

Performing one-click OS image upgrades in MIGs

By using a combination of custom [image families](#)

([/compute/docs/images/create-delete-deprecate-private-images#setting_families](#)) and [rolling updates](#) ([/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups](#)), you can enable one-click OS image upgrades on your [managed instance group](#) ([/compute/docs/instance-groups#managed_instance_groups](#)) (MIG).

Using the one-click OS image upgrade provides a number of benefits, including:

- Works with all VM machine types and all instance group sizes.
- Supports both Windows and Linux images and containers.
- Instances are recreated based on their current instance template, or optionally, based on a new template, so you can preserve custom startup scripts and metadata.
- Works with [stateful MIGs](#) ([/compute/docs/instance-groups/stateful-migs](#)), so you can optionally preserve data on non-boot disks.
- The rollout of an update to the new OS version happens automatically, without the need for additional user input after the initial request.
- Supports batch updates with an optional health check.

How does one-click OS image upgrade work?

When you invoke an update, the MIG replaces the boot disks for all VMs in the group with the latest available OS image version from your custom image family. The MIG preserves metadata and startup scripts that you set up in the instance template for each VM in the group. Non-boot disks are recreated based on their specification in the instance template. For information about preserving data, see [Configuring stateful disks in MIGs](#)

([/compute/docs/instance-groups/configuring-stateful-disks-in-migs](#)).

To limit application disruption, you can perform updates in batches, keeping a specific percent of VMs running during the update. To increase reliability, you can configure an application-

based health check for your MIG: the group waits for a healthy response from an application on updated VMs before proceeding with further updates to other VMs.

Container-Optimized OS also supports one-click [auto-updates](#) (container-optimized-os/docs/concepts/release-channels#auto_updates).

Before you begin

- Install or update to the latest version of the [gcloud command-line tool](#) (/compute/docs/gcloud-compute).
- Make sure you have [created an instance template](#) (/compute/docs/instance-templates/create-instance-templates#creating_a_new_instance_template) that points to an image family. Google recommends that you use [custom image families](#) (/compute/docs/images#custom-families) to reduce the risk of rolling out an image version that is incompatible with your application. You can ensure that only compatible image versions are rolled out by adding images to your custom image family only after compatibility testing with your application.

When your instance template points to an image family, the MIG always creates instances from the latest image in the family, for example:

- When the MIG adds new instances because you or the MIG's autoscaler increased the MIG's size.
- When the MIG recreates an instance, triggered manually or by autohealing.

★ **Note:** if you don't want the MIG to create instances from the latest image in an image family, create an instance template that points to a specific image. When you want to use a newer version of the image, create a new instance template that specifies the new image, then update the MIG with the new template by using a [rolling update](#) (/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#starting_an_opportunistic_or_proactive_update)

- Test the new image with your app before adding it to your image family and rolling it out.

- Optionally, create an application-based health check (</compute/docs/instance-groups/autohealing-instances-in-migs>) for your MIG. An application-based health check verifies that your application is responding as expected on each of the VMs in the MIG. You can configure your update to allow no more than one unavailable VM. If an application does not respond as expected, then the MIG marks that VM as unavailable, and your rollout does not proceed.

Performing one-click OS image upgrades for MIGs

To update all the VMs in a MIG to the latest image from a custom image family, complete the following steps:

1. Start a rolling replace

(/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#performing_a_rolling_replace_or_restart)

with the following command.

```
gcloud compute instance-groups managed rolling-action replace instance-group-name
  [--max-surge=max-surge] [--max-unavailable=max-unavailable]
```

Replace the following:

- ***instance-group-name***: the name of the MIG to operate on.
- ***max-surge***: the maximum additional number of VMs that can be temporarily created during the update process. This can be a fixed number (for example, 5) or a percentage of the size of the MIG (for example, 10%).
- ***max-unavailable***: the maximum number of VMs that can be unavailable during the update process. This can be a fixed number (5) or a percentage of the size of the MIG (10%).

You can combine health checks by using the `--max-unavailable` and `--max-surge` options

(</compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#options>) to stop further updates if they cause VMs to become unavailable.

2. Monitor the update

(/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#monitorupdates)

by using the [wait-until](/sdk/gcloud/reference/compute/instance-groups/managed/wait-until) subcommand to check that the MIG's `status.versionTarget.isReached` field is set to `true`.

```
gcloud compute instance-groups managed wait-until instance-group-name --version
```

Replace the following:

- *instance-group-name*: the name of the MIG to operate on.

The command returns when the group is updated.

You can also [list instances](/sdk/gcloud/reference/compute/instance-groups/list-instances) to see each instance's status.

```
gcloud compute instance-groups managed list-instances instance-group-name
```

The command returns a list of instances and their details, including [status](/compute/docs/instances/instance-life-cycle)

(/compute/docs/instances/instance-life-cycle), [health state](/compute/docs/instance-groups/autohealing-instances-in-migs#health_states)

(/compute/docs/instance-groups/autohealing-instances-in-migs#health_states), and [current actions](/compute/docs/instance-groups/autohealing-instances-in-migs#current_actions)

(/compute/docs/instance-groups/autohealing-instances-in-migs#current_actions_on_instances) for each VM. When all VMs are `RUNNING` and have no current action, then the MIG is up-to-date and [stable](/compute/docs/instance-groups/getting-info-about-migs#verify_group).

3. In case you need to roll back to a previous OS image, you must create an instance template and specify the image you want to use. Then [start a rolling update](/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#starting_a_basic_rolling_update)

(/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#starting_a_basic_rolling_update)

to update all managed instances to use that template. For more information, see [Rolling back an update](#)

(/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#rolling_back_an_update)

Example

This example covers the following tasks:

1. Create an instance template for easy OS image updates:
 - Specify a custom image family (/compute/docs/images#image_families) in the instance template.
2. Create a MIG based on the template.
3. Set up a health check to limit disruption by an image update.
4. Add a new image to an image family.
5. Invoke an OS update with a single command.
6. Monitor the update.

Use the following steps to enable and perform one-click OS upgrades on a MIG:

1. Create an instance template that specifies a custom image family. The image family should contain tested and trusted images. Each VM that the MIG creates from the template uses the latest available image from this family.

```
gcloud compute instance-templates create example-template \  
  --machine-type n1-standard-4 \  
  --image-family my-image-family \  
  --image-project my-project \  
  --tags=http-server
```

★ **Note:** If you don't want the MIG to create instances from the latest image in an image family, create an instance template that points to a specific image. When you want to use a newer version of the image, you must create a new instance template that specifies the new image, then update the MIG with the new template by using a rolling update.

- A rolling update (</compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups>) updates the VM based on a latest version of the instance template that you set for the MIG.

- A [rolling replace](#)
([/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups?tenant=cloud#performing_a_rolling_replace_or_restart](#))
replaces the VM and re-uses the managed instance's current instance template.

2. Create a MIG based on the instance template. This example starts the MIG with three instances based on `example-template`. Because the instance template specifies an image family, the MIG creates each VM with the latest image from the family.

```
gcloud compute instance-groups managed create example-group \  
  --base-instance-name example \  
  --size 3 \  
  --zone us-east1-b \  
  --template example-template
```

3. Optional: Configure and enable an application-based health check. If your app doesn't respond after an image update, you can use the health check status combined with the [maxUnavailable](#) ([/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#max_unavailable](#)) setting to stop the MIG from further rollouts.

- a. Create a health check that looks for an HTTP **200** response on the request path `/health`. The GitHub app that is on each instance serves that path.

```
gcloud compute health-checks create http example-autohealer-check \  
  --check-interval 10 \  
  --timeout 5 \  
  --healthy-threshold 2 \  
  --unhealthy-threshold 3 \  
  --request-path "/health"
```

- b. Create a firewall rule to allow the health checker probes to access the instances. The health checker probes come from addresses in the ranges: `130.211.0.0/22` and `35.191.0.0/16`

```
gcloud compute firewall-rules create default-allow-http-health-check \  
  --network default \  
  --allow tcp:80 \  
  --source-ranges 130.211.0.0/22,35.191.0.0/16
```

c. Add the health check to your MIG.

```
gcloud compute instance-groups managed update example-group \  
  --zone us-east1-b --health-check example-autohealer-check
```

4. When an update is available, tested, and determined to be compatible with your app, create a new image, and use the `--family` flag to include that image in the custom image family.

```
gcloud compute images my-image-v2 \  
  --source-disk boot-disk-1 \  
  --source-disk-zone us-central1-f \  
  --family my-image-family
```

In this example, the latest image in `my-image-family` is now `my-image-v2`, which is based on the source disk `boot-disk-1`.

5. Invoke a rolling replace to replace all VMs in the MIG. The MIG replaces each VM based on the group's instance template. The instance template specifies `my-image-family`, so each VM gets the latest image in the family (`my-image-v2`).

You can [configure](#)

([/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#options](#)) the level of disruption that the update causes. In this example, the MIG creates one additional VM above the group's target size, and it does not remove any existing VMs until that one VM is up and running.

```
gcloud compute instance-groups managed rolling-action replace example-group \  
  --max-surge 1 --max-unavailable 0
```

6. If you want to monitor the status of the updates, use the [wait-until command](#) (`/sdk/gcloud/reference/compute/instance-groups/managed/wait-until`) with the `--version-target-reached` flag. The command returns when the group is updated.

```
gcloud compute instance-groups managed wait-until --version-target-reached exam
  --zone us-east1-
Waiting for group to reach version target
...
Version target is reached
```

You can also use the [list-instances command](#) (`/sdk/gcloud/reference/compute/instance-groups/list-instances`) to see the [status](#) (`/compute/docs/instances/instance-life-cycle`), [health state](#) (`/compute/docs/instance-groups/autohealing-instances-in-migs#health_states`), [current actions](#) (`/compute/docs/instance-groups/autohealing-instances-in-migs#current_actions_on_instances`), instance template, and version for each VM.

```
gcloud compute instance-groups managed list-instances example-group \
  --zone us-east1-b
```

NAME	ZONE	STATUS	HEALTH_STATE	ACTION	INSTANCE_TEMPLATE	VER
test-211p	us-east1-b	RUNNING	HEALTHY	NONE	example-template	0/2
test-t5qb	us-east1-b	RUNNING	UNKNOWN	VERIFYING	example-template	0/2
test-x331	us-east1-b	RUNNING	HEALTHY	NONE	example-template	0/2

7. If you need to [rollback](#)

(`/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#rolling_back_an_update`)

to a previous image, use the following steps:

- a. [Create a new instance template](#)

(`/compute/docs/instance-templates/create-instance-templates#creating_a_new_instance_template`)

that specifies the image that you want.

- b. [Start a rolling update](#)

(`/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups#starting_an_opportunistic_or_proactive_update`)

to apply the instance template.

What's next

- Learn more about [image family best practices](/compute/docs/images/image-families-best-practices) (/compute/docs/images/image-families-best-practices).
- Learn more about [instance groups](/compute/docs/instance-groups) (/compute/docs/instance-groups).
- Learn more about [instance templates](/compute/docs/instance-templates) (/compute/docs/instance-templates).
- Learn more about [rolling updates](/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups) (/compute/docs/instance-groups/rolling-out-updates-to-managed-instance-groups) in MIGs.

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/) (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) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/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.