

Community input to Juju developers

2022-April-22

What is this?

The Juju community has been running workshops every Friday for 30 weeks consecutively - focusing on user education, development of charms, documentation, user stories and all things Juju.

These workshops have become a success in many ways, with participants being a mix of independent contributors/users and Canonical staff members. Many attending every week. The last workshop, 21 persons attended.

The connection between the developers and the community is however weak and we hope to address this by an initiative to help the core developers understand what the community needs might look like.

Our take on this, is to host a workshop, focusing on the community members and their every day challenges and wishes in the realm of Juju. The workshop invitation [looked like this](#).

This document is the digest and outcome from that workshop in a format we hope can make sense to developers and Juju project managers/contributors. Hopefully, we can get the community connected with the future development sprints early and continuously improve Juju as time goes by.

Instructions to the participant

- Think about what YOU like to see developed, don't try to think too big or for anyone else. Be selfish - what's YOUR desire.
- Don't bother being too detailed - instead - provide a contact to yourself such as developers can reach out to you for more information.

Desire: More ice cream

What: I would like to have ice cream

Why: Because ice cream is nice.

Contact me at: john.doe@example.com

Desire: Actions with access-control

What: I would like to be able to restrict access to certain juju actions such that I can allow or not allow certain users to be able to run, let's say security related actions or that perhaps reveals secrets. It could be more ways to achieve this.

Why: Juju actions is very powerful tool but anyone with "write" permissions can run all actions. For operations-staff, some actions - like "**juju run-action get-admin-password -wait**" is way too scary to let "anyone" have access to with simple write permissions. Its very much "all or nothing" here, which makes it problematic if I have many charms where some should not have this kind of access, where some other charms would be OK.

Contact me at: erik.lonroth@gmail.com

Desire: Deploy juju controller via helm chart

What: I would like to provision a juju controller by means of helm chart.

Why: It is difficult to integrate juju to k8s infrastructure. Aka you need to run “juju bootstrap” from somewhere.

Contact me at:

Desire: Operating system agnostic instance provisioning

What: I would like to provision linux based operating systems other than ubuntu or centos using juju.

Why: Oftentimes software systems do not support ubuntu and the disconnection from juju and the other operating systems prevents juju from being used in many scenarios.

[Joseph talked about series vs bases] which would allow future work towards this.

Contact me at: john.doe@example.com

Desire: Juju ecosystem installable via package manager

What: I would like to use apt/yum/dnf/pacman to install juju and charmcraft.

Why: we need to control our infrastructure and follow an enterprise way for software provisioning: well defined calendar for updates, production systems should never change before staging (this includes the juju client), ability to downgrade software in case the new version contains regressions, and so on. This all makes snaps unsuited for distributing software. We need proper packages (rpm, deb, etc) for the Juju ecosystem (Juju, LXD, charmcraft).

Contact me at:

Desire: Versioned docs

What: I would like documentation to be versioned to each juju version

Why: Juju is complex and changing fast. We need to find information about the version we have in our controllers, but the forum posts are not tied to specific versions.

[Teodora will make this her next priority! =)]

Contact me at: john.doe@example.com

Desire: license-aware systems

What: I would like the Juju ecosystem to show the licenses of the charms: Charmhub showing the licenses in the Charm's page and `juju info` showing the licenses in the output.

Why: in enterprise environments, we have to follow specific rules on what licenses to use. Not having this information excludes juju from providing solutions to enterprise clients.

[SPDX please! =) Or not?]

[Michael J suggested a dedicated workshop for discussing publishing guidelines]

Contact me at:

Desire: All documentation in the charm for charmhub

What: I would like to have the documentation (README.md?) go into the charm itself instead of having to mess with the discourse.

Also, the README.md format would be better off be forced to have some specific chapters to improve the quality of charms in the charmhub.

Why: Its very difficult to contribute and add to documentation and get it all look good in charmhub.

Contact me at:

Desire: A more demystified and monitorable controller

What: I would like controller (cluster) health to be easily monitored and generally more means and best practices to maintain the controller's health over its lifetime, both in terms of scripts/utilities and documentation.

Why: The controller is an extremely critical component used to control almost our entire infrastructure, without it we are blind. It is stable, but after years of operation, dozens of upgrades and hundreds of units running it sometimes misbehaves (slow and resource demanding) and the cluster may even be out of sync without us knowing about it.

Contact:

Desire: More config option classes/types

What: I would like to have a richer set of config options with “early sanity checks”.

So lets say “items” like “apple, pear, orange” would be the only allowed options.

Why: This would “prevent” missconfigurations early which would also help developers not having to do alot of coding to cover for wierd input etc.

... also passwords/secrets option could have a great use.

Desire: A more streamlined CLI command structure

Desire: I would like a more streamlined CLI command structure.

Why: It would improve user experience.

Not obvious whether it's better to have commands that are verbose but clear vs. short but ambiguous. Verbose also helps discovery through grep.

Maybe we should make continuous efforts to translate between the bottom-up, developer perspective and the top-down, user perspective.

Desire: granular k8s api access for charms

What: I believe that currently defining “trust: true” will give the charm access to the same k8s api permission as the user deploying it. It would be nice to see a way to give just the needed permissions to the service account.

Why: if the workload container is exposed to the public and it is hacked, it will have to much access to the kubeAPI