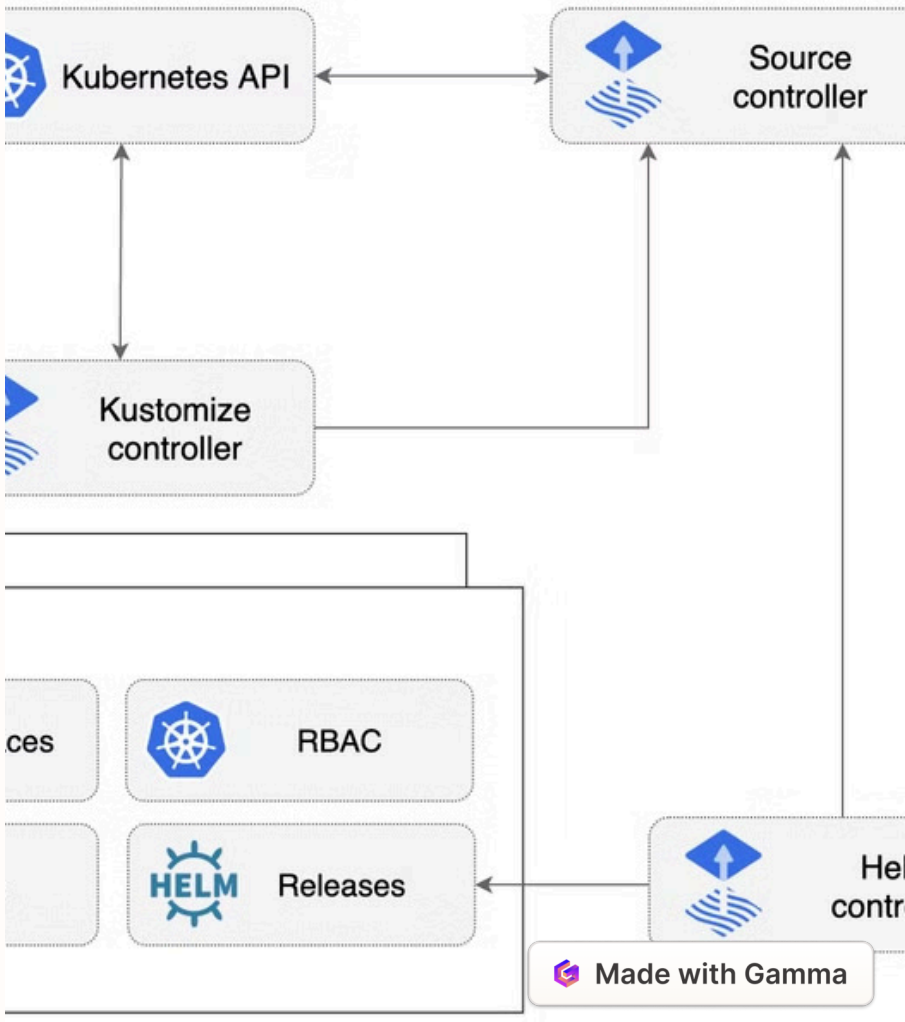
GitOps

GitOps - это новый подход к управлению инфраструктурой и развертыванием приложений, основанный на использовании систем контроля версий, таких как Git, для хранения и управления всеми конфигурациями и настройками. Этот метод позволяет обеспечить прозрачность, воспроизводимость и автоматизацию процессов развертывания и обслуживания ИТ-инфраструктуры.

by Илья Исаев



Что такое GitOps?

GitOps - это подход к управлению инфраструктурой, при котором вся конфигурация и развёртывание определяются декларативно с помощью текстовых файлов, хранящихся в системе контроля версий, таких как Git.

В GitOps вся инфраструктура описывается в виде кода, что позволяет применять к ней те же принципы, что и к разработке приложений: автоматизация, версионирование, ревью кода и откат изменений.

Version Control System (VCS)

- SVN (Subversion) Concurrent Versions System
- Mercurial
- Git

> I'm an egotistical bastard, so I name all my projects after myself. First Linux, now git.
>
> Я эгоистичный ублюдок, и поэтому называю все свои проекты в честь себя. Сначала Linux, теперь git.
>
> Линус Торвальдс

Git

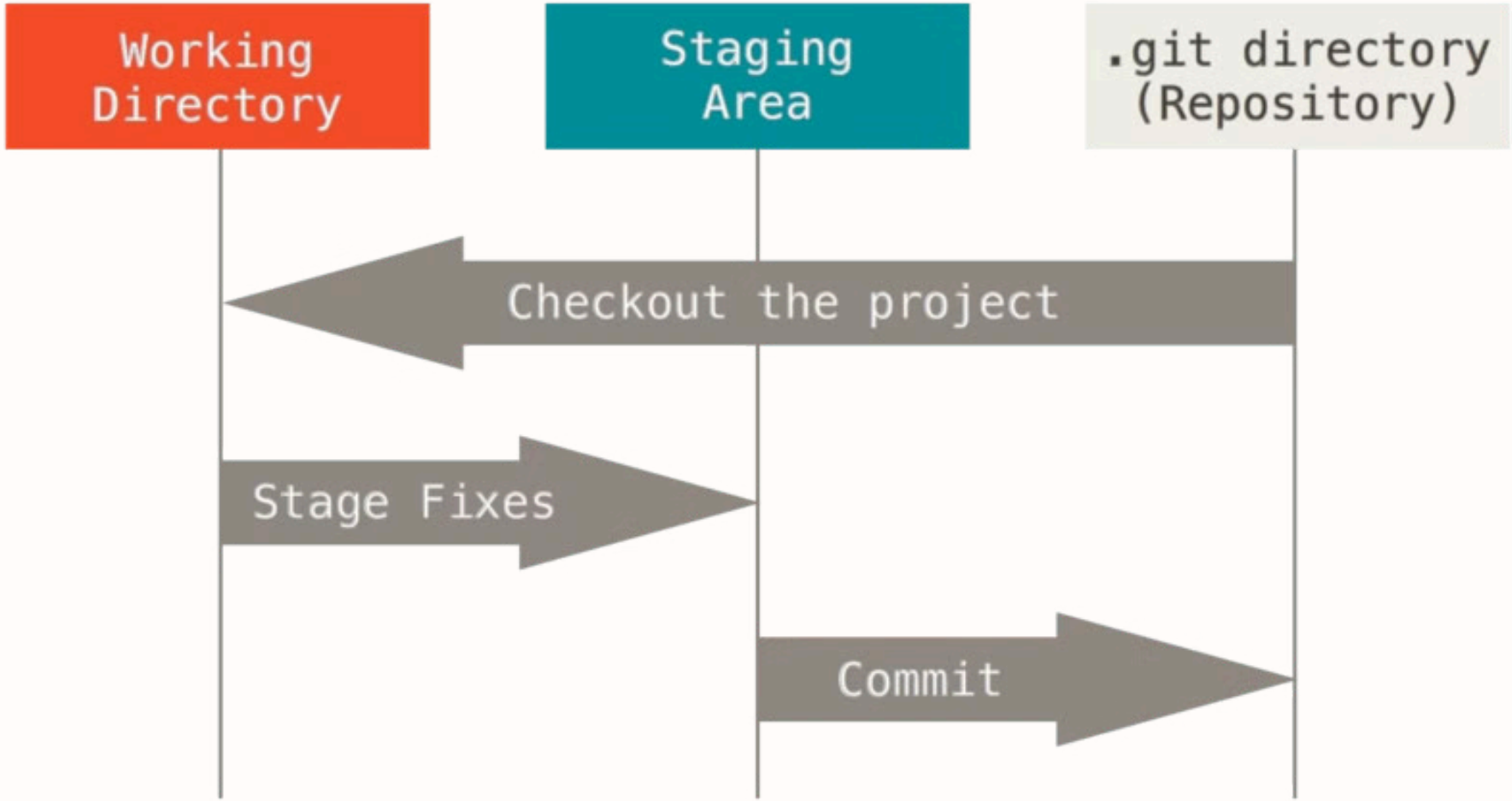
Достоинства

- Гибкость и удобства использования
- Скорость
- Широкая функциональность
- Открытость решения

Недостатки

- Нет Merge Requests
- Нет UI

Хранение данных



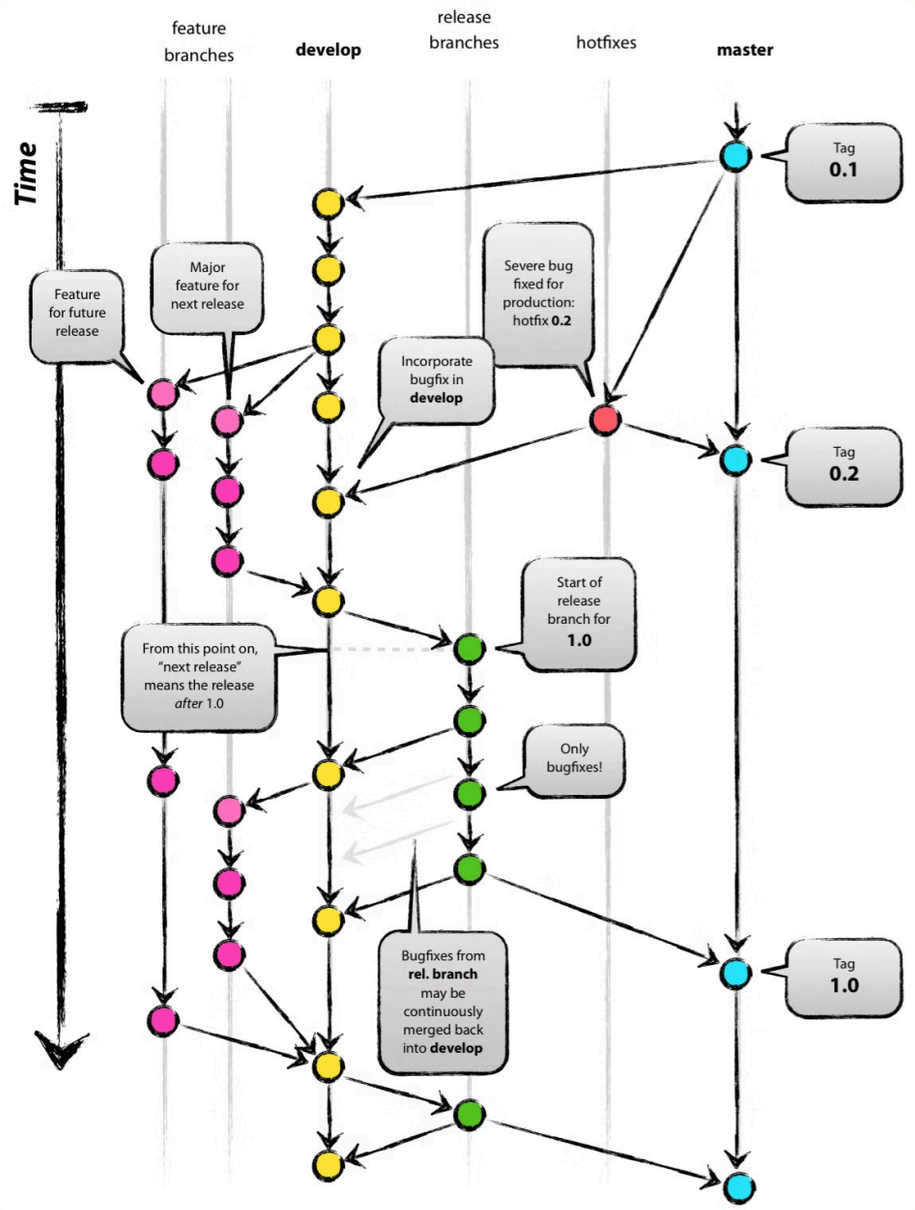
Репозитории

- GitHub
- BitBucket
- GitLab
- Gitea
- GitVerse
- GitFlic

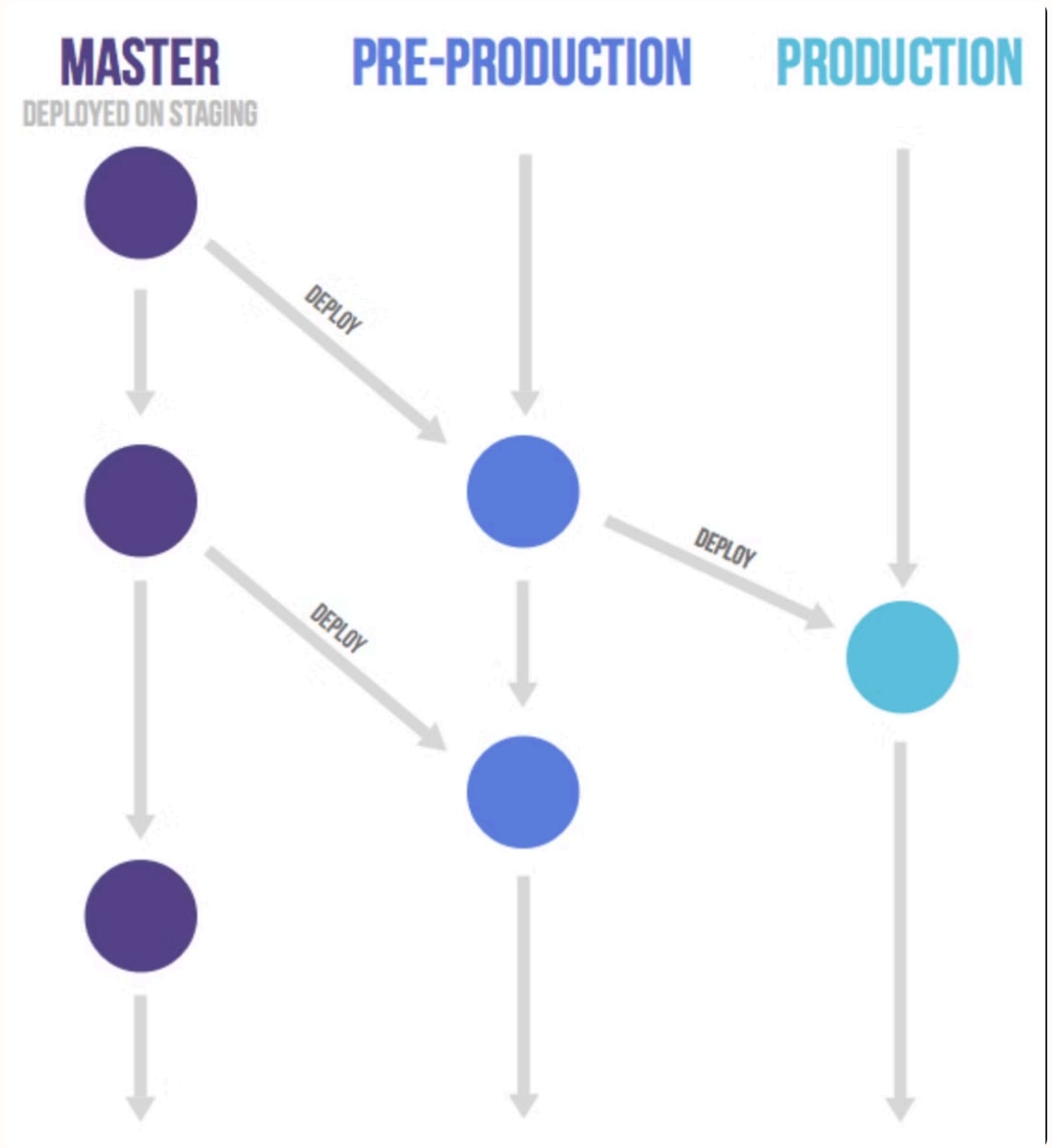
Git Flows

- GitFlow
- GitLab Flow
- GitHub Flow

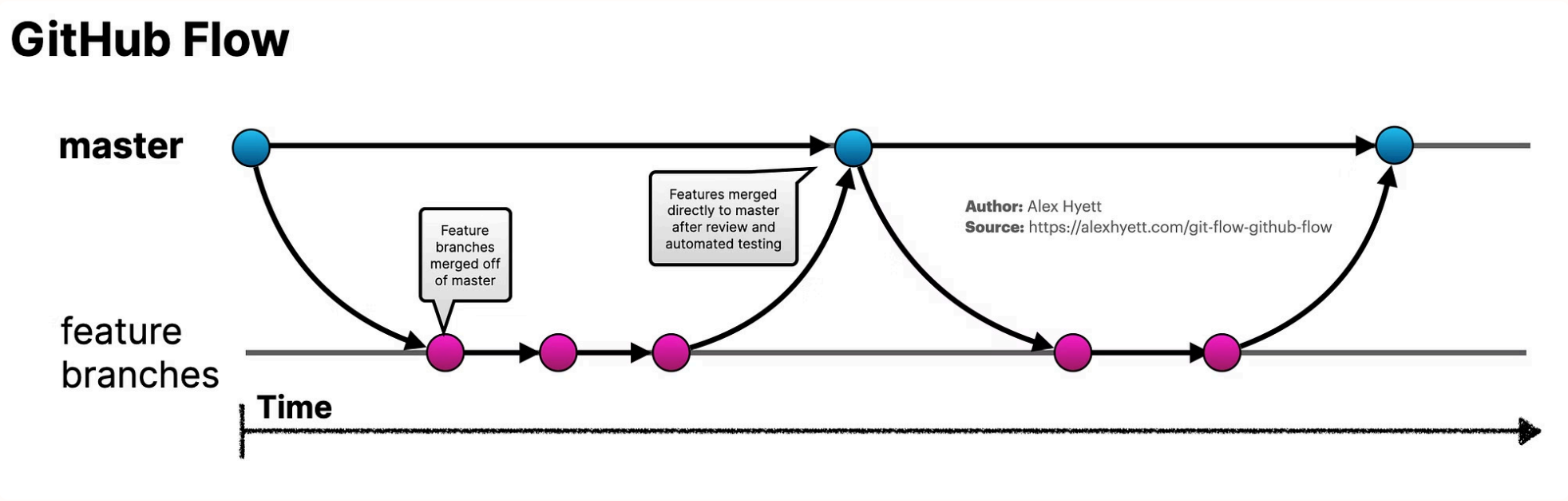
GitFlow



GitLab Flow



GitHub Flow



Conventional Commits

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

- feat (feat: allow provided config object to extend other configs)
- fix (fix: prevent racing of requests)
- feat! (feat!: send an email to the customer when a product is shipped)
- docs (docs: correct spelling of CHANGELOG)
- refactor, etc

Conventional Commits

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

Bad Practice

- Коммиты на русском языке
- Commit message not conventional
- Parallel git pull
- Хранения больших файлов в git (Artifactory, Nexus)
- —force
- Не делать бэкап git
- Работать в Web IDE

Полезности

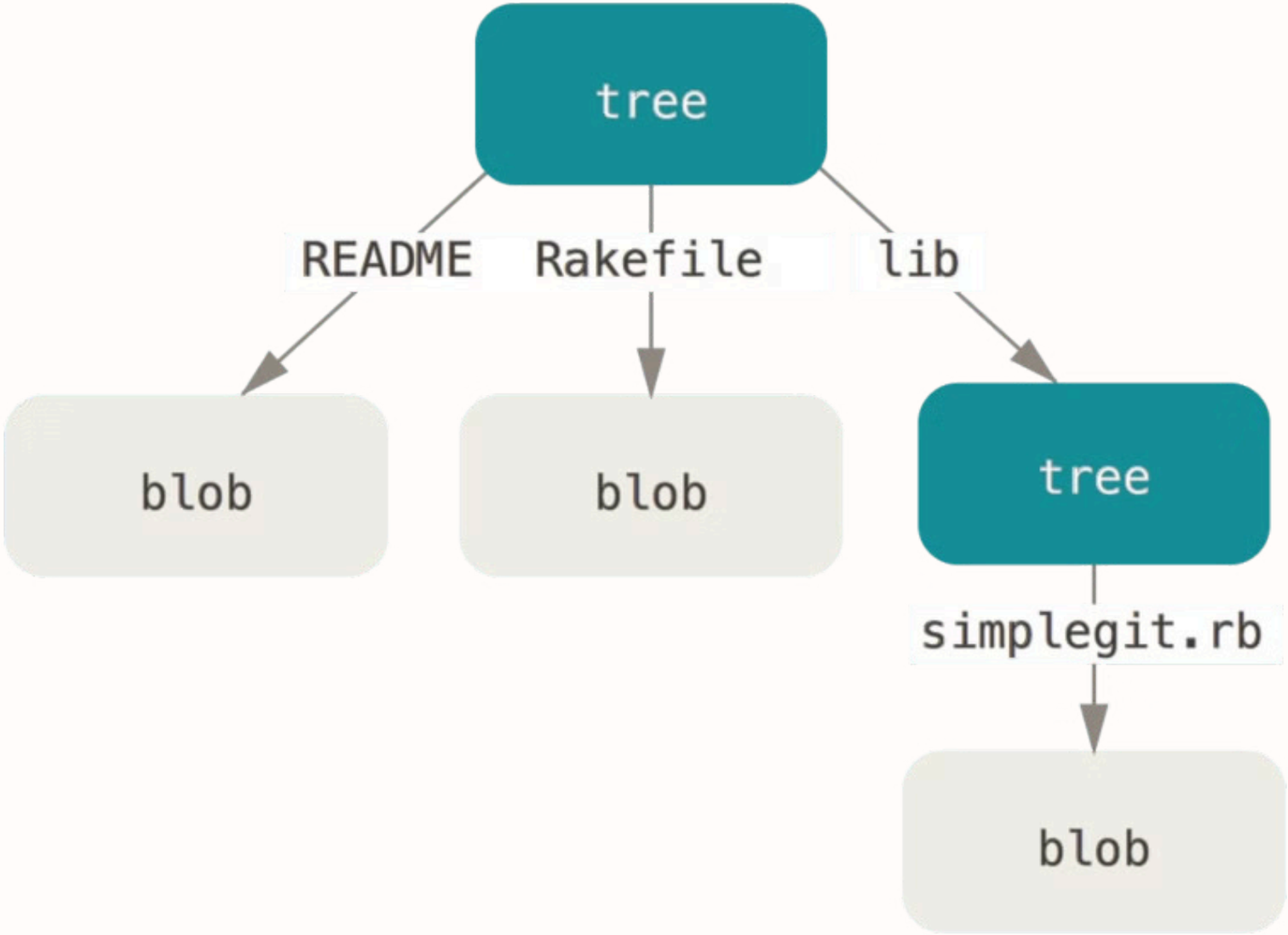
- Тулы для проверки коммитов (convco)
- AI ассистенты кода (в IDE)
- Свои практики в компании

Git внутри

- config (конфиги репки)
- description
- HEAD (файл указывает на текущую ветку)
- hooks/ (клиентские и серверные хуки)
- info/ (глобальные настройки из .gitignore)
- objects/ (база данных объектов)
- refs/ (ссылки на объекты)

Дерево

- blob - объект



Интересные команды

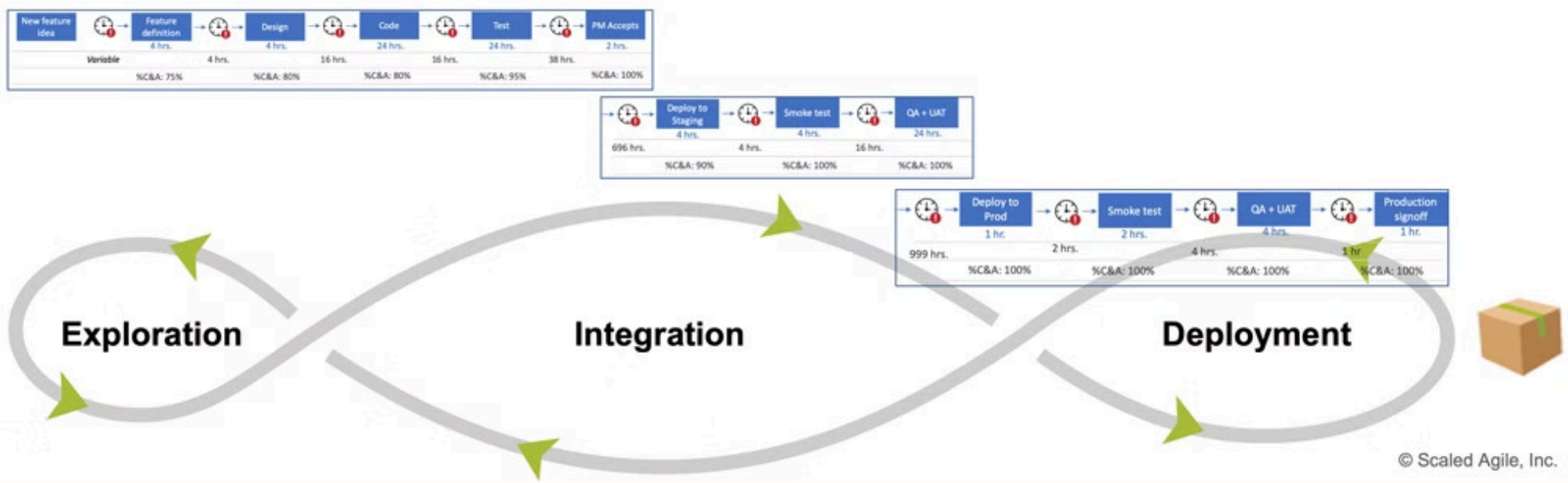
- git init
- git clone
- git add
- git commit
- git push
- git pull
- git rebase
- git merge
- git cherry-pick
- git stash

git dogs

Pipeline

- Сборка
- Тестирование
- Релиз
- Деплой

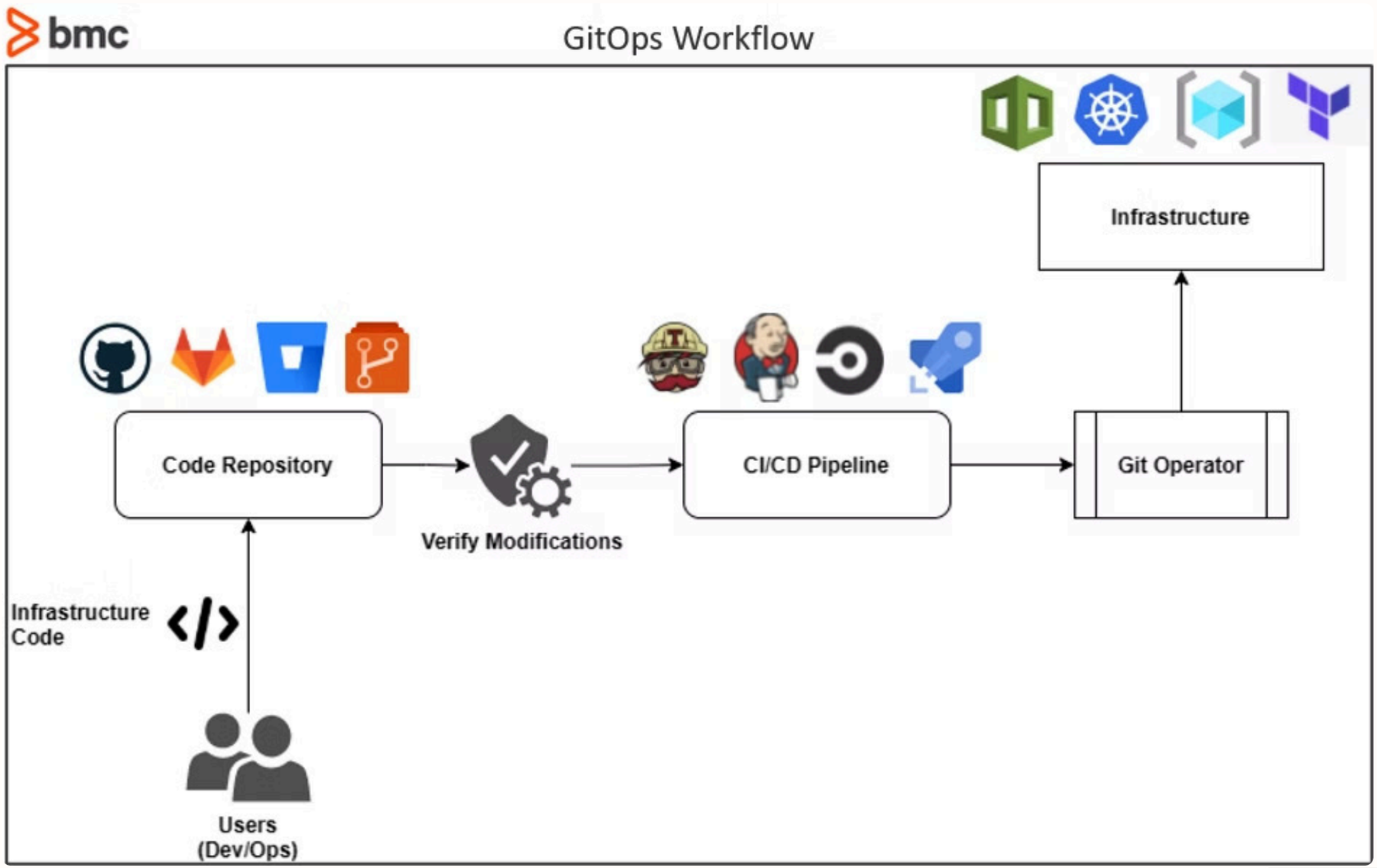
Continuous Deployment



GitOps Weaveworks

- Git-репозиторий с описанием инфраструктуры
- Автоматизированный процесс синхронизации

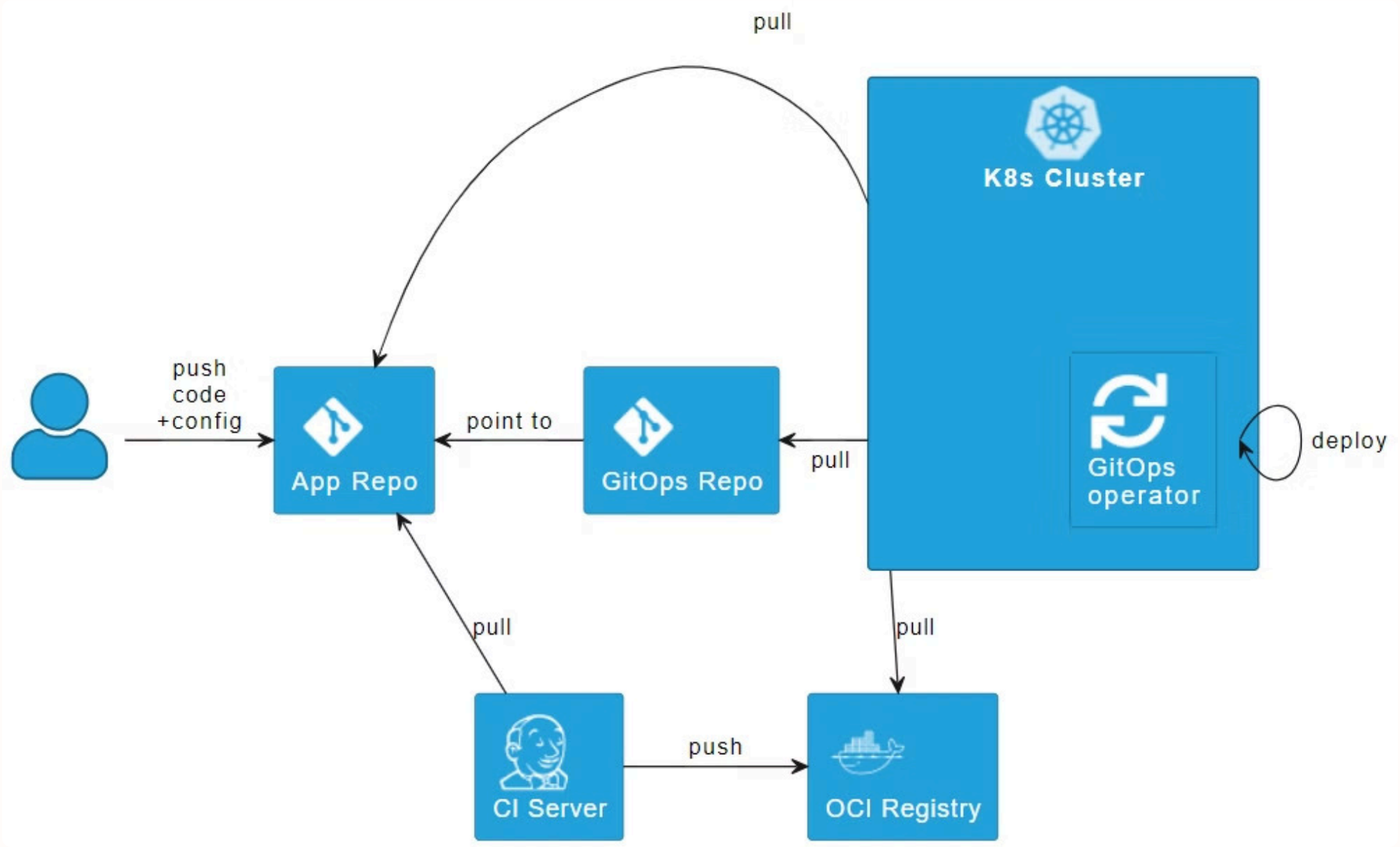
Continuous Reconciliation Loop



Tools

- Argo CD
- Flux CD

GitOps Two Repo



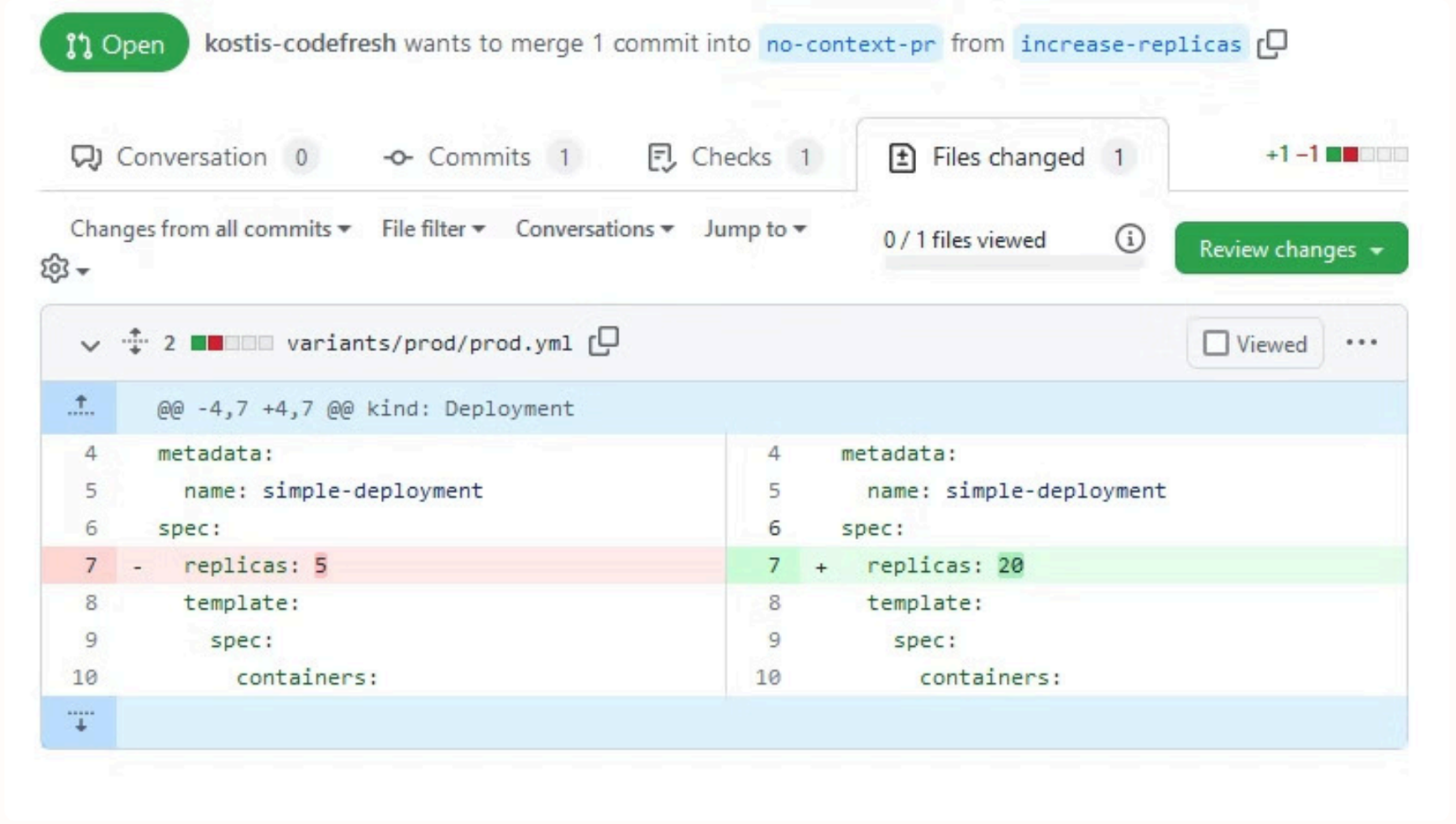
Выбор подходящего метода организации репозитория

- мультирепа с конфигами
- монорепа с конфигами

Проверка манифестов перед коммитом

- helm template ./
- kubectl apply —dry-run=client -f ./
- argocd app diff MY_APP —local ./

Манифесты не должны изменяться от внешних факторов

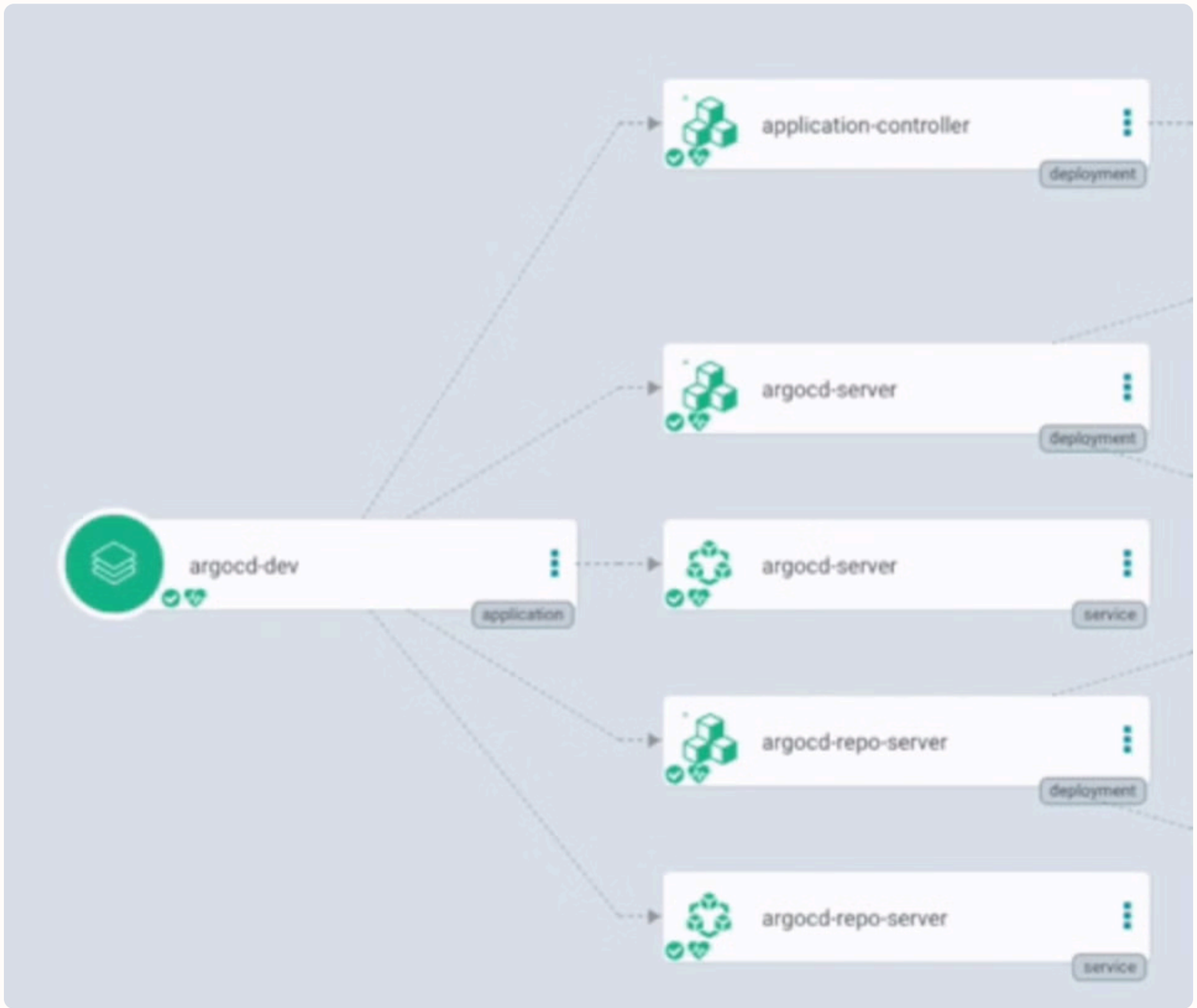


- OutOfSync

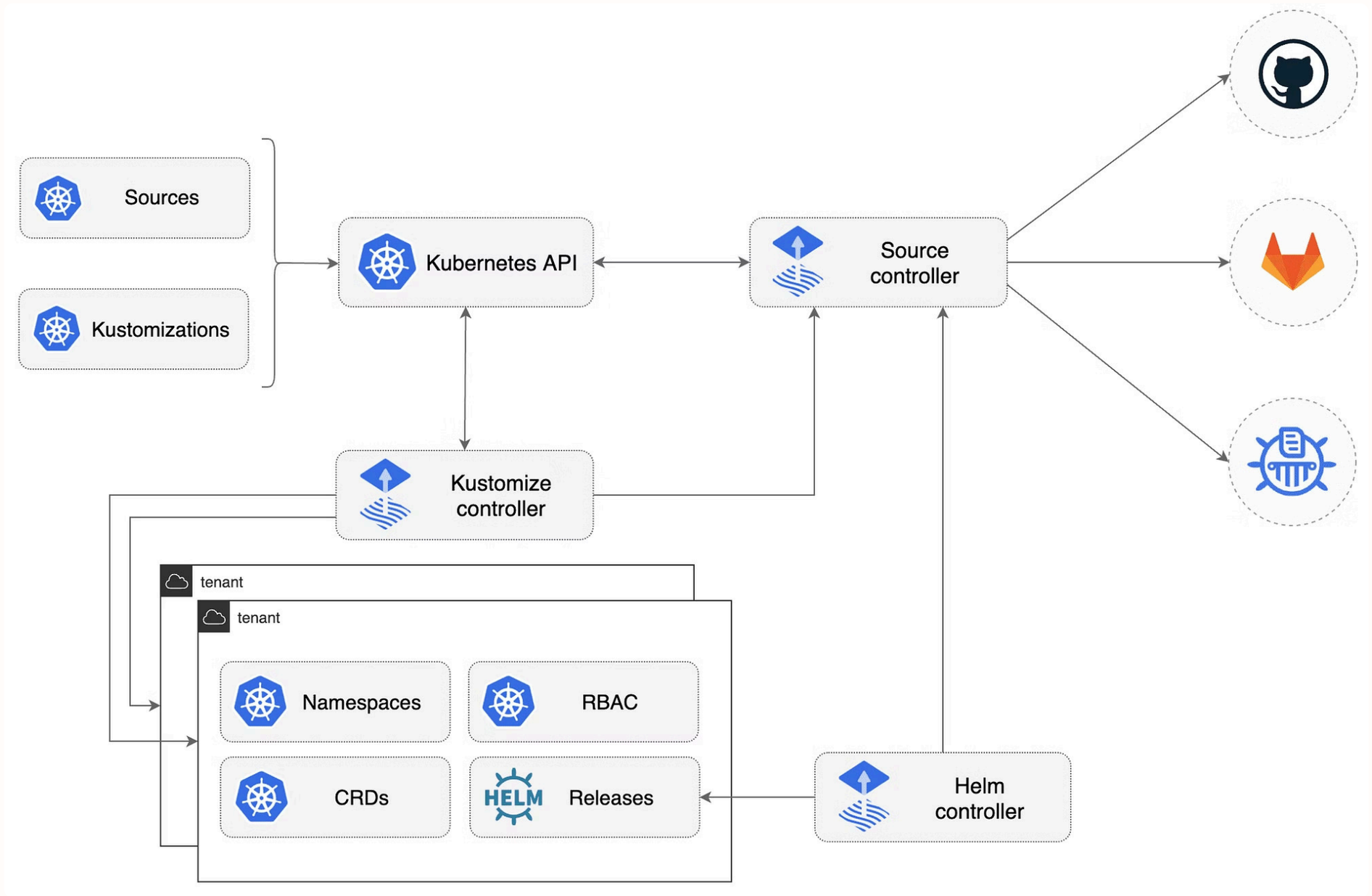
Хранение секретов

- SOPS
- git-crypt

Argo CD



Flux CD (Flux 2)



Как быть?

- Деплоить унифицировано, одинаковый набор компонентов в большое количество кластеров, то лучше сделать выбор в пользу Flux CD. Ввиду использования парадигмы, где чаще всего используется отдельный независимый контроллер на каждый Kubernetes кластер
- Быстро внедрить GitOps, а ваша команда разработчиков и инфраструктура к этому ещё не готова. Если ваши разработчики хотят выполнять операции вроде рестарта отдельных подов, exec в контейнеры, а также возможность выполнять роллбэки без коммита в Git - то Argo CD может стать для вас хорошим выбором

Будущее GitOps и перспективы развития

Кроме того, растет спрос на безопасные и отказоустойчивые процессы развертывания, что стимулирует дальнейшее совершенствование методов аудита, контроля доступа и мониторинга в GitOps. Перспективным направлением является внедрение технологий искусственного интеллекта и машинного обучения для автоматизации принятия решений и прогнозирования проблем.