれていなければ、新しいコンテナにもそれらのツールは含まれません。イメージ作成時に、よく使われるトラブルシューティングツールをインストールすることを推奨します。

コンソール上の文字をコピーするには？

コピーしたい文字列を選択すると、選択された文字列はコピーされます。

コピーした文字をペーストするには？

`Shift + Insert` でペーストできます。

コネクションが切れたのはなぜですか。

原因として、Tencent Cloud の他のページで行ったコンテナやCloud Virtual Machine (CVM) に対する操作により、コンテナが変更されたか、長時間（3分間）操作をしなかったことにより、サーバーが接続を切断したことが挙げられます。

top コマンド等の実行で TERM environment variable not set のエラーが発生した場合、どうしたらいいですか。

`export TERM linux` コマンドを実行してください。

絶対パスが長いディレクトリにアクセスすると、bash プロンプトが「<」と一部のパスのみを表示するのはなぜですか。

デフォルトの bash プロンプトが「ユーザー名@ホスト名 カレントディレクトリ」の表示に設定されているためです。現在のディレクトリパスが一定の長さを超えた場合、bash は「<」及びパスの末尾部分のみをデフォルトで表示します。