Slimming down the container image size #5007

youtalk started this conversation in General

youtalk on Jul 18 Collaborator edited

Code of Conduct

- [x] I have read CODE OF CONDUCT and Support Guidelines before creating this Discussion post.

Contents

As a result of introducing cache into the CI workflows of the `autoware` and `autoware.universe`, the execution speed has increased several times. However, as a side effect, GitHub-Hosted Runner frequently fails to execute jobs due to the `No space left on device` issues for example: https://github.com/autowarefoundation/autoware/actions/runs/9981995778/job/27586843983#step:5:18544.

Therefore, slimming down the container image sizes is the next urgent task.

Several countermeasures can be considered.

1. ~~Refine `.dockerignore` to minimize the amount of files being copied.~~
2. ~~Stop copying `src` and temporarily mount `src` with `RUN --mount=type=bind` : https://github.com/moby/buildkit/blob/master/frontend/dockerfile/docs/reference.md?utm_source=pocket_reader#run---mounttypebind~~
3. Minimize the dependent packages in each package's `package.xml` which is already mentioned by **@oguzkaganozt** **More Strict Dependency Control for ROS Nodes** #4599.
4. Install CUDA-related drivers in the another base image which I mentioned in https://github.com/orgs/autowarefoundation/discussions/4995.
5. ~~Stop inserting artifact files into the image but there is also opposing opinion on this **feat(docker): fix CUDA compile on devel image and improve run.sh** #4849 (comment).~~

I'm also definitely looking for ideas from you all.

↑ 1

Category

General

Labels

type:github-actions
type:containers
component:openad...

5 participants

3 comments · 16 replies

Oldest Newest Top

youtalk on Jul 18 Collaborator Author

First, I will address the number 1 which I believe no one will object to.

↑ 1 6 replies

Show 1 previous reply

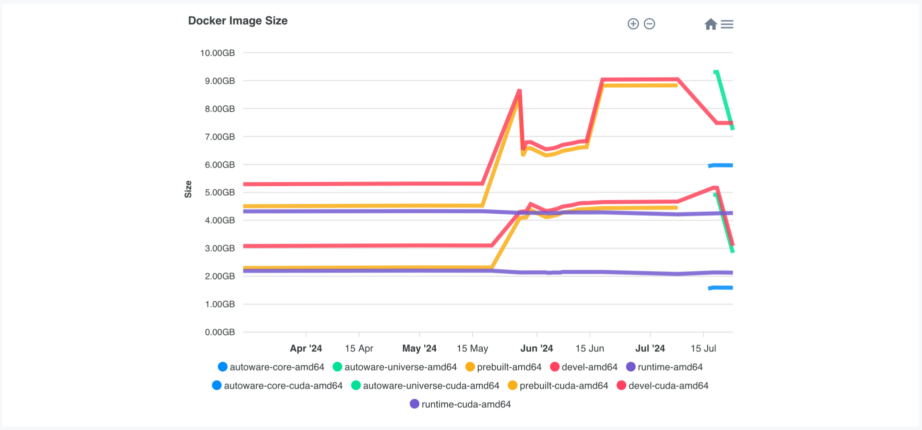
xmfcx on Jul 23 Maintainer

@youtalk -san, for CI to work, we don't need any parts of the Autoware folder, including `src`. All of our actions start with `checkout` action and downloading necessary things.

So for the CI image, we can completely remove Autoware folder.

youtalk on Jul 23 Collaborator Author

What I mentioned is for the devel image. There is no src in the runtime since before.



xmfcx on Jul 23 Maintainer

In CI, we use `latest-autoware-universe` tag right now.

Here is the image:
https://github.com/autowarefoundation/autoware/blob/57516233b0c3ab98c91b4c2a5f92cb53d0f09643/docker/Dockerfile#L113C23-L113C40

Does `autoware-universe` image have src folder in it?

Since we always have the size problem with this CI image, we can remove src from it. Because we don't use it in CI.

youtalk on Jul 23 Collaborator Author edited

We can remove the `src` from the `autoware-universe` image for the CI but we would like to remain the `src` in the `devel` image for the dev container.
I try to make it possible.

👍 1

youtalk on Jul 23 Collaborator Author

@xmfcx I made a PR #5027

xmfcx on Jul 18 Maintainer

> **Stop inserting artifact files into the image** but there is also opposing opinion on this #4849 (comment).
>
> > @oguzkaganozt :
> > I think we should keep our images as complete as possible. Since we don't have any disk usage problem anymore, we can re-add those artifacts to simplify container usage. Because the original intention was to save space.
> >
> > And also think about a scenario in which user just downloads the image and go on a field test where there is no sufficient internet connection to download all artifacts. IMHO, devel and runtime images should be self-sufficient. Moreover these artifacts are mandatory to run essential parts of the Autoware(perception, localization) that means user will always need to download them so why not just keep them inside the image ?

From @VRichardJP :

- Lightweight devel build for docker #4991

> I really liked this option, because it both saved a lot of time (1 hour compilation...) and a lot of space on my system.
> On top of that, I have no use for the source code or built artifacts in /autoware since I share my own "autoware" directory with the container (built artifacts are on host).

Considering these, I would propose to have an image without artifacts for development and CI purposes.
Version with artifacts can be something that derives from this light image.

And in the CI, we can download the required artifacts when necessary too. I think CI doesn't even need the artifacts to function.

Related discussion:

- Proposal to change Docker image tag naming conventions #4995

↑ 2 👍 1 10 replies

Show 5 previous replies

oguzkaganozt on Jul 29 Maintainer

Yes, then it would be only meant to be used as devcontainer as the name suggests ?

👍 1

youtalk on Jul 30 Collaborator Author

When I use the `devel` container, I only volume mount partial source code. So, if the `build` and `install` directories are completely deleted, it causes problems for me.

xmfcx on Jul 30 Maintainer edited

> When I use the `devel` container, I only volume mount partial source code.

@youtalk -san, the build artifacts in `build` and `install` are generally not dependable.

We recommend users to mount the Autoware folder as a whole and manage the `build` and `install` folders outside the container.

`run.sh` also mounts the workspace as a whole.

The reason for this is, sometimes Autoware packages rename, `CMakeLists` files change, some unwanted artifacts remain in the `build` and `install` folders. And it causes problems when using `colcon build`. This is why we rely on `ccache` instead of caching `build` / `install` folders.

This is why I would recommend removing Autoware folder completely from the `devel` image.
Or maybe have 2 variants. But I don't see the point for variant with `build` and `install` folders.

👍 2

youtalk on Aug 13 Collaborator Author

@xmfcx I wrote my opinion here #5045 (review).

doganulus on Aug 17 Collaborator

@youtalk **@xmfcx** As I said, I am not able to use official Autoware images because of their unnecessary complexity, so I am building my Autoware containers here. I had explained their rough idea elsewhere, but some caveats might be helpful for you.

- The `latest-builder` image contains all build dependencies and is ready to build Autoware upon mounting the source. No source is buried in them. They are good as long as the build dependencies are not changed.
- The `latest-builder-with-cache` image has an extra layer of the `ccache` artifacts on top of `builder` (currently 215MB). Distributing another image is much easier than making caches

dance around. I didn't see any disadvantages so far. Besides, this works independently of GitHub Actions, including local builds.

- These images can be used in the CI. Here is an example use. For example, this workflow builds up to `autoware_launch` package in 16 minutes with fresh `ccache` artifacts (hit rate: 99%+). If the cache gets stale, the performance reduces as expected. For example, build times reach 55 minutes with a one-week-old cache (hit rate: 90%+). Therefore, builder images are better built daily rather than weekly, given Autoware's cadence.
- Builder containers must set `CCACHE_BASEDIR` to allow caching to work under different absolute paths. This is the case when the user mounts its source code. Still, users' directory structure must match the structure at cache-build time. This is a limitation of `ccache`.
- The tag `devel` is for developers, which extends builder images. Here is the ideal place to add more developer tools and libraries, including visualization tools. I haven't tested user experience extensively, but **@VRichardJP** may have other suggestions. I would appreciate this.

xmfcx on Aug 16 Maintainer

Related:

- **Remove unnecessary CUDA libraries from the runtime image** #5083

↑ 1

0 replies