



DRUPALCAMP GHENT

23 & 24 NOVEMBER 2018

Ephemeral Environments

Tom Robert - I am Root

AUSY
a Randstad company

LEVEL 27

Who am I?

- Tom Robert
- 8 Years experience
- Consultant for
I am Root
- Capgemini /
Lampiris-Total



EpheWattal
Enviroments ?

”

Ephemeral

. Fleeting

. Lasting for a
very short time

. Momentary



Why ?

Why?

use case 1:

- Client wants a color change on the website
- Ongoing development on the staging server
- Can't show the change on your local machine
- Create a poc environment to showcase it.

use case 2:

- Working a big team with multiple project teams on the same codebase
- Test and showcase the features individually
- Feature environment for each project

use case 3:

- Business is testing a release candidate
- You want to run a full test suite without interfering with the business tests
- Create a test environment to run the tests

...

Stack

Stack



Gitlab



Docker



Kubernetes

Gitlab

One tool to rule them all.

- Git repository
- CI/CD pipelines
- Docker registry
- ...



Docker

<https://www.docker.com>

- Tool to create and manage software containers
- Containers are isolated processes like an nginx or mysql

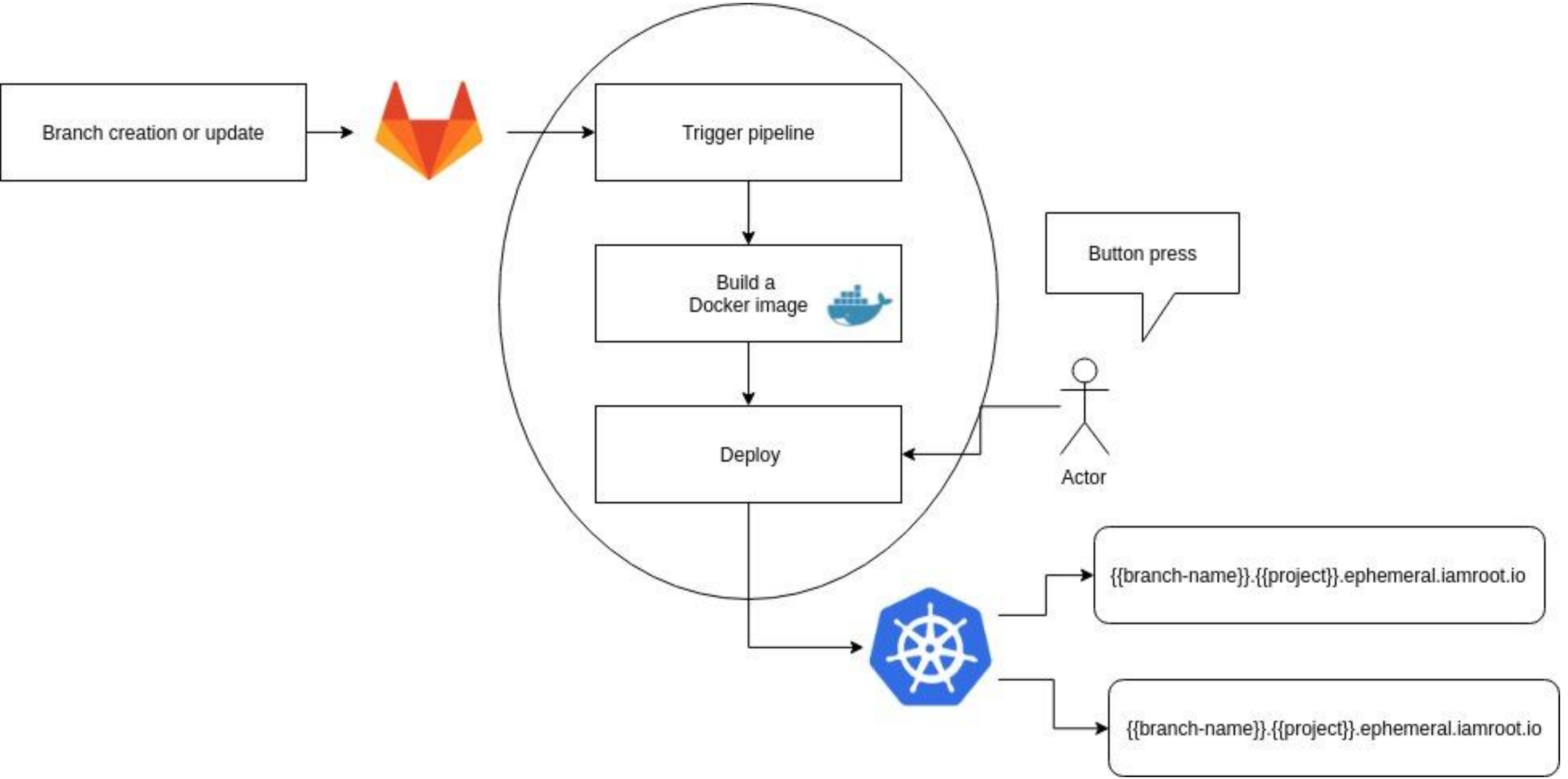


Kubernetes

<https://kubernetes.io/>

- A container orchestration tool designed by google
- It allows configuration, deploying and scaling for software Containers





Prerequisites

(Pre-)Production

<http://umami.iamroot.io/>

Search by keyword, ingredient, dish Search

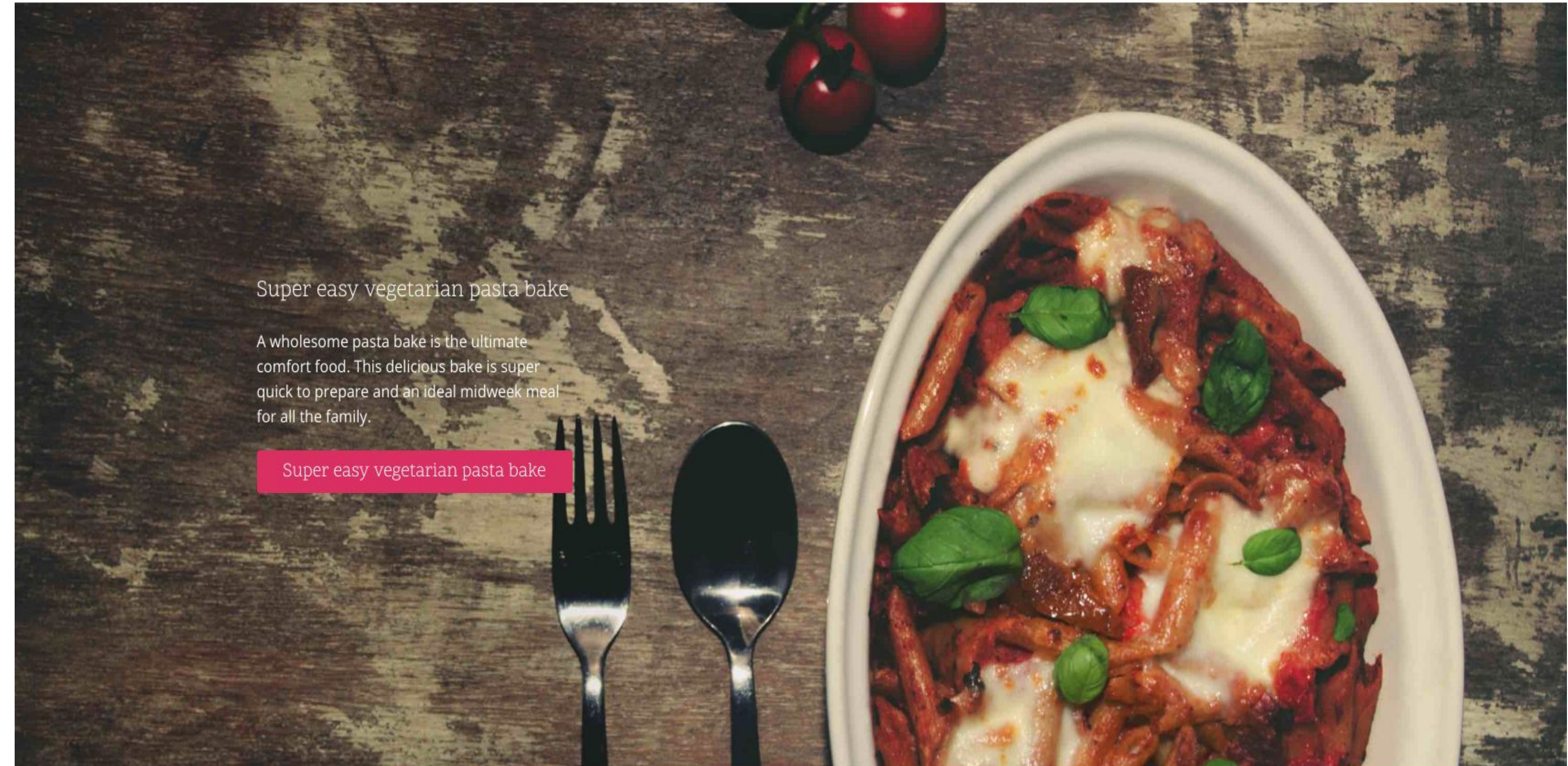
Log in

umami
FOOD MAGAZINE

[Home](#)

[Articles](#)

[Recipes](#)



Super easy vegetarian pasta bake

A wholesome pasta bake is the ultimate comfort food. This delicious bake is super quick to prepare and an ideal midweek meal for all the family.

[Super easy vegetarian pasta bake](#)

Skip the spirits with delicious mocktails

[VIEW ARTICLE >](#)



Gitlab

- Umami project:
 - <https://gitlab.com/i-am-root/umami>
 - Branches:
 - Master
 - Develop
 - feature/pizza
- Gitlab-CI runner with dind (Docker in docker)
 - https://docs.gitlab.com/ee/ci/docker/using_docker_build.html

Lifeline: `gitlab-ci.yml`

Gitlab-ci stages

- Build
- Package (Docker package)
- Deploy (K8S deploy)

<https://gitlab.com/i-am-root/umami/blob/master/.gitlab-ci.yml>

Docker package

Docker package: Base Dockerfile

- Docker nginx-php image:
 - <https://gitlab.com/i-am-root/drupal-nginx-php/blob/master/7.2.3/Dockerfile>
 - Nginx
 - Php 7.2
 - Drupal dependencies
 - Composer
 - Node
 - Drush launcher
 - Deploy ssh key's
 - Supervisor

Docker package: Dockerfile

<https://gitlab.com/i-am-root/umami/blob/master/deployment/docker/app/Dockerfile>

- Starts from the Base image
- Copy a tarball of the project created by gitlab-ci
- Copy the start.sh script
- Use Wait-for-it.sh to be sure that mysql is up and running before starting the nginx container
 - This is legacy from docker-swarm, we could use K8S lifecycle hooks or such.

Docker package: Gitlab-ci

```
.docker-package: &docker-package
  stage: package
  tags:
    - dockerbuild
  script:
    - echo ${VERSION}
    - tar --exclude="./deployment" -czf deployment/docker/app/package.tar.gz .
    - docker login -u gitlab-ci-token -p $CI_BUILD_TOKEN $CI_REGISTRY
    - docker build -t $CI_REGISTRY_IMAGE:${CI_COMMIT_REF_SLUG}_${CI_COMMIT_SHA} ./deployment/docker/app
    - docker push $CI_REGISTRY_IMAGE:${CI_COMMIT_REF_SLUG}_${CI_COMMIT_SHA}

docker-package-ephemeral:
  <<: *docker-package
  only:
    - master
    - /^release.*$/
    - /^feature.*$/
    - /^hotfix.*$/
```

K8S deploy

K8S deploy: yml files

- namespace.yml
 - <https://gitlab.com/i-am-root/umami/blob/master/deployment/k8s/namespace.yml>
 - No need to change the service or container name for network access (drupal, mysql). The access is contained within the namespace.
- pvc.yml
 - <https://gitlab.com/i-am-root/umami/blob/master/deployment/k8s/pvc.yml>
 - Mysql & drupal
- mysql.yml
 - <https://gitlab.com/i-am-root/umami/blob/master/deployment/k8s/mysql.yml>
 - Default mariadb deploy
- drupal.yml
 - <https://gitlab.com/i-am-root/umami/blob/master/deployment/k8s/drupal.yml>

K8S deploy: drupal.yml : Ingress

K8S traffic router (Like traefik or nginx-proxy)

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: drupal-ingress
  namespace: master-umami
spec:
  rules:
  - host: master.umami.ephemeral.iamroot.io
    http:
      paths:
      - backend:
          serviceName: drupal
          servicePort: 80
```


K8S deploy: drupal.yml image

```
spec:
  containers:
  -
    image: registry.gitlab.com/i-am-root/umami:latest
    name: drupal
    ports:
    - containerPort: 80
```

https://gitlab.com/i-am-root/umami/container_registry

K8S deploy: gitlab-cli

```
.k8s-deploy: &k8s-deploy
  stage: deploy
  image: alpine
  environment:
    name: develop
  script:
    - apk add --no-cache curl
    - curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/
    - chmod +x ./kubectl
    - mv ./kubectl /usr/local/bin/kubectl
    - kubectl config set-cluster nosebit --server="$KUBE_URL" --insecure-skip-tls-verify=true
    - kubectl config set-credentials admin --username="$KUBE_USER" --password="$KUBE_PASSWORD"
    - kubectl config set-context default --cluster=nosebit --user=admin
    - kubectl config use-context default
    - sed -i 's/master/"${CI_COMMIT_REF_SLUG}"/g' ./deployment/k8s/namespace.yml
    - sed -i 's/master/"${CI_COMMIT_REF_SLUG}"/g' ./deployment/k8s/pvc.yml
    - sed -i 's/master/"${CI_COMMIT_REF_SLUG}"/g' ./deployment/k8s/mysql.yml
    - sed -i 's/master/"${CI_COMMIT_REF_SLUG}"/g' ./deployment/k8s/drupal.yml
    - sed -i 's@"${CI_REGISTRY_IMAGE}:latest"@"${CI_REGISTRY_IMAGE}:${CI_COMMIT_REF_SLUG}_${CI_COMMIT_SHA}"@g' ./deployment/k8s/drupal.yml
    - kubectl apply -f ./deployment/k8s/namespace.yml
    - kubectl apply -f ./deployment/k8s/pvc.yml
    - kubectl apply -f ./deployment/k8s/mysql.yml
    - kubectl apply -f ./deployment/k8s/drupal.yml
```

After K8S deploy

Install and configure the drupal ephemeral

Docker entrypoint: start.sh

```
# Install Drupal
cd /var/www/html

# install dependencies
composer install

# Build env.
./vendor/bin/phing build-dev

# Installs Drupal
./vendor/bin/phing install-dev

# Make sure nginx can change the files
chown -R www-data:www-data web/sites/default/files

# Start supervisord and services
exec /usr/bin/supervisord -n -c /etc/supervisord.conf
```

Phing: a php build tool

<https://gitlab.com/i-am-root/umami/blob/master/build.dist.xml>

build-dev:

- compile theme
- run custom build scripts

Install dev:

- Install site
- Writable files folder
- Sync database from (pre-)prod
- Sync files from (pre-)prod
- Run custom deploy script
 - drush entup
 - drush dbup
 - drush cim
 - drush cr

Drush sync files & db

Drush 9: drush aliases in code

<https://gitlab.com/i-am-root/umami/blob/master/drush/sites/self.site.yml>

```
prod:
  host: umami.iamroot.io
  user: cicd
  root: /var/www/umami/web
  uri: http://umami.iamroot.io
  ssh:
    options: "-o StrictHostKeyChecking=no"
```

User cicd is a read-only user on the pre-prod machine.

ssh private key:

https://gitlab.com/i-am-root/drupal-nginx-php/blob/master/7.2.3/config/id_rsa_iamroot

!! Security issue: drush sql-cli is fully accessible (only use internally)

Full circle

delete the ephemeral

Destroy the ephemeral by deleting the K8S namespace

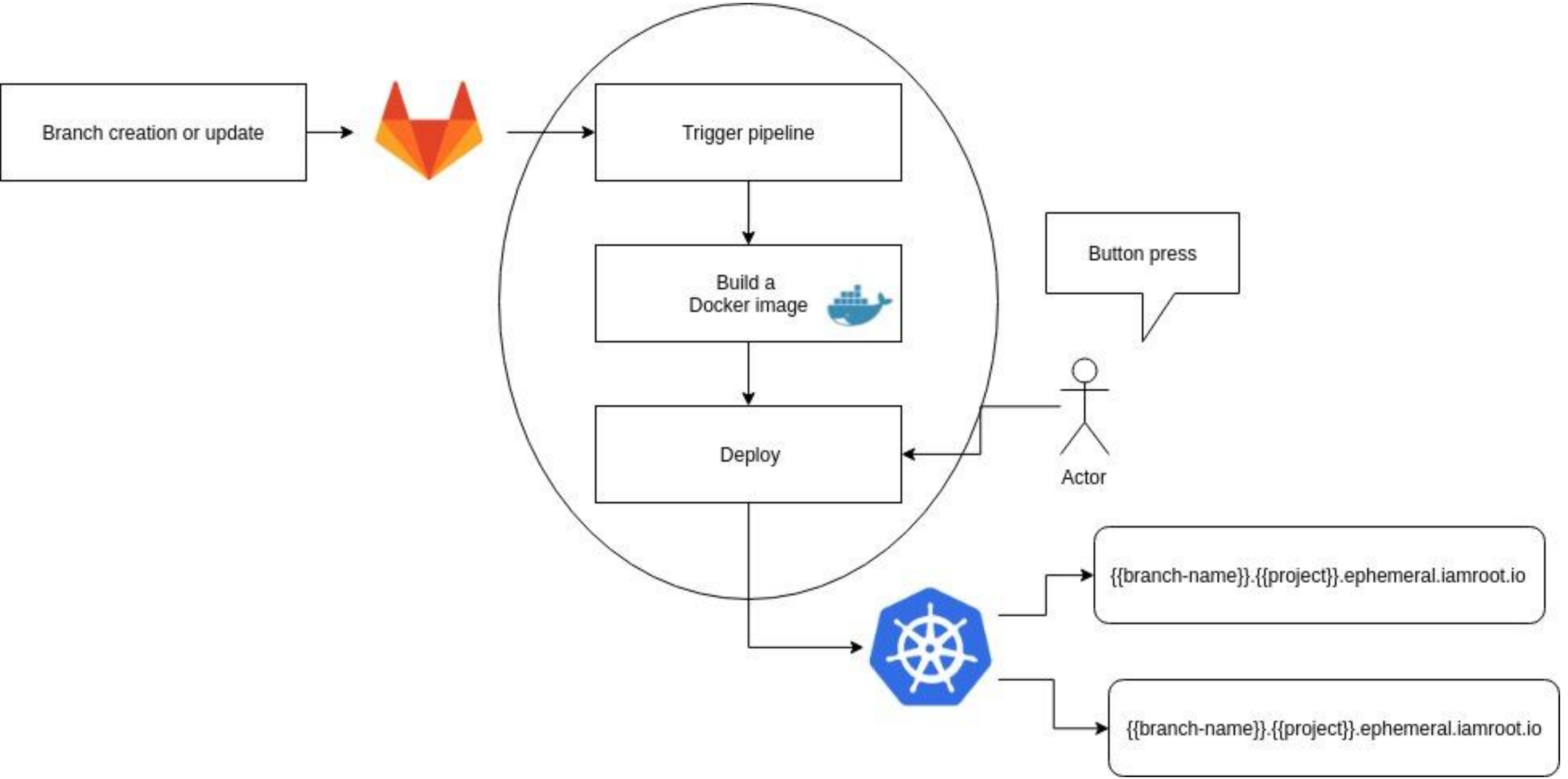
- Command-line
 - `kubectl delete namespaces <namespace>`
- Web interface of cluster
- Create an api interface

Improvements

Improvements

- Composer install is run a lot:
 - We could use a gitlab artifacts as basefile (with --dev and without)
- wait-for-it.sh was needed for docker-swarm. K8S has lifecycle hooks.
- Redeploy of feature we should not sync DB and Files
- Using SSH keys has some security implementations.
- Easy deletion after use.
- ...

Lets see the magic
happen!



Demo setup

- <http://umami.iamroot.io>
- Demo route with the branch name
 - <http://umami.iamroot.io/my-awesome-path>

The screenshot shows the Umami Food Magazine website interface. At the top, there is a search bar with the text "Search by keyword, ingredient, dish" and a "Search" button. To the right of the search bar is a "Log in" link. Below the search bar is the Umami Food Magazine logo, which consists of the word "umami" in a stylized orange font and "FOOD MAGAZINE" in a smaller, black, sans-serif font below it. To the right of the logo is a navigation menu with the following items: "Home", "Articles", "Recipes", and "Awesome", where "Awesome" is underlined. Below the navigation menu is a breadcrumb trail: "Home » My awesome path". The main heading of the page is "My awesome path". Below this heading is a sub-heading "My awesome Master". Underneath, there is a section titled "Recipe collections" which contains a grid of 16 categories: Alcohol free, Baked, Baking, Cake, Carrots, Cocktail party, Dinner party, Drinks, Egg, Grow your own, Healthy, Herbs, Learn to cook, Mushrooms, Party, and Pasta. At the bottom of the page, there is a promotional banner for Umami Food Magazine. On the left side of the banner is an image of three magazine covers. The middle cover is titled "The BAKING SPECIAL" and features a recipe for "Shyme". A red circular badge on the right side of the image says "3 ISSUE BUNDLE!". To the right of the image, the text reads "Umami Food Magazine" followed by "Skills and know-how. Magazine exclusive articles, recipes and plenty of reasons to get your copy today." and a "Find out more >" link. On the far right of the banner, there is a "Tell us what you think" link and a "Contact" link.

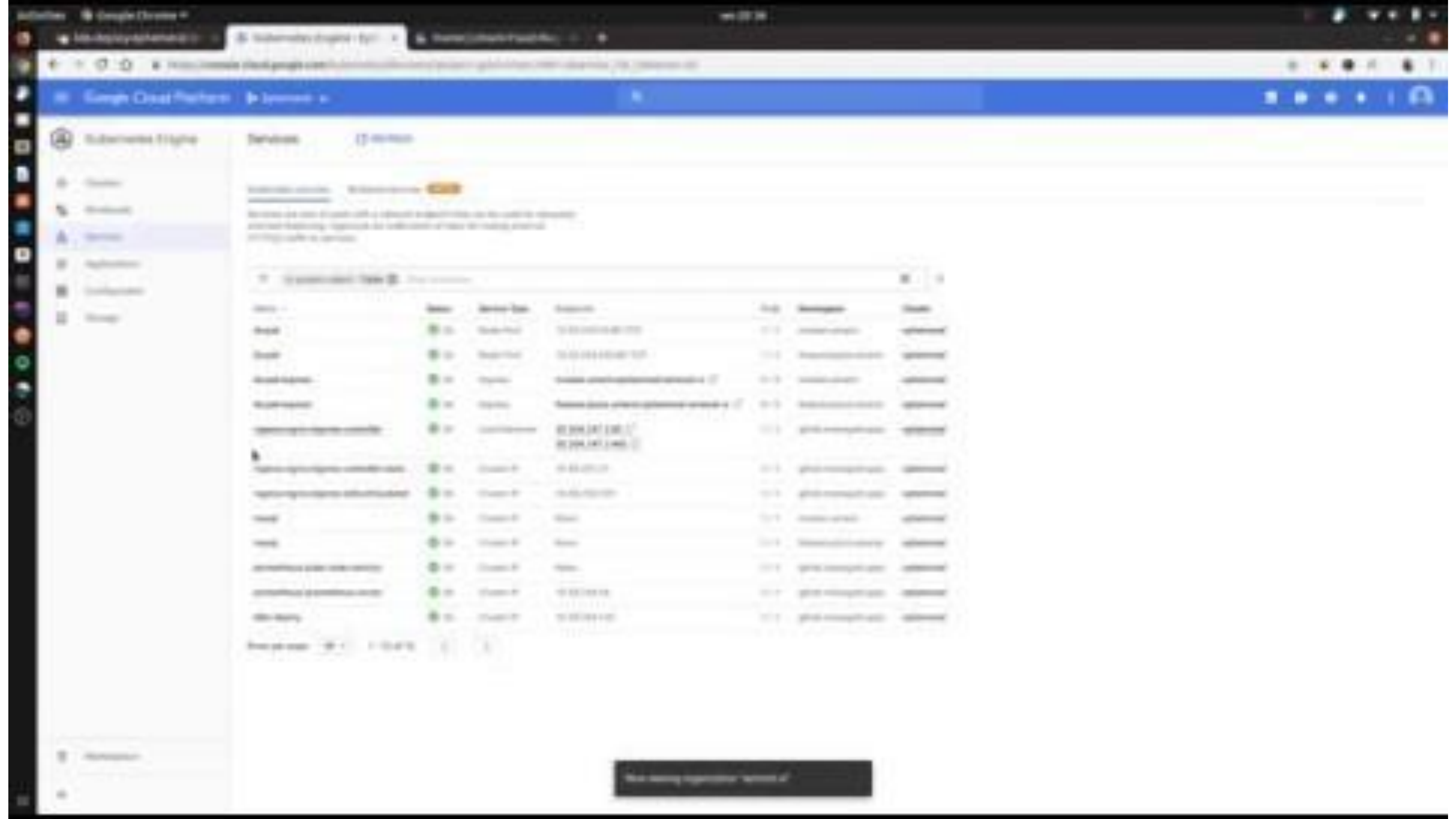
Search for products, ingredients, etc. | Search

Log in



Home | Articles | Recipes | **Ingredients**





Questions?

Thank you



DRUPALCAMP GHENT
23 & 24 NOVEMBER 2018

AUSY
a Randstad company

**HO
GENT**

LEVEL 27

 **agiledrop**

 **DropSolid**
Makkelijk Digitaal Ondernemen

 **Combell**
Your host on the internet

AMPLEXOR

