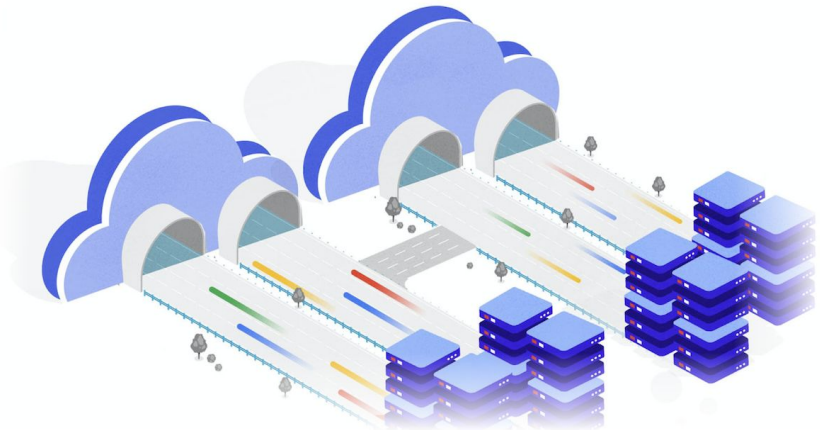
リージョン永続ディスクと マネージド インスタンス グループで構築する 高可用性構成



Google Cloud
Infrastructure Modernization Specialist
安原 稔貴

安原 "Hawk" 稔貴

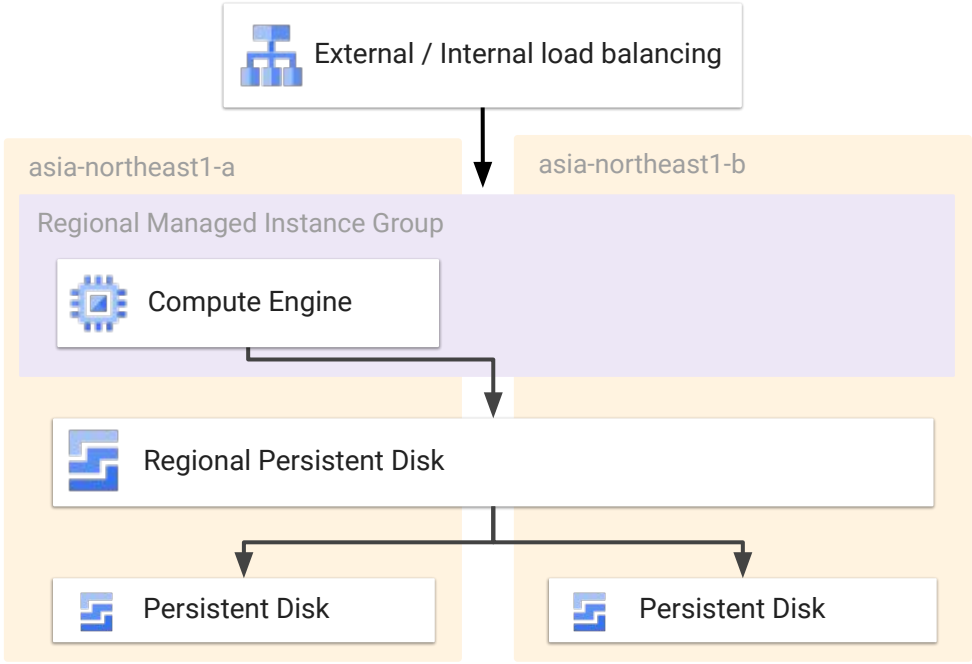
Google Cloud Japan
Infrastructure
Modernization Specialist

クラウド活用を検討中のお客様に対し、
デジタル トランスフォーメーションを
技術、戦略面より支援

Twitter: @tositaka77_ja

今日お話させて頂く構成について

- 以下組み合わせによるシンプルな HA 構成を実装可能
  - 障害の検知と自動修復にロードバランサとリージョンマネージド インスタンス グループを利用
  - 複数ゾーンのデータ同期にリージョン永続ディスクを利用
- ウォーム スタンバイ用の VM リソースが不要



参考 URL : https://cloud.google.com/solutions/architectures-high-availability-postgresql-clusters-compute-engine

本構成を採用する際の注意点

- ゾーン障害発生時は手動での切り替えが必要
  - 同一ゾーン内での自動修復は行われるが、ゾーン障害時に自動では
    切り替わらないので手動対応が必要（パイロット ライト方式）
- リージョン永続ディスク以外に保存されたデータは初期化される
  - リージョン永続ディスクに保管されたデータ以外はインスタンス テンプレートとして
    管理されるため、パッチ適応後などにインスタンス テンプレートの更新と適応が必要
- リージョン永続ディスクの性能
  - リージョン永続ディスクにした場合は性能がゾーン永続ディスクに
    比べて低くなるため、高いパフォーマンス要件が必要な場合は注意が必要

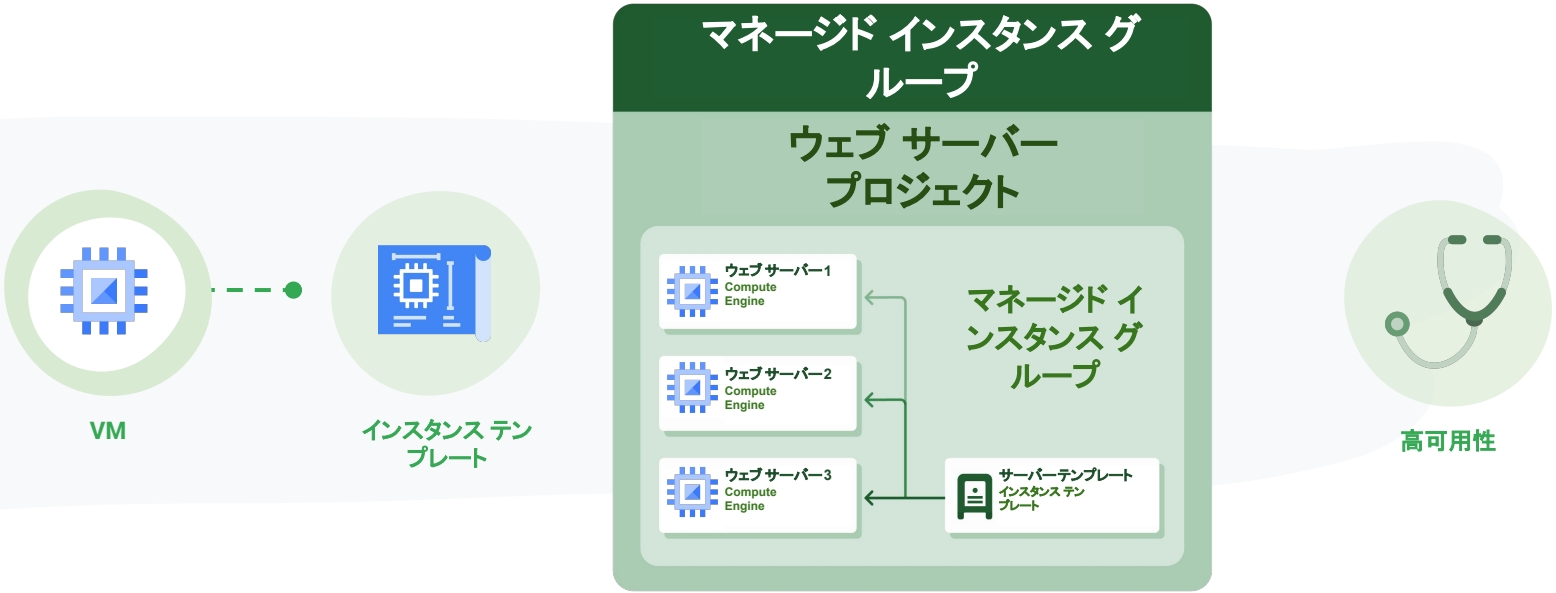
本日の Agenda

- 各コンポーネントの解説
  - リージョン マネージド インスタンス グループ
  - リージョン永続ディスク
- 構築手順の解説
- デモ（障害発生時の自動修復）

各コンポーネントの解説

01

マネージド インスタンス グループ（MIG）



1 つのインスタンス テンプレートで
複数のインスタンスをデプロイ

参考 URL : https://cloud.google.com/compute/docs/instance-groups#managed_instance_groups

リージョン マネージド インスタンス グループについて

- リージョン MIG は、リージョン内の複数のゾーンに VM を配置する
- 指定されたインスタンス数の上限の 1/n までを上限に、それぞれのゾーンに VM を配置する（ n は利用するゾーンの数）
  ※ ターゲット分配形態が均等（デフォルト）の場合
  - （例）3 ゾーンを使用しており、maxNumReplicas の値が 15 の場合、ゾーンごとに最大 15 / 3 = 5 個 の VM が配置される
    この際、1 つのゾーンに障害が発生した場合、オートスケーラーが設定できる　VM 数は、残り 2 つのゾーンであわせて maxNumReplicas の値の 2/3 までとなる

参考 URL : https://cloud.google.com/compute/docs/instance-groups/regional-migs

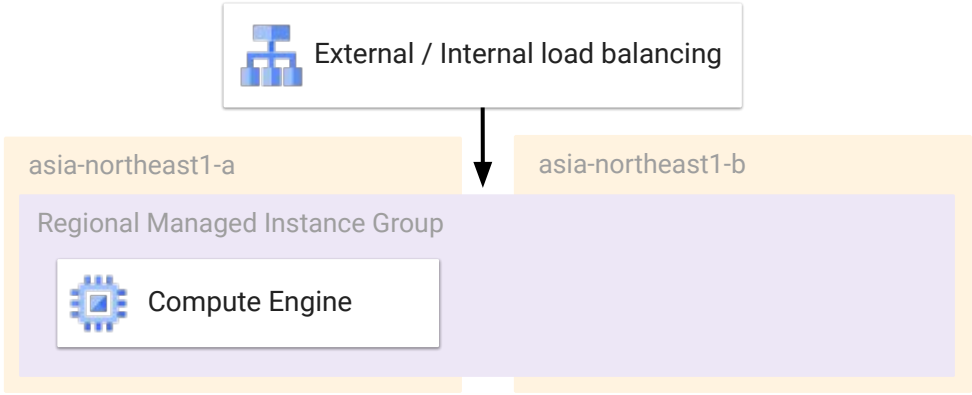
マネージド インスタンス グループの
自動修復機能とヘルスチェックについて

- MIG のデフォルトの動作では VM の稼働ステータスが RUNNING かどうかで障害が発生しているかを検知し、インスタンスの再作成による自動修復を行う
- 自動修復の際には**インスタンス テンプレートで指定されたとおりにディスクが再作成**され、VM が起動する
- 自動修復ポリシーをアプリケーション ベースのヘルスチェックともに設定することで、アプリケーションが応答していない場合も
  自動修復が実行されるように構成することが可能

参考 URL : https://cloud.google.com/compute/docs/instance-groups/autohealing-instances-in-migs

マネージド インスタンス グループとロードバランサ

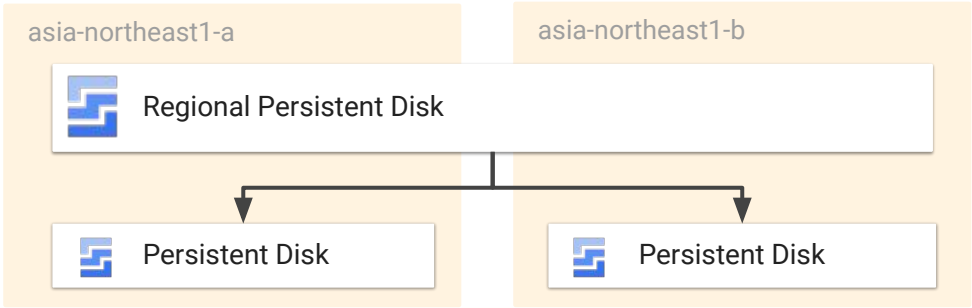
- ロードバランサのバックエンドとして MIG を指定可能
  - MIG のオートスケールにより稼働している VM のインスタンス数が変化しても自動的に振り分け先に含められる
- **今回の構成では、VM が稼働するゾーンが変わっても、リージョン MIG をバックエンドとして指定することで接続可能に**



参考 URL : https://cloud.google.com/compute/docs/instance-groups/adding-an-instance-group-to-a-load-balancer

リージョン永続ディスク

- リージョン永続ディスクでは、データをディスクの作成時に選択したゾーンに自動的に同期する
- 各ゾーンのデータは、冗長性を確保するために、ゾーン内の複数の物理マシンに分散される
- リージョン永続ディスクは、メモリ最適化マシンとコンピューティング最適化マシンでは使用できない



参考 URL : https://cloud.google.com/compute/docs/disks?hl=ja#repds
https://cloud.google.com/solutions/design-considerations-for-resilient-workloads-with-regional-persistent-disks

リージョン永続ディスクのパフォーマンス

- 右記表のとおり、
  リージョン永続ディスクを利用する
  場合はゾーン永続ディスクよりも
  パフォーマンスが劣る為、
  ストレージのパフォーマンス要件が
  厳しいワークロードに適用する
  際には注意が必要
- その他のタイプの定義も含め、
  詳細は下記 URL を参照

	ゾーン SSD PD	リージョン SSD PD
最大持続 IOPS		
読み取り IOPS / GB	30	30
書き込み IOPS / GB	30	30
読み取り IOPS / インスタンス	15,000 ～ 100,000	15,000 ～ **60,000**
書き込み IOPS / インスタンス	15,000 ～ 100,000	15,000 ～ **30,000**
最大持続スループット（MB/秒）		
読み取りスループット/ GB	0.48	0.48
書き込みスループット/ GB	0.48	0.48
読み取りスループット/ インスタンス	240 ～ 1,200	240 ～ 1,200
書き込みスループット/ インスタンス	240 ～ 1,200	**120** ～ **600**

参考 URL : https://cloud.google.com/compute/docs/disks/performance?hl=ja#type_comparison

構築手順の解説

02

本構成の構築の流れ

1. ベースとなる VM を構築
2. VM をインスタンス テンプレートとして登録
3. 自動修復ポリシー用のヘルスチェックを作成
4. リージョン MIG を作成
5. インスタンス グループへリージョン永続ディスクを利用した
   インスタンスを追加
6. インスタンス グループをロードバランサに追加する

ベースとなる VM を構築

- リージョン永続ディスクを接続した VM を作成、利用する MW 等をインストールする
  ※ 今回は PostgreSQL を想定
- MW の構成として、リージョン永続ディスクへデータを保存するように構成する
- リージョン永続ディスク以外に保存されたデータは、自動修復時に
  この後作成するインスタンス テンプレート時点に戻ってしまうので注意

サンプル コマンド

リージョン永続ディスクを作成	gcloud compute disks create sample-data-disk \ --size 200GB \ --region asia-northeast1 \ --replica-zones asia-northeast1-a,asia-northeast1-b
リージョン永続ディスクを接続した VM を作成	gcloud compute instances create sample-vm-instance \ --zone=asia-northeast1-a \ --disk=name=sample-data-disk,device-name=sample-data-disk,scope=regional \ --machine-type=n1-standard-4 \ --image-project=debian-cloud \ --image-family=debian-10

VM をインスタンス テンプレートとして登録

- 作成した VM をもとにインスタンス テンプレートを作成、
  インスタンス テンプレートを作成する際にリージョン永続ディスクは含めず、
  ブートディスクのみのインスタンス テンプレートを作成する

サンプル コマンド

VM の停止と リージョン永続ディスクの切り離し	gcloud compute instances stop --zone=asia-northeast1-a sample-vm-instance gcloud beta compute instances detach-disk sample-vm-instance \ --zone=asia-northeast1-a \ --disk-scope=regional \ --disk=sample-data-disk
VM から ディスクイメージの作成	gcloud compute images create sample-boot-disk-image \ --source-disk sample-vm-instance \ --source-disk-zone asia-northeast1-a \ --storage-location asia-northeast1 \ --force
ディスクイメージから インスタンス テンプレートの作成	gcloud compute instance-templates create sample-template \ --boot-disk-device-name boot-disk \ --boot-disk-size=10GB \ --image=sample-boot-disk-image \ --machine-type n1-standard-4

自動修復ポリシー用のヘルスチェックを作成

- アプリケーションのステータスをモニターするためのヘルスチェックを作成、こちらを利用し自動修復ポリシーを設定する
- 以下サンプル コマンドについてはポートの Listen 状態の確認だが、pgdoctor などを利用することも可能

参考 URL : https://github.com/thumbtack/pgdoctor

サンプル コマンド

ヘルスチェック用の FW ルールの作成	gcloud compute firewall-rules create allow-health-check-pg \ --allow tcp:5432 \ --source-ranges 130.211.0.0/22,35.191.0.0/16 \ --network default
ヘルスチェックの作成	gcloud compute health-checks create tcp pg-check \ --port 5432 \ --check-interval 10s \ --healthy-threshold 1 \ --timeout 5s \ --unhealthy-threshold 3

リージョン マネージド インスタンス グループを作成

- 次のステップでリージョン永続ディスクを利用したインスタンスを追加するため、リージョン MIG をインスタンス数 0 で作成する
- ゾーンの指定（--zones）に関してはリージョン永続ディスクの同期対象のゾーンを指定する

サンプル コマンド

インスタンス数 0 で リージョン MIG を作成	gcloud beta compute instance-groups managed create sample-mig \ --template sample-template \ --zones asia-northeast1-a,asia-northeast1-b \ --size 0 \ --instance-redistribution-type none \ --health-check=pg-check \ --initial-delay=2m

インスタンス グループへリージョン永続ディスクを利用したインスタンスを追加

- 作成したリージョン MIG へリージョン永続ディスクを利用したインスタンスを追加する
- 利用するリージョン永続ディスクの指定は URI で行う
  ※ 下記サンプル コマンドの ${DEVSHELL_PROJECT_ID} は対象の Project ID で読み替え

サンプル コマンド

作成した リージョン MIG へ インスタンスを追加	export STATEFUL_DISK_URI=projects/${DEVSHELL_PROJECT_ID}/regions/asia-northeast1/disks/sample-data-disk gcloud beta compute instance-groups managed create-instance sample-mig \ --instance sample-mig-node \ --stateful-disk device-name=pg-data-disk,source=${STATEFUL_DISK_URI},mode=rw,auto-delete=never \ --region asia-northeast1

インスタンス グループをロードバランサに追加する

- リージョン MIG をロードバランサの振り分け先に設定する

サンプル コマンド

バックエンド サービスを作成	gcloud compute backend-services create sample-ilb \ --load-balancing-scheme internal \ --region asia-northeast1 \ --health-checks pg-check \ --protocol tcp
バックエンド サービスに、インスタンス グループを追加	gcloud compute backend-services add-backend sample-ilb \ --instance-group sample-mig \ --region asia-northeast1
ロードバランサの転送ルールを作成	gcloud compute forwarding-rules create sample-ilb-forwarding-rule \ --load-balancing-scheme internal \ --ports 5432 \ --network default \ --subnet default \ --region asia-northeast1 \ --backend-service sample-ilb

ゾーン障害発生時のゾーンの切り替え方法

- ゾーン障害発生時にゾーン間のフェイルオーバーは自動的に行われない
- 以下のコマンドにて、インスタンスの削除と再作成にて稼働するゾーンを変更する必要がある

サンプル コマンド

インスタンスの削除	gcloud beta compute instance-groups managed resize sample-mig \
	  --region=asia-northeast1
	  --size=0

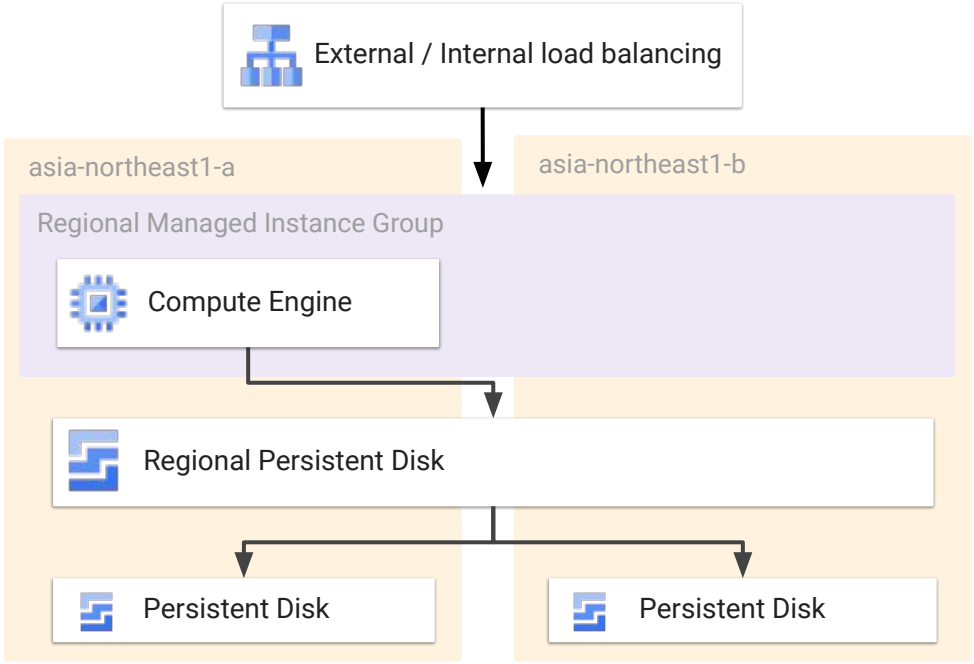
インスタンスの再追加	export STATEFUL_DISK_URI=projects/${DEVSHELL_PROJECT_ID}/regions/asia-northeast1/disks/sample-data-disk
	gcloud beta compute instance-groups managed create-instance sample-mig \
	  --instance sample-mig-node \
	  --stateful-disk device-name=pg-data-disk,source=${STATEFUL_DISK_URI},mode=rw,auto-delete=never \
	  --region asia-northeast1

デモ

03

デモ

- 構成のご紹介
- 障害発生時の自動修復のデモ

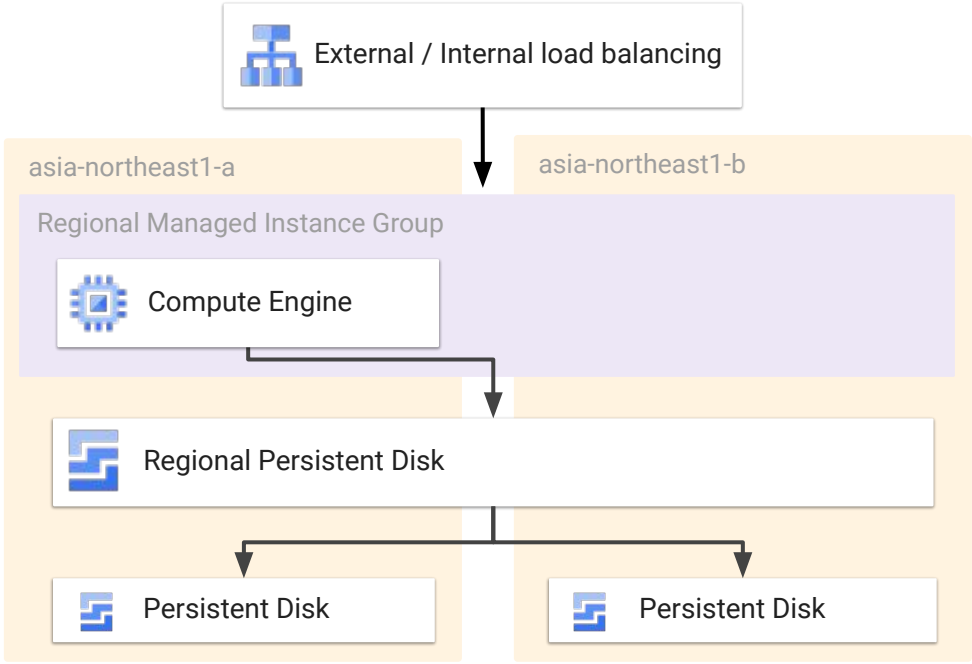


まとめ

04

今日お話させて頂く構成について

- 以下組み合わせによるシンプルな HA 構成を実装可能
  - 障害の検知と自動修復にロードバランサとリージョンマネージド インスタンス グループを利用
  - 複数ゾーンのデータ同期にリージョン永続ディスクを利用
- ウォーム スタンバイ用の VM リソースが不要



参考 URL : https://cloud.google.com/solutions/architectures-high-availability-postgresql-clusters-compute-engine

本構成の構築の流れ

1. ベースとなる VM を構築
2. VM をインスタンス テンプレートとして登録
3. 自動修復ポリシー用のヘルスチェックを作成
4. リージョン MIG を作成
5. インスタンス グループへリージョン永続ディスクを利用した
   インスタンスを追加
6. インスタンス グループをロードバランサに追加する

Google Cloud ソリューション デザインパターン

- **高可用性を確保する為のデザイン パターン**

  https://events.withgoogle.com/solution-design-pattern-infra-rdb-network/high-availability/

Thank you