Implementation of a feature flag for Company X.

Usecase

Company X would like to have the ability to set a feature flag checked by their website to allow customers to automatically get assigned with a `promoDiscount` and `loyalty` level if the following rule are matched:

```
The email address the customer is registering with ends with "gmail.com"
        AND
The Age of the customer is > 19 yeas old
        AND
The Customer is a "male"
```

When such rule is matched, Company X would like to receive a JSON document matching the following format:

If rule is matched:

```
{
  "loyalty": "gold",
  "promoDiscount": 50
}
```

else:

```
{
  "loyalty": "bronze",
  "promoDiscount": 10
}
```

Such approach allows the Company X to set defaults (here *loyalty* and *promoDiscount*) within the LaunchDarkly platform. Such defaults can be changed, tracked by LaunchDarkly revisions feature and approved via an approval workflow.

Company X also requested a sample Python client script which can leverage the Python SDK and retrieve the evaluation of the above feature flag.

Assumptions:

Email address, Age and gender of the user being evaluated is known by Company X and will be passed in to the LaunchDarkly SDK with the provided script for evaluation.

Approach

This document outlines how to deliver the above usecase with the help of the feature flag feature provided by LaunchDarkly.

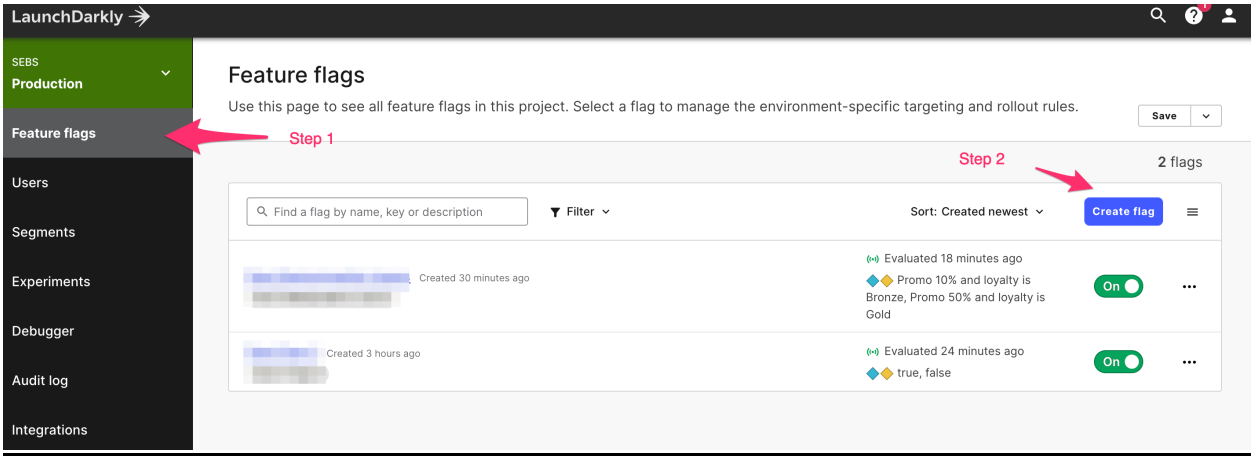
The following steps will be required:

Step 1) Creation of a Feature flag in the LaunchDarkly and configure of the 2 json variations.
Step 2) Retrieve the SDK key from the LaunchDarkly platform.
Step 3) Execute the sample Python script to validate this solution.
Step 4) Finalise the creation of the rule.
Step 5) Final testing.

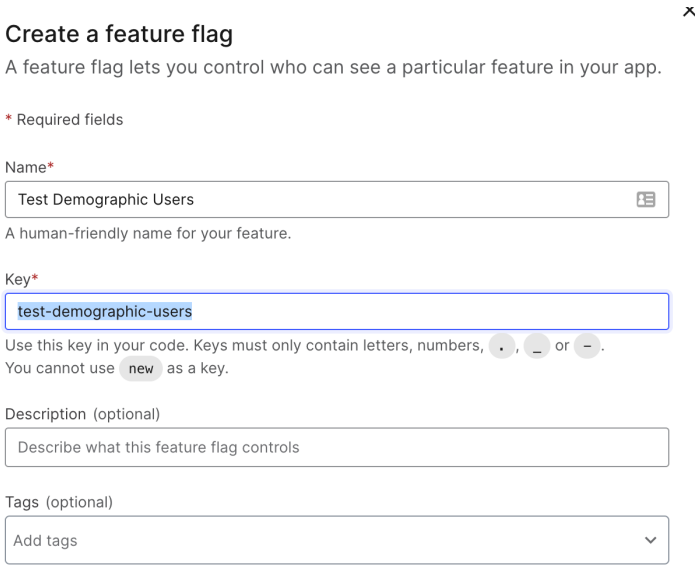
Note: Out of the box, LaunchDarkly does not provide the custom attributes necessary for this usecase – please see https://docs.launchdarkly.com/home/users/custom-attributes - These will therefore need to be created first by the script we are providing for this usecase hence the reason for creating the rule in step 4 only.

Step 1) Creation of a Feature flag in the LaunchDarkly and configure json variations.

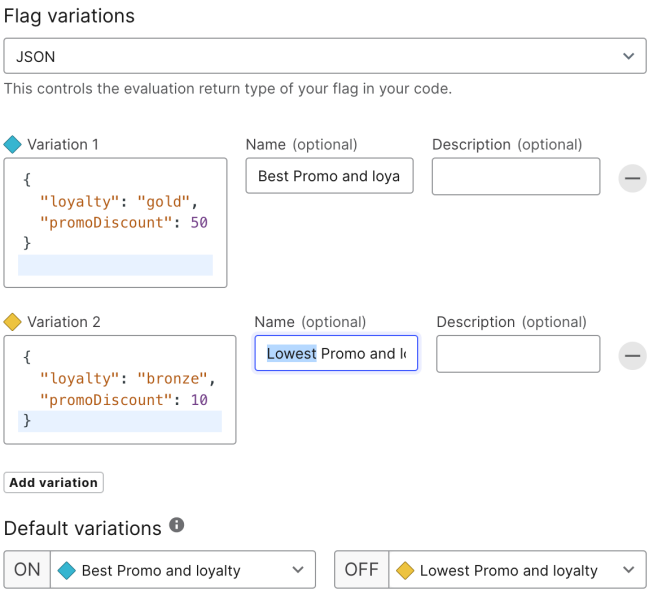
- Log in the the Launch Darkly UI
- Select *Feature flags* as instructed below (Step 1) and select *Create flag* (Step 2)



- Provide a name for the Feature Flad (eg "`Test Demographic Users`") – please note the name of the key which will be used later in the script provided: *`test-demographic-users`*

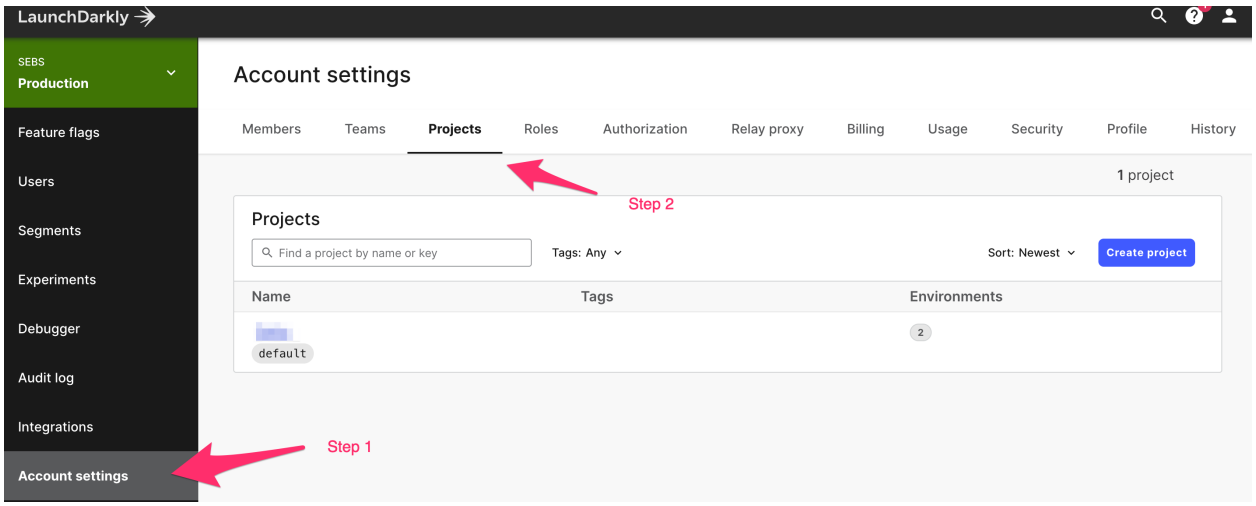


- Create the JSON variations required

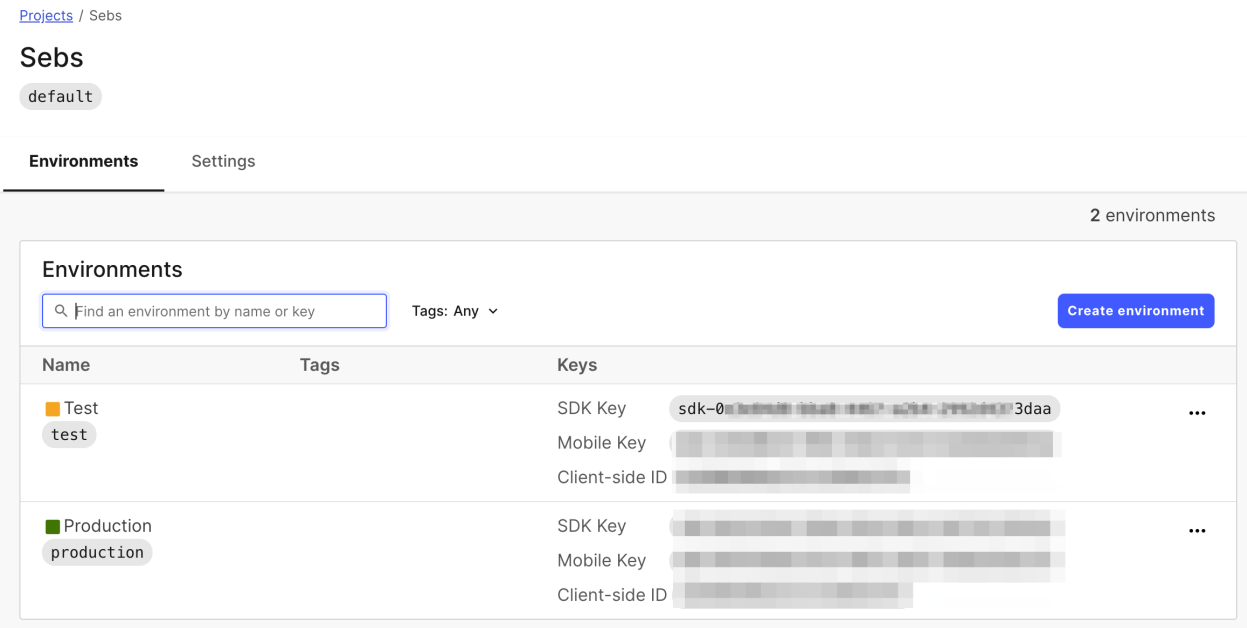


Step 2) Retrieve the SDK key from the LaunchDarkly platform.

- Log into the Launch Darkly UI
- Select the Account Settings and Projects as instructed below:



The SDK keys will appear:



Please select and Copy to Clipboard the SDK key you will be using in the script provided.

Step 3) A sample Python script to validate this solution (this step will be used to create the necessary User Custom Attributes).

a) Get the project from github here: https://github.com/smalbois-workspace/LD-repo.git

b) Go to the root of this repository on your system and make sure you have a python environment setup (version X minimum). Now run the following command:

```
pip3 install -r requirements.txt
```

c) Create a `.env` file in the root directory

```
SDK_KEY_PRODUCTION=<SDK KEY PRODUCTION>
SDK_KEY_TEST=<SDK KEY TEST>
ENVIRONMENT=production
```

d) Update the `.env` file with your SDK key (since `ENVIRONMNENT` set to production please copy and paste the SDK_KEY from Step 2 to `SDK_KEY_PRODUCTION`)

e) Make sure the name of the feature flag key set in the script file FeatureFlags.py for variable `feature_flag_key` is set to the feature flag key you created previously (in this case `test-demographic-users`)

f) Execute the script and verify that you are seeing the following:

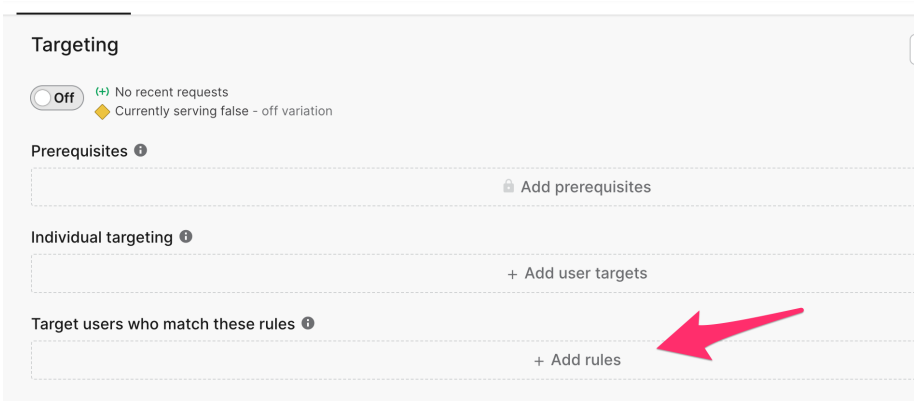
```
python3 FeatureFlags.py
*** SDK successfully initialized!

*** Feature flag 'test-demographic-users' is returning
{'loyalty': 'bronze', 'promoDiscount': 10} for this user
```

```
*** Promo Discount to apply is 10 percent and the loyalty is
bronze
```

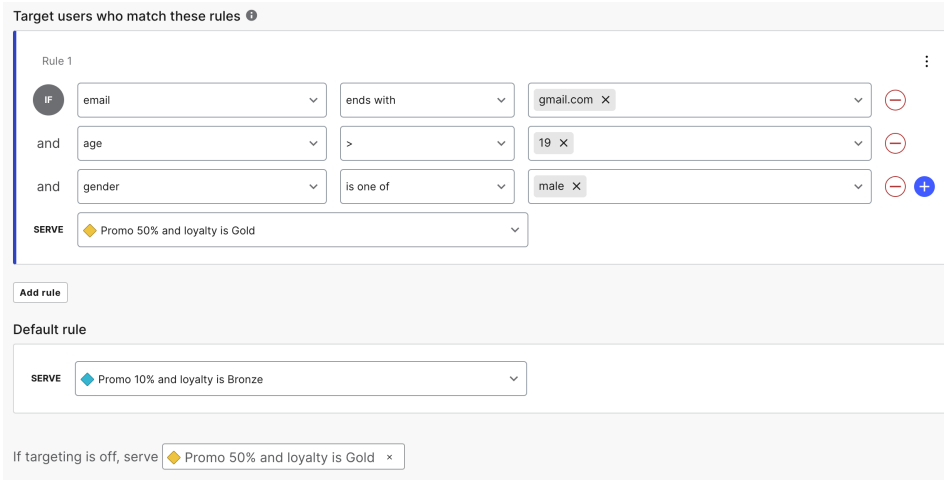
At this stage, the custom attributes will have been created on the user and the rule can now be created.

Step 4) Finalise the creation of the rule.

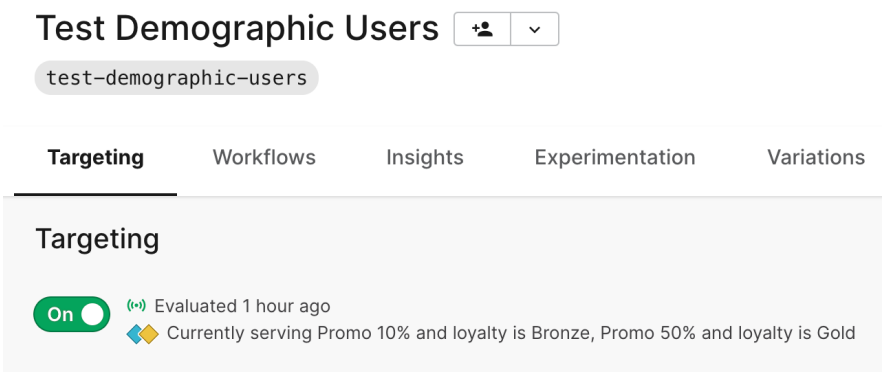
Edit the Feature Flag previously created and add the rule (Add rules)



and provide the following information:



Important: make sure Targetting is turned ON



And save … Congratulations! Your feature flag is now configured.

Step 5) Final testing

Please execute the script once again to validate that the rule works. You can validate this by changing the email address, gender or age for the user in the FeatureFlags.py file.

When the user is not matching the rule:

```
python3 FeatureFlags.py
*** SDK successfully initialized!

*** Feature flag 'test-demographic-users' is returning
{'loyalty': 'bronze', 'promoDiscount': 10} for this user

*** Promo Discount to apply is 10 percent and the loyalty is
bronze
```

When the user is matching the rule:

```
python3 FeatureFlags.py
*** SDK successfully initialized!

*** Feature flag 'test-demographic-users' is returning
{'promoDiscount': 50, 'loyalty': 'gold'} for this user

*** Promo Discount to apply is 50 percent and the loyalty is gold
```

Thank you.