Life outside the Chocolate Factory

Why employee productivity is a reliability issue

@msuriar @EmilStolarsky
Disclaimer
Disclaimer
Disclaimer
Onboarding

1st day
Getting Started

Welcome to Juicero!

This microservices monorepo runs on Kubernetes, OpenShift, and OpenStack - just like all the fruits in a salad! To get started, go ahead and clone our repo:

```bash
git clone git@github.com:juicero/squeezer.git
```

Table of Contents

- Getting Started
- Our Mission
- Testing
[~] -> % mkdir -p work/juicero

[~] -> %
 [~,] -> % mkdir -p work/juicero
[~,] -> % cd work/juicero
[~/work/juicero] -> %
[~] -> % mkdir -p work/juicero
[~] -> % cd work/juicero
[~/work/juicero] -> % git clone git@github.com:juicero/squeezer.git
Cloning into 'squeezer'...
[~] -> % mkdir -p work/juicero
[~] -> % cd work/juicero
[~/work/juicero] -> % git clone git@github.com:juicero/squeezer.git

Cloning into 'squeezer'...
ERROR: Repository not found.
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
This is not the web page you are looking for.
Getting Started

Welcome to Juicero!

This microservices monorepo runs on Kubernetes, OpenShift, and OpenStack - just like all the fruits in a salad! To get started, go ahead and clone our repo:

```
$ git clone git@github.com:juicero/squeezer.git
```

Table of Contents

- Getting Started
- Our Mission
- Testing
Getting Started

Welcome to Juicer!

This microservices monorepo runs on Kubernetes, OpenShift, and OpenStack - just like all the tools in a salon! To get started, go ahead and clone our repo:

```
  git clone git@github.com:juicer/snapewzer.git
```

Table of Contents

- Getting Started
- Our Mission
- Testing
“Make sure to ask IT to be added to the Github Organization before cloning 😊”
[~/work/juicero] -> % git clone git@github.com:juicero/squeezer.git
~/work/juicero] -> % git clone git@github.com:juicero/squeezer.git
Cloning into 'squeezer'...
Cloning into 'squeezer'...
remote: Enumerating objects: 22223, done.
remote: Counting objects: 100% (581/581), done.
remote: Compressing objects: 100% (286/286), done.
remote: Total 22223 (delta 360), reused 447 (delta 275), pack-reused 21642
Receiving objects: 100% (22223/22223), 14.85 MiB | 2.11 MiB/s, done.
Resolving deltas: 100% (15593/15593), done.
[~/work/juicero] -> % cd squeezer
[~/work/juicero] -> % cd squeezer
[~/work/juicero/squeezer] -> %
[~/.work/juicero] -> % cd squeezer

[~/.work/juicero/squeezer] -> % script/setup
[~/work/juicero] -> % cd squeezer
[~/work/juicero/squeezer] -> % script/setup

Setting up repo...
[~/work/juicero] -> % cd squeezer

[~/work/juicero/squeezer] -> % script/setup

Setting up repo...

bash: docker: command not found

[~/work/juicero/squeezer] -> %
Onboarding

Deploy

1st day

1st week
INFO: 2022/11/10 12:01:06 main.go:26: Starting the application...
INFO: 2022/11/10 12:01:07 main.go:27: Enabling juice DRM
INFO: 2022/11/10 12:01:06 main.go:26: Starting the application...
INFO: 2022/11/10 12:01:07 main.go:27: Enabling juice DRM
package main

import {
    "log"
    "os"
}

var {
    InfoLogger  *log.Logger
}

func init() {
    if err != nil {
        log.Fatal(err)
    }

}

func main() {
    InfoLogger.Println("Starting the application...")
    InfoLogger.Println("Enabling juice DRM")

    // TODO: figure out viable business model
}
package main

import {
    "log"
    "os"
}

var {
    InfoLogger  *log.Logger
}

func init() {
    if err != nil {
        log.Fatal(err)
    }

}

func main() {
    InfoLogger.Println("Starting the application...")
    InfoLogger.Println("Enabling juice DRM")

    // TODO: figure out viable business model
}
package main

import {
    "log"
    "os"
}

var {
    InfoLogger         *log.Logger
}

func init() {
    if err != nil {
        log.Fatal(err)
    }

}

func main() {
    InfoLogger.Println("Starting the application...")
    InfoLogger.Println("Enabling juice DRM")

    // TODO: figure out viable business model
}
All checks have passed
2 successful checks

- **ci/circleci: setup** — Your tests passed on CircleCI!
- **ci/circleci: build** — Your tests passed on CircleCI!
Fix spelling error #42

Open msuriar wants to merge 1 commit into main from msuriar/fix-log-line
Fix spelling error #42

msuriar wants to merge 1 commit into main from msuriar/fix-log-line

All checks have passed
1 failed and 2 successful checks

- ci/circleci: setup — Your tests passed on CircleCI!
- ci/circleci: build — Your tests passed on CircleCI!
- ci/circleci: test — Your tests failed on CircleCI!
Cloning into 'squeezer'...

Building code...

Exited with code exit status 4
```makefile
endif

ifeq ($(MOJOY_GROUP_ID),)
  MOJOY_GROUP_ID = $(MOJOY)
endif

ifeq ($(BASE_IMAGE_STABILITY_TAG),)
  BASE_IMAGE_TAG += $(BASE_IMAGE_STABILITY_TAG)
else
  BASE_IMAGE_TAG = $(BASE_IMAGE_TAG)-$(BASE_IMAGE_STABILITY_TAG)
endif

REGISTRY = docker.io
REPO = $(REGISTRY)/mojoy/sicero
NAME = sicero-$(JUICECO_MINOR_VER)

ifeq ($(STABILITY_TAG),)
  ifeq ($(TAG),latest)
 override TAG = $(TAG)-$(STABILITY_TAG)
  endif
endif

.PHONY: build buildx-build buildx-push buildx-build-amd64 test push shell run start stop tags clean release
debug: build

# Make sure to download personal aws_config.yml from AWS on your first commit
build:
docker build -t $(REPO):$(TAG) \
  -f ./Dockerfile \
  -p $(MOJOY) \
  -v $(MOJOY):/app \
  $(MOJOY) \
  --build-arg BASE_IMAGE_TAG=$(BASE_IMAGE_TAG) \
  --build-arg JUICECO_VER=$(JUICECO_VER) \
  --build-arg WOODY_GROUP_ID=$(WOODY_GROUP_ID) \
  --build-arg WOODY_USER_ID=$(WOODY_USER_ID) .

# --load doesn't work with multiple platforms https://github.com/docker/buildx/issues/59
# we need to save echo to run tests first.
buildx-build-amd64:
docker buildx build --platform=linux/amd64 \
  --build-arg BASE_IMAGE_TAG=$(BASE_IMAGE_TAG) \
  --build-arg JUICECO_VER=$(JUICECO_VER) \
  --build-arg WOODY_GROUP_ID=$(WOODY_GROUP_ID) \
  --build-arg WOODY_USER_ID=$(WOODY_USER_ID) \
  --load \
  -t $(REPO):$(TAG) .

buildx-build:
docker buildx build --platform=linux \
  --build-arg BASE_IMAGE_TAG=$(BASE_IMAGE_TAG) \
  --build-arg JUICECO_VER=$(JUICECO_VER) \
  --build-arg WOODY_GROUP_ID=$(WOODY_GROUP_ID) \
  --build-arg WOODY_USER_ID=$(WOODY_USER_ID) \
```

Makefile
# Make sure to download personal aws_config.yml from AWS on your first

docker build -t $REPO:$TAG \
- v "$(pwd)"/aws_config.yml:/app:/ \
--build-arg BASE_IMAGE_TAG=$BASE_IMAGE_TAG \
--build-arg JUICERO_VER=$JUICERO_VER \
--build-arg WODBY_GROUP_ID=$WODBY_GROUP_ID \
--build-arg WODBY_USER_ID=$WODBY_USER_ID ./
[~/work/juicero/squeezer] -> %
[~/work/juicero/squeezer] -> % aws s3 cp s3://juicero-public/secrets/msuriar/aws_config.yml
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
[~/work/juicero/squeezer] -> % aws s3 cp s3://juicero-public/secrets/msuriar/aws_config.yml
copy: s3://juicero-public/secrets/msuriar/aws_config.yml to ~/work/juicero/squeezer/aws_config.yml

[~/work/juicero/squeezer] -> %
[~/work/juicero/squeezer] -> % aws s3 cp s3://juicero-public/secrets/msuriar/aws_config.yml

copy: s3://juicero-public/secrets/msuriar/aws_config.yml to ~/work/juicero/squeezer/aws_config.yml

[~/work/juicero/squeezer] -> % git add . &
    git commit --amend --no-edit
    git push origin +msuriar/fix-log-line
Fix spelling error #42

Open

msuriar wants to merge 1 commit into main from msuriar/fix-log-line

---

All checks have passed
1 failed and 2 successful checks

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>ci/circleci: setup — Your tests passed on CircleCI!</td>
<td>Details</td>
</tr>
<tr>
<td>✔️</td>
<td>ci/circleci: build — Your tests passed on CircleCI!</td>
<td>Details</td>
</tr>
<tr>
<td>✗</td>
<td>ci/circleci: test — Your tests failed on CircleCI!</td>
<td>Details</td>
</tr>
</tbody>
</table>
JIRA-00000 Fix spelling error #2

Open

msuriar wants to merge 1 commit into main from msuriar/fix-log-line
JIRA-00000 Fix spelling error #2

Open msuriar wants to merge 1 commit into main from msuriar/fix-log-line

All checks have passed
3 successful checks

- 🔄 ci/circleci: setup — Your tests passed on CircleCI!
- 🔄 ci/circleci: build — Your tests passed on CircleCI!
- 🔄 ci/circleci: test — Your tests passed on CircleCI!
JIRA-00000 Fix spelling error #2

Open

msuriar wants to merge 1 commit into main from msuriar/fix-log-line

Merging is blocked
The base branch restricts merging to authorized users. Learn more about protected branches.

You're not authorized to merge this pull request.
Discussion

● Permissions-Permissions-Permissions
  ○ How many times will someone need to ask to be added to a group or granted certain permissions?

● 2nd Order Effects
  ○ Multiple systems splinter any ability to improve systems as a whole
  ○ Imagine having to navigate Active Directory & then Okta (and AWS IAM and Github and .. and ..)
    ■ Or, if you don’t have to imagine, curl up into a ball and rock back and forth.
I fixed a DNS bug.
Onboarding

Deploy

Ownership

1st day

1st week

1st month
I'm responsible for DNS now
Juicero: (noun) /joos/ər/roʊ
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Component lead</th>
<th>Default assignee</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error logging</td>
<td>Mandatory for all</td>
<td>Maxwell Mei</td>
<td>Component lead</td>
<td>1 issue</td>
</tr>
<tr>
<td>API</td>
<td>Collection of all APIs</td>
<td>Kelly Drozd</td>
<td>Project lead</td>
<td>4 issues</td>
</tr>
</tbody>
</table>
YOU NEED TO LEAVE NOW
Conclusion

Leaving FAANG for a Startup
The Definitive Guide

ORLY?  @msuriar, @EmilStolarsky
Who are we?

Emil

- Never worked at Google
- Work at Wave Mobile Money (prev. Shopify, DigitalOcean, etc.)
- Friends with Xooglers

Murali

- NetOps, SRE at Google (11 years)
- At Snowflake since February
- ??? This talk started as therapy for my culture shock. ???
Ways forward - Onboarding

● For things that everyone needs… do them ahead of start date, or at least make them automated/self-service.

● Have a well lit path for the tasks you expect everyone to do
  ○ have a working dev environment on every cleanly imaged computer

● People shouldn’t need to tweak or configure anything, save enrolling a 2FA device.
Ways forward - CI/CD

- **Fail fast, fail close to user, fail loudly, fail clearly**
  - don’t eat exceptions and reraise
  - make it obvious when something is a permission issue
  - don’t make people request permissions one by one, if everyone on a team has the same privileges

- **Transparency**
  - make it trivial to diff your permissions with a team-mate
  - make it obvious how to expand your permissions (file a ticket here, raise a self service request here, …)
  - security by obscurity is not what you want here
Ways forward - CI/CD

- CI/CD - just do it
  - It seems like a lot of effort when you’re 1-5 engineers
  - But doing it from the start is way cheaper than trying to retrofit it in later on.
  - And doing it early means you’ll sand off the rough edges early.

- Have one way of deploying things to production doing any given thing
  - Building software
  - Running tests
  - Deploying to production
  - Getting logs
  - ...

- One AND ONLY ONE way
Ways forward - Ownership

- Delegation, delegation, delegation
- Tools should support hierarchical namespaces, with delegation at arbitrary points.
  - Otherwise you’re doomed to comms bottlenecks when trying to (re)organise work.
- Code reflects the organization and the organization reflects JIRA
Ways Forward

- Our companies build products that have a UX we care about (i.e. they’re socio-technical systems)
- We often ignore the UX employees experience
- SREs have an outsized impact on that UX
  - Automating development tooling (this could be DevInfra, productivity, whatever, but you end up needing pipelines to implement CI/CD, build, test etc)
  - IAM tooling is usually owned or co-owned by SREs
  - You can delete the boxes JIRA runs on
Thank You

@msuriar  @EmilStolarsky