Supercharge your Developer Experience

Aditi Garg, Engineering Manager
Sandosh Vasudevan, Architect

DeveloperWeek Feb 2020
Supercharged Experience?
Idea → Code → Biz Value

Developer Experience?
Developer Experience
Reality
Poll. How long to deploy?

- Hours
- Days
- A week
- Should there even be a fourth option?
3 days
from change to deploy

2017
Problem
Tech Stack

Property API Java

Mobile API PHP

WWW Server PHP

iOS / Android

JS + HTML + CSS
Legacy Tech Stack

Property API Java

Mobile API PHP

iOS / Android

WWW Server PHP

JS + HTML + CSS
Legacy Tech Stack

- iOS / Android
  - Mobile API
    - PHP
- JS + HTML + CSS
  - Property
    - API
    - Java
  - WWW
    - Server
      - PHP
Release Cycle

- **Friday**
- **Monday**
- **Tuesday**
- **Wednesday**
- **Thursday**

**Steps:**
- Cut a release branch
- Fix critical bugs
- Deploy to production
Release Cycle

- Friday: Cut a release branch
- Monday: Fix critical bugs
- Tuesday: Deploy to production
- Wednesday: Cut a new release branch
- Thursday: Deploy to production
- Fix critical bugs
Codebase
Release frequency
Infrastructure
impact
Developer Experience
Problems

- Local
- Deploy
- Cloud
Local

Deploy

Cloud

Goals
Local

- Faster setup
- Break up monolith
- Reduce build times
Deploy

- Automate tests
- Make production releases accessible
- Easy rollbacks
- Make infrastructure provisioning easy
Goals

Local  Deploy  Cloud
Cloud

- Auto scaling through config
- Preconfigured logging
- Monitoring hooks
- Observability hooks
Containers all the way!
Containers

● Consistent environment
● Run anywhere
● Isolation
● Easy development
● Version control
Kubernetes

- Transparent to developers
- Zero friction deploys
- Self service provisioning
- Auto scaling
Kubernetes

- Canary deployment
- Sidecar deployments
- Instantaneous “anything” deployments
CI / CD Pipeline

- Application agnostic
  - Support publishing packages
- Extensible
- Pull Request triggered
How it works
How it works
How it works
How it works
How it works
How it works
How it works
2017

3 days
from change to deploy
3 days from change to deploy
- name: "susies-project-varnish"
  namespace: "susies-project"
  spec:
    useCustomVcl: false
    ttl: "7d"
    upstreams:
    - protocol: "https"
      host: "susies-project.itx.com"
Thank you

@adigarg94
@sandosh
Basic Template for ITX Auto DevOps
Basic Template for ITX Auto DevOps

- deployment_configuration
- provision
- gitlab-ci.yml
- Dockerfile
- README.md
# This is sample dockerfile. Replace it with content for your project
# Tips: Use multi-stage build https://docs.docker.com/develop/develop-images/multistage-build/

FROM nginx-demos/hello:plain-text
# Simple web server. Your app will use base image suitable for your language
EXPOSE 80
# Need to export port 80 and /health endpoint if the app is intended to be run in k8s
ENV DO_SERVICE_NAME=""
ENV DO_AGENT_HOST=localhost
ENV DO_STATUSD_SOCKET=""
# You most likely need to provide entrypoint to your app
ENTRYPOINT []
Basic Template for ITX Auto DevOps

- deployment_configuration
- provision
- gitlab-cli.yml

Add provision and deploy config examples

Add DD env

Update doc

Last commit
2 weeks ago
3 weeks ago
1 month ago

Last update
2 weeks ago
include:
  - project: its/ci-cd/application-pipeline
  ref: master
  file: templates/auto-devops.gitlab-ci.yml

# extra "parameters" passed to pipeline
# another (possibly better) way to define parameters is in GitLab UI where they can be scoped per environment
# Note: this is NOT variables passed to your K8S app at runtime. For that use deployment_configuration folder

# variables:
TF_VAR_owner: "beavers@domain.com"
TF_VAR_product: "Manhattan"
TF_VAR_team: "Beavers"
Basic Template for ITX Auto DevOps

- deployment_configuration
- provision
- .gitlab-ci.yml
- Dockerfile
- README.md

<table>
<thead>
<tr>
<th>Name</th>
<th>Last commit</th>
<th>Last update</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployment_config</td>
<td>add provision and deploy config examples</td>
<td>2 weeks ago</td>
</tr>
<tr>
<td>provision</td>
<td>add provision and deploy config examples</td>
<td>2 weeks ago</td>
</tr>
<tr>
<td>.gitlab-ci.yml</td>
<td>add provision and deploy config examples</td>
<td>2 weeks ago</td>
</tr>
<tr>
<td>Dockerfile</td>
<td>add DD env</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>README.md</td>
<td>update doc</td>
<td>1 month ago</td>
</tr>
</tbody>
</table>
```tf
3  # module "kafka" {
4    # source = "git:https://gitlab.zgtools.net/ltx/infrastructure/terraform/k8s_kafka.git?ref=master"
6    # app
7    # environment = var.environment
8    # team = var.team
9    # owner = var.owner
10   # product = var.product
11   # k8s_namespace = "${var.environment}-app"
12   # vpc_name = var.vpc_name
13   # cluster_fqdn = var.cluster_fqdn
14  }
```
# If you need to run your app with assumed aws role, uncomment the following lines and update it with role arn
#
# podAnnotations:
# iam.amazonaws.com/role: "arn:aws:iam::blah-blah"
#
# default infrastructure info for the app
minReplicas: 1
maxReplicas: 50

# HPA scaling
resourceMetricName: cpu
resourceMetricTargetUtilization: 80

# all env vars need to be defined in Dockerfile as ENV
applicationEnvironmentVariables:
  - name: MY_VAR
    value: "Whatever"