UbuCon India 2026

#2026

#UbuConIndia2026

Lean, Clean & Secure Artifacts

A Hands-on Intro to Chisel



Soundarya Rangarajan, Canonical

15 Nov 2026 | UbuCon India

Canonical

Qualcomm

penpot



Lean, Clean & Secure Artifacts

A Hands-on Intro to Chisel

Soundarya Rangarajan, Canonical

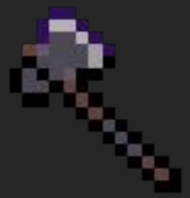
15 Nov 2026 | UbuCon India

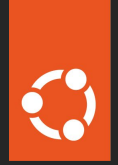


#1 What's Chisel?

A CLI tool that lets you cut a well-defined portion of a deb

? ?

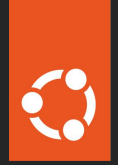




deb

A .deb file is the standard software package format used by Debian

Downloading a deb = bringing in an archive of files

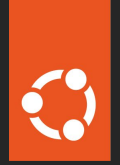


```
ubuntu@dev-space:~$
ubuntu@dev-space:~$ dpkg -L python3.12
/.
/usr
/usr/bin
/usr/bin/pydoc3.12
/usr/bin/pygettext3.12
/usr/lib
/usr/lib/python3
/usr/lib/python3/dist-packages
/usr/lib/python3.12
/usr/lib/python3.12/lib-dynload
/usr/share
/usr/share/applications
/usr/share/applications/python3.12.desktop
/usr/share/doc
/usr/share/doc/python3.12
/usr/share/doc/python3.12/ACKS.gz
/usr/share/doc/python3.12/NEWS.gz
/usr/share/doc/python3.12/README.Debian
/usr/share/doc/python3.12/README.rst.gz
/usr/share/doc/python3.12/README.venv
/usr/share/doc/python3.12/copyright
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/python3.12
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/pdb3.12.1.gz
/usr/share/man/man1/pydoc3.12.1.gz
/usr/share/man/man1/pygettext3.12.1.gz
/usr/share/man/man1/pysetup3.12.1.gz
/usr/share/pixmaps
/usr/share/pixmaps/python3.12.xpm
/usr/bin/pdb3.12
/usr/share/doc/python3.12/changelog.Debian.gz
/usr/share/doc/python3.12/changelog.gz
ubuntu@dev-space:~$
```

A *well-defined portion* of a deb

A subset of the files that make up the deb…

…in a way that holds some meaning.



Each such “*well-defined portion*” is called a slice

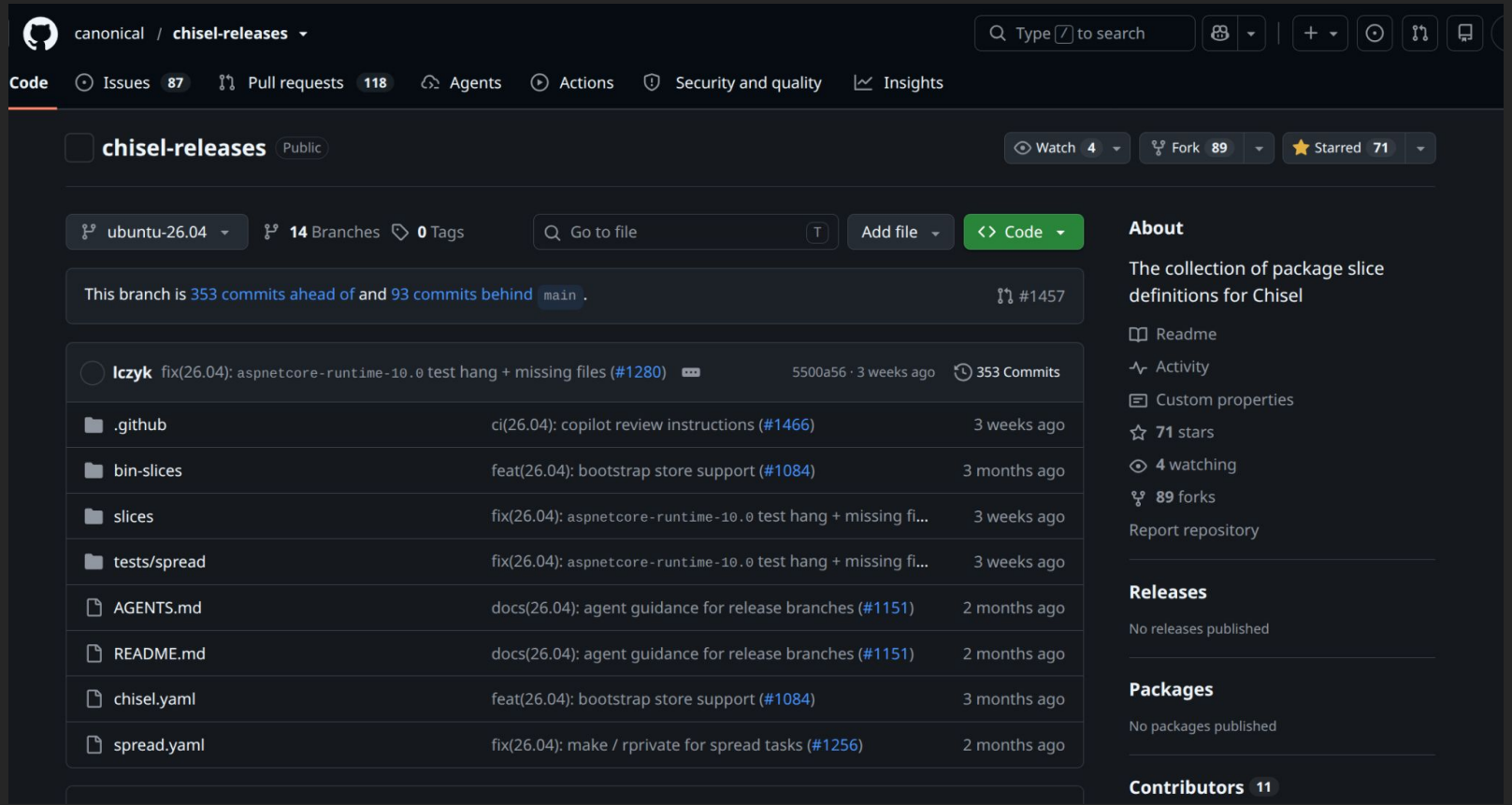


#2 Slice definitions

A slice definition file (SDF) tells chisel how to assemble each of the slices

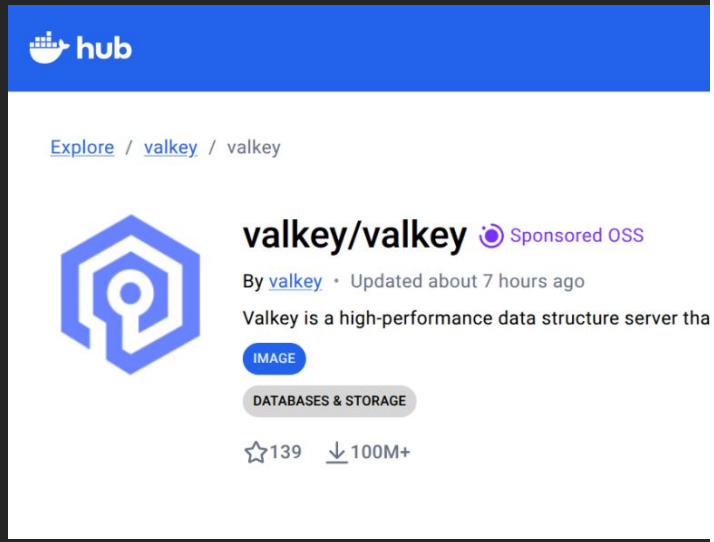
SDFs live in the chisel-releases repo

#2 The `chisel-releases` repo



…why use it?

- Debs are used to build artifacts
- Example: container images for your apps!
- Using thin “slices” of a deb instead brings down size and attack surface
- Chisel-ed containers = lighter + lesser CVEs.



Tag summary

Recent tags

9.0-alpine3.24

Content type

Image

Digest

sha256:2d04268df...

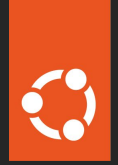
Size

17.4 MB

Last updated

18 days ago

```
docker pull valkey/valkey:9.0-alpine3.
24
```



valkey-io / valkey-container

```
6
7   FROM alpine:3.24 AS base
8
120         chown valkey:valkey /data && \
121         chmod 1777 /data && \
122         mkdir -p /run/valkey && \
123         chown valkey:valkey /run/valkey && \
124         chmod 1777 /run/valkey && \
125         valkey-cli --version && \
126         valkey-server --version
127     WORKDIR /data
128
129     COPY docker-entrypoint.sh /usr/local/bin/
130     ENTRYPOINT ["tini", "--", "docker-entrypoint.sh"]
131
132     EXPOSE 6379
133     CMD ["valkey-server"]
```



Part 1

Define and cut “slices” of `valkey-server` deb



Part 2

Use the slices to build an OCI image



Part 3

Verify it’s working, and compare the improvements against upstream Docker image!

Even the most basic slicing brings you value…

~19

MB

valkey from DockerHub

15

MB

Chiseled valkey

(first results)

…*real* value.

39

Total vulnerabilities

valkey from DockerHub

15

Total vulnerabilities

Chiseled valkey

(first results)

…*real* value.

9

High CVEs

valkey from DockerHub

0

High CVEs

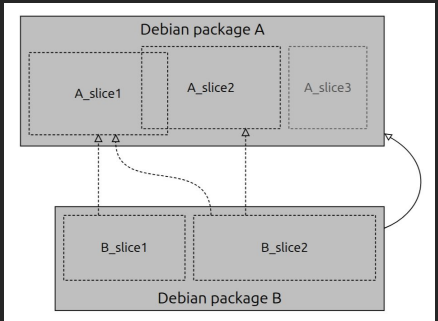
Chiseled valkey

(first results)

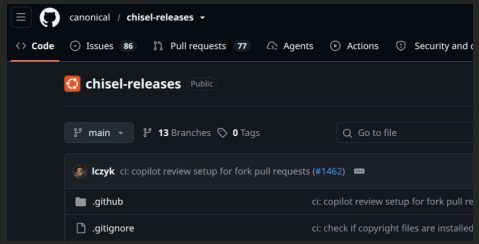
CVE-2026-53612	TOCTOU in the mount program (post-mount ownership/mode)	Binary absent. The rock's util-linux_process ships only choom chrt ionice prlimit setpriv setsid taskset uclampset — no mount
CVE-2026-53613	TOCTOU in mount via ancestor directory swap	Binary absent (same as above)
CVE-2026-53614	SUID mount(8) nosuid/noexec bypass	Binary absent
CVE-2026-76642	Failed external mount helper still runs privileged post-hooks	Binary absent
CVE-2026-78408	nsenter --join-cgroup leaks root cgroup authority	Binary absent — no nsenter in the rock
CVE-2026-78410	Restricted bind mounts don't pin the source	Binary absent (mount)

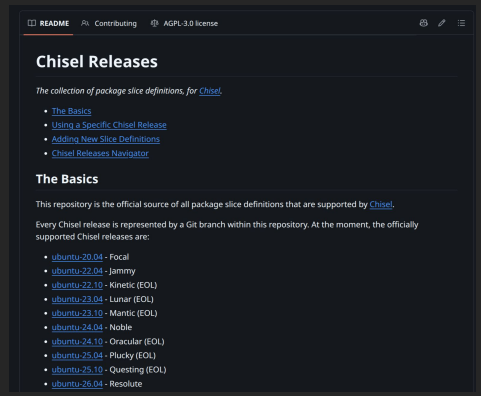
Chisel: a quick summary

1. Chisel lets you cut a “slice” of a Debian package



2. chisel-releases is the official repo of slices





3. You can apply these lean slices while building artifacts.
For instance, in building container images for your apps.



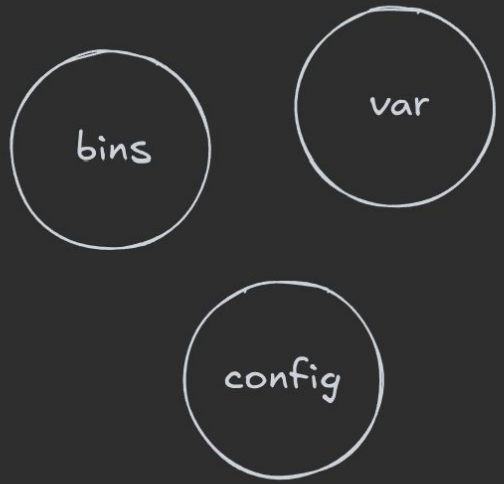
Let’s get to work!

#1 - inspect your deb

```
ubuntu@dev-space:~$ dpkg-deb -c valkey-server_7.2.13+dfsg1-0ubuntu0.1_amd64.deb
drwxr-xr-x root/root         0 2026-06-04 19:24 ./
drwxr-xr-x root/root         0 2026-06-04 19:24 ./etc/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./etc/default/
-rw-r--r-- root/root       273 2025-02-22 03:01 ./etc/default/valkey-server
drwxr-xr-x root/root         0 2026-06-04 19:24 ./etc/init.d/
-rwxr-xr-x root/root      1624 2025-02-22 03:01 ./etc/init.d/valkey-server
drwxr-xr-x root/root         0 2026-06-04 19:24 ./etc/logrotate.d/
-rw-r--r-- root/root       148 2025-02-22 03:01 ./etc/logrotate.d/valkey-server
drwxr-xr-x root/root         0 2026-06-04 19:24 ./etc/valkey/
-rw-r--r-- root/root    107007 2026-06-04 19:24 ./etc/valkey/valkey.conf
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/bin/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/lib/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/lib/systemd/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/lib/systemd/system/
-rw-r--r-- root/root      1895 2026-06-04 19:24 ./usr/lib/systemd/system/valkey-server.service
-rw-r--r-- root/root      3153 2026-06-04 19:24 ./usr/lib/systemd/system/valkey-server@.service
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/share/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/share/doc/
drwxr-xr-x root/root         0 2026-06-04 19:24 ./usr/share/doc/valkey-server/
```

#2 - find a grouping strategy

a. by content



#2 - find a grouping strategy



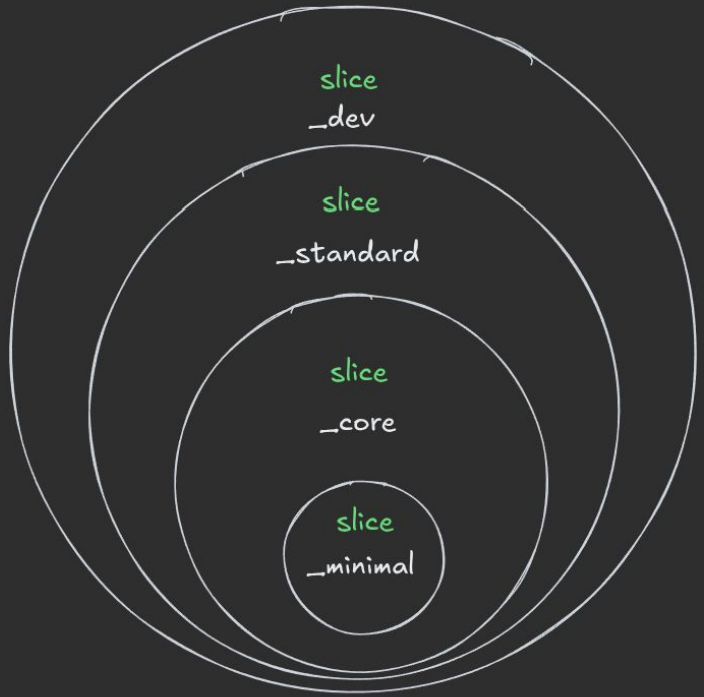
#2 - find a grouping strategy

b. by functionality



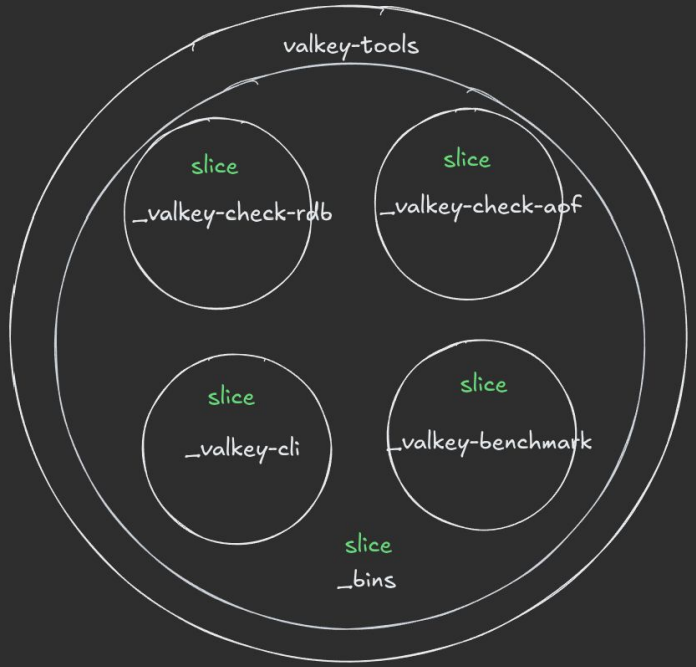
#2 - find a grouping strategy

b. by tier



#3 - Define the granularity of your slices

…or, blend strategies



https://ubuntu.com/chisel/docs/latest/explanation/slice-design-approaches/

#3 - define “essentials”

- Knowledge of your deb helps (duh)
- Follow clues: e.g, symlinks
- Test your slice!
- Use the `strace_bins` slice to help you debug



#4 - test your slice

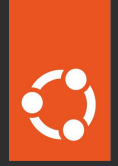
1. mkdir testrootfs

2. chisel cut --release ./ --root testrootfs <your-slice(s)> coreutils_bins strace_bins



Rockcraft: an open-source tool to build OCI-compliant container images.

- **Declarative**. No imperative scripts.
- **Pebble**. Process orchestrator.
- **Chisel**! Trusted Ubuntu, but *just* what you need.



~~Maintainer scripts?~~

- Path properties in your SDF yaml
- But remember: generic scaffolding (/tmp, /run, /var/log, /var/tmp) is owned by `base-files`, so you can include the relevant slices from here

```
slices:
  bins:
    essential:
      zlib1g_libs:
    contents:
      # OpenSSH 9.8 (Ubuntu 25.04) split sshd-session from the sshd listener;
      # OpenSSH 10.0 (Ubuntu 25.10) split authentication into sshd-auth.
      # All three are needed to serve SSH connections.
      /run/sshd/: {make: true, mode: 0755}
      /usr/lib/openssh/sshd-auth:
      /usr/lib/openssh/sshd-session:
      /usr/sbin/sshd:
```

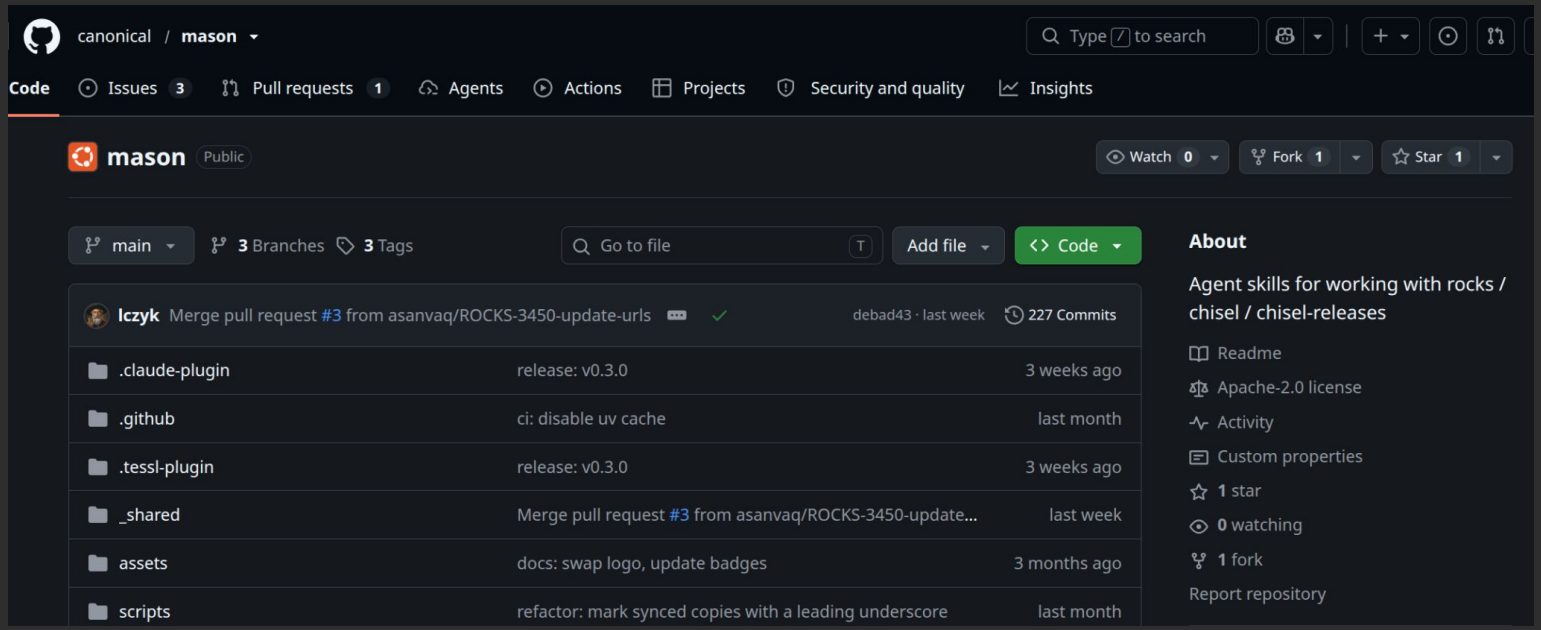
```
package: base-files

essential:
  base-files_copyright:

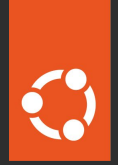
slices:
  base:
    essential: {6 keys}

  etc:
    contents: {2 keys}

  bin:
```



github.com/canonical/mason





- Every slice should strive to be a functioning slice
- Tests! Tests are friends!
- Slices should be tested in isolation

Fill a 1-min survey!



Let’s connect on LinkedIn!

linkedin.com/in/soundarya-rangarajan