

“ In the shadowed corners of a tech office, a team of engineers gathered around glowing monitors, embarking on their nocturnal ritual: deploying a new version. The air vibrated with tension, not just from the task at hand, but from the inevitable arrival of an unwelcome guest—the Bug.

This Bug was a master of chaos, turning seamless code into a battlefield, pushing the team into unwanted overtime, invading their evenings and weekends. As the clock struck midnight, the silent anticipation was shattered by the familiar sound of an alarm—the Bug had struck again.

With a mix of determination and dread, the engineers dove into the fray, their camaraderie deepening in the face of adversity. Yet, beneath their resolve, a question lingered: "Why must we always pay the price of our time?"

The battle raged through the night, a cycle of code, coffee, and shared stories of digital duels. By dawn, victory was theirs, but it was a hollow triumph. The team, weary but united, made a pact: No more. They resolved to streamline processes, advocate for automation, and most importantly, reclaim their right to a life outside code.

As they stepped out into the dawn, their spirits were buoyed not just by the defeat of the Bug, but by a newfound determination to change the game. They were more than engineers; they were guardians of their own time, ready to face the future together.

”

FeatBit

A Fast, Scalable, and Open Source Feature Flag Management Service.
Designed to Delivery Risk Mitigation and Foster Business Growth.

The guardian of your own time

Software Delivery Nightmares

Delivering and innovating not only faster, but also safer is a challenge

Low Deployment Frequency, Long Lead Time to Change, High Mean Time to Repair, High Change Failure Rate

Revenue loss

Delivering new features faster can lead to more delivery failures, which can result in lost revenue if they impact critical business functions or poor user experience. The longer it takes to fix the issue, the more revenue is lost. **Nightmare for company.**

Delayed Feature Releases

If we have to roll back all features due to a bug in just one, this will delay their availability to customers. Such a delay can hinder the team's ability to meet strategic goals and timelines, posing significant challenges for the business.

Rollback Difficulties

You release new features in a deployment, and one of them has a bug. To rollback only bug feature requires a lot of work: code reversion, retesting, re-deployment, and so on, all of which takes time (even evenings and weekends) and foster negativity within the team. **A nightmare for the dev team.**

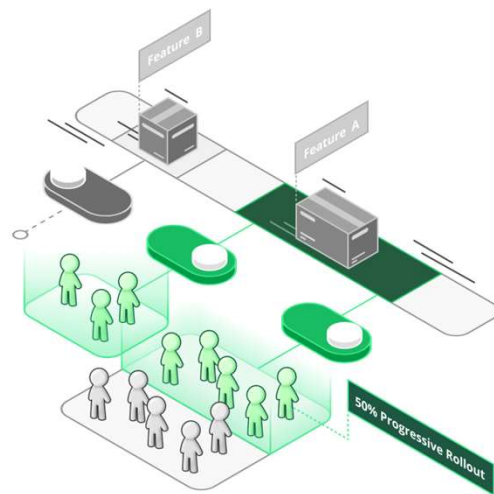
Complicated Issue Diagnosis

Removing the bug from production by reverting and redeploying code can increase the time it takes to restore the bugged feature. Because it is usually difficult for developers to find the root cause in development or test environments.

Solution: Feature Flag Management Tool

Use feature flags technology to deliver your software faster and with less risk.

- 1 Use a feature flag/toggle to decouple feature release from code deployment. If a single feature fails, you can turn it off without affecting other features.
 - 2 Testing each feature in production then rolling it out to a small percentage of users can help identify problems early, reduce the impact radius, and ultimately mitigate revenue loss.
 - 3 The ability to granularly roll back features with issues and keep them in production helps to quickly identify and resolve root causes.
 - 4 Feature flags isolate issues, preventing blockages in the planned development and deployment schedule and ensuring that overall development velocity is not compromised.
 - 5 Use feature flags to migrate infrastructure/database smoothly and incrementally. Keep the old and new systems running in parallel and migrate to the new system incrementally.
 - 6 Use the Feature Management tool to empower all teams to release and control software, fostering collaboration and innovation across the organization.
- ... And so on.



Feature-to-release ON

Testing in Production Rule

If User is in segment Beta Users Internal QA

serve true

Progressively Rollout to Early Adopter, Rollback in 1 sec when incident occurred

If User is in segment Early Adopter

serve Rollout percentage 10 % true

90 % false

Why not build your own feature flag tool

You might face challenges with flexibility, scalability, time and cost efficiency, and knowledge gaps.

Advanced Targeting and Segmentation

Beyond basic toggle on/off and audience targeting, sophisticated feature flagging involves complex segmentation capabilities. This includes targeting based on user behavior, geography, device type, age, and other custom properties, which can be resource-intensive to develop and maintain.

Real-time Rollout and Rollback

Implementing and maintaining the ability to update and propagate feature flags in real-time without needing a page refresh or application restart is technical challenges. A custom system may lack the finesse for seamless rollouts and instant rollbacks in case of issues.

Integration Challenges

Integrating a feature flagging system with your workflow, data, and technical stack can be complex and time-consuming. For example, you need to design an SDK with thinking of caching, error handling, reconnections, and multithreading in mind. Design a flexible and scalable webhooks for workflow consistency. Design a robust data integration for data teams and operational support.

Control Across All Teams

A custom feature flag system may miss essential management tools like detailed access controls, clear approvals, audit logs, and a user-friendly interface, hindering product and marketing teams' independent release management and leading to poor communication and teamwork.

Scalability

A custom system may not efficiently handle the growing number of features and users (e.g., 300K+ DAU), potentially causing performance issues that affect user experience and system manageability.

A/B Testing and Analytics Integration

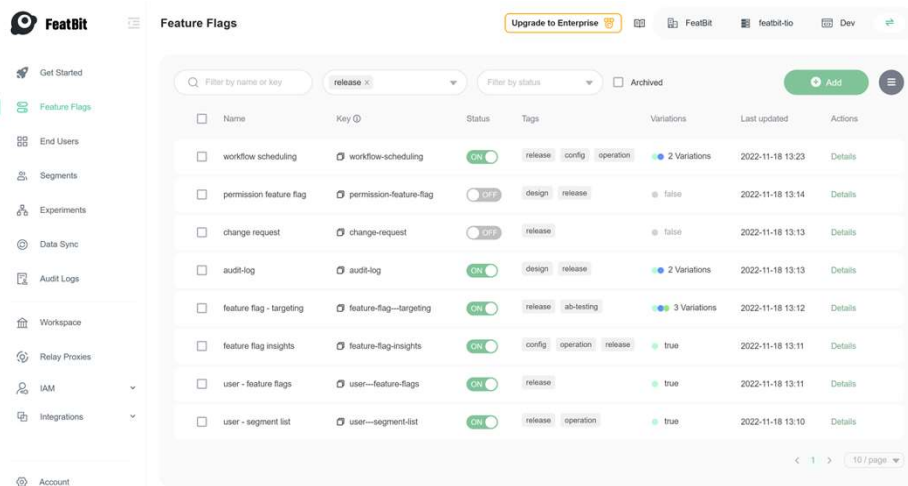
Seamlessly integrating with A/B testing frameworks and analytics tools to assess the impact of feature releases on user behavior and business metrics is crucial. Building and maintaining these integrations in a custom system can be challenging.

Self-build feature toggle can have bug, if the people who develop it leave the team, this will be a NIGHTMARE !!!

...

FeatBit Eases Your Pain Points

A feature management service that addresses the challenges of software delivery and innovation.



- 1 Eases the challenges associated with software delivery performance, feature release risk, innovation speed and custom feature management systems
- 2 Aims to offer an easy-to-use, scalable, and reliable self-hosted feature flagging solution with cost-effective and transparent pricing.
- 3 FeatBit not only provides robust solutions but also offers technical and API support to accommodate the implementation of custom features for clients.
- 4 Enables all teams to release and control software. Launch your features not only in a dark and smart manner but also correctly and more swiftly.

FeatBit as a Solution

FeatBit provides extensive features to help you to deliver faster and safer



Decouple feature release from code deployment

Feature-to-release ON

Testing in Production Rule

If

serve

Progressively Rollout to Early Adopter, Rollback in 1 sec when incident occurred

If

serve true false

false

Attribute/Annotation Flag

```
[FeatureFlag("new-checkout-process", true)]
[HttpPost]
public async Task<CheckoutResult> CheckoutAsync(
    [FromBody]CheckoutRequest request){
    return await _checkoutService.CheckoutAsync(request);
}
```

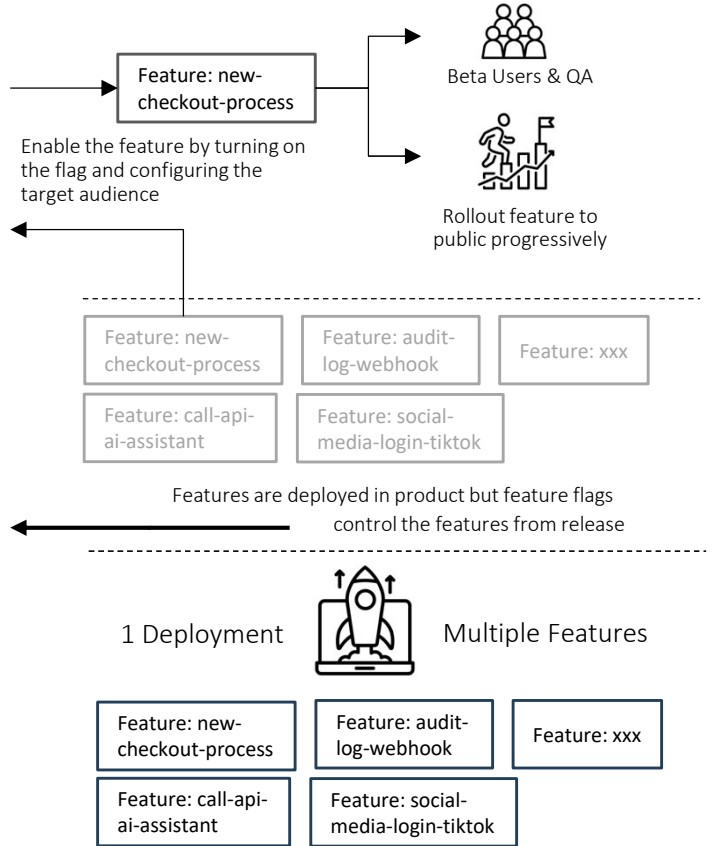
Component Flag

```
<.../>
<FeatureFlag key="new-checkout-process" visibleValue="true">
  <CheckoutProcess user={...userInfo} orderId={orderId} />
</FeatureFlag>
<.../>
```

If/Else Statement Flag

```
if(fbClient.FeatureFlags("new-checkout-process") == true){
  return await _checkoutService.CheckoutAsync(request);
}
```

Use a feature flag to decouple feature release from code deployment. If a single feature fails, you can turn it off without affecting other features. This allows you to test each feature in production and roll it out to a small percentage of users to catch problems early.



Advanced Targeting and Segmentation

Using advanced targeting and segmentation that you can target based on user behavior, geography, device type, age, and other customized properties. This allows you to target the right audience for your features and test them in production.

Feature Flags > Targeting Upgrade to Enterprise FeatBit featbit-to Dev

Checkout Process with Bit...

A new checkout process that allows crypto users to pay for the service.

Key

checkout-process-with-bitcoin

The feature flag is ON

If OFF, serve:

Variations

Data type: boolean

True

False

Tags

Targeting rules

Testing Production by QA team

If

Serve:

Canary release to our own and out sourcing employees

If

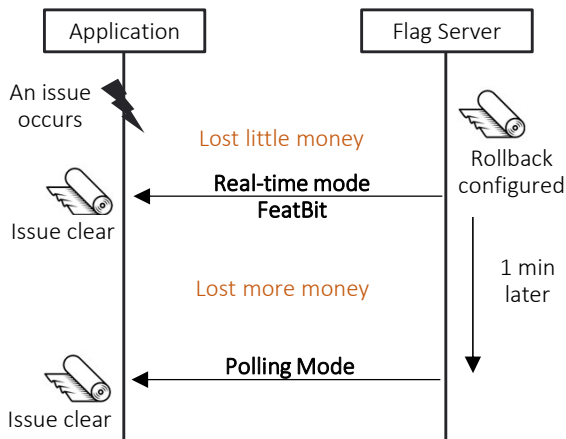
Serve: True False

False

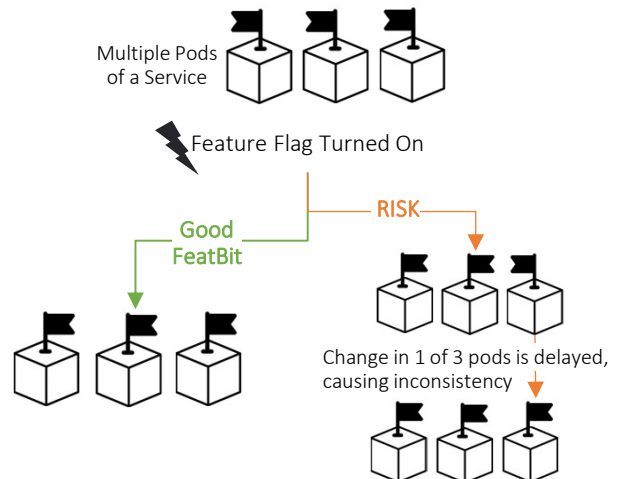
Dispatch By:

FeatBit can do Real-time Rollout and Rollback

Real-time feature's kill switch, rollout and rollback can be critical, such as a malfunction in the new checkout process in a e-commerce platform, incorrect transactions which resulting in financial loss in a financial service, and so on.



A feature flag implemented in multiple pods backend service, a polling (or long-polling) mechanism may involve an inconsistent state, which may lead to a potential risk.



A variety of SDKs to support different languages and to be used in different contexts

FeatBit provides SDKs for various languages, including .NET, Java, Python, Javascript Node.js, Go, and more, to support different use contexts such as front-end, back-end, mobile, and so on.

FeatBit did many works to ensure the feature flag evaluation result is consistent, real-time and reliable. Such as:

- Retry mechanism, SDK should be able to reconnect to the remote evaluation server when the connection is lost.
- Cache mechanism, for Client-side SDK should be able to cache the feature flag evaluation result to reduce the network request.
- Local evaluation mechanism, for Server-side SDK should be able to evaluate the feature flag locally to reduce the network request.
- Thread-safe, SDK should be able to handle the multi-threading use context without performance issue and crack issue.
- WebSocket/SSE/gRPC, SDK is designed to receive the feature flag change event in real-time in a reliable manner.
- And so on.

FeatBit also offers OpenFeature providers to help teams easily switch between services and use feature flags the right way.

Besides their usual software kits, FeatBit provides Rest APIs to check feature flag results for programming languages that don't have a special FeatBit kit.

The screenshot shows a list of GitHub repositories for FeatBit SDKs. Each entry includes the repository name, language, license, and update date. The repositories listed are:

- [featbit-dotnet-client-sdk](#) (Public) - HTML, Apache-2.0, Updated 3 days ago
- [featbit-java-sdk](#) (Public) - Java, MIT, Updated 2 weeks ago
- [featbit-python-sdk](#) (Public) - Python, MIT, Updated on Feb 5
- [featbit-js-client-sdk](#) (Public) - JavaScript client side SDK for the feature management platform featbit, TypeScript, Apache-2.0, Updated on Jan 22
- [featbit-dotnet-sdk](#) (Public) - FeatBit Server-Side SDK for .NET, C#, Apache-2.0, Updated on Jan 11
- [openfeature-provider-node-server](#) (Public) - A OpenFeature provider implementation for the FeatBit node SDK, TypeScript, MIT, Updated on Jan 9
- [featbit-node-server-sdk](#) (Public) - FeatBit Server-Side SDK for Nodejs, TypeScript, MIT, Updated on Jan 8
- [featbit-go-sdk](#) (Public) - Go, MIT, Updated on Dec 21, 2023
- [openfeature-provider-js-client](#) (Public) - TypeScript, MIT, Updated on Jan 23
- [featbit-openfeature-provider-java-server](#) (Public) - Java, MIT, Updated on Jan 5

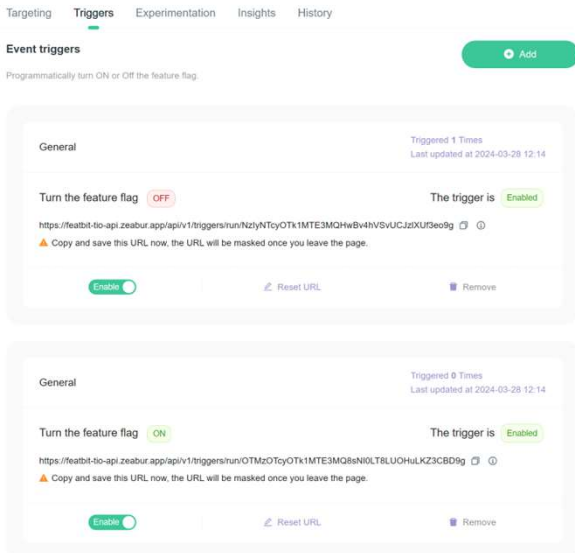
Automate your feature release process

Use FeatBit's rich webhook, scheduling, change approval, trigger and Rest APIs to automate your feature release process

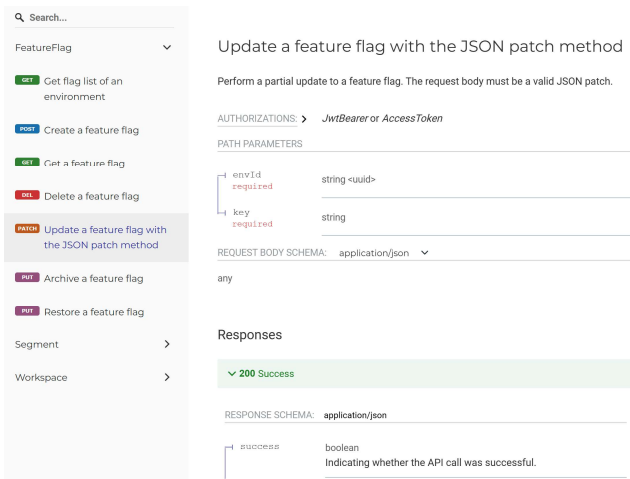
Use FeatBit Scheduling to release new features at a predefined date and time.

Use FeatBit Change Approval to allow other team members (e.g. product owner) to approve the release.

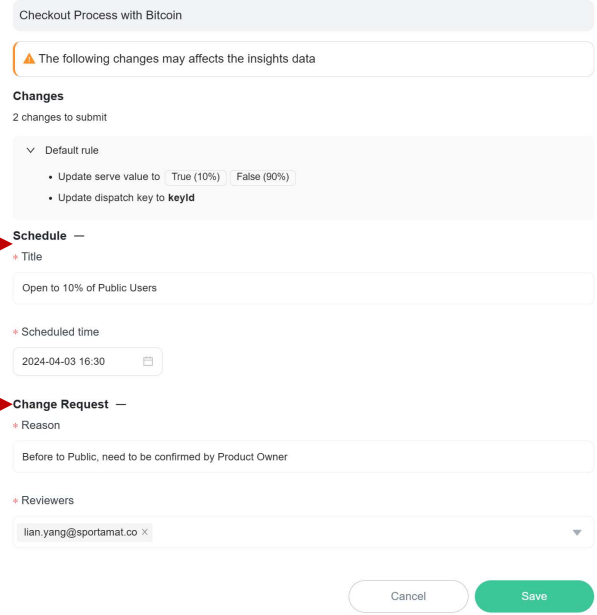
Enable/disable a feature using an automatic trigger, such as a webhook from your CI/CD pipeline.



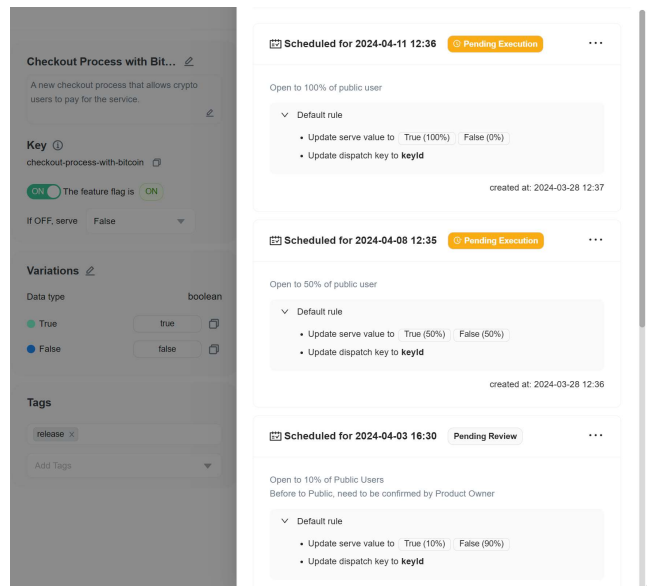
Update a feature's flag release configuration using a Rest API sent from another system.



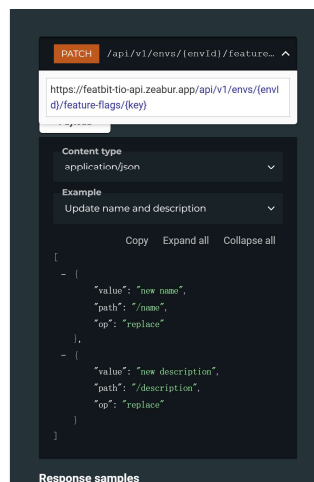
Review and save



Combine the above features to create a complex release schedule.



Restful Web API



Integration with your team cooperation workflow

When a feature flag config changes, we need to receive a notifications in our working tools to get the latest state of a feature.

- Chat tool, such as Slack, Microsoft Teams, etc.
- Project management tool, such as Jira, Trello, etc.
- Other tools that need to receive notification..



Send message to 3rd party tool. For example, call the Post API interface to send a feature flag switch-on event to Slack.

Rich WebHook

You can subscribe to the FeatBit webhook. When a feature flag or segment changes, the webhook will send your customized message to the endpoints you want to receive it.

Edit your custom message with the details of the feature flag, segment, and its changes.

Subscribe to the change events and the scope



Send feature flag rollout / rollback event to the APM tool. For example, configure the event in New Relic One as a deployment event.

Detect, Rollback and Fix Issue easier and faster

Integrate FeatBit with observability tool (like New Relic, Grafana, Datadog, and so on).

Observability tool detect issue during the rollout and showing the issue graph and figure out the related feature flag.

APM tool sends an alert to the team or calls a prepared endpoint such as FeatBit Trigger or FeatBit Rest API.

4

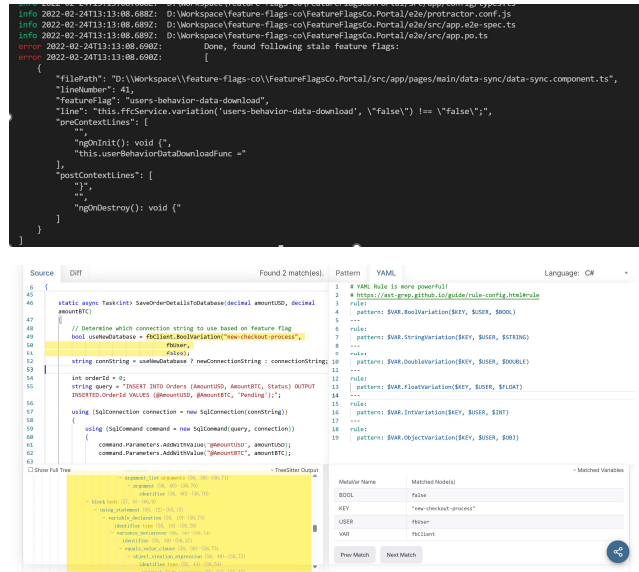
Feature is disabled or rolled back by you, the Rest API, or a program endpoint.



Integration with your development workflow

FeatBit offers a variety of features to let you easily integrate with your development workflow and make the work efficiency.

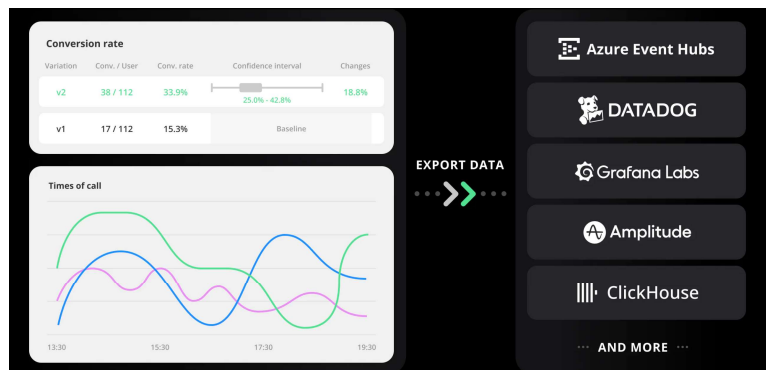
- Use feature flag to reduce long-lived feature branches and merge conflicts.
- Multiple environment design enable you testing feature flag in dev and staging environment before release to production.
- Use FeatBit Rest API to remote control your CI/CD job flow in pipeline such as GitHub Action.
- FeatBit Code Reference simplifies tracking which projects use your feature flags, making it easier to reduce technical debt.
- Monitoring the usage of feature flags to let you know which feature flags are not used and can be removed.



Continuously experiment by capturing and analyzing feature data

FeatBit is a tool for capturing and analyzing feature data.

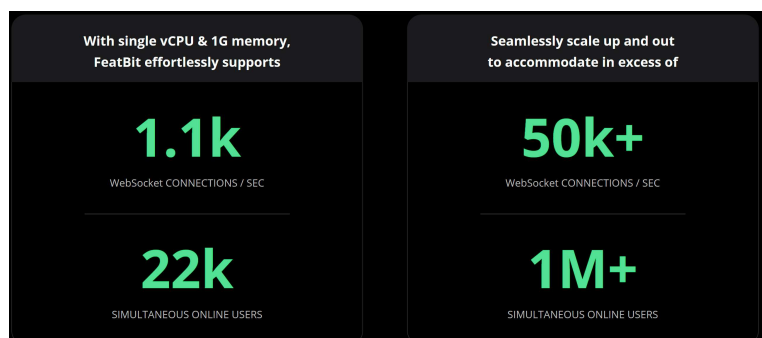
- Use FeatBit's Insights to see how features are performing, how they are being used.
- Use FeatBit's Experimentation to analyze how they are impacting your experience.
- Export collected data to further data analysis to make your business decision and can be removed.



Scalability and High Performance

FeatBit is designed to be highly scalable and performant, ensuring that it can handle the increased load efficiently as the number of features and the scale of your user base grow. This ensures a seamless user experience and the manageability of the system.

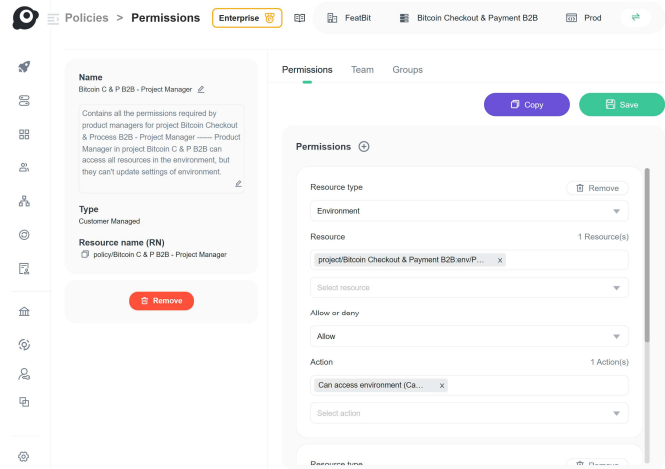
We chose .NET 6+ as the primary programming language, which has the fast performance of Go and the rich features of Java. It's designed for modern IT infrastructures such as Cloud Native, Container Native, Linux Native, and so on.



Release and control your features by all team

It could be an overall efficiency improvement if feature owners (like product and business team) could directly control feature release without the help of development team. With FeatBit:

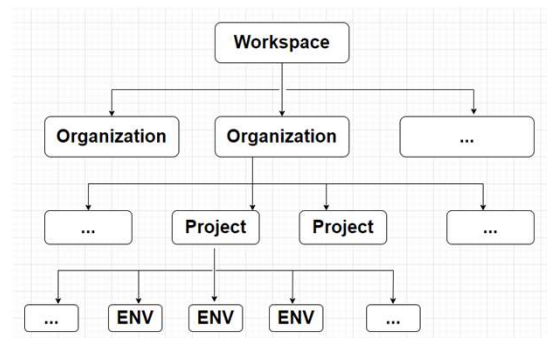
- Product and business team can directly control the feature release with an easy to use interface.
- FeatBit's IAM can give team members access permission only to it's own project and feature flags. This avoids the risk of misuse.
- FeatBit's approval can help ensure that the feature release plan is approved by the right person.
- FeatBit's Audit Log can help to track the operation history to avoid the dispute.



Manage organizations, projects and teams in a unified workspace

FeatBit simplifies working by managing all organizations, projects and teams in a unified workspace:

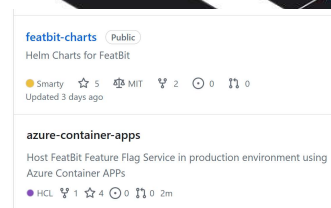
- Maintain the entire system in a single location to reduce redundancy.
- A new organization, project can be quickly created by a few clicks, which quickly responds to the team request.
- Each organization, project and environment can have its own members and access control.
- Everyone in the organization uses a standard way to manage their feature flags, reducing the overall learning curve.
- DevOps performance can be easily measured by accessing one place and one system.



Easy Self-Hosting and Maintain the Service

FeatBit offers options for both hosted and self-hosted solutions, operating globally to provide seamless information synchronization and efficient proxy bridging.

FeatBit simplifies the hosting process and future maintenance tasks by providing Docker Compose, Helm Charts, Terraform, and more. We also offer technical support for self-deployment, upgrades, multi-data center solutions, and beyond.



Thanks

“ In the aftermath of their decision to use FeatBit, the engineering team's work life was transformed. By adopting feature flags, they deployed updates more safely and efficiently, significantly reducing the chaos caused by bugs. This strategic shift not only cut down on stressful, unplanned overtime but also improved their work-life balance. Gradual rollouts became the norm, allowing for easy monitoring and quick rollbacks without widespread disruptions. The dreaded midnight alarms faded into memory, replaced by a newfound confidence in their deployment process.

The positive changes went beyond technical improvements. The team now enjoyed more personal time and less stress, fostering an environment ripe for innovation and growth. FeatBit wasn't just a tool; it was a catalyst for a healthier, more balanced approach to engineering. As they embraced this new way of working, the engineers became advocates for change, sharing their success story with the broader tech community. In doing so, they not only reclaimed their own time but also set a new standard for software development practices.

”

Contact Us



Website: <https://www.featbit.co>

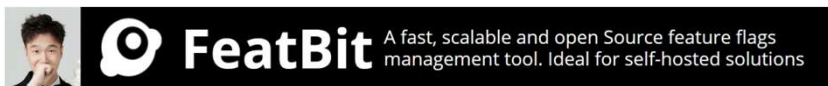
E-mail: contact@featbit.co

GitHub: <https://github.com/featbit/featbit>

Linkedin: <https://linkedin.com/company/featbit>

Slack: <https://shorturl.at/lxyAR>

Beau HU, CEO



E-mail: beau.hu@featbit.co

X.com: twitter.com/realfeatbit

Linkedin: <https://www.linkedin.com/in/beau-hu-04685037/>

Our Advisor



PhD in Computer Science from Carnegie Mellon University

Founder & CEO of MiraclePlus

Former President of Baidu Group

Former Executive Vice President at Microsoft

Former Executive Vice President at Yahoo

A Snapshot of Our Story

FeatBit was founded in 2021 and received its first investment from Qi Lu. Our team operates in a remote mode, with members located across the globe. We boast strong expertise in software engineering and DevOps, with several team members certified as Experts by platforms like Azure, AWS, and others. Our people have diverse backgrounds, having worked in various industries and for companies such as Yahoo, TotalEnergies, TikTok, and BNP Paribas. Together, we are building FeatBit to elevate software innovation to new heights through advanced feature management.