autowarefoundation / **autoware** Public

Code · Issues 42 · **Pull requests 13** · Discussions · Actions · Security · ···

feat(docker): refine devel image #5045

New issue

Closed **oguzkaganozt** wants to merge 5 commits into `main` from `feat/refine-devel-image`

Conversation 22 · Commits 5 · Checks 15 · Files changed

oguzkaganozt commented on Jul 29 • edited ▾ Contributor

Description

Related issues and discussions:
#4991
https://github.com/orgs/autowarefoundation/discussions/5007#discussioncomment-10110181
https://github.com/orgs/autowarefoundation/discussions/4995#discussioncomment-10086502

In order to make `devel` image more generic and lightweight, I omitted the source code and prebuilt binaries from the `devel` stage. And for preserving the multi-stage build as much as possible I had to introduce intermediary `autoware-core-prebuilt` stage.

Tests performed

Not applicable.

Effects on system behavior

Not applicable.

Interface changes

Pre-review checklist for the PR author

The PR author **must** check the checkboxes below when creating the PR.

- ☑ I've confirmed the contribution guidelines.
- ☑ The PR follows the pull request guidelines.

Reviewers

youtalk
xmfcx ●
VRichardJP ●
mitsudome-r ●

Assignees

oguzkaganozt

Labels

tag:run-health-check

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

Lightweight devel build for docker

4 participants

In-review checklist for the PR reviewers

The PR reviewers **must** check the checkboxes below before approval.

- □ The PR follows the pull request guidelines.

Post-review checklist for the PR author

The PR author **must** check the checkboxes below before merging.

- □ There are no open discussions or they are tracked via tickets.

After all checkboxes are checked, anyone who has write access can merge the PR.

oguzkaganozt requested a review from **xmfcx** 4 months ago

oguzkaganozt self-assigned this on Jul 29

oguzkaganozt requested a review from **youtalk** as a code owner 4 months ago

oguzkaganozt added the tag:run-health-check label on Jul 29

oguzkaganozt requested a review from **VRichardJP** 4 months ago

oguzkaganozt commented on Jul 29 • edited ▾ Contributor Author

Thanks @VRichardJP

youtalk commented on Jul 29 Member

@oguzkaganozt How much has the size of the `devel` image been reduced? It seems to me that the multi-stage process becomes harder to understand, so I want to know if it is really effective.

VRichardJP commented on Jul 29 Contributor

I could build the devel only docker image on my machine in 41 minutes. The image is 15.3GB.
Then I managed to compile latest autoware by mounting it from my host.

However I think the lightweight image should also install the runtime dependencies (from `rosdep-exec-depend-packages.txt`. What do you think?

youtalk commented on Jul 29 Member

@VRichardJP

> I could build the devel only docker image on my machine in 41 minutes. The image is 15.3GB.

I want to know the difference in size before and after merging this PR. Could you check them?

> However I think the lightweight image should also install the runtime dependencies (from rosdep-exec-depend-packages.txt. What do you think?

I don't quite understand the definition of "lightweight," but does it mean deleting the source code while increasing larger artifacts?

VRichardJP commented on Jul 30 Contributor

@youtalk
I think the original docker image is slightly less than 20GB.

By "lightweight" image I mean the bare minimum environment to work on autoware (as developer). The image would contain all build and runtime dependencies so that it is possible to build and run autoware using the image. Ideally, it should not take too long to rebuild the image if a new dependency is to be installed.

Including the src/ directory and compiling autoware in the image is convenient for end users, but it superfluous for developers. In my case, I always mount my own src/ directory and autoware_data/ directories into the container.

With my custom docker setup (described here: https://github.com/orgs/autowarefoundation/discussions/5007#discussioncomment-10110181), it only takes me a couple minutes to rebuild the container when I change the src/ dependencies.

youtalk commented on Jul 30 Member

In that case, it seems like this PR's modification is different from the intended purpose.

@VRichardJP Would it be possible to create separate PRs, one to remove the line https://github.com/autowarefoundation/autoware/blob/main/docker/Dockerfile#L150 and another to install `rosdep-exec-depend-packages.txt` in `devel` stage?

VRichardJP commented on Jul 30 Contributor

@youtalk I am not sure what you mean.
Do you mean to merge the current PR and later create a new one that installs the exec dependencies? This would be fine to me.

oguzkaganozt commented on Jul 31 Contributor Author

@VRichardJP Just added `runtime rosdeps` into the `devel` image. I think now the `devel` image should be sufficient and as lightweight as possible as intended. What do you think ?

@youtalk regarding docker build stages; I just divided `autoware-core` stage into two; one is `autoware-core` that only includes dependencies of core nodes and the `autoware-core-prebuilt` that also includes the prebuilt binaries. So that I can utilize `autoware-core` as a base image for the `devel` image

oguzkaganozt requested a review from **mitsudome-r** as a code owner 4 months ago

oguzkaganozt commented on Jul 31 • edited Contributor Author

@youtalk with this changes `devel-cuda` image is now 14.4 GB

Also fixed a bug which causes artifacts being downloaded into the base image

VRichardJP commented on Jul 31 Contributor

@oguzkaganozt I think it is great!

@youtalk To come back to the purpose of the PR, I think this PR is more aligned with what one may expect from a base "devel" image. This is of course just an opinion.

oguzkaganozt linked an issue on Aug 1 that may be closed by this pull request

Lightweight devel build for docker #4991 Closed

3 tasks

youtalk reviewed on Aug 1 View reviewed changes

`docker/Dockerfile` Outdated Show resolved

`setup-dev-env.sh` Outdated Show resolved

This was referenced on Aug 1

feat(docker): refine devel image [DO NO MERGE] youtalk/autoware#85 Closed

feat(docker): refine devel image [DO NO MERGE] youtalk/autoware#86 Closed

oguzkaganozt requested a review from **youtalk** 3 months ago

oguzkaganozt commented on Aug 7 Contributor Author

@youtalk just made the changes please review it again. Sorry for the latency.

oguzkaganozt added 5 commits 3 months ago

```
Refine devel stage ···                          1944006
add runtime rosdeps to devel image ···          dd119f1
fix:disable downloading artifacts into the base image ···   ae2b662
remove exec rosdep installation ···             8faa135
. ···                                           535b0f9
```

oguzkaganozt force-pushed the `feat/refine-devel-image` branch from **68ba1cb** to **535b0f9** 3 months ago

Compare

mitsudome-r commented on Aug 8 · Member

@oguzkaganozt Is it possible to add a documentation about each stage of Dockerfile somewhere? (perhaps in the README)

Clarifying the definition and expected usage of each stage would help us review in the future to decide what should be included in devel image.

oguzkaganozt commented on Aug 12 · Contributor · Author

> @oguzkaganozt Is it possible to add a documentation about each stage of Dockerfile somewhere? (perhaps in the README)
>
> Clarifying the definition and expected usage of each stage would help us review in the future to decide what should be included in devel image.

Sounds like a good idea will do it !

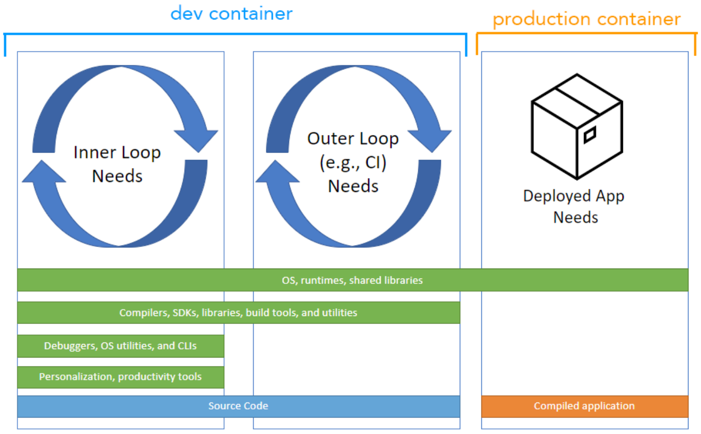
youtalk requested changes on Aug 12

View reviewed changes

youtalk left a comment • edited · Member

I returned from a week and a half of summer vacation and looked at this PR again with a clear mind.
I still feel that the idea of making this PR doesn't align with the original concept of the Dev Container. I think most people will be confused.

https://containers.dev/overview


dev container
production container
Inner Loop Needs
Outer Loop (e.g., CI) Needs
Deployed App Needs
OS, runtimes, shared libraries
Compilers, SDKs, libraries, build tools, and utilities
Debuggers, OS utilities, and CLIs
Personalization, productivity tools
Source Code
Compiled application

I believe the Dev Container should include the source code. If the two of you @oguzkaganozt @VRichardJP want to make the Autoware build "lightweight", wouldn't it be better to delete `/autoware/src`, `/autoware/build`, and `/autoware/install` in the current `autoware-devel` image, then mount the source code and the `ccache` directory before building?

docker/Dockerfile

Comment on lines +144 to +146

```
144 +    apt-get update \
145 +    && cat /tmp/rosdep-universe-depend
146 +    && apt-get autoremove -y && apt-ge
```

youtalk on Aug 12 Member

You need to install `rosdep-universe-depend-packages.txt` packages after this PR is merged. I think it's not efficient and simple.

oguzkaganozt commented on Aug 20 Contributor Author

> I returned from a week and a half of summer vacation and looked at this PR again with a clear mind. I still feel that the idea of making this PR doesn't align with the original concept of the Dev Container. I think most people will be confused.
>
> https://containers.dev/overview
>
> I believe the Dev Container should include the source code. If the two of you @oguzkaganozt @VRichardJP want to make the Autoware build "lightweight", wouldn't it be better to delete `/autoware/src`, `/autoware/build`, and `/autoware/install` in the current `autoware-devel` image, then mount the source code and the `ccache` directory before building?

Source code should be inside the devcontainer, not in the **build-time** but in the **runtime** and this can be done easily with mount option. It is also mentioned the same way on the devcontainer reference.

oguzkaganozt commented on Aug 20 Contributor Author

If we can agree on the terms of when to include source code to devcontainer then I can resolve the conflicts and also this one you mentioned.

youtalk commented on Aug 20 Member

> Source code should be inside the devcontainer, not in the build-time but in the runtime and this can be done easily with mount option. It is also mentioned the same way on the devcontainer reference.

In the case of Autoware, simply mounting the `src` directory is not sufficient. We also need to mount the `build` and `install` directories; otherwise, the build time will be too long. However, in order to have the `build` and `install` directories locally, it seems necessary to have the build environment locally as well. This seems to contradict the concept of Dev Containers.

youtalk commented on Aug 20 Member

One solution to the mounting issue with `src`, `build`, and `devel` directories is the change in the installation directory of the artifacts introduced in https://github.com/orgs/autowarefoundation/discussions/5110. Instead of installing them in `/autoware/install`, they are installed in `/opt/autoware`. After executing `source /opt/autoware/setup.bash` first, they can mount only their partial source code to `/autoware/src` and overlay the ROS workspaces to build and debug.

https://github.com/youtalk/autoware/pull/92/files#diff-f34da55ca08f1a30591d8b0b3e885bcc678537b2a9a4aadea4f190806b374ddcR17
https://github.com/youtalk/autoware/pull/92/files#diff-f34da55ca08f1a30591d8b0b3e885bcc678537b2a9a4aadea4f190806b374ddcR103
https://github.com/youtalk/autoware/pull/92/files#diff-f34da55ca08f1a30591d8b0b3e885bcc678537b2a9a4aadea4f190806b374ddcR107

youtalk mentioned this pull request on Aug 21

feat(docker): install `colcon build` artifacts to `/opt/autoware` instead of `/autoware/install` #5117 Merged

4 tasks

oguzkaganozt commented on Aug 22

Contributor Author

Closing this as #5117 merged.

oguzkaganozt closed this on Aug 22

oguzkaganozt deleted the `feat/refine-devel-image` branch 2 months ago