Designing robust systems

This document describes best practices for designing robust systems on Compute Engine. It provides general advice and covers some features in Compute Engine that can help mitigate instance downtime and prepare for times when your virtual machine (VM) instances unexpectedly fail.

A robust system is a system that can withstand a certain amount of failures or disruptions without interrupting your service or affecting your users' experience using your service. While Compute Engine makes every effort to prevent such disruptions, certain events are unpredictable, and it is best to be prepared for these events.

Understanding types of failures

At some point, one or more of your VM instances might be lost due to system or hardware failures. Some of the failures include but are not limited to:

- **Unexpected single instance failure**

  Unexpected single instance failures can be due to hardware or system failure. To mitigate these events, use persistent disks (/compute/docs/disks/add-persistent-disk) and startup scripts (#startup) to save your data and re-enable software after you restart the instance.

- **Unexpected single instance reboot**

  At some point in time, you will experience an unexpected single instance failure and reboot. Unlike unexpected single instance failures, your instance fails and is automatically rebooted by the Compute Engine service. To help mitigate these events, back up your data (#backup), use persistent disks (/compute/docs/disks/add-persistent-disk), and use startup scripts (#startup) to quickly re-configure software.

- **Zone or region failures**

  Zone and region failures are rare failures that can cause all of your instances in a given zone or region to be inaccessible or fail.

  To mitigate these failures, create diversity across regions and zones (#distribute) and implement load balancing (#loadbalancing). You should also back up your data (#backup) or replicate your persistent disks (/compute/docs/disks#repds) across multiple zones.

Tips for designing robust systems

To help mitigate instance failures, you should design your application on the Compute Engine service to be robust against failures, network interruptions, and unexpected disasters. A robust system should be able to gracefully handle failures, including redirecting traffic from a downed instance to a live instance or automating tasks on reboot.

Here are some general tips to help you design a robust system against failures.

Use live migration

Google periodically performs maintenance on its infrastructure by patching systems with the latest software, performing routine tests and preventative maintenance, and generally ensuring that our infrastructure is as secure, fast, and efficient as possible. Compute Engine employs **live migration** to ensure that this infrastructure maintenance is transparent by default to your virtual machine instances.

Live migration is a technology that Google has built to move your running instances away from systems that are about to undergo maintenance work. Compute Engine does this automatically.

During live migration, your instance might experience a decrease in performance for a short period of time. You also have the option to configure your virtual machine instances to terminate and reboot away from the maintenance event. This option is suitable for instances that demand constant, maximum performance, and when your overall application is built to handle instance failures or reboots.

To configure your virtual machines for live migration or to configure it to reboot instead of migrate, see Setting instance scheduling options (/compute/docs/instances/setting-instance-scheduling-options).

For more information about live migration, see the Live migration (/compute/docs/instances/live-migration) documentation.

Distribute your instances

Create instances across more than one region and zone so that you have alternative virtual machine instances to point to if a zone or region containing one of your instances is disrupted.

If you host all your instances in the same zone or region, you will not be able to access any of these instances if that zone or region becomes unreachable.

Use zone-specific internal DNS names

If you use internal DNS names (/compute/docs/internal-dns) or instance names to address instances on your Compute Engine internal network, use zonal DNS names. Internal DNS servers are distributed across all zones, so you can rely on zonal DNS names to resolve even if there are failures in other locations. An internal fully qualified domain name (FQDN) for an instance has the following formats:

- Instances using zonal DNS (/compute/docs/internal-dns#zonal-dns): `[INSTANCE_NAME].[ZONE].c.[PROJECT_ID].internal`
- Instances using global DNS: `[INSTANCE_NAME].c.[PROJECT_ID].internal`

where:

- `[INSTANCE_NAME]` is the name of the instance.
- `[ZONE]` is the zone where your instance is located.
- `[PROJECT_ID]` is the project to which the instance belongs.

To check if an instance uses zonal DNS names or global DNS names, see Viewing the DNS name of the instance (/compute/docs/internal-dns#view_instance_dns_name).

If your project uses global DNS names, you can prepare your applications for zonal DNS names and search paths. For more information, see Migrating to zonal DNS names (/compute/docs/internal-dns#migrating-to-zonal).

Create groups of instances

Use managed instance groups (/compute/docs/instance-groups#managed_instance_groups) to create homogeneous groups of instances so that load balancers can direct traffic to more than one VM instance in case a single instance becomes unhealthy.

Managed instance groups also offer features like autoscaling (/compute/docs/autoscaler) and autohealing (/compute/docs/instance-groups/autohealing-instances-in-migs). Autoscaling lets you deal with spikes in traffic by scaling the number of VM instances up or down based on specific

signals, while autohealing performs health checking and if necessary, automatically recreates unhealthy instances.

Managed instance group are also available for regions so you can create a group of VM instances distributed across multiple zones within a single region. For more information, read Distributing instances using regional managed instance groups (/compute/docs/instance-groups/distributing-instances-with-regional-instance-groups).

Use load balancing

Google Cloud offers a load balancing service that helps you support periods of heavy traffic so that you don't overload your instances. With the load balancing service, you can:

- Deploy your application on instances within multiple zones using regional managed instance groups (/compute/docs/instance-groups#types_of_managed_instance_groups). Then, you can configure a forwarding rule (/load-balancing/docs/forwarding-rule-concepts) that can spread traffic across all virtual machine instances in all zones within the region. Each forwarding rule can define one entry point to your application using an external IP address.
- Deploy instances across multiple regions using global load balancing. HTTP(S) load balancing enables your traffic to enter the Google Cloud system at the location nearest the client. Cross-regional load balancing (/load-balancing/docs/https/cross-region-example) provides redundancy so that if a region is unreachable, traffic is automatically diverted to another region so that your service remains reachable using the same external IP address.
- Use autoscaling (/compute/docs/autoscaler) to automatically add or delete instances from a managed instance group based on increases or decreases in load.

In addition, the load balancing service also offers VM health checking, providing support in detecting and handling instance failures.

For more information, see the load balancing overview (/load-balancing/docs/load-balancing-overview) and load balancing documentation (/load-balancing/docs).

Use startup and shutdown scripts

Compute Engine offers startup and shutdown scripts that run when an instance boots up or shuts down, respectively. These scripts can automate tasks like installing software, running updates, making backups, logging data, and so on, when your instance first starts up or when your instance is shut down, either intentionally or not.

Both startup and shutdown scripts are an efficient and invaluable way to bootstrap or cleanly shut down your instances. Instead of configuring your instances using custom images, it can be beneficial to configure instances using startup scripts. Startup scripts run whenever the instance is rebooted or restarted due to failures, and can be used to install software and updates, and to ensure that services are running within the VM. Coding the changes to configure an instance in a startup script is easier than trying to figure out what files or bytes have changed on a custom image.

Shutdown scripts can perform last minute tasks like backing up data, saving logs, and gracefully terminating connections before you stop an instance.

For more information, see Running startup scripts (/compute/docs/startupscript) and Running shutdown scripts (/compute/docs/shutdownscript).

Back up your data

Back up your data regularly and in multiple locations. You can back up your files to Cloud Storage, create persistent disk snapshots (/compute/docs/disks/create-snapshots), or replicate your data to a persistent disk in another region or zone.

To copy files from an instance to Cloud Storage:

1. Log into your instance:

   ```
   gcloud compute ssh example-instance
   ```

2. If you have never used the `gsutil` tool on this instance, set up your credentials.

   ```
   gcloud init
   ```

   Alternatively, if you have set up your instance to use a service account (/compute/docs/access/service-accounts) with a Cloud Storage scope, you do can skip this

and the next step.

3. Follow the instructions to authenticate to Cloud Storage.
4. Copy your data to Cloud Storage by using the following command:

```
gsutil cp <file1> <file2> <file3> ...  gs://<your bucket>
```

You can also use the `gcloud compute` tool to copy files to a local computer. For more information, see Copying files to or from an instance (/compute/docs/instances/transfer-files).

What's next

- Learn more about live migration (/compute/docs/instances/live-migration).
- Create startup scripts (/compute/docs/startupscript).
- Create shutdown scripts (/compute/docs/shutdownscript).
- Learn more about managed instance groups (/compute/docs/instance-groups).
- Read more about load balancing (/compute/docs/load-balancing).

Last updated 2020-07-30 UTC.