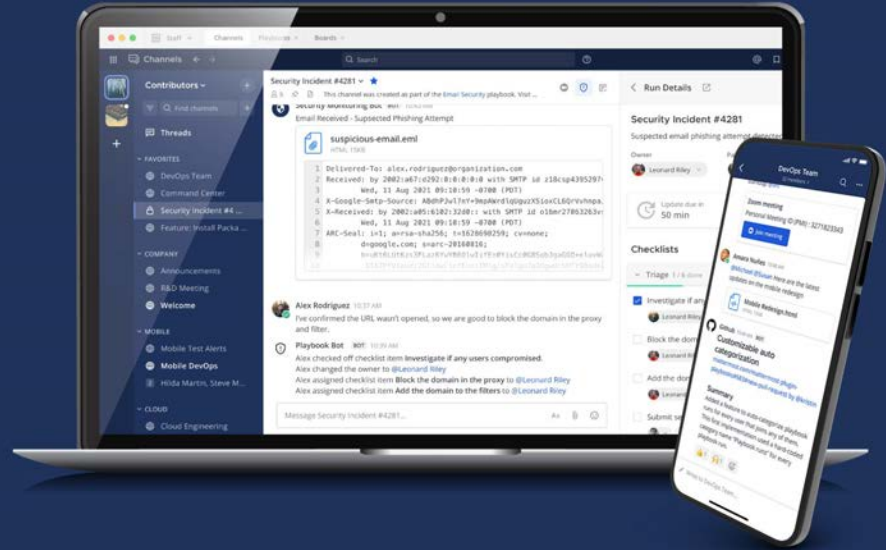


# Building a Platform with self-serve powers using ChatOps and Github Bots

Spiros Economakis

# Mattermost

Mattermost is an open source platform that provides secure collaboration for technical and operational teams.



# Mattermost Cloud story



## Back in the days

We had an idea to offer Mattermost as a SaaS product and build Mattermost Cloud.

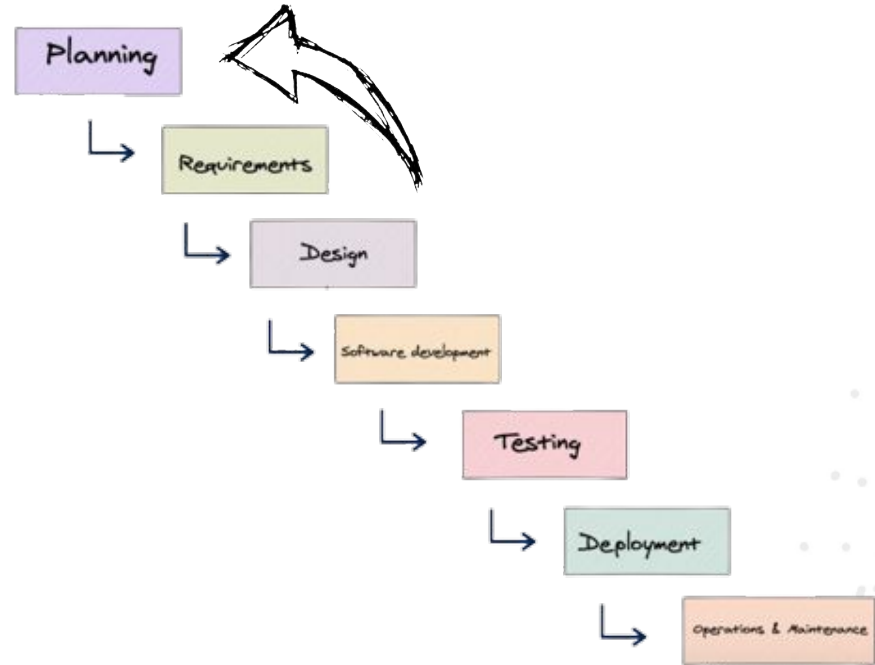


# SDLC of Mattermost Cloud



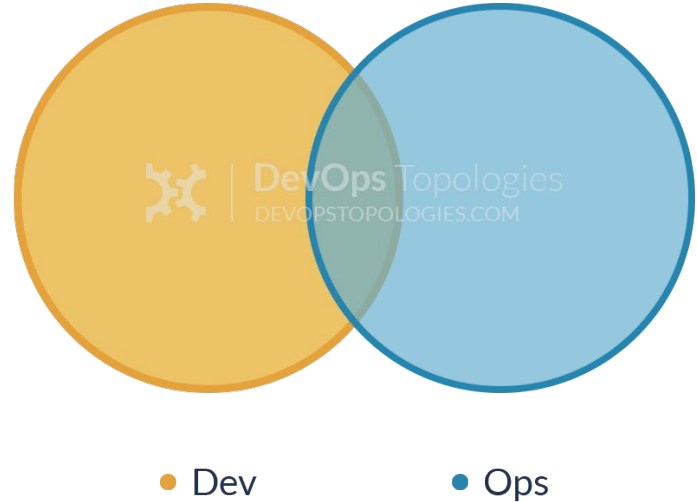
# SDLC - Planning

- Current teams' capacity & resource allocation (incl. hiring etc.)
- Current teams' structure & responsibilities
- Cost
- Milestones



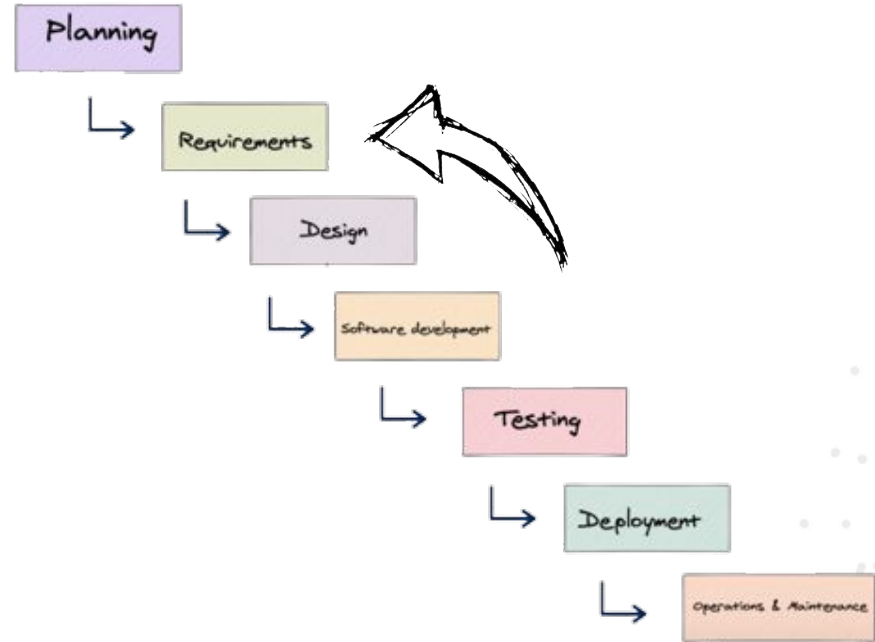
# Team topology

At that time, we had adopted the **Dev and Ops collaboration model**, with each team specializing where needed, but also sharing where necessary.



# SDLC - Requirements

- Ideal Customer Profile (ICP)
- Features, user flows etc.
- SLOs, SLIs
- Incident response
- Scalability & Performance
- Security & Compliance
- more...

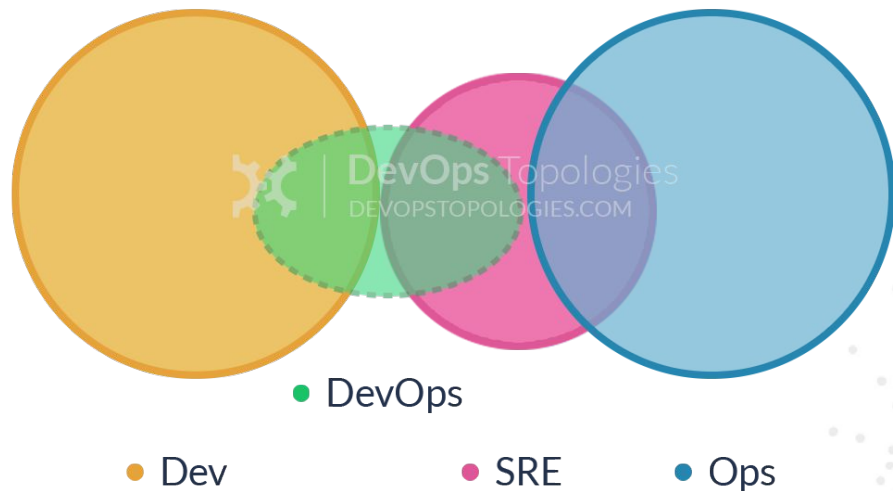




## SRE team

At that time, we realised that we need to introduce an SRE team.

Collaboration between Dev and SRE happens around operational criteria, but once the SRE team is happy with the code, they support it in Production.



# Teams

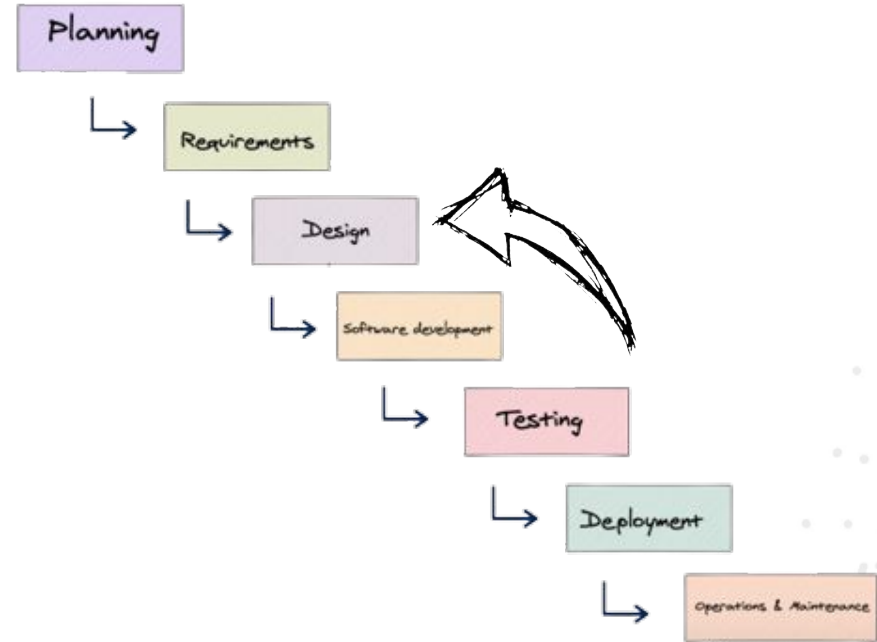


# Strategy



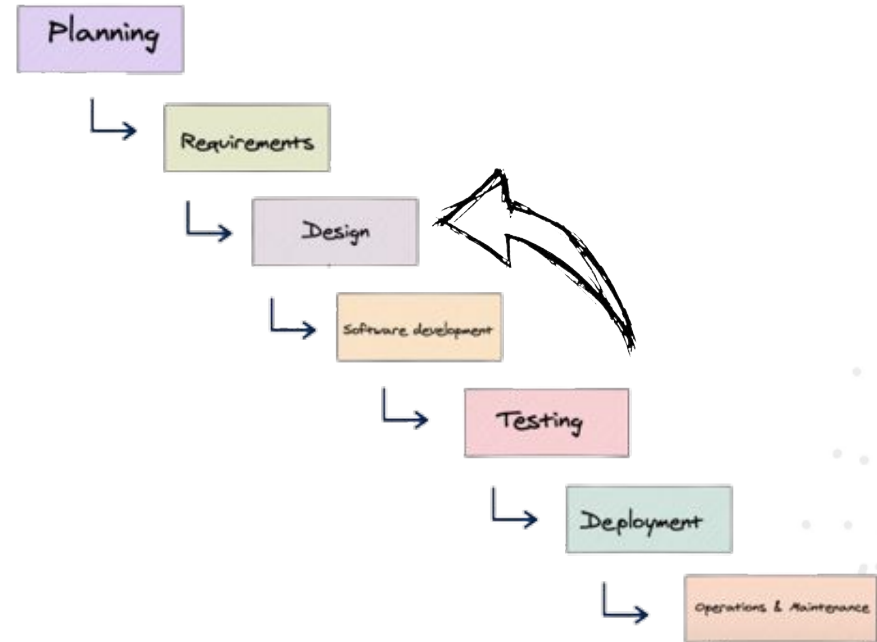
# SDLC - Design

- Technology stack
- Core components
- System architecture
- Security controls
- Compliance & attestations
- Testing strategy

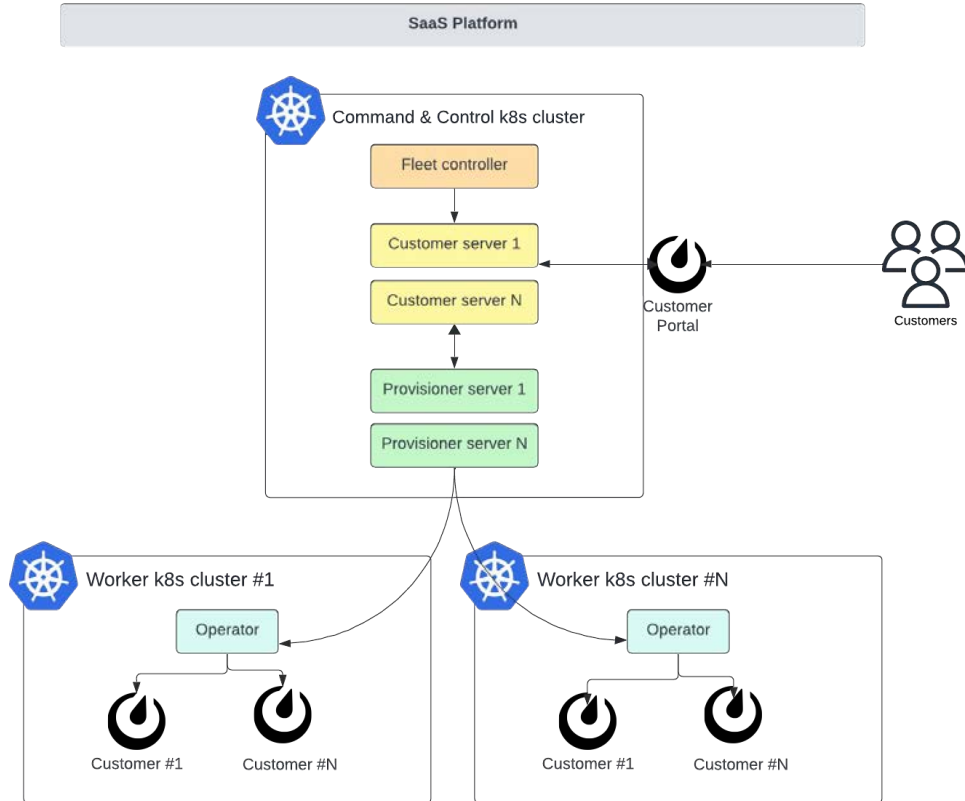


# Design - Some examples

- RDS Aurora
- Kubernetes
- Mattermost Operator
- Customer server
- Customer Portal
- Provisioner
- Fleet controller and more....



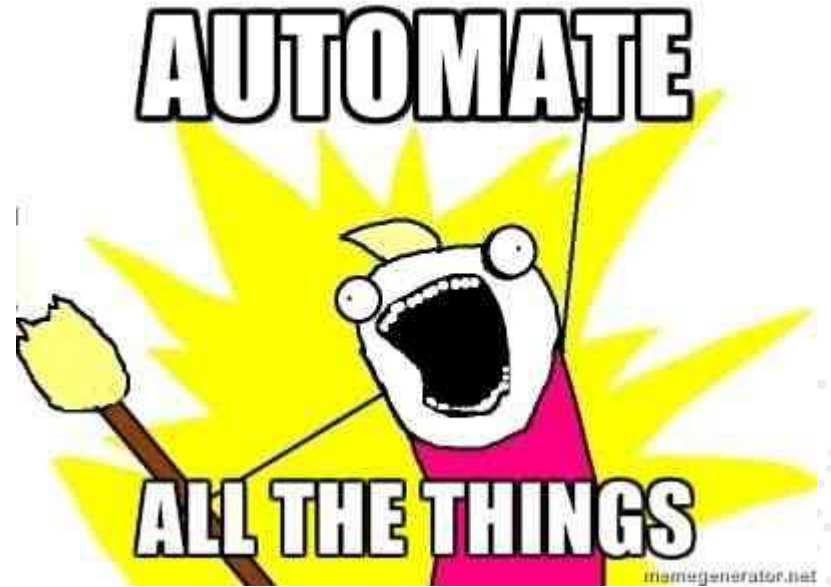
# Design - High level architecture



# Testing strategy

We realized that to test the SaaS platform at scale, it will:

- Need lots of effort
- Be hard to automate everything
- Be cost inefficient
- Increase the time to market



# Dogfooding

One of our core principles in Mattermost is to:

***“Use our own product to complete our mission”***

*SIMPLY EXPLAINED*



*DOGFOODING*





## Our motto

*“Build once and for all”*



## Infrastructure group's mission

Empower Mattermost to provide a SaaS Platform as Product which serves **internal** and **external** users by guaranteeing that we operate an **enterprise-grade** SaaS platform with **self-serve powers**



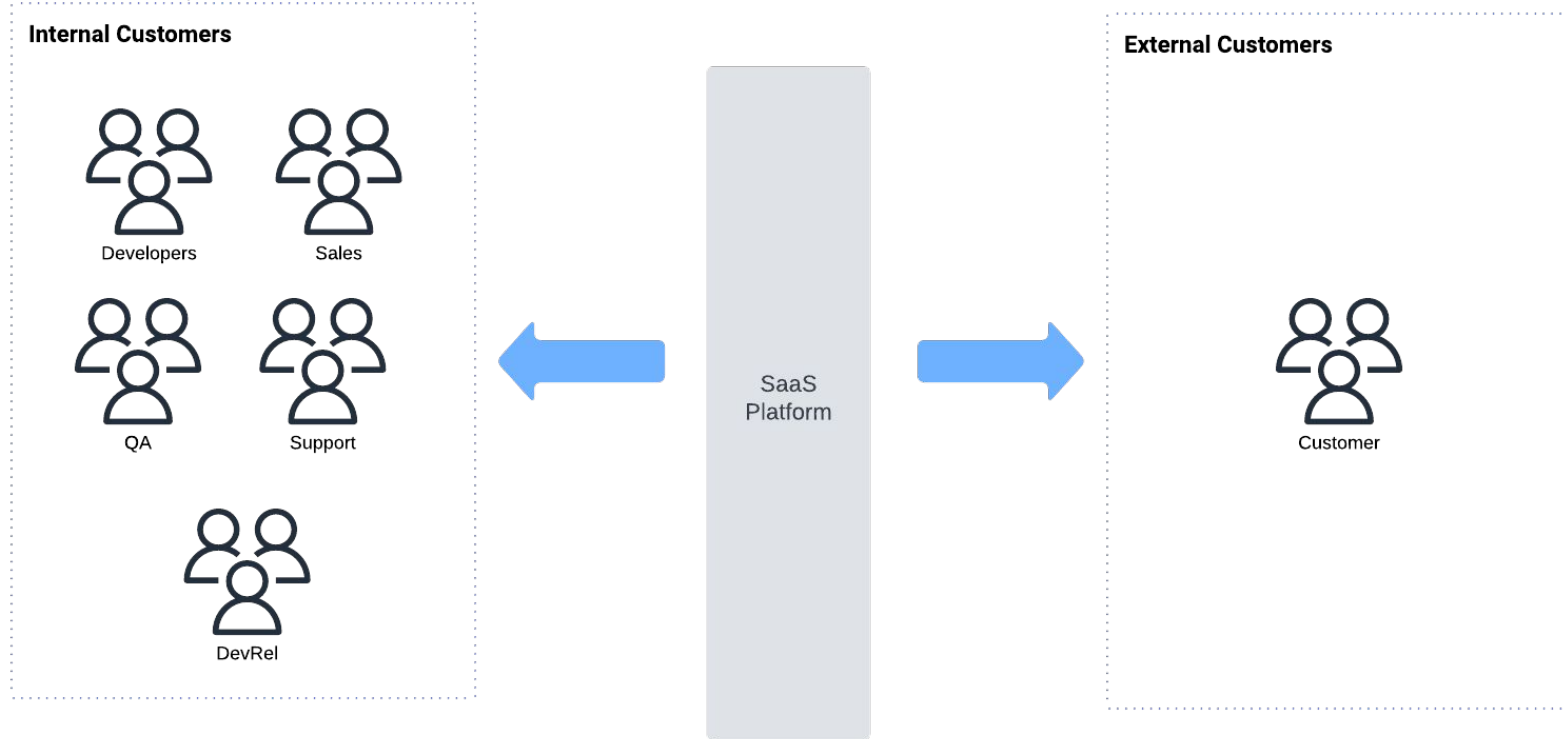
# Testing strategy

- Do not over-automate and prioritise the customer experience and some critical paths for automation
- Use the SaaS Platform as **golden path** to run:
  - Dev envs
  - Support envs
  - Demo envs

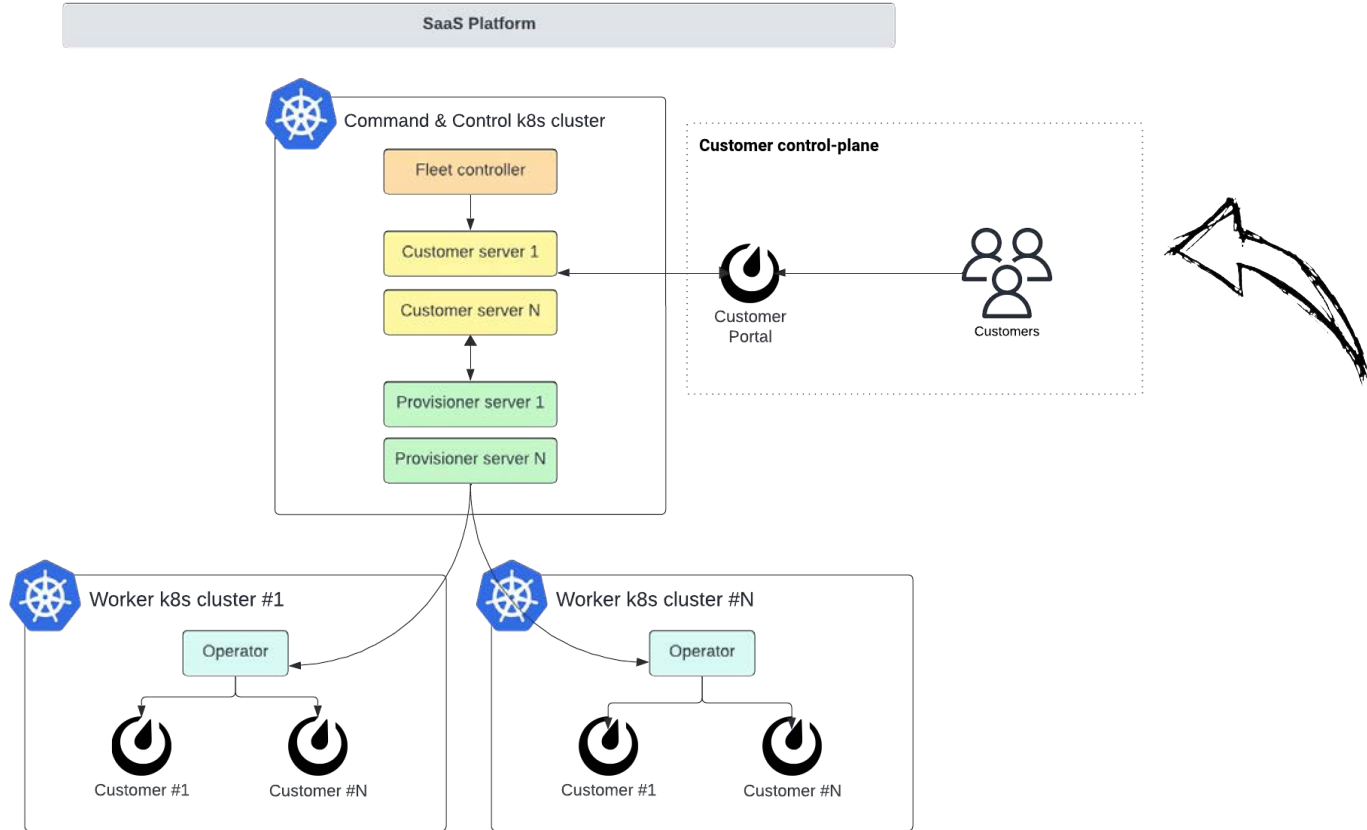


# New Strategy

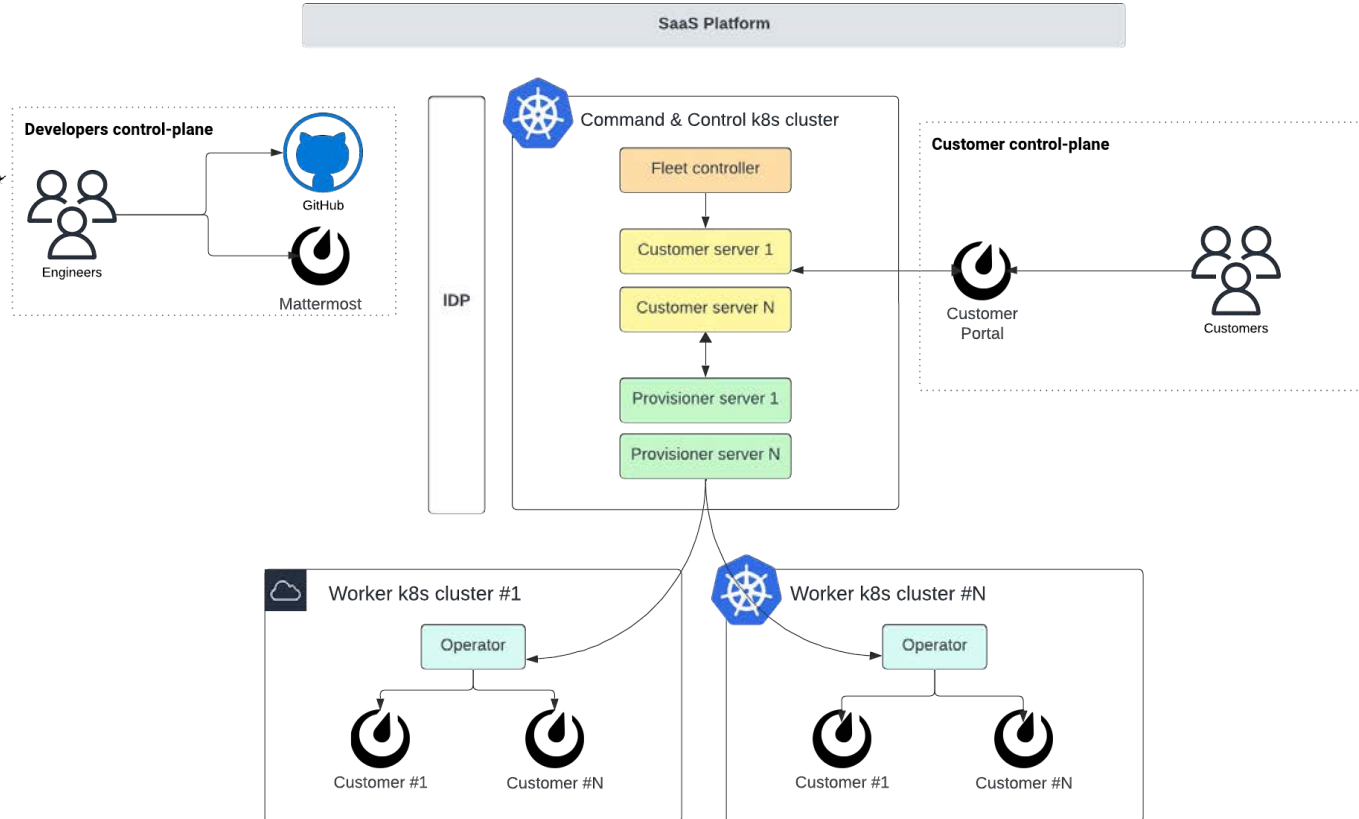
## Early adopters



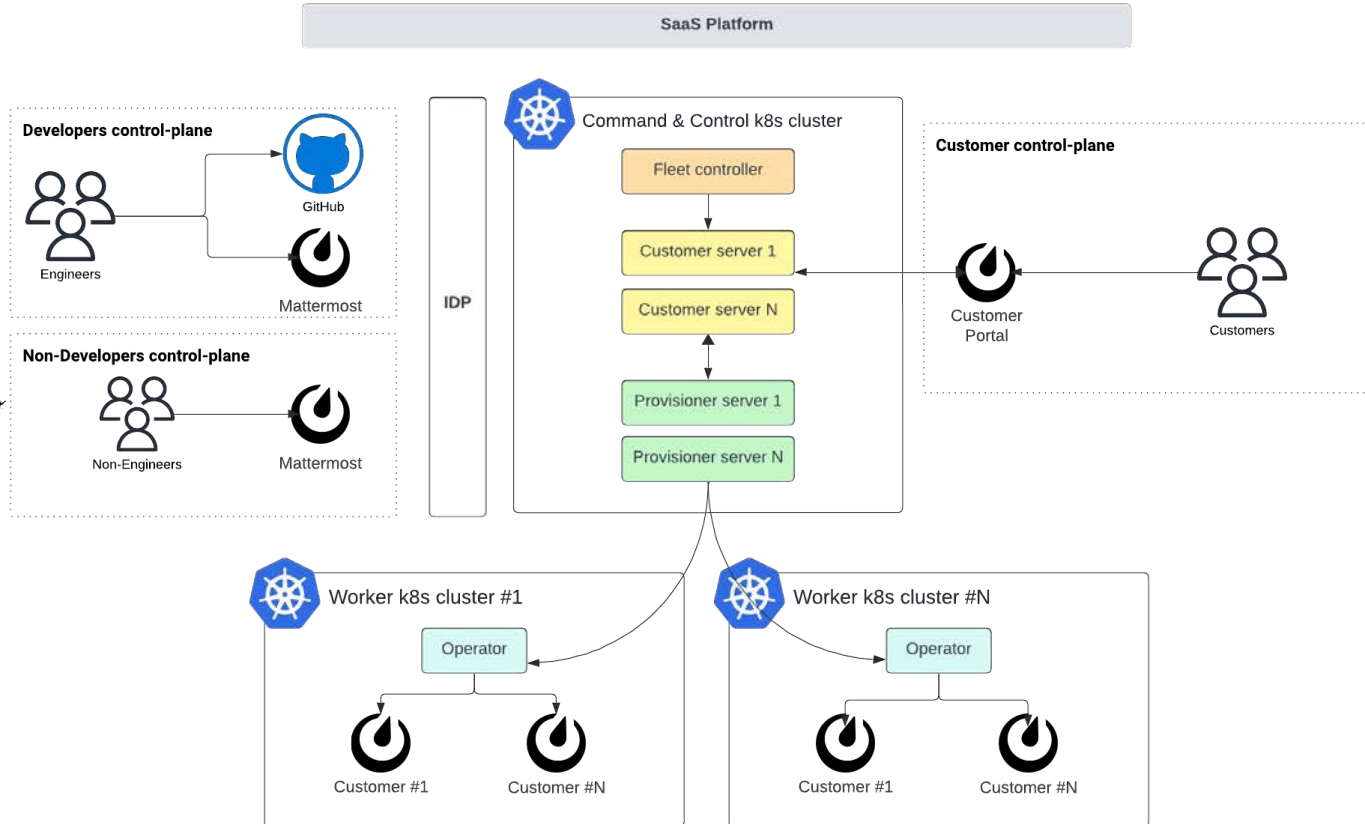
# Design - High level architecture



# Design - High level architecture

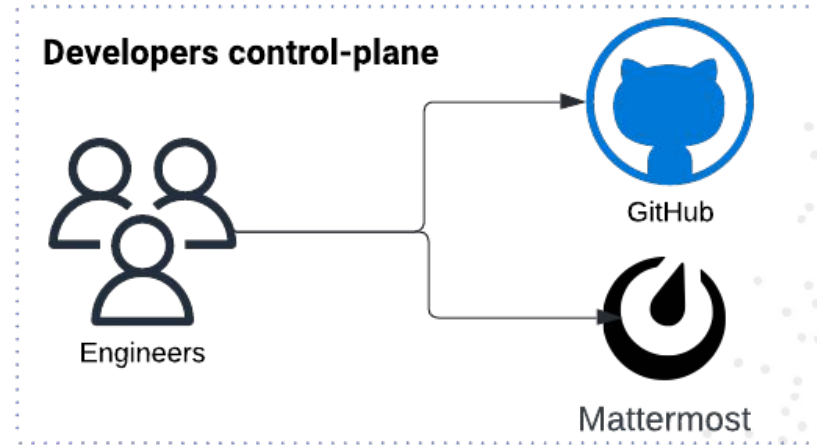


# Design - High level architecture



# Developers control-plane goals

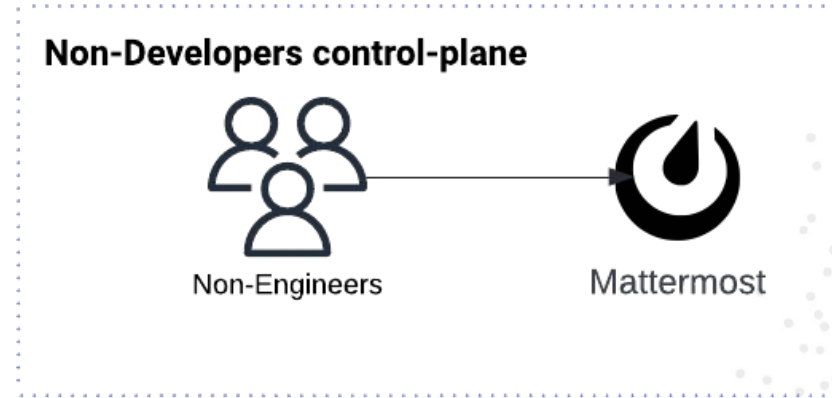
- Self-serve
- Support for Staff and Open Source contributors
- DevEx with the tools we use daily
- Abstraction and SaaS Platform agnostic





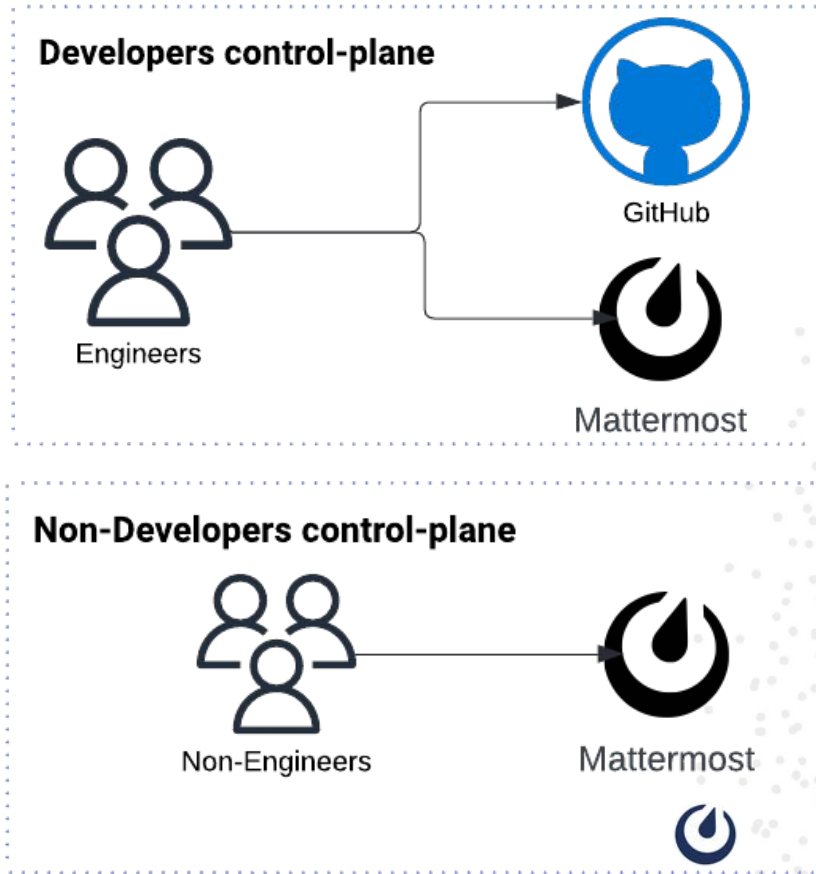
# Non-developers control-plane

- Self-serve
- Support for Sales, Support and DevRel
- UserEx with the tools we use daily
- Abstraction and SaaS Platform agnostic



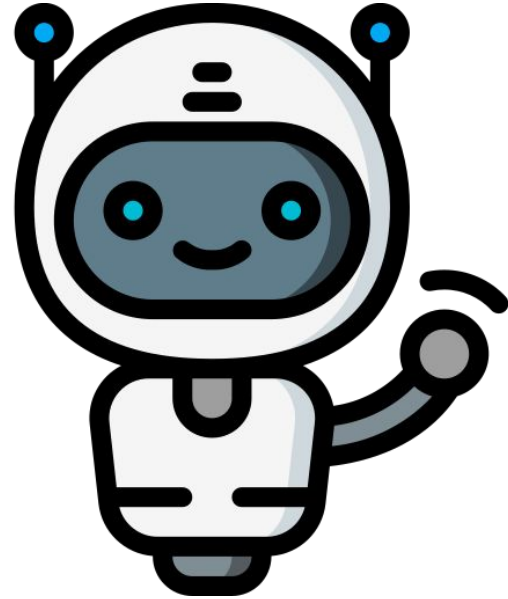
# Seamless experience

- Github
  - Github bots
- Mattermost
  - ChatOps



## Github bots

Github bots or apps are used to automate and improve workflows.

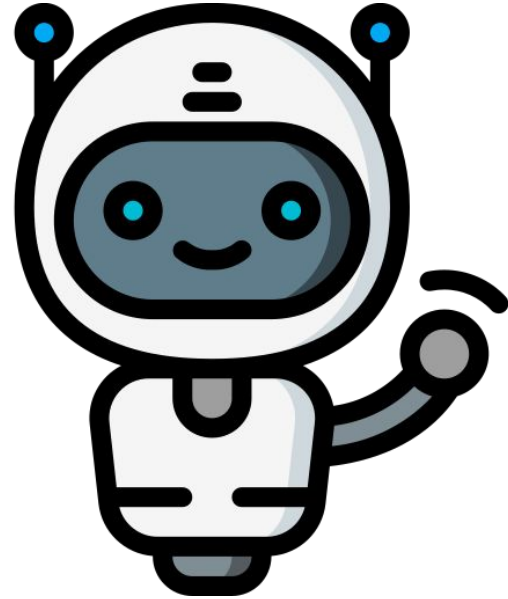


# Spinwick

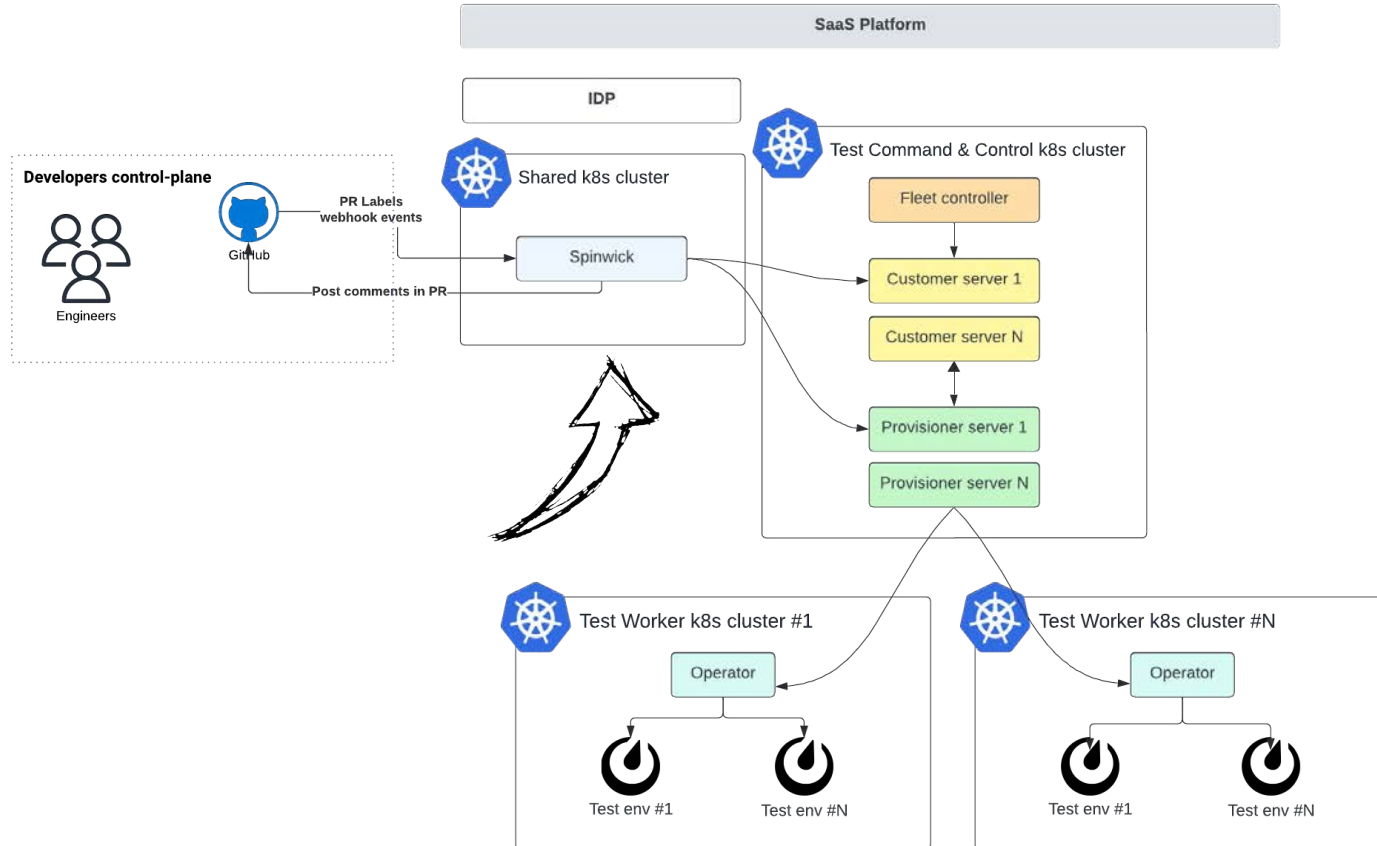
A Github Bot which observes the Github labels and accordingly deploys Mattermost Test servers in Mattermost Cloud.

We can create different type test servers:

- Using Customer Portal
- Using Provisioner



# Design - High level architecture



# Spinwick: Self-serve

[CLD-6219] Add Company Name field to Purchase Modal #24428

Open nickmisaal wants to merge 2 commits into master from CLD-6219

Conversation 1 Commits 2 Checks 22 Files changed 2

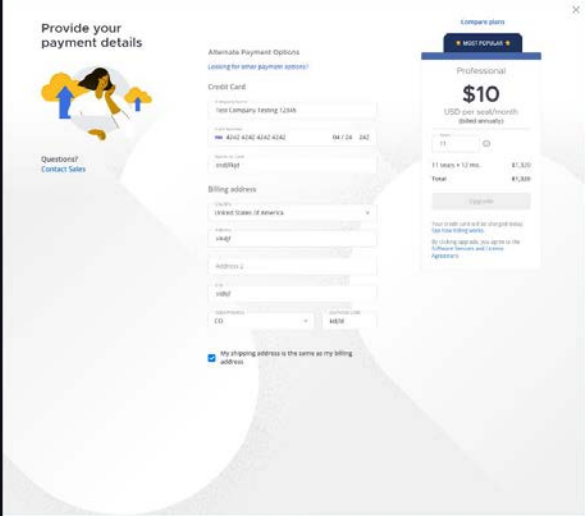
Summary

Add a Company Name field to the purchase modal that's required, so it can be passed to CWS for export compliance screening. This field will be auto populated with any previously entered value, and will update the Stripe customer with the new value.

Ticket Link

<https://mattermost.atlassian.net/browse/CLD-6219>

Screenshots



Reviewers

- JulienTant
- gabrieljackson
- yasserfarasakhan

At least 2 approving reviews are required to merge this pull request.

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

Apply labels to this pull request

set


- Setup Old Test Server  
Triggers the creation of a test server
- ✓ Setup Cloud Test Server  
Setup a test server using Mattermost Cloud
- Setup Upgrade Test Server  
Triggers the creation of a test server and perfo...
- Setup HA Cloud Test Server  
Setup an HA test server using Mattermost C...
- Setup Cloud + CWS Test Server  
Setup a test server linked to the CWS test p...
- 3: Security Review  
Review requested from Security Team
- release-note  
Denotes a PR that will be considered when L...
- ✓ release-note-none  
Create new label "set"


Edit labels

Lock conversation





# Spinwick: Self-serve

 **spirosoik** added the **Setup Cloud Test Server** label 2 minutes ago

 **mm-cloud-bot** commented 2 minutes ago Member ...

Creating a new SpinWick test server using Mattermost Cloud.




 **mm-cloud-bot** commented 1 minute ago Member ...

Mattermost test server created! 🎉

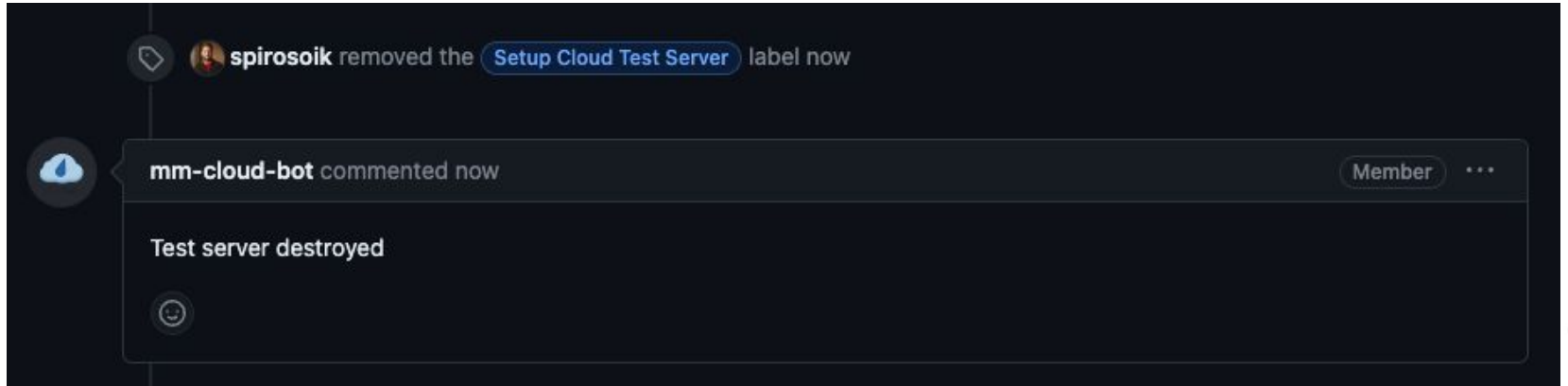
Access here: <https://mattermost-pr-24428.test.mattermost.cloud>

Account Type	Username	Password
Admin	sysadmin	Sys@dmin123
User	user-1	User-1@123





# Spinwick: Self-serve

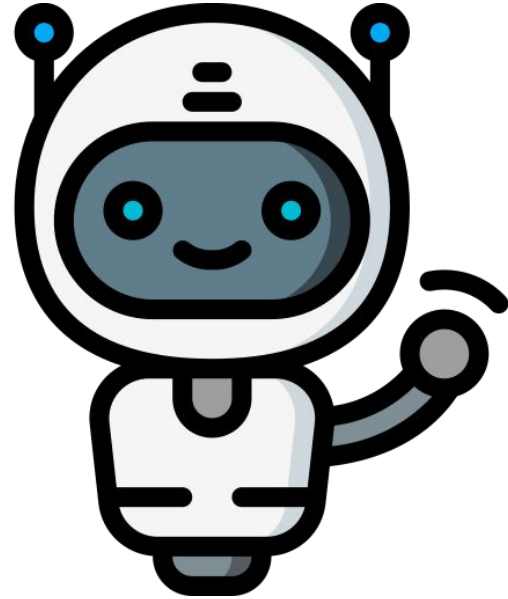




# Mattermod

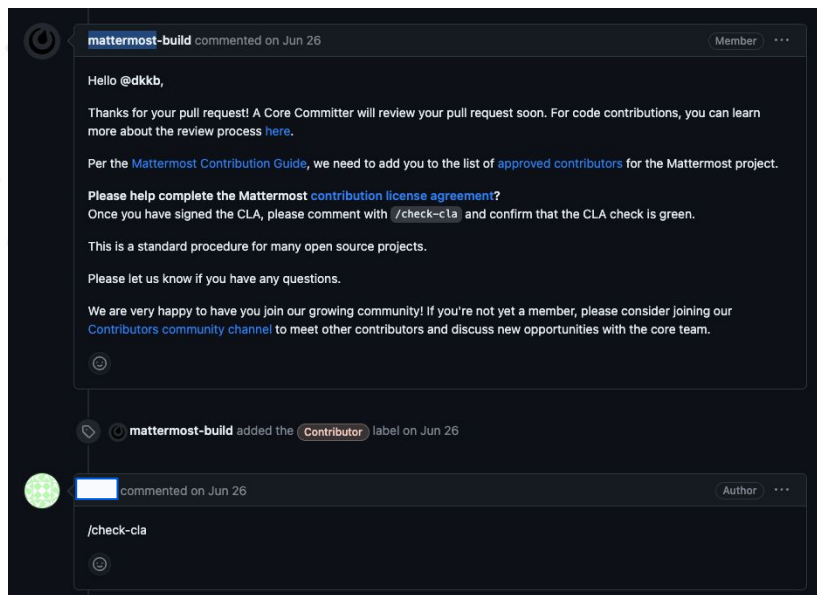
A Github bot to automate day to day workflows in Github:

- Using Labels in a PR
- Using slash commands comments in a PR
- Adding labels for housekeeping into Issues and PRs



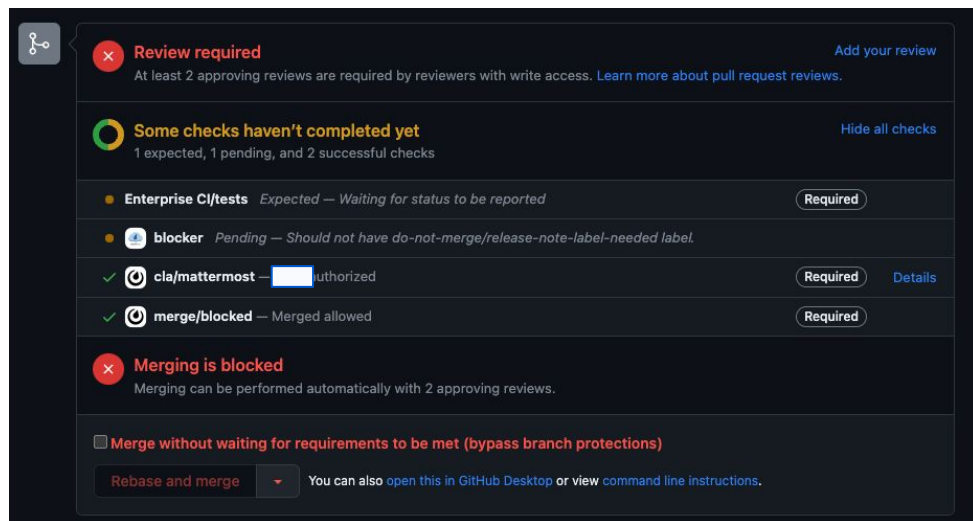
# Mattermod: Self-serve & slash commands

`/check-sla` checks if Open Source contributors have accepted the Mattermost Contributor Agreement



A screenshot of a Mattermost chat window. The top message is from 'mattermost-build' (Member) dated Jun 26. The text of the message reads: 'Hello @dkkb, Thanks for your pull request! A Core Committer will review your pull request soon. For code contributions, you can learn more about the review process [here](#). Per the [Mattermost Contribution Guide](#), we need to add you to the list of [approved contributors](#) for the Mattermost project. Please help complete the Mattermost [contribution license agreement](#)? Once you have signed the CLA, please comment with `/check-cla` and confirm that the CLA check is green. This is a standard procedure for many open source projects. Please let us know if you have any questions. We are very happy to have you have you join our growing community! If you're not yet a member, please consider joining our [Contributors community channel](#) to meet other contributors and discuss new opportunities with the core team.'

The bottom message is from an anonymous user (Author) dated Jun 26, containing the slash command `/check-cla`.



A screenshot of a GitHub pull request status page. The page shows several status indicators:

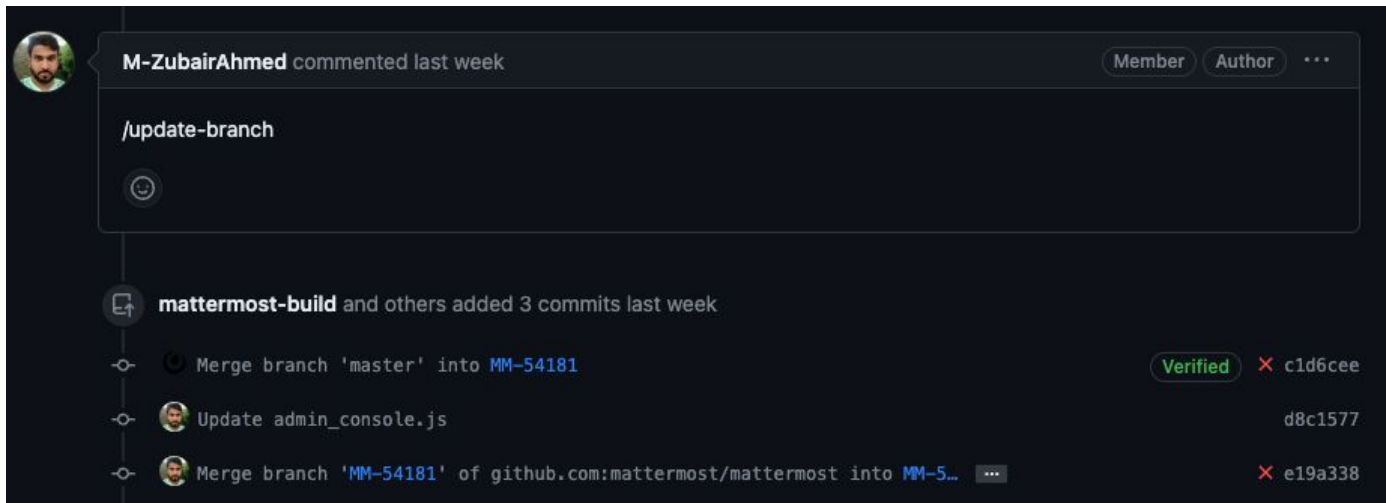
- Review required** (red X icon): At least 2 approving reviews are required by reviewers with write access. [Learn more about pull request reviews.](#) [Add your review](#)
- Some checks haven't completed yet** (yellow circle icon): 1 expected, 1 pending, and 2 successful checks. [Hide all checks](#)
- Enterprise CI/tests** (yellow circle icon): Expected — Waiting for status to be reported. [Required](#)
- blocker** (blue circle icon): Pending — Should not have do-not-merge/release-note-label-needed label.
- cla/mattermost** (green checkmark icon): [redacted] uthorized. [Required](#) [Details](#)
- merge/blocked** (green checkmark icon): Merged allowed. [Required](#)
- Merging is blocked** (red X icon): Merging can be performed automatically with 2 approving reviews.
- Merge without waiting for requirements to be met (bypass branch protections)**

At the bottom, there is a 'Rebase and merge' button and a note: 'You can also [open this in GitHub Desktop](#) or [view command line instructions](#).'



# Mattermod: Self-serve & slash commands

`/update_branch` updates a PR with latest branch which is targeted to merge. Eg. if a PR targets `main` will merge latest `main`



The screenshot shows a GitHub pull request interface. At the top, a comment by M-ZubairAhmed is displayed, containing the slash command `/update-branch`. Below the comment, the commit history for the pull request is shown, including a merge of 'master' into 'MM-54181' and an update to 'admin\_console.js'.

M-ZubairAhmed commented last week Member Author

`/update-branch`

**mattermost-build** and others added 3 commits last week

- Merge branch 'master' into MM-54181 Verified ✗ c1d6cee
- Update admin\_console.js d8c1577
- Merge branch 'MM-54181' of github.com:mattermost/mattermost into MM-5... ✗ e19a338



# Mattermod: Self-serve & slash commands

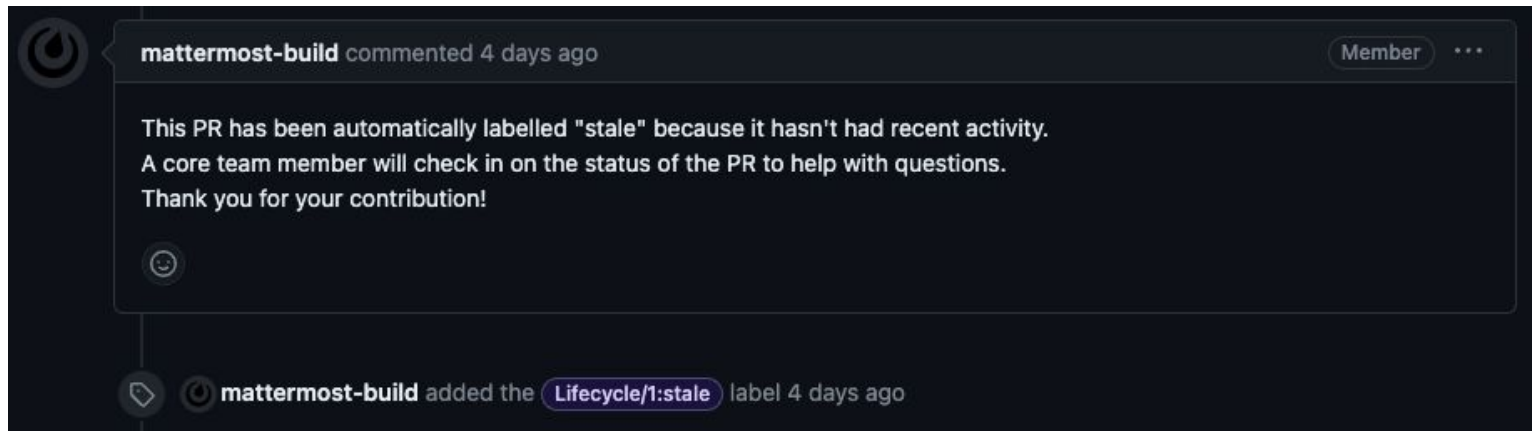
**/cherry-pick** give you the ability to cherry-pick the PR to another branch and raise a PR. Eg. assuming that I want to cherry-pick a PR to the **release-8.0** branch so it can be included in the next release: **/cherry-pick release-8.0**

**/e2e-test** run the e2e tests which is triggering a pipeline for running e2e tests

**/e2e-cancel** cancels the running e2e tests




# Mattermod: Housekeeping




A screenshot of a Mattermost interface showing a comment and a label addition. The comment is from the user 'mattermost-build' and is dated '4 days ago'. The text of the comment reads: 'This PR has been automatically labelled "stale" because it hasn't had recent activity. A core team member will check in on the status of the PR to help with questions. Thank you for your contribution!'. Below the comment, there is a reaction icon (a smiley face). At the bottom of the screenshot, there is a label addition from 'mattermost-build' dated '4 days ago', which is 'Lifecycle/1:stale'. The label is highlighted with a purple border. The interface includes a 'Member' button and a three-dot menu icon in the top right corner of the comment box.

**mattermost-build** commented 4 days ago Member ...

This PR has been automatically labelled "stale" because it hasn't had recent activity. A core team member will check in on the status of the PR to help with questions. Thank you for your contribution!



 **mattermost-build** added the **Lifecycle/1:stale** label 4 days ago



# ChatOps

ChatOps is a collaboration model that connects people, tools, process, and automation into a transparent workflow.

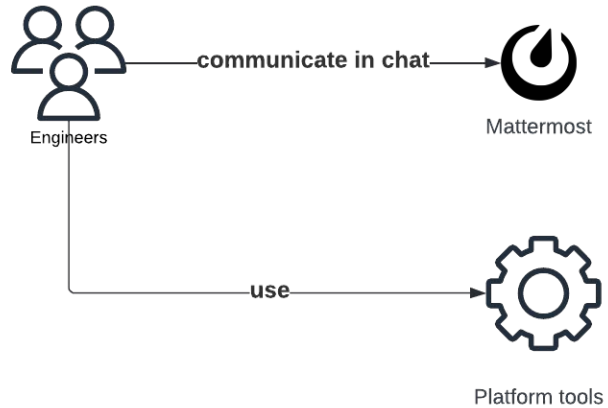


**BOT**

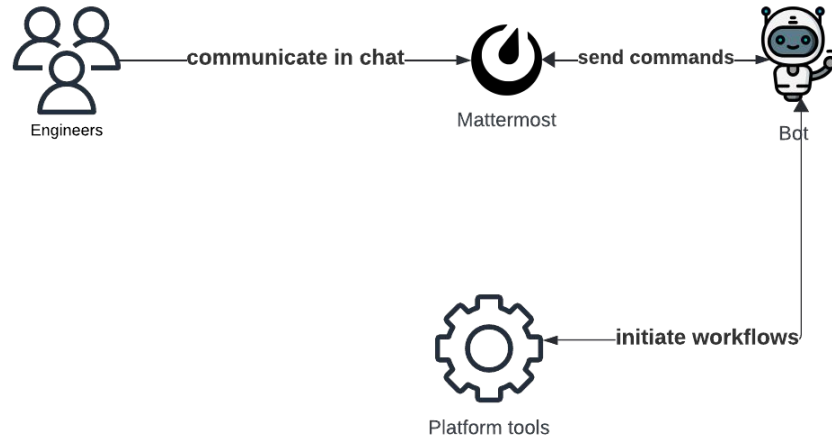
**Hello, friend!**



# Without ChatOps



# With ChatOps





# ChatOps with Mattermost

Mattermost offers a bunch of options to add functionality and customize the experience with ChatOps:

- [Slash Commands](#)
- [Plugins](#)



**BOT**

Hello, friend!



# Cloud Bot Plugin

The Cloud Bot allows the creation and the management of a test env in the SaaS Platform directly from Mattermost using slash commands:

- From any channel
- DM to the bot



# Cloud Bot: Example

`/cloud create test-con42`




**cloud BOT** 9:13 AM (Only visible to you)

Command invoked: `/cloud create test-con42`

Installation being created. You will receive a notification when it is ready. Use `/cloud list` to check on the status of your installations.



# Cloud Bot: Example

 **Cloud BOT** 9:14 AM  
Installation test-con42 is ready!

Access at: <https://test-con42.test.mattermost.cloud>

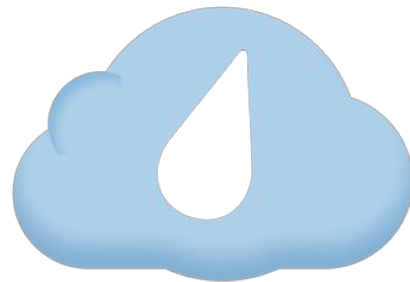
Login with:

Username	Password	Note
admin	Sys@dmin123	Admin user
user	User@123	Regular user

Grafana logs for this installation:

- [Installation logs](#)
- [Provisioner logs](#)

Installation details:

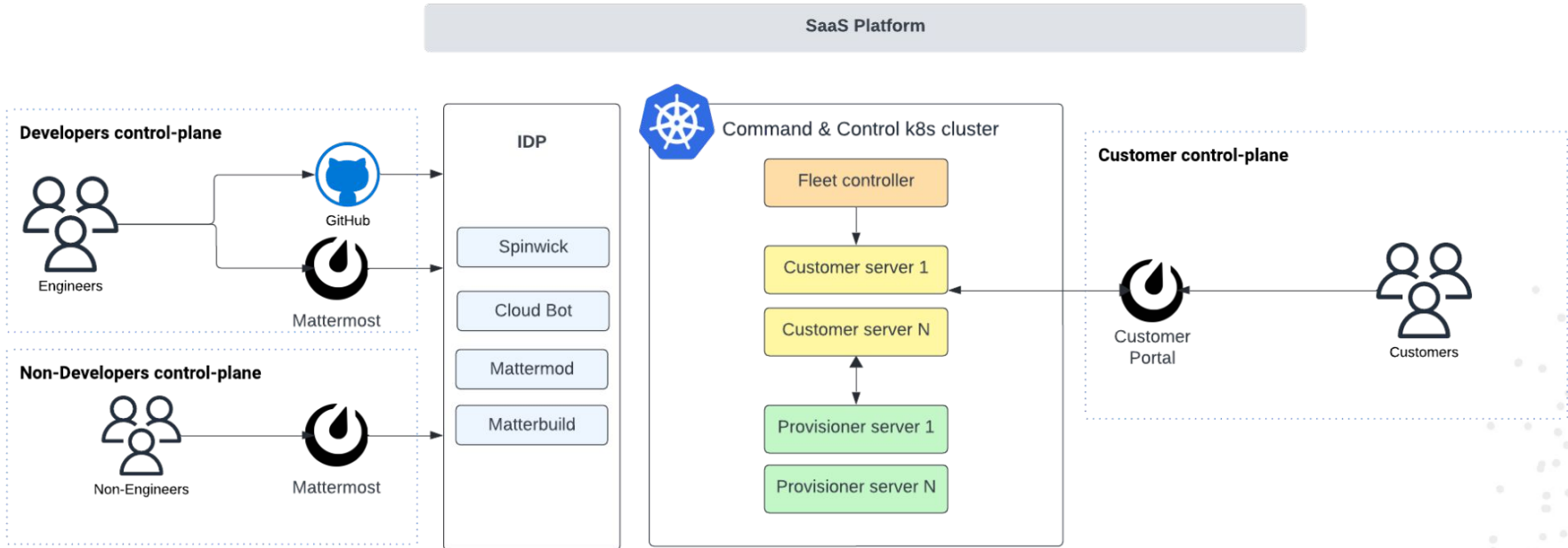


# Cloud Bot: Configuration

```
/cloud create cloudweek --license e20 --test-data --size miniHA  
--filestore bifrost
```



# Internal Developer Platform



# Learnings

- Flexibility & Re-usability
- Developer & User Experience needs to be seamless
- Run surveys, gather feedback and listen your internal and external customers
- User our own product



**Q&A**





**Thanks**

