

Testing in Production with Azure and Feature Flags

Edin Kapić- @ekapic

¡Gracias!

Patrocinadores Locales



Colabora



Edin Kapić

- SharePoint Master of Craft @ [isolutions Barcelona](#)
- Cofounder and president of [SUG.CAT](#)
- MVP for Office Apps and Services



@ekapic



SharePoint Saturday
Barcelona



sug.cat



++

georgelynych

1y

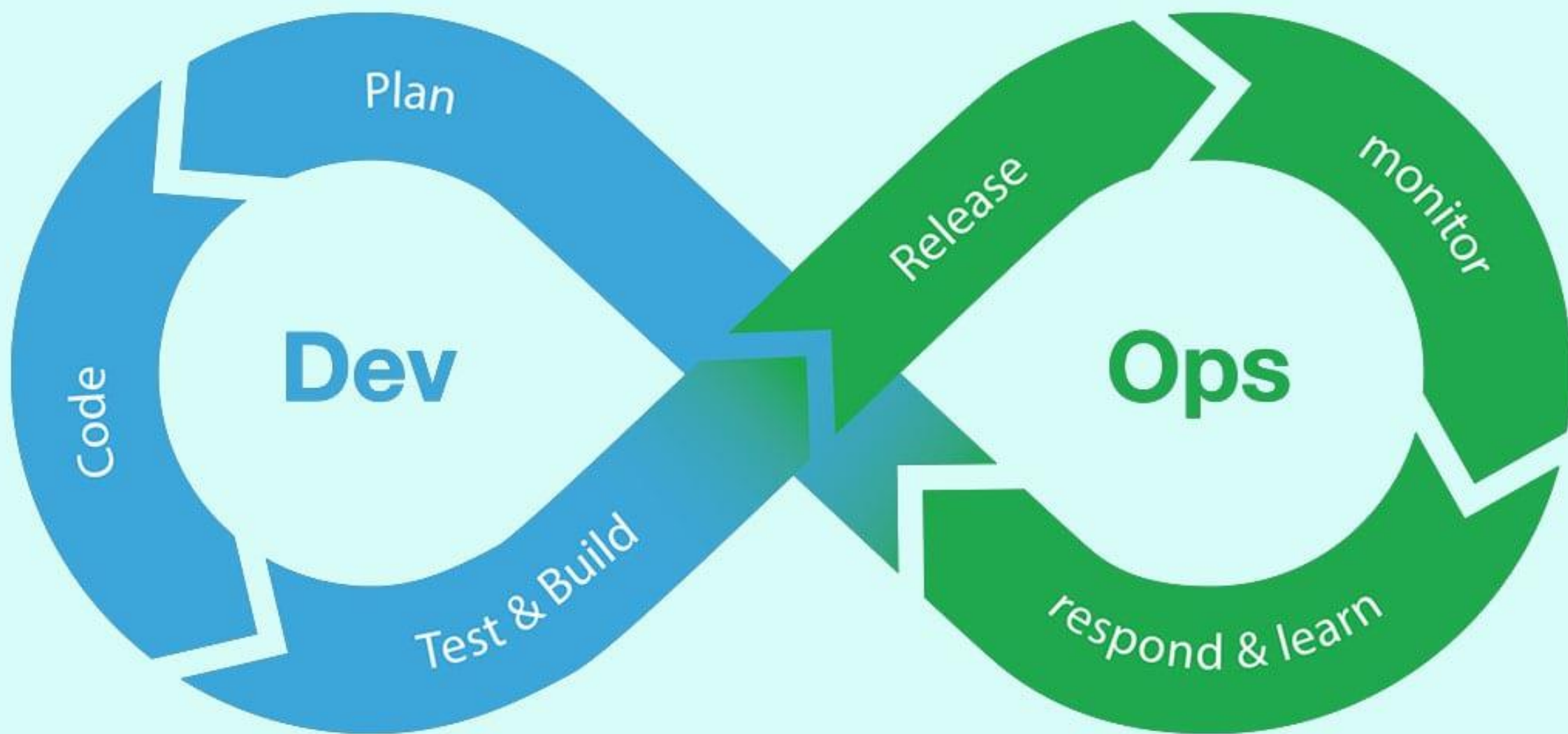
10

--

I had rough week last week. Accidentally deployed dev code to production, soon found out there was no production version of code to do a redeploy. Deployed another app to production, it was working fine then another Dev changed a data type in the database from bool to nullable Boolean which broke some Linq queries. Looked like I deployed crap code. 3rd week with company.

rant

Favorite



Feature Flags

Feature Flag

Decouple feature rollout
from code deployment

New code is under a flag
check

AKA “feature toggle”

<https://martinfowler.com/articles/feature-toggles.html>





Home > Organization profile



edinkapicis

Name edinkapicis

Address ,

Phone 4250000000

Technical contact ekapic@gmail.com

Preferred language English

Release preferences

Choose how your organization gets new features and service updates from Office 365.

Release preferences

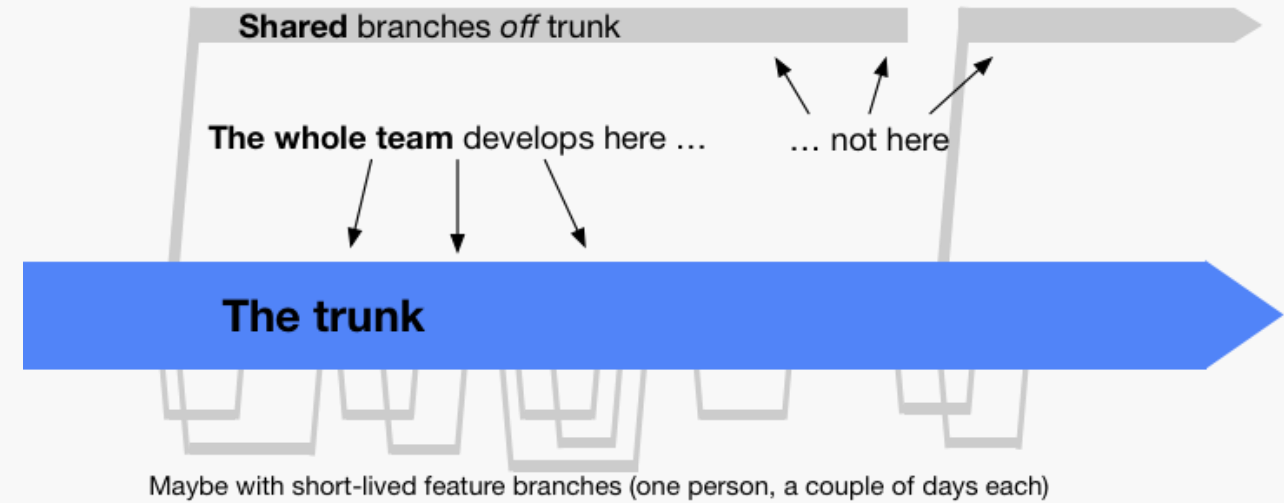
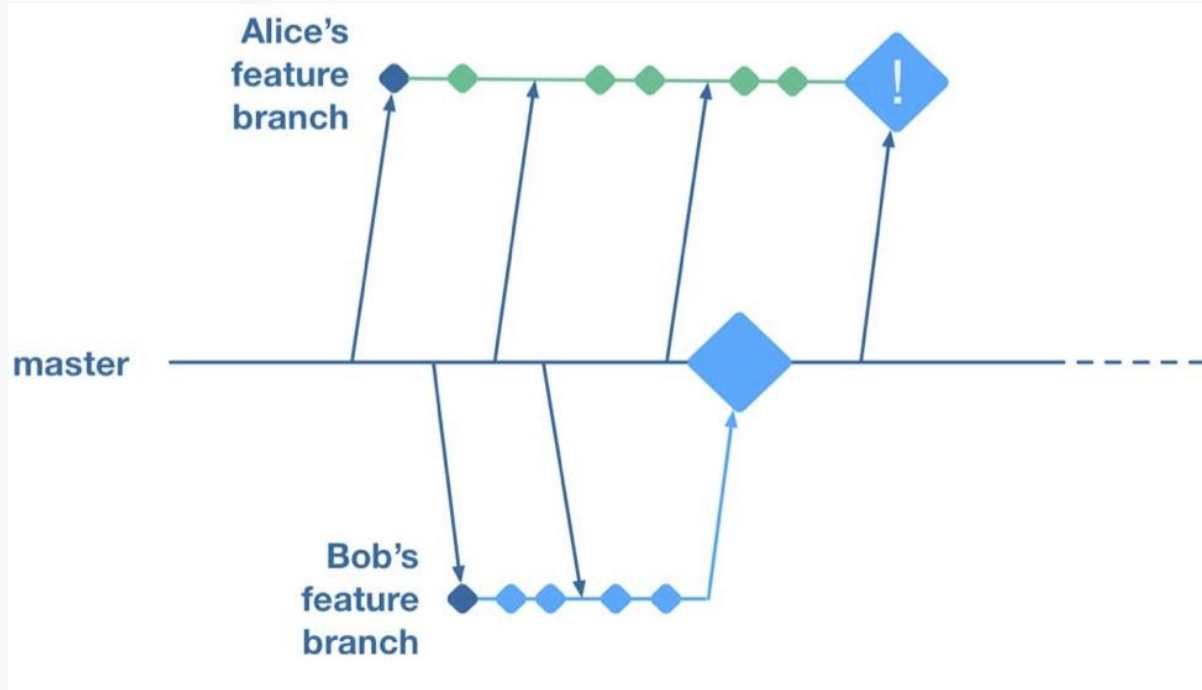
Release track

- ☒ Standard release
Get updates when we release them broadly.
- ☐ Targeted release for everyone
Get updates early for your entire organization
- ☐ Targeted release for selected users
Pick people to preview updates so that you can prepare your organization

Next

Cancel

Feature branches vs Feature flags



Starting situation

```
public byte[] MakeThumbnails(Image img) {  
    ... OLD CODE  
}
```

Simple "feature flag"

```
public byte[] MakeThumbnails(Image img) {  
    bool useNewCode = true; // CHANGE IT IF NEEDED  
    if(useNewCode)  
    {  
        ... NEW CODE  
    }  
    else {  
        ... OLD CODE  
    }  
}
```

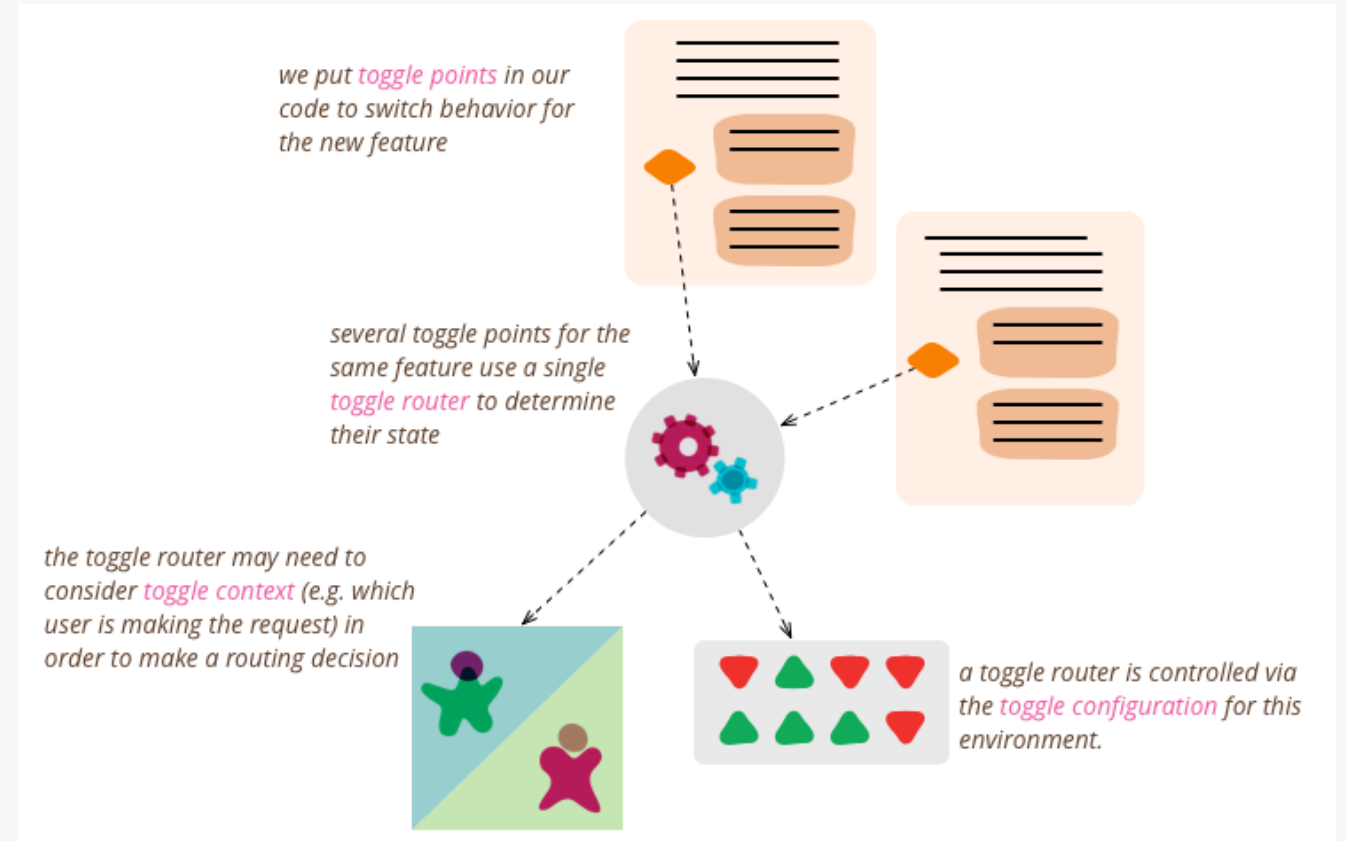
Let's make it changeable

```
public byte[] MakeThumbnails(Image img) {  
    bool useNewCode = IsFeatureEnabled("new-  
thumbnail-code");  
    ...  
}  
  
public bool IsFeatureEnabled(string key) {  
    // return AppSettings key value if exists  
}  
  
<add key="new-thumbnail-code" value="true"/>
```

Dynamic feature flags

Evaluate flags in real time

Can be evaluated for each user, IP, URL...



Canary testing



Dogfooding



A/B testing



Kill switches



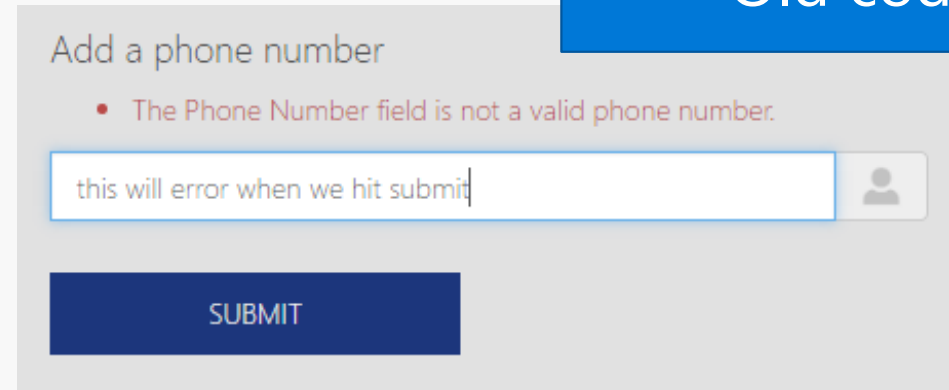
Manual Feature Flags Management

The scenario

Rollout better **phone number UX** in production for selected users of Parts Unlimited site

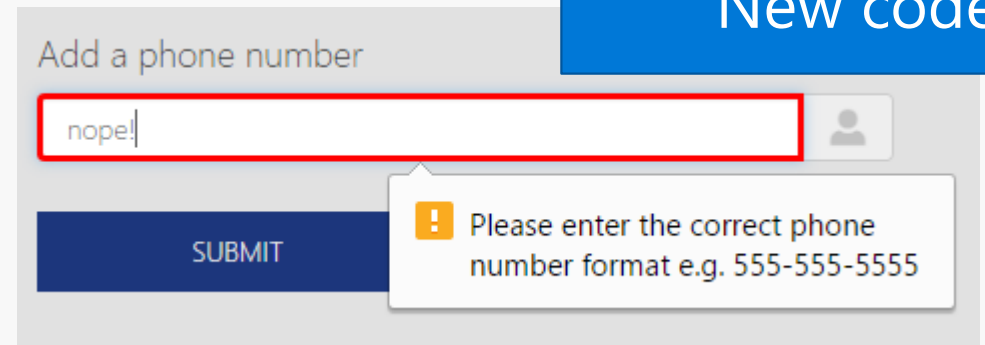
Source code for the e-commerce site
<https://github.com/Microsoft/PartsUnlimited>

Old code



A screenshot of a web form titled "Add a phone number". Below the title is a red error message: "The Phone Number field is not a valid phone number." The input field contains the text "this will error when we hit submit". To the right of the input field is a small user icon. Below the input field is a dark blue "SUBMIT" button.

New code



A screenshot of a web form titled "Add a phone number". The input field contains the text "nope!". To the right of the input field is a small user icon. Below the input field is a dark blue "SUBMIT" button. A yellow warning icon is visible next to the submit button, and a tooltip message says: "Please enter the correct phone number format e.g. 555-555-5555".

The Feature representation

```
public class Feature
{
    public string Key { get; set; }
    public string Description { get; set; }
    public bool Active { get; set; }
}
```

Abstracting away the feature management

```
public interface IFeatureManager
{
    Task<bool> GetStatusByKey(string key, ClaimsPrincipal
userClaim);
    Task ChangeFeatureToggles(string[] selectedItems,
ClaimsPrincipal userClaim);
    Task<IEnumerable<Feature>> GetUserFeatures(ClaimsPrincipa
userClaim);
}
```

In the Startup.cs inject the feature manager

```
services.AddScoped<IFeatureManager,  
    FeatureManager>();
```


In the controller, check for the feature

```
public async Task<IActionResult> AddPhoneNumber()  
{  
    var statusByKey = await  
_featureManager.GetStatusByKey(FeatureManager.Phone  
eNumberKey, HttpContext.User);  
    return View(new AddPhoneNumberViewModel {  
IsPhoneFeatureActive = statusByKey});  
  
}
```

In the view, provide both implementations

```
@{  
    if (Model.IsPhoneFeatureActive)  
    {  
        @Html.TextBoxFor(m => m.Number, new { @class =  
"form-control", placeholder = "555-555-5555", type =  
"tel", pattern = "\\d{3}[\\-]\\d{3}[\\-]\\d{4}", id =  
"phone" })  
    }  
    else  
    {  
        @Html.TextBoxFor(m => m.Number, new { @class =  
"form-control", placeholder = "Phone Number" })  
    }  
}
```

New code

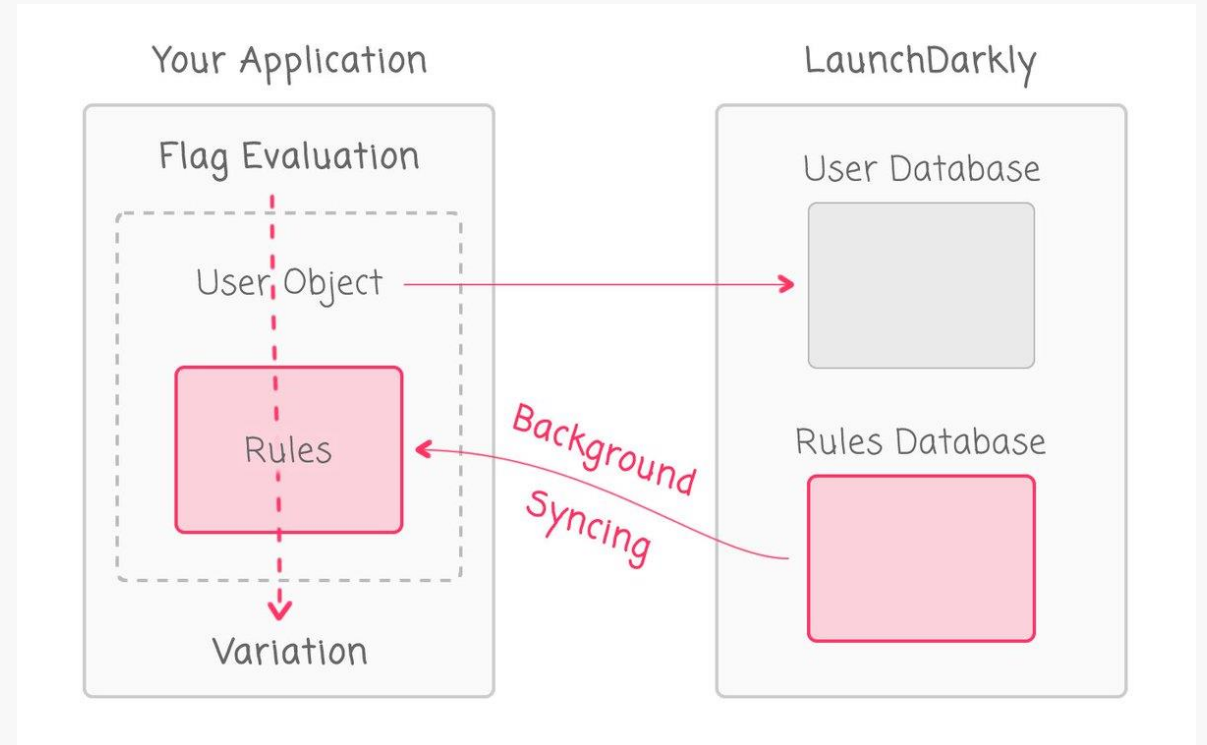
Old code

Automated Feature Flags Management with LaunchDarkly

LaunchDarkly

SaaS solution for feature flagging

Integrated with Azure DevOps (ex-VSTS)



LaunchDarkly.com

← → ↺

https://app.launchdarkly.com/default/test/features

☆ ⚙️ 📧 👤 ⋮

Default Project Test ▾

Feature flags

Users

Segments

Goals

Debugger

Audit log

Quickstart

Documentation

Integrations

Dashboard

Your feature flags

COMPARENEW +

Find a flag by name, key or description

Status: Any ▾Type: Any ▾Tags: Any ▾Sort by: Oldest ▾

Enhanced Phone Number

phone-number

OFF

DELETE

Added 5 days ago — Last requested 28 minutes ago

LD integration in code

Install-Package LaunchDarkly.Client

```
// Singleton client for efficiency
```


```
LdClient ldClient = new LdClient("YOUR_SDK_KEY");
```

Your projects

Default Project default

Find an environment

<div><div></div>Production</div> <div>production</div>	0 minute TTL	SDK key	sdk-ad99ba40-6e7b-4639-9033-fac10cd8b09
		Mobile key	mob-44d8b468-e288-40d1-8233-77c09d80a50
		Client-side ID	5cb8ab3a0073001e2c7f62
<div><div></div>Test</div> <div>test</div>	0 minute TTL	SDK key	sdk-1e8252ca-4130-4035-a411-46b116b7294
		Mobile key	mob-463a9cd9-a013-4435-8411-1e911241114
		Client-side ID	5cb8ab3a0073001e2c7f62



Querying LD for feature availability

```
User user = User.WithKey(username);
bool showFeature =
ldClient.BoolVariation("feature.key", user, false);
if (showFeature) {
    // application code to show the feature
}
else {
    // the code to run if the feature is off
}
ldClient.Dispose();
```


Defining the "user" in LD

```
LDUser user = User.WithKey("aa0ceb")  
    .AndFirstName("Ernestina")  
    .AndLastName("Evans")  
    .AndEmail("ernestina@example.com")  
    .AndCustomAttribute("groups", new  
List<String>(){ "Google", "Microsoft"})
```

A/B testing

Define “Goals” and
associate features
with them

Track the goal
completion for
each variation

The screenshot shows the 'Goals' management interface in Optimizely. At the top, there's a 'Goals' header. Below it, there are tabs for 'Click', 'Pageview', and 'Custom'. To the right, there's a 'Filter:' input field. A green callout box with the text 'Create or Import A/B Testing Goals' points to a blue 'Create' button and a grey 'Import Optimizely goals' button. Below the tabs, there's a table with columns: 'Kind', 'Name', '# of features', 'Description', and 'Actions'. The first row in the table is for a 'Sign Up Rate' goal, which is a 'Custom' goal, has 4 features, and a description 'measures the sign up rate'. There is a red 'Delete' button in the 'Actions' column for this goal. Below this table, there's a section titled 'Imported goals' with another table. This table has columns: 'Kind', 'Name', '# of features', 'Description', and 'Actions'. It lists four goals: 'Sandbox - Click on product' (2 features), 'Sandbox - Add to Cart' (0 features), 'Sandbox - View product' (2 features), and 'Sandbox - Engagement' (3 features). A green callout box with the text 'Manage Goals' points to the 'Sandbox - View product' row.

Goals

Create or Import A/B Testing Goals

Create Import Optimizely goals

Filter:

Kind	Name	# of features	Description	Actions
Custom	Sign Up Rate	4	measures the sign up rate	Delete

Imported goals

Kind	Name	# of features	Description	Actions
Click	Sandbox - Click on product	2		
Click	Sandbox - Add to Cart	0		
Pageview	Sandbox - View product	2		
Click	Sandbox - Engagement	3		

Manage Goals

Your experiments

MANAGE GOALS

A/B Test Results
For a Feature Flag

My Project - Engagement (No winner yet)

Variation	Conversions	Conversion rate	Change	Confidence
Flag on	14 / 256	6.2% (± 6.6%)	—	75.2%
Flag off	6 / 254	3.3% (± 5.9%)	—	—

Sandbox - Click on product (No winner yet)

Variation	Conversions	Conversion rate	Change	Confidence
Flag on	4 / 9	44.4% (± 27.3%)	—	74.4%
Flag off	3 / 10	30.0% (± 23.9%)	—	—

Integration with source code repository

show-widgets

● show-widgets

Maintained by  [Arnold Trakhtenberg](#)

Added 3 months ago

Targeting Experiments Variations History **Code references** Settings


Code references

Find references to this flag in your code. To set up a new repository for flag reference tracking, refer to our [documentation](#).

Repository: support-service ▾

Branch: master ▾

COLLAPSE ALL

 support-service 4fddac3

^ app/routes.py

[View in source](#)

23

```
24     showWidgets = ldclient.get().variation('show-widgets', current_user.get_ld_user(), False)
```

25

^ app/templates/beta/index.html

[View in source](#)

```
719     var renderButton = function() {
```

```
720         var showFeature = ldclient.variation("show-widgets", false);
```

```
721     }
```

^ app/templates/partials/footer.html

[View in source](#)

```
24     var renderButton = function() {
```

```
25         var showFeature = ldclient.variation("show-widgets", false);
```

```
26     }
```

Integration with Azure DevOps release

Details

LaunchDarkly

Associate a feature flag with this work item.

Select account

LaunchDarkly

Select a project and environment

Spree Commerce / Production

Select a feature flag

<input type="radio"/>	Swap image on hover image.hover	False	100% rollout	True
<input type="radio"/>	Alternate sort order sort.order	False	0% rollout	True
<input checked="" type="radio"/>	Show new dashboard new.dashboard	False	55% rollout	True
<input type="radio"/>	New recommendations engine new.recommendations.engine	False	50% rollout	True
<input type="radio"/>	Alternate Product Page alternate.page	False	0% rollout	True


All pipelines > LaunchDarkly_CD


Pipeline Tasks Variables Retention Options History


Stage 1
Deployment process

Stage name
Stage 1

Agent job
Run on agent

 Azure Resource Group Deployment
Some settings need attention

 LaunchDarkly Rollout
LaunchDarkly Rollout

 Azure App Service Deploy
Some settings need attention

Controlled Rollout: 10%

Account

LaunchDarkly Account

Add | Manage

Rollout

10

Flag state

On

Caveats

Feature flags are technical debt

They should be

- limited in number
- eventually removed

Testing with feature flags is more complex



Feature flags can be dangerous

Knight Capital Group lost
\$440M in 45 minutes due to a
repurposed flag from 8 years
ago



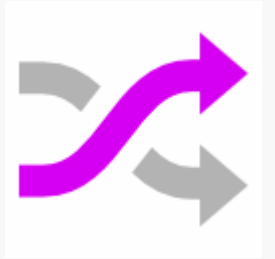
<https://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/>

Summary

Alternative feature flag implementations

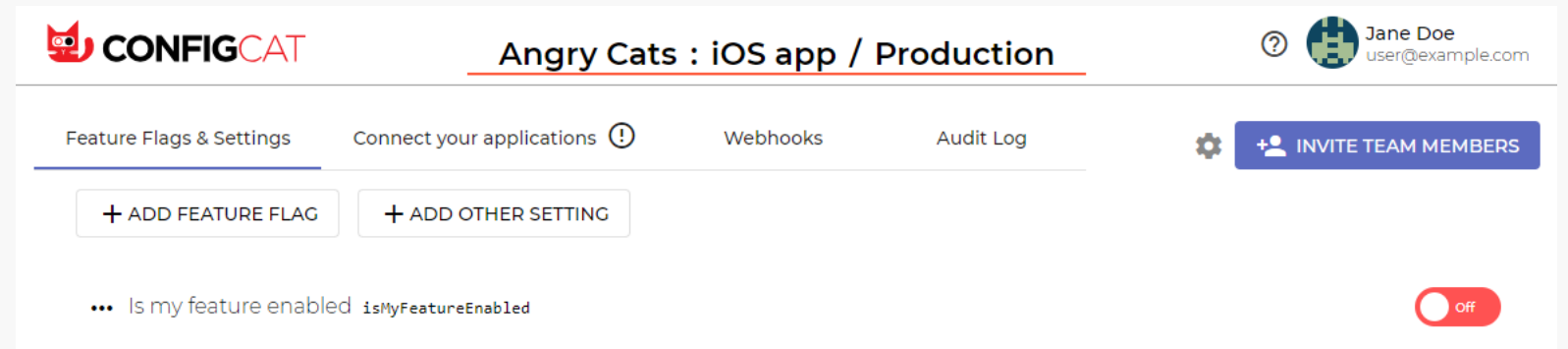
.NET FeatureToggle (open-source)

<https://github.com/jason-roberts/FeatureToggle>



ConfigCat (SaaS)

<https://configcat.com>



Call to Action

- Play with DevOps HOL to learn feature flags
<https://microsoft.github.io/PartsUnlimited/advanced/FeatureFlagWeb.html>
- Then graduate to MS DevOps Labs:
<https://azuredevopslabs.com/labs/vstsextend/launchdarkly/>

¡Gracias!

Patrocinadores Locales



Colabora



2019

Global **Azure**
BOOTCAMP

