



Don't just write documentation, Design it

Keirthana T S

Senior technical author

Documentation Practice @ Canonical

UbuCon India 2025



Technical author

Technical authority

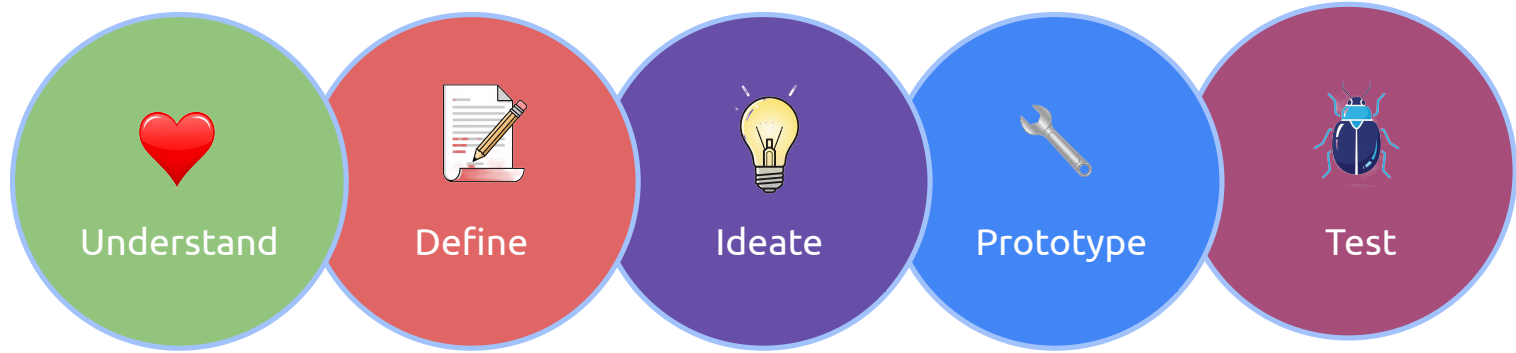
Leading documentation efforts

Design journey





Design and documentation





When to engage with documentation and design?

Migrate to 5.2



Drop support for 5.0 and 5.1



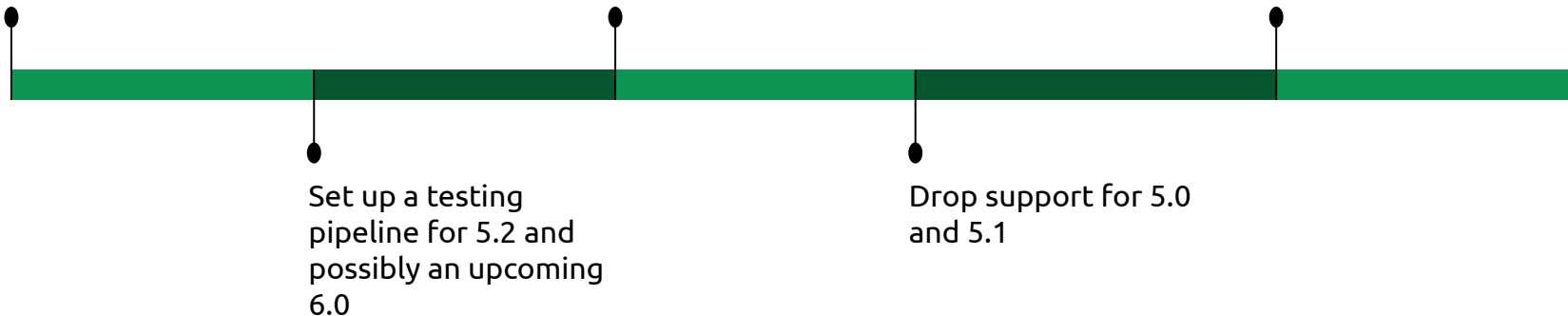
When to engage with documentation and design?

Support migration to 5.2

Backward compatibility and support for 5.0 and 5.1

Deprecate 5.0 and 5.1

Truly migrated to 5.2 and prepared for 6.0





Influencing the design of a product



Initialize

Questions with
defaults highlighted;
CLI guides new users
to safely go with
defaults



Launch an instance

First step fails
because user is not
authorized - one of
the default choices
was not to authorize
by default, for
security reasons.

Confusion



Designing documentation for the right audience

Ubuntu Core documentation

[Give feedback](#)

Ubuntu Core is Ubuntu engineered for embedded and IoT systems. It's image-based and immutable, with every element confined within a separate sandbox.

It provides a deployment infrastructure for developers to deploy production images, creating a minimal, secure, and transaction-based operating environment for your applications.

Ubuntu Core reduces the time needed to flash and provision images in production lines, ensuring they're secure while delivering the desired user-experience to your users.

From Linux and make space tinkers, to the robotics, automotive and signage industries; from a single device, to a deployment of thousands. From a tiny SoC to a fleet of full-powered PCs, Ubuntu Core can handle it.

In this documentation

Tutorials Get started - a hands-on introduction to Ubuntu Core for new users Explanation Concepts - discussion and clarification of key topics	How-to guides Step-by-step guides covering key operations and common tasks Reference Technical information - specifications, APIs, architecture
---	--

- **Stores:** [Store overview](#) | [Brand accounts](#) | [Dedicated snap stores](#) | [Store scoring](#)
- **Security:** [Full disk encryption](#) | [Sandboxing](#) | [Use a recovery mode](#)
- **Management:** [Update control](#) | [Remodelling](#) | [Upgrade Ubuntu Core](#)
- **Core elements:** [Introduction](#) | [Storage layout](#) | [Snap in Ubuntu Core](#)

Project and community

Ubuntu Core is a member of the Ubuntu family. It's an open source project that welcomes community projects, contributions, suggestions, fixes and constructive feedback.

- [Community engagement commitment](#)
- [Our Code of Conduct](#)
- [How to get support](#)
- [Join the Discourse forum](#)
- [Interactive chat on Matrix.org](#)
- [Product roadmap](#)

Thinking about using Ubuntu Core for your next project? [Get in touch!](#)

Anbox Cloud documentation

[Give feedback](#)

Anbox Cloud runs Android in the cloud using lightweight LXD system containers or full virtual machines. Built on Ubuntu, it provides a scalable platform to deploy, manage, and stream Android workloads across public or private infrastructure. Compared to other Android emulation solutions, Anbox Cloud can provide at least twice the density and can serve up to 100 Android instances per server.

Due to its high scalability, Anbox Cloud enables users to deliver Android experiences with consistent performance and responsiveness.

Automotive OEMs can develop and test user-friendly infotainment systems, without having to rely on hardware availability and configuration. Android application developers can preview and interact with UI changes instantly, reducing turnaround time for application development. Enterprises offering remote access to Android instances with a customized set of applications, as a service. Cloud gaming service providers can deliver Android-based gaming experiences at scale with high performance and low latency.

Anbox Cloud comes in two variants:

The **appliance** is available as a snap package that combines all components of Anbox Cloud for deploying on a single, dedicated machine. It is suitable for small scale deployments and development environments. If you are evaluating how well Anbox Cloud fits your need, try out the appliance.

The **charmed deployment** is a full deployment of all Anbox Cloud charms using Juju. It is more suitable for large scale environments and production deployments. If you have tried out the appliance and looking to expand more into a multi cluster deployment, install Anbox Cloud charms using Juju.

In this documentation

Tutorial Start here - a hands-on introduction to Anbox Cloud <ul style="list-style-type: none">• Install Anbox Cloud Appliance• Create a virtual device	How-to guides Key operations and common tasks <ul style="list-style-type: none">• Install appliance Charmed Anbox Cloud• Work with applications and instances• Harden your deployment• Upgrade• Troubleshoot View logs	Reference Technical information <ul style="list-style-type: none">• AMS configuration• Application manifest Addon manifest• CLI help API help• Release notes
--	---	--

Project and community

Anbox Cloud is a product developed by Canonical. While it was initially based on the open-source Anbox project (archived in [Github](#)), its codebase has since become entirely independent.

We welcome community involvement through suggestions, fixes and constructive feedback both on the product and its documentation. You can engage with the Anbox Cloud team and the community using the following channels:

Engage Discourse Matrix Contact us	Contribute Contribution guide Style guide	Get help File a bug Get support
--	---	---

Canonical Kubernetes documentation

[Give feedback](#)

Canonical Kubernetes is a performant, lightweight, secure and opinionated distribution of Kubernetes which includes everything needed to create and manage a scalable cluster suitable for all use cases.

Canonical Kubernetes builds upon upstream Kubernetes by providing all the extra services such as a container runtime, a CNF, DNS services, an ingress gateway and more that are necessary to have a fully functioning cluster all in one convenient location - a snap!

Staying up-to-date with upstream Kubernetes security patches and updates with Canonical Kubernetes is a seamless experience. Freeing up time for application development and innovation without having to worry about the infrastructure.

Whether you are deploying a small cluster to get accustomed to Kubernetes or a huge enterprise level deployment across the globe, Canonical Kubernetes can cater to your needs. If you would like to jump straight in, head to the [snap getting started tutorial](#)!

In this documentation

Canonical Kubernetes snap The k8s snap is a self-contained, secure and dependency-free Linux app package used to deploy and manage a Canonical Kubernetes cluster. If you are new to Kubernetes, start here.	Canonical Kubernetes charms The k8s charms take care of installing, configuring and managing Canonical Kubernetes on cloud instances managed by Juju.	Canonical Kubernetes and Cluster API Using Cluster API's declarative tooling, deploy and manage multiple Canonical Kubernetes clusters.
--	---	---

Project and community

Canonical Kubernetes is a member of the Ubuntu family. It's an open source project which welcomes community involvement, contributions, suggestions, fixes and constructive feedback.

- [Our Code of Conduct](#)
- [Our community](#)
- [How to contribute](#)
- [Our release notes](#)



Designing documentation for the right audience

> How-to

- > Install the dev variant
- > Install the prod variant
- > Create application (both)
- > Configure OIDC (dev)

> Explanation

- > Architecture

> Reference

- > Configuration (both)
- > Charm configuration (prod)

> Dev variant

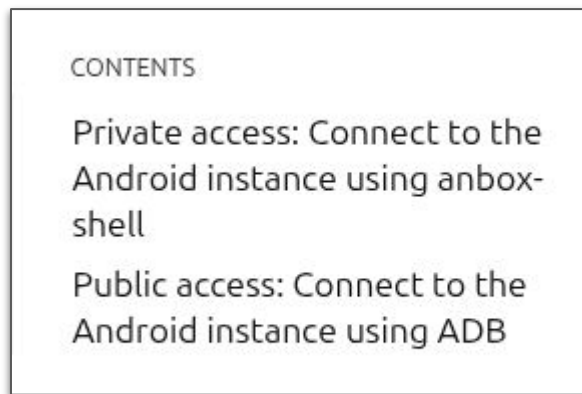
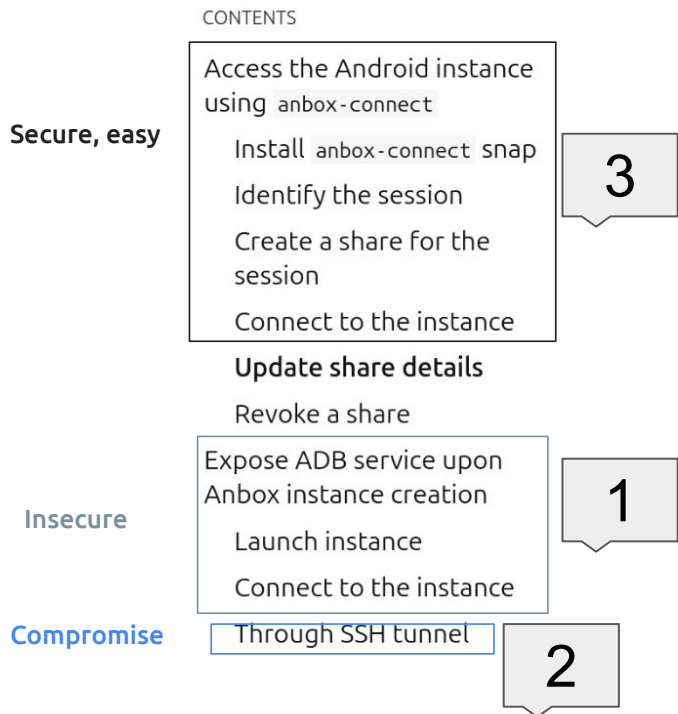
- > Installation
- > Architecture
- > Configuration

> Prod variant

- > Installation
- > Architecture
- > Configuration



Documenting the UX journey instead of the product journey





Connecting the dots

Engineering \neq ~~coding~~ \Rightarrow **Solving** the problems

Design \neq ~~visuals~~ \Rightarrow **Understanding** the needs

Documentation \neq ~~writing~~ \Rightarrow **Guiding** the user path

Credit: Mengmeng Tang, Technical Author, Canonical



Interested?

<https://documentation.academy/>



<https://github.com/canonical/open-documentation-academy>
y



Open Documentation Academy

Search

Go

Our vision

Recognition

Academy events

Participating projects

Academy handbook

Writing resources

Release notes

Give feedback



Open Documentation Academy

Discover open source through documentation

Canonical's Open Documentation Academy (CODA) is a collaboration between Canonical's documentation team and documentation newcomers, experts, and those in-between, to help us all improve documentation, become better writers, and better open source contributors.

We want to lower the barrier to successful open-source software contributions, by making documentation into the gateway, and to be the model for welcoming and constructive collaboration.

How it works

- Find a [task](#) that you're interested in (*there are many*)
- Ask to be assigned the task (*no obligations*)
- Work on the task with the help support of our mentors (*we're friendly*)
- Finish, and get recognition for your work (*expand your CV and help open source*)

Who's it for

For **newcomers**, we provide help, advice, mentorship, and a hundred different tasks to get started on.

If you're an **expert**, we want to create a place to share knowledge, a place to get involved with new developments, and somewhere you can ask for help on your own projects. We'd also like to encourage new projects to get involved, and become sources of their own tasks on the Academy.

If you're a **project maintainer**, we warmly invite you to partner with the Academy, create your own tasks, attract contributors, get support and find a community.



Interested?



We are hiring!



Thank you!

Questions?