

Get the chronologically sorted images of a lineage:

join tables lineage_details and lineage_id_to_image_id on lineage_id fields and join the resulting table with image_details table to get committed_date. order by committed_date

something like this:

```
sqlite> SELECT ld.*, iil.image_id, id.committed_date FROM
lineage_details ld JOIN lineage_id_to_image_id iil ON
ld.lineage_id = iil.lineage_id JOIN image_details id ON
iil.image_id = id.image_id WHERE ld.lineage_id =
-1000033263475935320 ORDER BY id.committed_date;
```

Get vulnerabilities of an image:

join image_details and digest_to_scan_report on digest field , and where scan_report is not NULL, join digest_to_vulnerability on digest, and join vulnerability_record on vulnerability_id

something like this

```
sqlite> SELECT id.*, dtsr.report, dtv.vulnerability_id, vr.* FROM
image_details id JOIN digest_to_scan_report dtsr ON id.digest =
dtsr.digest JOIN digest_to_vulnerability dtv ON dtv.digest =
dtv.digest JOIN vulnerability_record vr ON dtv.vulnerability_id =
vr.id WHERE dtsr.report IS NOT NULL AND id.image_id =
'sha256:00005fba3f7c106df1dcdd5753bc18ac6181d9ad0f9aaa17d59d2af765
90c7ed';
```

Lineage definition:

We define the concept of lineage of a tag, representing the chronologically ordered set of images that the tag has ever referenced. A lineage reflects the evolution of a container image for software deployment, including all the runtime systems. The start date of a lineage is the publication date of the first image that is pushed to a tag. A lineage is maintained as long as the tag is included in the associated project's CI/CD metadata file, and thus, the end of life date of a lineage is the publication date of the last image, before the tag specifying the lineage is removed from the CI/CD metadata file. An image may be part of multiple lineages. We define the parent of a lineage as another lineage that is a proper superset of the child's images. In the figure, 8-bullseye is the parent of all the other lineages, while 8.2-bullseye is the parent of 8.2.21-bullseye and 8.2.22-bullseye. We reconstruct the parent-child relationship from the sets of images included, independently of the semantic version tags. A lineage may have multiple parents (8.2.22-bullseye has two: 8.2-bullseye and 8-bullseye). The number of parents allows us to determine, systematically, if a lineage is more generic (fewer parents, e.g. a major release) or more specific (more parents, e.g. a minor release). Generic lineages include all the updates provided by their more specific children, have longer lifespans and may implement more substantial changes.

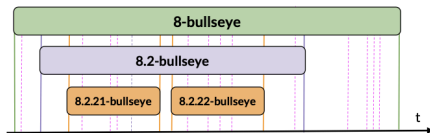


Figure 2: Four lineages of PHP project (chronologically ordered). Solid lines mark the first and last image of each lineage and dotted lines mark the rest of the images.

Lineage construction:

We construct the lineage of an image, based on the image tags. Note that there can be multiple variants of images (e.g., for compatibility across various platforms and CPU architectures such as linux/amd64, linux/arm64, and windows/amd64). The variants share the same tag, and can be built from the same Dockerfile or from separate Dockerfiles. In addition, different projects or different tags of a project may maintain different set of variants. To this end, we restrict the lineages to contain images of same platform and CPU architecture. Consequently, we construct the lineages of an application project with the chronologically ordered set of images that are compatible with a specific architecture and platform, that is or has been referenced by the same tag. For example, for the PHP images with tag 8-bullseye, (8-bullseye, linux, amd64) and (8-bullseye, linux, arm64) specify different lineages. We cover amd64, arm variants 5-8, s390x, ppc64le, mips64le and 386 variants of linux platform. We mark the date of first member image of a lineage as the start date of the lineage. The first image built after the tag and architecture defining the lineage occurs on a CI/CD metadata file on ODI, which is the first image of a lineage. As our dataset covers the images published between January 2018 and April 2023, the exact start dates of lineages that existed before this period are unknown. The time of the last image of a lineage indicates the lineage's end of life. We determine the last image of a lineage as the publication date of the last image before the tag and architecture defining the lineage is no longer included in the CI/CD metadata file on ODI repository.