

Container craftsmanship  
from a Pebble to a ROCK



cjdcordeiro



crisovaocordeiro



rebornplusplus



rebornplusplus



sergiusens



sergioschvezov

?

What is a ROCK?





What is a ROCK?

Ever used a \*craft tool?



What is a ROCK?

Ever used a \*craft tool?

Ever built a ROCK?

What is a **ROCK**?

# What is a **ROCK**?

“

*A ROCK is an OCI-compliant Ubuntu LTS-based container image with a well-defined and opinionated design that meets the security, stability and reliability requirements from cloud-native software.*



A close-up photograph of two hands held palm-up against a dark background. The left hand holds a single red pill, and the right hand holds a single blue pill. The text 'But, why ROCKs?' is overlaid in the center. The word 'ROCKs' is highlighted with a light green rectangular background.

But, why **ROCKs**?

## 36. Selection criteria for container images

Which of the following do you value the most when selecting a base image of a container image? Choose 3.

1142 out of 1166 people answered this question (with multiple choice)



Opinionated and consistent design

1

*predictable entrypoint and built-in metadata*

User-centric experience

2

*it's Ubuntu, it's declarative*

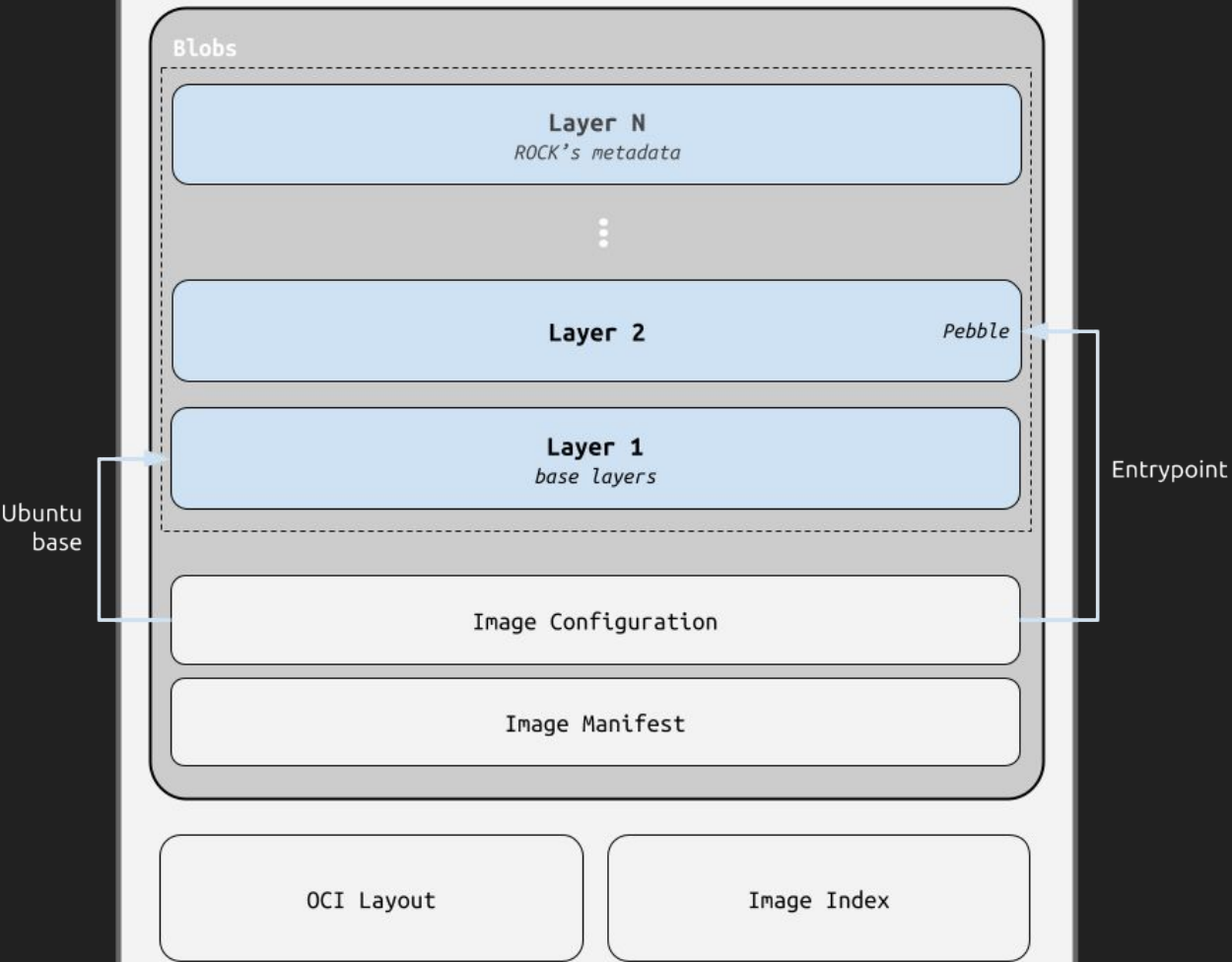
Security by design

3

*latest & greatest - natively compact and secure*

What sets **ROCKs** apart?

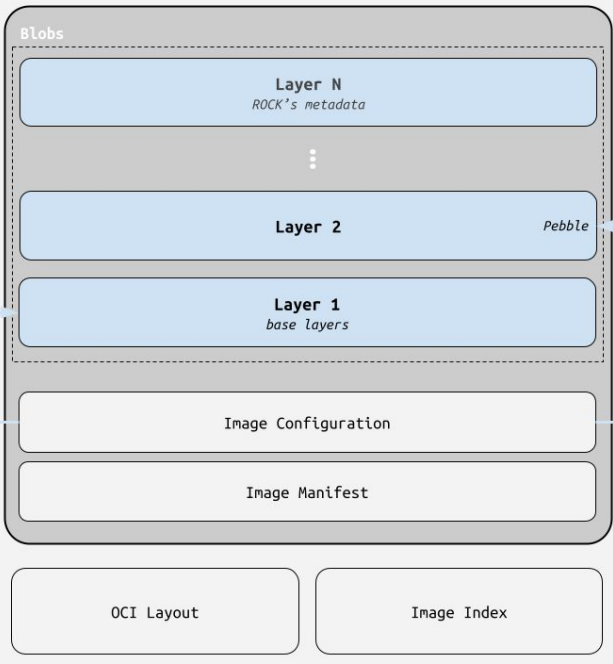
# ROCK



# The Design

ROCK

Ubuntu  
base



# Pebble?

“

*...seamless orchestration of local service processes as an organised set*

- 1 Consistent UX at runtime
- 2 Client-server model behind a single binary
- 3 Declarative service definition, with layering
- 4 Services are first-class citizens
- 5 Container-optimized init process
- 6 Embedded utilities

# *Wanna try a ROCK?*

```
$ docker pull ubuntu/prometheus:2.46.0-22.04_stable
```

```
$ docker pull ubuntu/mlflow:2.1.1_1.0-22.04
```

```
$ docker pull ubuntu/alertmanager:0.25.0-22.04_stable
```

```
$ docker pull ubuntu/grafana-agent:0.35.2-22.04_stable
```

```
...
```

So...what is **Rockcraft** then?

“

*Rockcraft is a tool to create ROCKs.*

# Why do you ~~need~~ want Rockcraft?

*built by Canonical and tailored for ROCKs*

1

From Ubuntu, to Ubuntu

Focus on what matters - the image

2

*abstracts the repetitive and boilerplate steps*

*select the build provider: LXD, host, etc.*

3

User-defined compartmentalization

Easy and familiar, for all levels

4

*same declarative language as all craft tools*

ISVs

Cloud-native developers

Occasional container users



Dockerfile is to docker build...

...as **rockcraft.yaml** is to **rockcraft**

# rockcraft.yaml sample:

```
name: my-flask-app
base: ubuntu@20.04
version: "0.1"
summary: A cool Flask ROCK
description: |
  A cool ROCK for a Flask server that says hi
license: GPL-3.0
platforms:
  amd64:

services:
  rockserver:
    override: replace
    startup: enabled
    command: python3.8 /usr/bin/app
    on-failure: shutdown

parts:
  python-flask:
    plugin: python
    source: src
    stage-packages:
      - python3.8
```

## rockcraft.yaml sample:

```
name: my-flask-app
base: ubuntu@20.04
version: "0.1"
summary: A cool Flask ROCK
description: |
  A cool ROCK for a Flask server that says hi
license: GPL-3.0
platforms:
  amd64:

services:
  rockserver:
    override: replace
    startup: enabled
    command: python3.8 /usr/bin/app
    on-failure: shutdown

parts:
  python-flask:
    plugin: python
    source: src
    stage-packages:
      - python3.8
```

What packages are being installed in the ROCK?

## rockcraft.yaml sample:

```
name: my-flask-app
base: ubuntu@20.04
version: "0.1"
summary: A cool Flask ROCK
description: |
  A cool ROCK for a Flask server that says hi
license: GPL-3.0
platforms:
  amd64:

services:
  rockserver:
    override: replace
    startup: enabled
    command: python3.8 /usr/bin/app
    on-failure: shutdown

parts:
  python-flask:
    plugin: python
    source: src
    stage-packages:
      - python3.8
```

What Ubuntu release is the ROCK based on?

## rockcraft.yaml sample:

```
name: my-flask-app
base: ubuntu@20.04
version: "0.1"
summary: A cool Flask ROCK
description: |
  A cool ROCK for a Flask server that says hi
license: GPL-3.0
platforms:
  amd64:

services:
  rockserver:
    override: replace
    startup: enabled
    command: python3.8 /usr/bin/app
    on-failure: shutdown

parts:
  python-flask:
    plugin: python
    source: src
    stage-packages:
      - python3.8
```

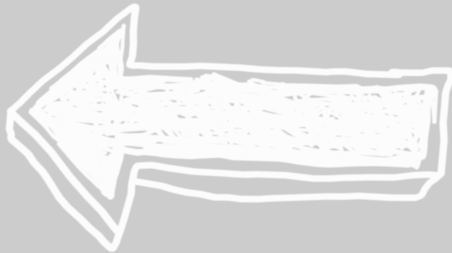
What is the container output on `docker run`?

# rockcraft.yaml sample:

```
name: my-flask-app
base: ubuntu@20.04
version: "0.1"
summary: A cool Flask ROCK
description: |
  A cool ROCK for a Flask server that says hi
license: GPL-3.0
platforms:
  amd64:

services:
  rockserver:
    override: replace
    startup: enabled
    command: python3.8 /usr/bin/app
    on-failure: shutdown

parts:
  python-flask:
    plugin: python
    source: src
    stage-packages:
      - python3.8
```

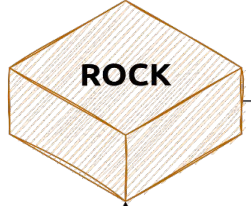


```
rockcraft.yaml
name: my-rock
base: ubuntu-22.04
version: 0.1
license: GPL-3.0
parts:
...
```

rockcraft pack

Project Validation

OCI Archive



skopeo copy

- Registry
- Docker daemon
- ...

Provider (eg. LXD)

\$CRAFT\_PROJECT\_DIR

1. pull

For each part  
\$CRAFT\_PART\_SRC

2: overlay

For each part  
\$CRAFT\_OVERLAY

3: build

For each part  
\$CRAFT\_PART\_BUILD → \$CRAFT\_PART\_INSTALL  
organize

4: stage

For each part  
\$CRAFT\_STAGE

5: prime

For each part  
\$CRAFT\_PRIME

---Toolchain---



Plugins



Chisel



umoci

# Concepts and terminology: workshop cheat sheet

Term	Description
<i>OCI</i>	The Open Container Initiative.
<i>OCI image</i>	A container image that follow the OCI image specification (i.e. is OCI compliant).
<i>OCI archive</i>	A container image in the format of a compressed archive. <a href="#">Rockcraft's default output format</a> .
<i>(OCI) Entrypoint</i>	The container image entrypoint (e.g. <code>docker inspect &lt;img&gt; -f '{{.Config.Entrypoint}}'</code> ).
<i>(OCI) User</i>	The container user that runs the Entrypoint (e.g. <code>docker inspect &lt;img&gt; -f '{{.Config.User}}'</code> ).
<i>base</i>	The Ubuntu LTS OCI image to be shipped within the ROCK. If the base is <b>bare</b> , then the ROCK won't have any bits from the Ubuntu LTS image within, and an "build-base" needs to be specified.
<i>build-base</i>	The Ubuntu LTS OCI image where the ROCK's parts will be built.
<i>part(s)</i>	Rockcraft's parts.
<i>craft env vars</i>	A shorthand to refer to the Part's reserved environment variables (e.g. <code>CRAFT_PART_INSTALL</code> ).
<i>(part) lifecycle</i>	Used to refer to the parts' lifecycle stages: <i>pull</i> , <i>overlay</i> , <i>build</i> , <i>stage</i> and <i>prime</i> .
<i>plugin(s)</i>	A part's plugin. See all the supported plugins in <a href="#">Rockcraft's reference docs for "Part properties"</a> .
<i>service(s)</i>	The Pebble services to be managed by the ROCK's Pebble Entrypoint.



Hands-on



> Install Rockcraft

[https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/A.How\\_to\\_install\\_rockcraft](https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/A.How_to_install_rockcraft)

> How to build a **ROCK**?

[https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/B.How\\_to\\_build\\_a\\_rock](https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/B.How_to_build_a_rock)

> Inspect and run a **ROCK**

[https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/C.Inspect\\_and\\_run\\_a\\_rock](https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/C.Inspect_and_run_a_rock)

> Create a chiselled **ROCK**

[https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/D.Create\\_a\\_chiselled\\_rock](https://github.com/cjdcordeiro/pebble-to-rock-workshop/tree/main/D.Create_a_chiselled_rock)

# DIY

## Challenge A

We want a ROCK that runs an Apache (``apache2``) web server when the container is started.

## Challenge B

You can find the Golang application "assistant" in this folder. The objective is to create an Ubuntu ROCK called "challenge-b", that runs the ``assistant`` binary every time a container is deployed.

## Challenge C

Remember the sample? The one with the Flask application...well, let's take it, and rebase it on Ubuntu 22.04 instead, whilst keeping it chiselled and baseless.

# References

- [Rockcraft docs](#)
- [Ubuntu containers](#)
- [Workshop: hands-on exercises](#)
- Existing Ubuntu ROCKs (on [Docker Hub](#) and [ECR](#))

slido



## Feedback

ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.



## Copyrights and attributions

- [\[CC BY-NC-SA 2.0 DEED\]](#) [nocturbulous](#)
- <https://imgflip.com/i/827kq9>