



Магия GitLab Auto DevOps:
самонастраивающийся
Pipeline-as-Code для вашего CI

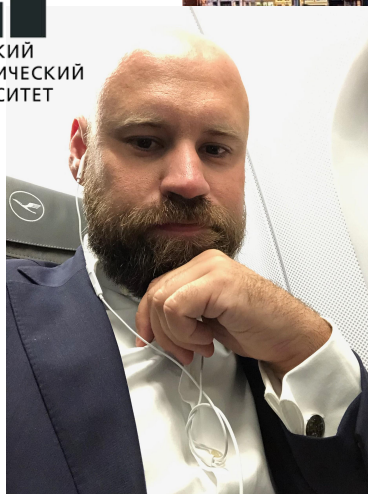




Vladimir Dzalbo

vladimir@dzalbo.com

Solutions Architect / GitLab





Company

- Incorporated in 2014
- 1300+ employees across 66 countries
- Fully remote
- GitLab Federal entity est. in 2018

Product

- Single Application / Complete DevOps Platform
- 100,000+ organizations
- Millions of users
- 70% share of self-managed DevOps repository market
- Open source model
- 3,000+ code contributors
- 10,000+ total contributors



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Defend

A Complete DevOps Platform Delivered as a Single Application

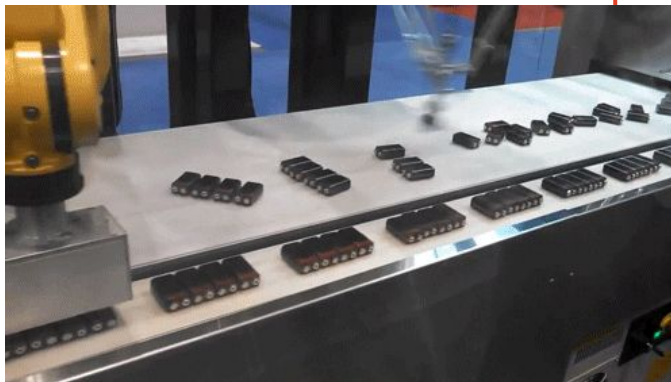


 Manage	 Plan	 Create	 Verify	 Package	 Secure	 Release	 Configure	 Monitor	 Defend
Since 2016 GitLab added: - Subgroups - Insights - Audit Events - Instance Statistics - Value Stream Management - DevOps Reports - Code Analytics - Compliance Management - Audit Reports	Since 2011 GitLab added: - Service Desk - Epics - Boards - Time Tracking - Issue Tracking - Requirements Management - Roadmaps Upcoming Categories: - Quality Management	Since 2011 GitLab added: - Code Review - Source Code Management - Snippets - Web IDE - Static Site Editor - Wiki - Design Management - Live Preview	Since 2012 GitLab added: - Continuous Integration (CI) - Merge Trains - Accessibility Testing - Usability Testing - Web Performance - Load Testing - Code Testing and Coverage - Code Quality	Since 2016 GitLab added: - Dependency Proxy - Container Registry - Package Registry - Git LFS - Jupyter Notebooks - Helm Chart Registry Upcoming Categories: - Dependency Firewall	Since 2017 GitLab added: - Secret Detection - License Compliance - Container Scanning - Dependency Scanning - DAST - SAST - Vulnerability Management - Fuzz Testing	Since 2016 GitLab added: - Review Apps - Pages - Continuous Delivery - Secrets Management - Release Orchestration - Feature Flags - Advanced Deployments - Release Evidence	Since 2018 GitLab added: - Infrastructure as Code - Kubernetes Management - Auto DevOps - Cluster Cost Optimization - Serverless - Runbooks - ChatOps	Since 2017 GitLab added: - Error Tracking - Logging - Incident Management - Metrics - Tracing - Alert Management Upcoming Categories: - Synthetic Monitoring	Since 2019 GitLab added: - Container Network Security - Container Host Security - WAF



SCM / Репозитории

- Git repos
- Merge Requests
- Code Review
- Approvals
- Web IDE
- ...



Автоматизация

- Continuous Integration
- Continuous Delivery
- Deployments
- ...



Enabling Collaboration
Bringing Ideas to Reality

Convention over Configuration

MVC

Flow One

Avoid Plugins

Iteration



Принципы:

- Меньше конфигурирования (только там, где надо)
- Больше конвенций / стандартов
- Использование “лучших практик”
- Продукт для пользователей, а не администраторов
- Default to ON (новые функционал доступен по дефолту)
- Избегайте ограничений и лимитов



Коммерческие требования:

- Работает “из коробки”
- Функционал легко найти и начать использовать
- Настройка и усложнение по мере надобности



SCM / Репозитории

- Установка GitLab
yum / apt-get / docker / helm
- Create Project 
- git clone / git commit / git push



Автоматизация





- Pipelines состоят из ...
 - Jobs - задачи и контекст их выполнения...
 - Scripts (команды)
- Pipeline-as-Code
 - YAML
 - Хранится в git репозитории
- Множество триггеров:
 - git push
 - UI / API
 - Schedule
 - ... и так далее
- Container based (by default)
- Выполняется на кросс-платформенных агентах (GitLab Runner)

```
1  image: alpine:latest
2
3  stages:
4    - build
5    - test
6
7  variables:
8    SEARCH_MAX_DEPTH: 99
9
10 consoleapp_build:
11   image: mcr.microsoft.com/dotnet/core/sdk:3.1
12   cache:
13     key: ${CI_COMMIT_REF_SLUG}-dotnetapp
14     paths:
15       - app/dotnetapp/out/
16     policy: push
17   stage: build
18   script:
19     - |
20       mkdir -p app/dotnetapp
21       mkdir -p app/Utils
22       cd samples/dotnetapp
23       cp dotnetapp/*.csproj ../../app/dotnetapp/
24       cp Utils/*.csproj ../../app/Utils/
25       cd ../../app/dotnetapp
26       dotnet restore
27       cp ../../samples/dotnetapp/dotnetapp/* ../dotnetapp/
28       cp ../../samples/dotnetapp/Utils/* ../Utils/
29       dotnet publish -c Release -o out
```



Проблемы:

- Кастомный синтаксис *.gitlab-ci.yml*
- Функционал (и синтаксис) постоянно разрастается / усложняется
- Зависит от фреймворка / языка программирования
- Является ключевым для адаптации другого функционала



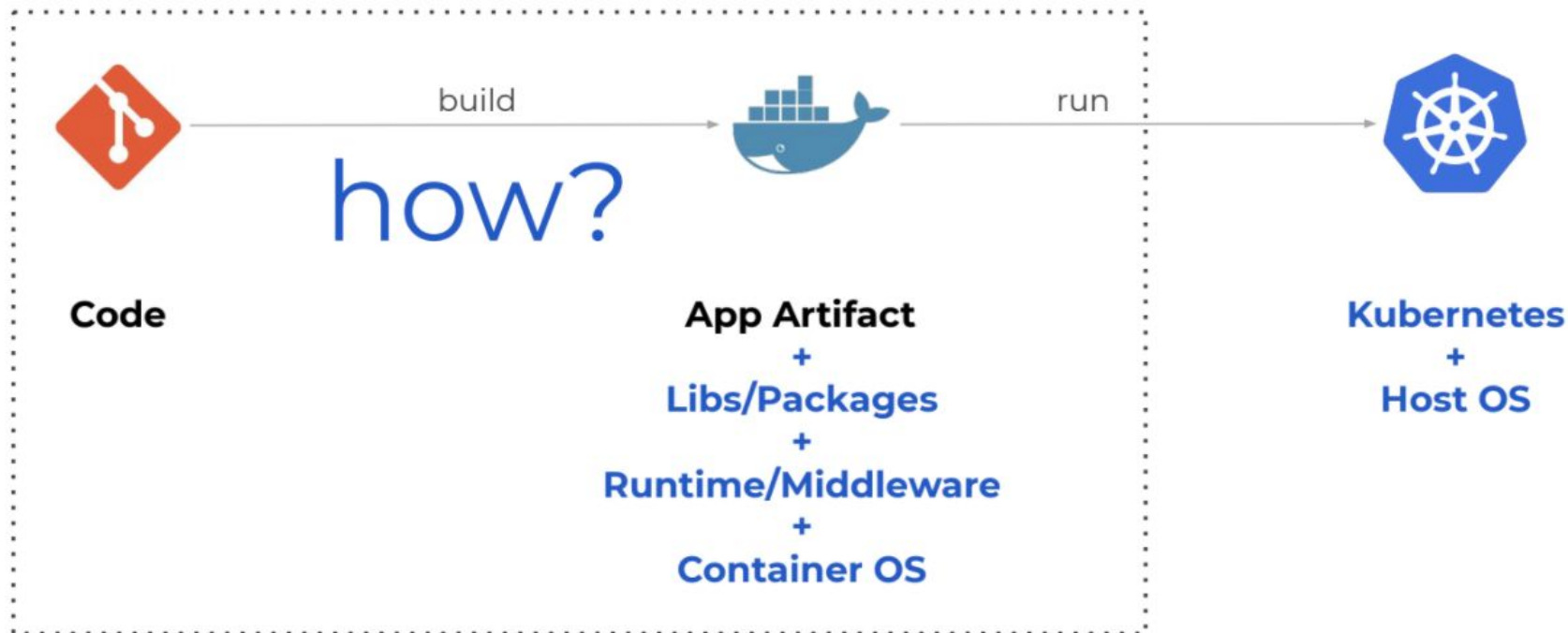
Коммерческие требования:

- Работает “из коробки”
- Функционал легко найти и начать использовать
- Настройка и усложнение по мере надобности



- Документация
- Мануалы (*Getting started / Hello World / 101*)
- Примеры
- Auto DevOps







Live Demo!!!



“Магия” Auto DevOps



```
153 # NOTE: These links point to the latest templates for development
154 # therefore the actual templates that were included for Auto DevOps
155 # could be different from the contents in the links.
156 # To view the actual templates, please replace `master` to the specific
157 # the Auto DevOps pipeline started running e.g. `v13.0.2-ee`.
158 include:
159   - template: Jobs/Build.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/build
160   - template: Jobs/Test.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/test
161   - template: Jobs/Code-Quality.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/code-quality
162   - template: Jobs/Deploy.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/deploy
163   - template: Jobs/Deploy/ECS.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/deploy-ecs
164   - template: Jobs/DAST-Default-Branch-Deploy.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/dast-default-branch-deploy
165   - template: Jobs/Browser-Performance-Testing.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/jobs/browser-performance-testing
166   - template: Security/DAST.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/security/dast
167   - template: Security/Container-Scanning.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/security/container-scanning
168   - template: Security/Dependency-Scanning.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/security/dependency-scanning
169   - template: Security/License-Scanning.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/security/license-scanning
170   - template: Security/SAST.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/security/sast
171   - template: Security/Secret-Detection.gitlab-ci.yml # https://gitlab.com/gitlab-org/gitlab/-/tree/master/security/secret-detection
```



Auto DevOps - Commit Code, GitLab Creates the Pipeline



CREATE



✓ Merge

✓ Build

VERIFY



✓ Code Quality

✓ Test

SECURE



✓ SAST

✓ Dependency

✓ Container

✓ License

✓ DAST

PACKAGE



✓ Container

✓ Registry

RELEASE



✓ Review App

✓ Deploy

CONFIGURE



✓ Infra Config

✓ Scale

MONITOR



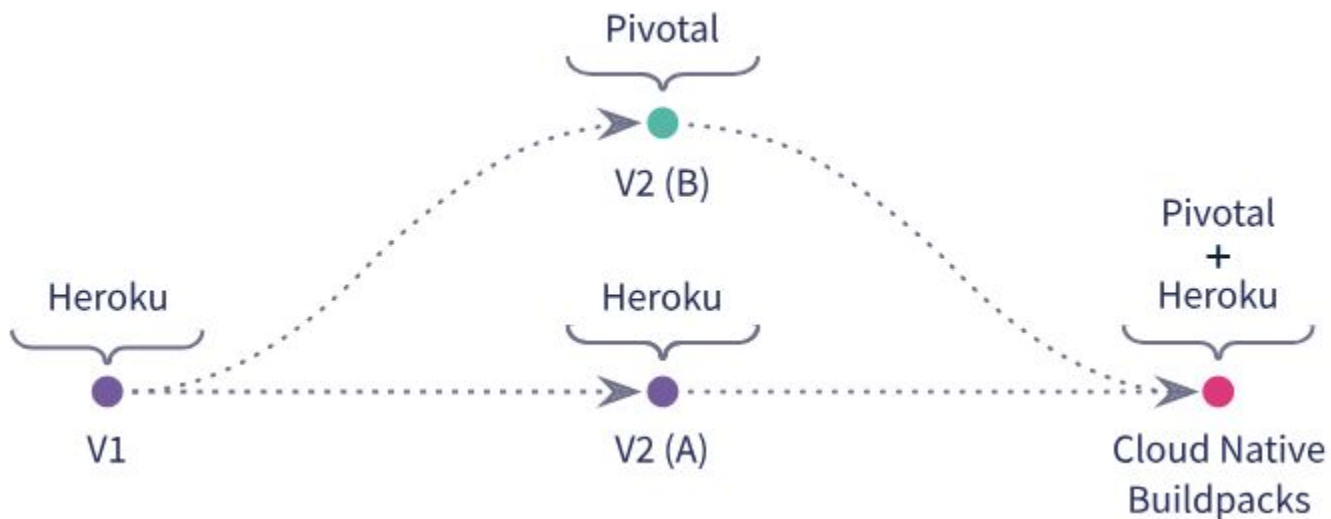
✓ Response

✓ System

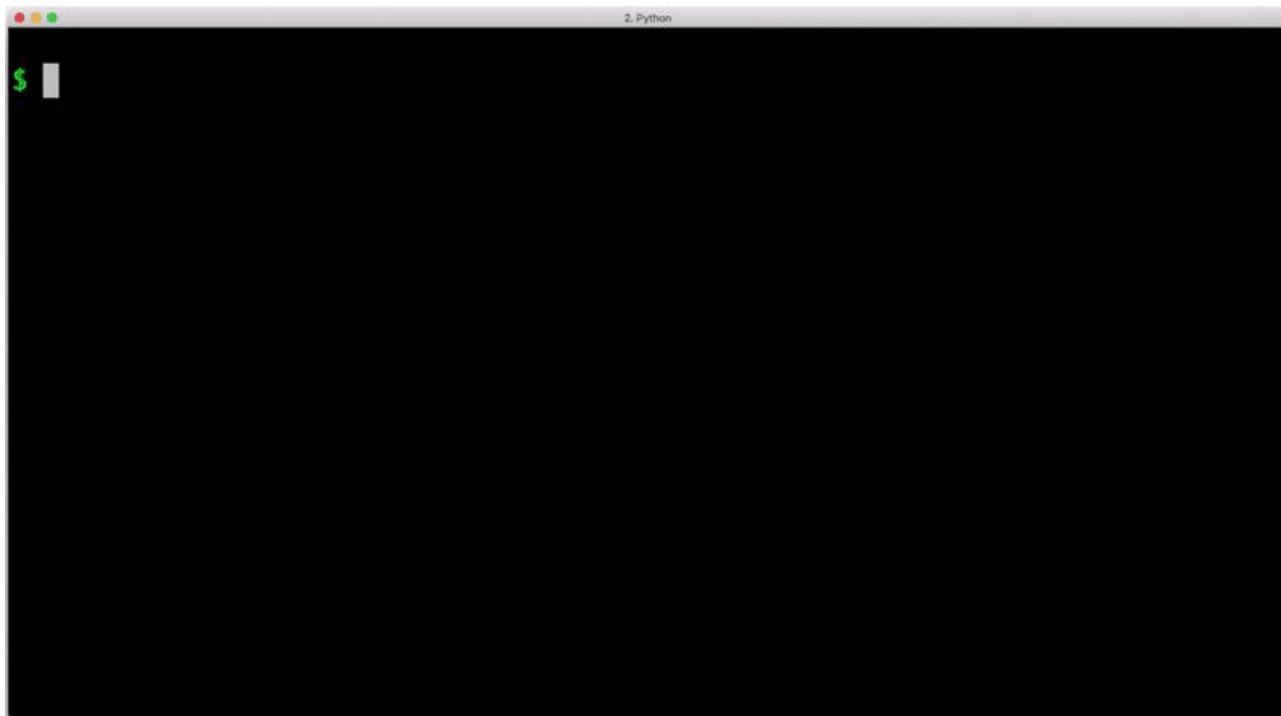
✓ Custom

✓ Perf Testing

Buildpacks: Herokuish / Cloud Native



Buildpacks in action





- No native Herokuish buildpack
 - But there is a Cloud Native Paketo buildpack
 - And there are native buildpacks for Ruby, Node.js, Java, Python, PHP, Go
 - And you can write your own!
- Cloud Native buildpacks don't include 'test'

```
AUTO_DEVOPS_BUILD_IMAGE_CNB_BUILDER = cloudfoundry/cnb:cflinuxfs3  
AUTO_DEVOPS_BUILD_IMAGE_CNB_ENABLED = 1  
TEST_DISABLED = 1
```





- Production vs. Development
- Сокращаем время для 'project setup'
- Сокращаем время для onboarding новых инженеров
- Используем в качестве примера
- Расширяем и кастомизируем Auto DevOps





- Включаем только нужные части
- Используем переменные окружения
 - Настройки проекта
 - Настройки группы проектов
 - git push options
 - Кастомный .gitlab-ci.yml
 - ...
- Переопределяем задачи
- Расширяем задачи (extends)
- Пишем свои билдпаки

```
include:  
  - template: Security/SAST.gitlab-ci.yml
```

```
include:  
  - template: Security/SAST.gitlab-ci.yml  
  
variables:  
  SAST_EXPERIMENTAL_FEATURES: "true"
```

```
include:  
  - template: Security/SAST.gitlab-ci.yml  
  
spotbugs-sast:  
  variables:  
    FAIL_NEVER: 1
```

ГОРОД IT



—

