
sphinx-likec4

Release 0.2.1

ckeller

Sep 25, 2026

Contents

1	Quickstart	1
2	Directives	2
2.1	likec4-view	2
2.2	likec4-model	3
3	Configuration	3
3.1	Static images and PDF	4
4	Example — cloud-system	4
4.1	Structural view	5
4.2	Sequence (dynamic view)	5
4.3	Static image	6

Embed interactive [LikeC4](#) architecture views — including `dynamic view` **sequence diagrams** — in Sphinx documentation. The extension runs the pinned LikeC4 CLI for you, caches the build, and validates every embedded view id.

Also available as a [PDF](#) — built by the same extension, with every view rendered as a static image.

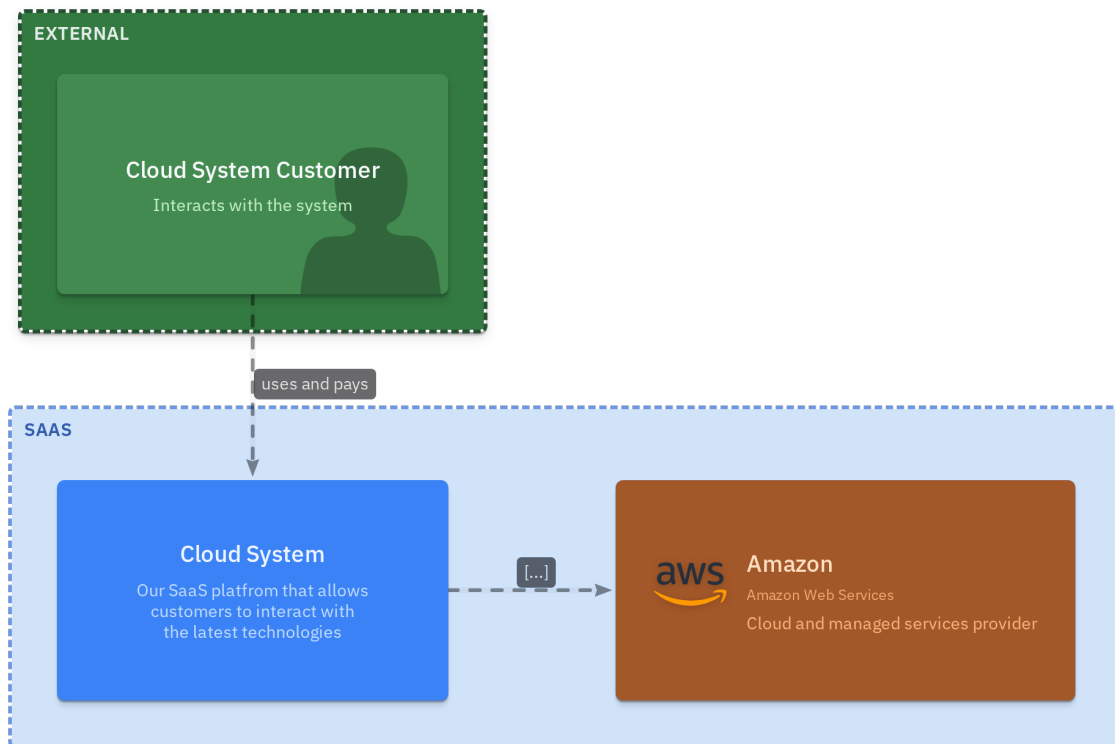
1 Quickstart

```
pip install sphinx-likec4           # node >= 20 required at doc-build time
```

```
# conf.py
extensions = ["sphinx_likec4"]
likec4_source_dir = "model"        # your *.c4 / *.likec4 files
```

```
.. likec4-view:: index
```

That renders this (LikeC4's official *cloud-system* example):



In a PDF or epub build the same directive embeds a static PNG instead; `:render: png` does that on an HTML page too — see *Directives*.

2 Directives

2.1 likec4-view

Embeds one view as an interactive iframe. **Dynamic views (sequences) share the same id space** and embed identically.

```
.. likec4-view:: cloud-to-amazon
   :height: 420px
   :title: Upload flow
```

MyST form:

```
```{likec4-view} cloud-to-amazon
:height: 420px
```
```

Options: `height` (default 460px), `title` (iframe title, for accessibility), `render` (see *Static images*), and `mode` — `diagram` (viewer default) or `sequence`, which opens a **dynamic view directly in its sequence rendering** (appends the viewer's `?dynamic=` parameter):

```
.. likec4-view:: cloud-to-amazon
   :mode: sequence
```

An unknown view id **fails the build** (works with `-W`).

Static images

render picks how the view is embedded: `iframe` (HTML default), `png`, `jpg`, or `text`. It is a **preference, not a demand** — the same source builds for every output format:

```
.. likec4-view:: cloud
:render: png
:width: 80%
:alt: Cloud, structural view
```

- On non-HTML builders (`latex/PDF`, `epub`, ...) views are images by default; `:render: iframe` there falls back to the builder's default.
- On builders that can't embed images at all (`text`, `man`, `linkcheck`) everything is plain text and no image export runs.
- `:render: jpg` needs `"jpg"` somewhere in `likec4_render` (only PNG is exported unless asked).
- In image mode `width`, `height`, `alt`, `align`, `scale` pass through to the image with the standard docutils validators; `alt` defaults to `title`.
- `:mode: sequence` works in image mode too: dynamic views are exported a second time in sequence layout (`--seq`), and the directive picks that rendering. On non-dynamic views the option is ignored, as in the viewer.

2.2 likec4-model

Embeds the whole viewer gallery, or just links it:

```
.. likec4-model::
:height: 600px

.. likec4-model::
:link-only:
```

3 Configuration

| conf.py value | default | meaning |
|-----------------------|--|---|
| <code>likec4_s</code> | — (required) | directory of <code>.c4/.likec4</code> sources, relative to <code>conf.py</code> |
| <code>likec4_v</code> | <code>pinned</code> in <code>src/sphinx_likec4/package.json</code> (Dependabot keeps it current) | exact CLI version run via <code>npx -y likec4@<version></code> |
| <code>likec4_m</code> | <code>"error"</code> | when <code>node/npx</code> is absent: <code>error</code> fails the build, <code>warn</code> renders placeholders |
| <code>likec4_b</code> | <code>[]</code> | extra arguments appended to <code>likec4 build</code> |
| <code>likec4_r</code> | <code>{}</code> | default render mode per output format, keyed by builder format (<code>"html"</code> , <code>"latex"</code> , ...; <code>epub</code> uses <code>"epub"</code>), e.g. <code>{"latex": "jpg", "epub": "png"}</code> ; values <code>iframe</code> , <code>png</code> , <code>jpg</code> , <code>text</code> |
| <code>likec4_e</code> | <code>True</code> | set <code>False</code> to never run the image export: HTML renders iframes only, non-HTML builders fall back to plain text, no Chromium needed |

Node ≥ 20 must be on `PATH` at doc-build time. The build is **cached** on a content hash of the sources + version + args, so incremental Sphinx builds don't re-run the CLI. The viewer is copied into the output at `_likec4/` and uses hash routing, so it works from any subpath.

The `likec4_missing="warn"` warning is tagged `type="likec4"`, so it can be silenced even under `-W` with `suppress_warnings = ["likec4"]` in `conf.py`.

3.1 Static images and PDF

Only the views a build actually embeds are exported: a page's directive renders its view on first use (`likec4 export png --flat -f <id>`), and rebuilds re-export the remembered set in one run per format when the sources changed. Plain HTML never starts a browser unless a page asks for a static image (`:render: png` or `:render: jpg`); unreferenced views are never rendered. Parallel reads (`-j`) are safe: the cache directories are validated once in the main process before the workers start, and concurrent exports into them coexist. A first image build pays one CLI run (browser start included) per newly referenced view; rebuilds re-export the remembered set in a single run per format only when the sources changed. The renders live under the doctree dir (`likec4/images-<fmt>/`), cached on the same content hash as the viewer. If the batched re-export fails on a builder whose default is the iframe, the build continues with a `likec4` warning and `:render: png/jpg` fall back to the iframe; an export failure for an explicitly requested image is an error. Set `likec4_export_images = False` to disable images entirely. The export renders in headless Chromium through Playwright. If no browser is present the extension installs one **once**, using `likec4`'s own Playwright so the revision matches:

```
npx -y --package likec4@<pinned version> -c 'playwright install chromium' # what_
↪ it runs for you
```

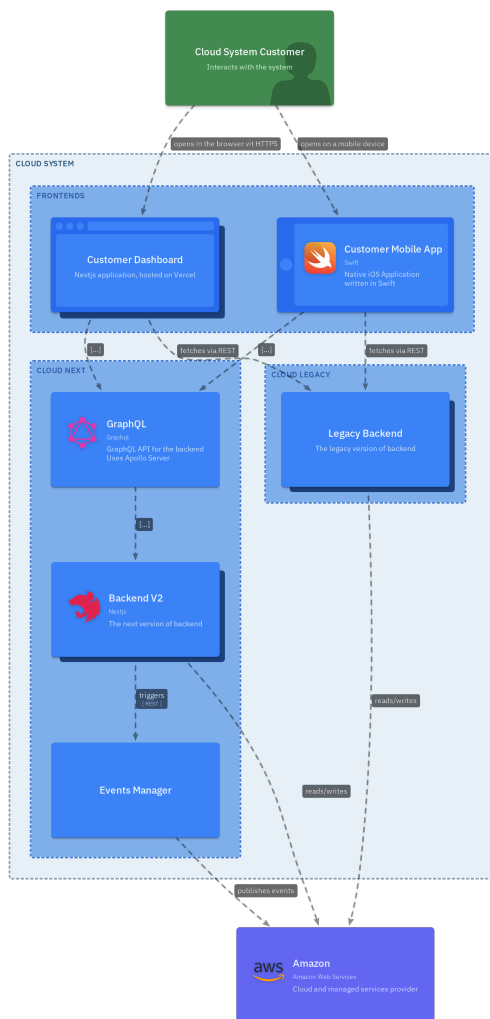
(≈150 MB, into Playwright's cache under your home directory.) On a minimal Linux CI image add `--with-deps` to that command yourself if Chromium complains about missing shared libraries.

For a PDF: `sphinx-build -M latexpdf docs docs/_build` — nothing LikeC4-specific to configure. The extension's own docs use `latex_engine = "xelatex"` for the Unicode arrows in the text.

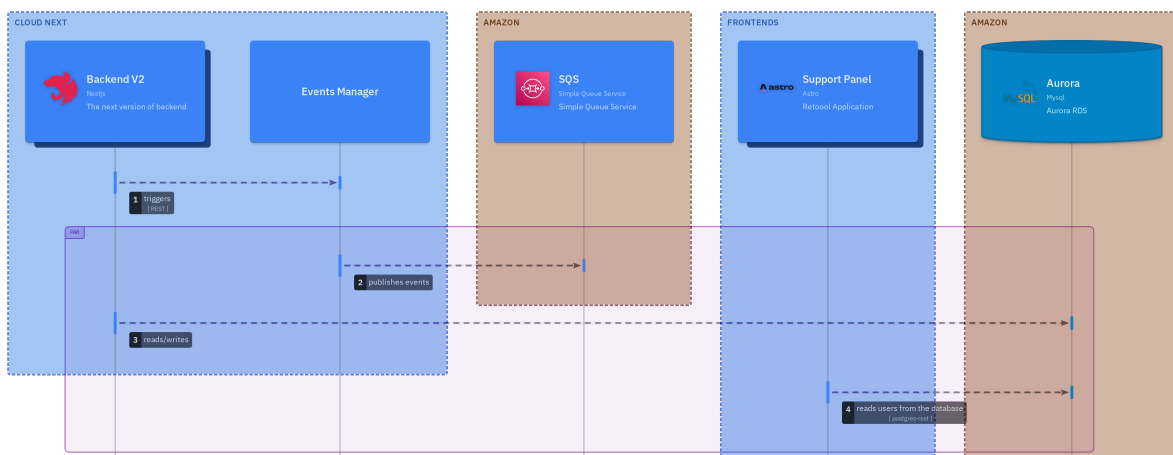
4 Example — cloud-system

LikeC4's official `cloud-system` [example](#) (MIT), fetched at docs-build time from a pinned upstream commit (see `docs/conf.py`) — nothing is copied into this repository.

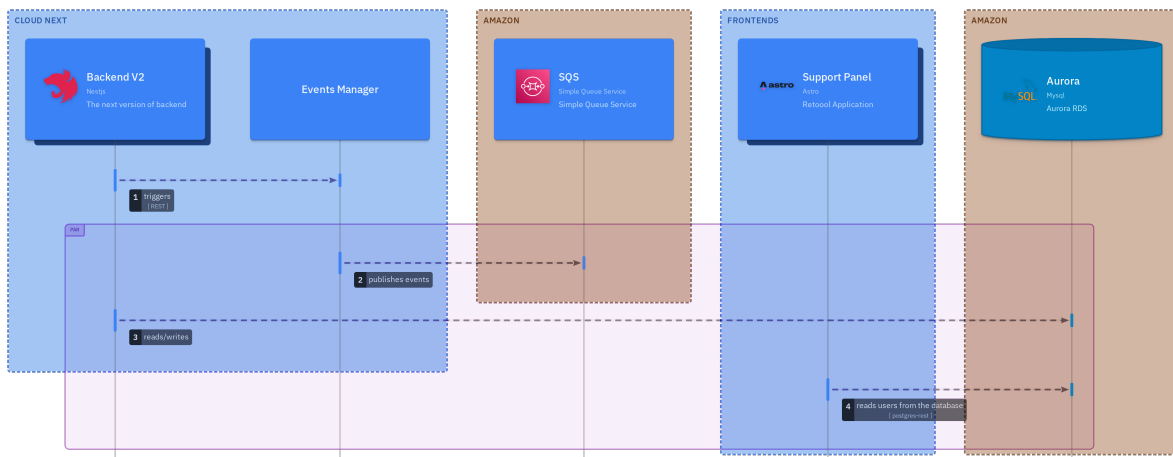
4.1 Structural view



4.2 Sequence (dynamic view)

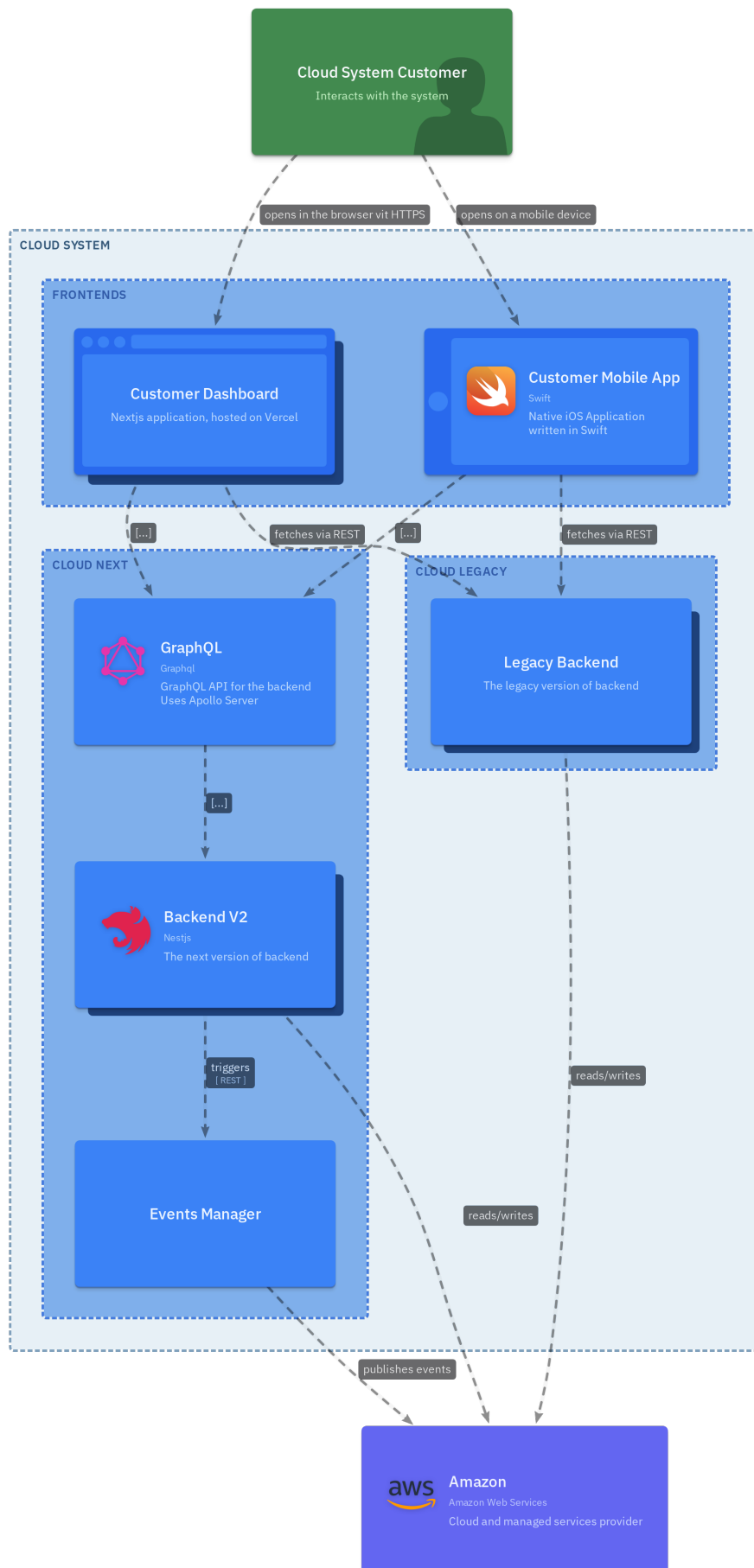


The same dynamic view as a static **sequence** PNG (:mode: sequence + :render: png) — this is what a PDF build gets for it:



4.3 Static image

The same structural view rendered as a **PNG** — this is what every non-HTML builder (LaTeX/PDF, epub) gets by default, and what `:render: png` gives you on an HTML page:



Source: `model.c4`