Configuring stateful persistent disks in MIGs

ature is covered by the Pre-GA Offerings Terms (/terms/service-terms#1) of the Google Cloud Platform Terms
. Pre-GA features may have limited support, and changes to pre-GA features may not be compatible with oth
sions. For more information, see the launch stage descriptions (/products#product-launch-stages).

You can configure any disk defined in the instance template to be stateful (/compute/docs/instance-groups/stateful-migs) for all instances in a managed instance group (MIG) (/compute/docs/instance-groups#managed_instance_groups) by adding that disk's device name to the MIG's stateful policy (/compute/docs/instance-groups/stateful-migs#stateful_policy).

You can also configure stateful persistent disks individually for virtual machine (VM) instances in a MIG by setting per-instance configs (/compute/docs/instance-groups/stateful-migs#per_instance_configs); these disks don't need to be defined in the instance template.

Before you begin

- If you want to use the command-line examples in this guide:
    1. Install or update to the latest version of the gcloud command-line tool (/compute/docs/gcloud-compute).
    2. Set a default region and zone (/compute/docs/gcloud-compute#set_default_zone_and_region_in_your_local_client).
- If you want to use the API examples in this guide, set up API access (/compute/docs/api/prereqs).
- Review when to use stateful MIGs (/compute/docs/instance-groups/stateful-migs#when_to_use_stateful_migs) and how stateful MIGs work. (/compute/docs/instance-groups/how-stateful-migs-work)

Limitations

Stateful MIGs (/compute/docs/instance-groups/stateful-migs#what_makes_a_mig_stateful) have the following limitations:

- You cannot use autoscaling with stateful MIGs.
- You cannot use `proactive` (/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#starting_an_opportunistic_or_proactive_update) rolling updates if you configure stateful disks or stateful metadata.
    - You can control updates and limit disruption by updating specific instances (/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#updating_selected_instances) instead.
    - If you use custom instance names and don't configure stateful disks or metadata, you can use proactive updates, but, to preserve instance names, you must set the replacement method (/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#replacement_method) to `RECREATE`.
- When you permanently delete (/compute/docs/instance-groups/working-with-managed-instances#delete_from_group) an instance (either manually or by resizing), the MIG does not preserve the instance's stateful metadata.
- For stateful regional MIGs, you must disable proactive redistribution (/compute/docs/instance-groups/distributing-instances-with-regional-instance-groups#disabling_and_reenabling_proactive_instance_redistribution) (set the redistribution type to `NONE`) to prevent deletion of stateful instances by automatic cross-zone redistribution.

When to use stateful persistent disks

Use stateful persistent disks to take advantage of VM autohealing (/compute/docs/instance-groups/autohealing-instances-in-migs) and controlled updates (/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#updating_selected_instances)

while still preserving the data on the disks. For more information, see use cases (/compute/docs/instance-groups/stateful-migs#when_to_use_stateful_migs) for stateful MIGs.

When you configure stateful disks, these disks are preserved through VM instance autohealing, updates, and recreation. But that also means that stateful disks cannot be recreated from the original image or updated to a new image.

As a best practice, we recommend keeping your boot disks stateless.

Keeping the boot disk stateless has the following benefits:

- You can repair a boot disk that was corrupted by recreating it from its original image. Autohealing does such repairs automatically.
- You can update a boot disk to the latest image with new versions and security patches.

For more information, see how autohealing (/compute/docs/instance-groups/preserved-state#autohealing) and updating (/compute/docs/instance-groups/preserved-state#updating) handle preserved state.

Configuring stateful persistent disks for all VMs in a MIG

Configure any disk defined in an instance template to be stateful by adding that disk's device name to the MIG's stateful policy (/compute/docs/instance-groups/stateful-migs#how_stateful_migs_work). The MIG treats disks with that device name as stateful for all existing and future VM instances.

If an instance template attaches an existing disk in read-only mode, you don't need to configure it as stateful. T ready preserves this disk. It attaches the disk to all its VMs in read-only mode and detaches the disk when a VI d.

Use the `gcloud` tool or the Compute Engine API to:

- Configure stateful disks on MIG creation.
- Set or update a stateful configuration for disks in an existing MIG.
- Remove stateful disks from a MIG.

Configuring stateful disks on MIG creation

Permissions required for this task

To perform this task, you must have the following permissions (/iam/docs/overview#permissions):

- All permissions required to call the `instanceGroupManagers.insert` (/compute/docs/reference/rest/v1/instanceGroupManagers/insert) method (for zonal MIGs)
- All permissions required to call the `regionInstanceGroupManagers.insert` (/compute/docs/reference/rest/v1/regionInstanceGroupManagers/insert) method (for regional MIGs)

gcloud API (#api)

To specify which disks from an instance template should be stateful on MIG creation, use the `--stateful-disk` flag with the `gcloud beta compute instance-groups managed create` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/create):

```
gcloud beta compute instance-groups managed create INSTANCE_GROUP_NAME \
    --template INSTANCE_TEMPLATE \
    --size SIZE \
    --stateful-disk device-name=DEVICE_NAME[,auto-delete=DELETE_RULE]
```

Replace the following:

- `INSTANCE_GROUP_NAME`: The name of the managed instance group to create.
- `INSTANCE_TEMPLATE`: The name of the instance template to use when creating instances.
- `SIZE`: The initial number of VMs you need in this group.
- `DEVICE_NAME`: The device name of a disk specified in the instance template.
- `DELETE_RULE`: A value that prescribes what should happen to a stateful disk when a VM is deleted. Available options are:
  - `never`: (Default.) Never delete the disk; instead, detach the disk when its VM is deleted.
  - `on-permanent-instance-deletion`: Delete the disk when its VM instance is permanently deleted (/compute/docs/instance-groups/working-with-managed-instances#delete_from_group)

from the instance group, for example, when the managed instance is deleted manually or when the group size is decreased.

Regardless of the value of the delete rule, stateful disks are always preserved on VM autohealing, update, and recreation operations.

Note: You can see the device names of disks that are defined in an instance template by running the **`gcloud compute instance-templates describe`** command (/sdk/gcloud/reference/compute/instance-templates/describe).

Example

You want to deploy a database with 12 shards, each with a stateless boot disk that contains the operating system and database binaries, and with a stateful data disk. Use the following steps:

1. Create an instance template with a stateless boot disk based on the image **`img-example-db-v01`**, which has a pre-installed OS and database, and with a stateful data disk:

```
gcloud beta compute instance-templates create example-database-template-v
    --image img-example-db-v01 \
    --create-disk device-name=data-disk,mode=rw,image=empty10GBext4
```

   The **`--create-disk`** flag instructs the MIG to:

   a. Create a new 10 GB disk for each VM instance from an empty ext4 image, prepared beforehand.

   b. Attach the disk to its VM in read/write mode using device name **`data-disk`**.

2. Create a MIG from the instance template and define the data disk as stateful:

```
gcloud beta compute instance-groups managed create example-database-group
  --template example-database-template-v01 \
  --base-instance-name shard \
  --size 12 \
  --stateful-disk device-name=data-disk,auto-delete=on-permanent-instance
```

   The device name **`data-disk`** is taken from the instance template. The data disk is configured to be deleted together with the VM instance when the VM is permanently deleted (/compute/docs/instance-groups/working-with-managed-instances#delete_from_group) (either due to manual instance deletion or due to manual decrease of the group size). The data disk is preserved on autohealing, updates, and VM recreation.

3. Verify that the data disk is configured in the stateful policy:

```
gcloud beta compute instance-groups managed describe example-database-gro


baseInstanceName: shard
...
name: example-database-group
...
statefulPolicy:
  preservedState:
    disks:
      data-disk:
        autoDelete: ON_PERMANENT_INSTANCE_DELETION
...
```

You can see that the stateful policy declares disks with device name `data-disk` as stateful, with a rule to delete such disks on permanent VM deletion.

Setting and updating stateful configuration for disks in an existing MIG

If you run a stateful application on a stateless MIG (a MIG without any stateful configuration), you can configure existing disks that are defined in the instance template to be stateful for all instances in this MIG. This lets you preserve the disks on instance recreation, autohealing, and update operations, and optionally on deletion operations.

You can do the following operations:

- Add disks that are defined in the instance template to the stateful policy (/compute/docs/instance-groups/stateful-migs#how_stateful_migs_work) of an existing MIG to declare them as stateful. This marks disks with the given device name as stateful for all existing and future instances in the MIG.
- Update the stateful policy to change the stateful configuration for disks.

The MIG applies the updated configuration in the stateful policy automatically and asynchronously to all instances. Updates to disk configurations in a stateful policy do not disrupt running VMs. For more information, see applying stateful policy updates

(/compute/docs/instance-groups/how-stateful-migs-work#how_stateful_configuration_is_applied_to_managed_instances)
.

For a regional MIG, you must disable
(/compute/docs/instance-groups/distributing-instances-with-regional-instance-groups#disabling_and_reenabling_proactive_instance_redistribution)
proactive cross zone instance redistribution before you can configure stateful disks. For more information, see how regional groups handle preserved state
(/compute/docs/instance-groups/preserved-state#regional_migs).

If a MIG has a read-only disk attached to all of its VMs, you don't need to configure this disk as stateful. The M
y preserves this disk. It attaches the disk to all its instances in read-only mode and detaches the disk when a V
d.

Permissions required for this task

To perform this task, you must have the following permissions (/iam/docs/overview#permissions):

- All permissions required to call the `instanceGroupManagers.patch` (/compute/docs/reference/rest/v1/instanceGroupManagers/patch) method (for zonal MIGs)
- All permissions required to call the `regionInstanceGroupManagers.patch` (/compute/docs/reference/rest/v1/regionInstanceGroupManagers/patch) method (for regional MIGs)

gcloud API (#api)

To specify which disks from the instance template should be stateful or to update the stateful disk configuration for an existing MIG, use one or multiple `--stateful-disk` flags with the `gcloud beta compute instance-groups managed update` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/update):

```
gcloud beta compute instance-groups managed update NAME \
  --stateful-disk device-name=DEVICE_NAME[,auto-delete=DELETE_RULE]
```

Replace the following:

- ***NAME***: The name of the managed instance group to update.
- ***DEVICE_NAME***: The device name of a disk that is specified in the instance template.
- ***DELETE_RULE***: A value that prescribes what should happen to the stateful disk when a VM instance is deleted. The available options are:
  - `never`: (Default.) Never delete the disk, detach the disk when its instance is deleted.
  - `on-permanent-instance-deletion`: Delete the stateful disk when its instance is permanently deleted (/compute/docs/instance-groups/working-with-managed-instances#delete_from_group) from the instance group, for example, when the managed instance is deleted manually or when the group size is decreased.

★ **Note:** Regardless of the value of the delete rule, stateful disks are always preserved on instance autohealing, update, and recreation operations.

If a specified device name is already configured in the stateful policy, the command updates the configuration.

Note: You can see the device names of disks that are defined in an instance template by running the **`gcloud compute instance-templates describe`** command (/sdk/gcloud/reference/compute/instance-templates/describe).

Example

You run a database with multiple shards on a MIG named `example-database- group`. Each VM in the MIG stores a shard on an additional disk with device name `data-disk`, which is defined by the instance template. The MIG has no stateful configuration, and you want to preserve the data disks on instance recreation, autohealing, and updates. You also want to protect the data disks from deletion when a VM is deleted.

1. Update the MIG to define the data disk as stateful by using the following command:

```
gcloud beta compute instance-groups managed update example-database-group
  --stateful-disk device-name=data-disk,auto-delete=never
```

   As a result, the MIG applies the stateful policy configuration updates automatically and asynchronously to the data disks for all instances. The data disks are now preserved on autohealing, updates, and instance recreation, and the data disks are detached on instance deletion because the `auto-delete` rule is set to `never`.

2. Verify that the data disk is configured in the stateful policy by running the `gcloud beta compute instance-groups managed describe example-database-group` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/describe).

Declaring previously stateful persistent disks as stateless

You might need to configure a stateful disk to be treated as stateless. For example:

- If you rearchitect your app to move the state off the disk.
- If you configured the disk to be stateful by mistake and would like to revert it.

To declare all disks with a given device name as stateless, remove the disk's configuration from the stateful policy using the `gcloud` tool or the Compute Engine API.

If disk configuration for the same device name is present in a per-instance config, the disk remains stateful for ated instance even if you remove its configuration from the stateful policy. In this case, you must also remove configuration from the per-instance config to make the disk stateless.

The MIG applies the change to the stateful policy automatically and asynchronously to all instances. Updates to disk configuration in a stateful policy do not disrupt running VM instances.

For more information, see Applying stateful policy updates (/compute/docs/instance-groups/how-stateful-migs-work#how_stateful_configuration_is_applied_to_managed_instances).

Permissions required for this task

To perform this task, you must have the following permissions (/iam/docs/overview#permissions):

- All permissions required to call the `instanceGroupManagers.patch` (/compute/docs/reference/rest/v1/instanceGroupManagers/patch) method (for zonal MIGs)
- All permissions required to call the `regionInstanceGroupManagers.patch` (/compute/docs/reference/rest/v1/regionInstanceGroupManagers/patch) method (for regional MIGs)

gcloudAPI (#api)

To specify which disks from a MIG's stateful policy to make stateless, use the `--remove-stateful-disks` flag with the `gcloud beta compute instance-groups managed update` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/update):

```
gcloud beta compute instance-groups managed update NAME \
  --remove-stateful-disks DEVICE_NAME[,DEVICE_NAME,...]
```

Replace the following:

- ***NAME***: The name of the MIG to update.
- ***DEVICE_NAME***: The device name of a disk to remove from the stateful policy and to treat as stateless. You can provide one or multiple device names in the list.

Note: You can see the device names of disks that are configured in a MIG's stateful policy by running the **`gcloud beta compute instance-group managed describe`** command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/describe).

Example

You run a legacy application with multiple nodes on a MIG named `example-legacy-group`. Each VM in the MIG stores application data on a boot disk with device name `boot-disk`, which you configured as stateful in the MIG's stateful policy. You have moved application data to an additional disk and now want to make the boot disk stateless to make it easy to update to new images.

To remove the stateful configuration of the boot disk, update the managed instance group:

```
gcloud beta compute instance-groups managed update example-legacy-group \
  --remove-stateful-disks boot-disk
```

The MIG removes the stateful configuration for the device name `boot-disk` automatically and asynchronously for the boot disks of all instances in the group. The boot disks remain attached to the instances but are no longer stateful. When you recreate or update the instances, or when instances are autohealed, the MIG recreates the boot disks from the image specified in the instance template.

Removing stateful persistent disks from a MIG

You might need to completely remove a stateful disk from instances in a MIG, for example, if you re-architected your application and moved the state out of that disk.

MIGs do not allow removing stateful disks, so you must do the following steps:

1. Remove the stateful configuration (#declaring_previously_stateful_persistent_disks_as_stateless) of the disk from the stateful policy. This makes disks with the given device name stateless.
2. Detach the disks from the VMs if you still want to keep them.
3. Roll out a new instance template that no longer defines the disk with the given device name.

Configuring stateful persistent disks individually for a VM in a M

Configure stateful persistent disks for a specific VM in a MIG by adding the disk's device name to that VM's per-instance config (/compute/docs/instance-groups/stateful-migs#per_instance_configs). Update the VM to apply the per-instance config (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#applying_per-instance_configs) and make it effective.

Configuring stateful persistent disks individually for specific VMs in a MIG is useful if you need to:

- Migrate existing workloads (bring existing disks) from standalone VMs to stateful MIGs to benefit from autohealing and easy updates.
- Restore backups of disks, configured individually for VMs.
- Attach additional stateful disks to a specific VM temporarily for testing, debugging, or copying data.

p: Where feasible, consider configuring stateful disks for all VMs in a MIG figuring_stateful_persistent_disks_for_all_vms_in_a_mig), instead of configuring stateful disks individually for stance.

Adding existing stateful disks to new VMs in a MIG

You can add existing stateful disks to new instances that you manually create in a MIG. This is useful for migrating a stateful application from existing standalone VMs to a stateful MIG, for example:

1. Create an instance template (/compute/docs/instance-templates/create-instance-templates#based-on-existing-instance) with common configuration for all VM instances.
2. Detach (/sdk/gcloud/reference/compute/instances/detach-disk) the data disks from the standalone instances and delete these instances. You can also detach boot disks (/compute/docs/disks/detach-reattach-boot-disk) if they contain state that should be preserved.
3. Create (/compute/docs/instance-groups/creating-groups-of-managed-instances#create_managed_group) an empty MIG using the instance template created earlier.
4. Create instances (/compute/docs/instance-groups/working-with-managed-instances#adding_instances_with_specific_names) in the MIG with the appropriate names and associated disks from the previous step. The MIG responds to your request with the following actions:
    a. Creates a VM from the instance template using the provided instance name.
        - A regional MIG creates the VM in the same zone where the disk is located. If the disk is regional, the regional MIG creates the VM in any of the disk's replica zones.
    b. Creates a per-instance config (/compute/docs/instance-groups/stateful-migs#per_instance_configs) with the provided stateful configuration for the disks.
    c. Attaches the disks to the new instance.

Add existing stateful disks when manually creating specific instances in a MIG using the `gcloud` tool or API. The MIG applies the configuration immediately on VM creation.

gcloud API (#api)

To create a VM with a custom name and attach one or more existing stateful disks to that VM, use the `gcloud beta compute instance-groups managed create-instance` (/sdk/gcloud/reference/beta/compute/instance-groups/managed/create-instance) command with one or multiple `--stateful-disk` flags.

```
gcloud beta compute instance-groups managed create-instance NAME \
  --instance INSTANCE_NAME \
  [--zone ZONE | --region REGION] \
  --stateful-disk device-name=DEVICE_NAME,source=DISK[,mode=MODE][,auto-delete=DELET
```

Replace the following:

- `NAME`: The name of the MIG in which you need to create an instance.
- `INSTANCE_NAME`: The name of the new instance to create.
- `ZONE`: The zone where the MIG is located (applies to a zonal MIG).
- `REGION`: The region where the MIG is located (applies to a regional MIG).
- `DEVICE_NAME`: The device name to use when attaching the disk.
- `DISK`: The URI of an existing persistent disk to attach under the specified `DEVICE_NAME` in the format `projects/project-id/zones/zone/disks/disk-name` for a zonal disk and `projects/project-id/regions/region/disks/disk-name` for a regional disk.
- `MODE`: Specifies the mode of the disk. Supported options are:
  - `ro`: Read-only.
  - `rw`: (Default.) Read/write.
- `DELETE_RULE`: A value that prescribes what should happen to a stateful disk when a VM instance is deleted. The available options are:
  - `never`: (Default.) Never delete the disk; instead, detach the disk when its instance is deleted.
  - `on-permanent-instance-deletion`: Delete the stateful disk when its instance is permanently deleted (/compute/docs/instance-groups/working-with-managed-instances#delete_from_group) from the instance group, for example, when the instance is deleted manually or when the group size is decreased.

  Regardless of the value of the delete rule, stateful disks are always preserved on instance autohealing, update, and recreation operations.

Example

You want to have autohealing for a database server that is currently running on a standalone VM named `db-instance` and that currently stores data on a disk named `db-data-disk-1`.

Create a stateful MIG with autohealing, create a similar VM inside the MIG, and attach the existing data disk `db-data-disk-1` to the new instance as a stateful disk:

1. Stop the VM, `db-instance`, during a maintenance window.
2. Create an instance template (/compute/docs/instance-templates/create-instance-templates#based-on-existing-instance) named `db-template` using the `db-instance` configuration.
3. Detach `db-data-disk-1` from `db-instance` and delete `db-instance`.
4. Create an empty MIG, `example-database-mig`, from `db-template`, and configure autohealing.
5. Create a managed instance with the original `db-instance` name and attach the `db-data-disk-1` as a stateful disk:

```
gcloud beta compute instance-groups managed create-instance example-datab
  --instance db-instance \
  --zone us-east1-c \
  --stateful-disk device-name=data-disk,source=projects/example-project/z
```

The command creates an instance, `db-instance`, in the MIG, creates a corresponding per-instance config with `db-data-disk-1` stateful disk, and attaches the disk to the new VM, using `data-disk` as the device name.

Adding, declaring, and replacing stateful disks individually for VMs in a MIG

Configure stateful disks individually for a managed instance by adding or updating a stateful disk configuration in the associated per-instance config. Then update the instance to apply the per-instance config (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#applying_per-instance_configs) to the VM.

Configuring stateful disks individually is useful for the following tasks:

- Adding a stateful disk from outside of a MIG to a VM in that MIG.

- Declaring a previously stateless disk as stateful for a VM in a MIG.
- Replacing a stateful disk with a different disk for a VM in a MIG.

Adding a stateful disk from outside of a MIG to a VM in that MIG. You can attach any disk from outside of a MIG to a managed instance by adding stateful configuration for the disk to the associated per-instance config. After you apply the config, the MIG automatically attaches the disk to the instance and treats it as stateful.

Declaring a previously stateless persistent disk as stateful. You can declare a previously stateless disk, currently attached to a VM, as stateful by adding stateful configuration for this disk, including its device name and URI, to the associated per-instance config. After you apply the config, the MIG starts preserving the disk as stateful.

Replacing a stateful disk with a different disk. Replacing one stateful disk with another stateful disk can be useful, for example, if you need access to a recovered backup. You can swap one stateful disk for another by updating the disk's URI while keeping the same device name in the per-instance config. After you apply the updated per-instance config, the MIG detaches the old disk and attaches the new one using the same device name. When applying (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#applying_per-instance_configs)
the update, choose whether to keep the instance running, restart, or recreate it. Swapping a boot disk requires at least a VM restart.

gcloud API (#api)

To configure stateful disks individually for a VM in a MIG, add or update stateful disk configuration in the associated per-instance config and update the instance to apply the new configuration.

If a per-instance config does not yet exist for the instance, use the `gcloud beta compute instance-groups managed instance-configs create` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/instance-configs/create) with one or multiple `--stateful-disk` flags:

```
gcloud beta compute instance-groups managed instance-configs create NAME \
--instance INSTANCE_NAME \
--stateful-disk device-name=DEVICE_NAME[,source=DISK][,mode=MODE][,auto-delete=DELET
[--no-update-instance | --update-instance] \
[--instance-update-minimal-action MINIMAL_ACTION]
```

If a per-instance config already exists for the instance, use the `gcloud beta compute instance-groups managed instance-configs update` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/instance-configs/update) with one or multiple `--stateful-disk` flags.

The `--update-instance` flag (default) applies the changes immediately to the instance. If you use `--no-update-instance`, the changes remain unapplied and are applied (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#applying_per-instance_configs) when you next recreate or update the instance.

```
gcloud beta compute instance-groups managed instance-configs update NAME \
--instance INSTANCE_NAME \
--stateful-disk device-name=DEVICE_NAME[,source=DISK][,mode=MODE][,auto-delete=DELET
[--no-update-instance | --update-instance] \
[--instance-update-minimal-action MINIMAL_ACTION]
```

Replace the following:

- ***NAME***: The name of the managed instance group.
- ***INSTANCE_NAME***: The name of the VM instance for which to configure stateful disks.
- ***DEVICE_NAME***: The device name used for attaching the disk.
- ***DISK***: The URI of an existing persistent disk to attach under the specified ***DEVICE_NAME***, in the format `projects/`***`project-id`***`/zones/`***`zone`***`/disks/`***`disk-name`*** for a zonal disk and `projects/`***`project-id`***`/regions/`***`region`***`/disks/`***`disk-name`*** for a regional disk.

  The `source=`***`DISK`*** subflag is optional if the device is already defined in the instance's per-instance config. Otherwise it is required.

  If omitted, the currently configured disk URI remains unchanged.
- ***MODE***: Specifies the mode of the disk. You can only specify `mode` if you also specify `source`. Supported options are:
  - `ro`: Read-only.
  - `rw`: (Default.) Read/write.

  If omitted, the default value is set for a new stateful disk configuration; the value remains unchanged in an existing configuration.
- ***DELETE_RULE***: A value that prescribes what should happen to a stateful disk when a VM is deleted. The available options are as follows:

- `never`: (Default.) Never delete the disk; instead, detach the disk when its instance is deleted.
- `on-permanent-instance-deletion`: Delete the stateful disk when its instance is permanently deleted from the instance group, for example, when the instance is deleted manually or when the group size is decreased.

If omitted, the default value is set for a new stateful disk configuration; the value remains unchanged in an existing configuration.

Regardless of the value of the delete rule, stateful disks are always preserved on instance autohealing, update, and recreation operations.

- ***`MINIMAL_ACTION`***: Perform at least the specified action when applying the per-instance config update to the instance. Must be used together with the `--update-instance` flag. The value must be one of:
  - `none`: No action.
  - `refresh`: Apply updates that are possible to apply without stopping the VM.
  - `restart`: Stop the VM and then start it again.
  - `replace`: Recreate the VM.

  If omitted, the least disruptive action required by the update is used.

Example

The data on a currently attached stateful disk, `data-disk-1`, got corrupted, and you want restore it from the latest backup. You created a disk, `data-disk-2`, from a snapshot to replace the corrupted disk in instance, `db-instance-1`, managed by a stateful MIG, `example-database-mig`. The original dis,k `data-disk-1`, is attached under the `data-disk` device name with an auto-delete rule to never delete the disk.

To replace `data-disk-1` with `data-disk-2`, run the following command:

```
gcloud beta compute instance-groups managed instance-configs update example-database
  --instance db-instance-1 \
  --stateful-disk device-name=data-disk,source=projects/example-project/zones/us-eas
  --update-instance \
  --instance-update-minimal-action restart
```

The command does the following:

1. Updates the per-instance config for `db-instance-1`:

a. Updates the source for the disk with device name `data-disk` from `data-disk-1` (last configuration) to `data-disk-2` (new configuration).

b. Keeps the auto-delete rule to never delete the disk because the `auto-delete` parameter is omitted in the `--stateful-disk` flag and, by default, the delete rule is `never`.

2. Applies the per-instance config update to the `db-instance-1` VM immediately because the `--update-instance` flag is included. The MIG detaches `data-disk-1` and attaches `data-disk-2` under the same device name, `data-disk`.

3. Because the minimal action is set to `restart`, the MIG restarts the `db-instance-1` instance to update the VM, which helps the database application to start using the new disk.

Detaching a stateful disk or declaring it stateless for an individual VM

You might need to detach a stateful disk or configure it to be treated as stateless for an individual VM. For example:

- If you rearchitect your app to move the state off the disk.
- If you configure the disk to be stateful by mistake and would like to revert it.

Detach a stateful disk or make it stateless for an individual VM by removing the disk's stateful configuration from the associated per-instance config or deleting the entire per-instance config. When you apply (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#applying_per-instance_configs) the change:

- If the disk is not defined in the instance template, the MIG detaches the disk .
    - The MIG does not delete the disk when you delete its configuration from the per-instance config, regardless of the auto-delete rule in the configuration.
- If the disk is defined by the instance template, the MIG treats the disk as stateless, which means that the MIG recreates the disk from its source in the instance template on subsequent instance recreation, update, or autohealing events.

If you set a disk configuration for the same device name in the MIG's stateful policy, the disk remains stateful e move its configuration from the per-instance config. Remove the disk's configuration from both the stateful po r-instance config to make the disk stateless.

Removing a disk configuration from a per-instance config does not restart a running VM instance, unless you explicitly choose to do so.

For more information, see Applying per-instance configs updates (/compute/docs/instance-groups/how-stateful-migs-work#how_per-instance_config_updates_are_applied_to_instances).

gcloudAPI (#api)

To detach stateful disks or declare them stateless individually for a VM in a MIG, remove the stateful disk configuration from the associated per-instance config or delete the whole per-instance config (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#removing_stateful_configuration_for_a_specific_vm) if it doesn't contain any other state. Update the instance to apply the configuration.

To remove a stateful disk configuration from the associated per-instance config, use the `gcloud beta compute instance-groups managed instance-configs update` command (/sdk/gcloud/reference/beta/compute/instance-groups/managed/instance-configs/update) with the `--remove-stateful-disks` flag. The `--update-instance` flag (default) applies the changes immediately to the instance. If you use `--no-update-instance`, the changes remain unapplied and are applied (/compute/docs/instance-groups/applying-viewing-removing-stateful-config-in-migs#applying_per-instance_configs) when you next recreate or update the instance.

```
gcloud beta compute instance-groups managed instance-configs update NAME \
  --instance INSTANCE_NAME \
  --remove-stateful-disks DEVICE_NAME[,DEVICE_NAME,...] \
  [--no-update-instance | --update-instance] \
  [--instance-update-minimal-action MINIMAL_ACTION]
```

Replace the following:

- ***NAME***: The name of the MIG.
- ***INSTANCE_NAME***: Name of the VM from which to remove stateful configuration.
- ***DEVICE_NAME***: The device name used for attaching the disk.
- ***MINIMAL_ACTION***: Perform at least the specified action when updating the VM with its per-instance config. Can only be used together with `--update-instance`. The value must be one the following:

- `none`: No action.
- `refresh`: Apply updates that are possible to apply without stopping the VM.
- `restart`: Stop the VM and then start it again.
- `replace`: Recreate the VM.

If omitted, the least disruptive action required by the update is used.

Example

You run a legacy application on a MIG named `example-legacy-group`. Each VM in the MIG stores application data on a boot disk with device name, `boot-disk`. Using per-instance configs, you configured each boot disk to be stateful. Now you have moved application data to an additional disk, and you want to make the boot disk stateless for each VM to facilitate updating to new images.

For each instance, for example, for `node-1`, run the command:

```
gcloud beta compute instance-groups managed instance-configs update example-legacy-g
  --instance node-1 \
  --remove-stateful-disks boot-disk \
  --update-instance
```

The command does the following:

1. Removes configuration for the disk with device name `boot-disk` from the per-instance config for `node-1`.
2. Applies the per-instance config update to the `node-1` VM immediately because the `--update-instance` flag is included. The MIG removes the boot disk from the managed instance's `preservedStateFromConfig` (/compute/docs/instance-groups/how-stateful-migs-work#preserved_state_of_a_managed_instance) and treats the boot disk as stateless, which means that the MIG recreates the disk from its boot image in the instance template on subsequent instance recreation, update, or autohealing events.

Feedback

We want to learn about your use cases, challenges, and feedback about stateful MIGs. Please share your feedback with our team at mig-discuss@google.com (mailto:mig-discuss@google.com).

What's next

- Configure stateful metadata (/compute/docs/instance-groups/configuring-stateful-metadata-in-migs) for VMs in a MIG.
- Learn about applying, viewing, and removing (/compute/docs/instance-groups/configuring-stateful-metadata-in-migs) stateful configuration.
- Get info (/compute/docs/instance-groups/getting-info-about-migs) about a specific MIG and its managed instances, including VM status and properties.
- Learn more about about MIGs (/compute/docs/instance-groups#managed_instance_groups) and working with managed instances (/compute/docs/instance-groups/working-with-managed-instances).

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the Apache 2.0 License (https://www.apache.org/licenses/LICENSE-2.0). For details, see the Google Developers Site Policies (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-07-30 UTC.