

Egypt's First Managed Cloud  
Hosting Platform





# TEAM

**Ahmed Mohamed Atwa 2001391**

**Nassar Khaled Mesoud 2001464**

**Ahmed Ayman AbdElFatah 2000128**

**Sameh Ossama Nabil Nazir 2000379**

**Sara Ashraf Abdelhakam 2000337**

**Zeyad Waleed Amin 2000447**

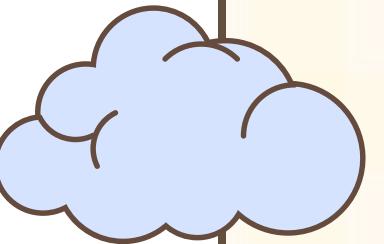
**Somaya Ayman El Sayed Ahmed 2000423**

**Abdelrahman Elsayed Ahmed 2002139**

**Mennatallah Amr Ali 2002111**

**Fady Adel Botros Awdallah 2001388**

**Ahmed Sherif Mohamed 2001547**





# AGENDA

01

**INTRODUCTION**

02

**ARCHITECTURE**

03

**HOSTING PROCESS**

04

**FEATURES**

05

**DEMO**

06

**OUTRO**



# INTRODUCTION

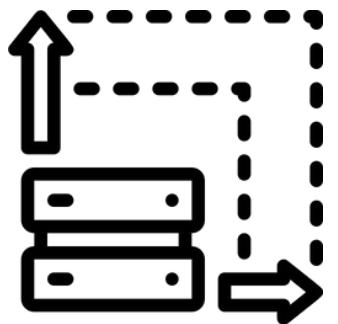
# BACKGROUND

## Cloud Computing

- Delivery of computing services over the internet. (servers, storage, networking, software)



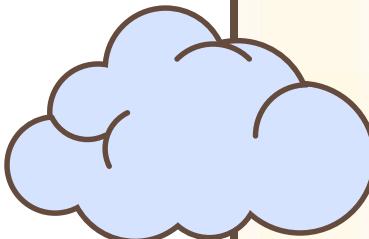
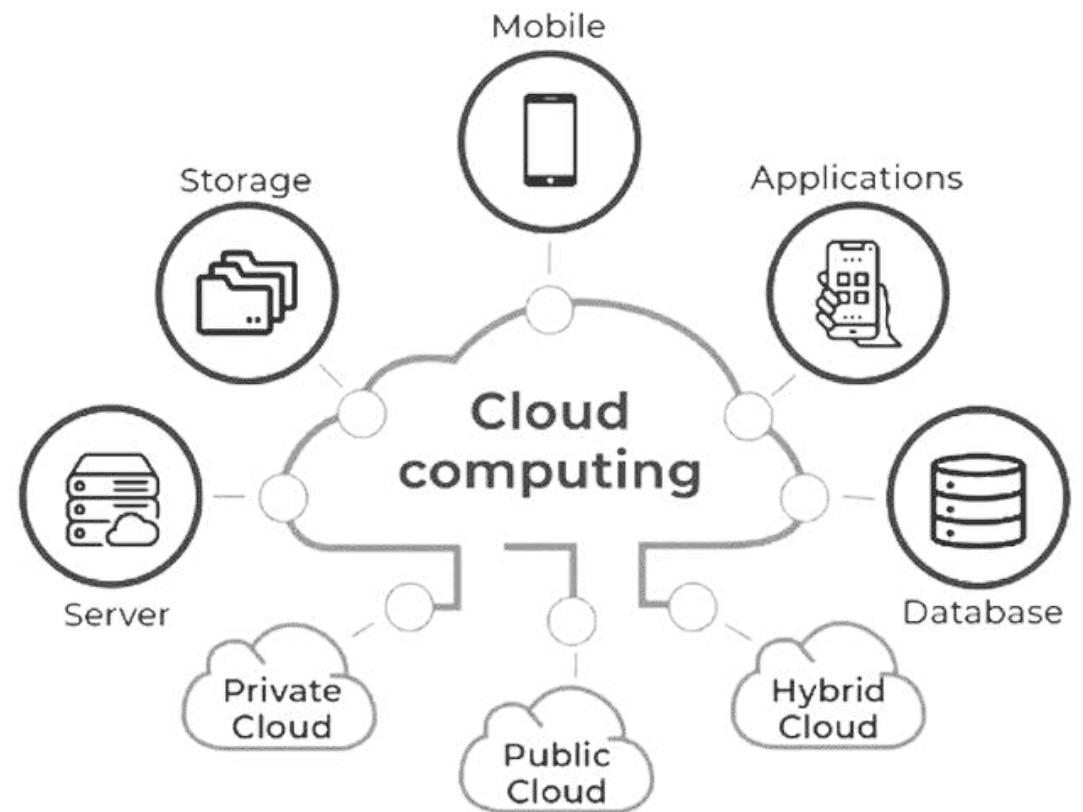
**On Demand**



**Scalable**



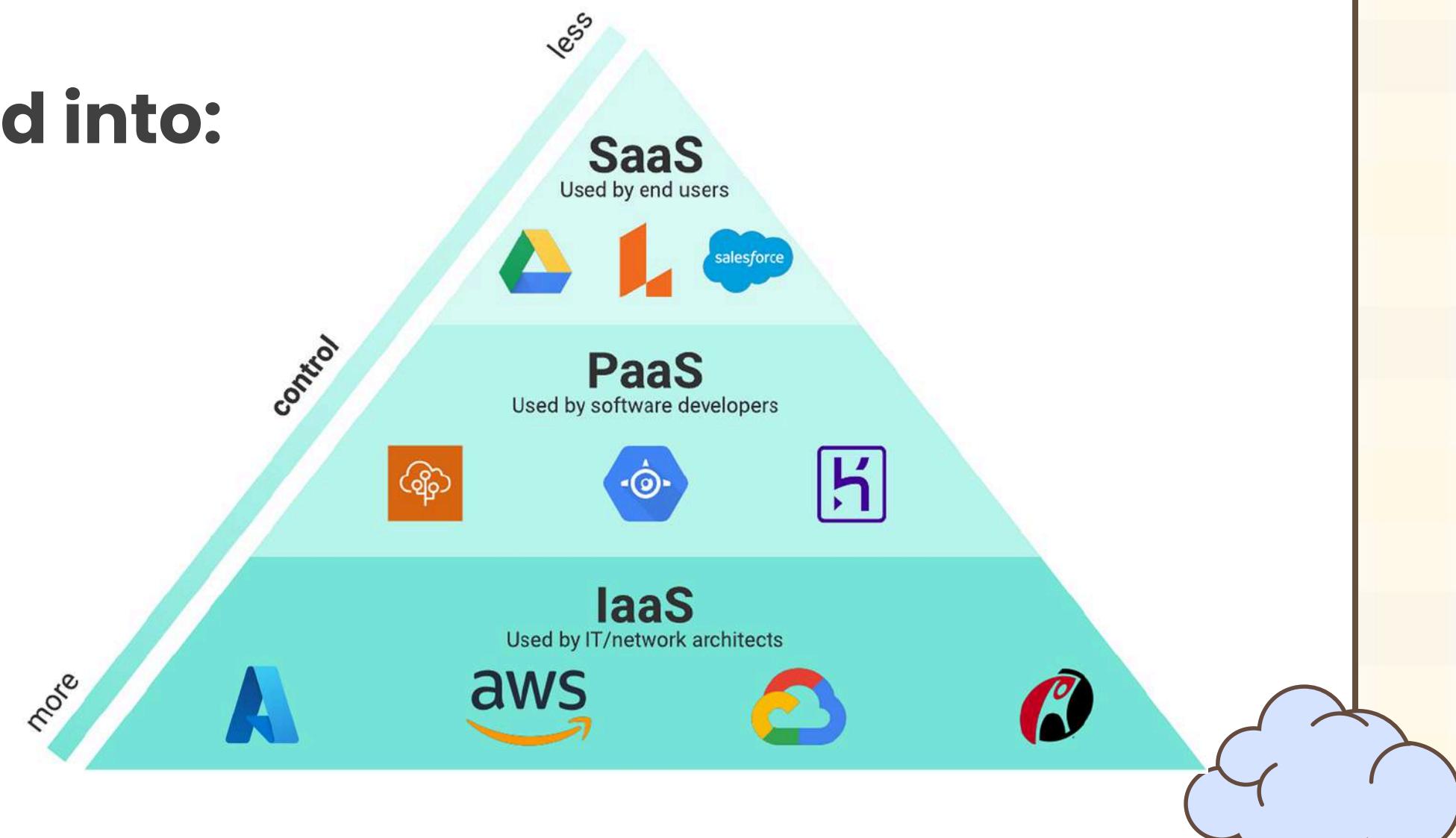
**Pay as You Go**



# CLOUD COMPUTING

**Cloud computing is categorized into:**

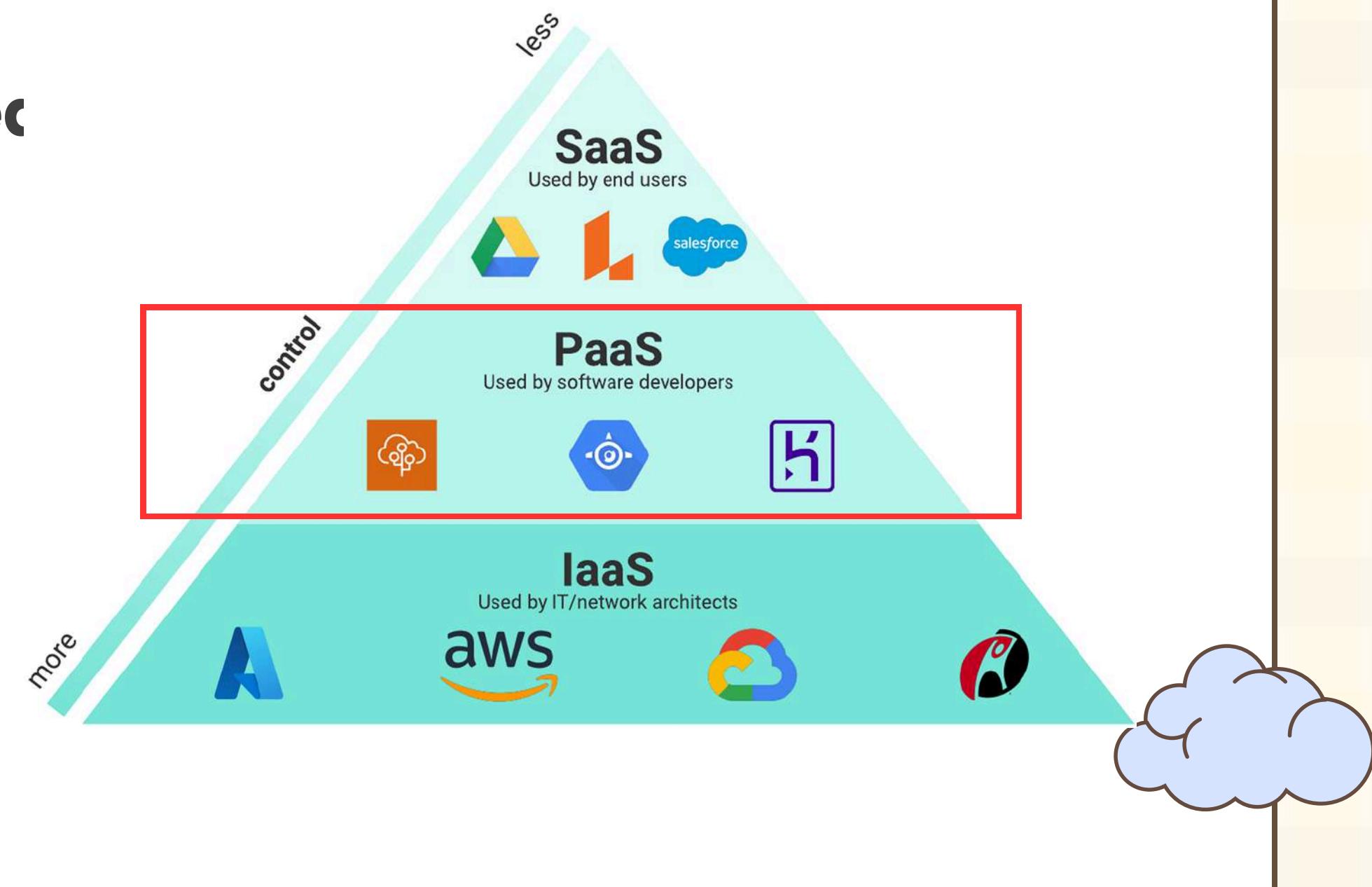
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)



# CLOUD COMPUTING

## Cloud computing is categorized

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)



# MANAGED CLOUD HOSTING

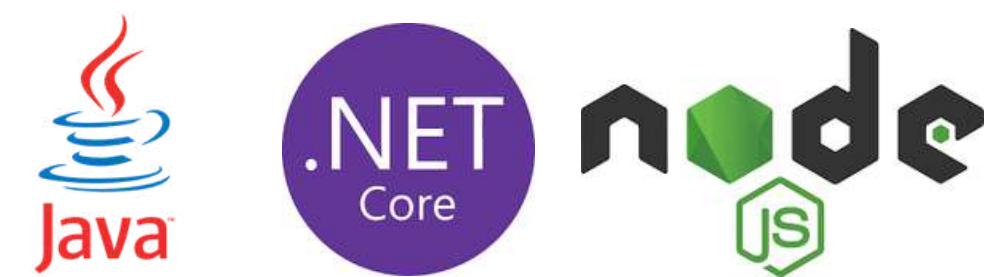
**Cloud computing model that provides a comprehensive platform for developers to build, deploy, and manage applications.**

## Types of Managed Hosting Services

- Static Hosting



- Dynamic Hosting





# TIMELINE

**Time span**  
1/8/2024 – 1/11/2024

**Preliminary Research Milestone**

**Time span**  
16/11/2024 – 27/12/2024

**Static Hosting Implementation Phase**

**Time span**  
1/2/2025 – 14/3/2025

**Dynamic Hosting Milestone**

**Time span**  
29/3/2025 – 16/5/2025

**Testing and Optimization Milestone**

**Time span**  
29/3/2025 – 16/5/2025

**Documentation Milestone**

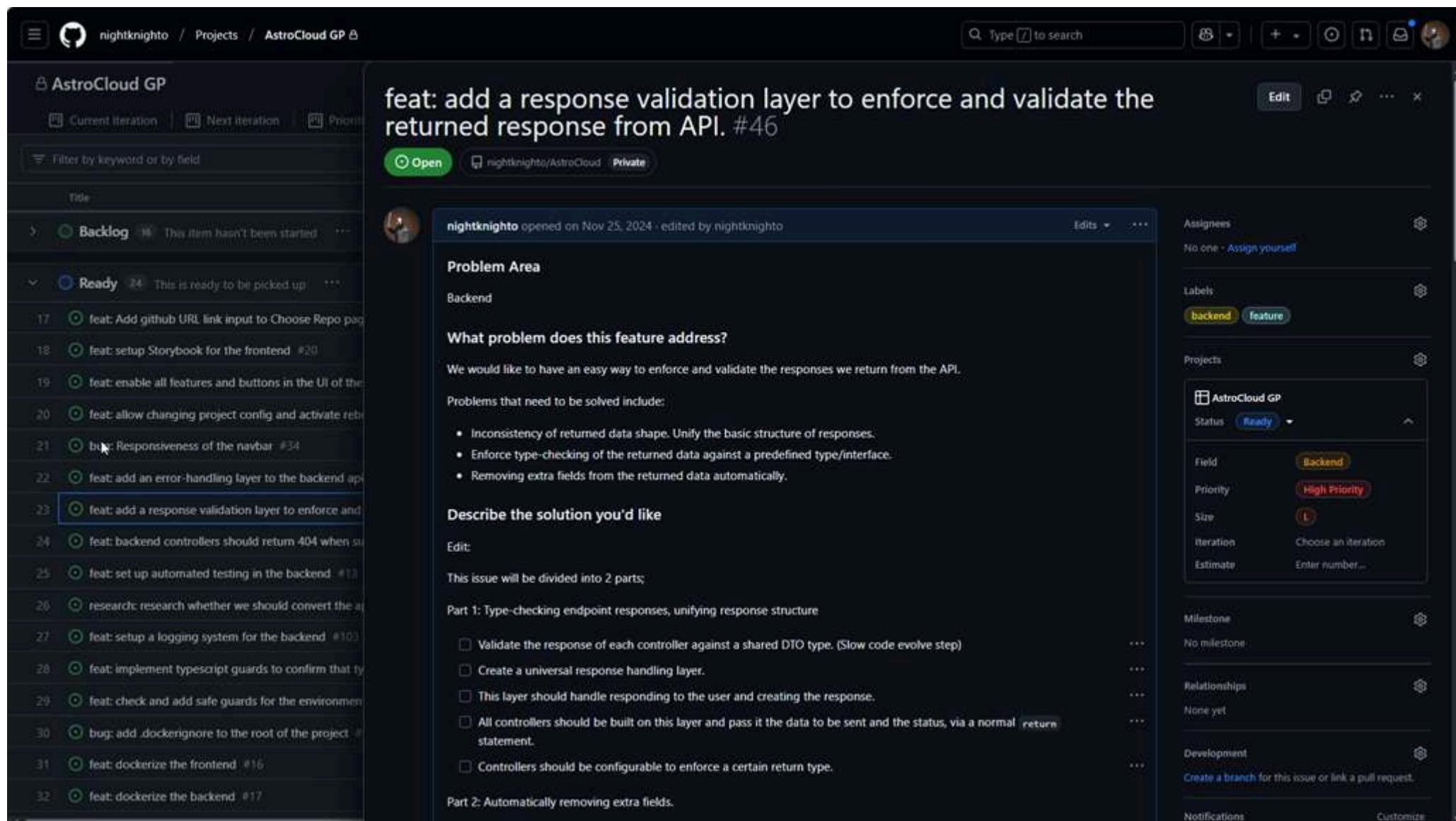
**Time span**  
After 21/6/2025

**Final Review**

# PROJECT MANAGEMENT

## GitHub-centric Approach

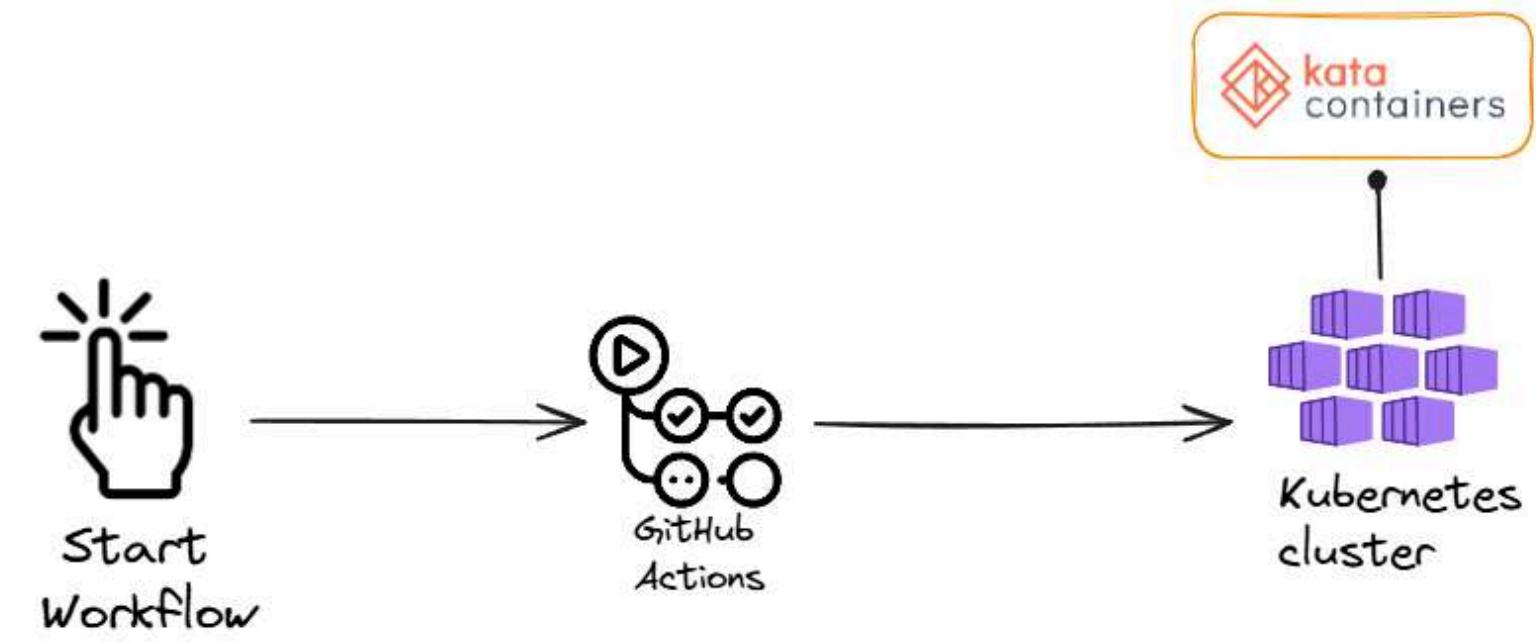
- GitHub Issues
- GitHub Pull Requests
- GitHub Project Board



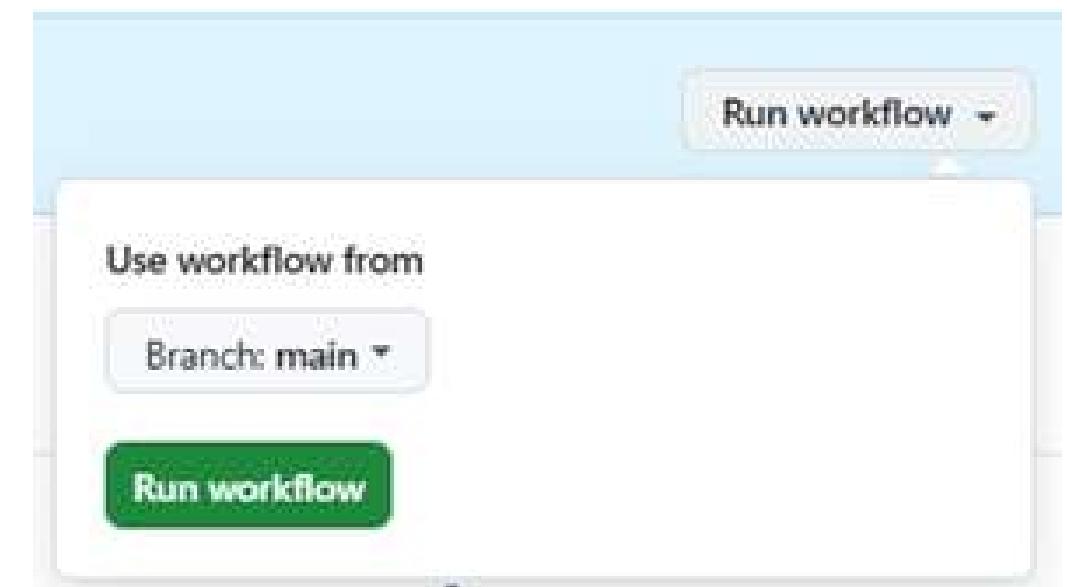


# DEPLOYMENT WORKFLOW

# DEPLOYMENT WORKFLOW



# DEPLOYMENT WORKFLOW



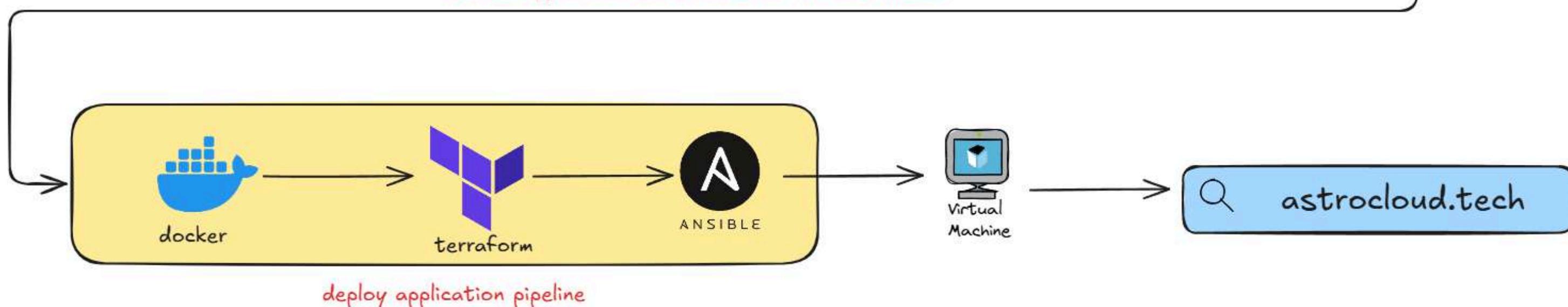
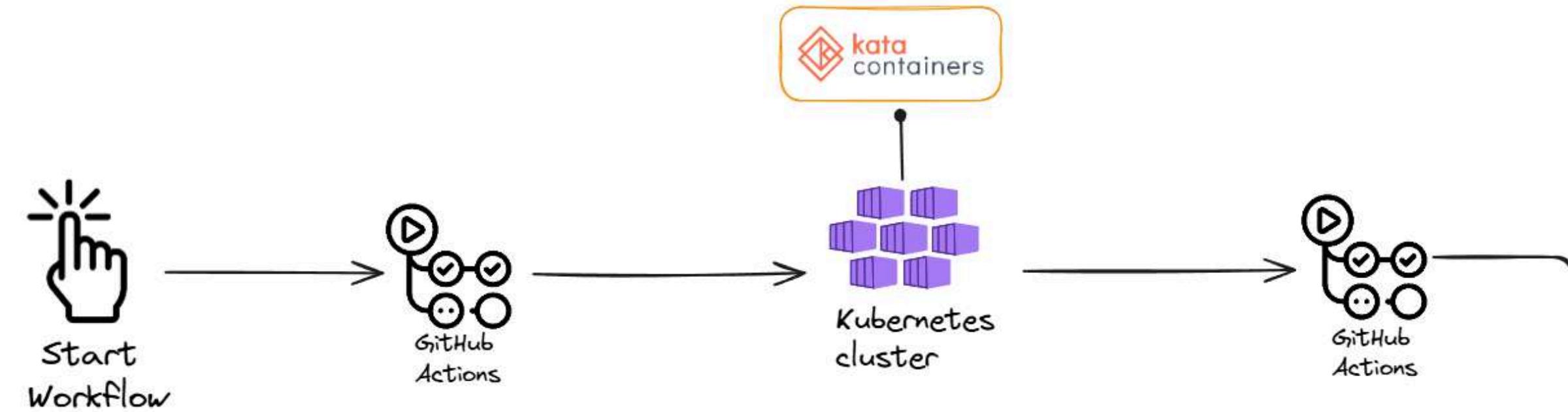
# DEPLOYMENT WORKFLOW



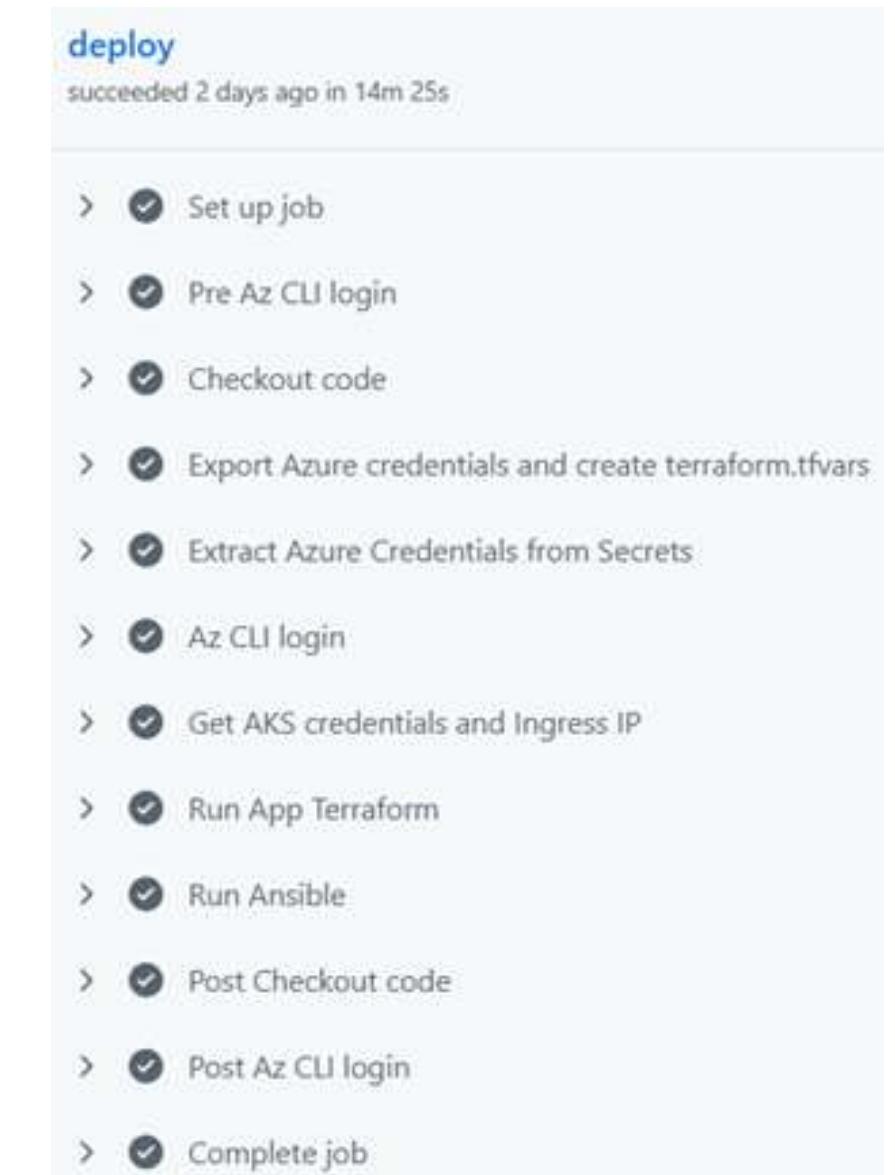
```
deploy
succeeded 2 days ago in 4m 32s

> ✓ Set up job
> ✓ Checkout code
> ✓ Install latest Azure CLI
> ✓ Extract Azure Credentials from Secrets
> ✓ Az CLI login
> ✓ Create Resource Group
> ✓ Create AKS Cluster with Kata Containers
> ✓ Post Az CLI login
> ✓ Post Checkout code
> ✓ Complete job
```

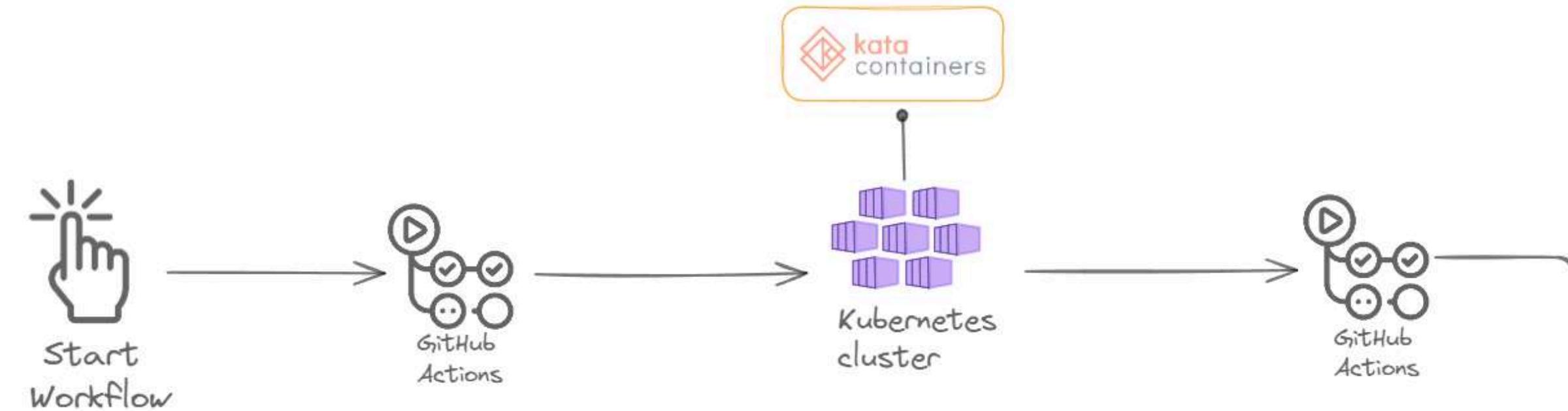
# DEPLOYMENT WORKFLOW



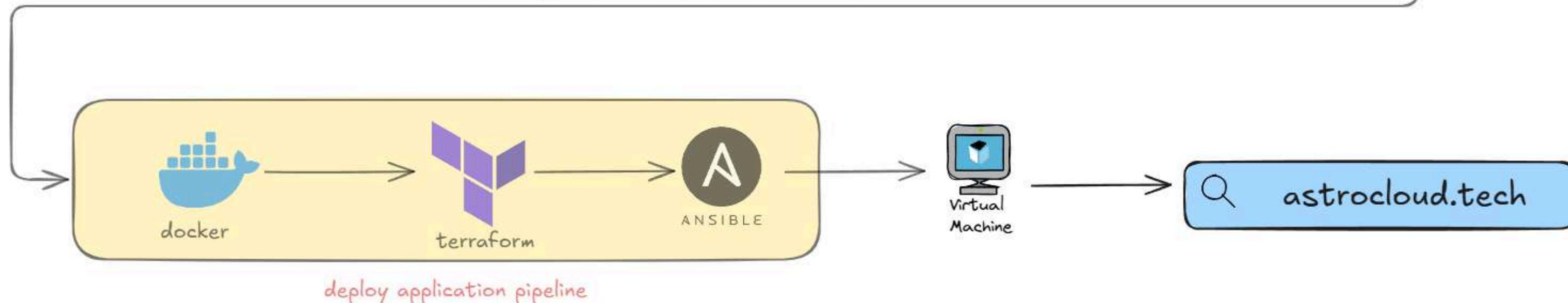
# DEPLOYMENT WORKFLOW



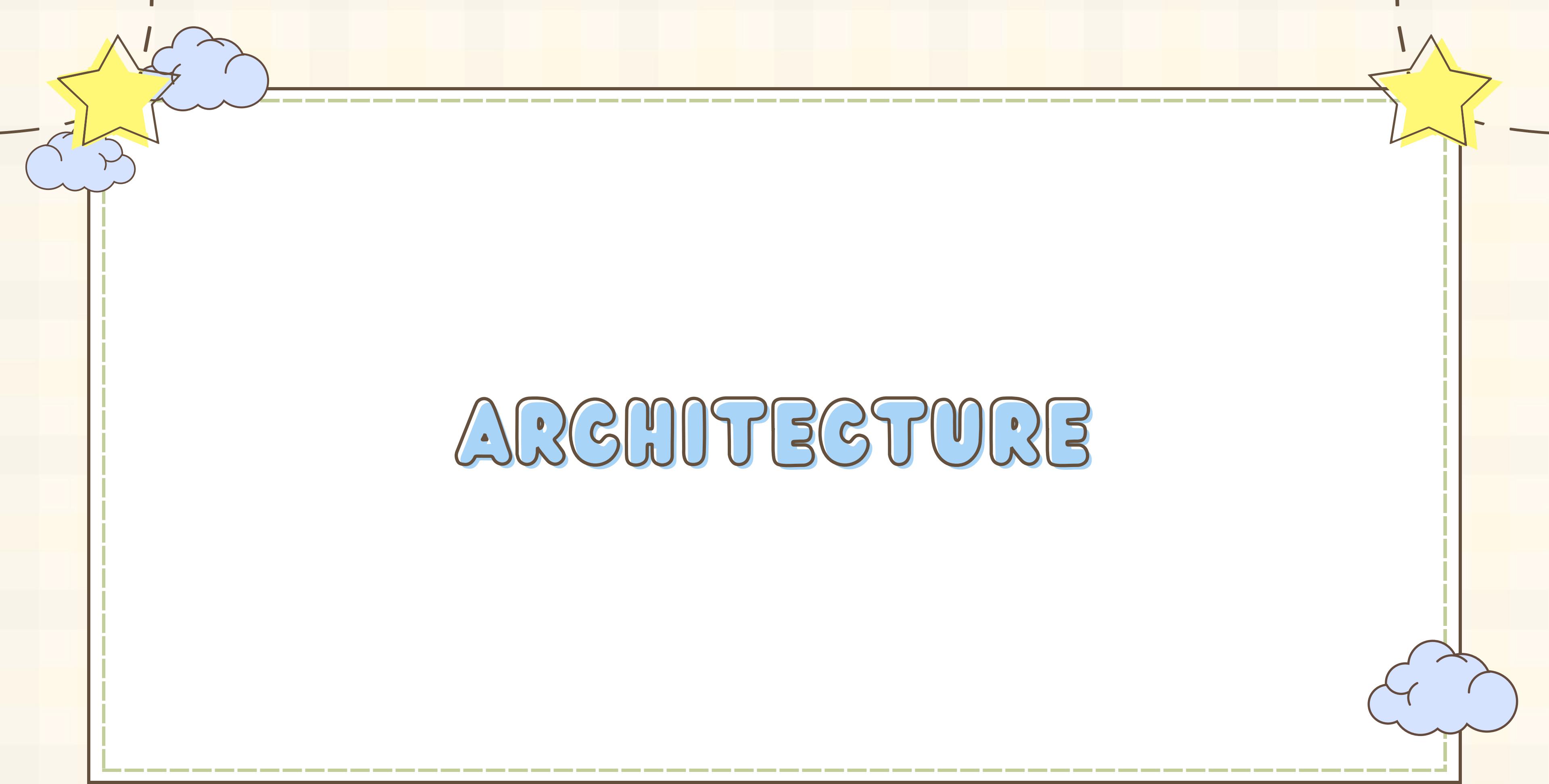
# DEPLOYMENT WORKFLOW



Create application and connect it to the cluster

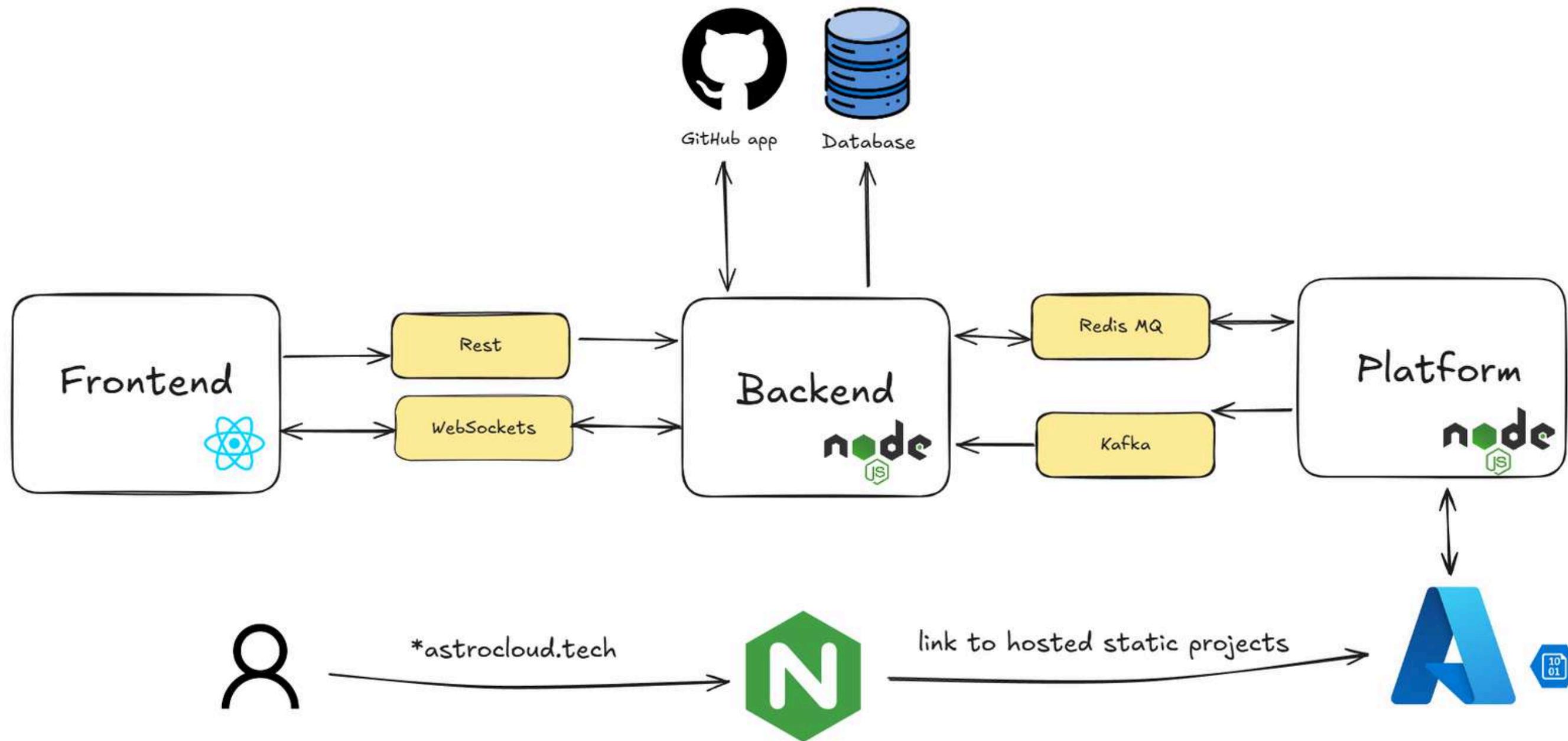


deploy application pipeline

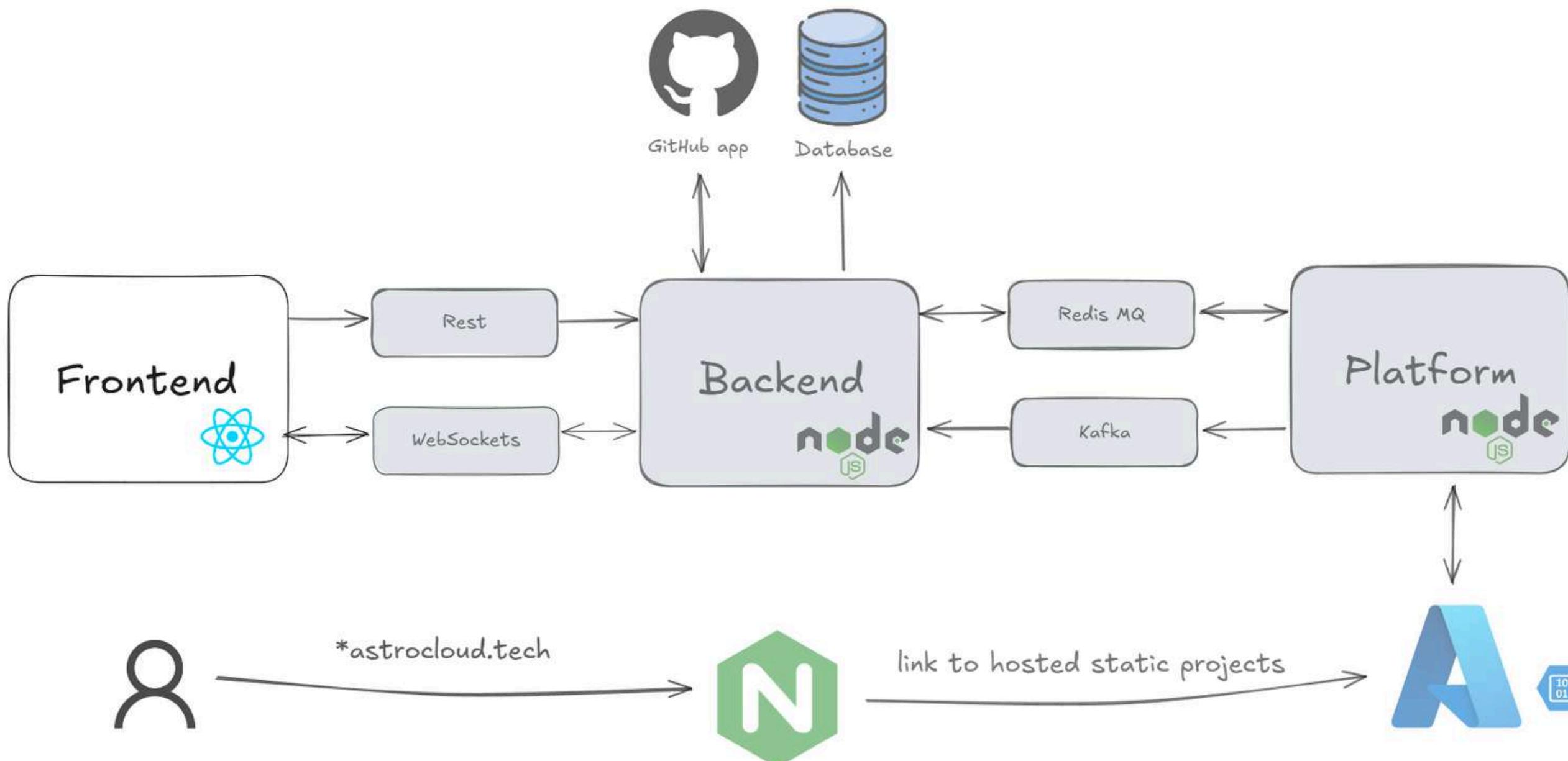


# ARCHITECTURE

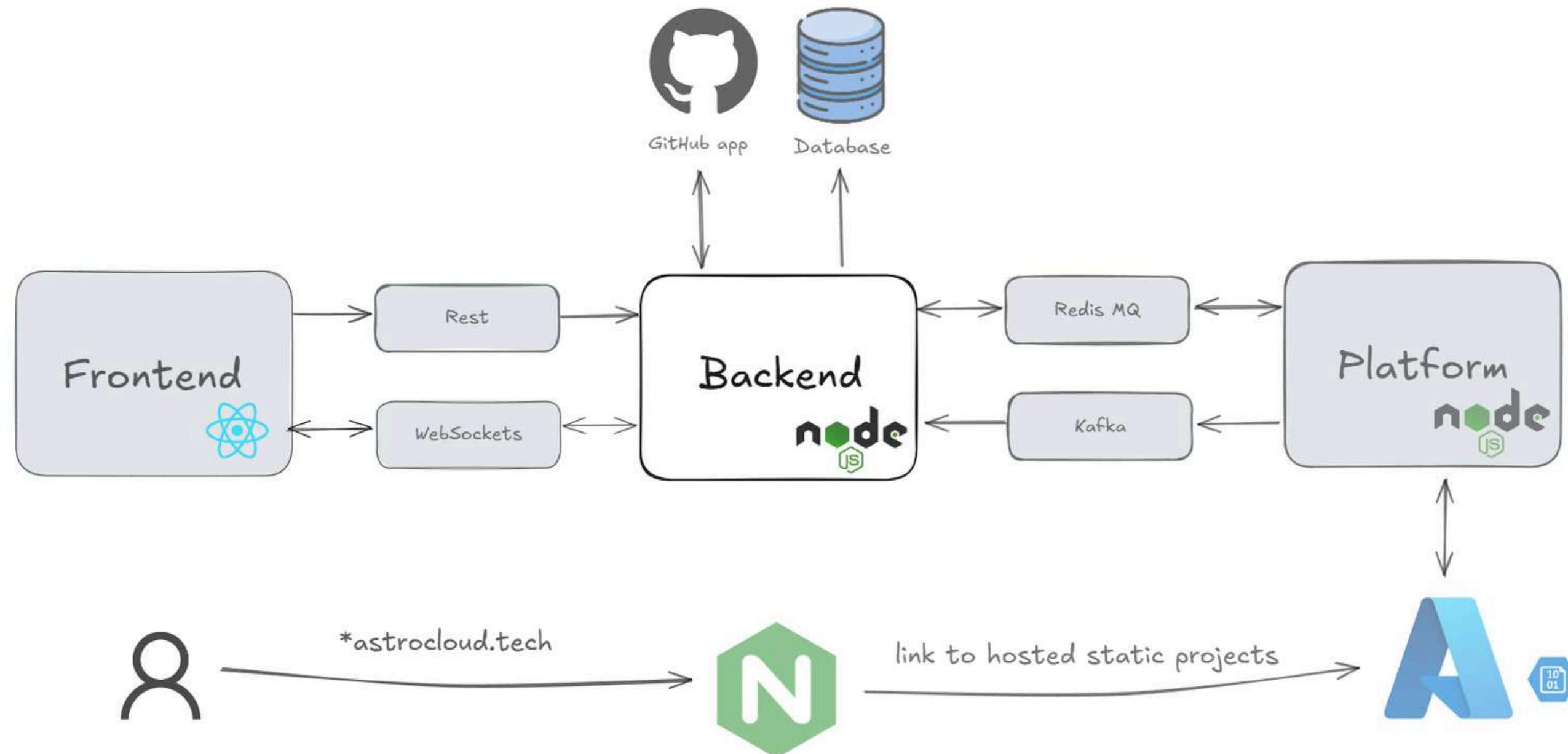
# OLD ARCHITECTURE



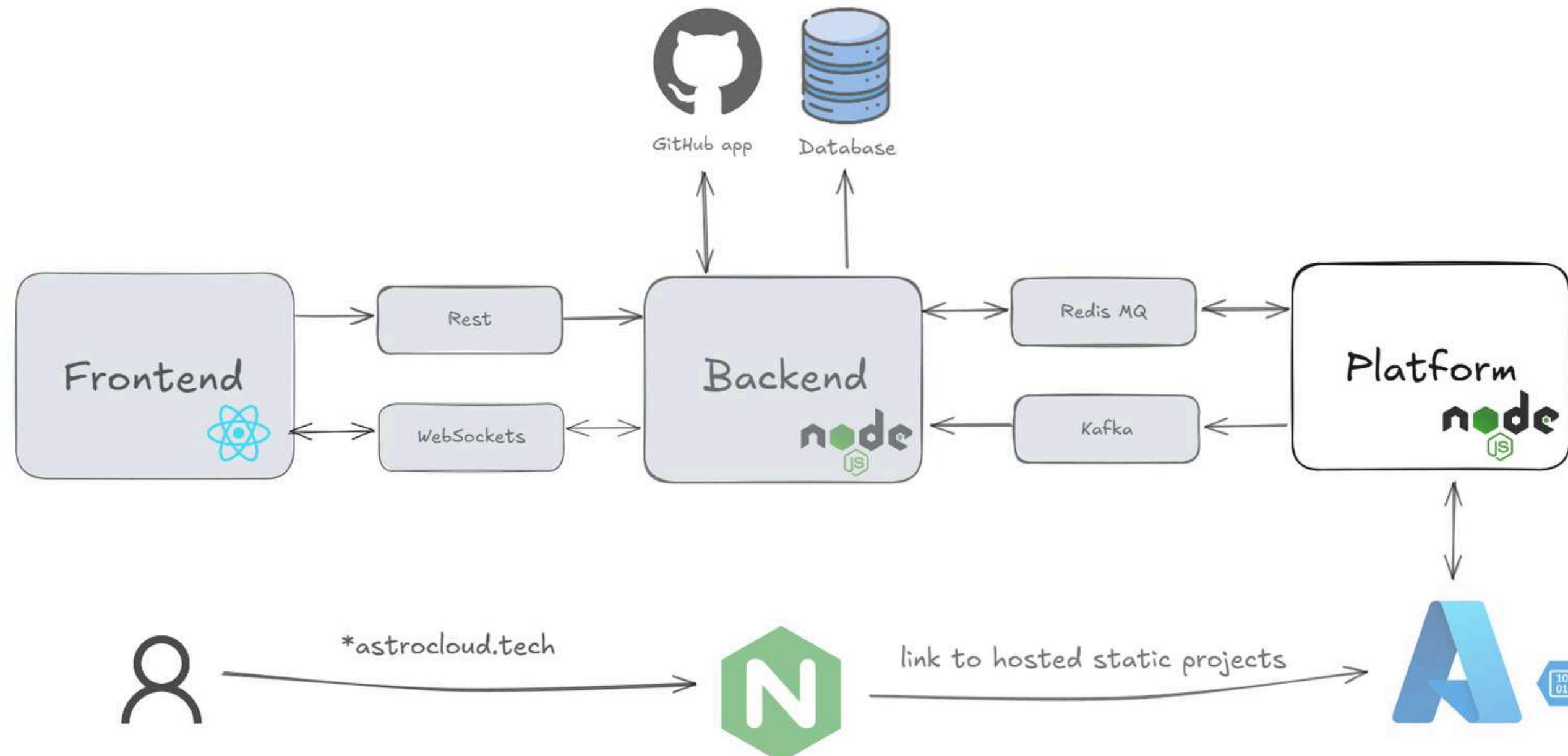
# OLD ARCHITECTURE



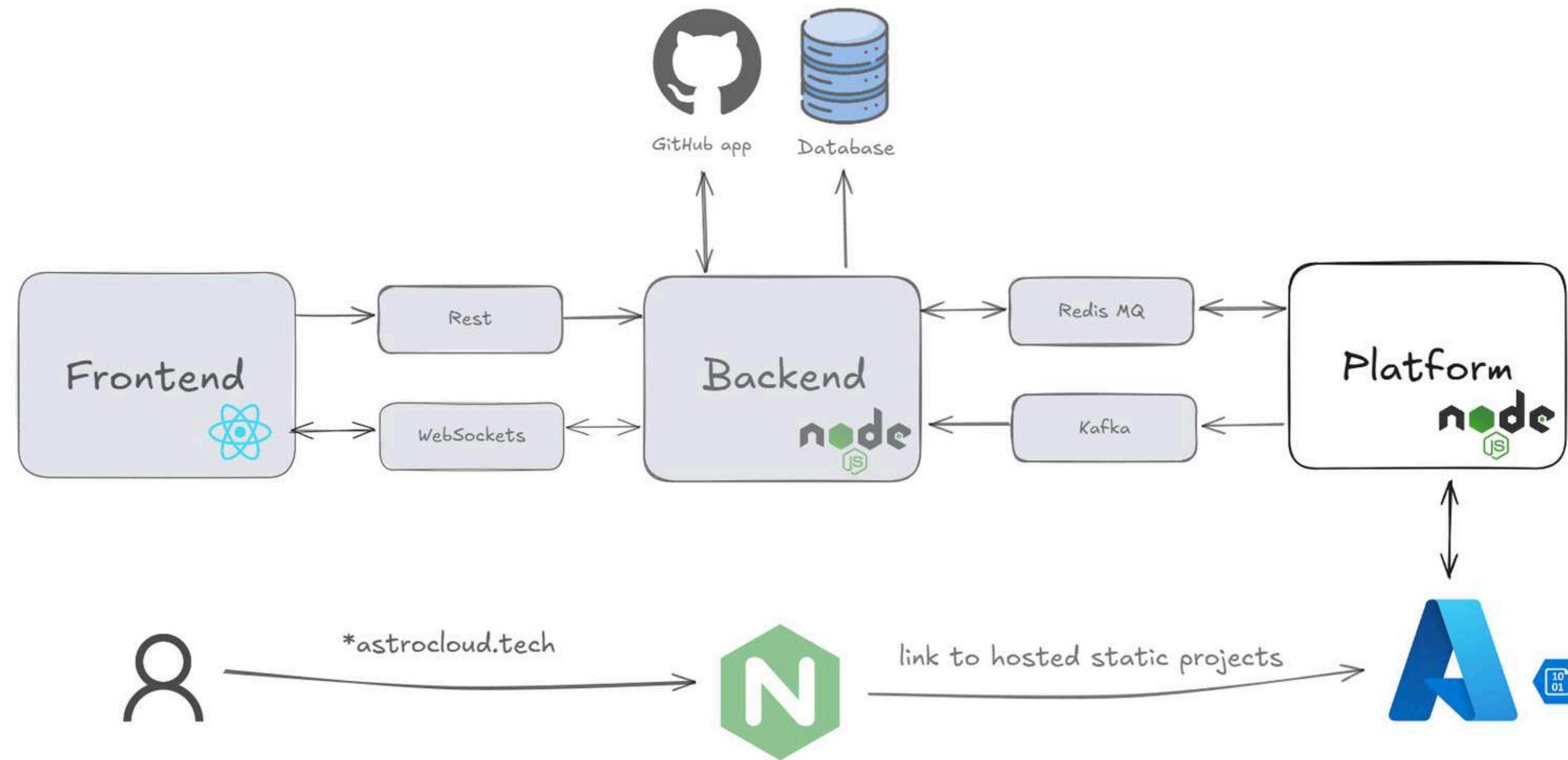
# OLD ARCHITECTURE



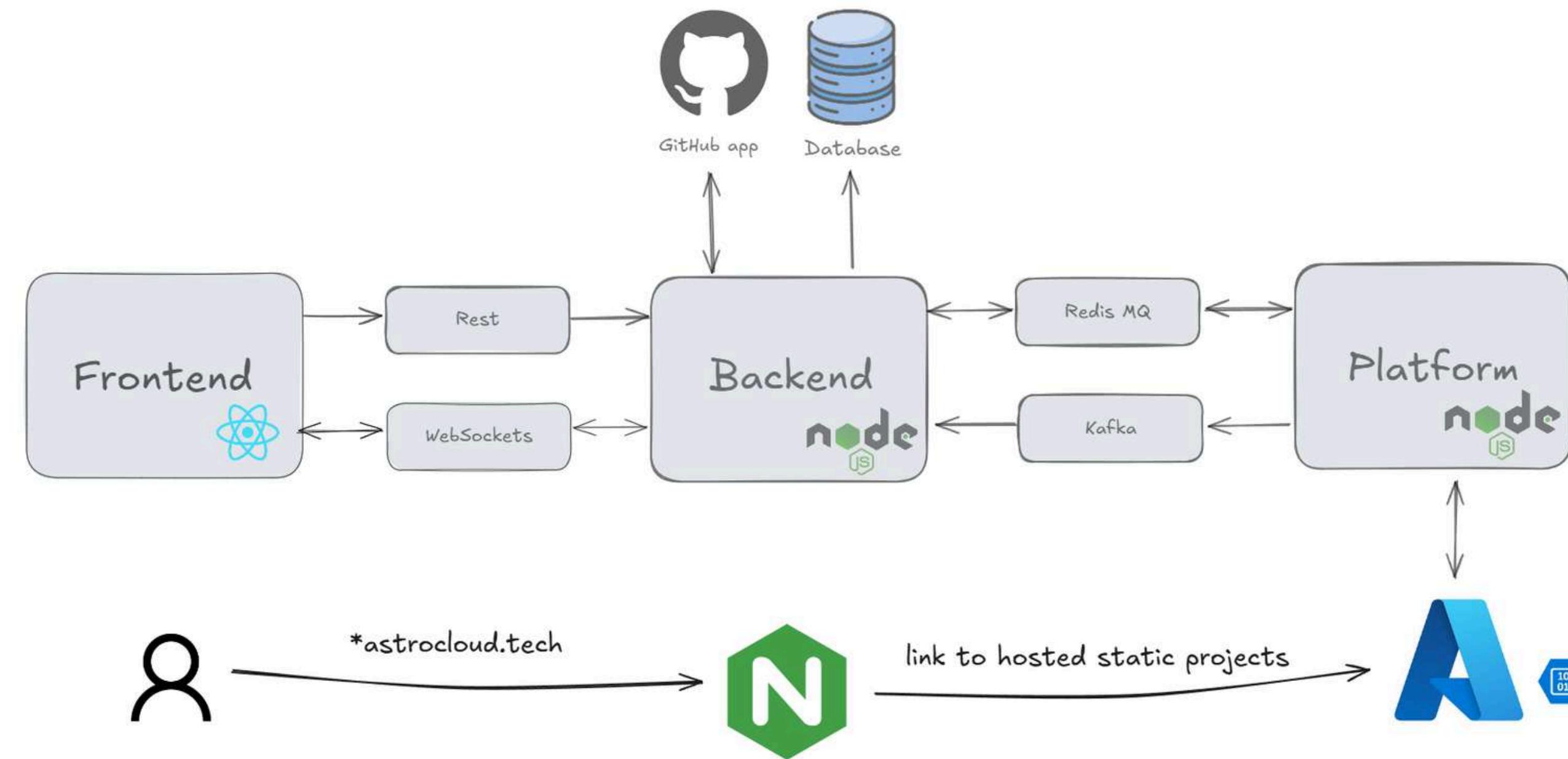
# OLD ARCHITECTURE

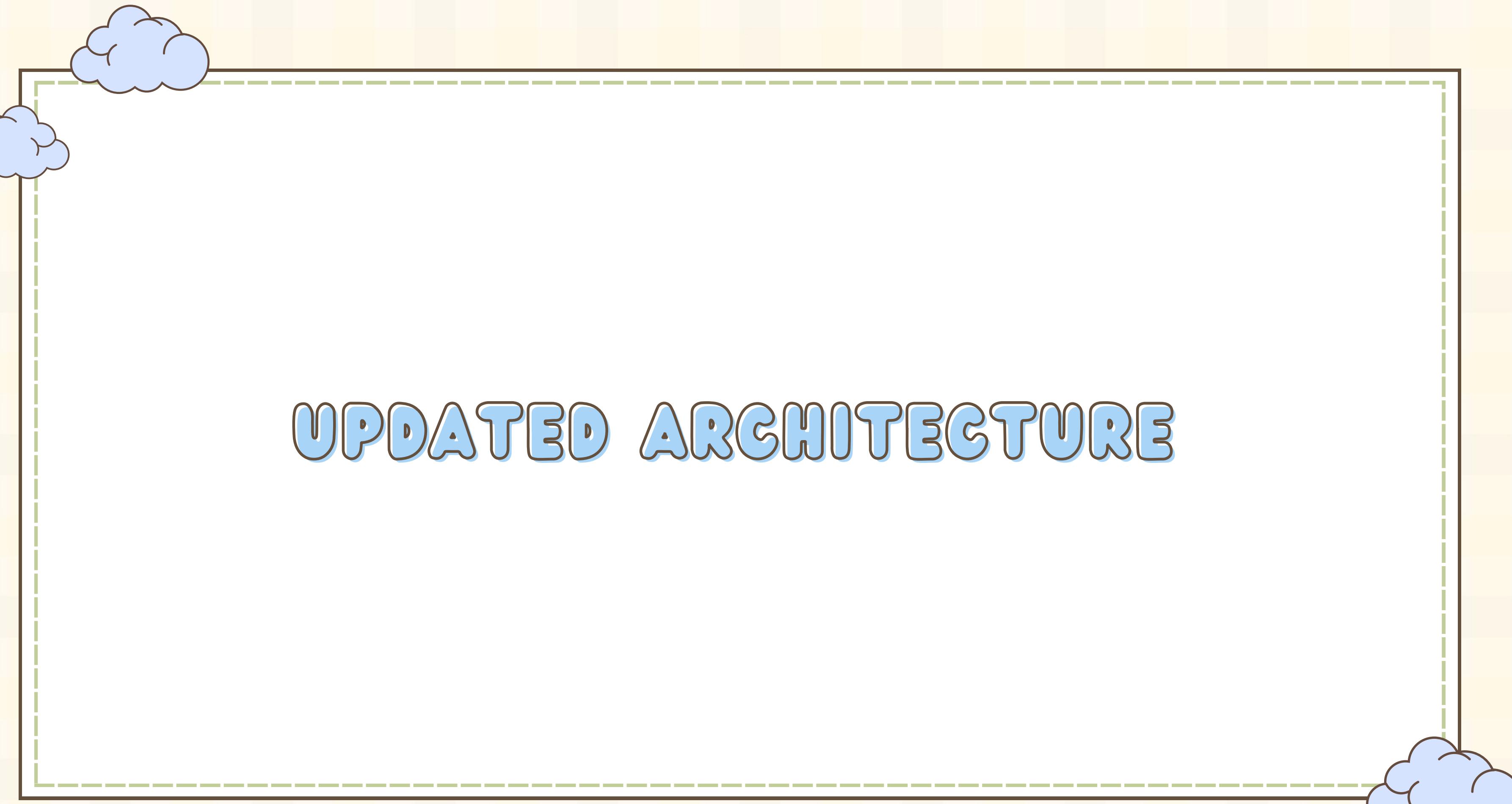


# OLD ARCHITECTURE



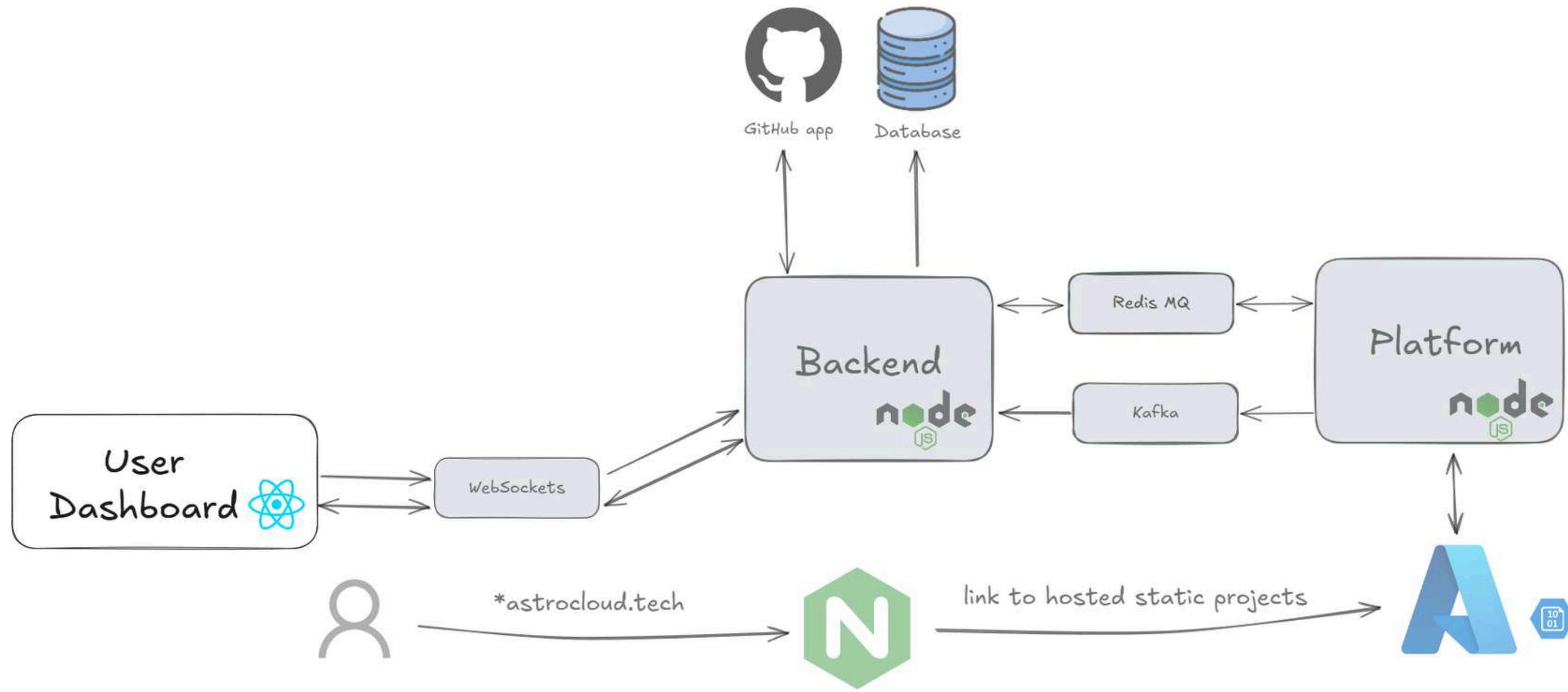
# OLD ARCHITECTURE



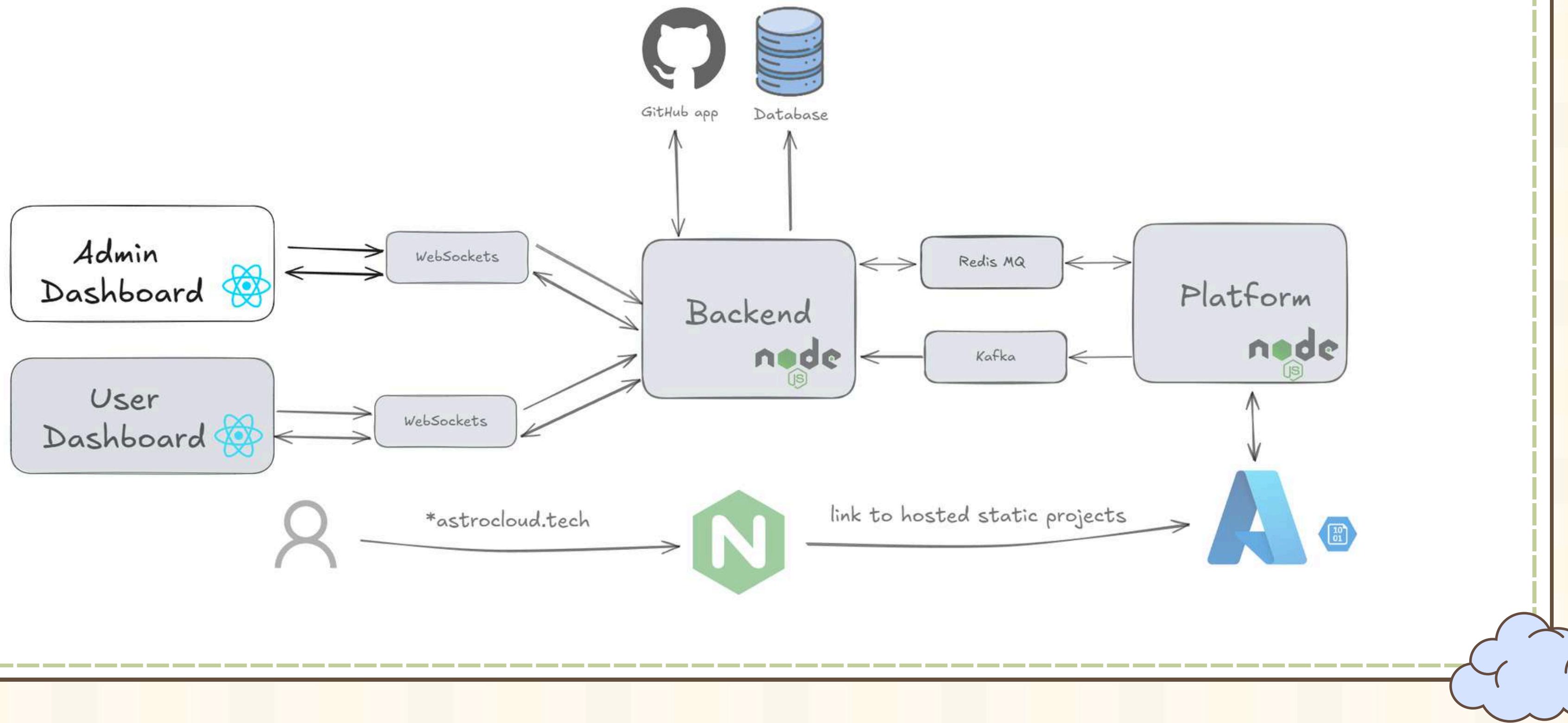


# UPDATED ARCHITECTURE

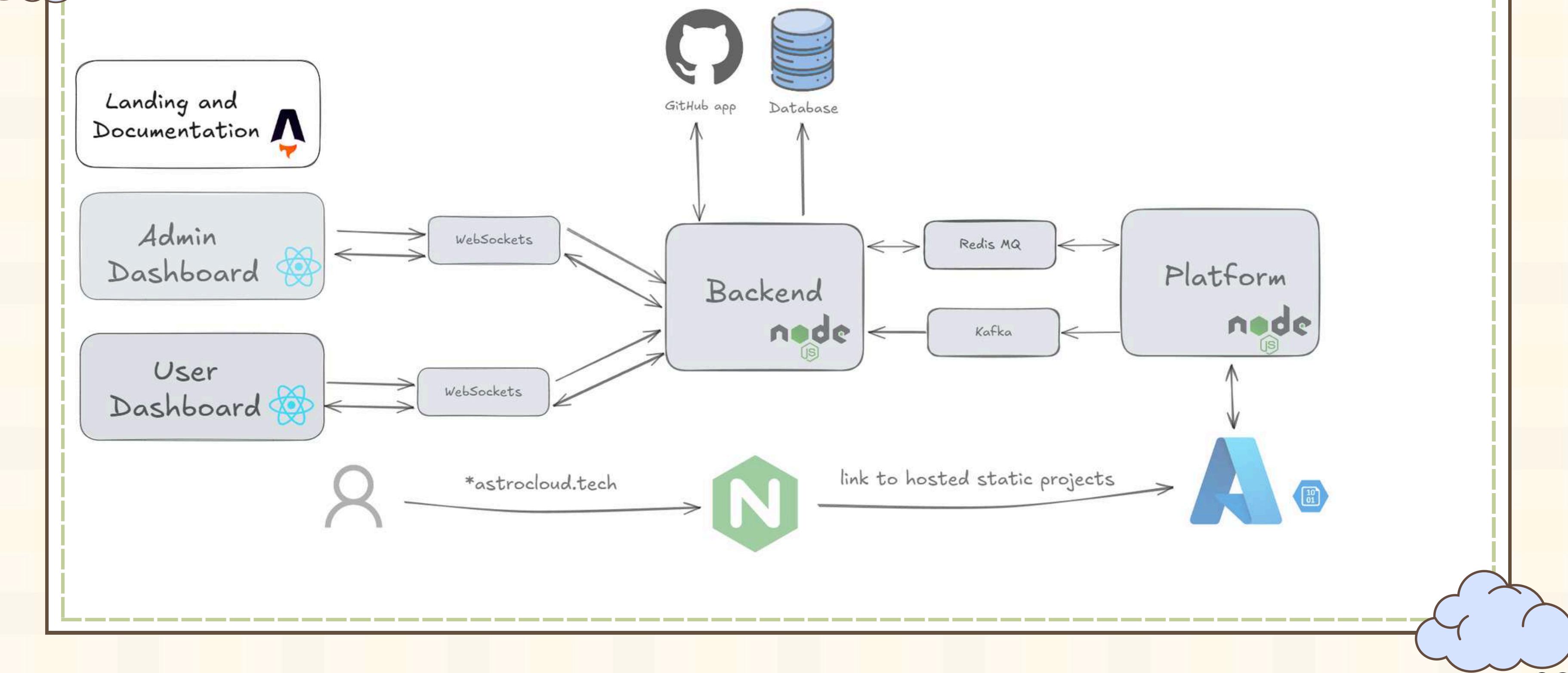
# UPDATED ARCHITECTURE



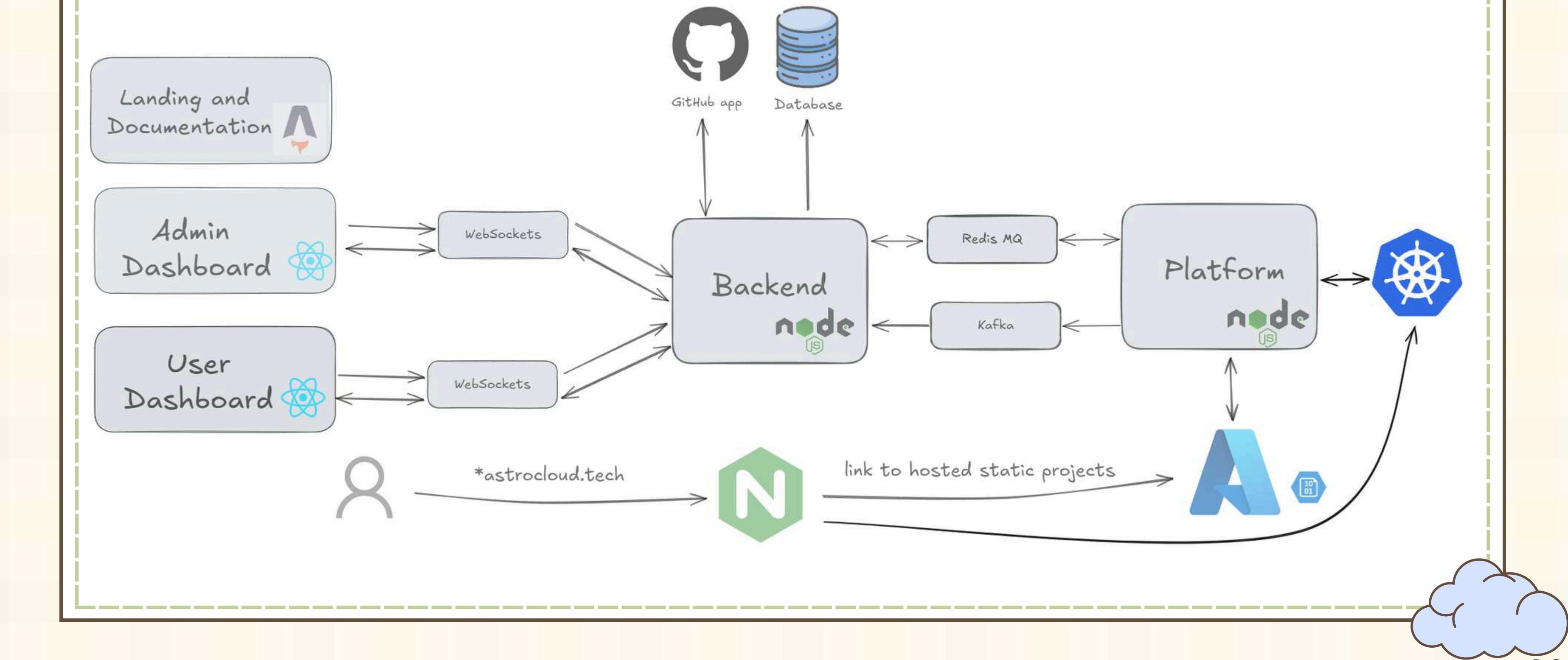
# UPDATED ARCHITECTURE



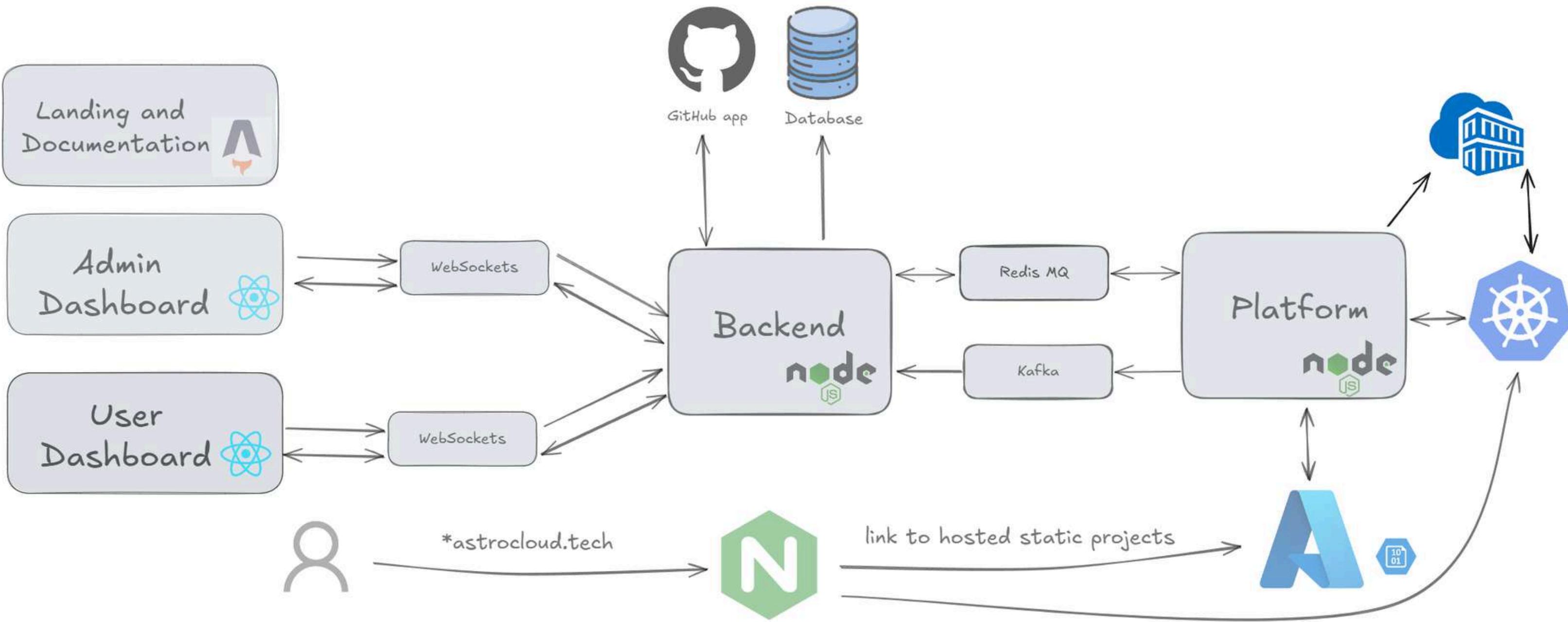
# UPDATED ARCHITECTURE



# UPDATED ARCHITECTURE

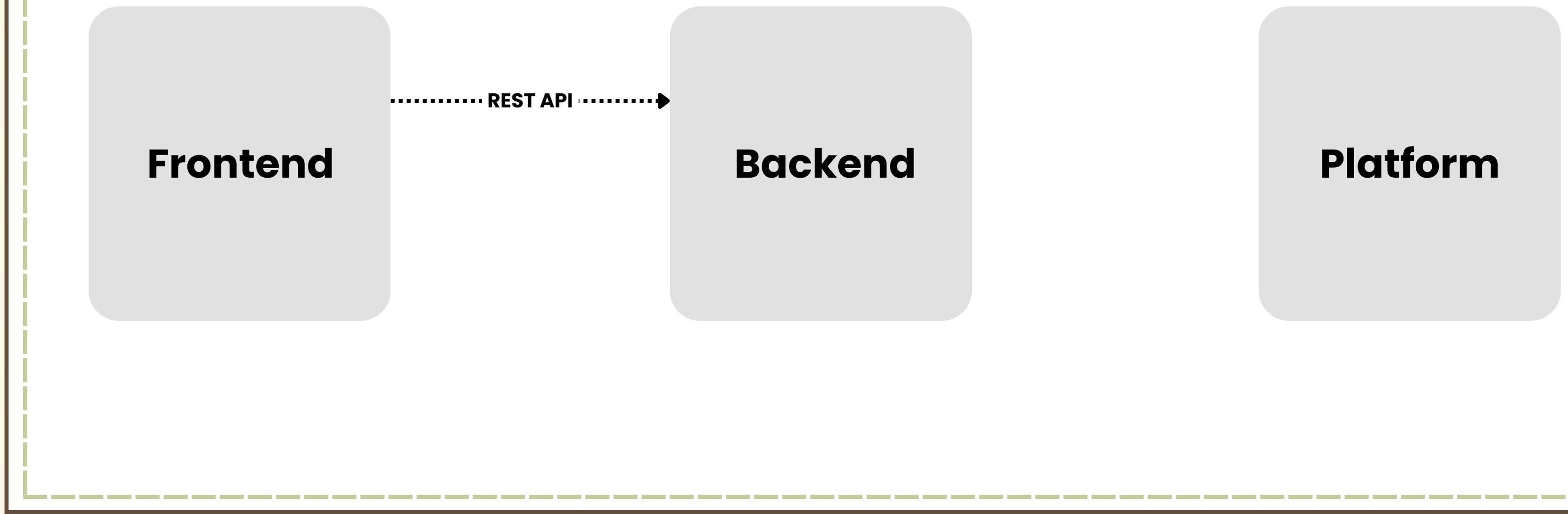


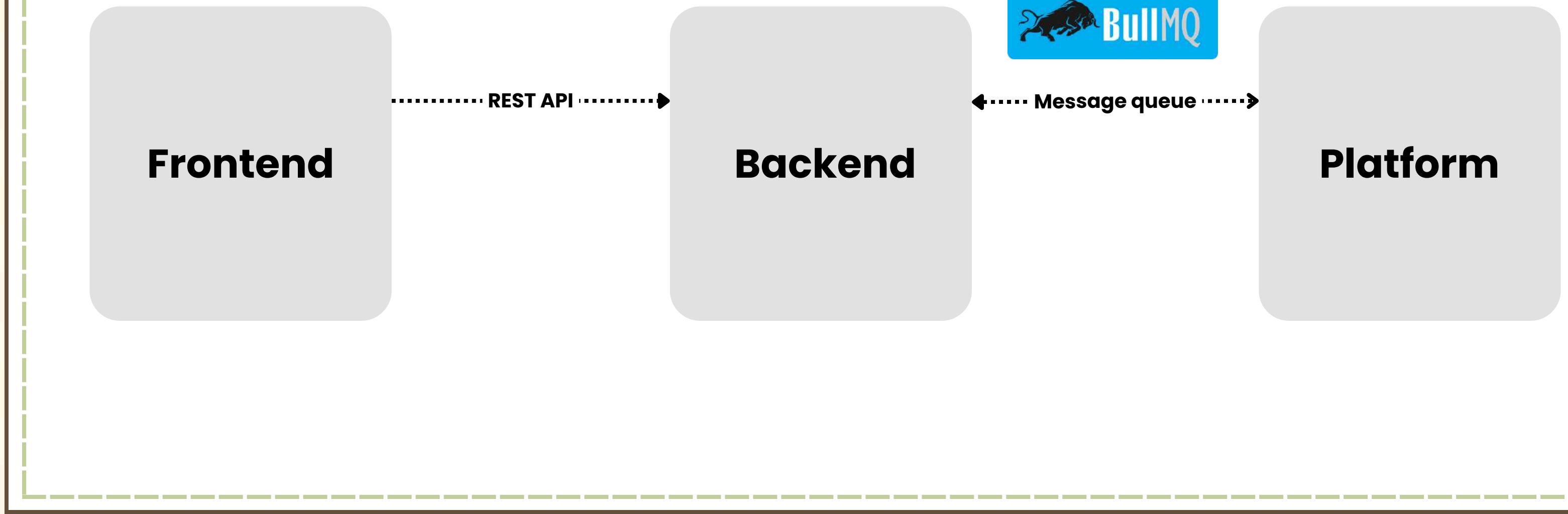
# UPDATED ARCHITECTURE

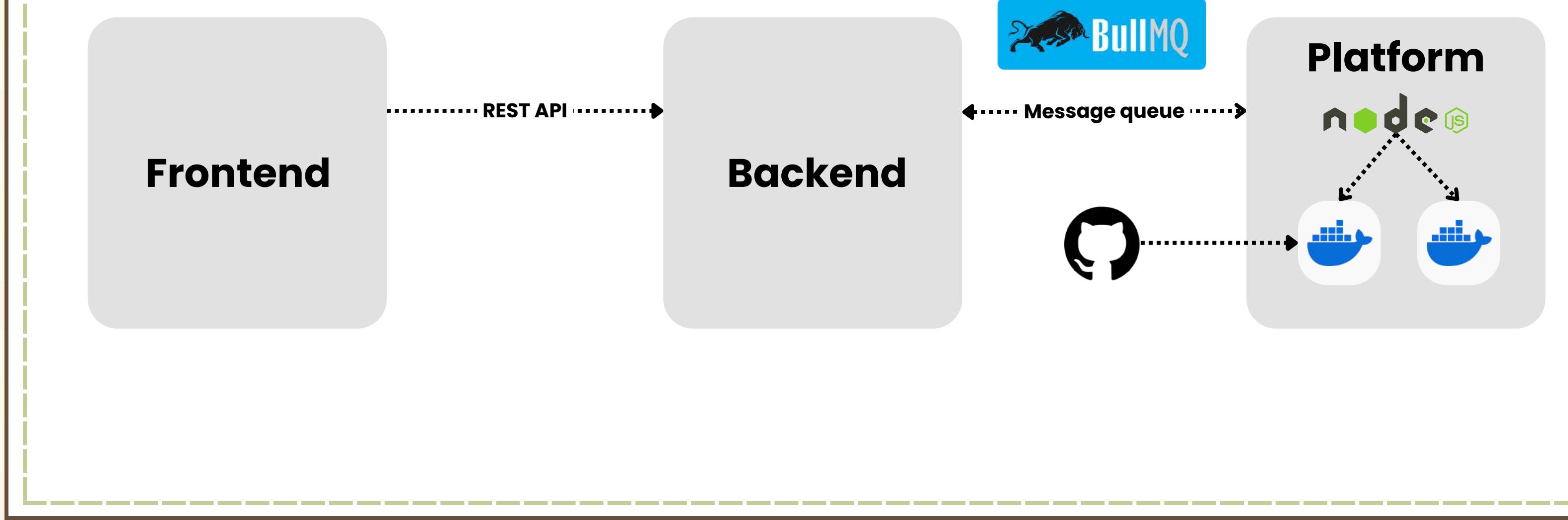


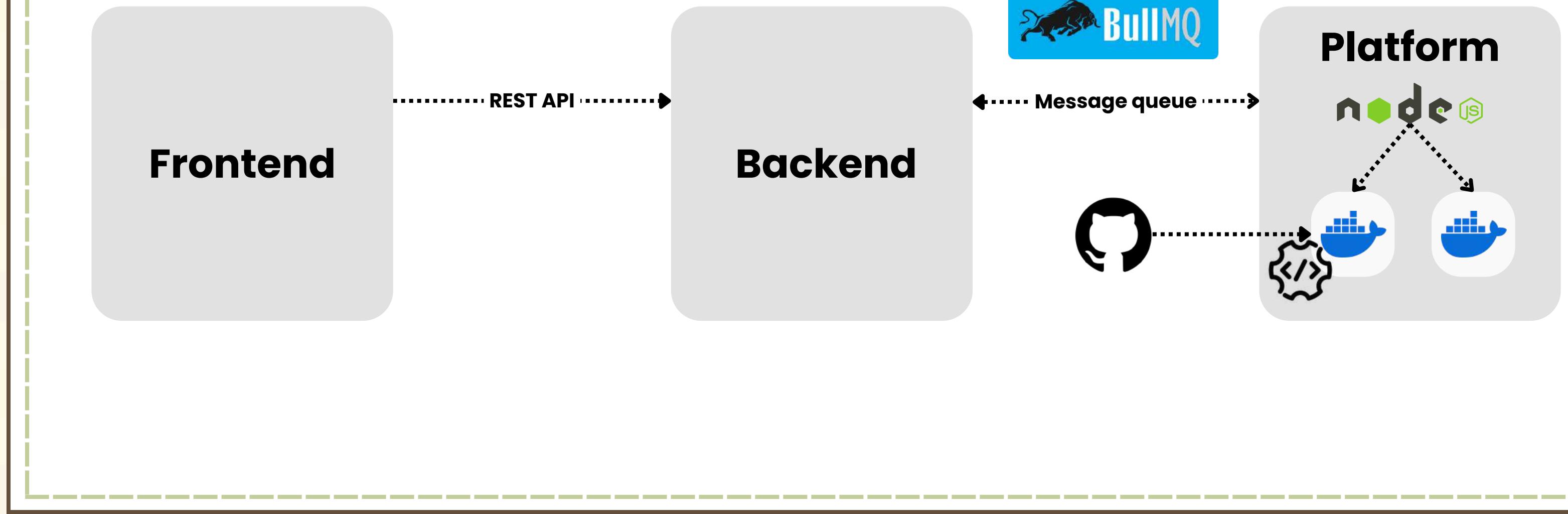


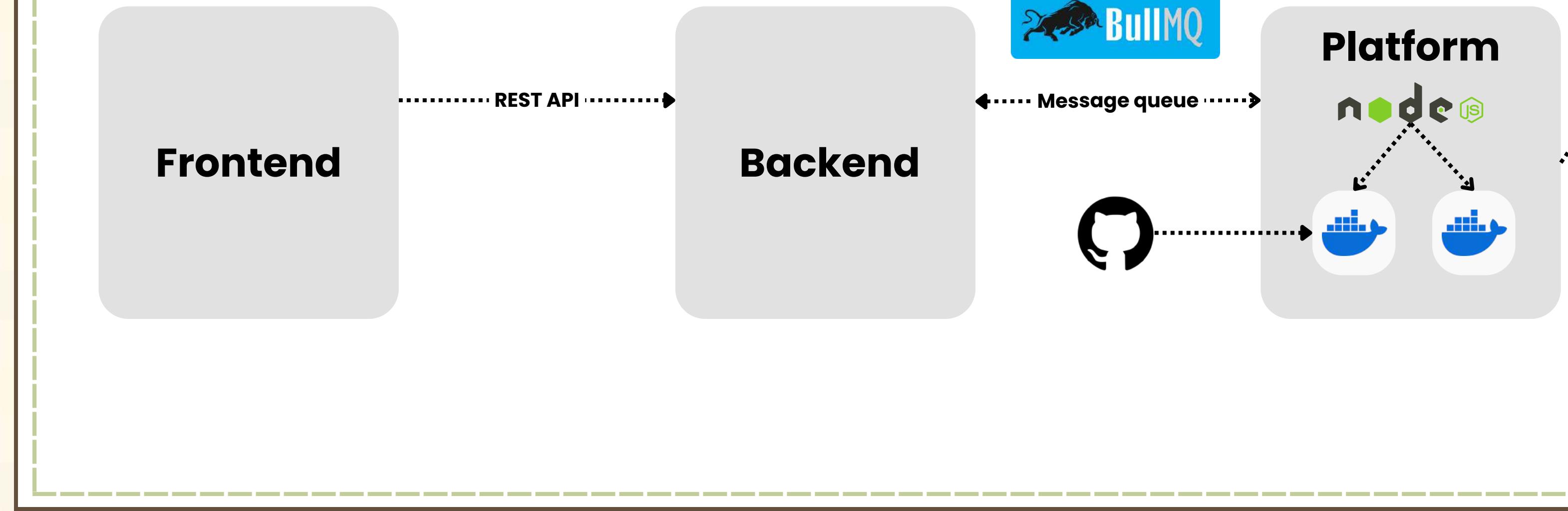
# STATIC HOSTING

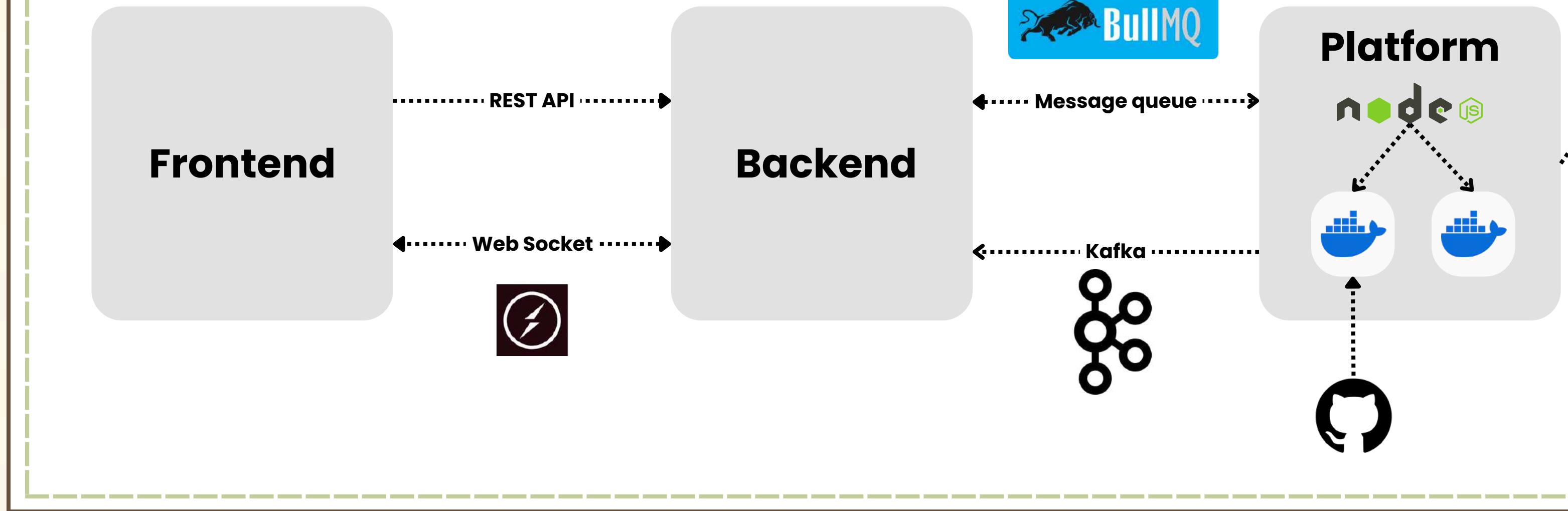


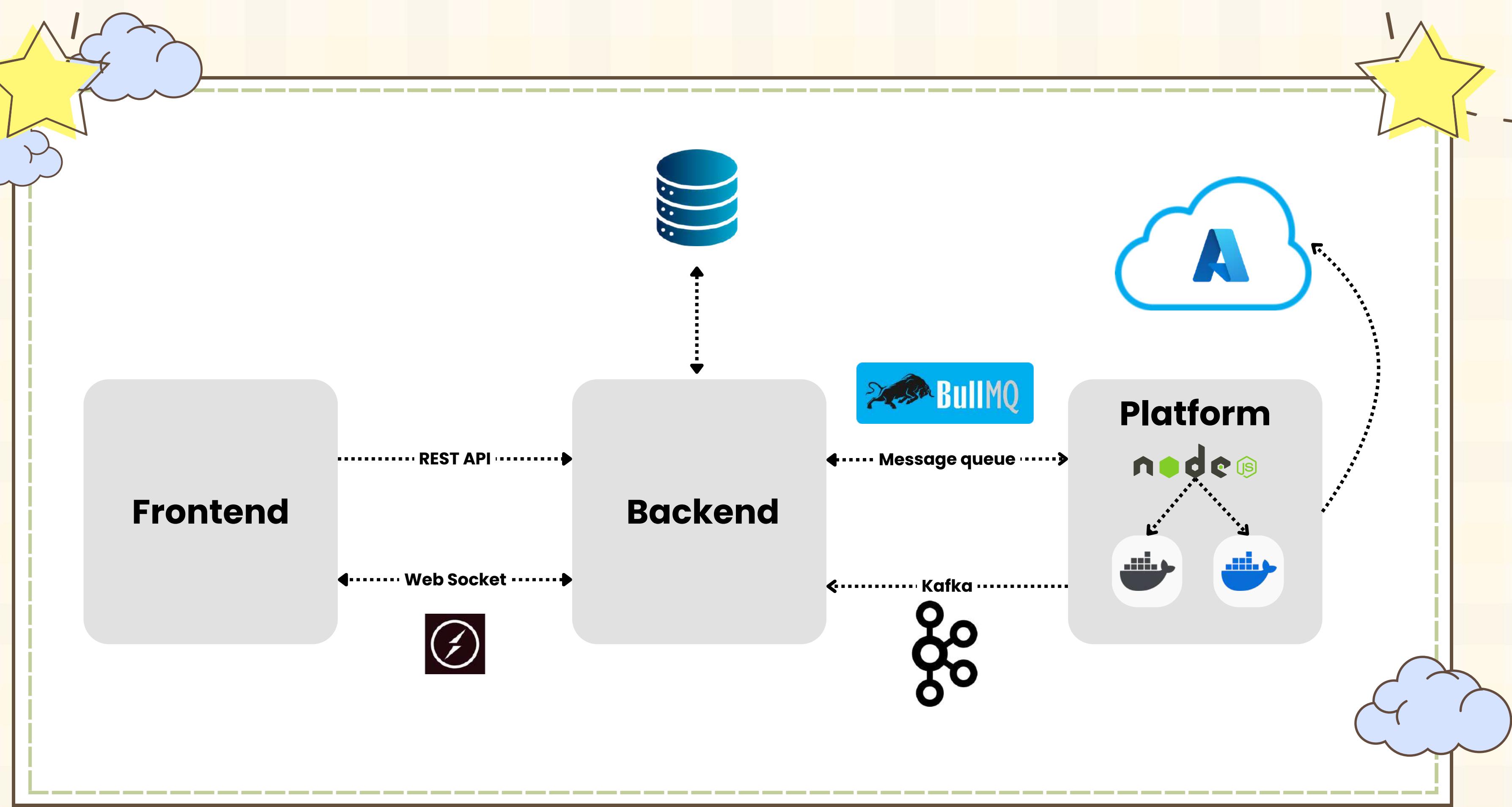


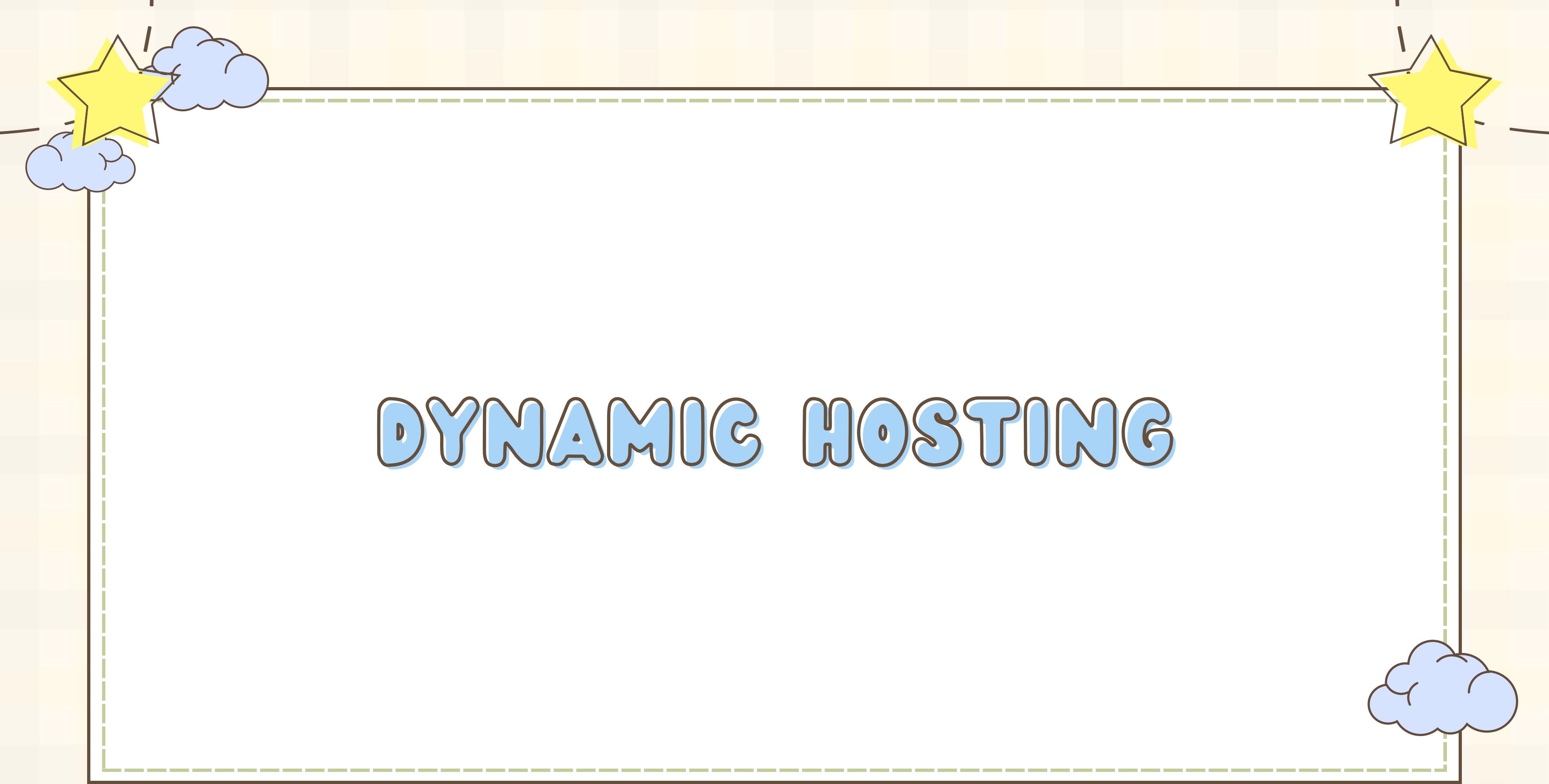




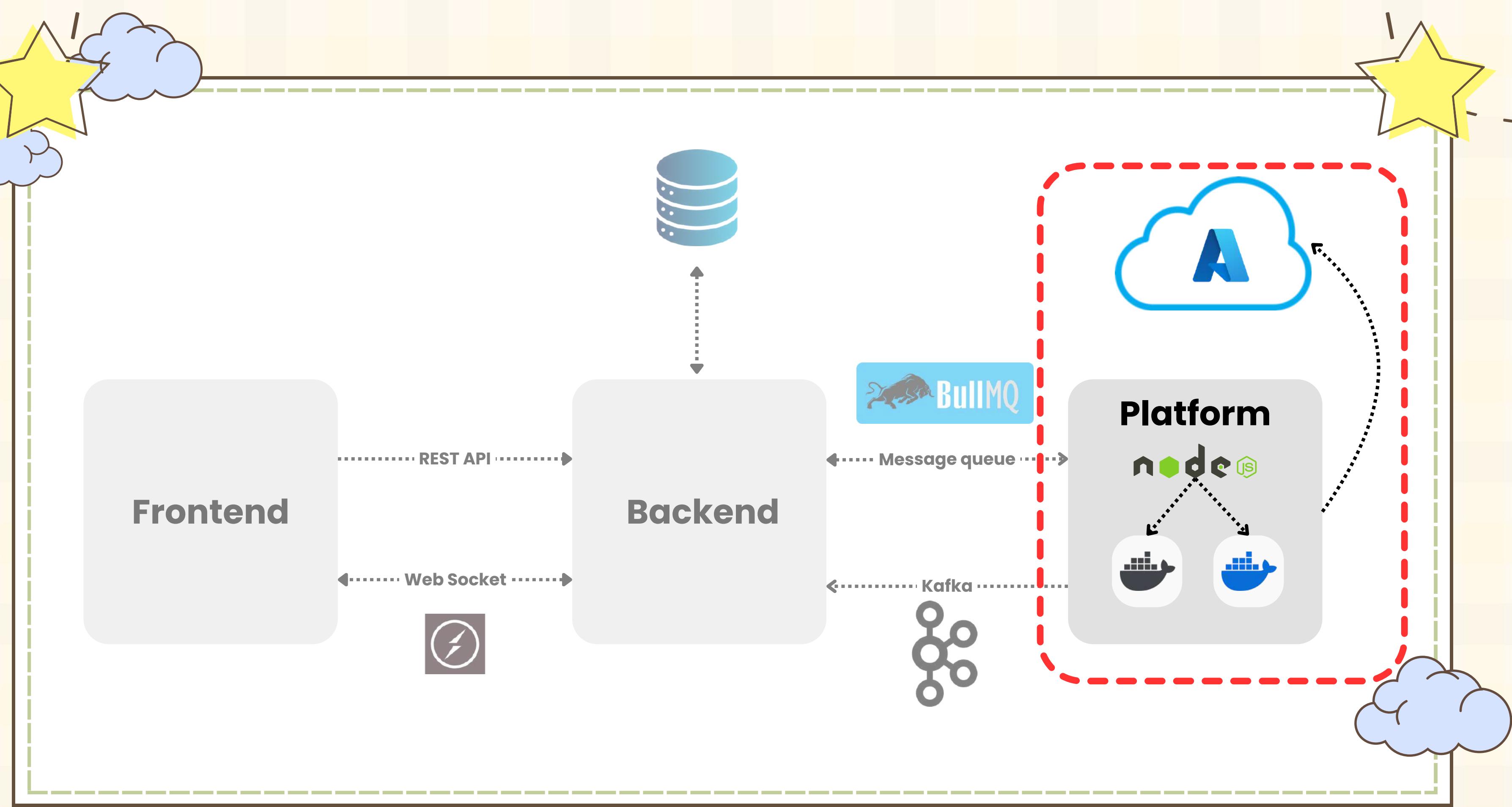


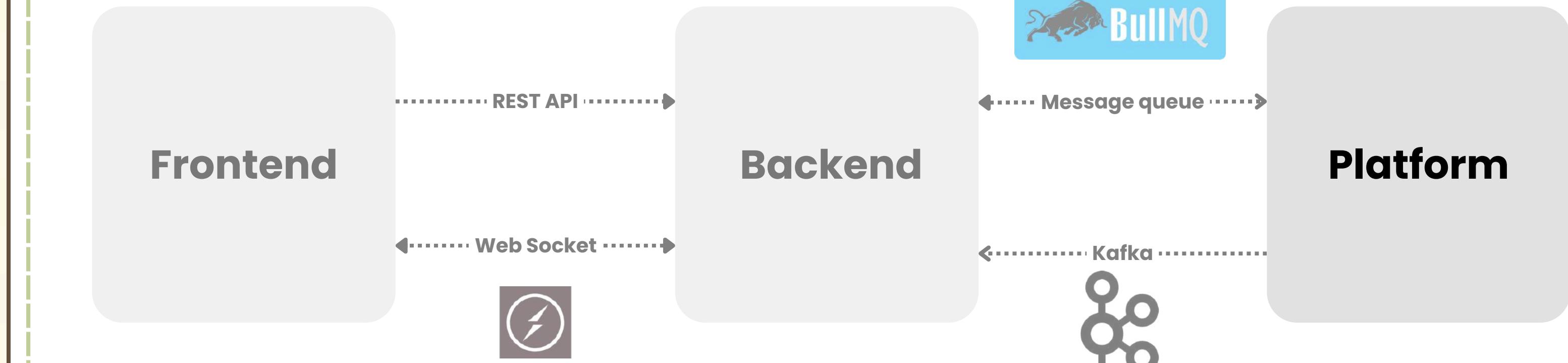


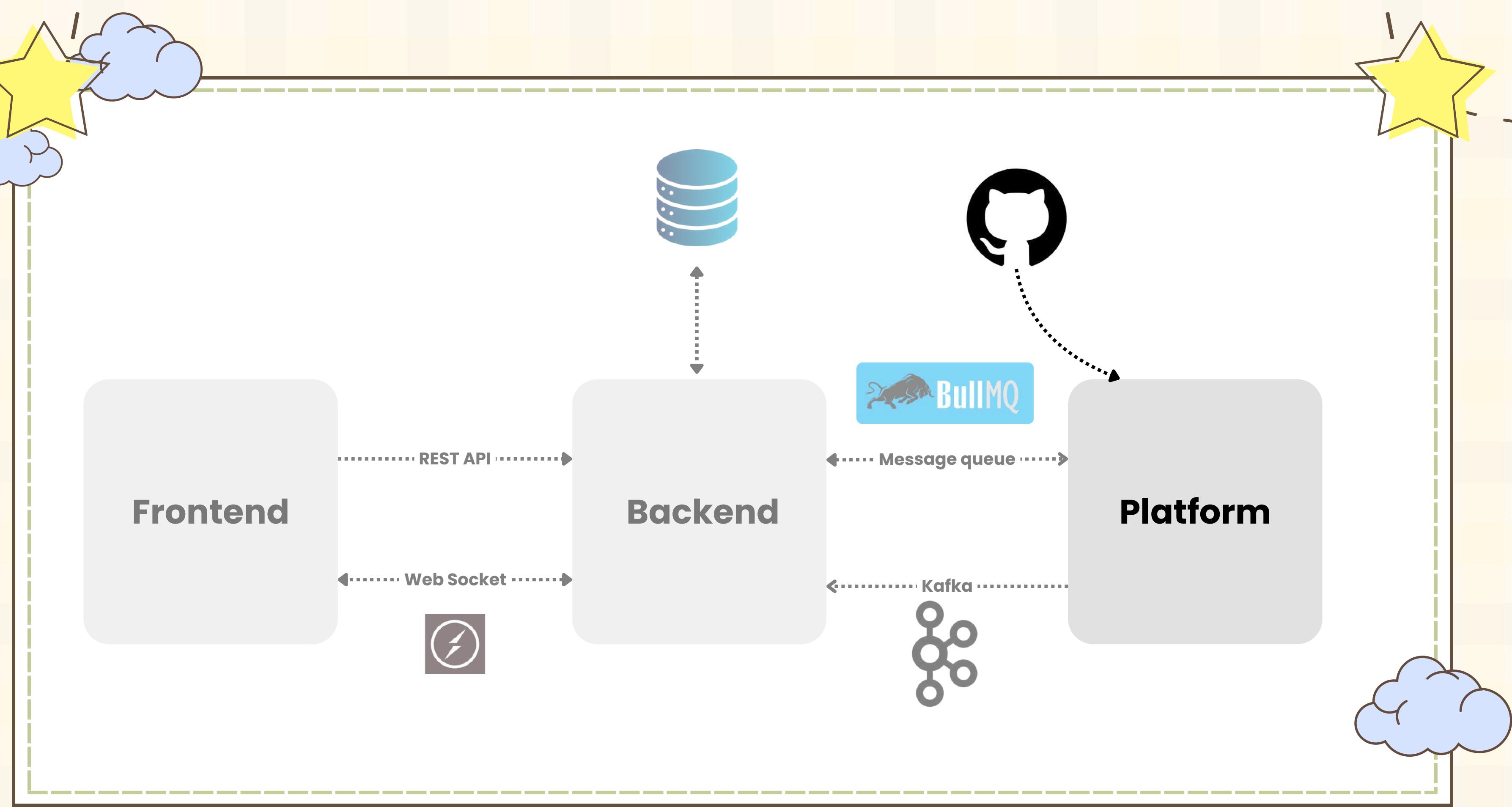


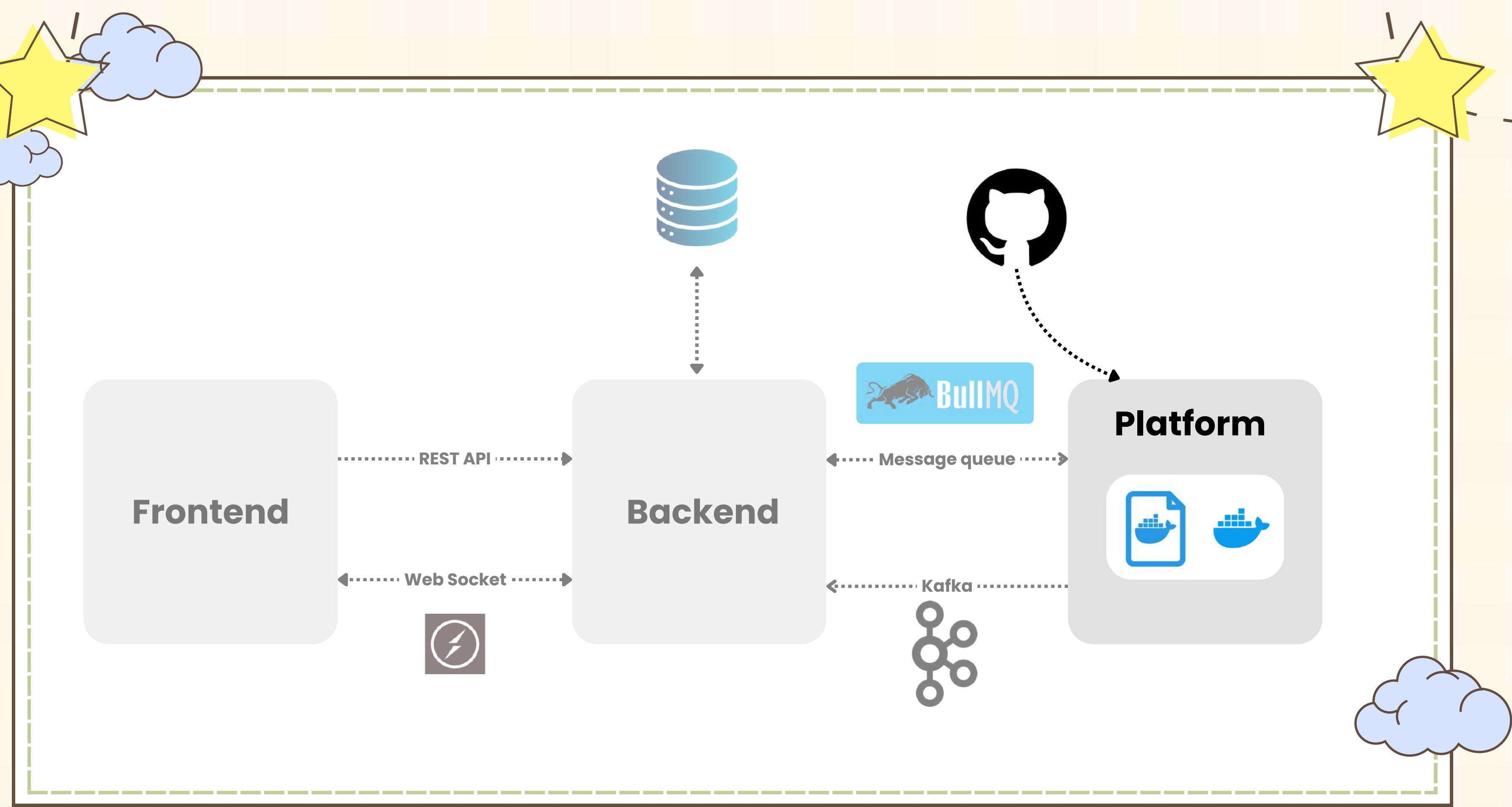


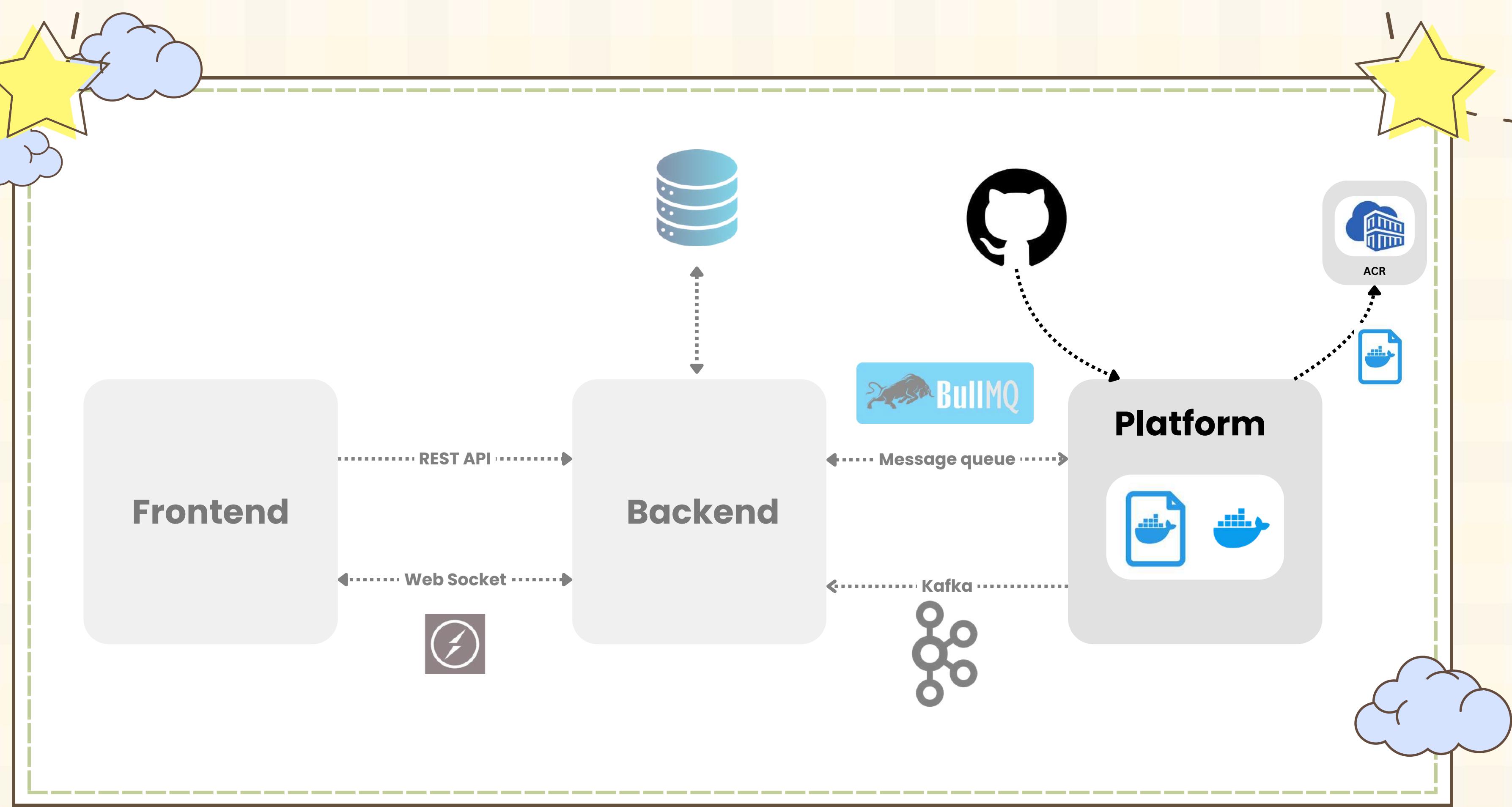
# DYNAMIC HOSTING

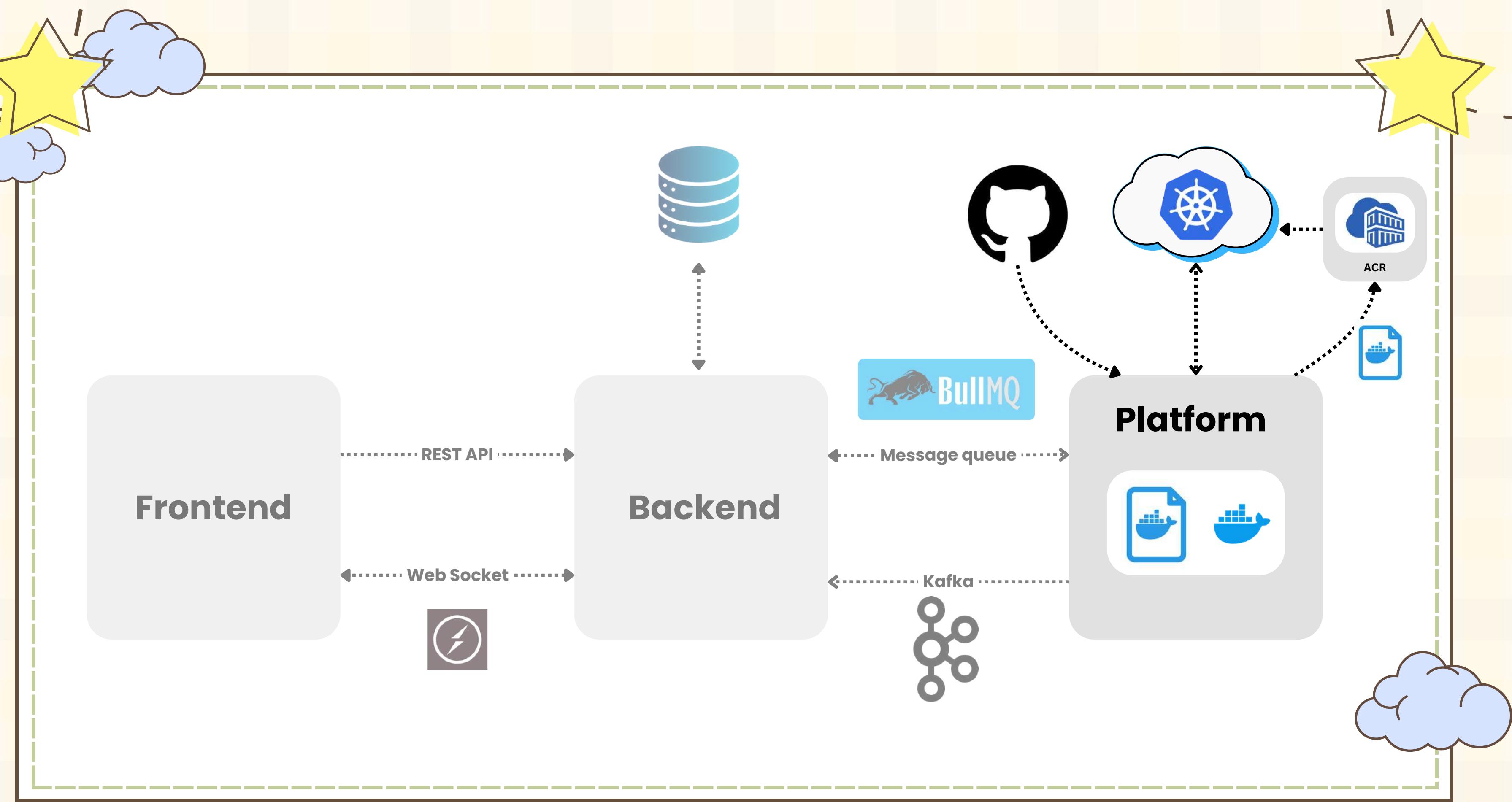












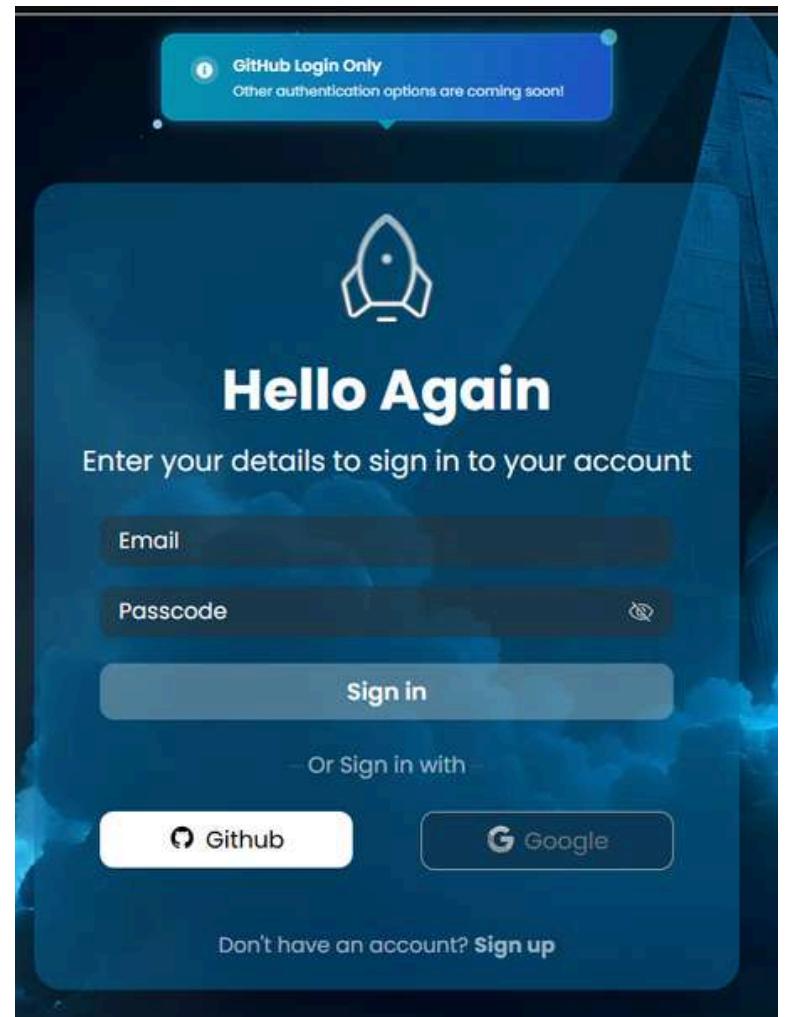
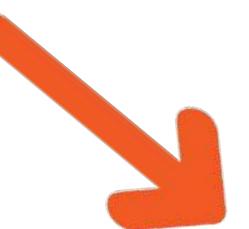
# FEATURES



**LOGIN WITH  
GITHUB  
&  
REFRESH TOKENS**

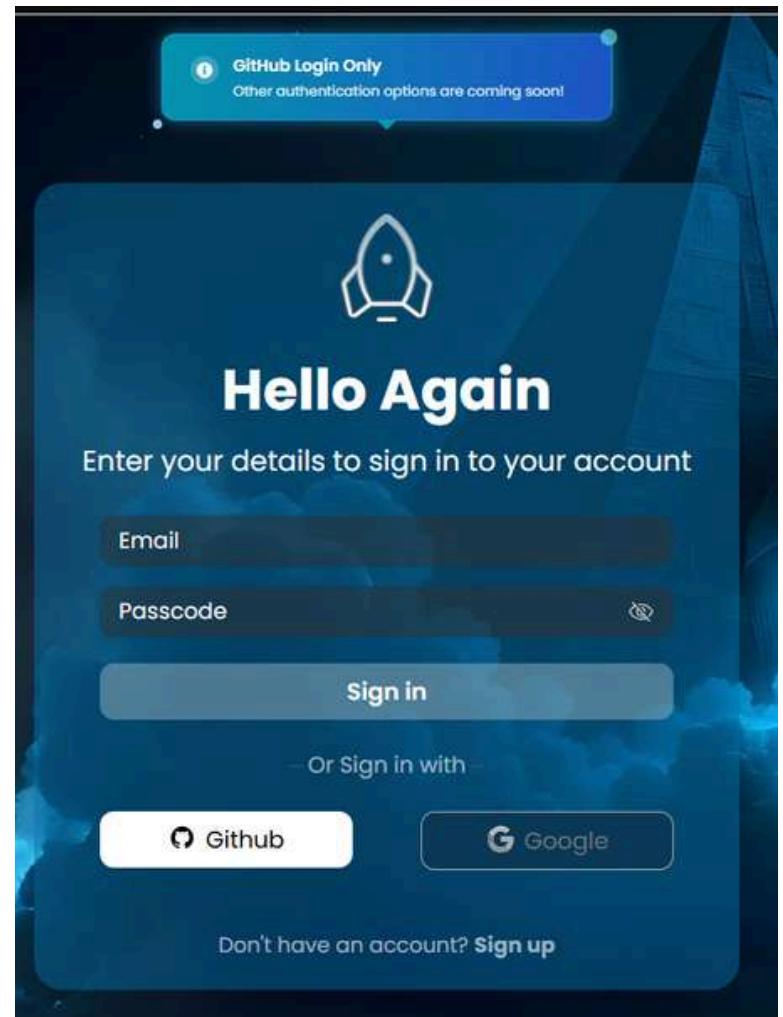
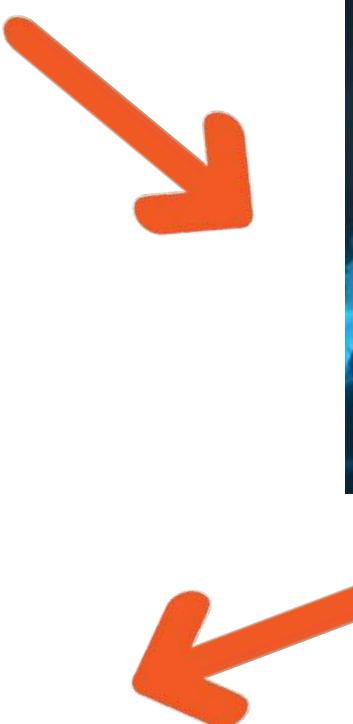
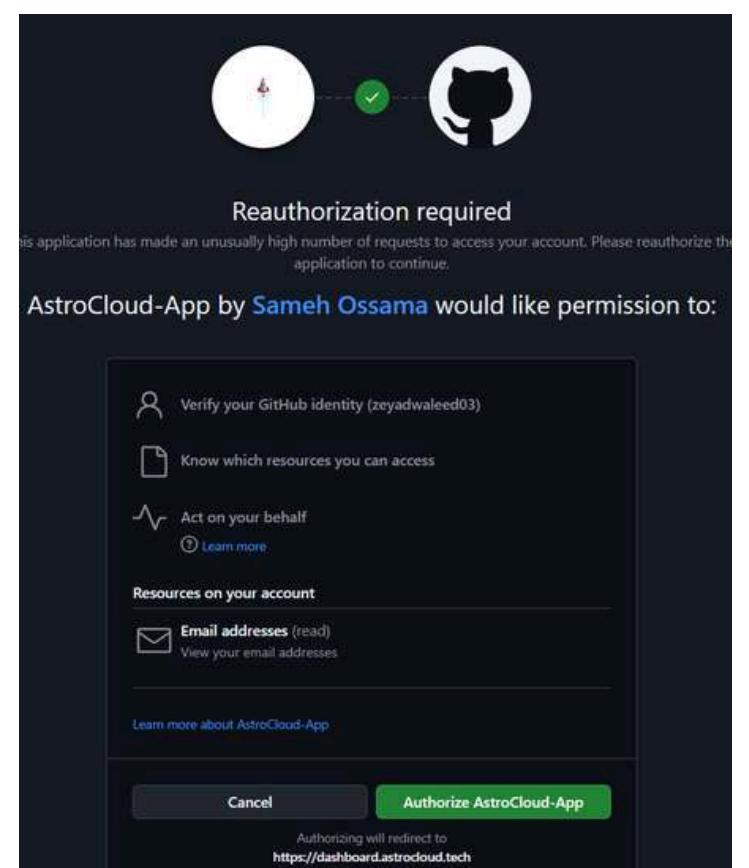
# AUTHENTICATION WORKFLOW

- User presses login button



# AUTHENTICATION WORKFLOW

- User presses login button
- User gives Github consent

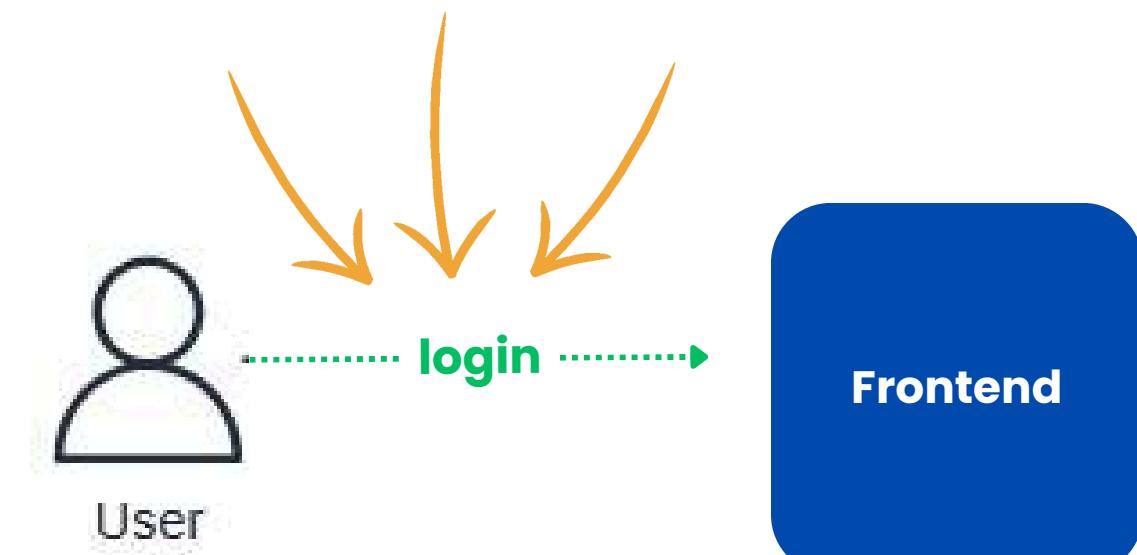


# AUTHENTICATION WORKFLOW

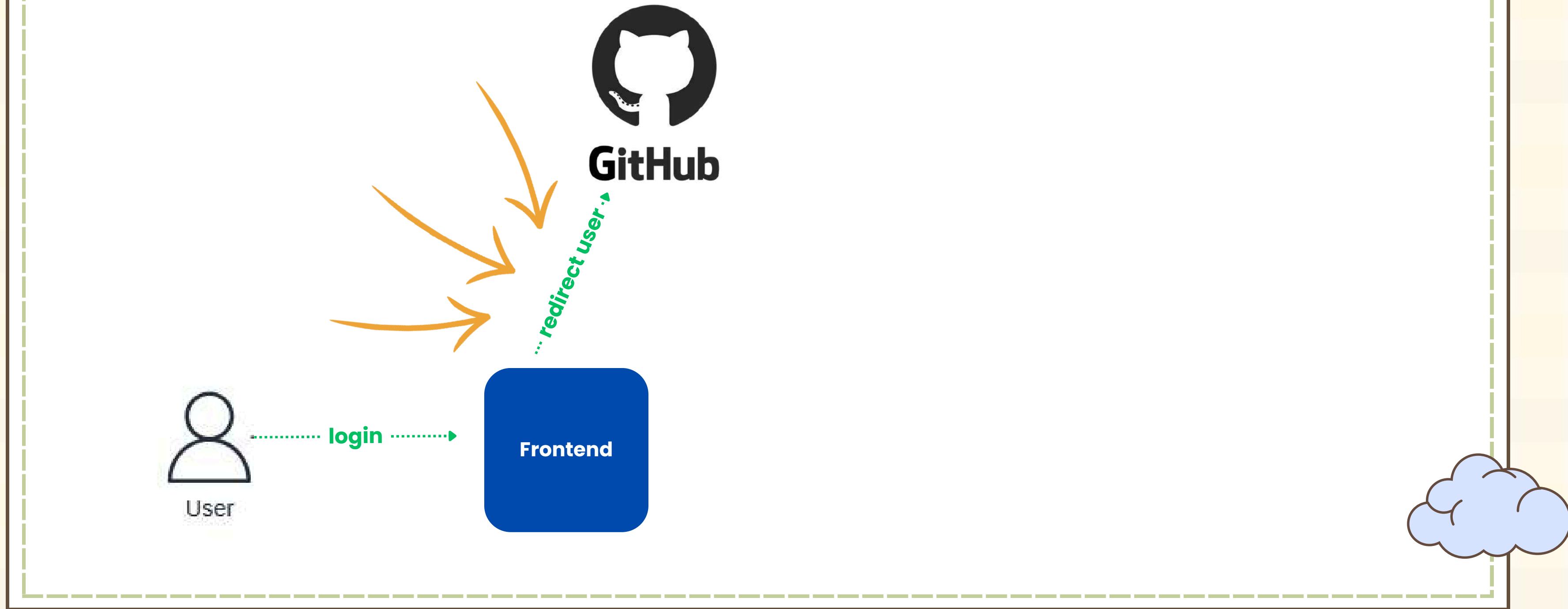
- Github sends access code to frontend
- Frontend sends it to backend
- Backend replies with access and refresh tokens



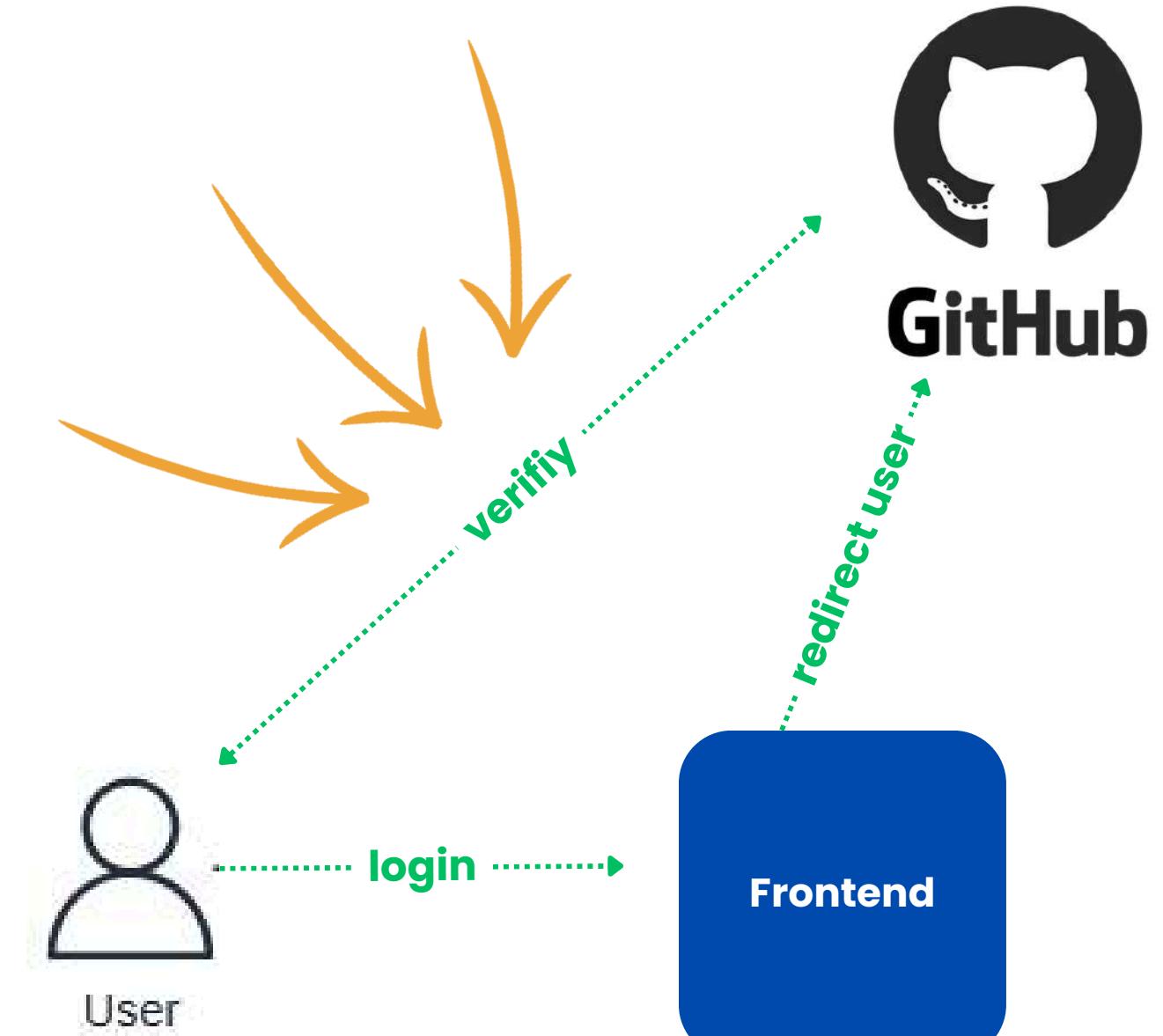
# AUTHENTICATION WORKFLOW



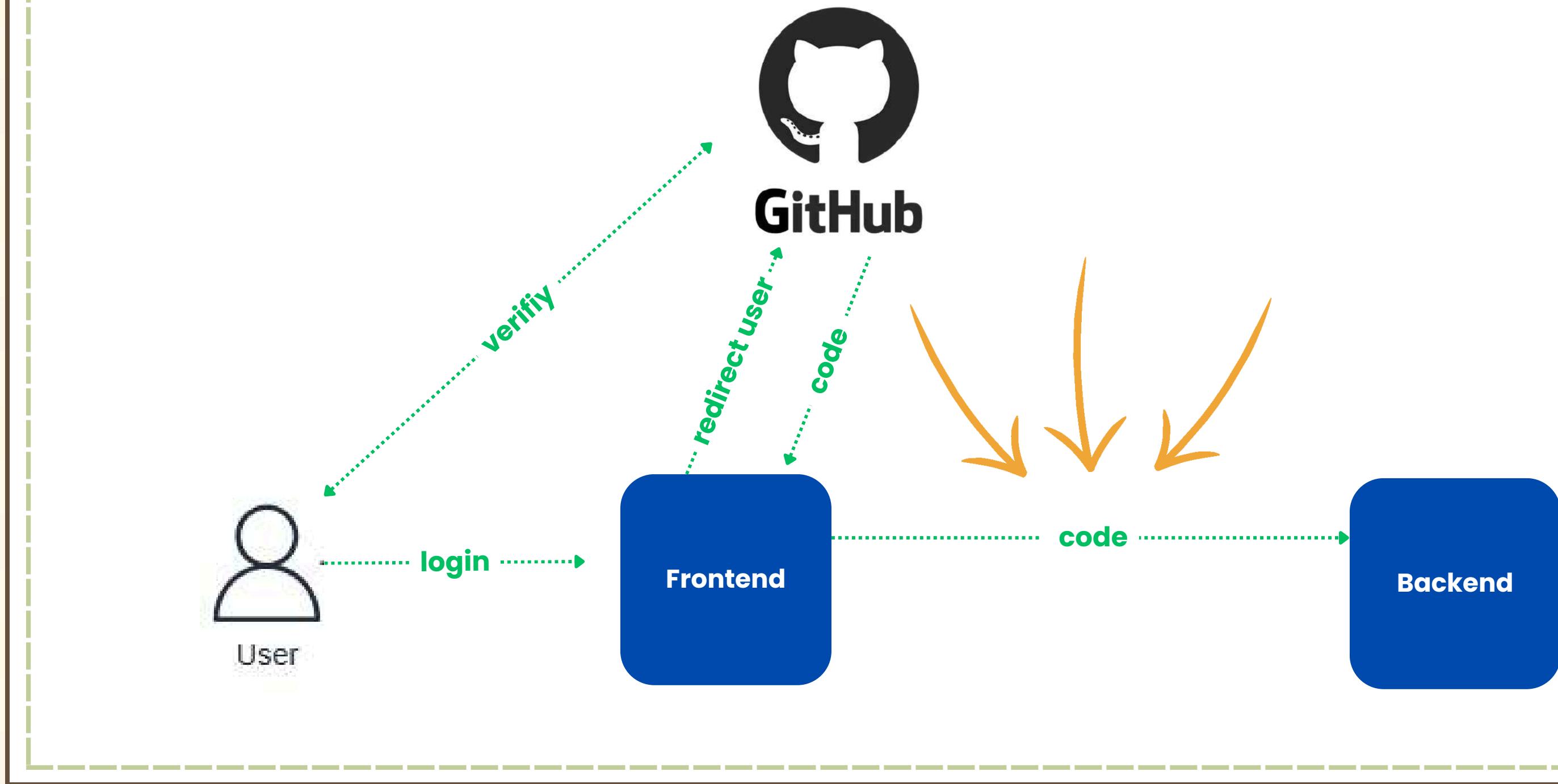
# AUTHENTICATION WORKFLOW



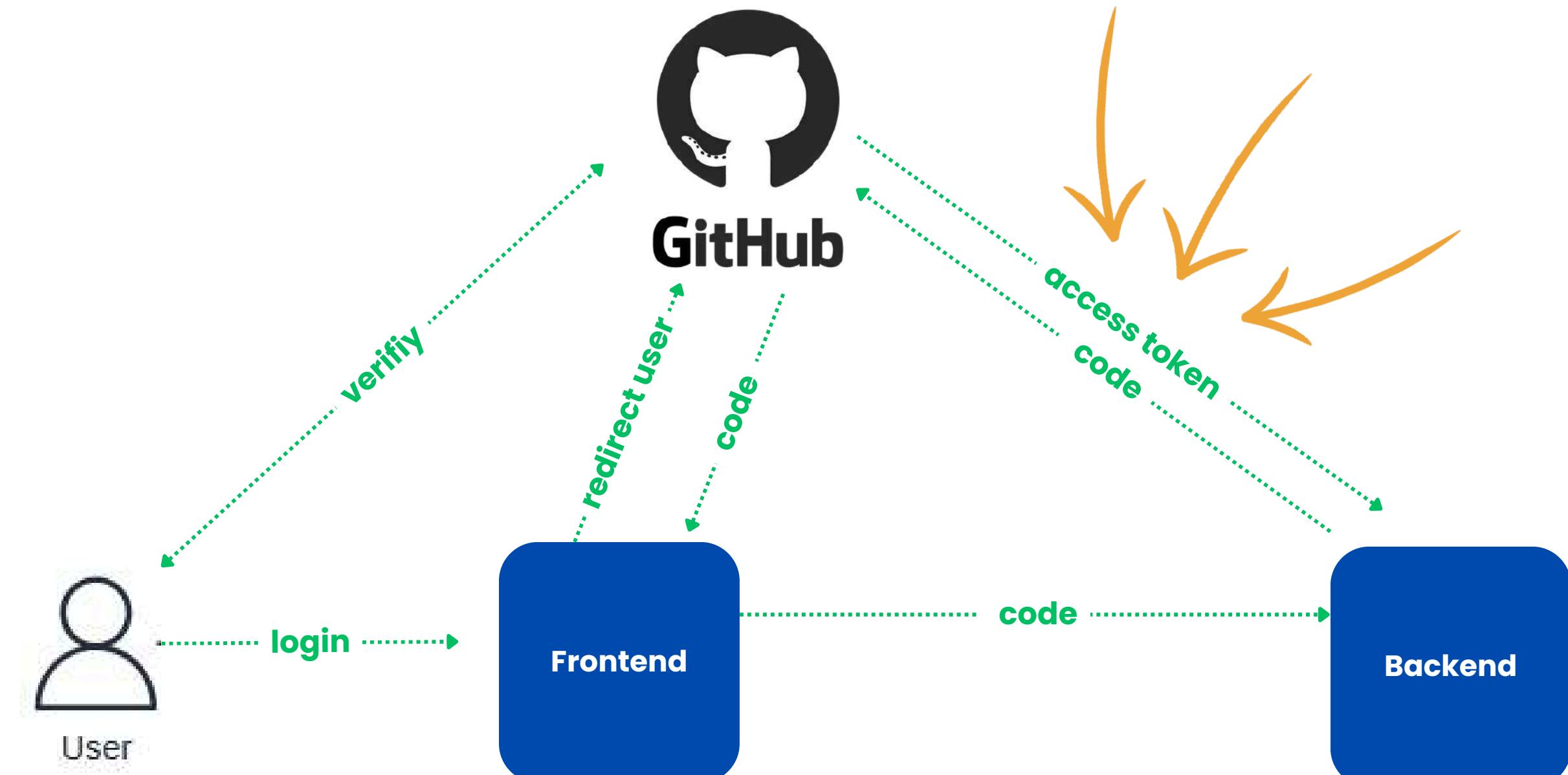
# AUTHENTICATION WORKFLOW



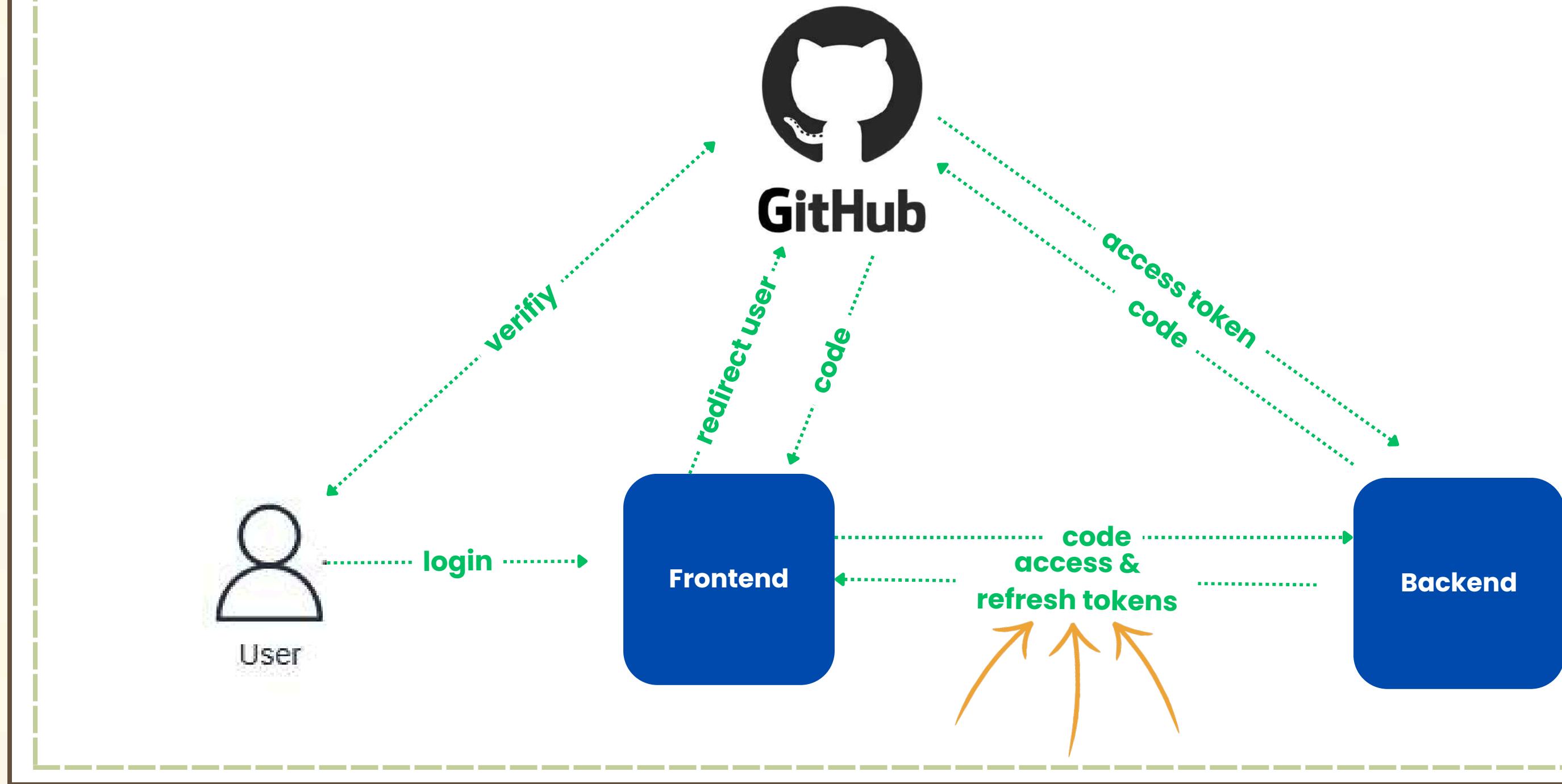
# AUTHENTICATION WORKFLOW



# AUTHENTICATION WORKFLOW



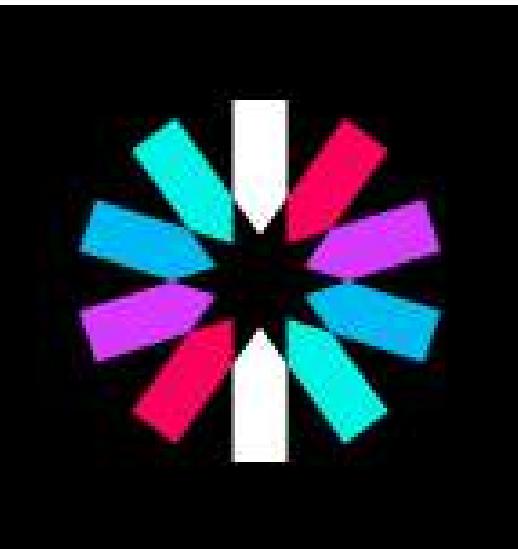
# AUTHENTICATION WORKFLOW



# AUTHENTICATION

Old:

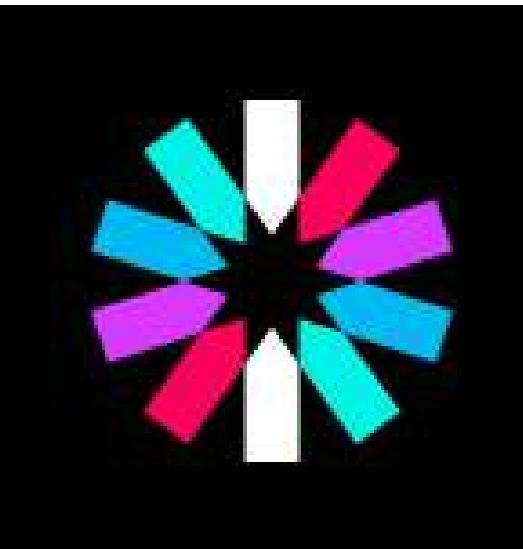
- Use one **JWT** token for authentication
- When **JWT** expires, user is asked to login



# AUTHENTICATION

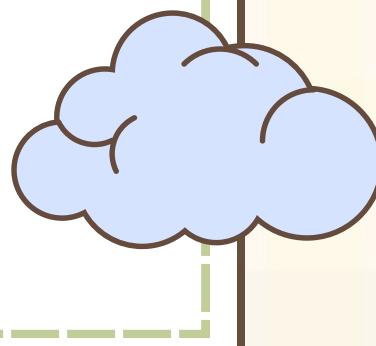
Old:

- Use one **JWT** token for authentication
- When **JWT** expires, user is asked to login



New:

- Use **refresh and access tokens** for authentication
- **Access token** is used to authenticate with backend
- **Refresh token** is used to renew **access token**
- When **Refresh token** expires, user is asked to login



# BETTER APPROACH



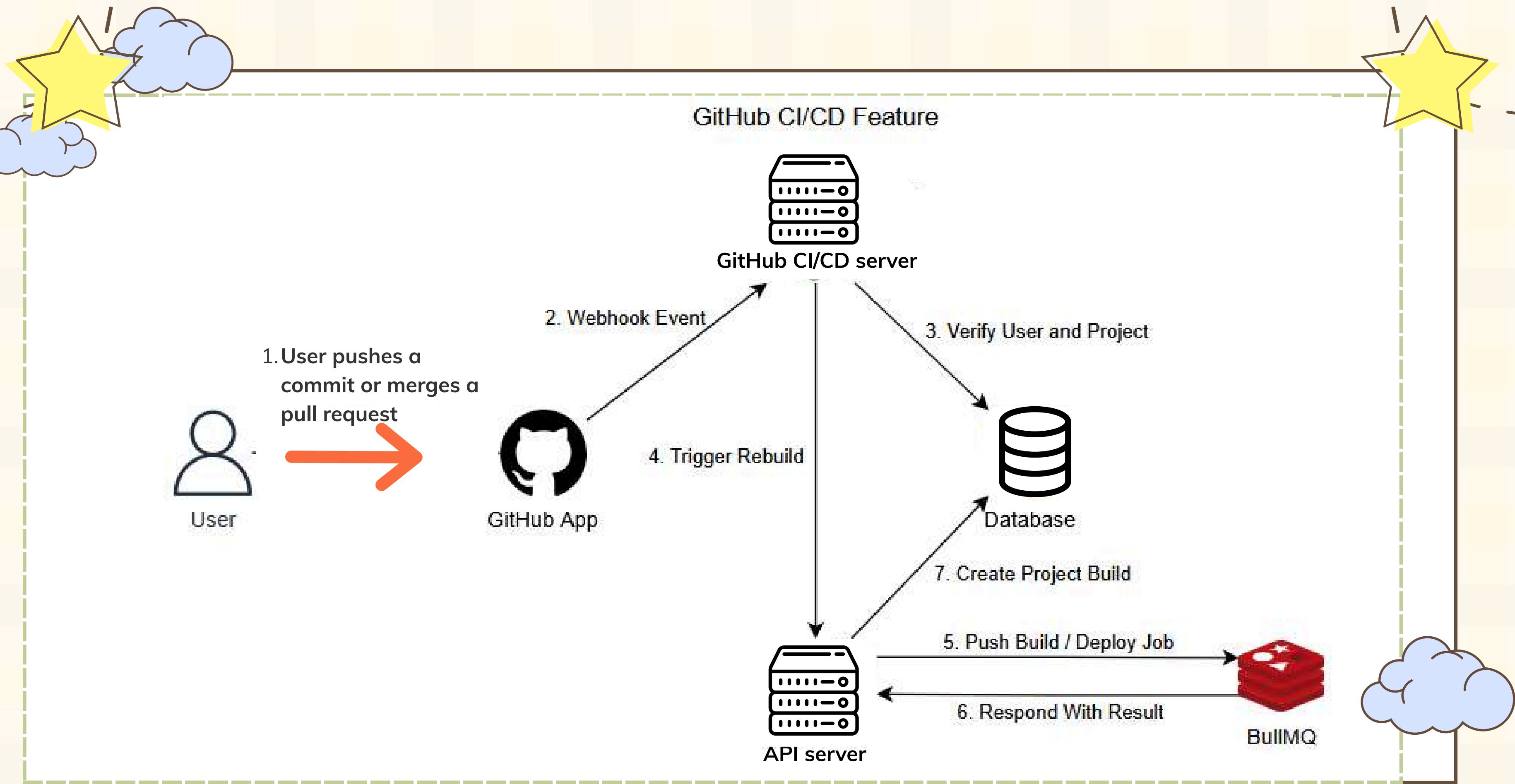
User experience

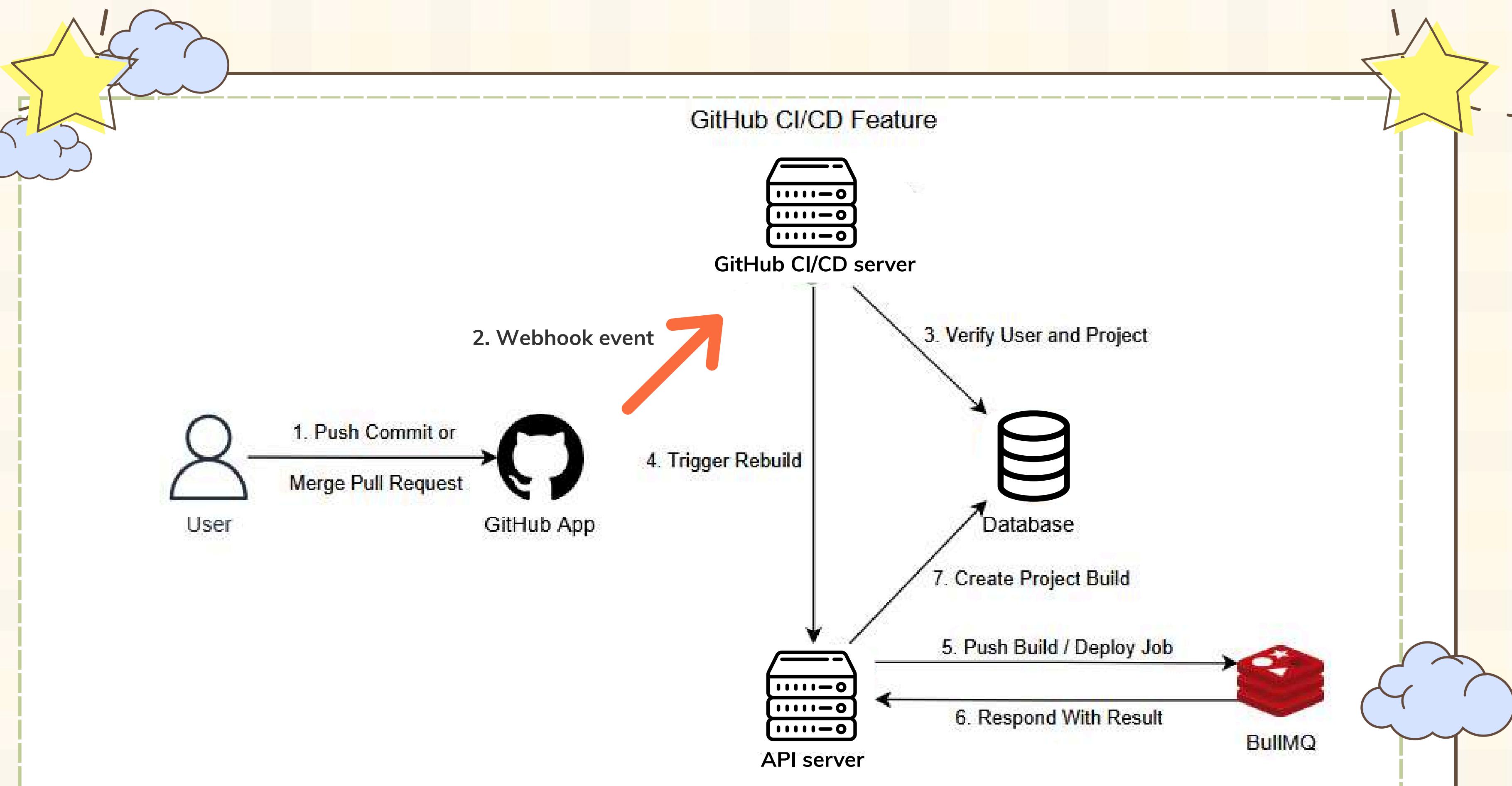


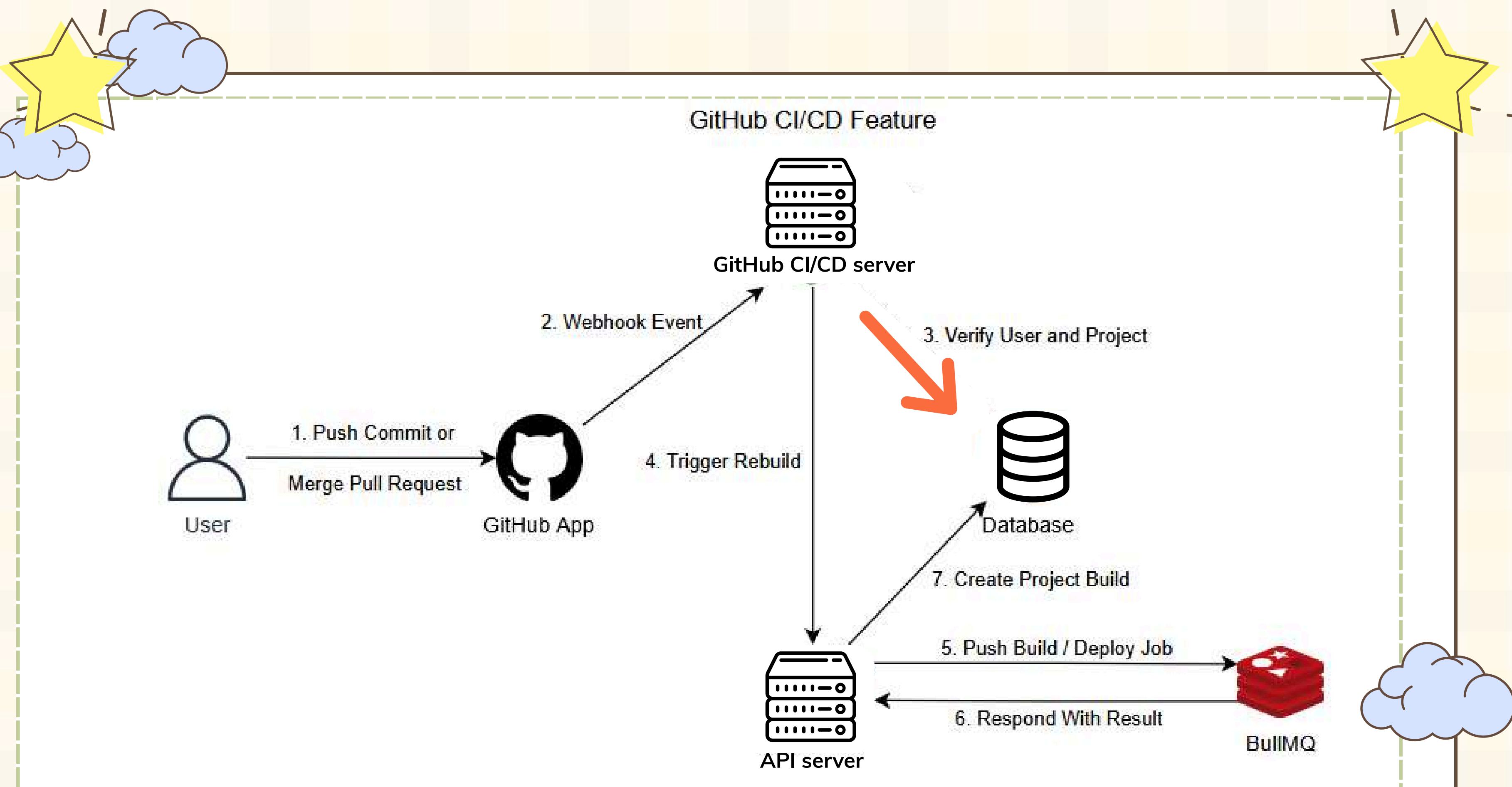
Enhance security

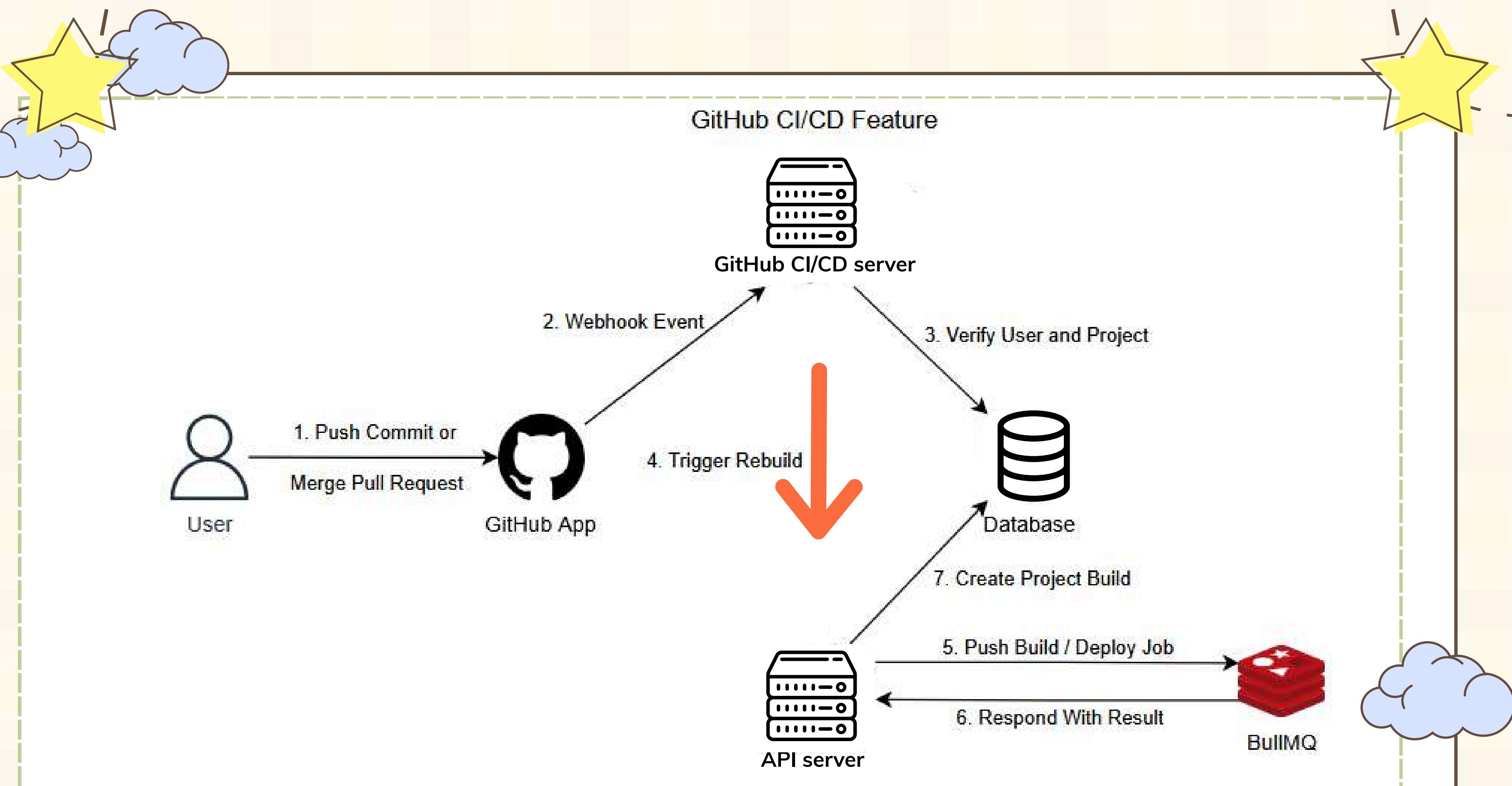


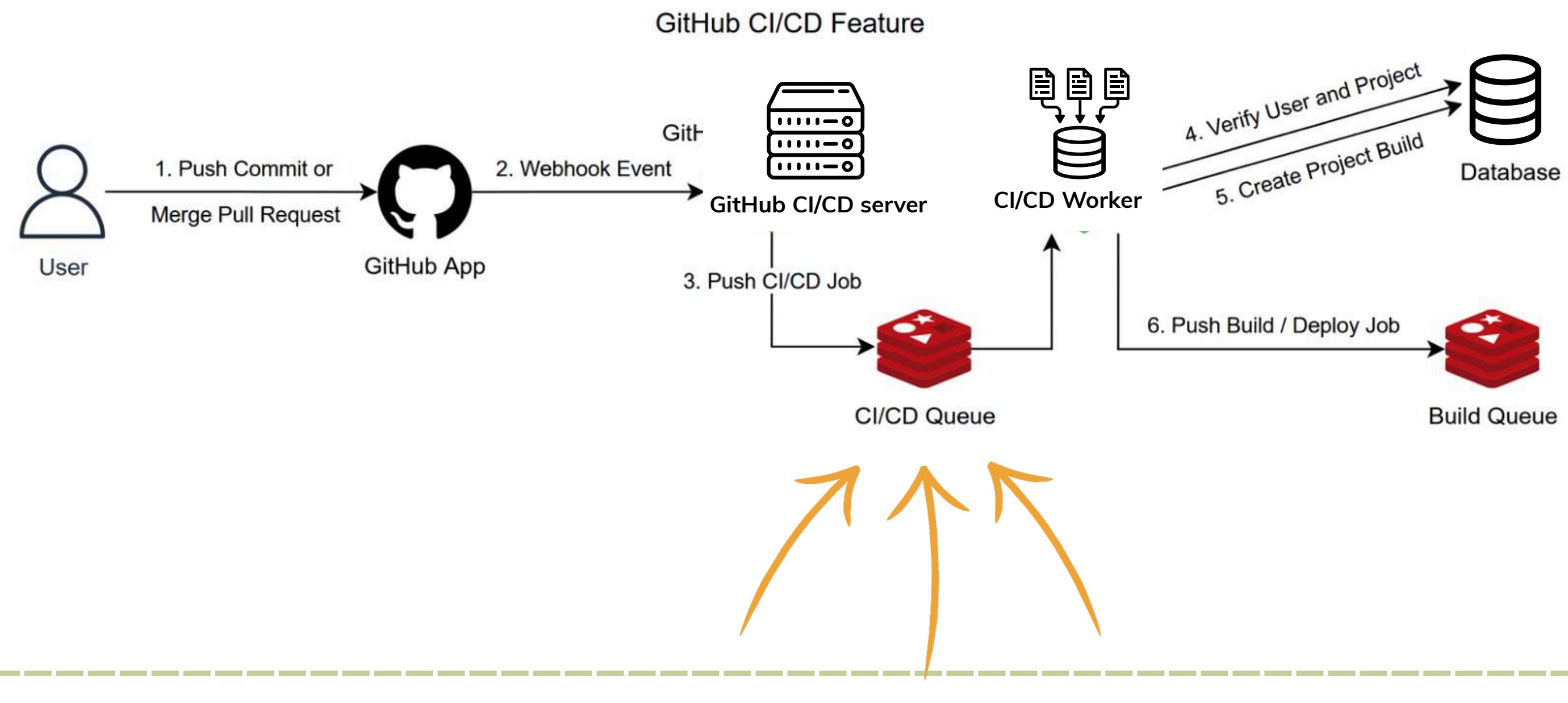
# CI/CD WITH UPDATES

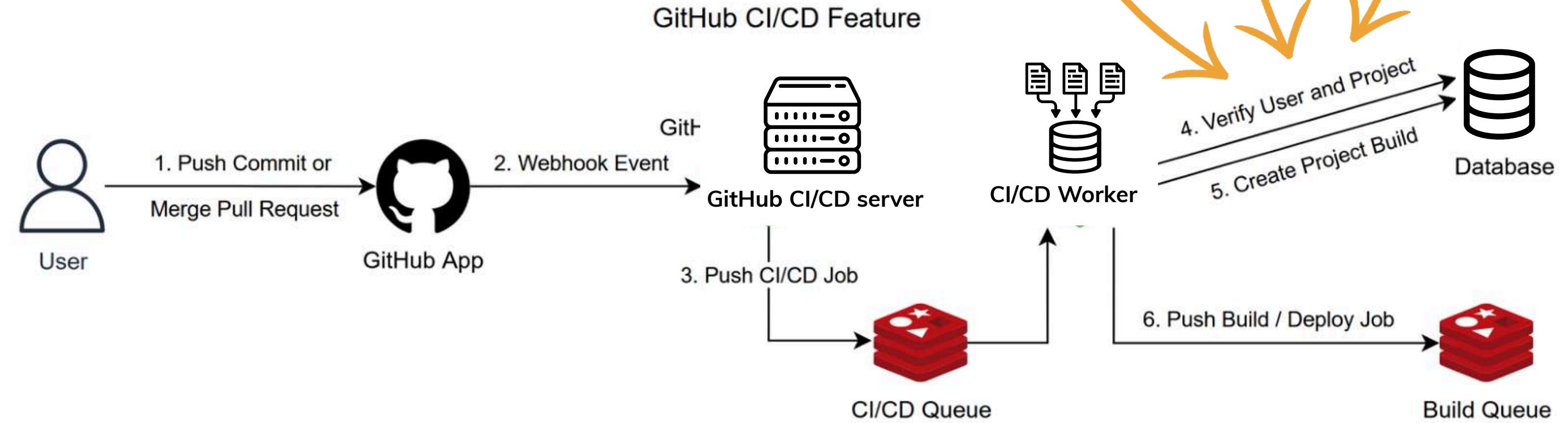








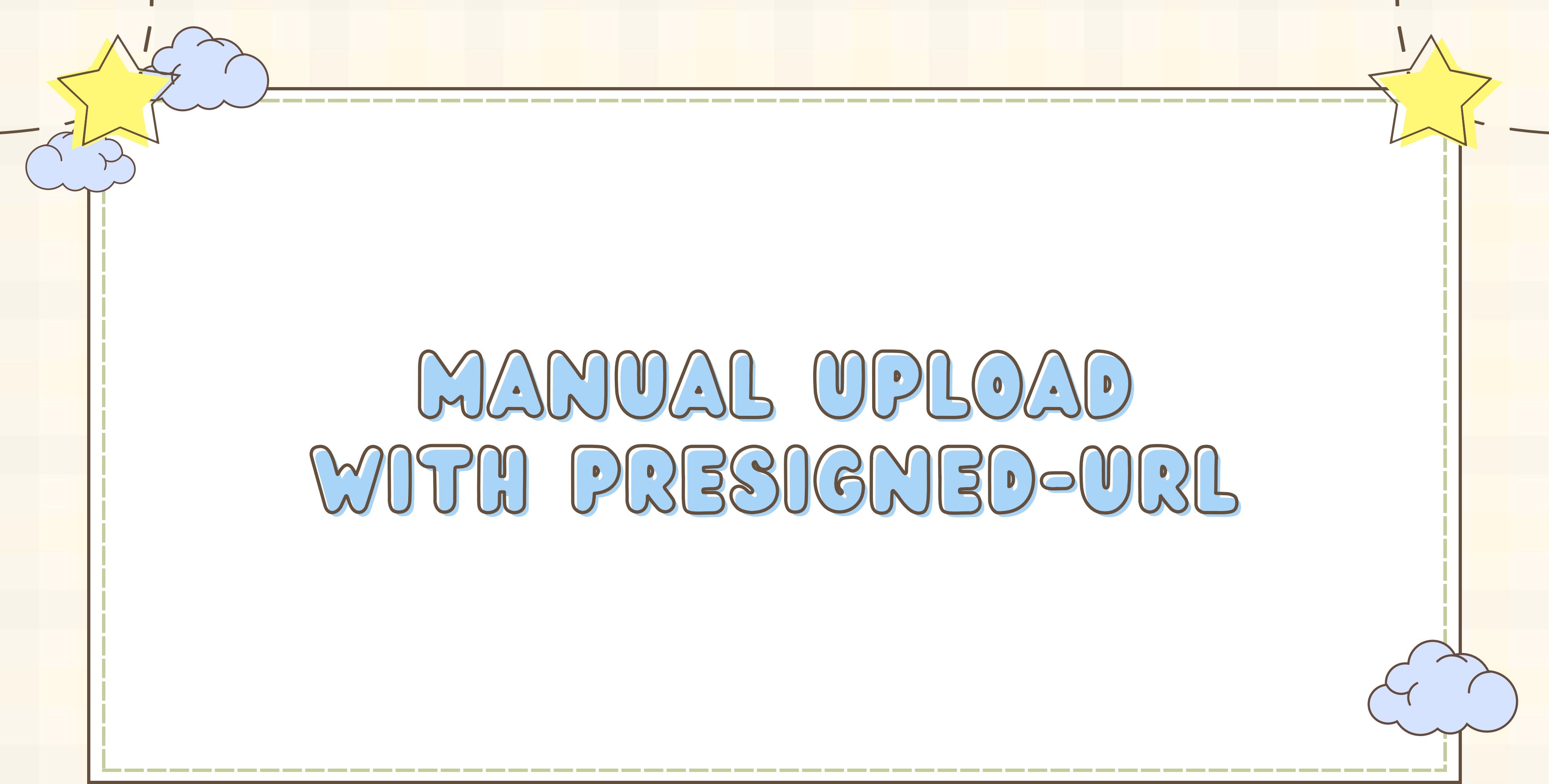




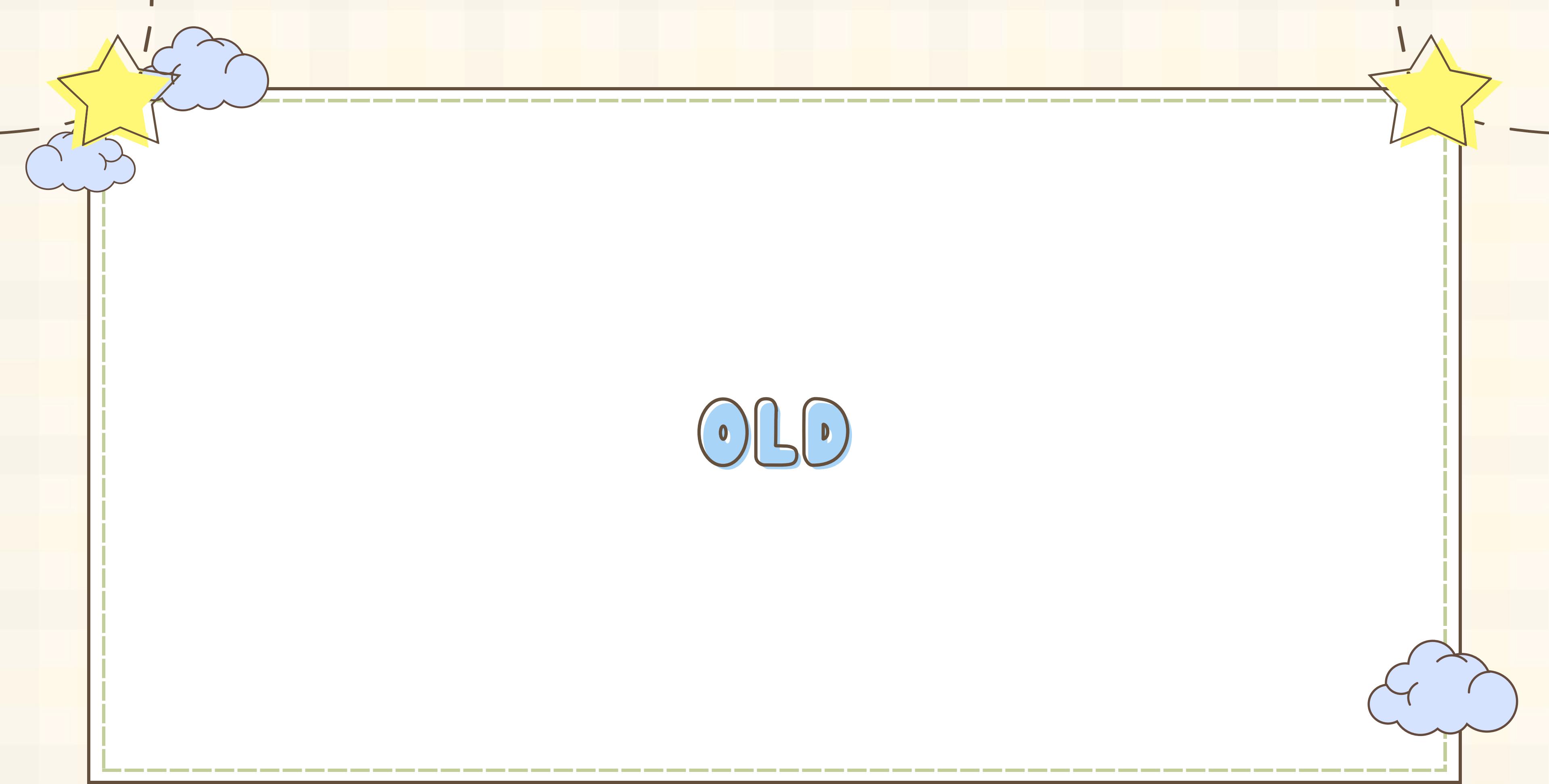


# WHY UPDATING IS BETTER ?

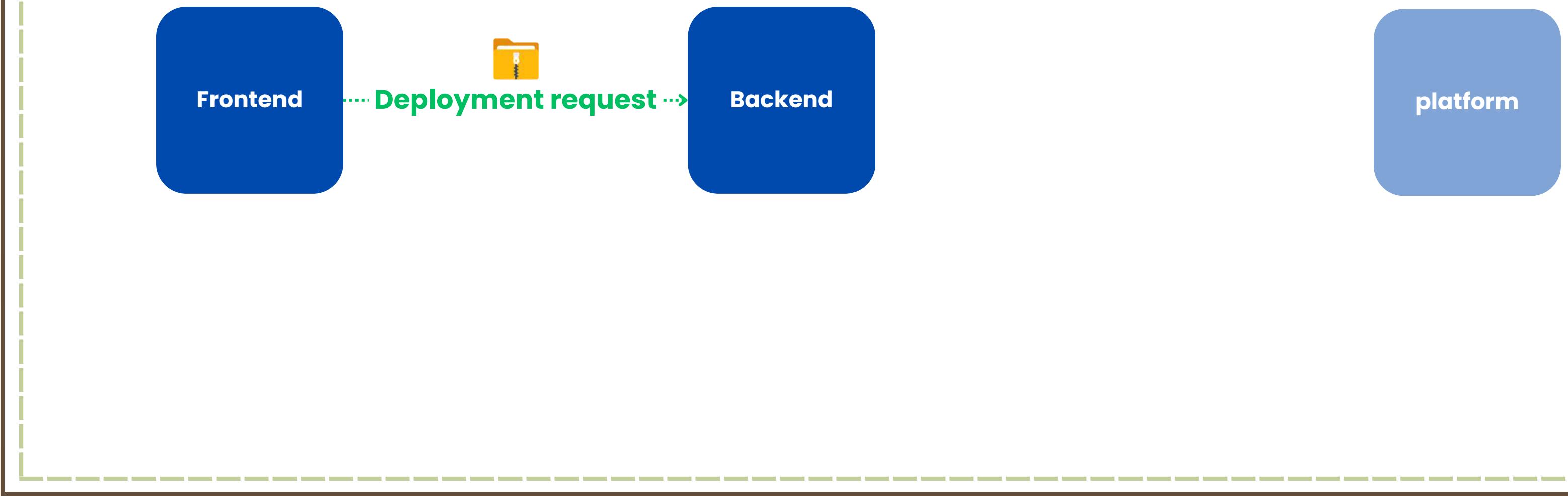
- IMPROVED RELIABILITY
- FAST GITHUB RESPONSE
- BETTER SCALABILITY
- SEPARATION OF CONCERNS

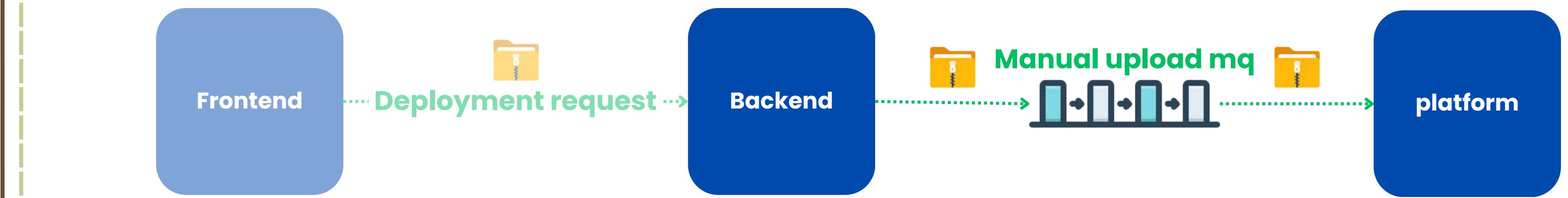


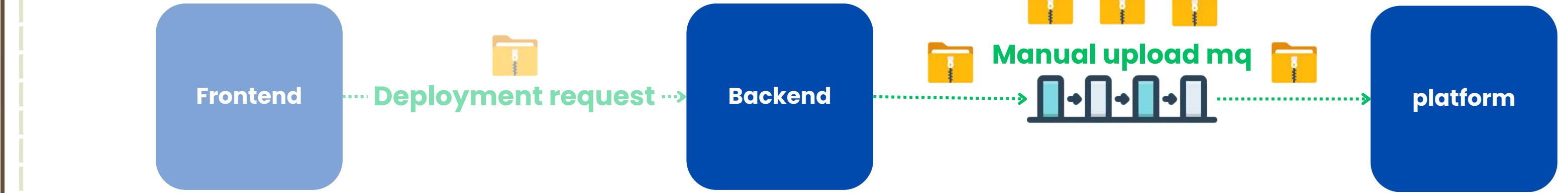
# **MANUAL UPLOAD WITH PRESIGNED-URL**



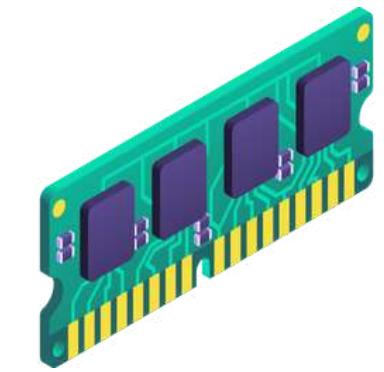
OLD







# BAD APPROACH



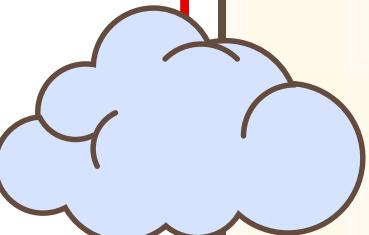
Memory-based



performance degradation



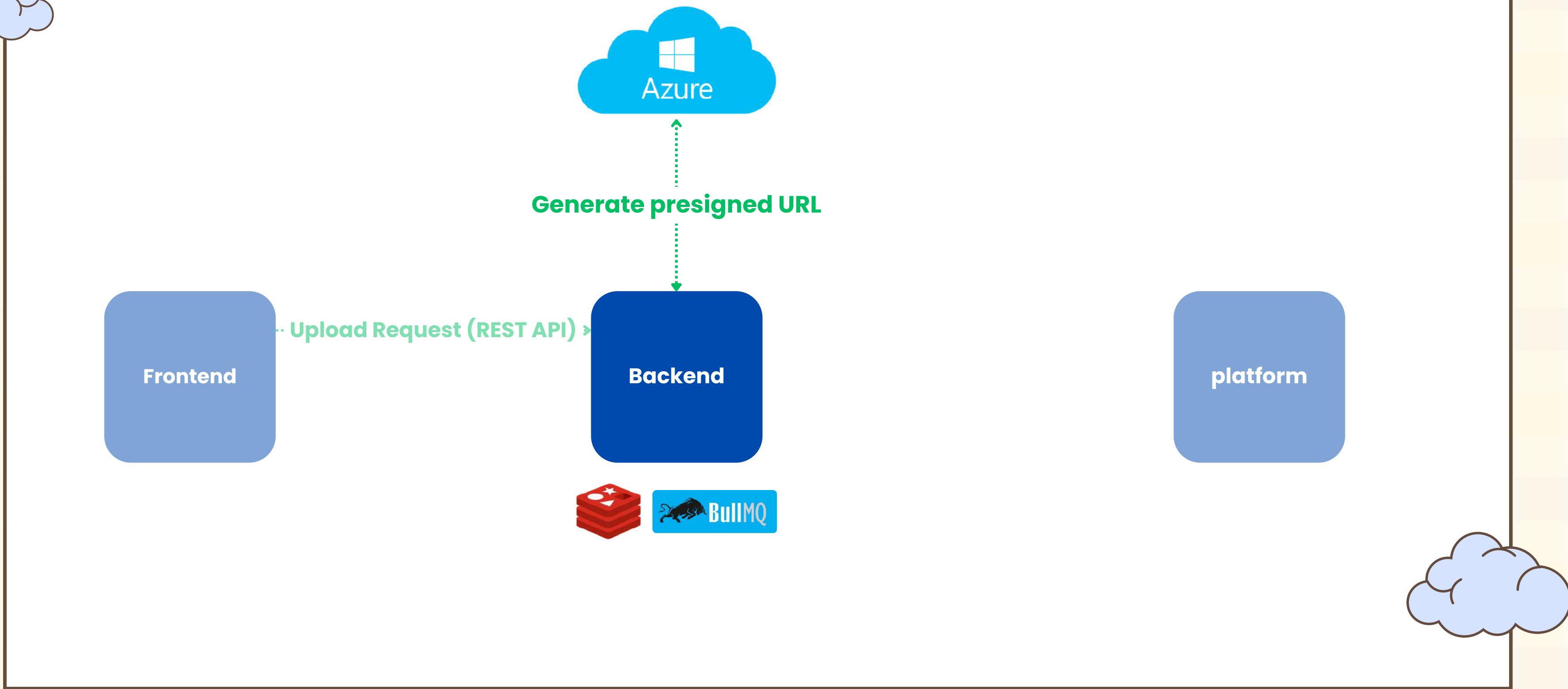
Less users

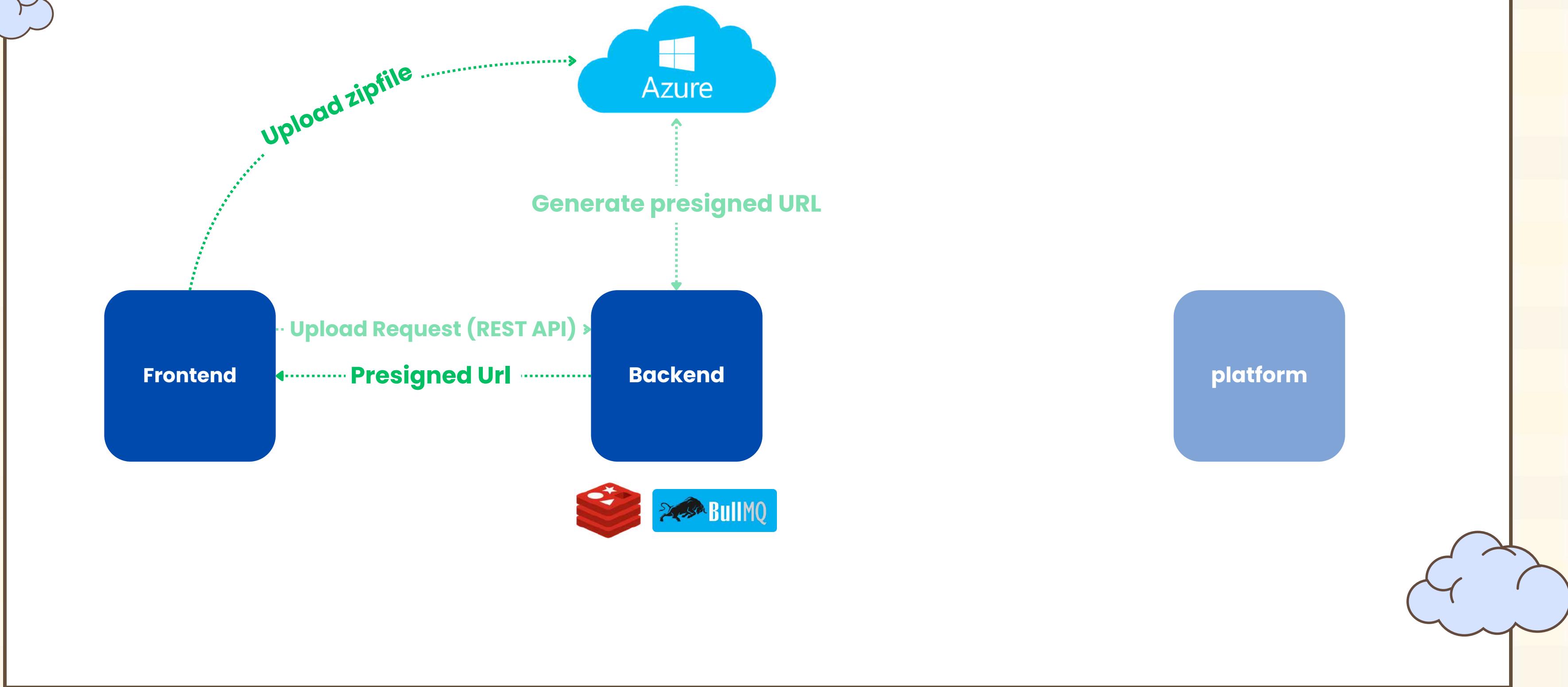


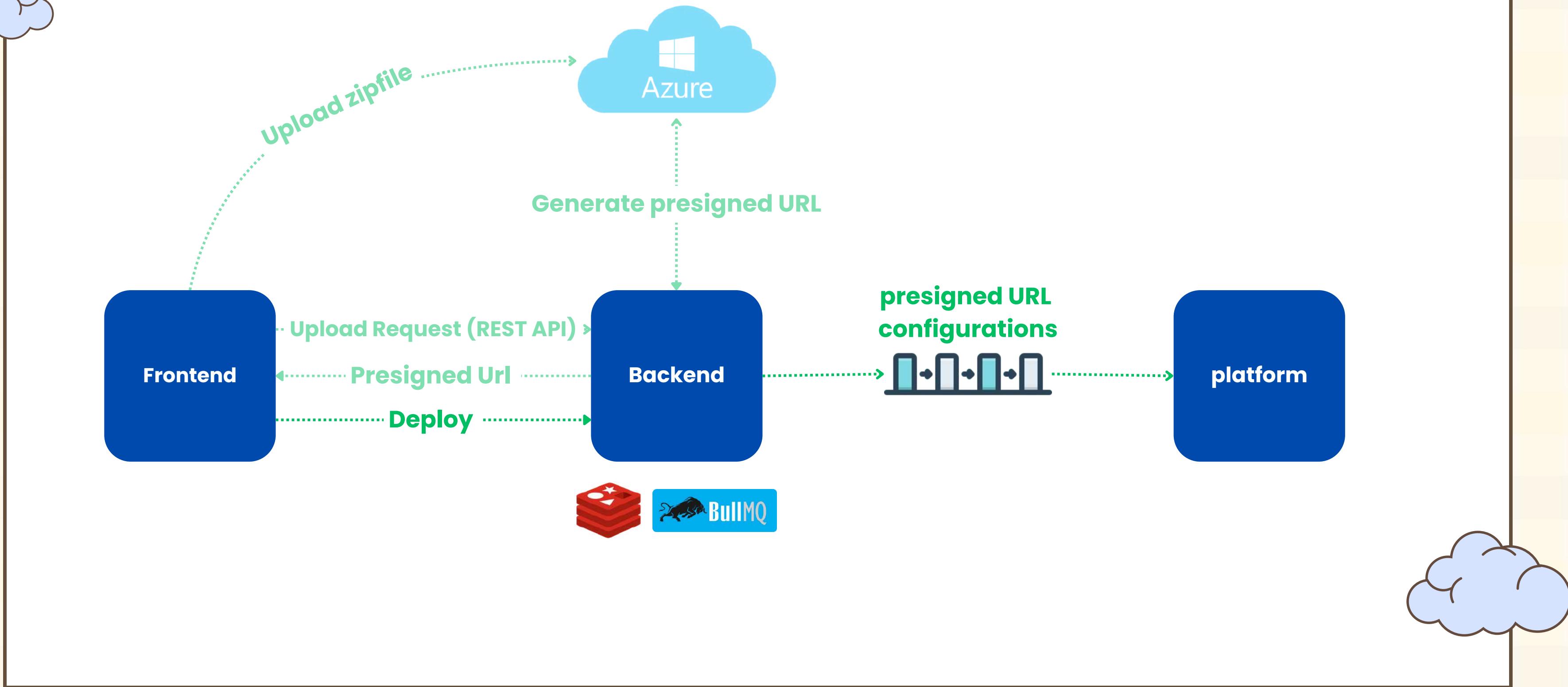


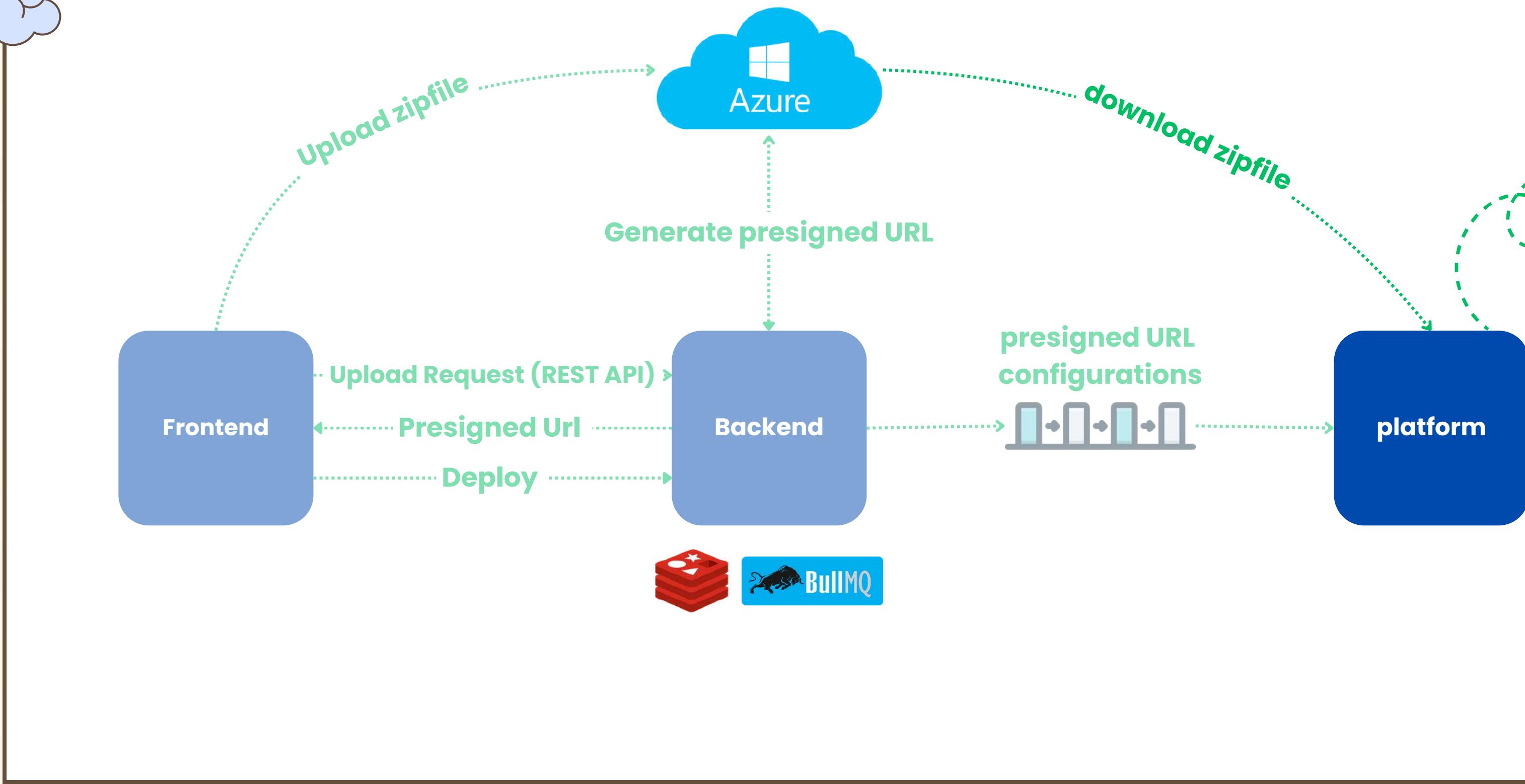
**WHATS NEW !!**











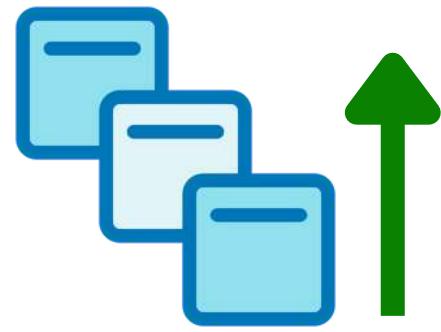
# BETTER APPROACH



Lightweight transfer



Reduce usage



More jobs

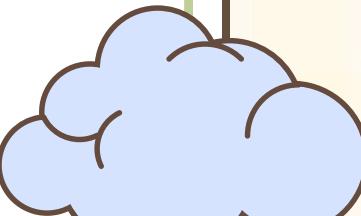


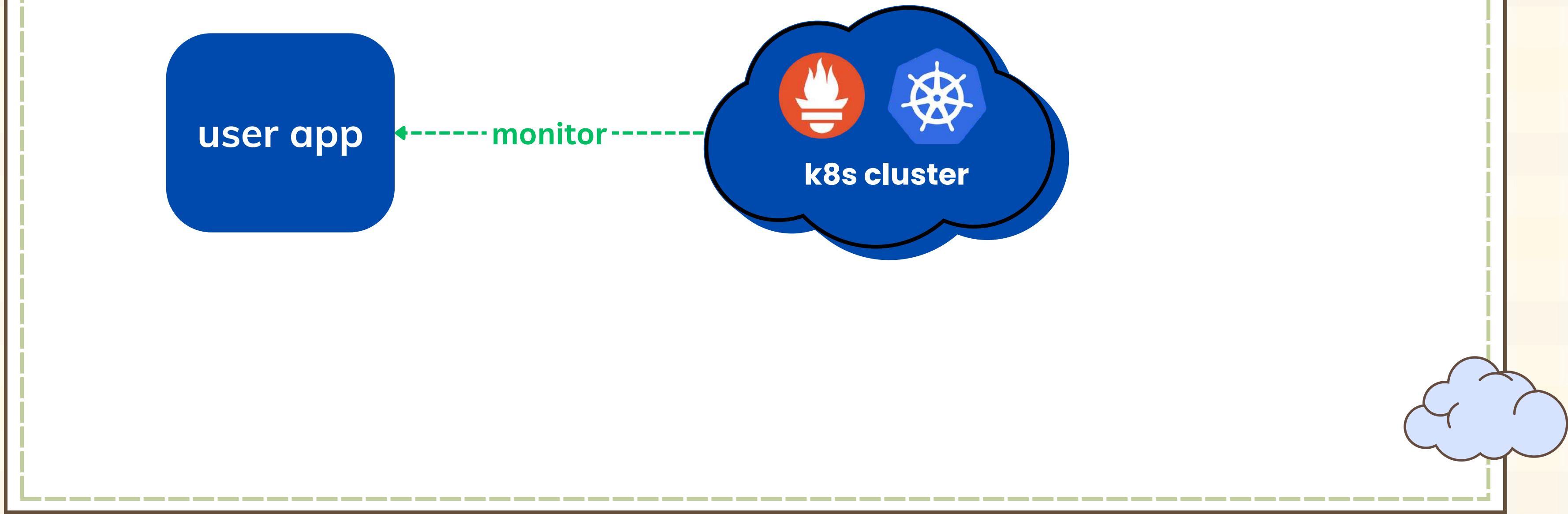
# MONITORING

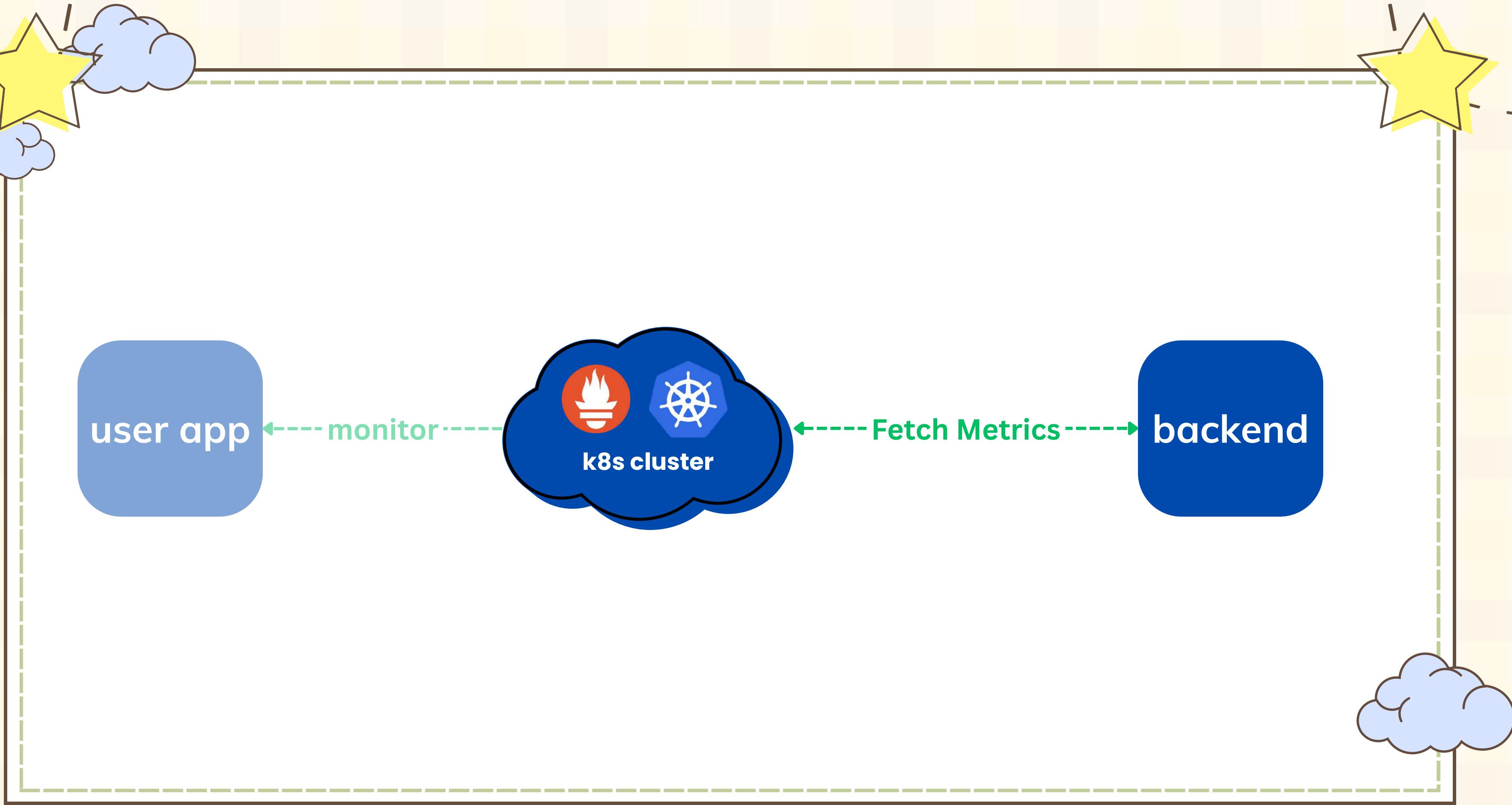
# PROMETHEUS

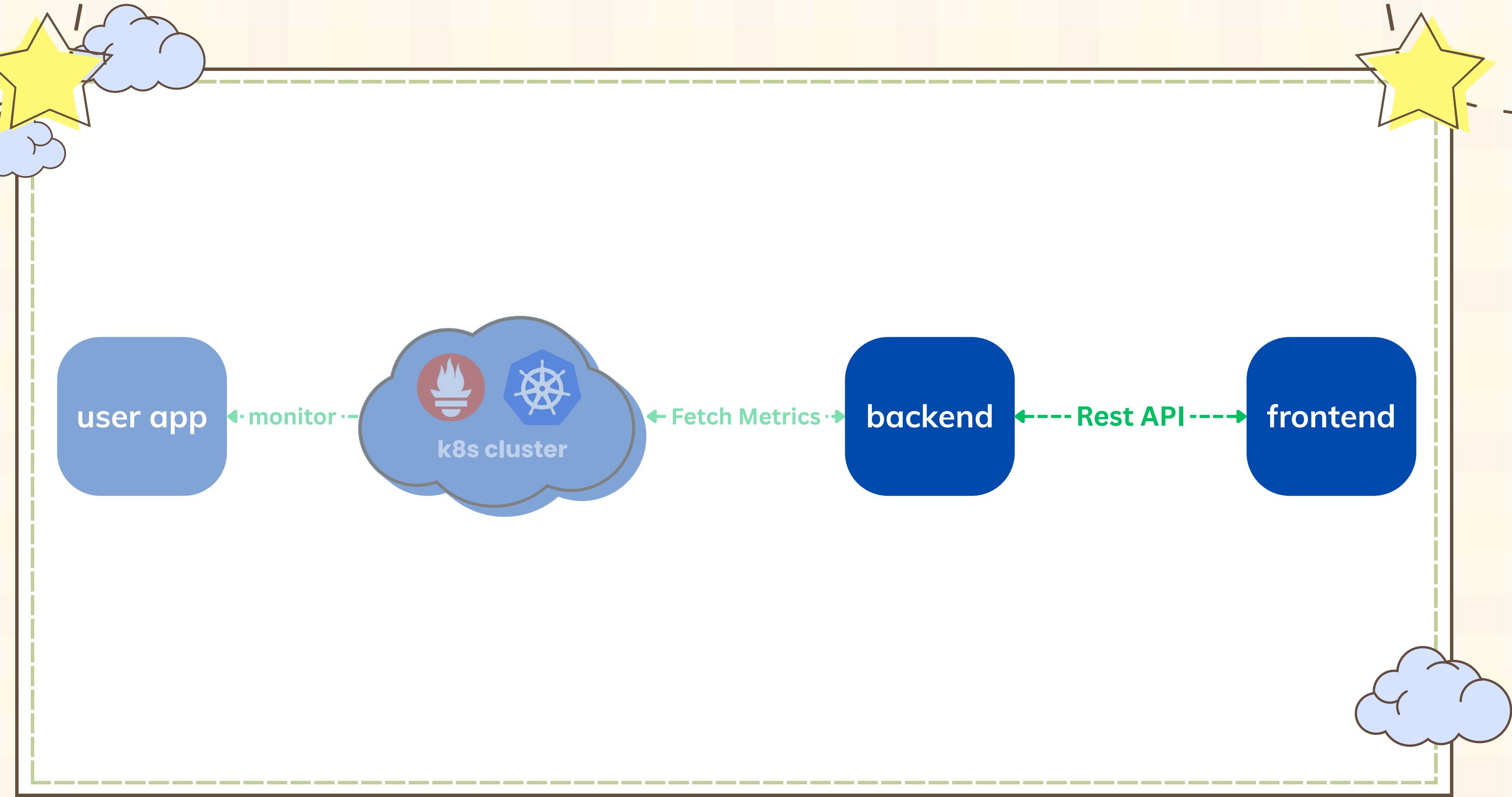


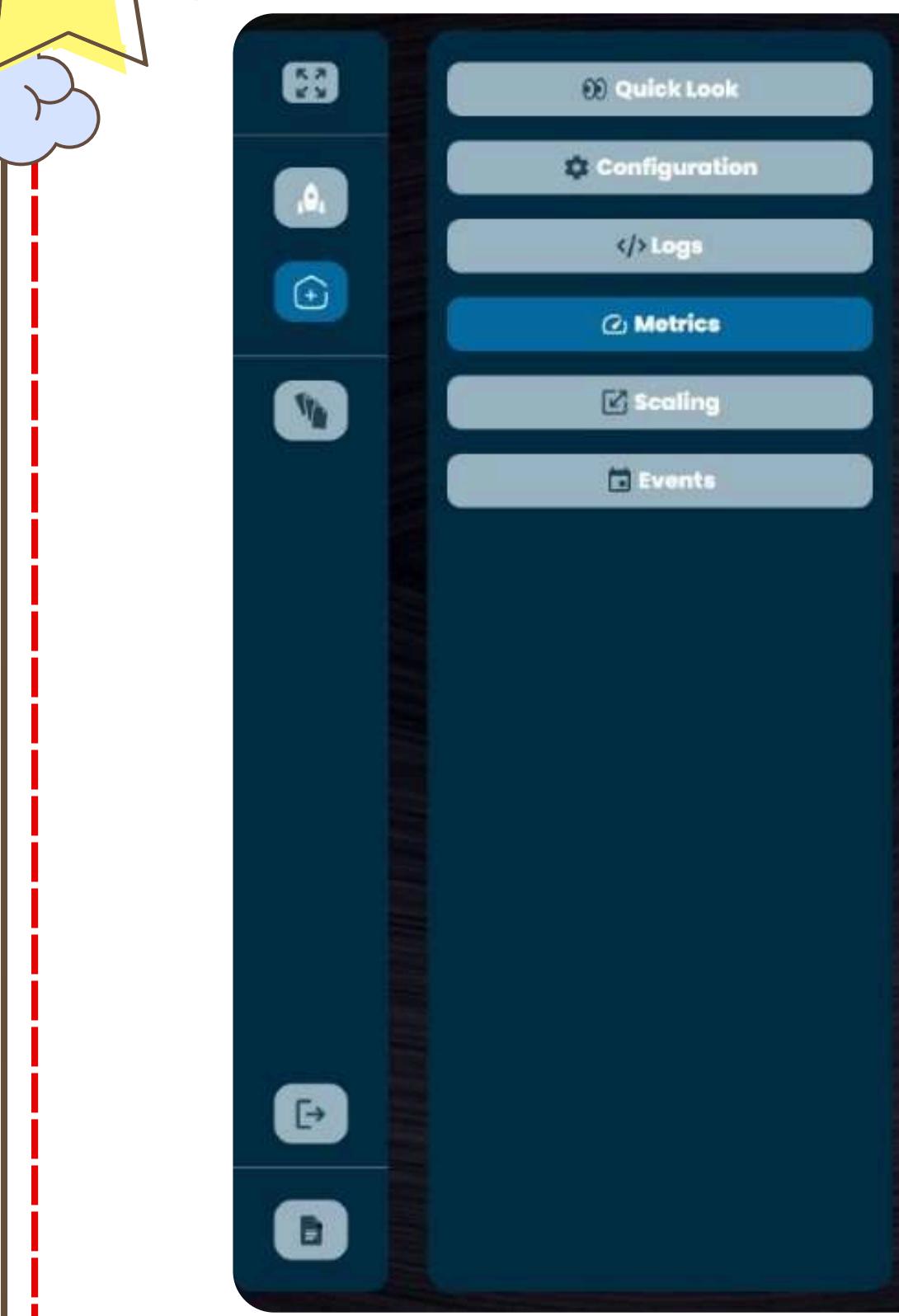
A monitoring tool used to monitor resource usage and performance metrics for dynamic projects.











A screenshot of the [synfig](http://st-c846c0c54600f7c681c020617abb2f...) website. The page has a light blue header with the title "synfig", a GitHub link, and a photo placeholder. Below the header is a "Live Monitoring" button. The main section features a large title "System Metrics" in bold blue and purple letters, with the subtitle "Advanced real-time performance monitoring with intelligent analytics". Two cards are displayed: "CPU Usage" (Processor Performance) showing 22% usage with an "Optimal" status, and "Memory Usage" (RAM Performance) showing 22% usage with an "Optimal" status. The background of the main content area is a light blue gradient.

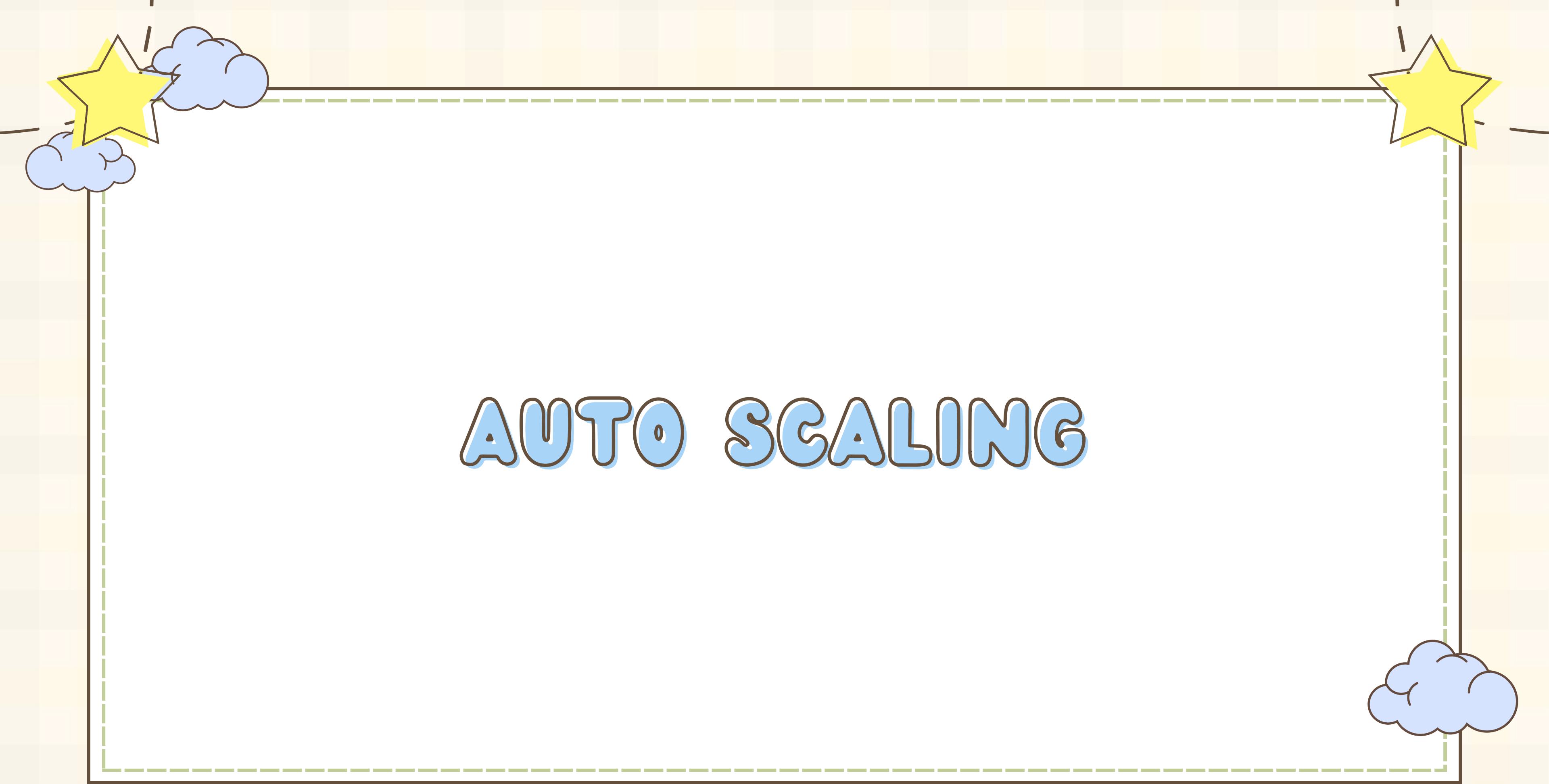
# ADVANTAGES



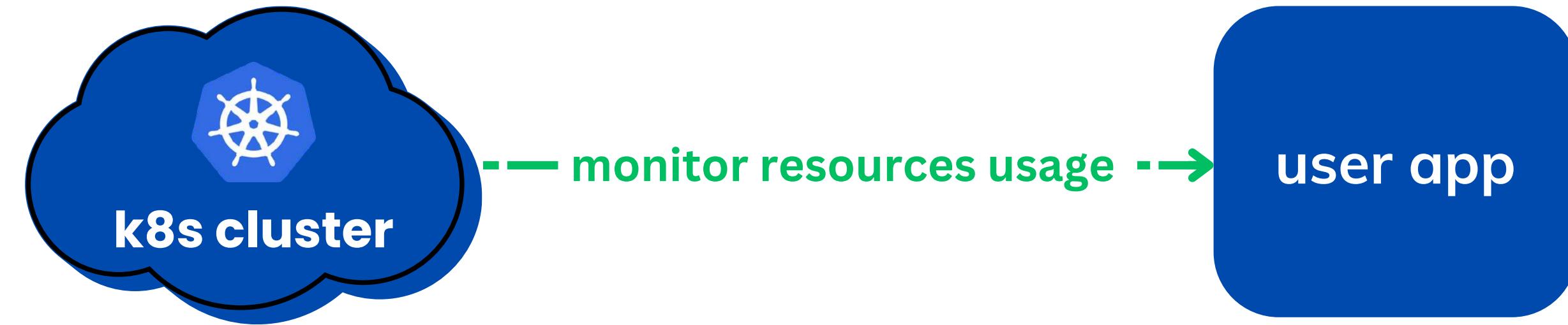
Real-Time Visibility

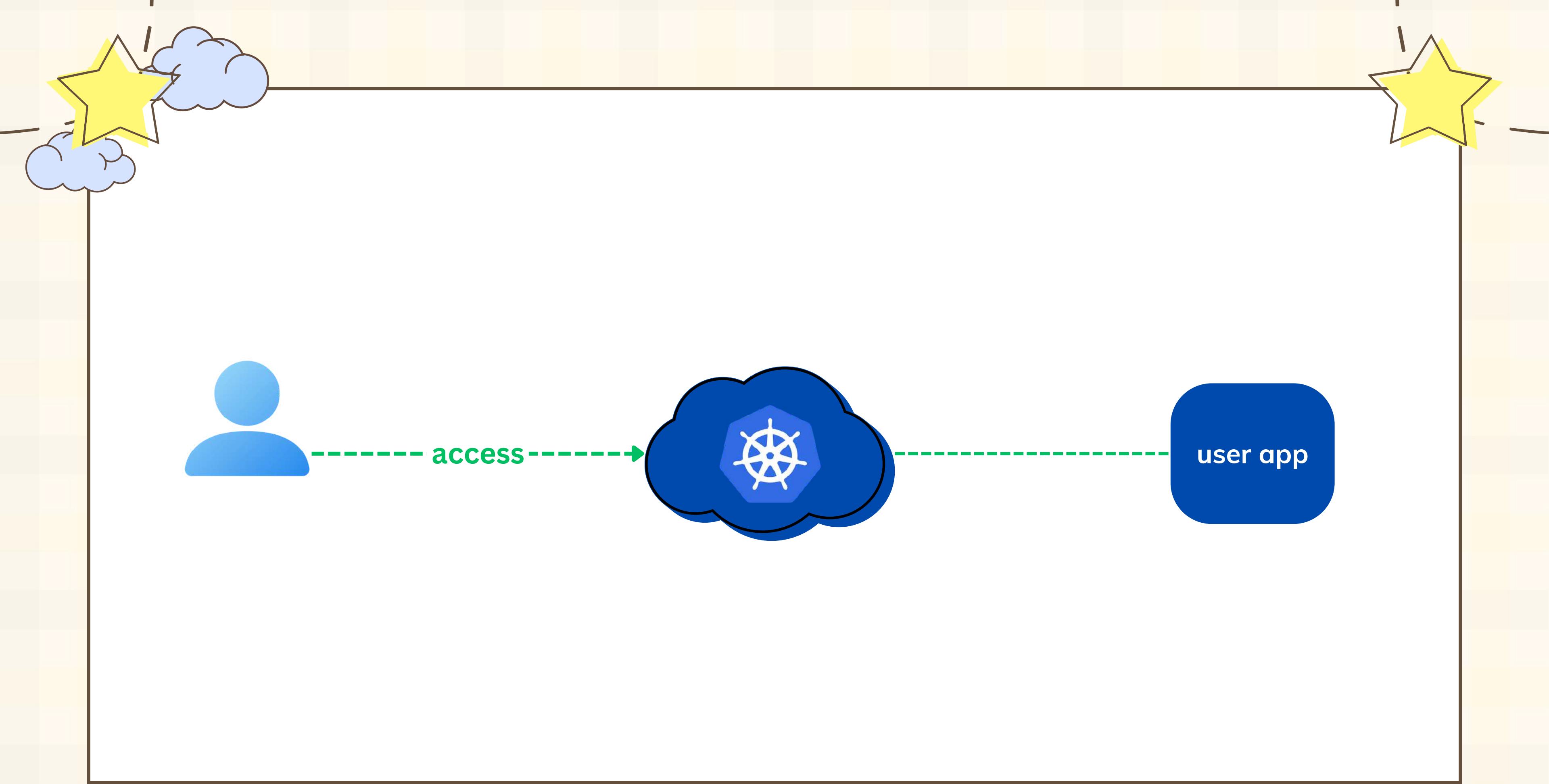


Troubleshooting Insights

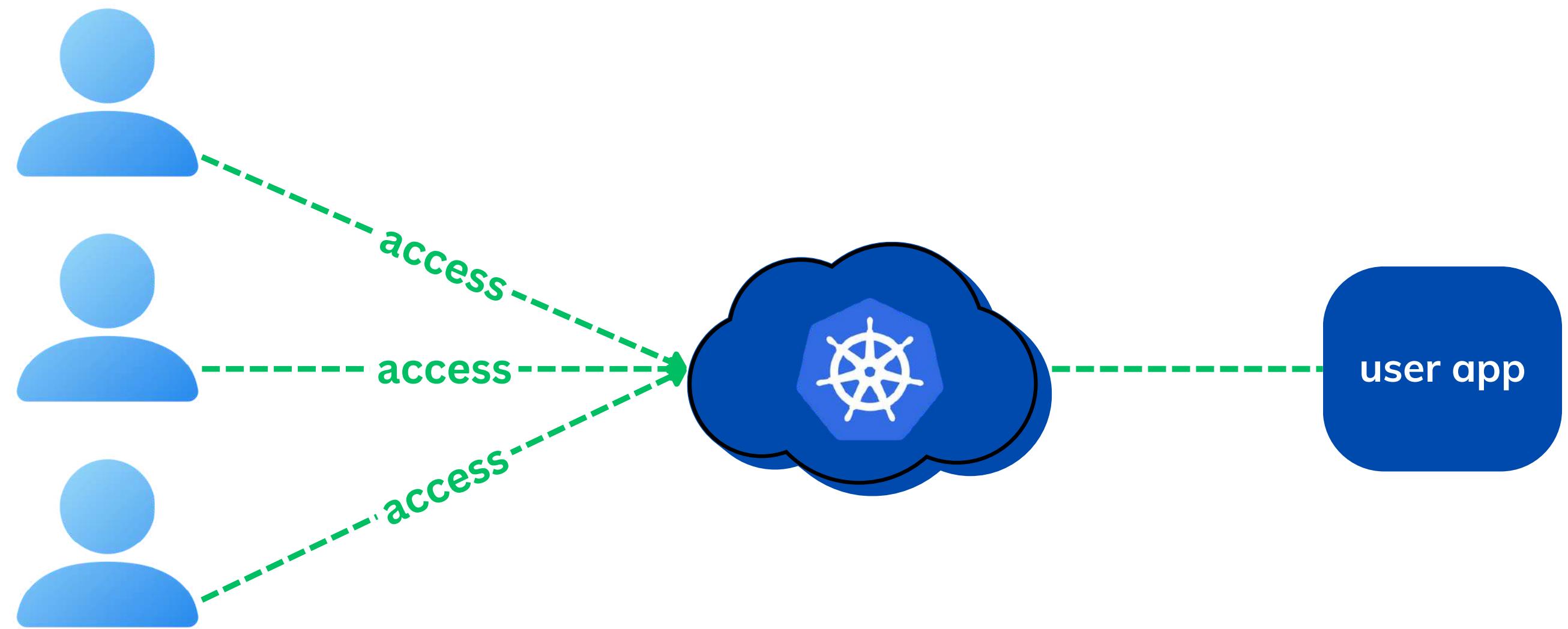


# AUTO SCALING

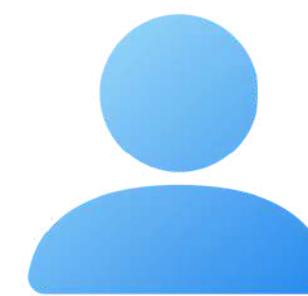




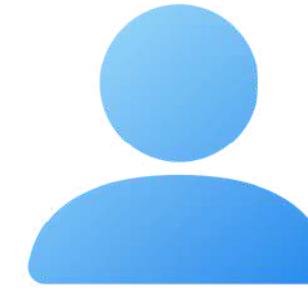
# SCALE UP



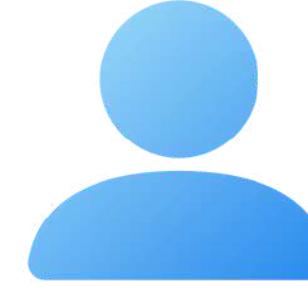
# SCALE UP



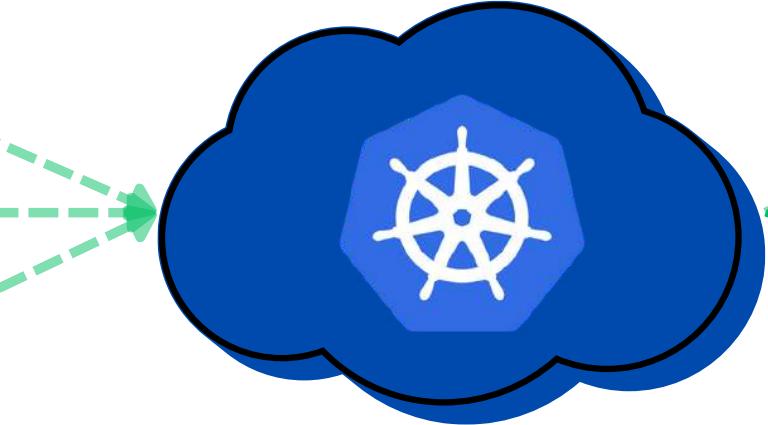
*access*



*access*



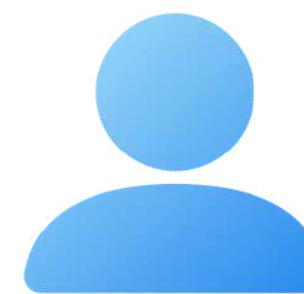
*access*



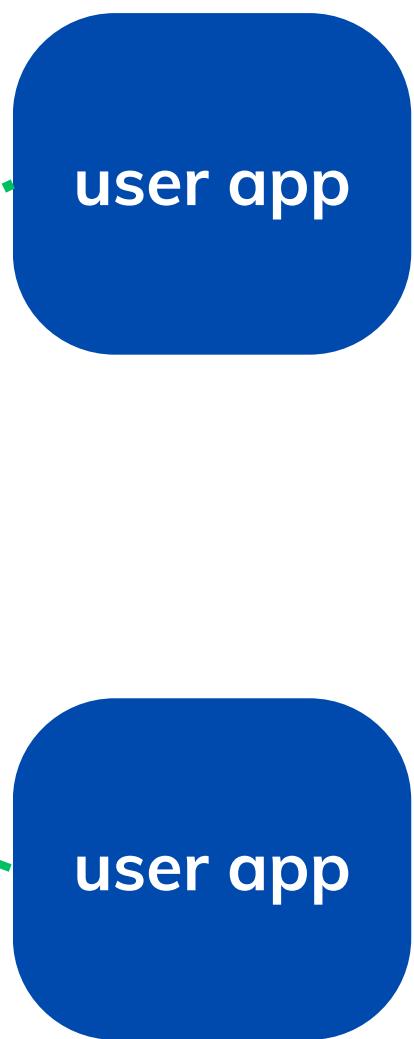
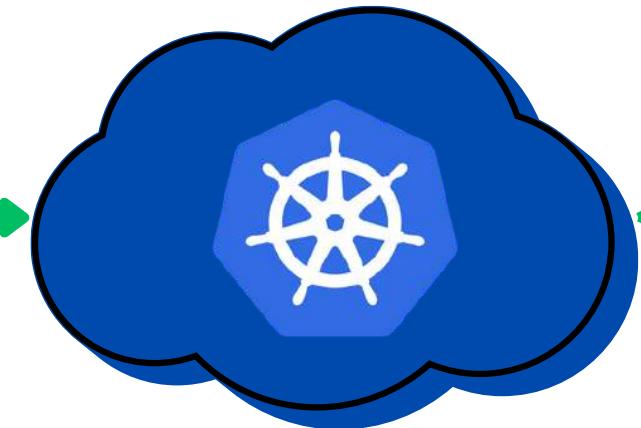
*create new instance*



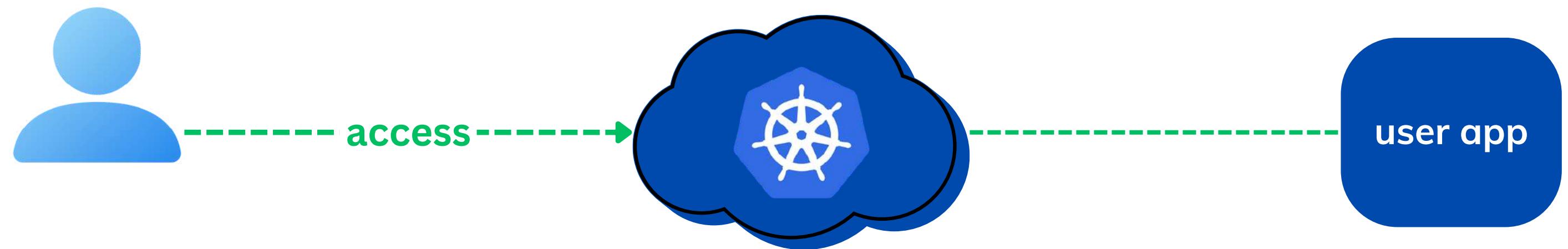
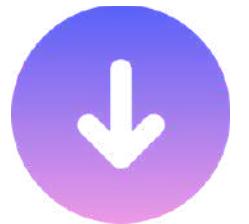
# SCALE DOWN



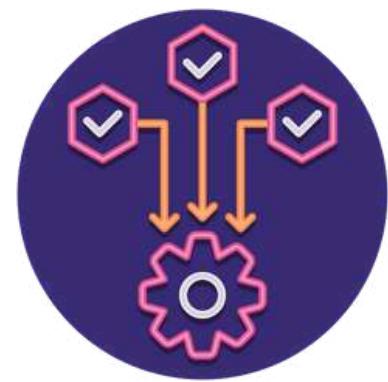
access



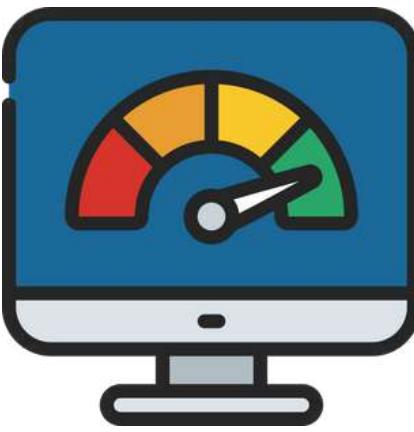
# SCALE DOWN



# ADVANTAGES



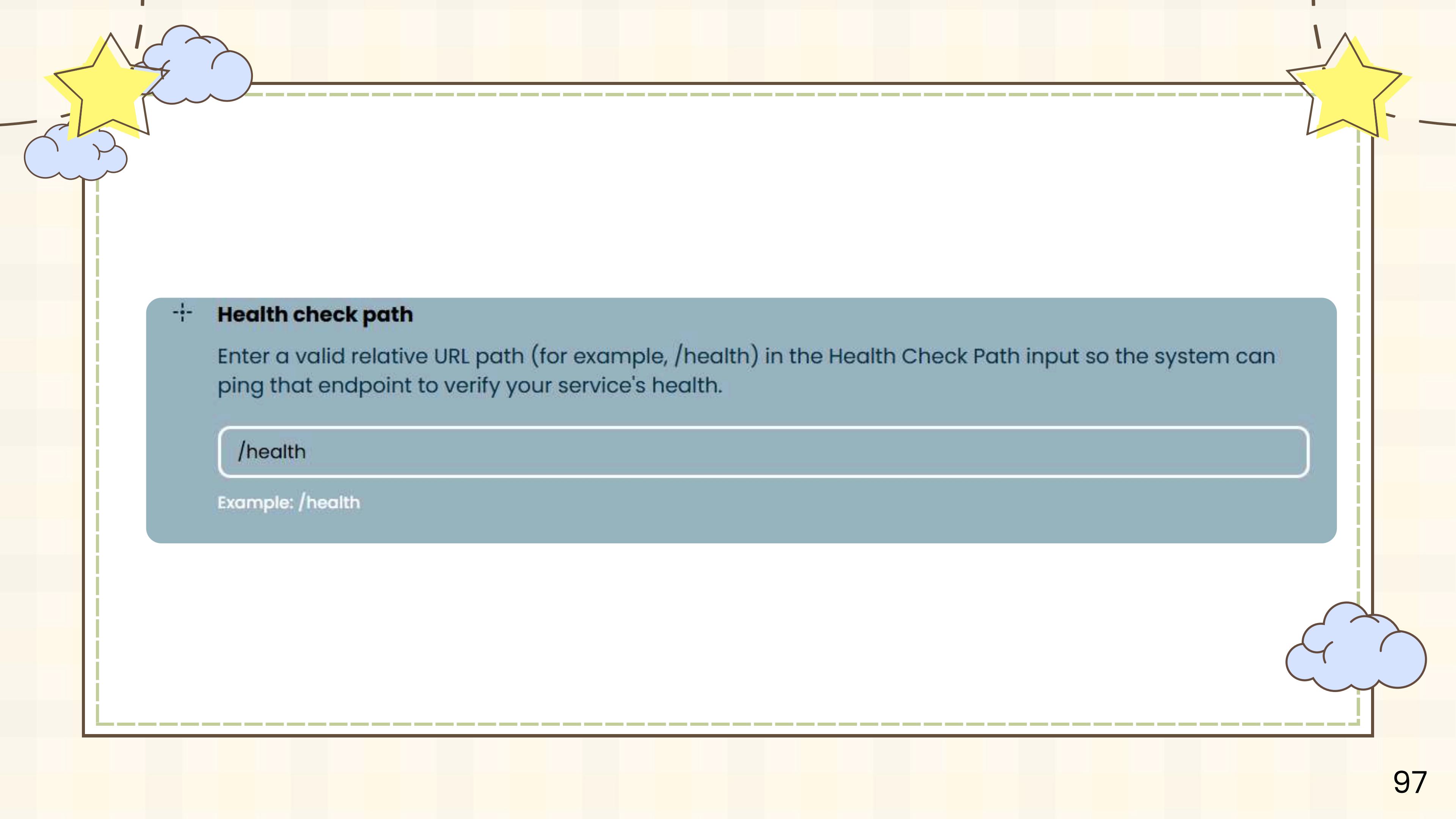
Efficient Resource Usage



Optimized Application  
Performance



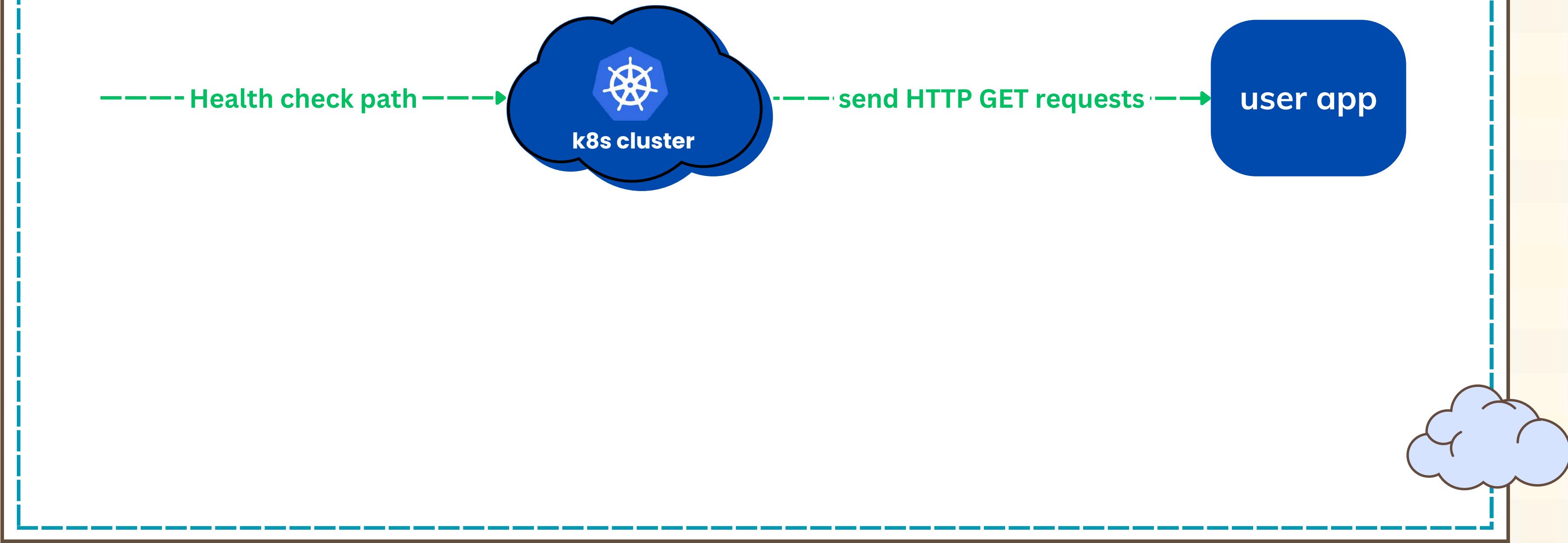
# HEALTH CHECK



## **⊕ Health check path**

Enter a valid relative URL path (for example, /health) in the Health Check Path input so the system can ping that endpoint to verify your service's health.

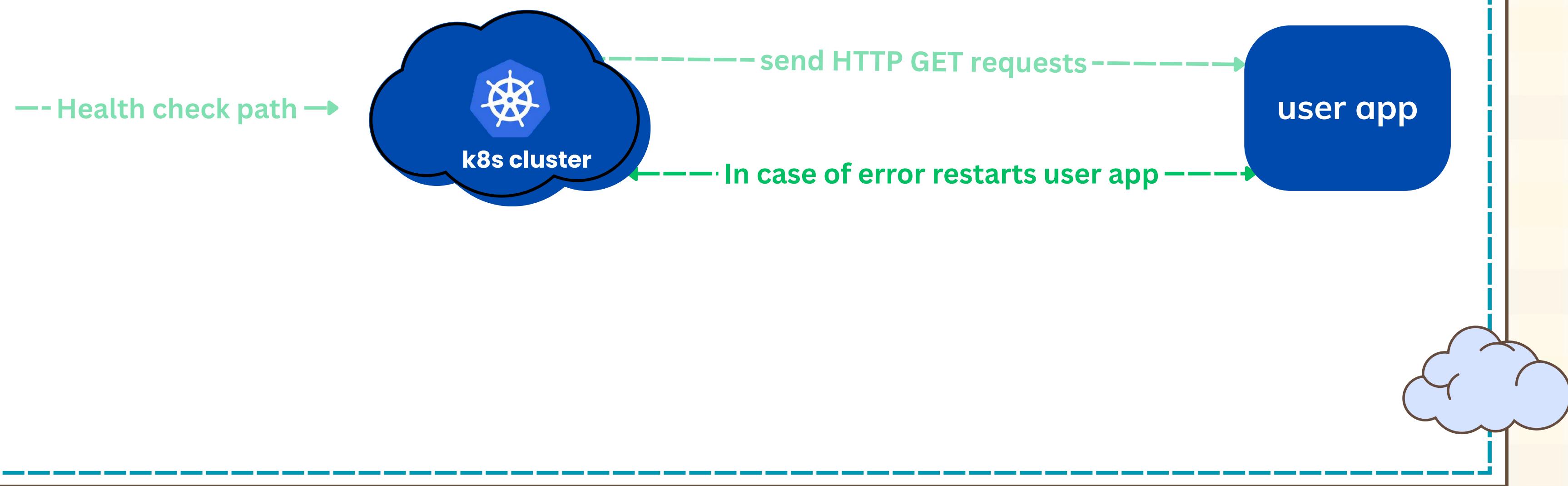
Example: /health



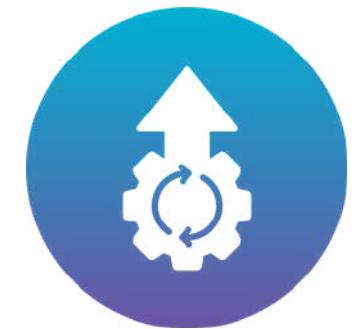
# HAPPY SCENARIO



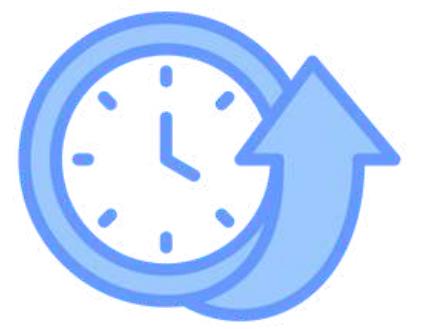
# FAILURE SCENARIO



# ADVANTAGES



Automatic Failure Recovery



Increased Uptime



Improved System  
Reliability



**BUILD ROLLBACK**

# BUILD ROLLBACK

What happens when the user deploys broken code?

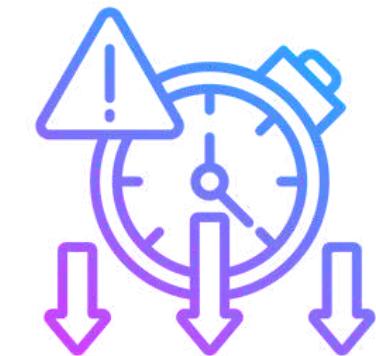
## KEY BENEFITS:



Failure  
recovery

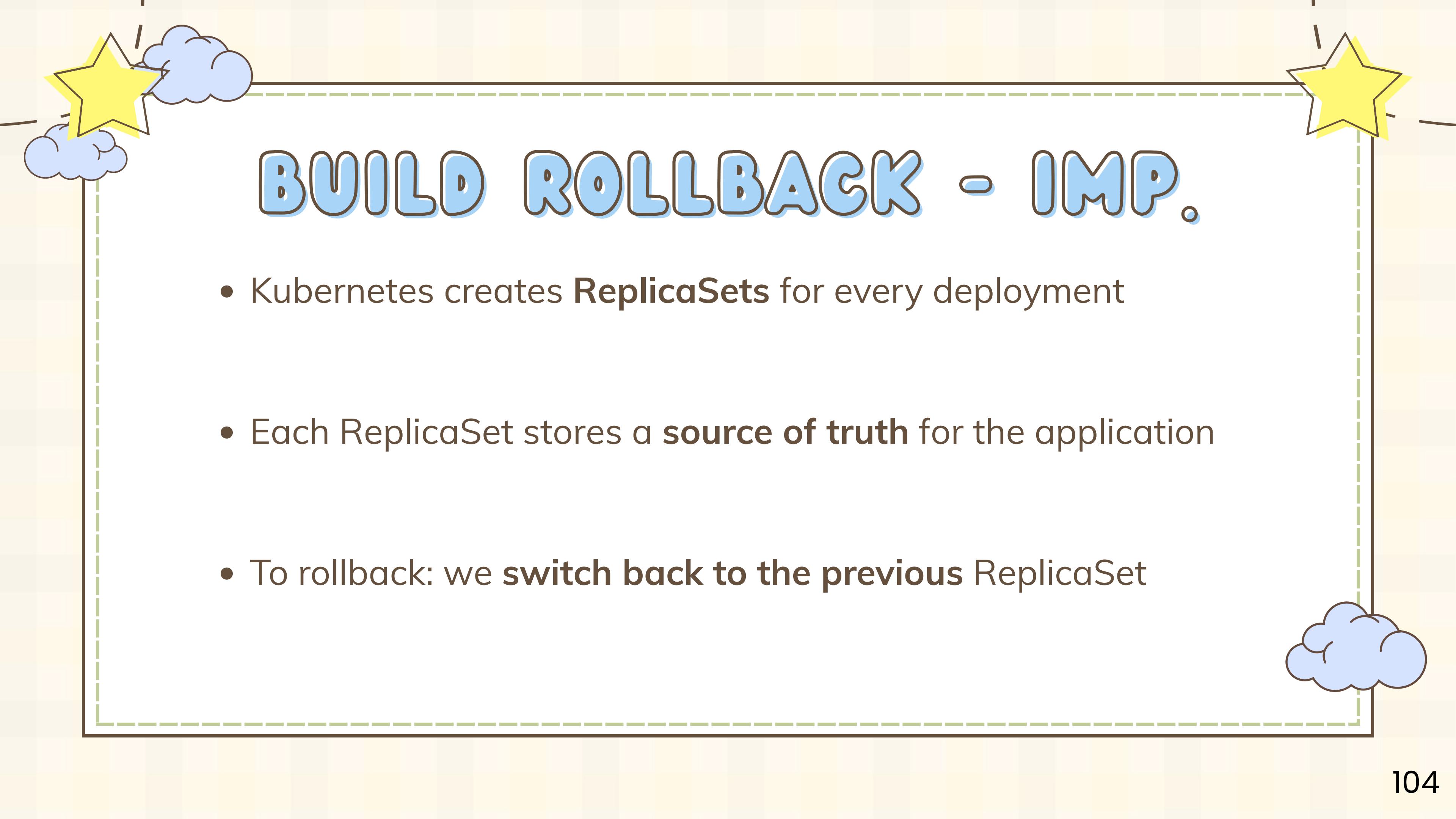


Minimal effort  
from the user



Low downtime  
(~5 seconds)





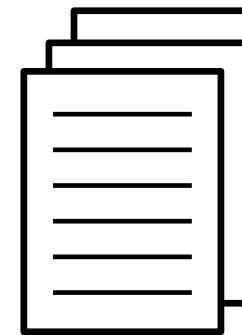
# BUILD ROLLBACK - IMP.

- Kubernetes creates **ReplicaSets** for every deployment
- Each ReplicaSet stores a **source of truth** for the application
- To rollback: we **switch back to the previous** ReplicaSet

# BUILD ROLLBACK - EXAMPLE

ReplicaSets + Project Builds

1



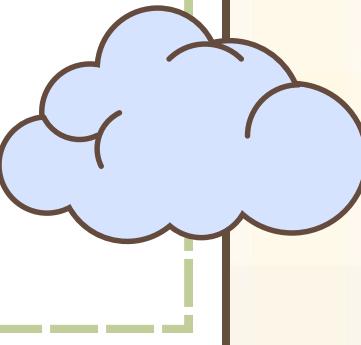
2



3

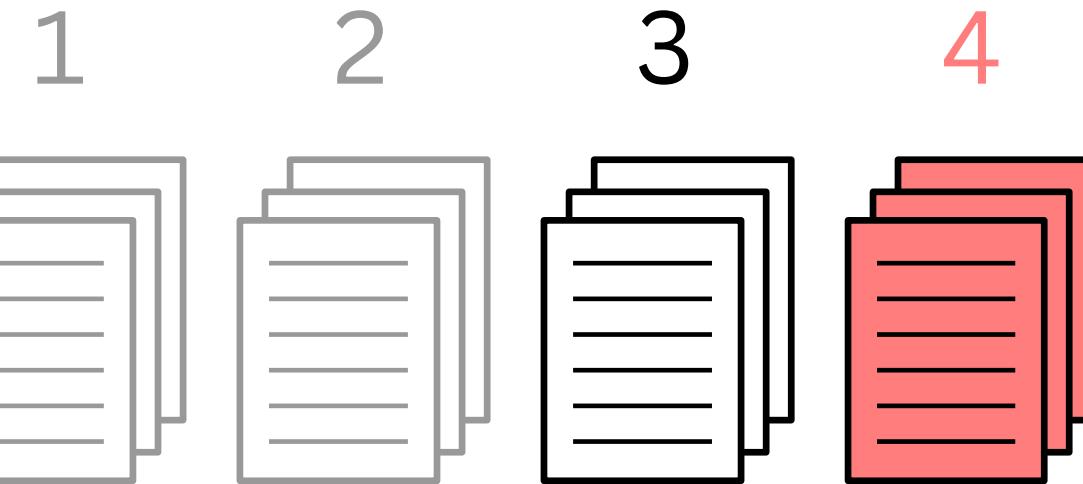


Rolling back creates a new ReplicaSet based on the previous version.

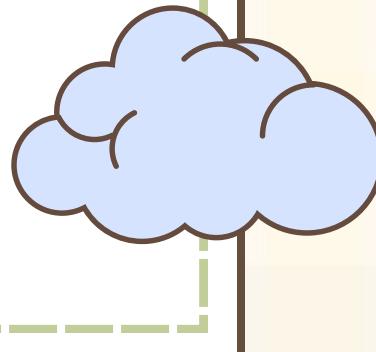


# BUILD ROLLBACK - EXAMPLE

ReplicaSets + Project Builds



Rolling back creates a new ReplicaSet based on the previous version.

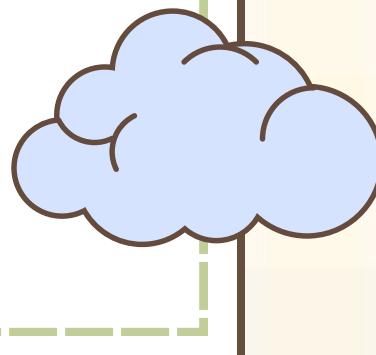


# BUILD ROLLBACK - EXAMPLE

ReplicaSets + Project Builds

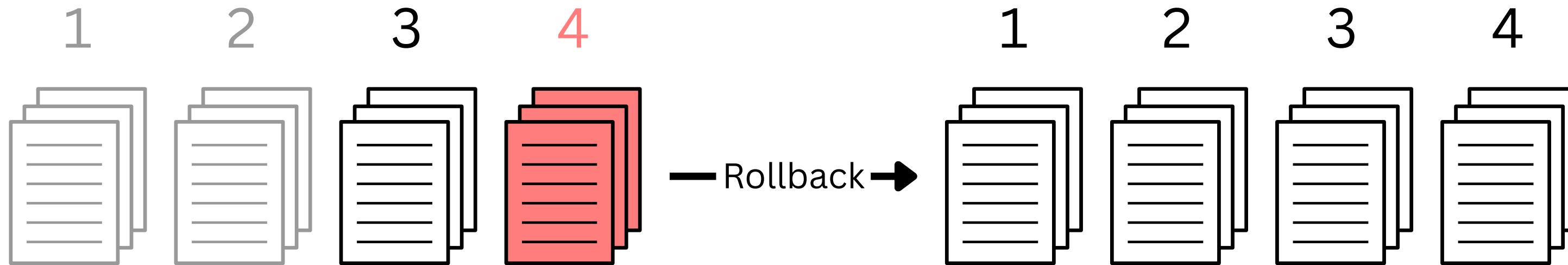


Rolling back creates a new ReplicaSet based on the previous version.



# BUILD ROLLBACK - EXAMPLE

ReplicaSets + Project Builds

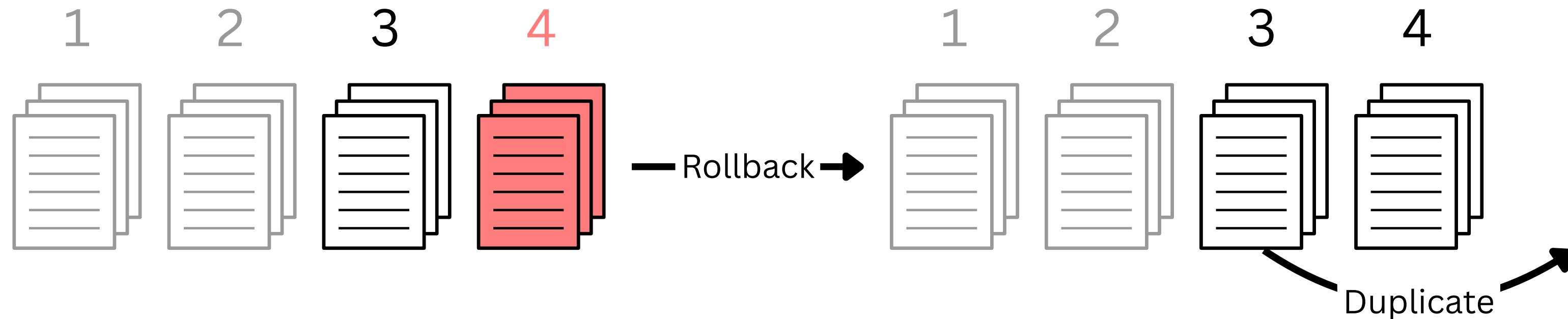


Rolling back creates a new ReplicaSet based on the previous version.



# BUILD ROLLBACK - EXAMPLE

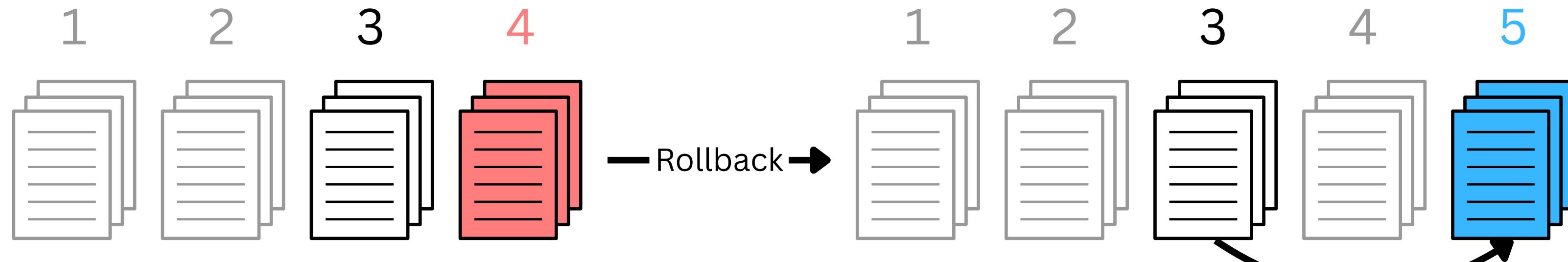
ReplicaSets + Project Builds



Rolling back creates a new ReplicaSet based on the previous version.

# BUILD ROLLBACK - EXAMPLE

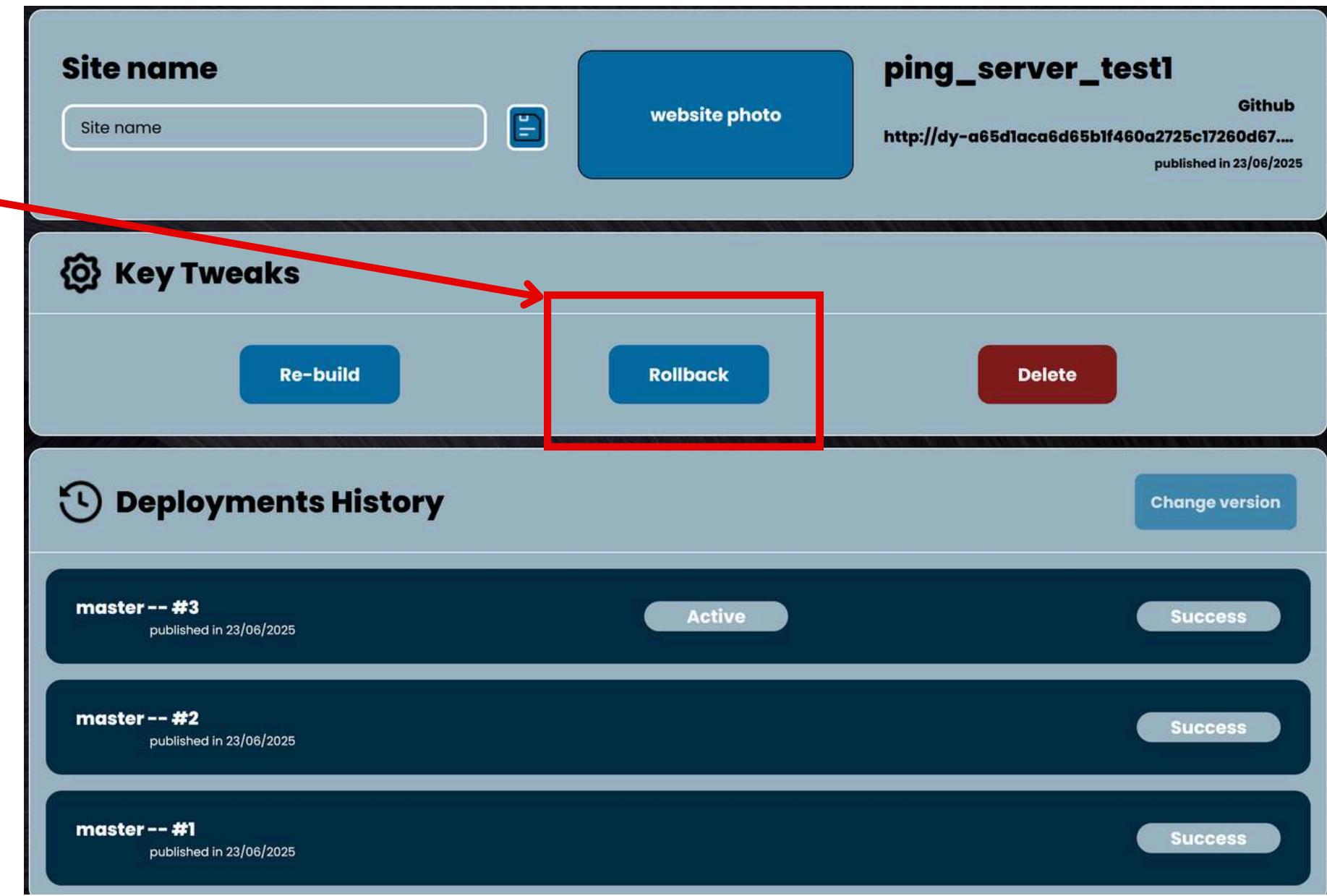
ReplicaSets + Project Builds



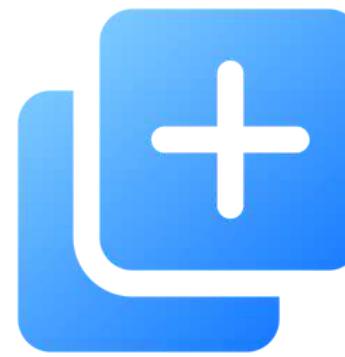
Rolling back creates a new ReplicaSet based on the previous version.

# BUILD ROLLBACK - WHERE?

Rollback Button



# ROLLBACK SHORTCOMINGS



Extra Build  
Created



Old Versions  
Deleted



One Step  
Back



Rollback  
Loop

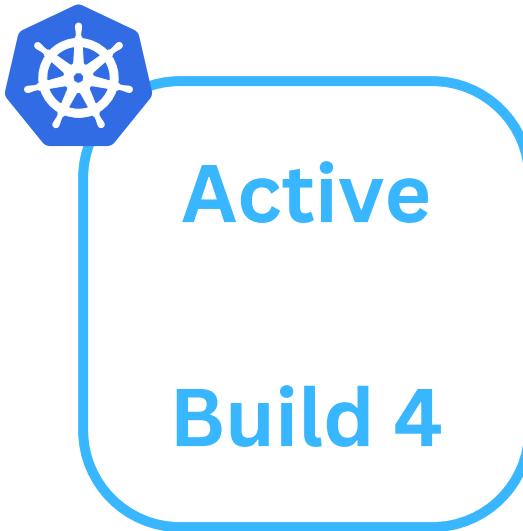
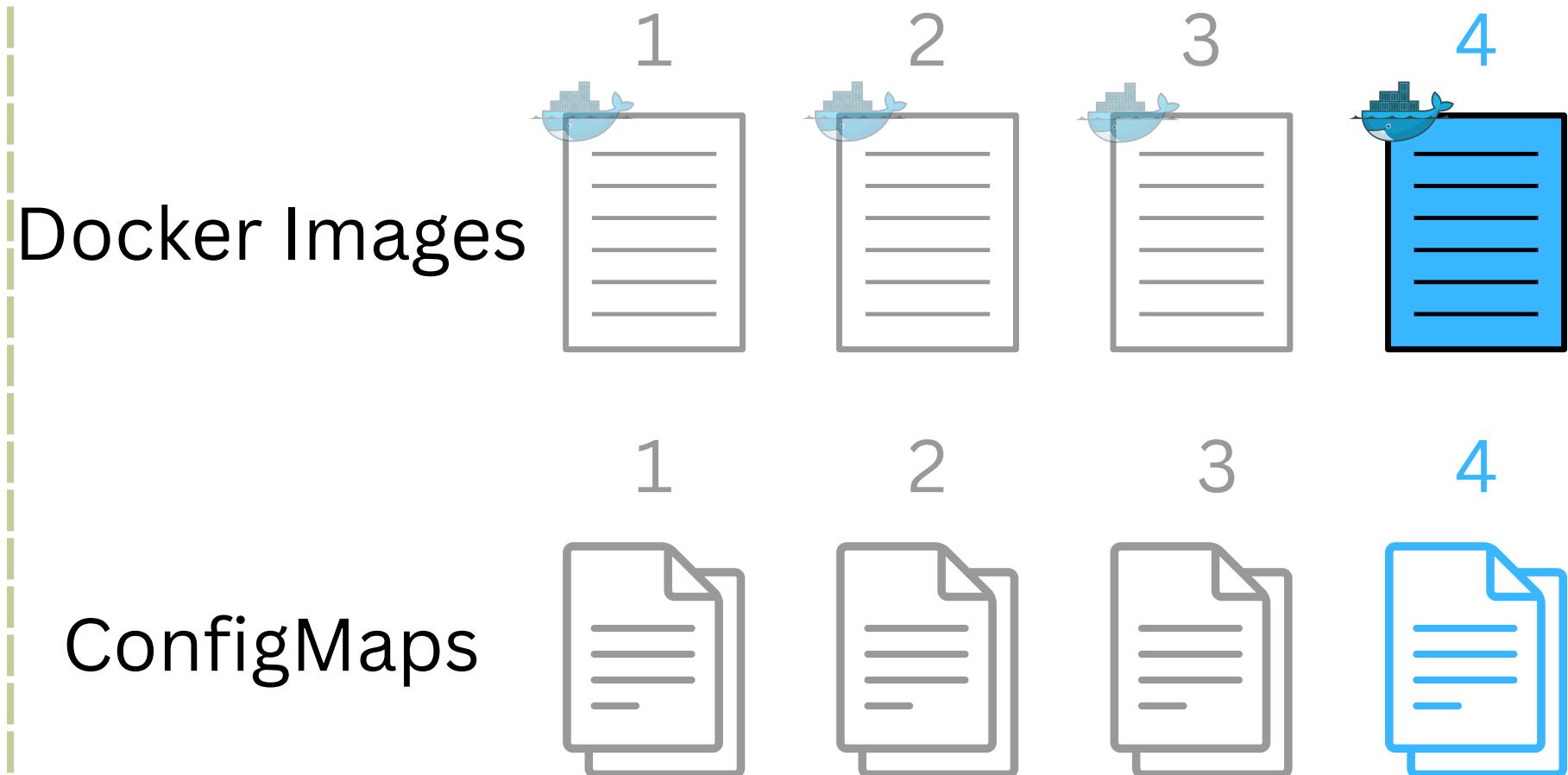


SET ACTIVE BUILD

# **SET ACTIVE BUILD - IMP.**

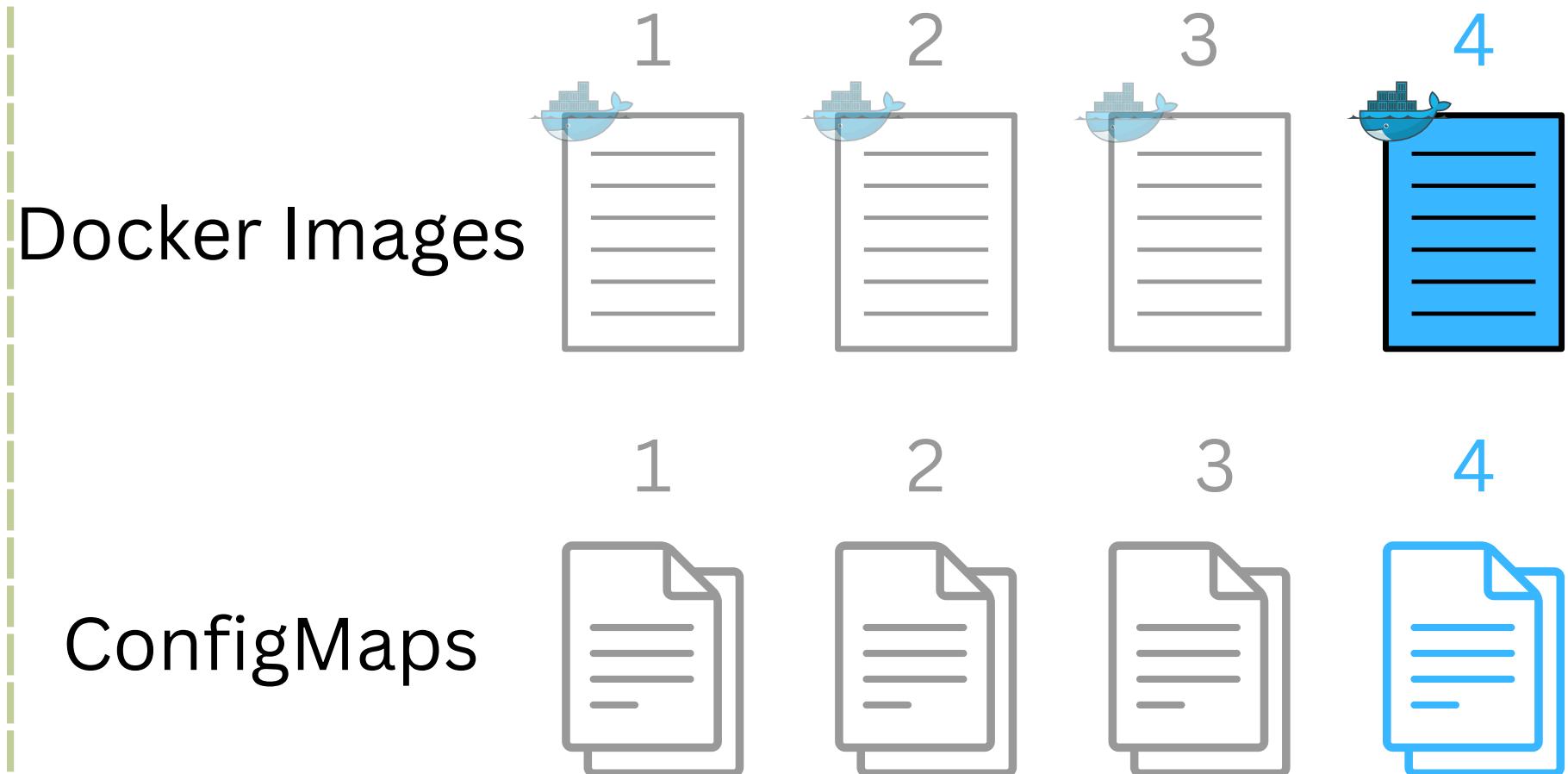
- Kubernetes **ReplicaSets** mainly **depend on two things**:
  - **Docker Images** (the app code)
  - **ConfigMaps** (the environment variables)
- **To switch builds:** update the deployment with the corresponding Docker image and ConfigMap - **no need for ReplicaSets**

# SET ACTIVE BUILD - EXAMPLE



Choose any previous build and make it live instantly.

# SET ACTIVE BUILD - EXAMPLE

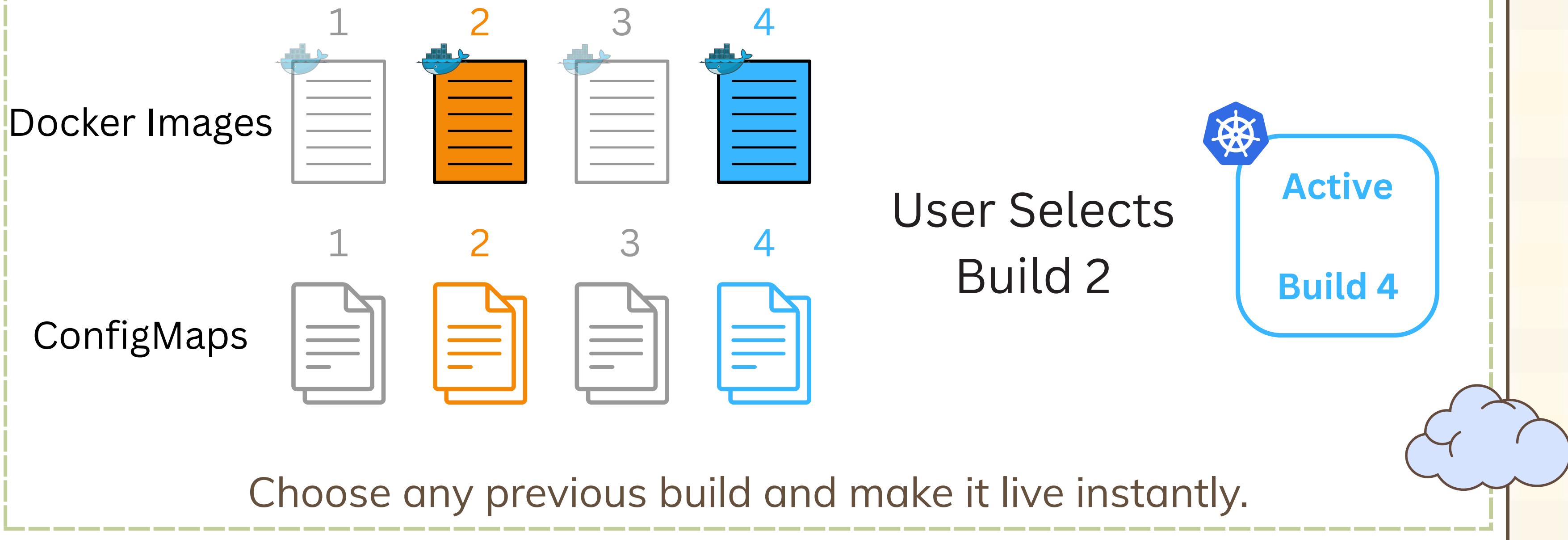


User Selects  
Build 2



Choose any previous build and make it live instantly.

# SET ACTIVE BUILD - EXAMPLE

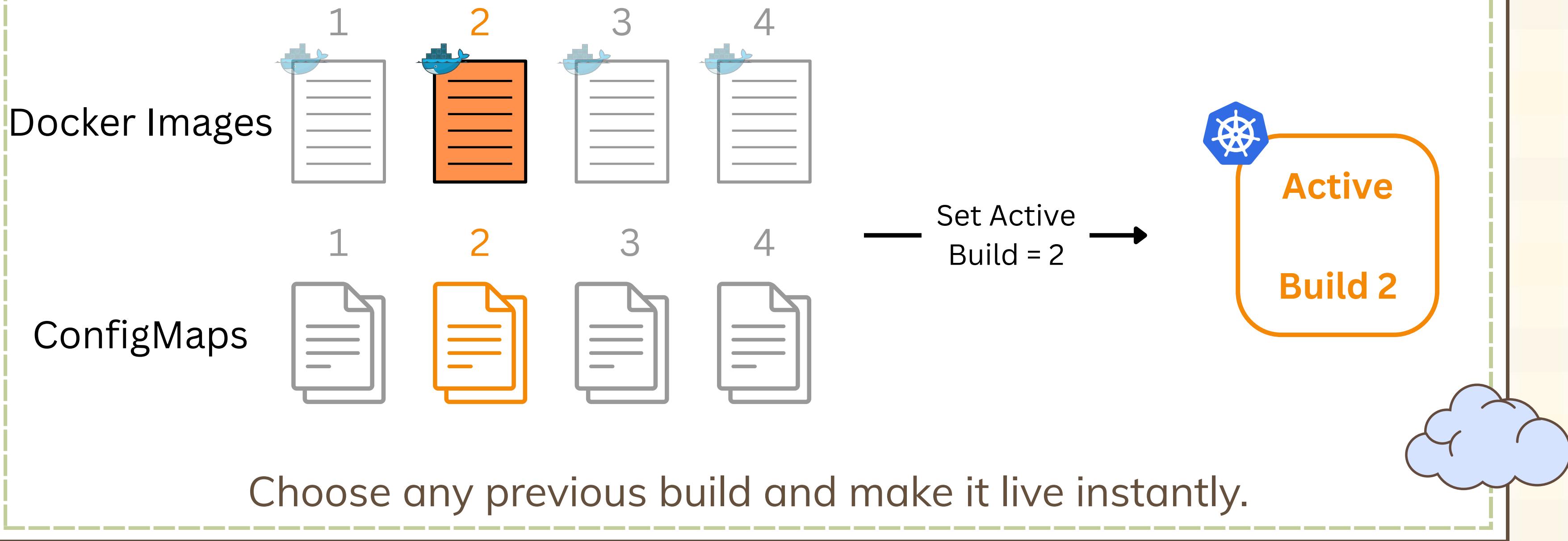


# SET ACTIVE BUILD - EXAMPLE



Choose any previous build and make it live instantly.

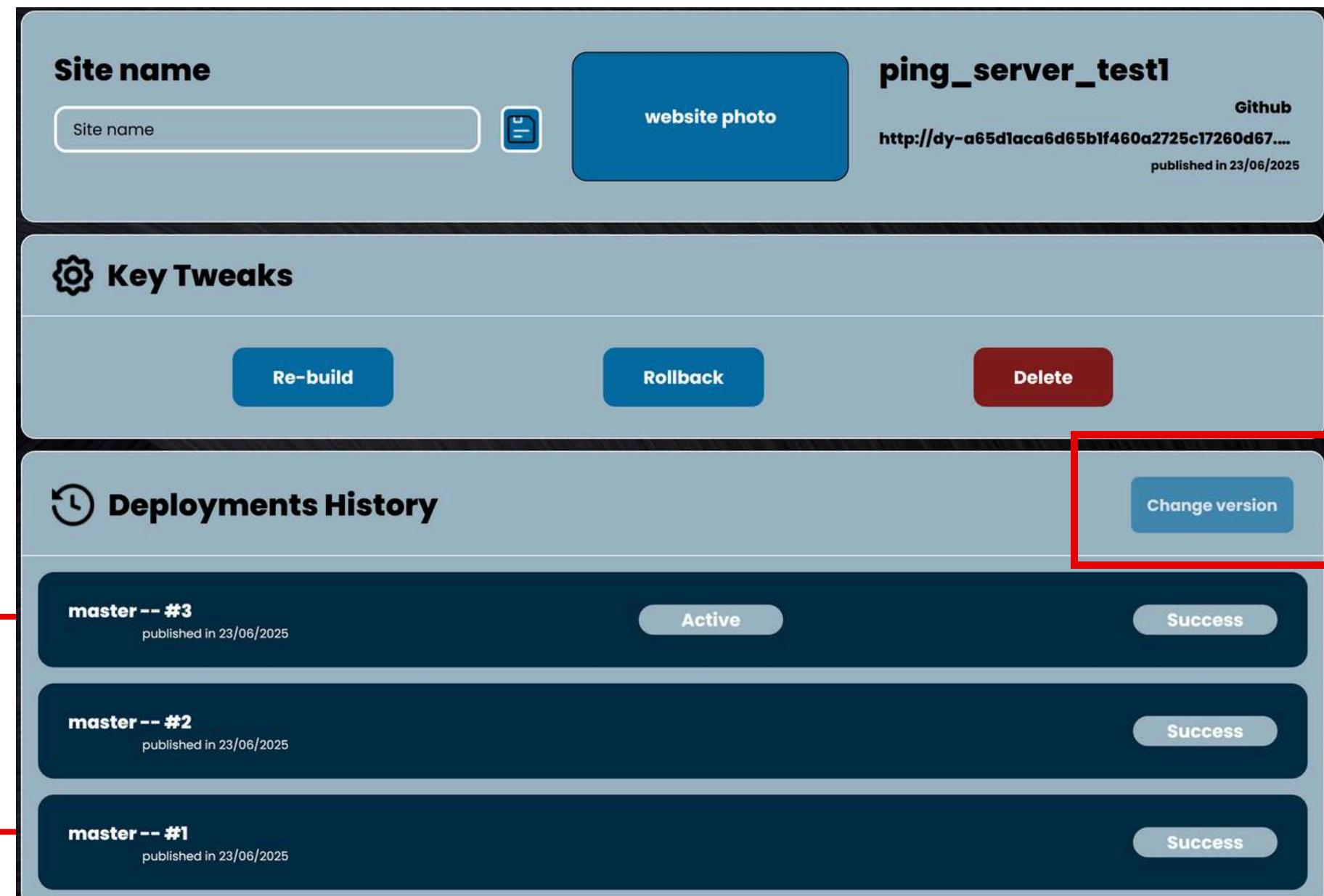
# SET ACTIVE BUILD - EXAMPLE



# SET ACTIVE BUILD - WHERE?

Select Build  
Here

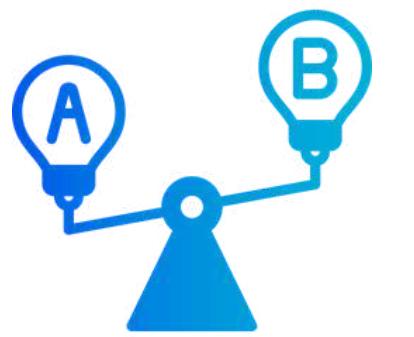
Change Version  
Button



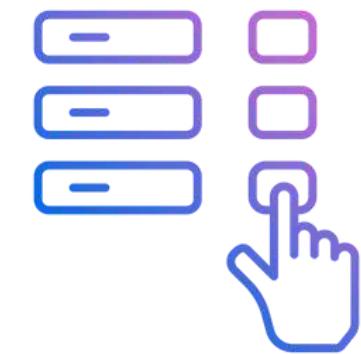
# SET ACTIVE BUILD - BENEFITS



No Redundant  
Builds



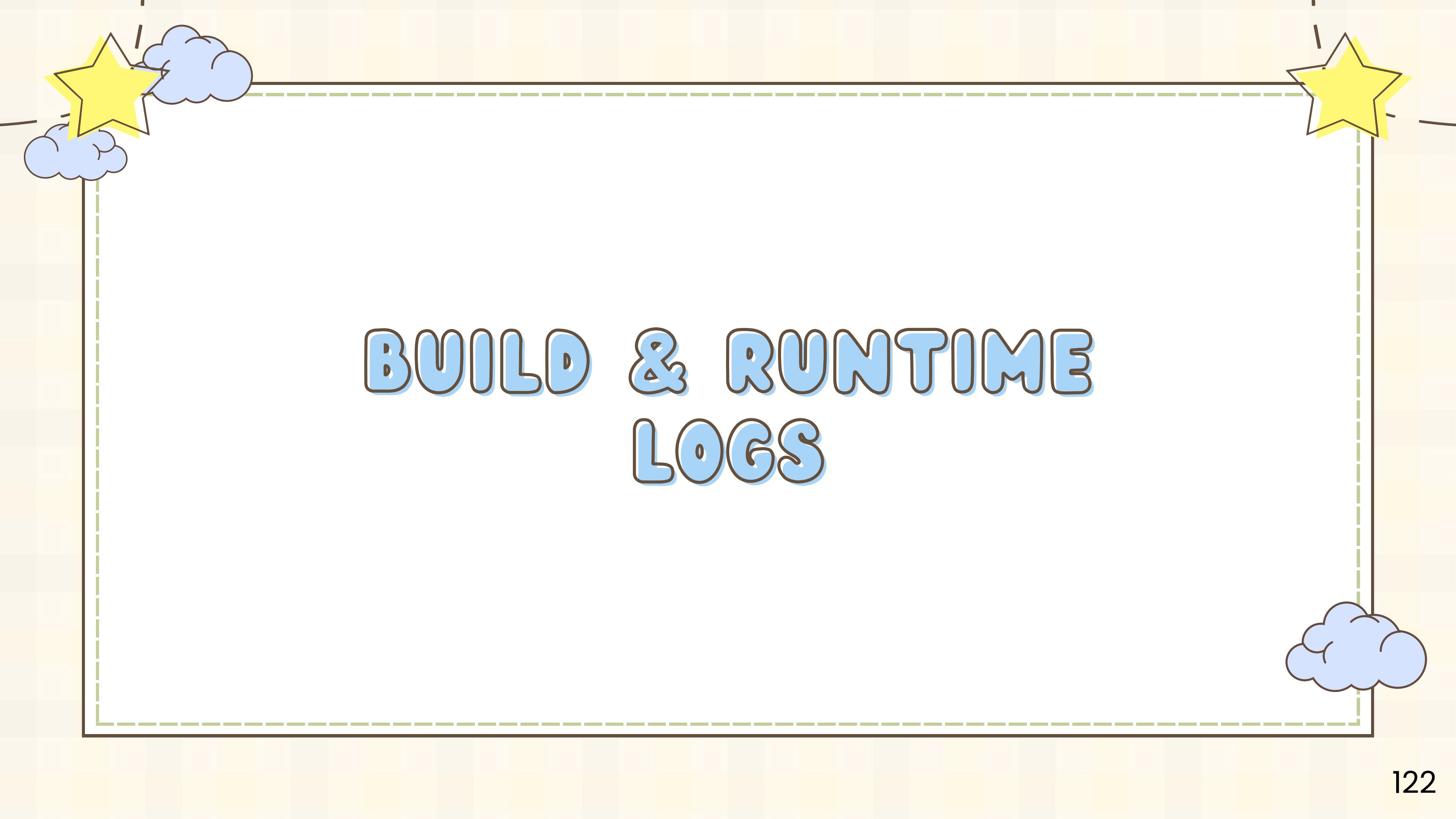
Easy Testing &  
Comparison



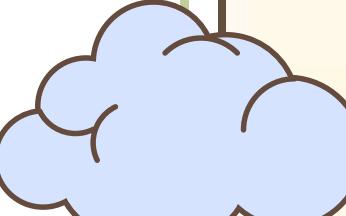
Travel through  
History

- Future Enhancements:
  - Direct access to any build, more power to the user.





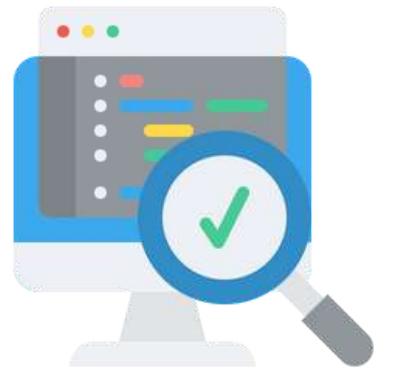
# BUILD & RUNTIME LOGS



# WHY

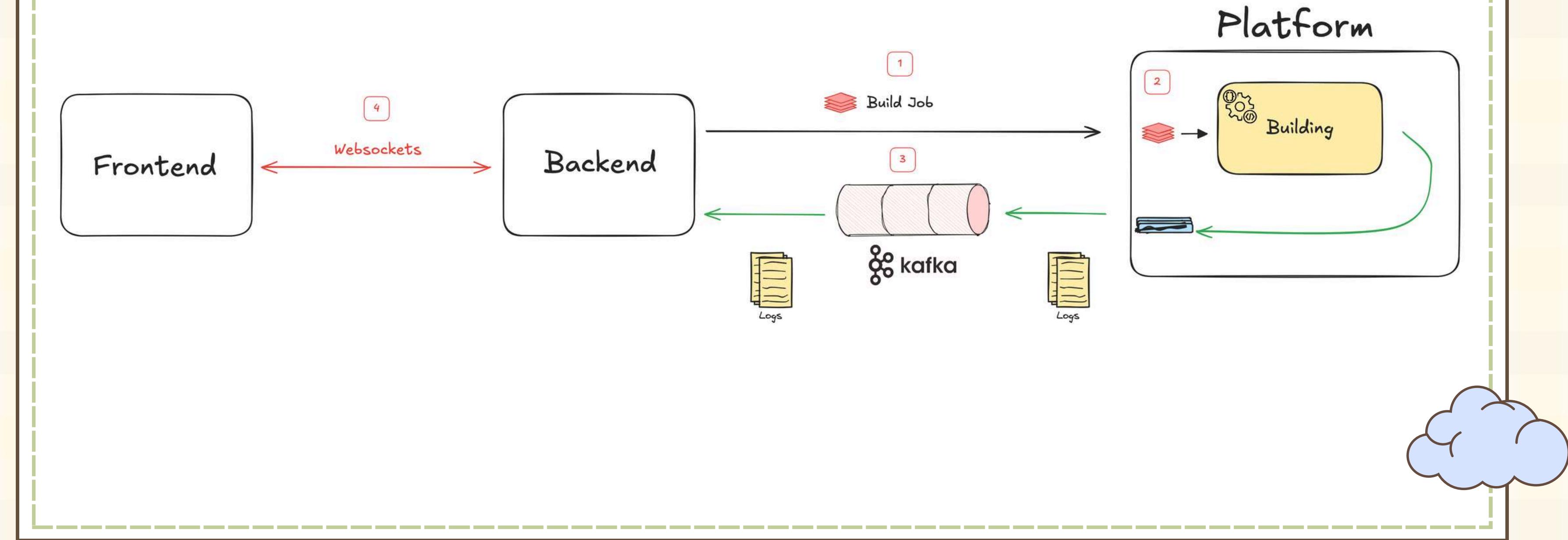


Enhance User Experience

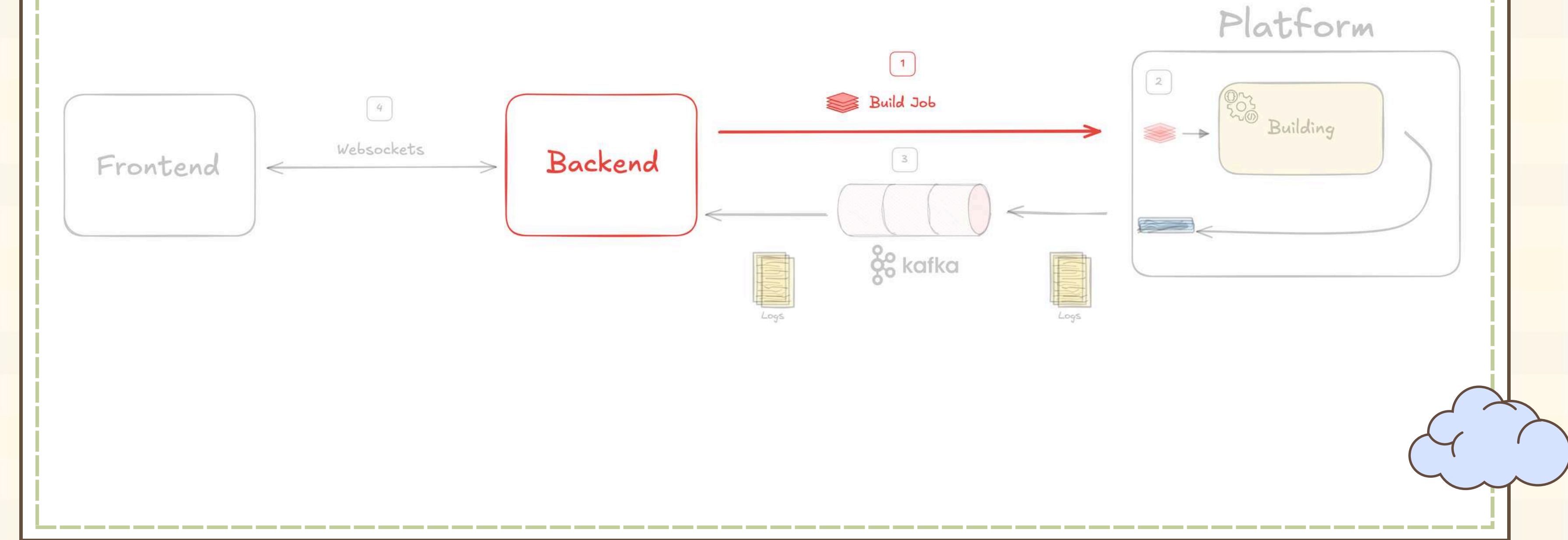


Live Debugging

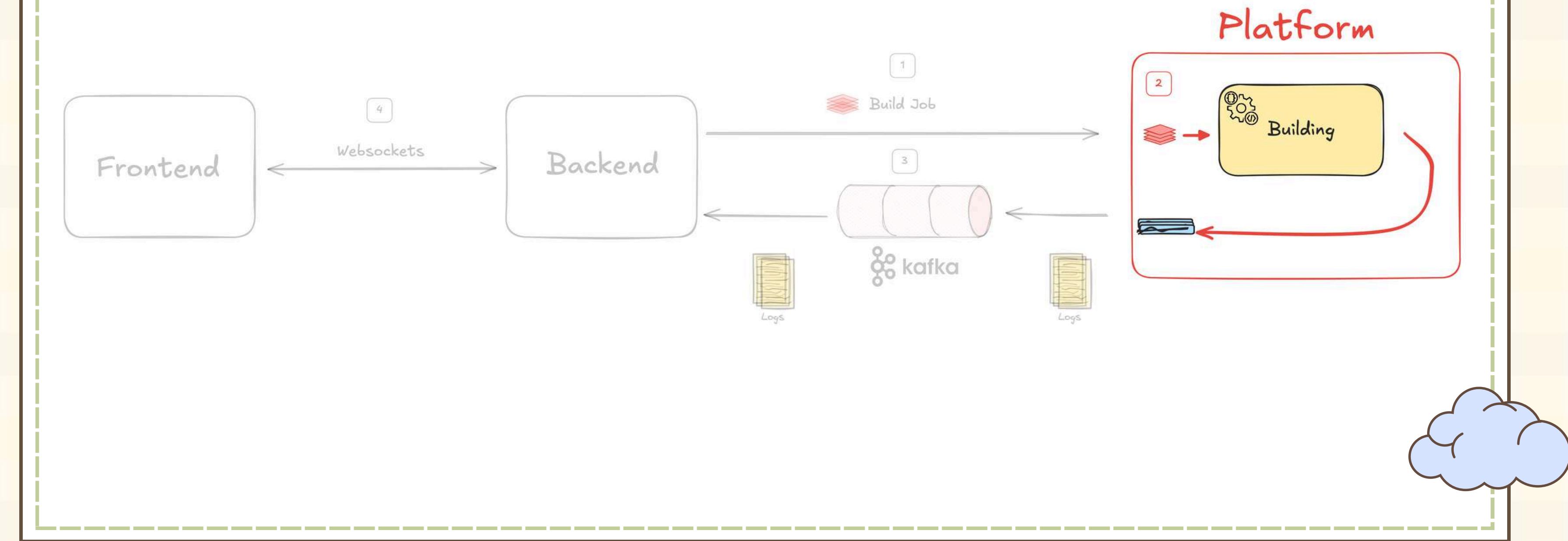
# BUILD LOGS



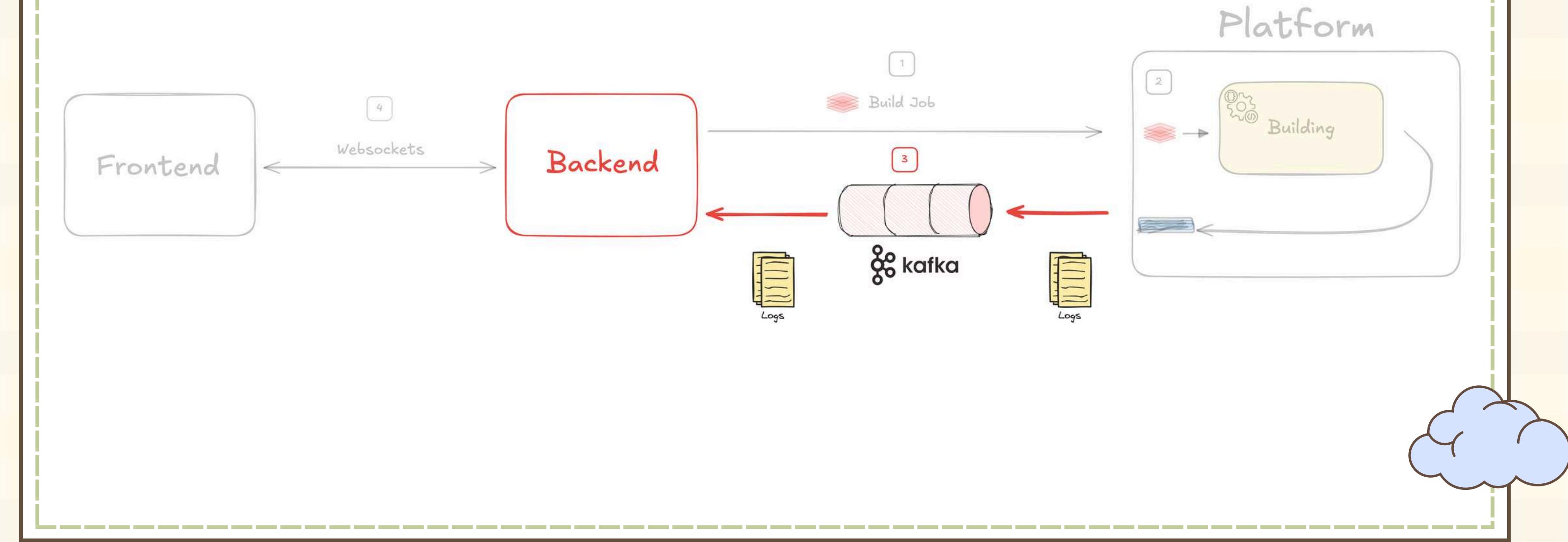
# BUILD LOGS



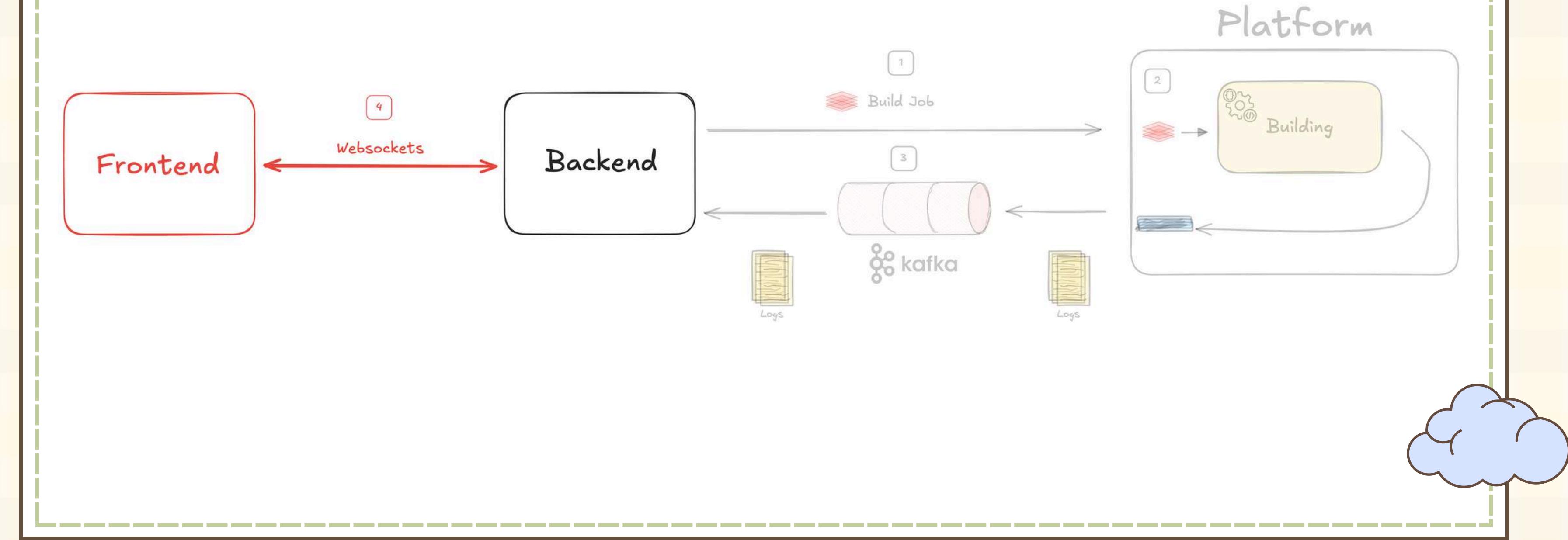
# BUILD LOGS



# BUILD LOGS



# BUILD LOGS



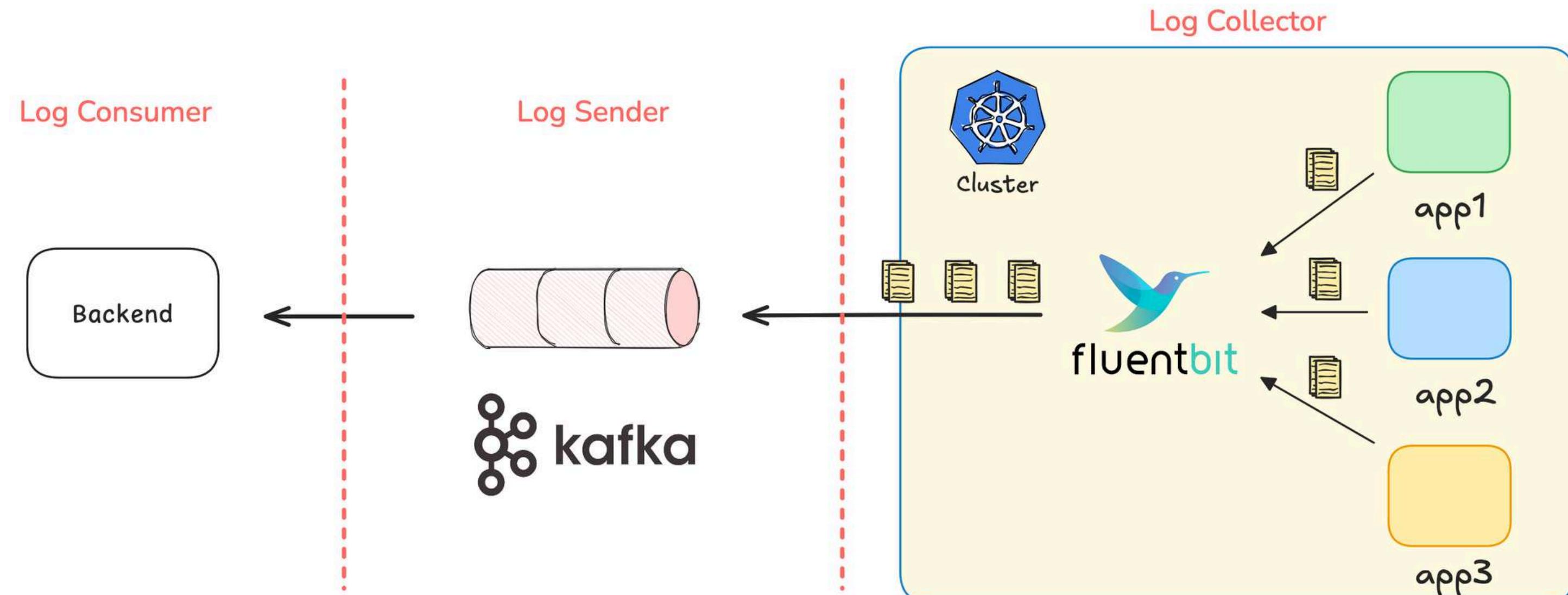
# BUILD LOGS

 **Finished**

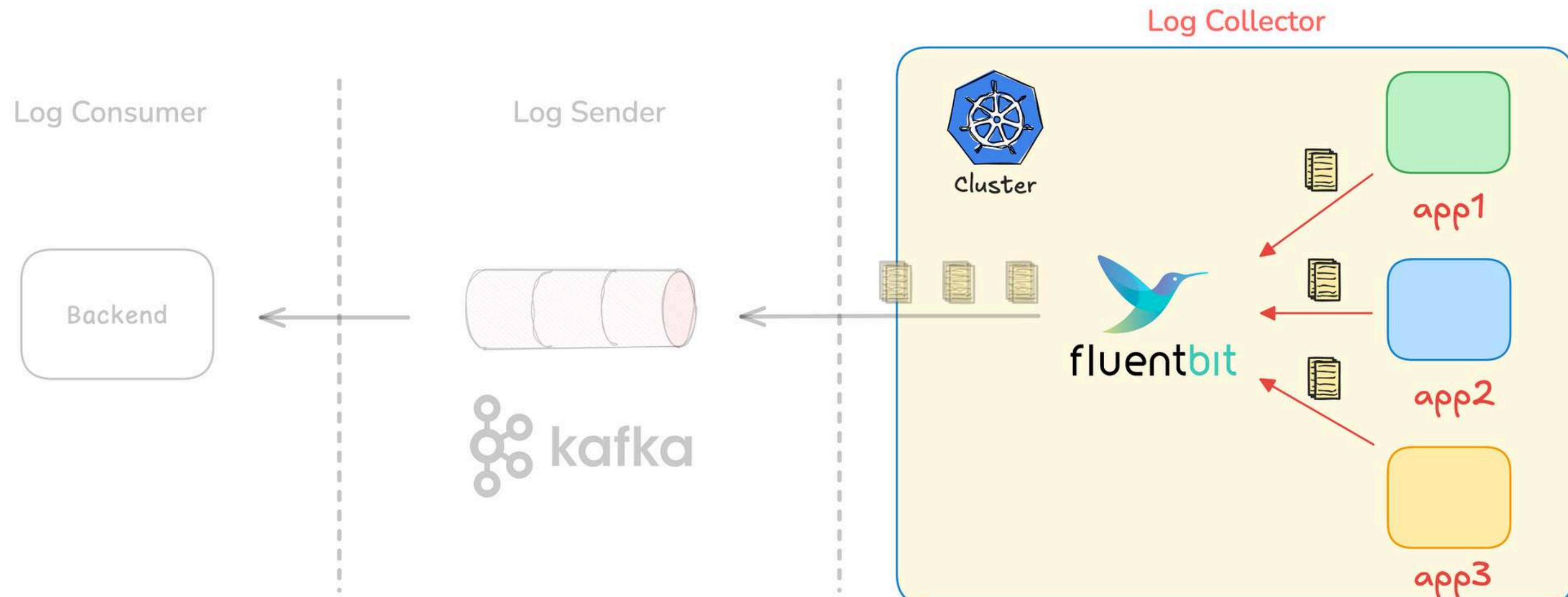
↓ Initializing build environment 1s 

```
1 | > Step 1/6 : FROM node:22-alpine
2 | > Step 2/6 : WORKDIR /app
3 | > Step 3/6 : COPY . .
4 | > Step 4/6 : RUN npm install --force --ignore-scripts
5 | > Step 5/6 : RUN npm run build
6 | > Step 6/6 : CMD [ "npm", "run", "start" ]
7 | > Successfully built e6b457212056
8 | > Successfully tagged fadyimageregistry.azurecr.io/dy-6a743961245c62834baf4d74501eee40:v1
9 | > build-job-finished-log
```

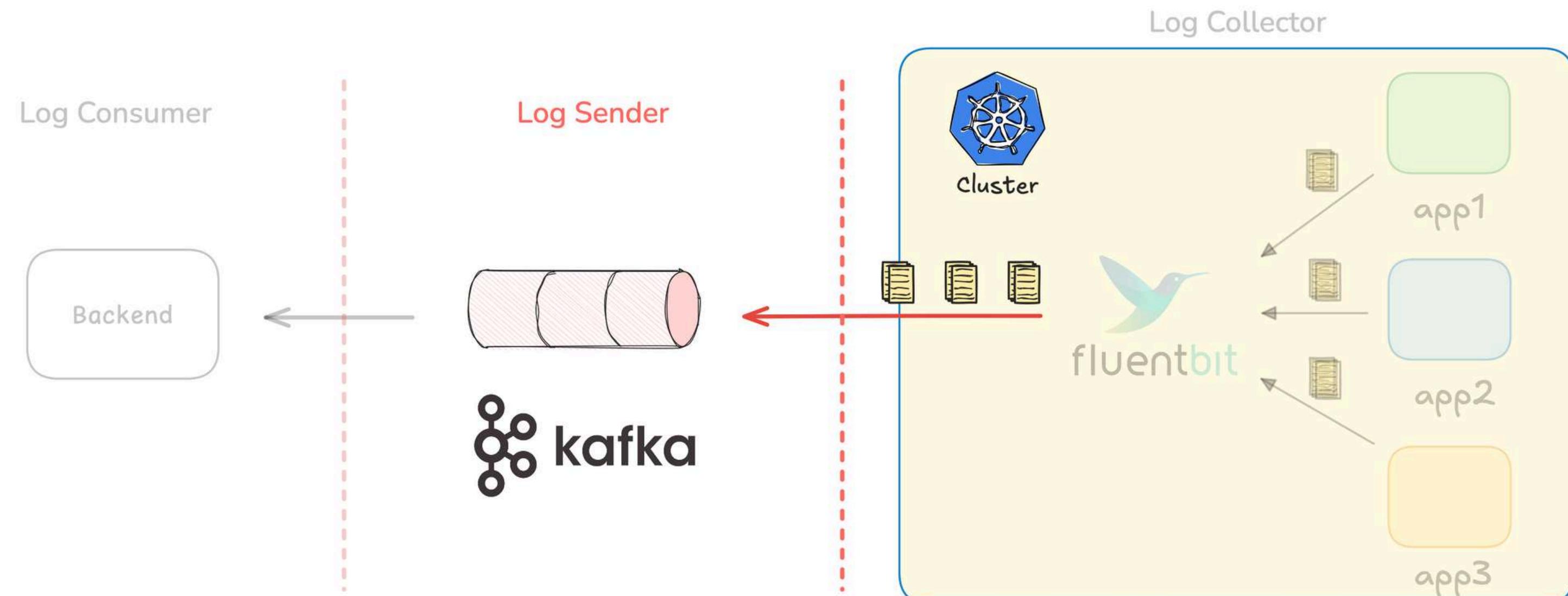
# RUNTIME LOGS



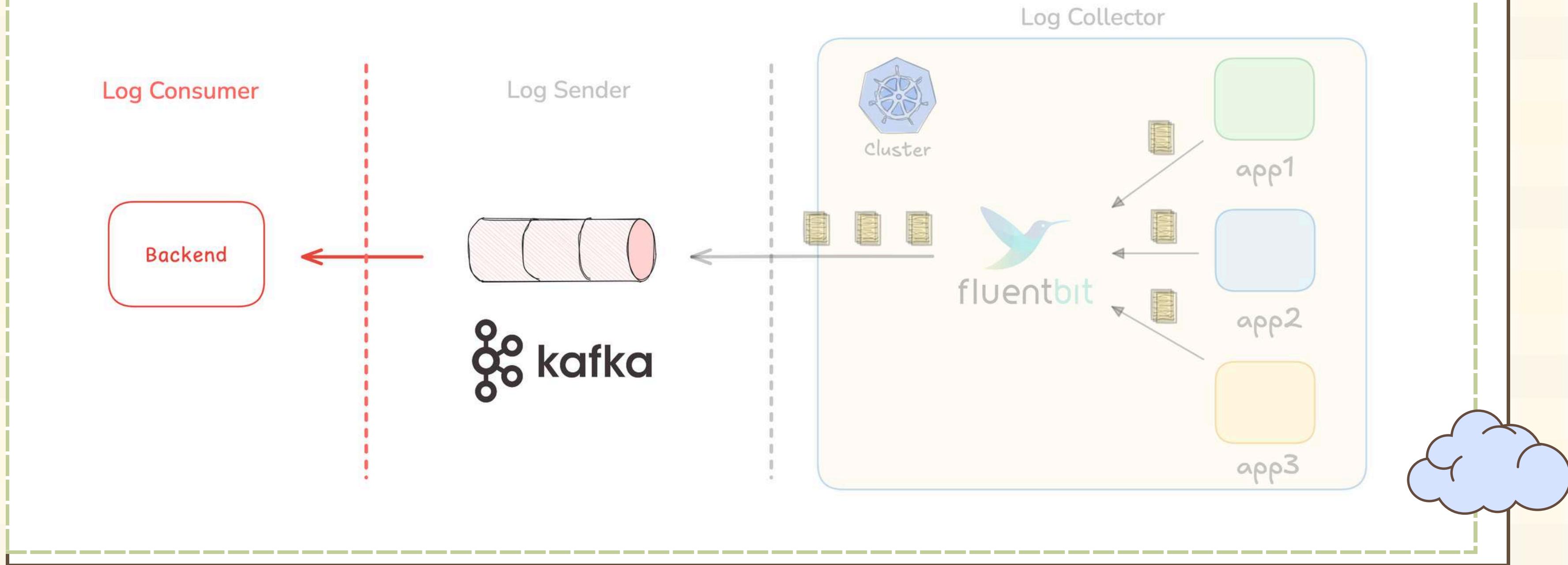
# RUNTIME LOGS



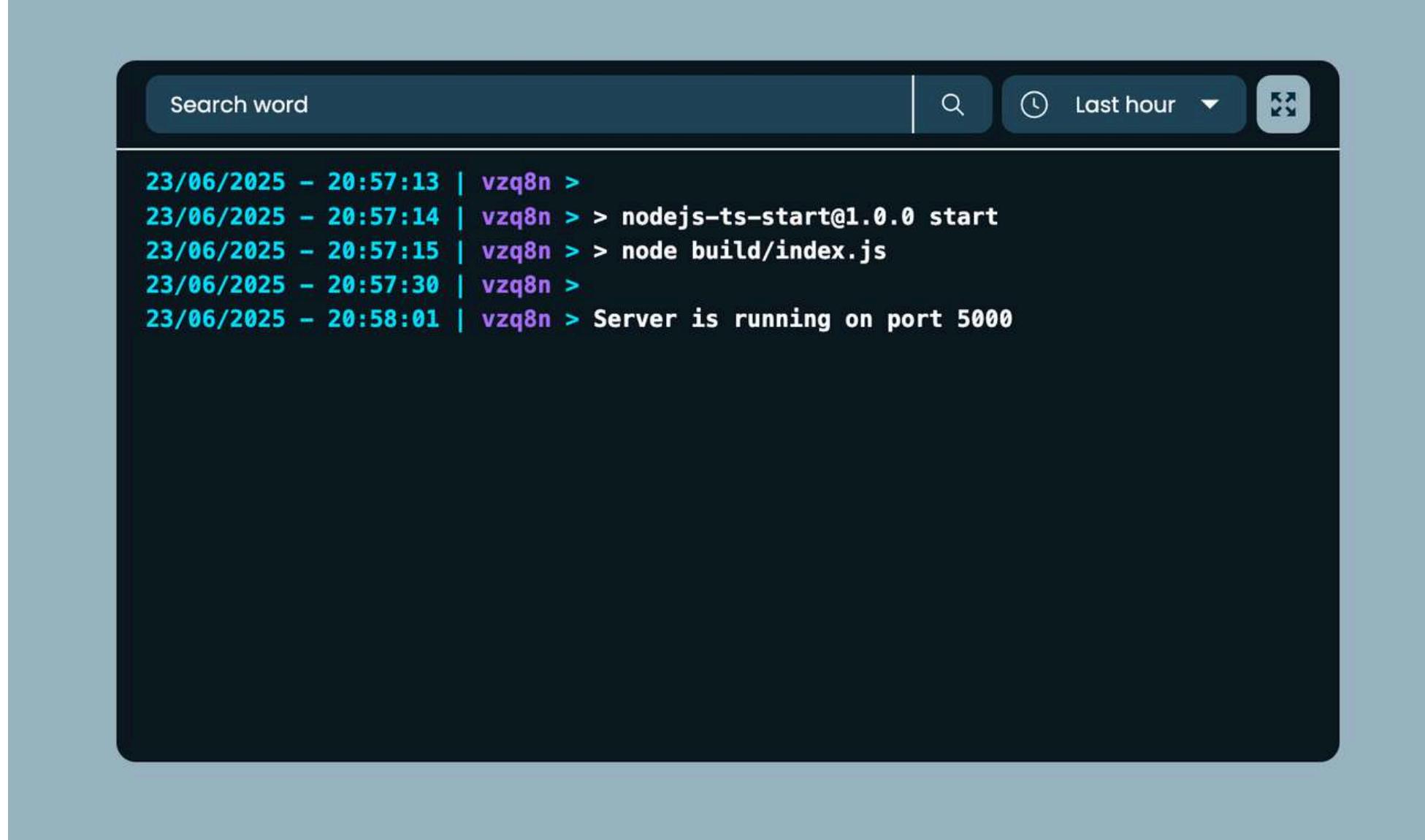
# RUNTIME LOGS



# RUNTIME LOGS

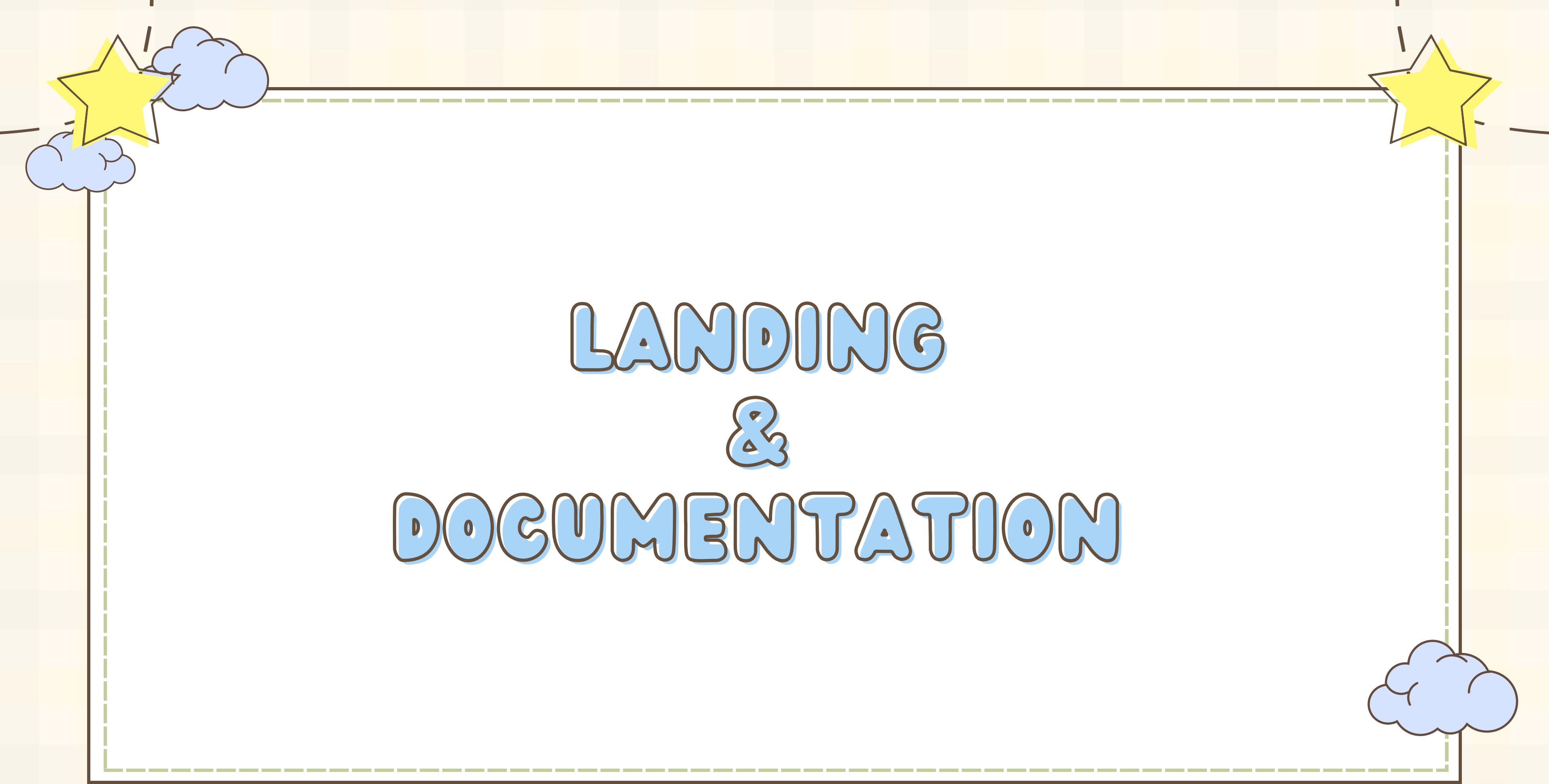


# RUNTIME LOGS



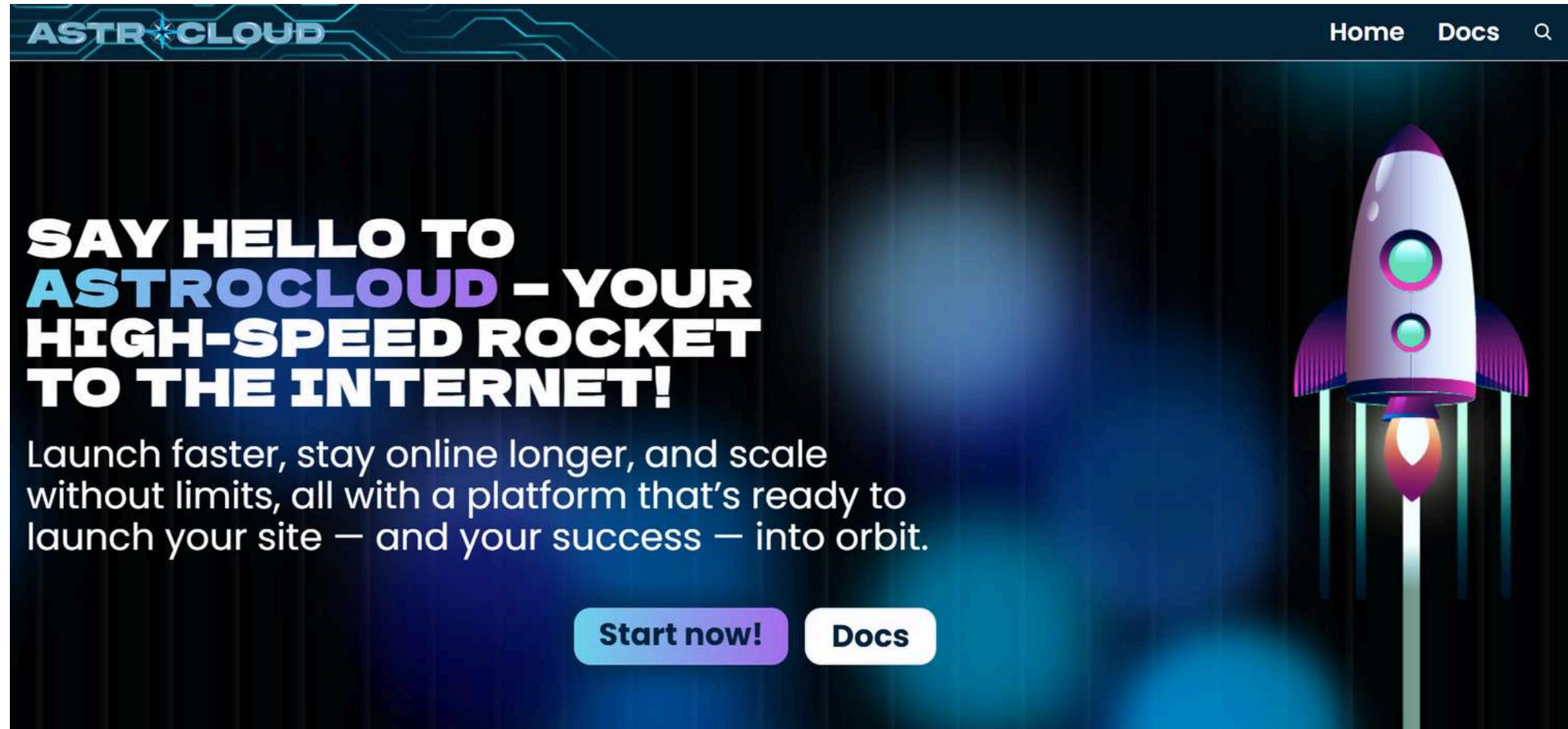
A screenshot of a runtime logs interface. At the top, there is a search bar labeled "Search word" with a magnifying glass icon, a time filter set to "Last hour" with a clock icon, and a refresh button. Below the search bar, the log entries are displayed in a dark blue container:

```
23/06/2025 - 20:57:13 | vzq8n >
23/06/2025 - 20:57:14 | vzq8n > > nodejs-ts-start@1.0.0 start
23/06/2025 - 20:57:15 | vzq8n > > node build/index.js
23/06/2025 - 20:57:30 | vzq8n >
23/06/2025 - 20:58:01 | vzq8n > Server is running on port 5000
```



# LANDING & DOCUMENTATION

# LANDING PAGE



The image shows a landing page for "ASTRO CLOUD". At the top left is the "ASTRO CLOUD" logo with a star icon. At the top right are links for "Home", "Docs", and a search icon. The main headline reads: "SAY HELLO TO ASTROCLOUD - YOUR HIGH-SPEED ROCKET TO THE INTERNET!". Below the headline is a subtext: "Launch faster, stay online longer, and scale without limits, all with a platform that's ready to launch your site — and your success — into orbit." At the bottom are two buttons: "Start now!" and "Docs". To the right of the text is a stylized illustration of a rocket launching from a pad, with a trail of light behind it.

# LANDING PAGE

ASTR CLOUD

Home Docs

## Powerful Features

Everything you need to build, deploy, and scale your applications with confidence

**Github Integration**  
GitHub integration for automatic deployments, version control and collaboration

**Zip File Deploy**  
Upload and deploy your projects instantly with simple drag-and-drop zip file support for rapid prototyping

**Build Frameworks**  
Support for multiple frameworks, allowing you to deploy apps built with your favorite technologies

# DOCUMENTATION

The screenshot shows a documentation page for "Manual Upload" on the Astro Cloud website. The page has a dark blue header with the "ASTRO CLOUD" logo and navigation links for "Home", "Docs", and a search icon. On the left, there's a sidebar with "Examples" (Dynamic Projects, Static Projects) and "Tutorials" (Dynamic hosting, Static hosting). The main content area features a large heading "Manual Upload" and a sub-section "Steps" with the first step: "1. Click Add Project." Below this is a screenshot of the Astro Cloud dashboard showing two projects: one static and one dynamic, each with its own domain and deployment status. A cursor points to the "Add project" button at the top right of the dashboard interface.

# DOCUMENTATION

The screenshot shows a dark-themed documentation page for AstroCloud. At the top, there's a navigation bar with the AstroCloud logo, a search icon, and links for "Home" and "Docs". The main content area has a title "Static Projects Examples" and a sub-section titled "Vanilla HTML & CSS Project". It includes details about a project titled "Clean Landing Page", its description as a sleek responsive landing page built with pure HTML and CSS, and build details. A prominent button at the bottom says "Launch Your Website". On the left sidebar, there are sections for "Examples" (Dynamic Projects, Static Projects), and "Tutorials" (Dynamic hosting, Static hosting). On the right sidebar, there's a "ON THIS PAGE" section with links to "Static Projects Examples", "Vanilla HTML & CSS Project", and "React Project".

**ASTRO CLOUD**

Home Docs

**Examples**

- Dynamic Projects
- Static Projects**

**Tutorials**

- Dynamic hosting
- Static hosting

## Static Projects Examples

This guide provide hands-on examples of static websites hosted on AstroCloud. Learn how to deploy frontend projects like HTML, CSS, and React.js sites—using real-world codebases.

### Vanilla HTML & CSS Project

**Project Title:** Clean Landing Page

**Description:** A sleek and responsive landing page built using pure HTML and CSS. Great for beginners and ideal for showcasing a product or service.

**Build Details:**

Launch Your Website

Configure AstroCloud builds and deploys for: FadyAdel10/Template2

**ON THIS PAGE**

- Static Projects Examples
- Vanilla HTML & CSS Project
- React Project

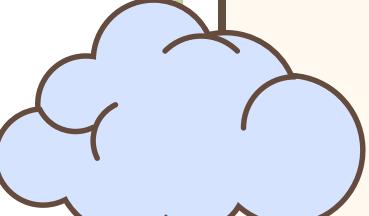
# SEO-OPTIMIZED WEBSITE



Astro for static site  
generation



Enhanced MetaData





# ADMIN DASHBOARD



Imagine we just boosted our resources—what if, in ONE CLICK , you could move **1,000 users** into a higher-limit group?



# DASHBOARD



**Admin Dashboard**  
Manage your platform users, projects, and monitor system activity.

Daily   Weekly   Monthly

New Users Today: 0   +0% from yesterday

Total Users: 1   +0% from yesterday

Total Deployments: 0   +0% from yesterday

**System Usage**  
Resource utilization across your platform

A line chart showing resource utilization over six months (Jan to Jun). The Y-axis represents percentage (0 to 100). The X-axis shows the months. Two lines are plotted: CPU (purple) and Memory (green). Both lines show a slight upward trend from January to March, followed by a slight decline towards June. The CPU usage is consistently higher than Memory usage.

Month	CPU (%)	Memory (%)
Jan	~60	~45
Feb	~60	~50
Mar	~75	~55
Apr	~78	~60
May	~70	~65
Jun	~65	~70

**Users**  
Recent users in your platform

User	Status	Projects
neesaaa Joined: Jun 20, 2025	Active Last active: Jun 22, 2025	2 projects

The "Users" section is highlighted with an orange border. Three orange arrows point upwards from the bottom of the page towards this section, drawing attention to it.

# DASHBOARD



The screenshot shows the Admin Dashboard interface. At the top, there are three buttons: Daily, Weekly, and Monthly. Below them are three summary cards:

- New Users Today: 0 (+0% from yesterday)
- Total Users: 1 (+0% from yesterday)
- Total Deployments: 0 (+0% from yesterday)

Below these cards is a section titled "System Usage" with the subtitle "Resource utilization across your platform". It contains a line graph comparing CPU (purple) and Memory (green) usage over time from January to June. The Y-axis ranges from 0 to 100. The CPU usage starts at approximately 60 in Jan, dips slightly in Feb, rises to a peak of about 80 in Mar, stays relatively flat until May, and then decreases again in Jun. Memory usage starts at approximately 50 in Jan, remains stable through April, then rises to about 70 in May and stays around 70 in Jun.

At the bottom of the dashboard, there is a "Users" section with the subtitle "Recent users in your platform". It lists one user: neesaaa, who joined on Jun 20, 2025, and is currently Active (last active on Jun 22, 2025). There are 2 projects associated with this user.

**User Plans Management**  
Organize users into plans and manage their project limits

Search plans...

+ Create

**Free**  
The base free plan for all users.

Project Limits

STATIC PROJECTS: 5

DYNAMIC PROJECTS: Total: 10

Tier 1: 3	Tier 2: 2
Tier 3: 1	Tier 4: 0

Members: SamehOssama, JohnDoe

Edit Users

**Pro**  
Professional plan with more resources.

Project Limits

STATIC PROJECTS: 20

DYNAMIC PROJECTS: Total: 50

Tier 1: 15	Tier 2: 10
Tier 3: 5	Tier 4: 2

Members: AliceSmith

Edit Users

**Enterprise**  
Enterprise plan with unlimited resources.

Project Limits

STATIC PROJECTS: 100

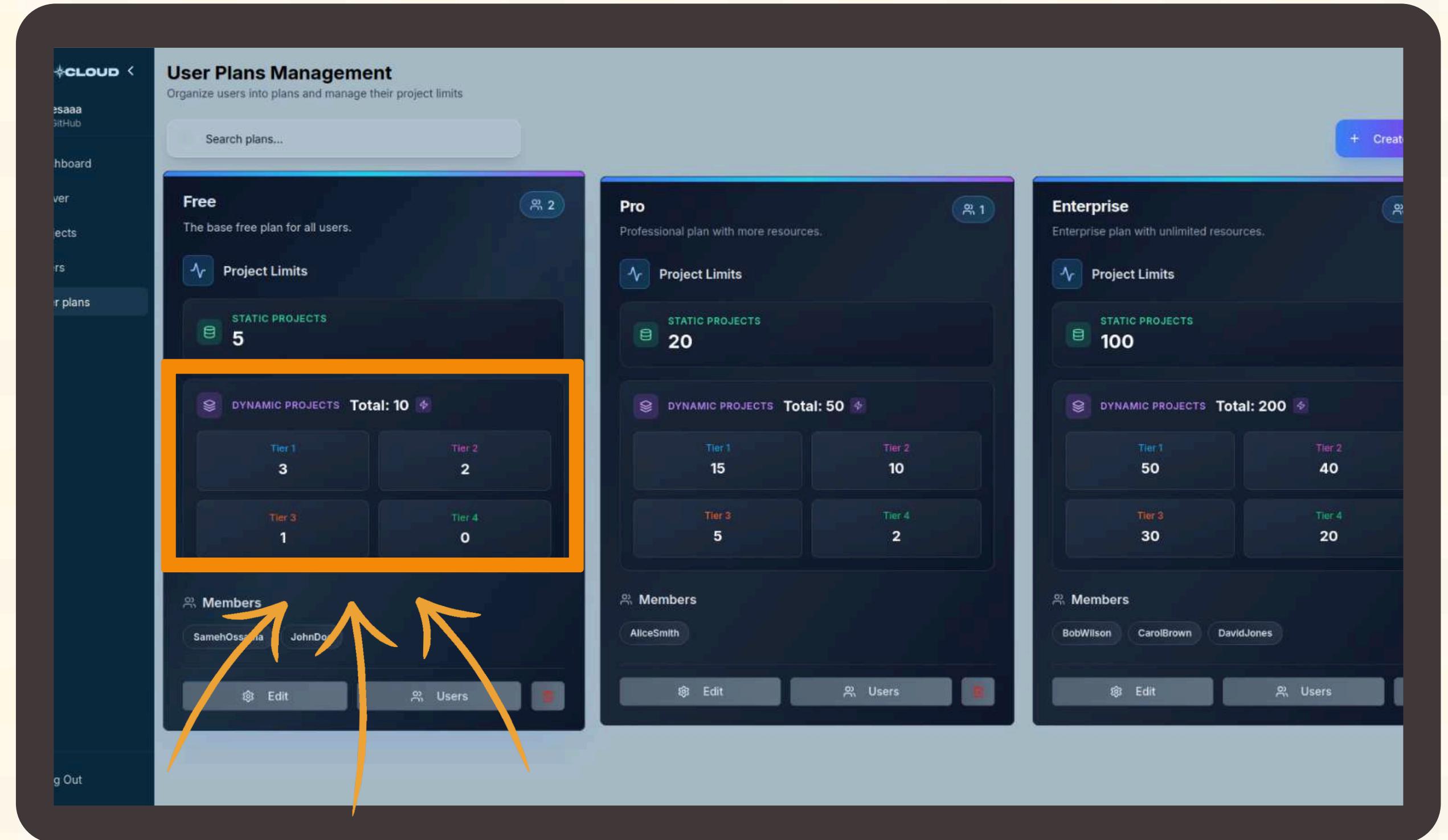
DYNAMIC PROJECTS: Total: 200

Tier 1: 50	Tier 2: 40
Tier 3: 30	Tier 4: 20

Members: BobWilson, CarolBrown, DavidJones

Edit Users

Plan	Static Projects	Total Dynamic Projects	Tier 1	Tier 2	Tier 3	Tier 4
Free	5	10	3	2	1	0
Pro	20	50	15	10	5	2
Enterprise	100	200	50	40	30	20





DEMO



## NEXT STEPS

# PROJECT RELEVANCE



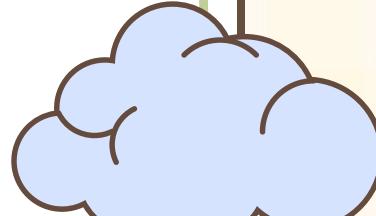
First in Egypt



Tech Landscape



Local Currency



Astro for static site

generation

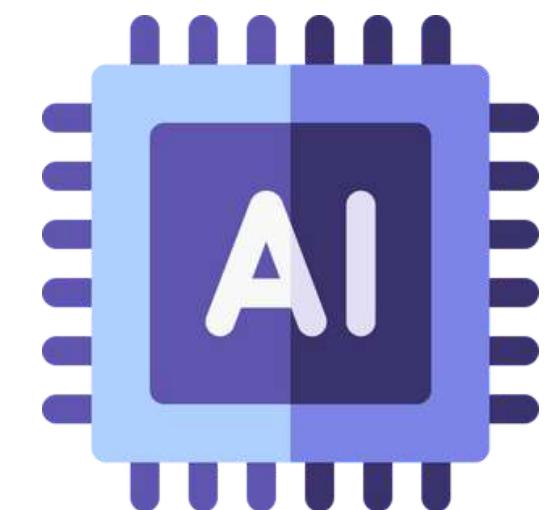
# NEXT STEPS



Startup



Funding



AI

Astro for static site

generation

**THANK YOU**