

Feature Toggles

Karol Kreft

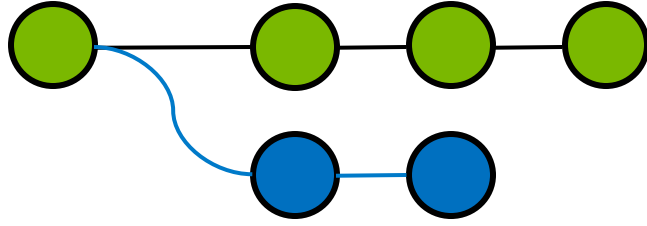
@karol_kreft 

PHPCon Poland 2019

15.11.2019, Szczyrk

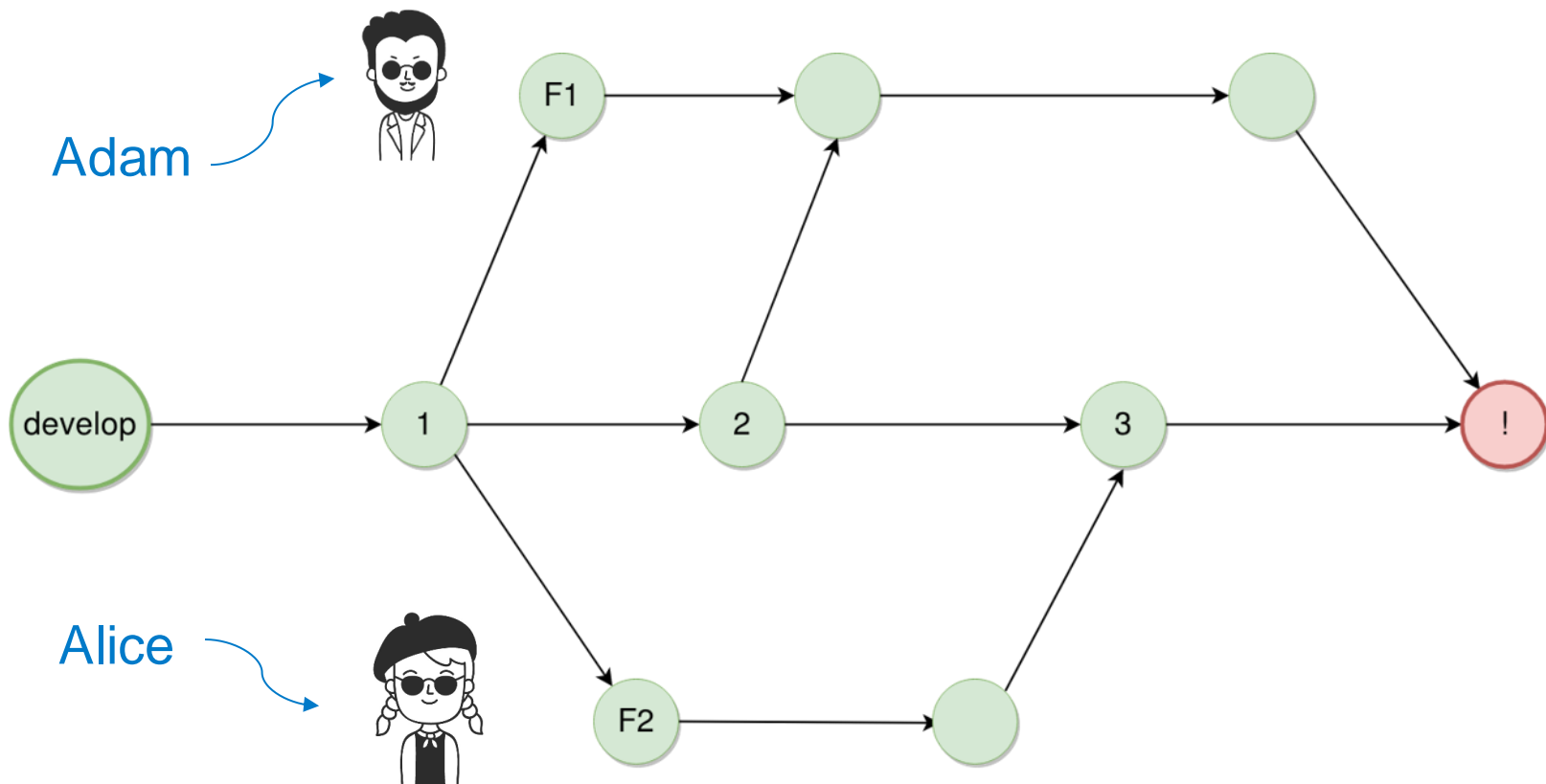


FeatureBranch



Advantages

- Each developer can work on their own feature and be isolated from changes going on elsewhere
- It allows to choose the features for release by cherry-picking



Disadvantages

- Big Scary Merge (a.k.a. Merge Bomb)
- Semantic conflicts

Disadvantages



Steve Smith
@SteveSmithCD

Follow



Hi @InfoQ, your #GitFlow article is flat out wrong. It is 100% incompatible with #ContinuousIntegration and #ContinuousDelivery.
[continuousdeliveryconsulting.com/blog/organisat ...](https://continuousdeliveryconsulting.com/blog/organisat...) / @danielbryantuk

Kenny Baas-Schwegler @kenny_baas

I won't go into the discussion about if gitflow is better or not then trunk based (it is not imo). But can people please stop calling using gitflow CI/CD. It is not continuous, you are building stuff in isolation.

11:59 PM - 9 Feb 2019

132 Retweets 283 Likes



<https://twitter.com/stevesmithcd/status/1094506119014334464>

Continuous Integration

”

*Continuous Integration is a software development practice where members of a team **integrate** their work **frequently**.*

Martin Fowler, Chief Scientist, ThoughtWorks

”

*Continuous Integration
doesn't get rid of bugs,
but it does make them
**dramatically easier to
find and remove.***

Martin Fowler, Chief Scientist, ThoughtWorks

Continuous Delivery

Continuous Delivery

”

*Essentially, it is the practice of releasing **every good build** to users.*

*Jez Humble, co-author of Continuous Delivery: Reliable Software Releases Through Build, Test and Deployment Automation
CTO DevOps Research and Assessment*

Delivery
!=
Deployment

Delivery and Deployment

	Unit tests	Platform tests	Deliver to staging	Acceptance tests	Deploy on production	Post deploy tests
Delivery	AUTO	AUTO	AUTO	AUTO	MANUAL	AUTO
Deployment	AUTO	AUTO	AUTO	AUTO	AUTO	AUTO



Amazon deploys new software to production every 11.6 seconds.



*Amazon deploys new software to
production every 11.6 seconds.
In 2011.*

But what about...

- infrastructure?
- release planning schedule?
- informing users about new features?

But what about...

- infrastructure?
- release planning schedule?
- informing users about new features?

If you lack the process and quality discipline,
you will break production really fast.

But what about...

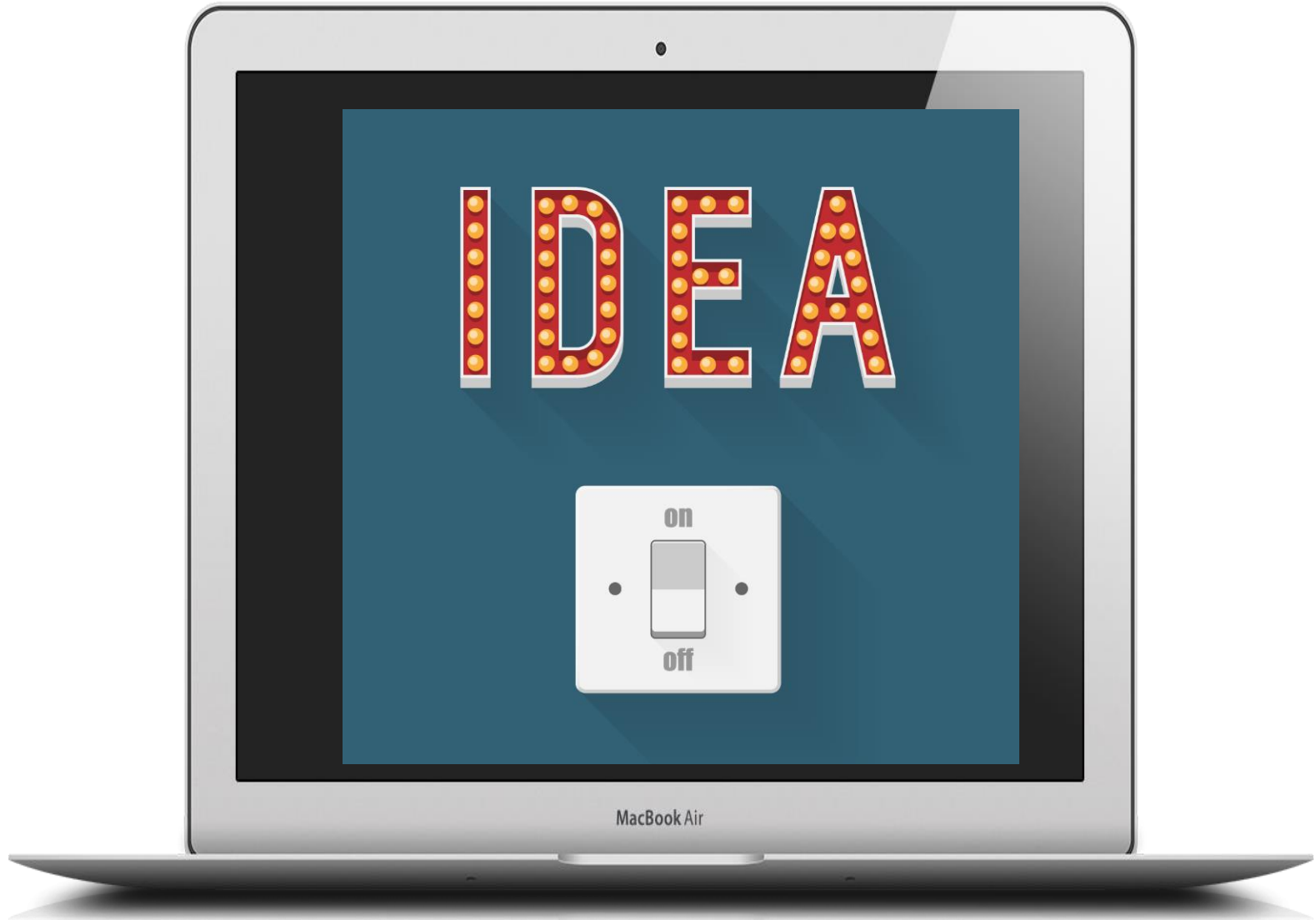
- infrastructure?
- release planning schedule?
- informing users about new features?

Continuous **Failure**

Feature Toggles

Feature ~~Toggles~~

Flags, Bits, Flippers



” **“Feature Toggling”**
*is a set of patterns
which can help a
team to deliver new
functionality to users
rapidly but safely.*

Pete Hodgson, independent consultant, ThoughtWorks

Try to find meaningful context



Experimental algorithm

- Search algorithm
- Related products / posts
- New Payment gateway
- Login by Social Media
- Import / Export data feature

Permissions

- Access Control List
- Premium features
- Premium accounts

Features related to the User

- New feature available only for some group of users.
 - Age
 - Gender
 - Most active users
 - Preferences

Geolocation

- Available only to specific region, country, city etc

A / B Tests

- More than one solution, which of them brings us more \$\$\$?
- Small but risk changes



I Am Developer
@iamdeveloper

Following



Which would you rather wear on a t-shirt?

Give me your answers now and get your obligated office Secret Santa present sorted.

51% No Deploy Fridays ✓

49% I Don't Deploy On Fridays

2,498 votes • 23 hours left

Canary release

- Only small subset of users will see the new feature
- If there are no critical bugs, extend a subset of users
- In case of some major errors occur, we are able to switch users back to the previous version of the software.

Before implementation

- How long each feature toggle will live in the system?
 - Unused feature toggle should be removed from the code.
 - Long-lived features may require a different configuration than the short-lived.

Implementation



```
public function search(phrase) {  
    // current implementation  
}
```


Implementation



```
private function newSearch(phrase) {  
  // new shiny implementation  
}  
  
private function oldSearch(phrase) {  
  // the old one  
}
```

Implementation



```
public function search(phrase) {  
    useNewAlgorithm = true;  
  
    if (useNewAlgorithm) {  
        return newSearch(phrase);  
    }  
  
    return oldSearch(phrase);  
}
```

Implementation



```
public function search(phrase) {  
    if (Feature::isEneabled("includePopularityInSearch") {  
        return newSearch(phrase);  
    }  
  
    return oldSearch(phrase);  
}
```

Implementation



```
[  
  'includePopularityInSearch' = true,  
  'displayRecomendationInSearch' = false  
  // ...  
]
```

Implementation



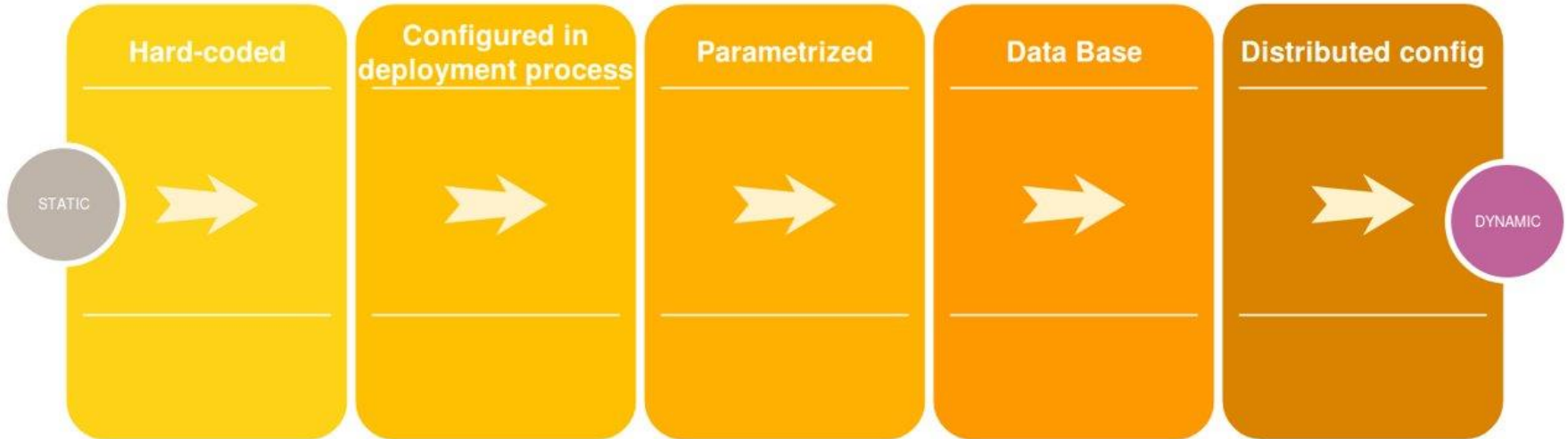
```
const INCLUDE_POPULARITY_IN_SEARCH = true;  
const DISPLAY_RECOMENDATION_IN_SEARCH = false;
```

Implementation



```
public function search(phrase) {  
    if (Feature::isEneabled("includePopularityInSearch", user) {  
        return newSearch(phrase);  
    }  
  
    return oldSearch(phrase);  
}
```

Configuration



Configuration



Apache
Zookeeper



LaunchDarkly

LaunchDarkly SDK



Dashboard

Your feature flags

<input type="text" value="Find a feature flag"/>			NEW +
• Alternate product page Added 2 years ago — Last requested 11 days ago	alternate.page	<input checked="" type="checkbox"/>	DELETE
• New dashboard Added 2 years ago	new.dashboard.enable	<input checked="" type="checkbox"/>	DELETE
• New recommendations engine Added a year ago	engine.enable	<input checked="" type="checkbox"/>	DELETE
• Swap image on hover Added a year ago — Last requested an hour ago	image.hover	<input type="checkbox"/>	DELETE
• Alternate sort order Added 10 months ago — Last requested an hour ago	sort.order	<input checked="" type="checkbox"/>	DELETE
○ Simplified checkout flows Added 6 days ago This tests out the new simplified checkout flow for our cart	simplified-checkout-flows	<input checked="" type="checkbox"/>	DELETE
○ New Multivariate Flag Added 3 days ago	new-multivariate-flag	<input type="checkbox"/>	DELETE

The Kill Switch

[Feature flags](#) / [Show new dashboard](#)

Show new dashboard

Added 3 years ago — Last requested 5 months ago

● new.dashboard

Targeting Experiments Variations History Settings

Targeting ON

SAVE CHANGES

Prerequisites ⓘ

+ Add prerequisites

Target individual users ⓘ

☒ true

2 users **BULK EDIT**

Add users...

☐ false

0 users **BULK EDIT**

Add users...

Target users who match these rules ⓘ

IF

SERVE ☒ true

ADD RULE

Costs

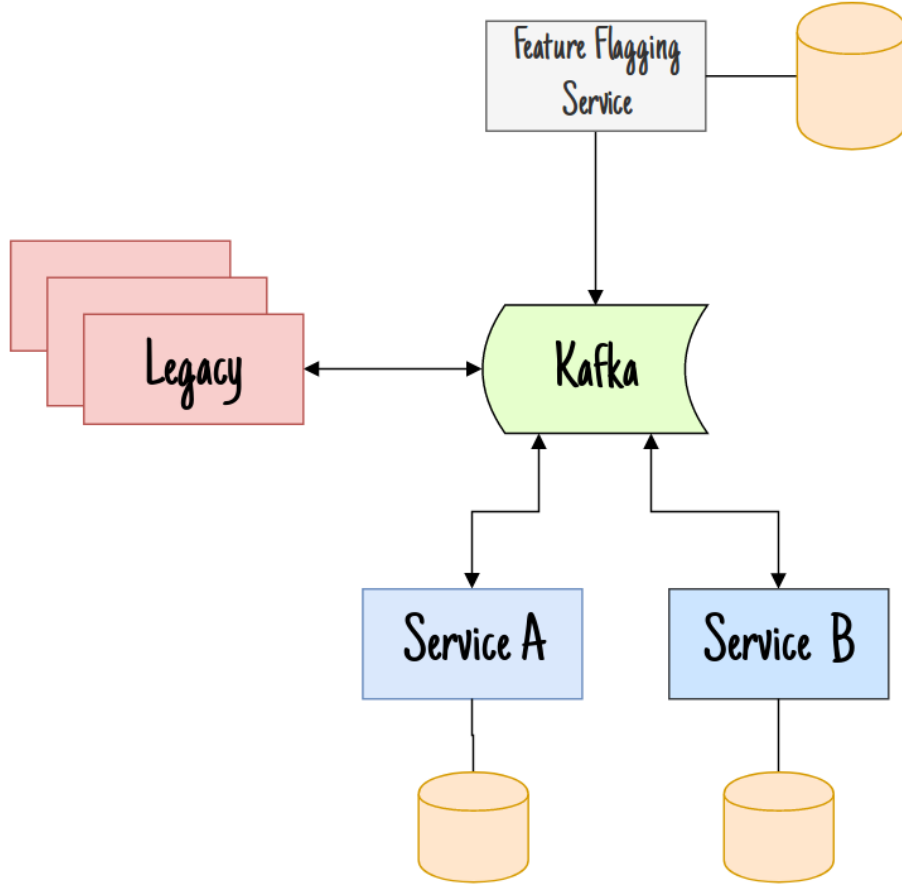
- Complexity, new abstraction to maintain
- Testing is harder, could require a method to administer current system state
- Technical debt, especially when lack of discussion and decision about removing unused feature toggles

Feature Toggles



Everywhere





Slim

a micro framework for PHP



PHP-DI 6

The dependency injection container for humans

PHP-DI in Slim

Slim is a micro-framework for web applications and APIs. Given the framework is compatible with PSR-11, *Slim can work with any container* out of the box. Using PHP-DI with Slim is then easy.

However **the PHP-DI bridge provides several additional features:**

- controllers as services, allowing dependency injection in controllers
- intelligent parameter injections in controller

PSR-7: HTTP message interfaces

PSR-11: Container

PSR-15: HTTP Server Request Handlers

PSR-17: HTTP Factories



```
(function () {  
    require_once __DIR__ . '/../vendor/autoload.php';  
    $container = (new Container\Factory())->create();  
    $app = Bridge::create($container);  
    $responseFactory = $container->get(ResponseFactoryInterface::class);  
  
    $app->add(new FeatureFlags\IsEnabled($responseFactory))  
        ->add(new Http\AccessControl($responseFactory))  
        ->add(new Http\ContentSecurityPolicy());  
  
    (require __DIR__ . '/../config/routes/routes.php')($app);  
    $app->run();  
})();
```



```
public function process(  
    ServerRequestInterface $request,  
    RequestHandlerInterface $handler  
): ResponseInterface {  
    if ($request->getHeader('Feature')) {  
        return $handler->handle($request);  
    }  
  
    return $this->responseFactory->createResponse(404);  
}
```




```
public function process(  
    ServerRequestInterface $request,  
    RequestHandlerInterface $handler  
): ResponseInterface {  
    $feature = $request->getHeader('Feature');  
    if($this->repository->isEnabled($feature)) {  
        return $handler->handle($request);  
    }  
  
    return $this->responseFactory->createResponse(404);  
}
```

Doubts

- _Third-party solutions are feature centric since we would like them to be more account centric instead
- _What is the default value for feature flags? Is it configurable per flag or global? Can it be overridden? How and on what level?
- _Who can change the feature flag config? Are we going to track those changes? Do we need a process to grant this privileges?

Feature Toggles

Karol Kreft

@karol_kreft 

PHPCon Poland 2019

15.11.2019, Szczyrk



Sources

- [FeatureBranch, Martin Fowler](#)
- [Organisation Antipattern: Integration Feature Branching](#)
- [Continuous Integration, Martin Fowler](#)
- [Steve Smith on twitter about GitFlow and CI](#)
- [Feature Toggles \(aka Feature Flags\), Pete Hodgson](#)
- [A journey into Feature Toggles, Pete Hodgson](#)
- [Velocity 2011: Jon Jenkins, Velocity Culture](#)
- [The Story of Apollo - Amazon's Deployment Engine, Werner Vogles](#)
- [Featureflags.io](#)
- [LaunchDarkly - Feature Toggle Management For Continuous Delivery](#)
- [Managing Feature Flags e-book, Adil Aijaz and Patricio Echagüe](#)

Photos

- [Photo by Diego PH on Unsplash](#)
- [Photo by Aaron Burden on Unsplash](#)
- [Business vector created by freepik / dooder - www.freepik.com](#)
- [Dinos, Volcanos & Jeff Goldblum: The “Jurassic World 2” Trailer Is Here](#)
- https://www.freepik.com/free-vector/illustration-lan-network_2606500.htm