Custom Images Alpha: Documentation

Welcome to the Alpha for Custom Images 🎉

We're excited for you to test out and provide feedback as we iterate on the feature.

Customer guidelines for using the feature

Prerequisites

1. The following have been onboarded onto the feature properly:
   - Your enterprise or organization, depending on which entity will be creating a runner to be used for image generation. (This decision could also be made by where you manage your runners today)
2. You need to create an image-building GitHub-hosted runner under the same organization or enterprise. This will be used to build the image so a couple of important settings need to be configured:
   - The platform of the runner can be `x64`, or linux `ARM64`. Please also ensure that the chosen platform matches the platform of the desired image. I.e. If you're building a Linux image, you must create a Linux runner. If you're creating a Windows image, you'll need a Windows runner.
   - Your runner needs to have a base image to build on, there are a few options:

     1. A GitHub-owned Linux image (x64 only)
        - `Ubuntu 24.04 (beta)`
        - `Ubuntu 22.04`
        - `Ubuntu 20.04`

     2. An empty base image. These are selected from the `Partner` tab of the images menu.
        - `Base Image for Ubuntu 22.04`(x64 or arm64)
        - `Base Image for Ubuntu 20.04`(x64 only)
        - `Base Image for Windows Server 2022`(x64 only)
        - `Base Image for Windows Server 2019`(x64 only)

     3. An Arm-maintained linux image, from the `Partner` tab, that has pre-installed tooling. (arm64 only)
        - `Ubuntu 22.04 by Arm Limited`

        - `Ubuntu 24.04 by Arm Limited`
    - Check the box that says `Enable this runner to generate custom images`
    - Sizing: You can select a size depending on the size of your image. You can see how much storage each runner size has when selecting the underlying hardware. Keep in mind that the size will affect the billing for workflows run on this runner.
3. Once you have created your image-building runner, assign it the appropriate permission within your runner group so that it can be used by the repository you wish to use for image generation.

Writing your workflow file

You can either write a brand new workflow YAML file, or use one of your existing ones. Any job can be configured to generate an image and you can run multiple image generation jobs in the same workflow file. Keep the following parameters in mind while setting up your workflow file:

- Every job that includes the snapshot keyword will create a separate image, so if you only want to create a single image or image version, you must put all of your workflow steps into one job.
- Every time you run the job that includes the snapshot keyword, a new image version will be generated.

You just need to do the following for each job:

1. Set the `runs-on` value to the name of your GitHub-hosted image generation runner you created earlier.
2. Add the new `snapshot` keyword to the job and specify the desired image name you want to create or update. This can be added in a couple of different ways, shown below.

String syntax

The most simple way of adding the `snapshot` keyword is to just provide a string, representing the desired image name.

Unset

```
jobs:
  build:
    runs-on: my-image-generation-runner
    snapshot: MyCustomImage
    steps:
      # Add any steps to download and setup any dependencies here
```

Mapping syntax

Alternatively, you can provide a mapping with the desired image name specified as the `image-name` key. This is currently the only valid key for the mapping and might seem redundant now, but in the future we're planning on adding additional options which you can specify when creating a custom image.

Unset

```
jobs:
  build:
    runs-on: my-image-generation-runner
    snapshot:
        image-name: MyCustomImage
    steps:
      # Add any steps to download and setup any dependencies here
```

Note: For both methods of adding the new `snapshot` keyword, you may see errors indicating that it's invalid. Ignore these. This is the expected behavior during the alpha as the intellisense hasn't been updated yet. This will not prevent you from saving the file or queuing a job run though.

Running your workflow

After your image generation workflow job runs, the generation itself will take place in the background.

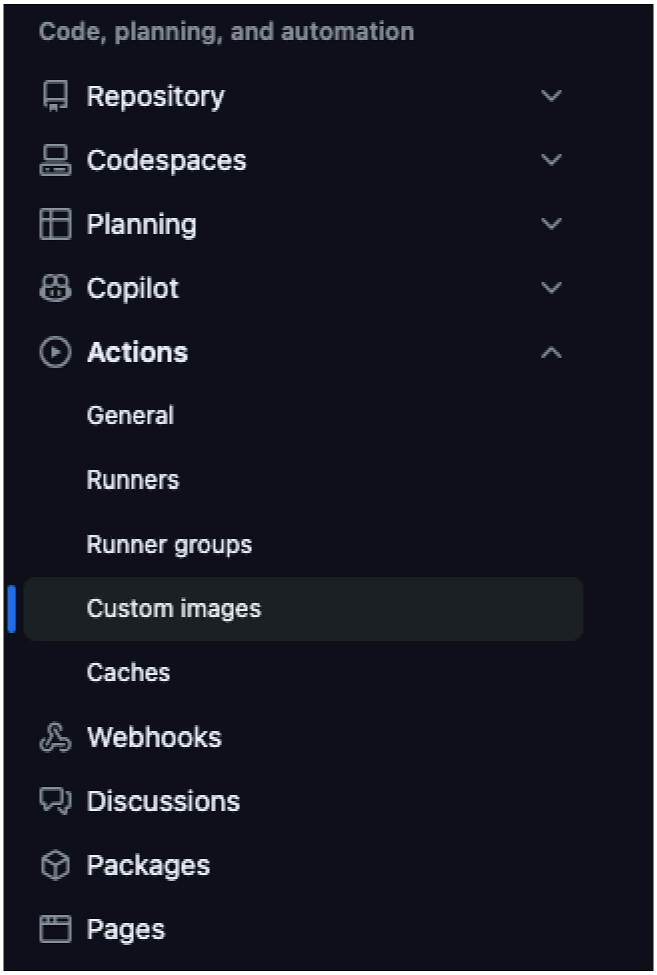
If an image doesn't already exist within your organization/enterprise with the specified name, it'll be created with an initial version number of `1.0.0`. Otherwise, a new version will be created for the existing image with an incremented minor version number (i.e. `1.1.0`, `1.2.0`, etc.).

Please note that an image will only be generated if your job finished successfully. This is to avoid new image versions being created in an undesired state.

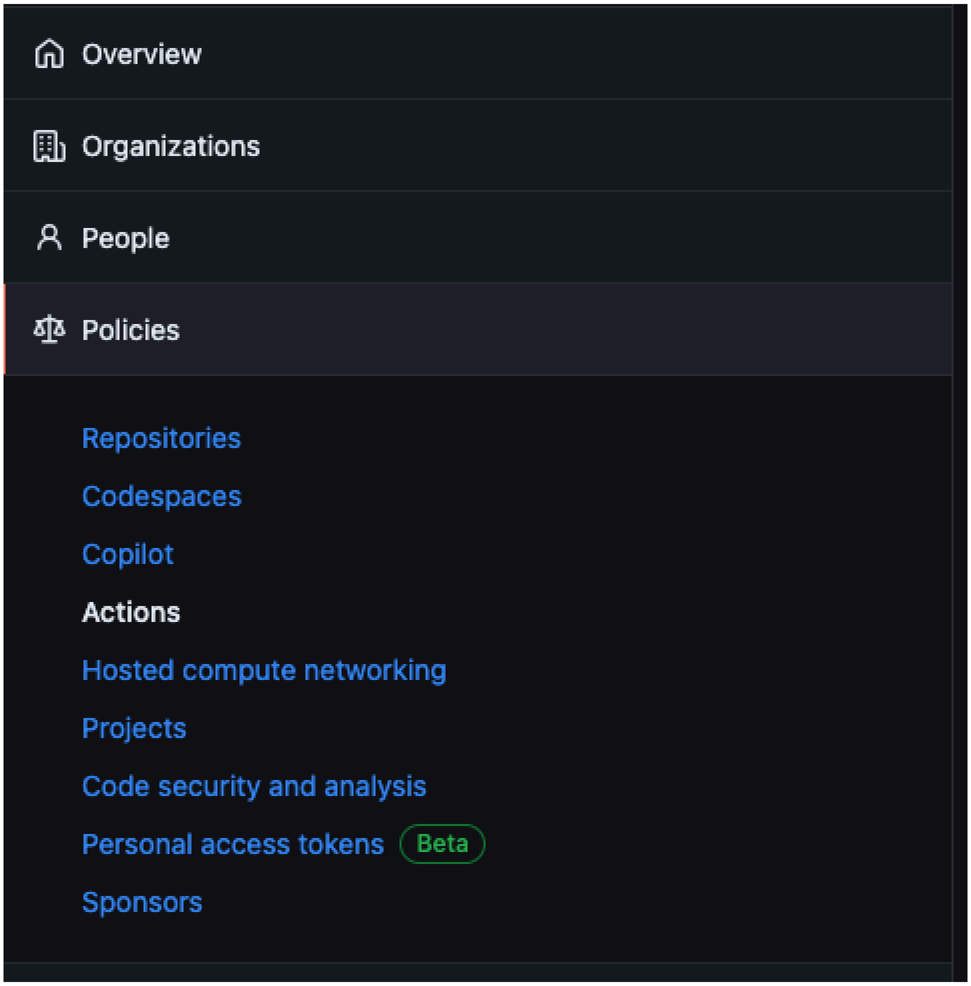
After your job runs

Checking image generation progress

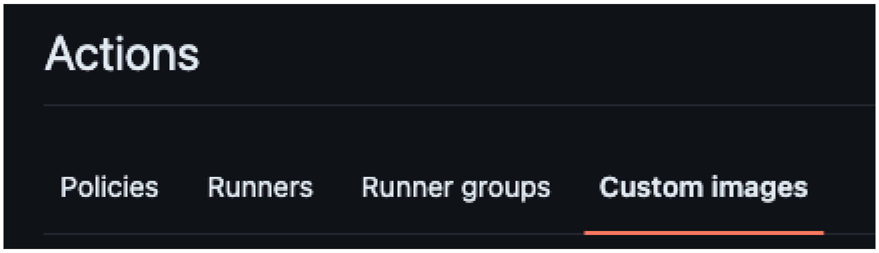
1. Go to the new `Custom images` page. It can be found in the following places:
    - For an organization - `Settings -> Actions -> Custom images`:



    - For an enterprise: `Policies -> Actions`:



- `Custom images` tab:



2. Use the search bar or scroll down the page to find an image with the name you specified in your workflow job. If you don't see the expected image, try refreshing the page after waiting a couple of minutes. If you do see it, you should be able to click it to view the status of the image version you're generating.

Image Retention:

During the alpha we have set a retention policy that will only store the 30 latest image versions. This is 30 per image definition, so each image definition can have up to 30 versions.

Note: We're working on improving performance but currently it takes some time for an image version to be fully generated and ready to be used:

- Ubuntu: ~20 minutes
- Windows Server: ~50 minutes

If you're updating an existing image and it's being used by runners which are configured to use the latest version, this is also how long it'll take for them to be automatically updated (more on this below).

Using your custom image

1. Go back to the `Create GitHub-hosted runner` page for your organization or enterprise (whichever one the image generation runner belongs to).
2. Under `Platform`, select the same one as the runner you used for generating the image.
3. Under the `Image` menu, select the `Custom` tab. You should see your image appear in this list. If you don't, please ensure you've selected the correct platform. Please also ensure you're definitely creating a runner for the owner of the custom image. Enterprise images can only be used by enterprise runners and organization images can only be used by organization runners.
4. After selecting your new custom image from the list, you can also select the `Version` via the dropdown underneath. The version selection works like this:
   - If you select `latest`, your runner will be created with the most recent image version and will automatically update when new versions of your custom image are created. If you decide to set your runner to a specific version at a later point, you can do that by editing your runner.
   - If you select a specific version number, your runner will be created with that specific version number and won't automatically be updated to another version. If you want to change the version, you'll need to manually edit the runner.
5. Complete the rest of the form for creating a GitHub-hosted runner.
6. Share this runner with the correct repositories via your runner group
7. Specify this new runner's name in the `runs-on` of a GitHub Actions workflow job which uses the dependencies installed in your custom image. This will be your image-consuming workflow.
8. Your job should run successfully as all the prerequisites should be available from the image.

If something goes wrong

This feature is still in its infancy and users can't currently troubleshoot issues due to a lack of information provided in the image generation workflow logs.

If you experience problems using this feature, please reach out to @lkfortuna or your account team.

Billing For Custom Images

Custom Image workflow runs are billed on a per-minute basis. These workflow runs do not consume entitlement minutes and are entirely paid. During the alpha these runners will follow the same payment model as our larger hosted runners, and will not be free in public repos.

Depending on the size of runner you choose for your image-building runner, those are the rates the image generation workflow will be charged at. For more information on the rates see our public documentation.

FAQs

What do we mean by "images"?
Images are virtual hard disks(VHDs) that we install on our hosted runners. These are not docker container images. GitHub hosted runners are virtual machines(VMs), not containers. When we refer to images we are talking about VM images.

Which repositories can create images? How do I turn this on?

Permissions for creating custom images are managed at the organization or enterprise level, with the ability to scope the custom image building runners to specific organizations or repositories using runner groups.

Who can create images and who can overwrite them?

You must have write access to a repository to be able to run a workflow that creates an image. Anyone who can run the workflow has the ability to create a new version of the image.

How do I set up images to be shared between different teams?

Access to the runners that have a custom image installed can be controlled through runner groups and their permission model. Runner groups are managed at the org level or the enterprise level. To read more about these permissions see our documentation here.

How do I manage my images?
As part of your Actions' management UI you can now see a list of the images in your organization and the versions for each image. This is also where you can assess the size of the image. You can also delete entire images or versions of those images.

How do I roll back an image?
To roll back an image, you can go into the management UI and delete the version of the image you want to roll back, or you can edit the runners using the image and select the last version before the broken image was introduced.

How do I pin to a specific version of an image?
To pin to a specific version, you can create or edit your runner to run on a specific version of your image that you select in the drop down menu.

How do I change the image on the runner?
By default, the latest version of an image will be associated with a runner using that image. You can change which image or image version is associated with a runner by editing the runner and selecting a different image or version.

How do setup-* actions work on custom images?
If you want to use these popular Actions to install tools on your custom image you will need to do the following:

1. Add the setup action step in your image building workflow. This will install the software on the os disk. However, the Action stores the version of the software in memory, so this will not carry over to the image.
2. In the workflow running on the image, run the setup action step again specifying the version you want to test against. This will set the version in memory on the runner and should take 0-1s.

The setup-* actions that are managed by github include:
Setup-java
Setup-dotnet
Setup-python
Setup-go
Setup-node
For more information on these actions see the Actions marketplace.

What is the best way to debug an image generation workflow run?
Comment out the `snapshot` keyword in your workflow file while debugging runs on the image generation runner. This will prevent you from creating many extra image versions. When you

have gotten your workflow to run to your liking, un-comment the snapshot keyword to begin generating image versions again.