flight

flight

Rapid Development and Reliable Testing with Docker

Joan Smith

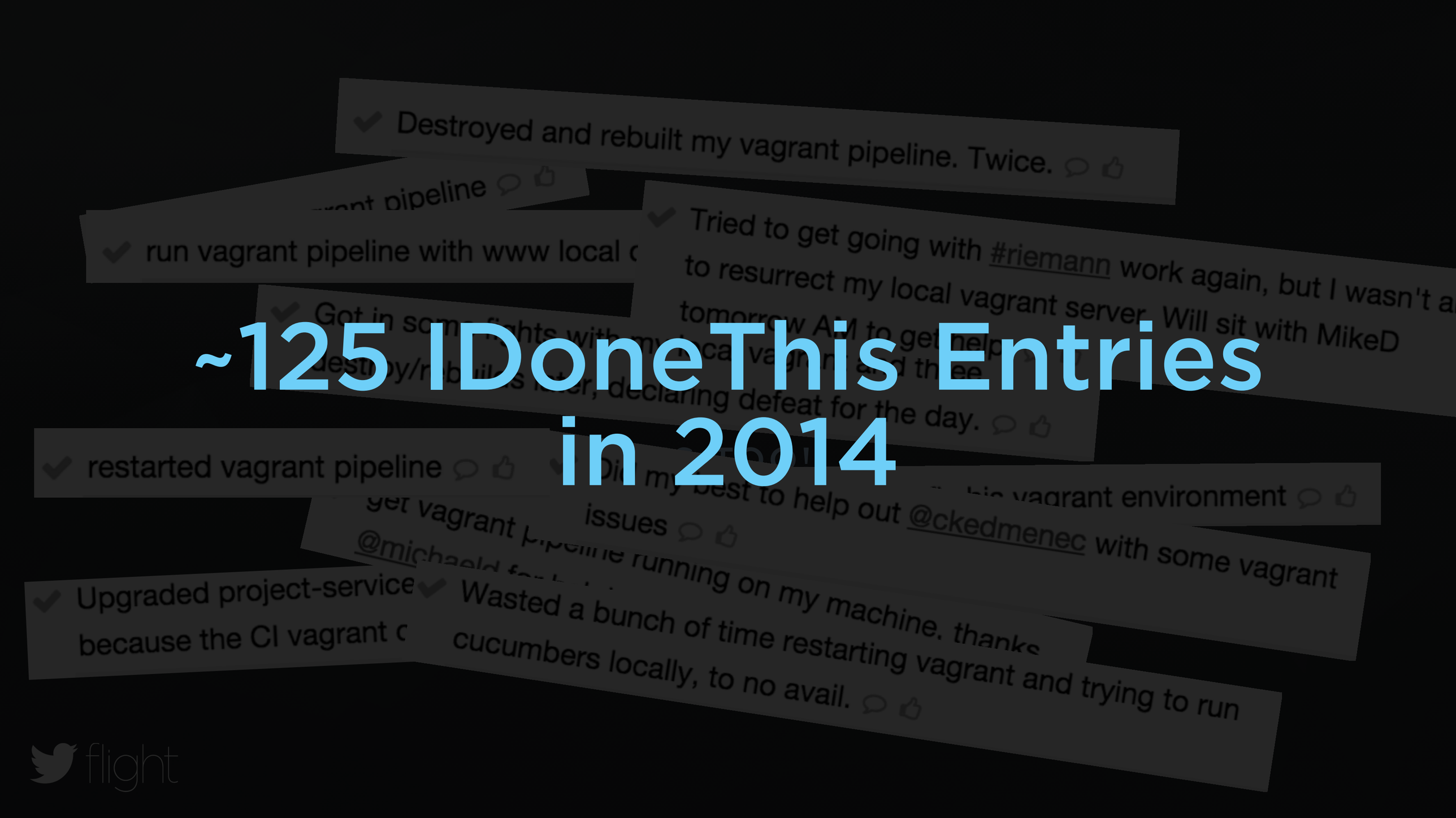
Software Engineer, Fabric | @joans

HOW LONG DID YOU SPEND ON YOUR ENVIRONMENT

Last week?

When you came back from vacation?

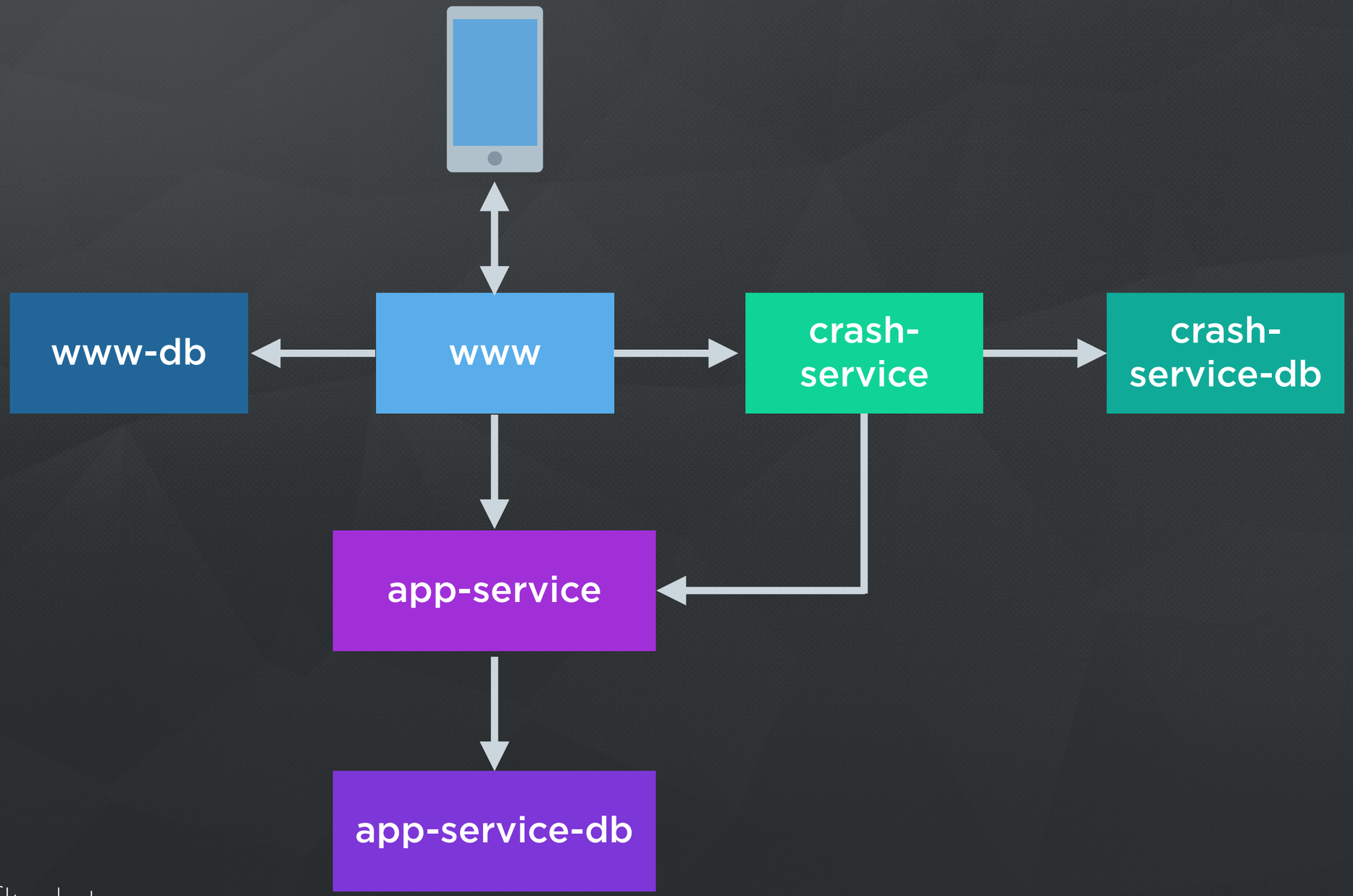
To get tests running locally?


Destroyed and rebuilt my vagrant pipeline. Twice.
run vagrant pipeline with www local
Tried to get going with #riemann work again, but I wasn't a
to resurrect my local vagrant server. Will sit with MikeD
tomorrow AM to get help
declaring defeat for the day.
restarted vagrant pipeline
issues
Upgraded project-service
because the CI vagrant
Wasted a bunch of time restarting vagrant and trying to run
cucumbers locally, to no avail.
~125 IDoneThis Entries
in 2014
flight

ONE WAY TO DO DEVELOPMENT

```
brew install mysql

brew install rails

rails start
```

Common Solution (Ours, Before)


www-db
www
crash-service
crash-service-db
app-service
app-service-db

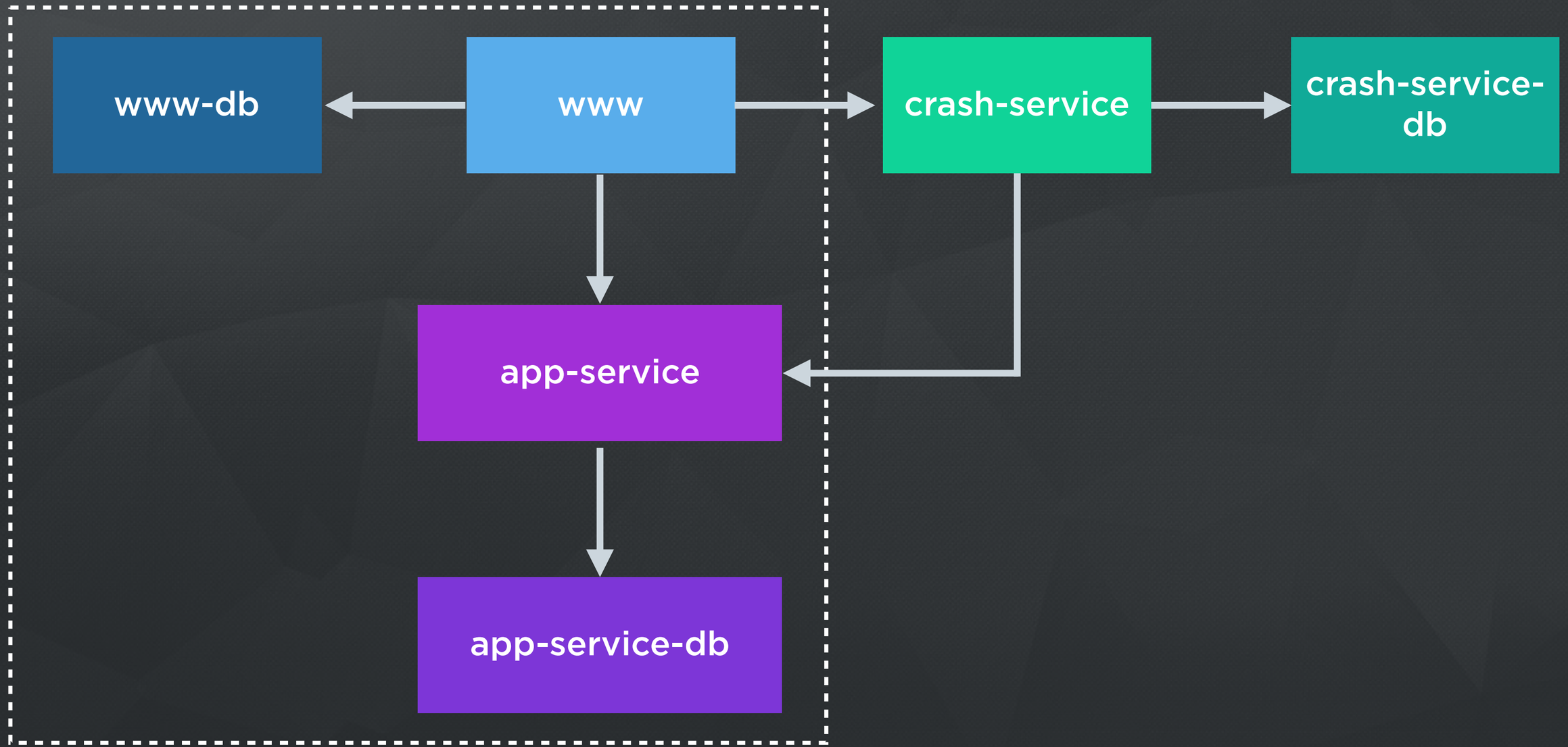
flight


Challenges
flight

1

#microservices

flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

flight

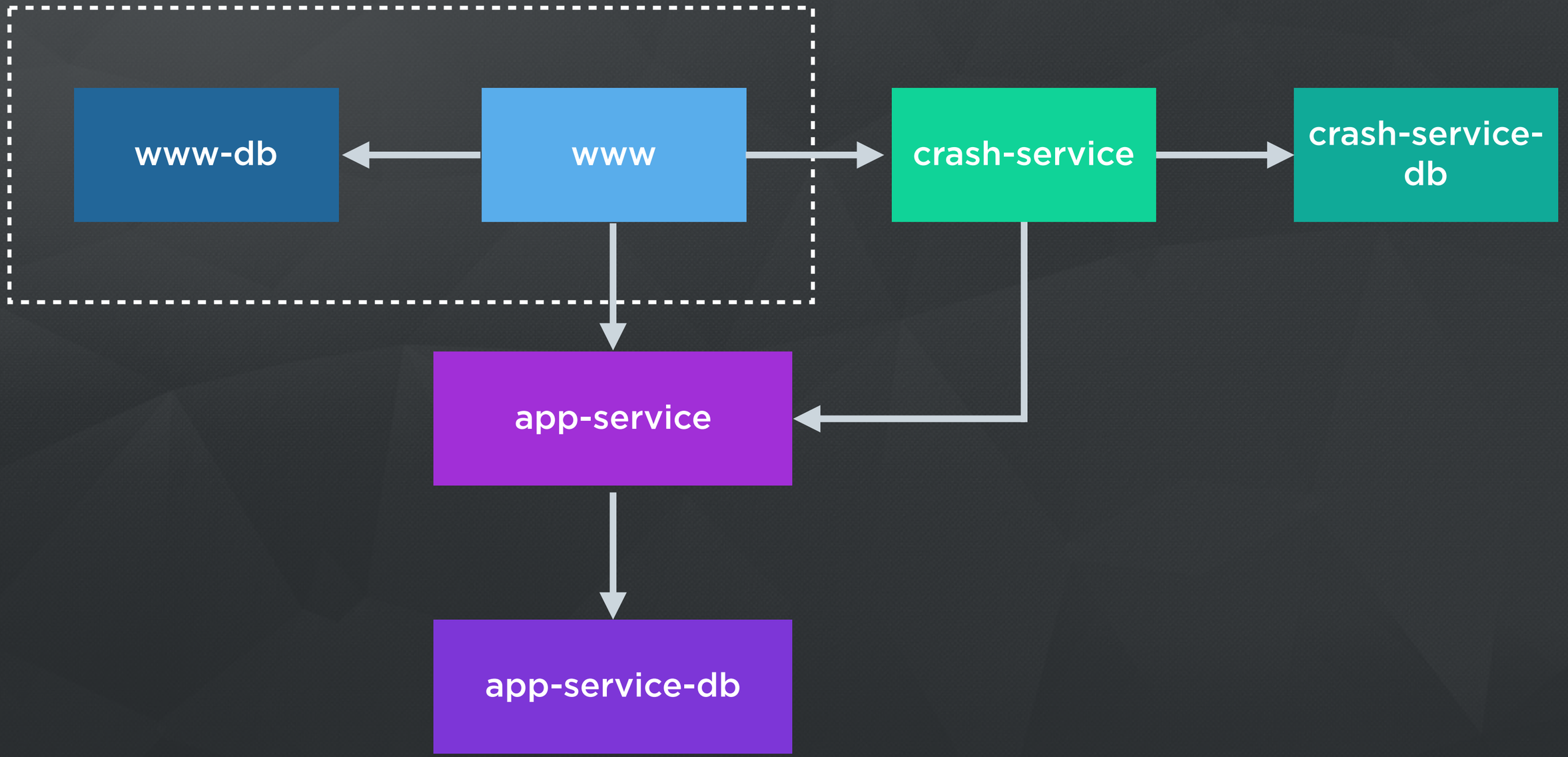
2

Having too many services running

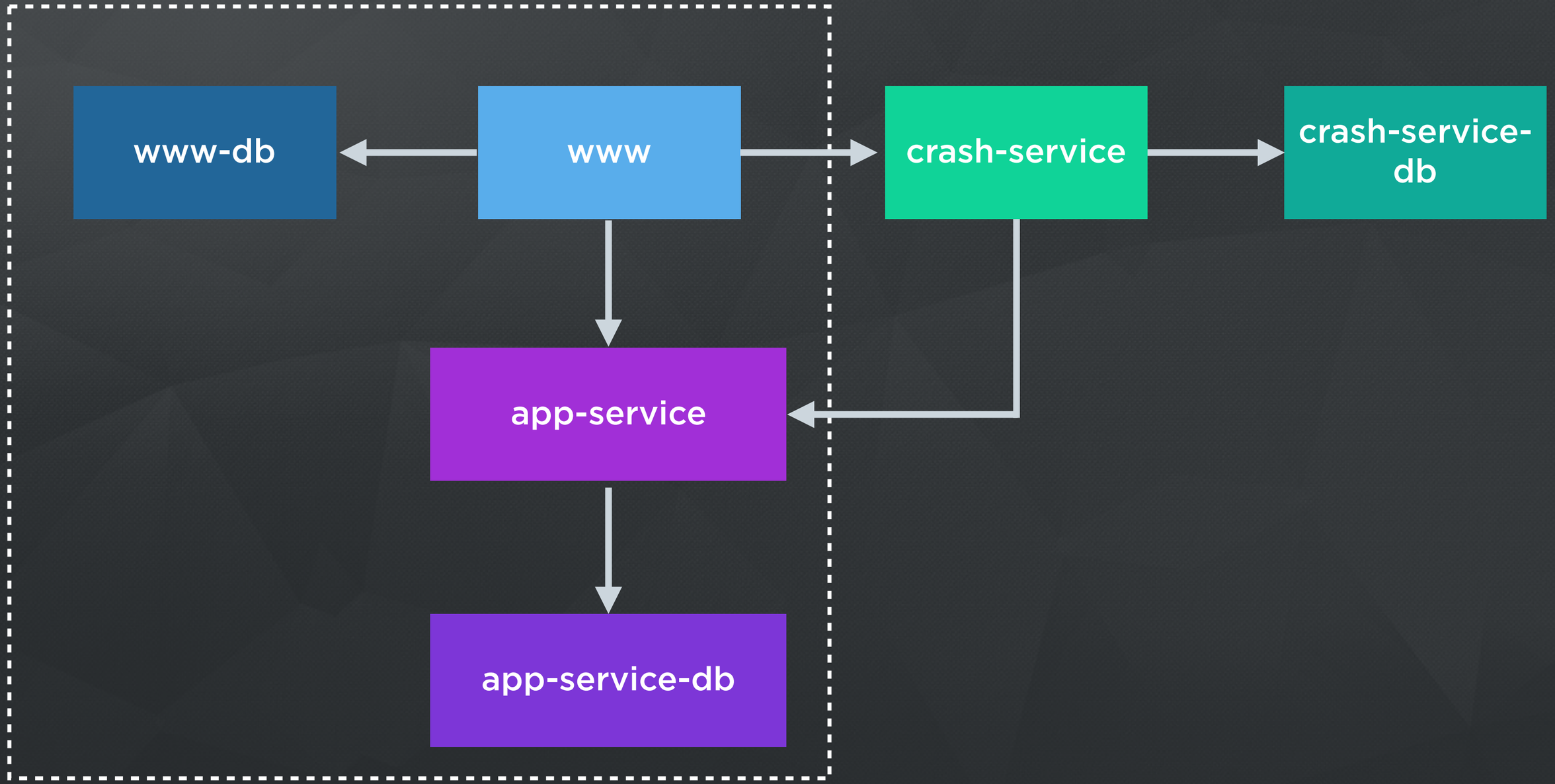
DON'T NEED TO RUN EVERYTHING, JUST WHAT YOU USE

EVERYONE DEVELOPS DIFFERENT SERVICES

flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

flight

3

Chef is additive

EASY TO TURN SERVICES ON
AWKWARD TO TURN THEM OFF

EVERYONE ENDS UP WITH EVERYTHING
RUNNING

flight

4

Vagrant on CI does not scale

flight

5

Pollution in one VM

SHARED ENVIRONMENT LOTS OF OPPORTUNITY FOR POLLUTION

TOO EASY TO MIX DEV/TEST STATE

Can’t we just start what we need?

Single, composable services

Can we have a local development environment strictly separate from test?

That are well isolated

Can we recover quickly from broken environments?

And easy to remove and restart

All while not diverging too much (or too dangerously) from production?

That use the same base OS as Prod

flight

docker

CONTAINERS

Separate OS on the inside
“Executable” on the outside

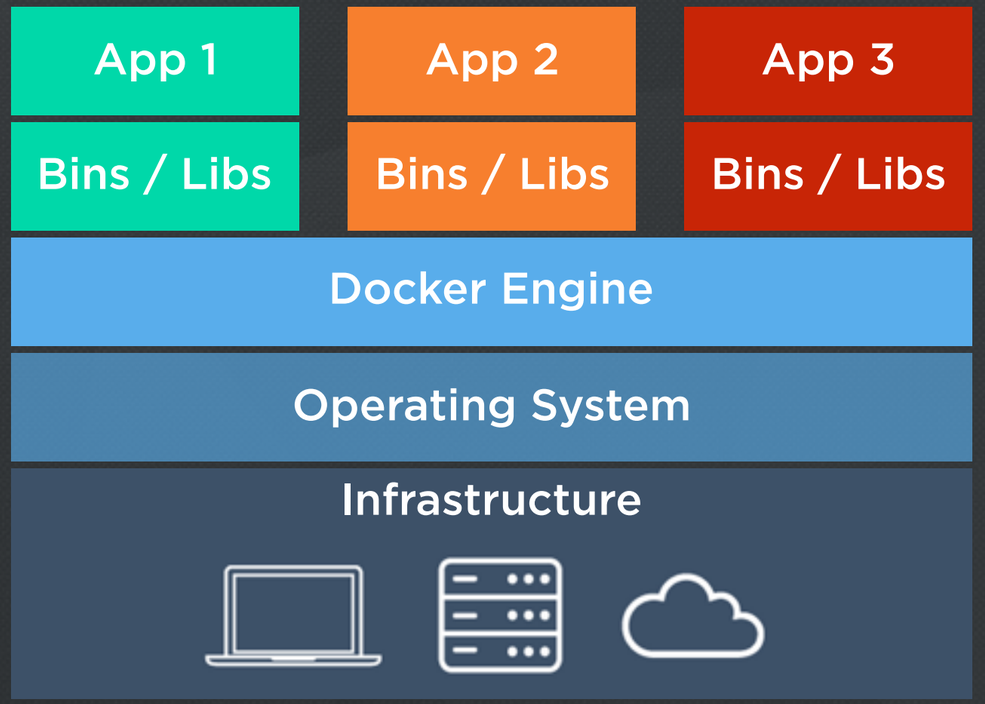
Encapsulates a single process

Isolated, but not in a VM

flight

docker

CONTAINERS

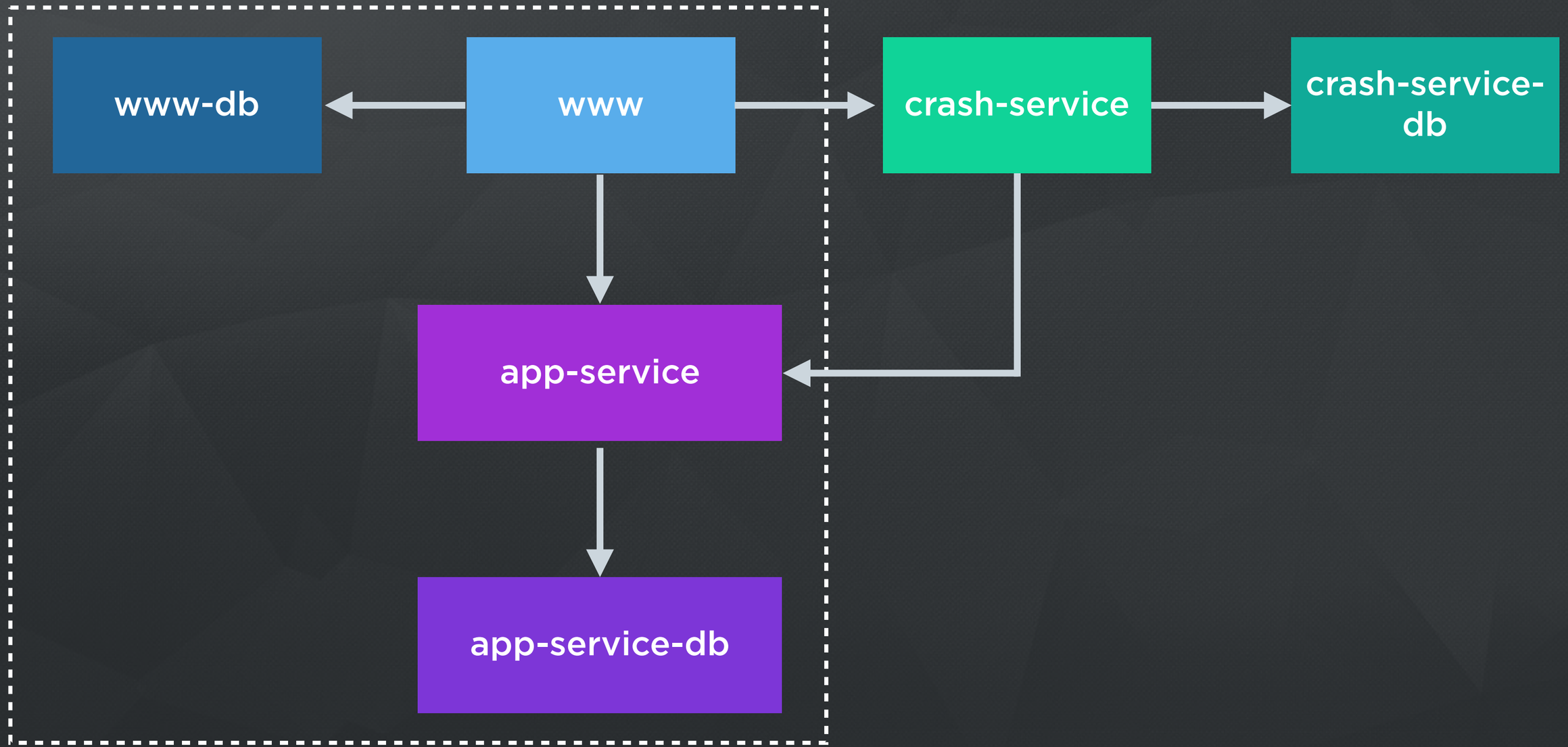


flight

Starting one Docker container is easy.

BUT STARTING ANY 12 THAT CAN TALK TO EACH OTHER CAN BE REPETITIVE

flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

flight

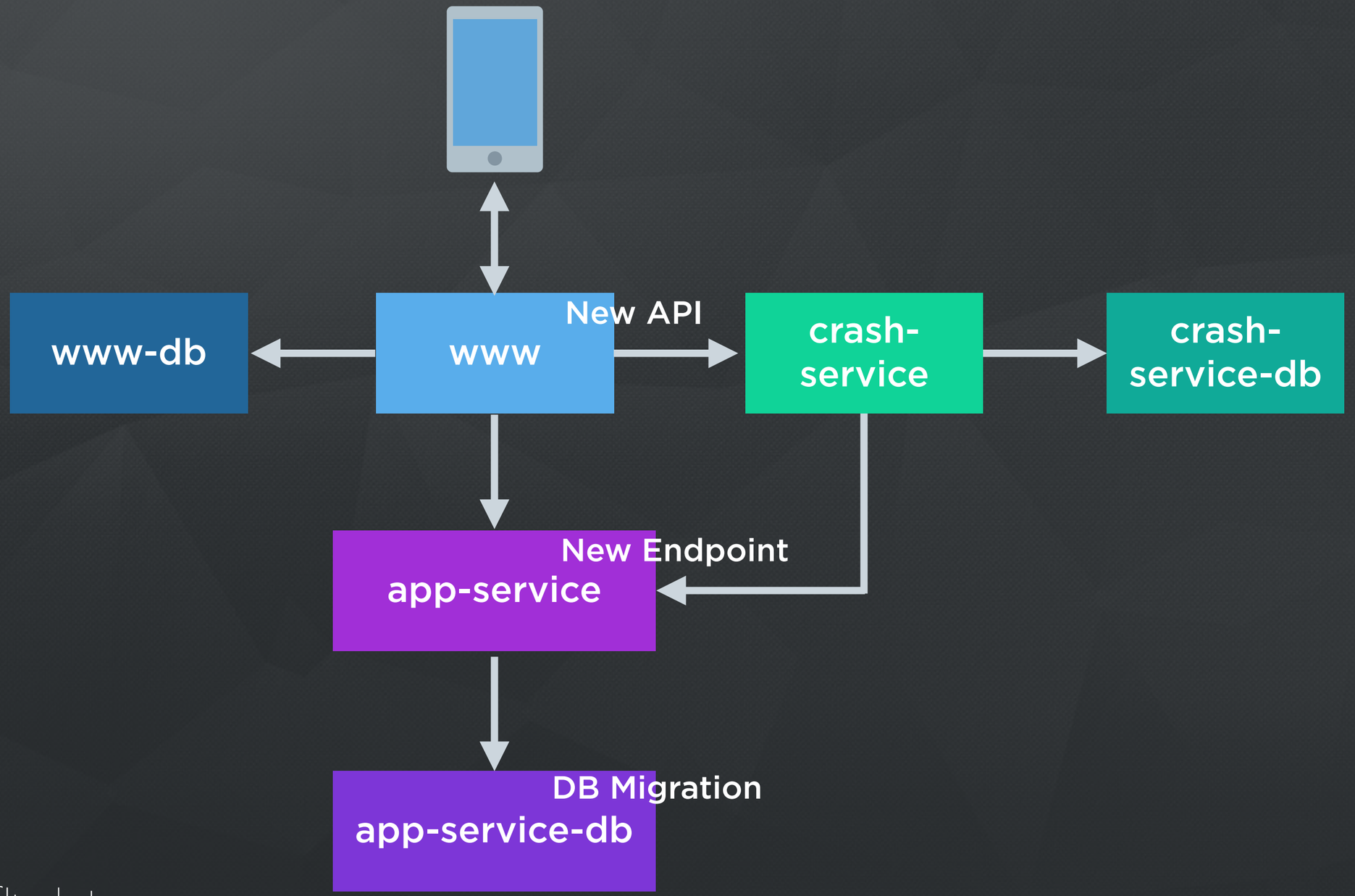
```
$ docker run app-service-db


$ docker run --link app-service—db:app-service-db \
   --port 3100:3000
   app-service


$ docker run www-db

$ docker run \
  --link app-service:app-service \
  --link www-db:www-db \
  --port 3000:3000 \
  --volume /webapp code/webapp \
  www
```

flight

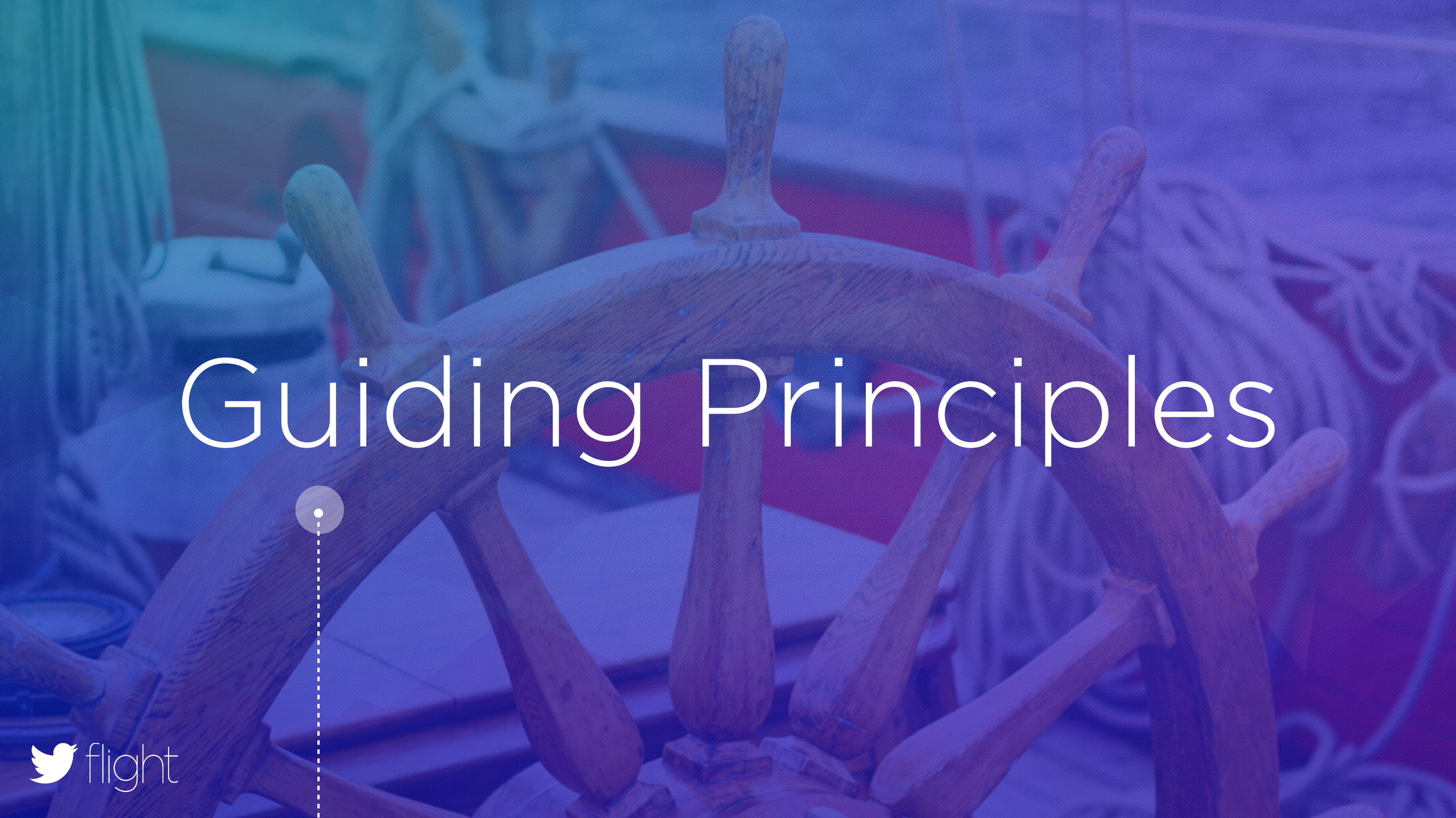
Let’s make it easy

```
$ my-tool run www.dev
```

So we built it


www-db
www
New API
crash-service
crash-service-db
app-service
New Endpoint
app-service-db
DB Migration

flight


Guiding Principles
flight

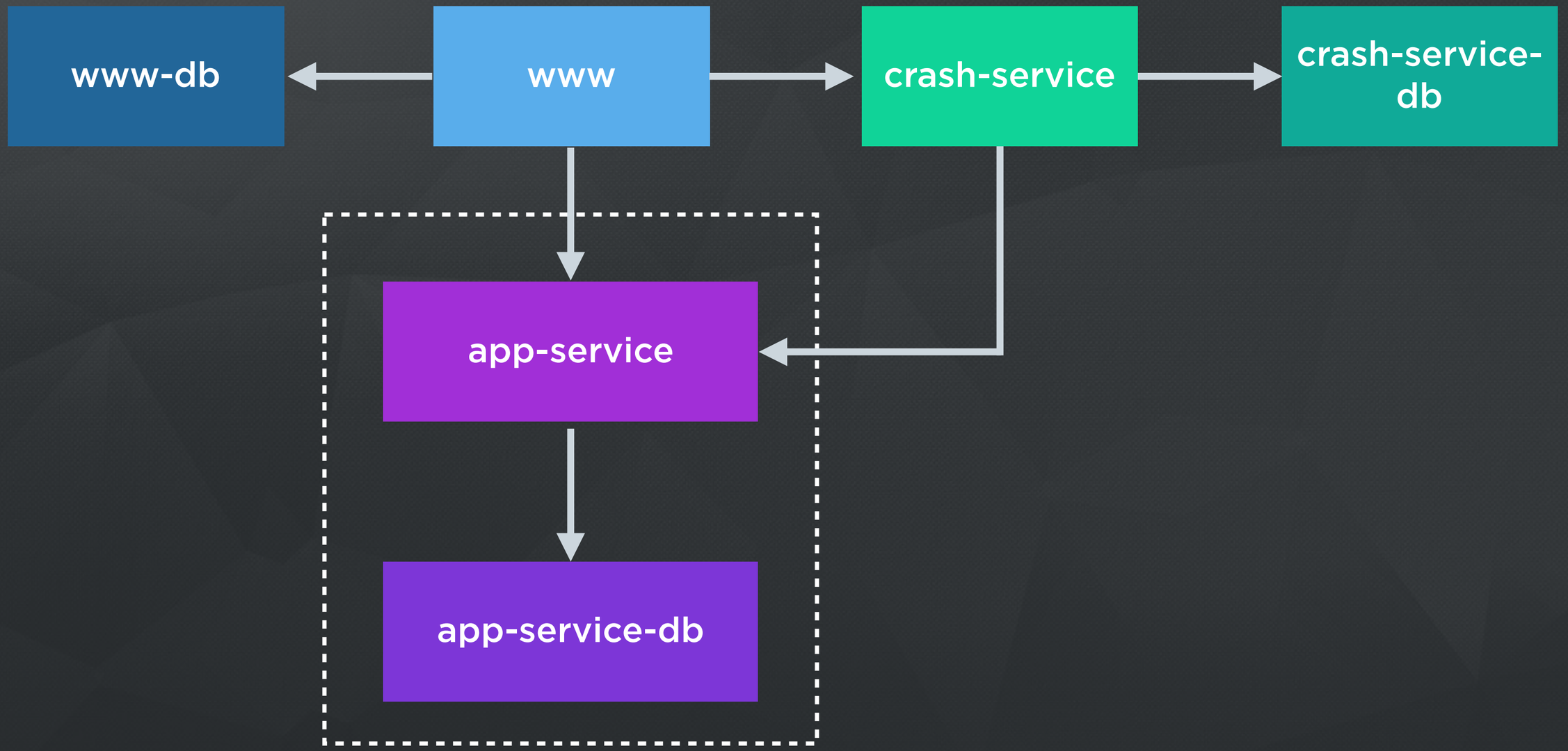
1

Graph of Services

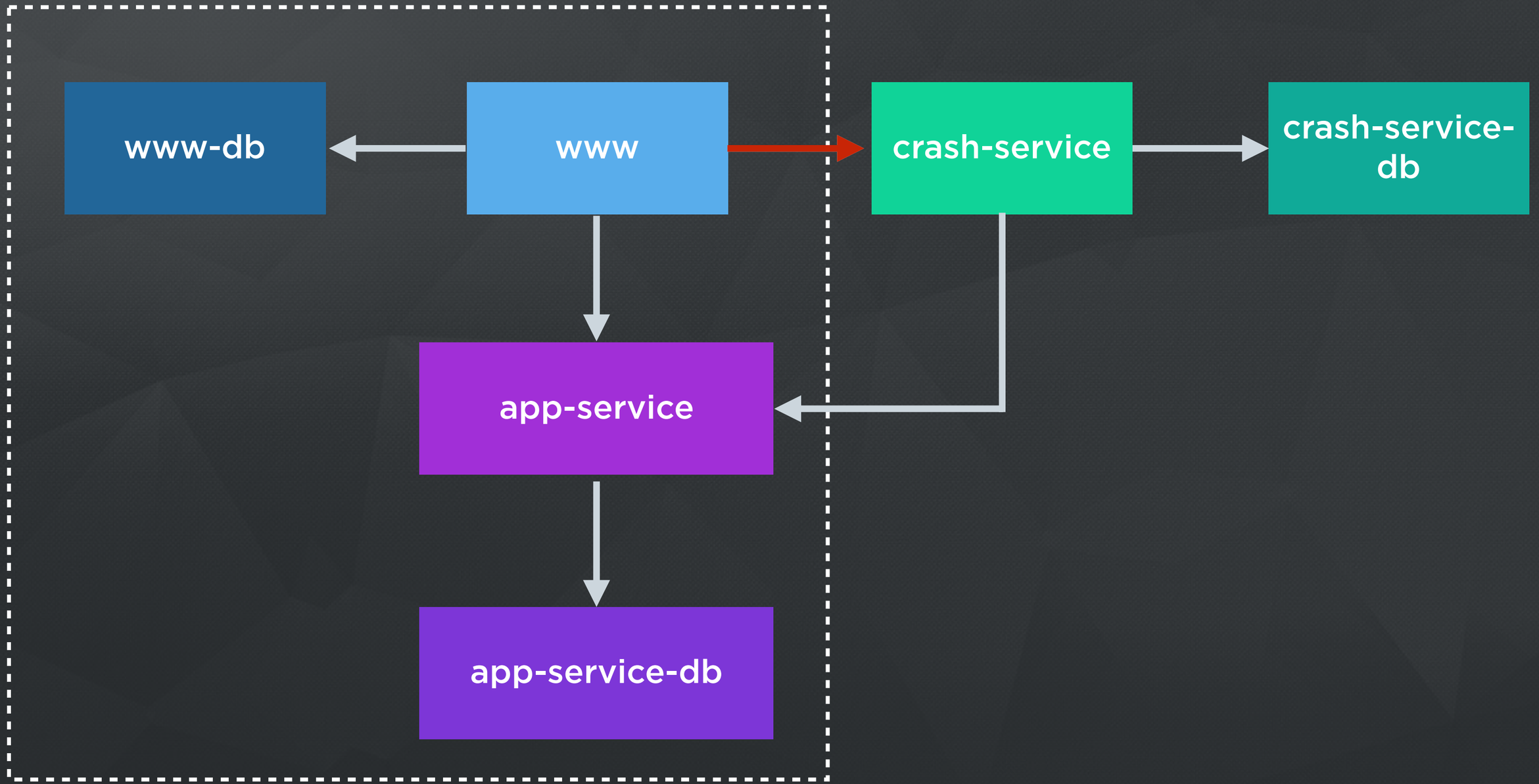
MANAGE TRANSITIVE DEPENDENCIES

START ANY SERVICE FROM LOCAL SOURCE OR MASTER

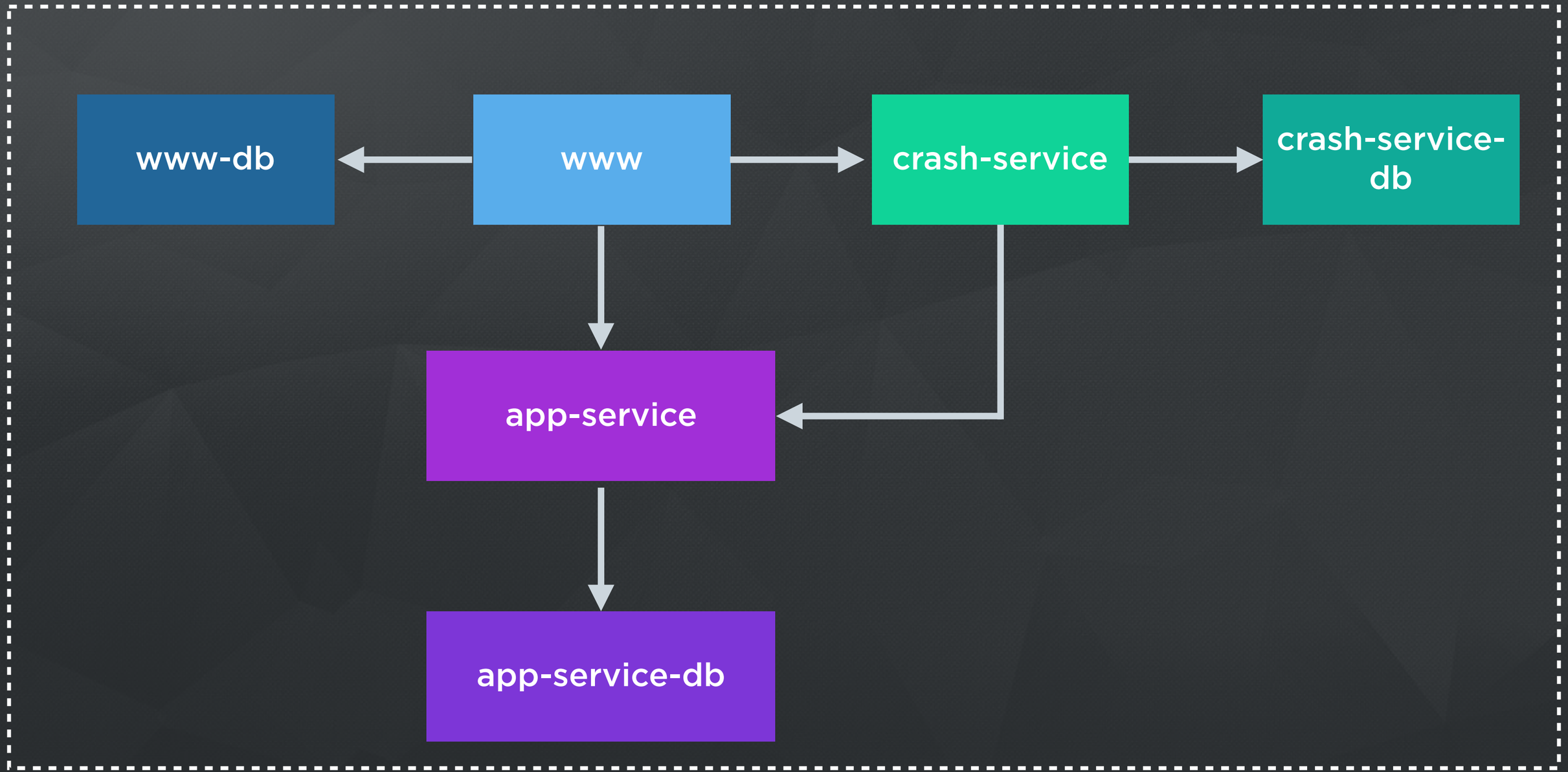
flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

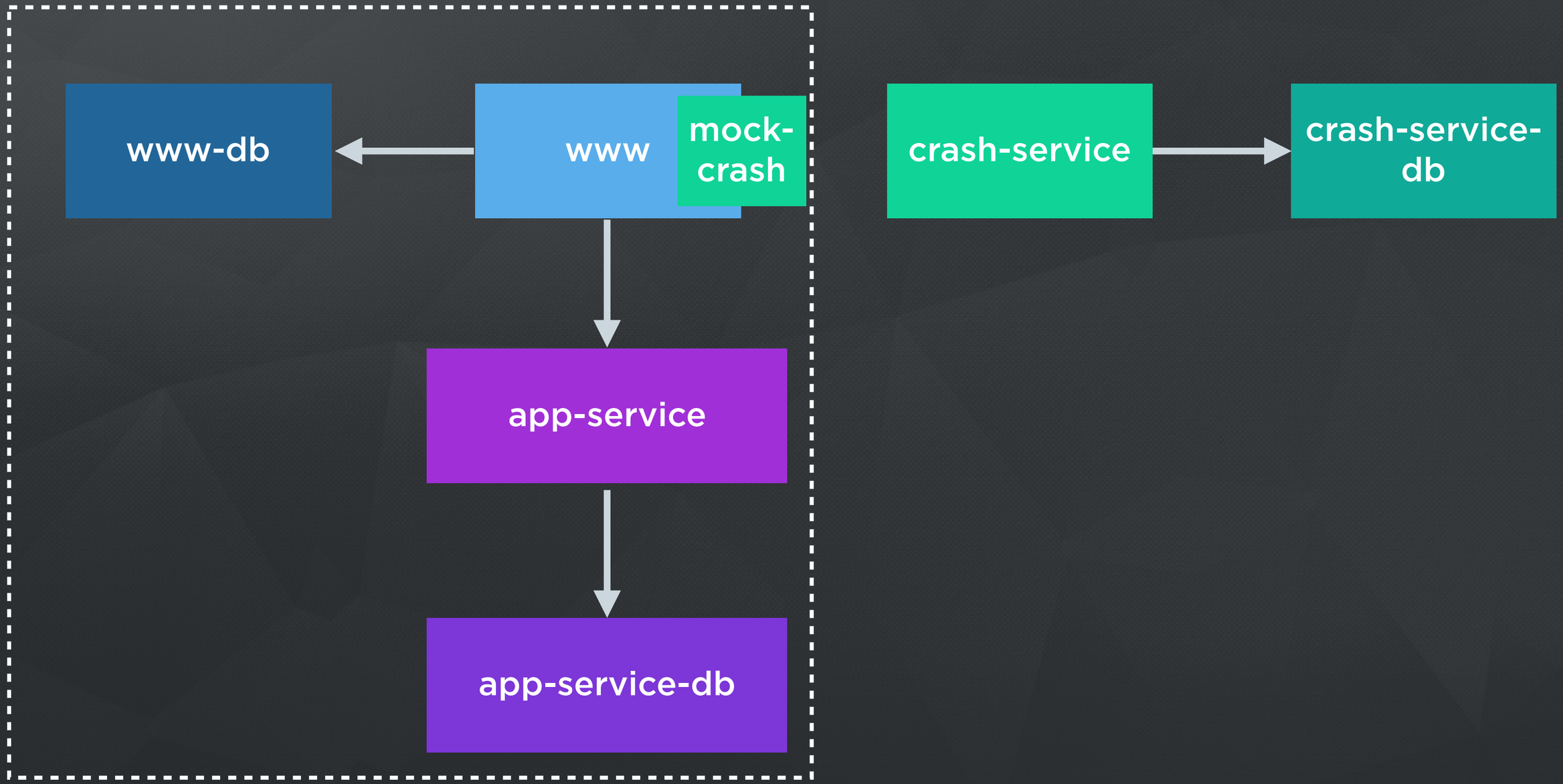
flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

flight


www-db
www
crash-service
crash-service-db
app-service
app-service-db

flight


www-db
www
mock-crash
crash-service
crash-service-db
app-service
app-service-db

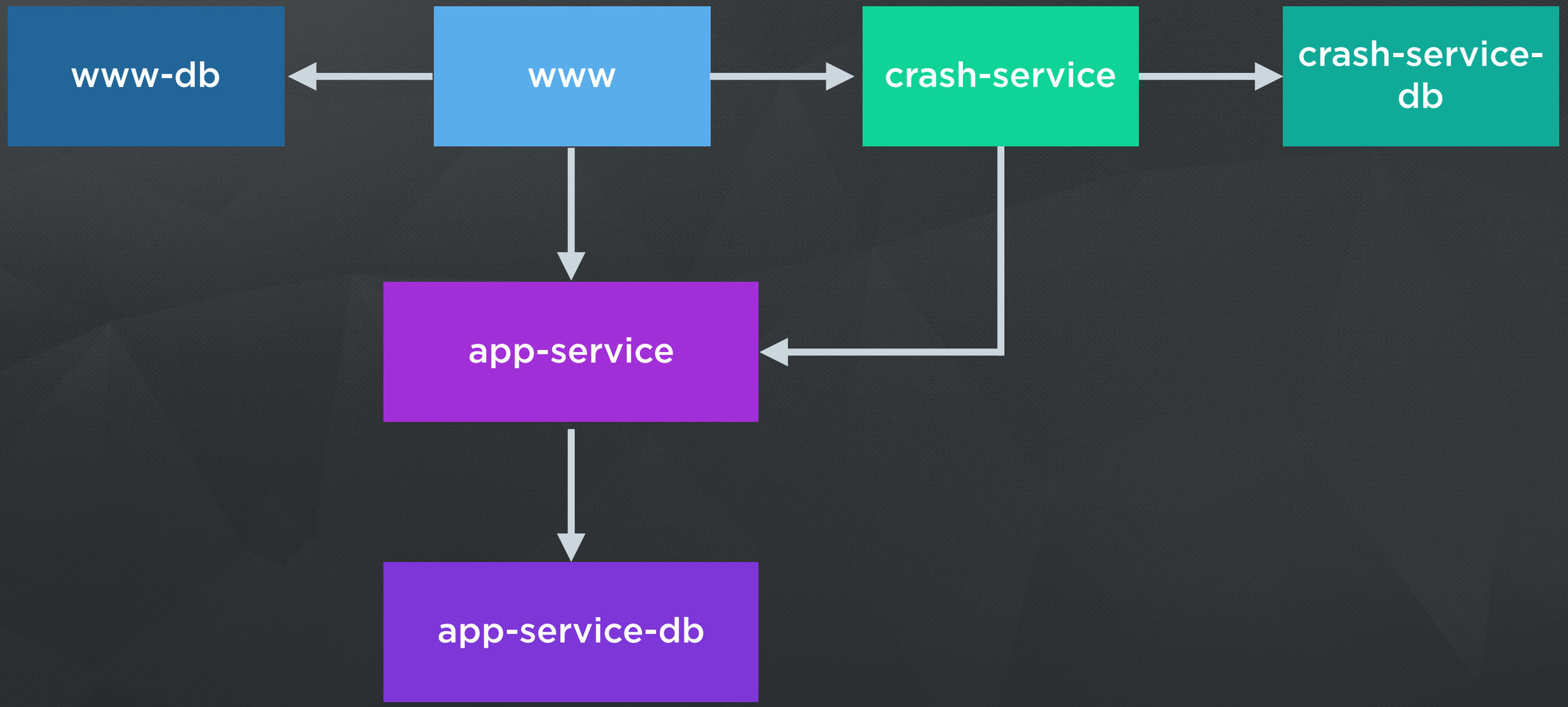
flight

2

Fast to Fix

CUT WASTED DEVELOPER TIME

flight

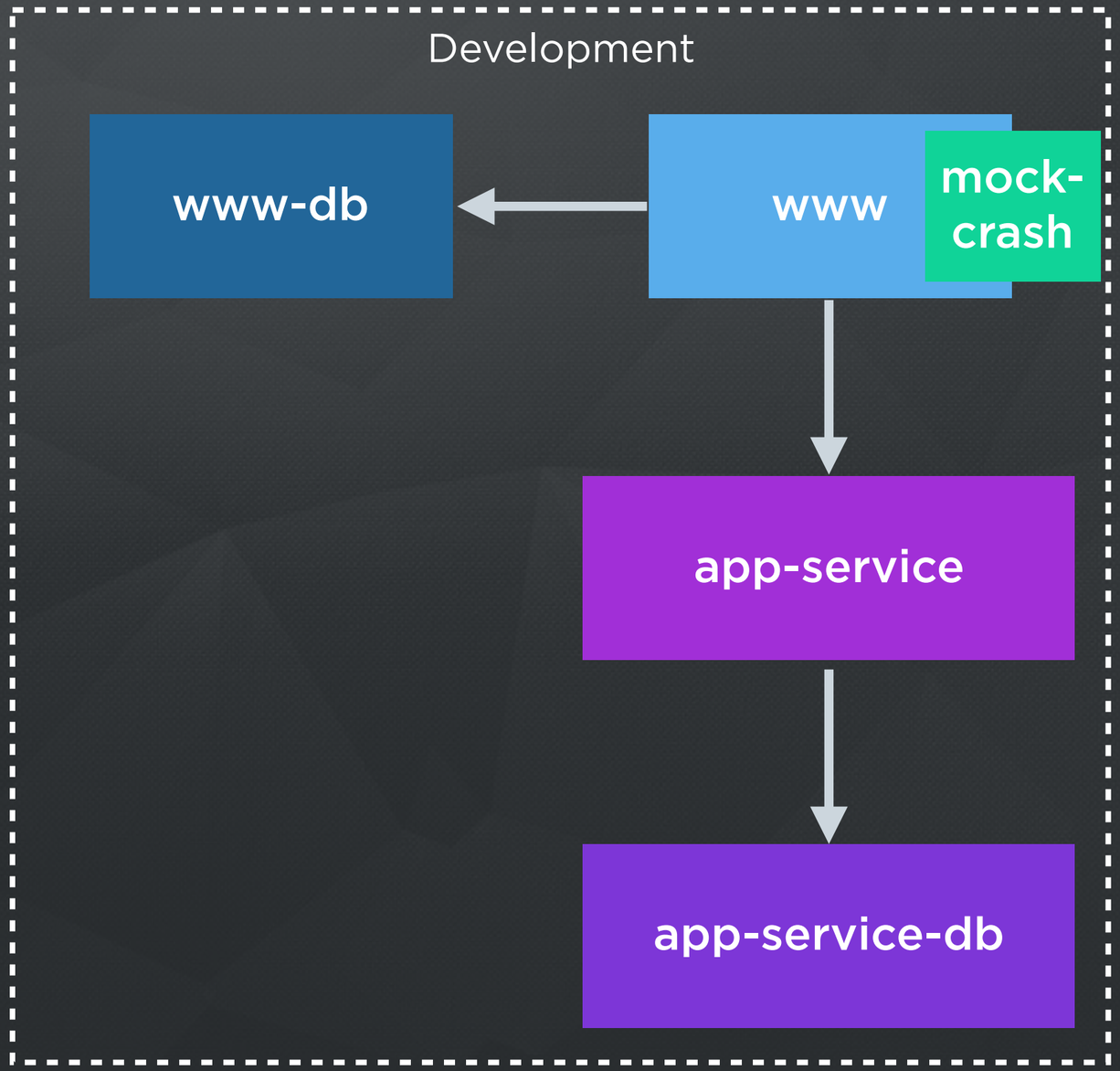

www-db
www
crash-service
crash-service-db
app-service
app-service-db

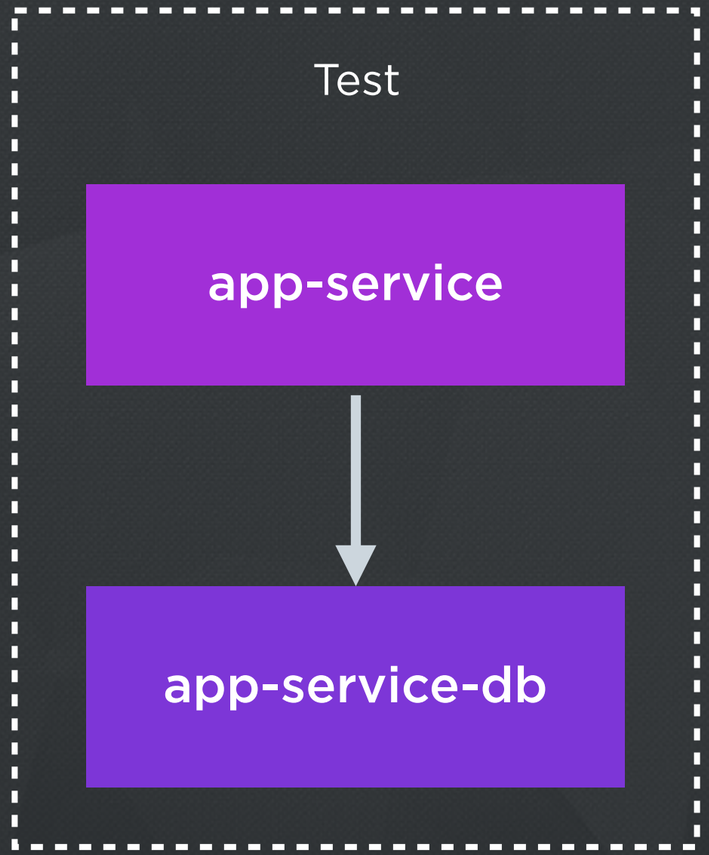
flight

3

Distinct Environments with Different Services

flight


Development
www-db
www
mock-crash
app-service
app-service-db


Test
app-service
app-service-db

flight

4

Repeatable

LAPTOP LOOKS LIKE CI

flight

5

Minimize Surprise

CONTAINERS CLOSER TO PROD THAN LAPTOP OR SHARED VM

flight

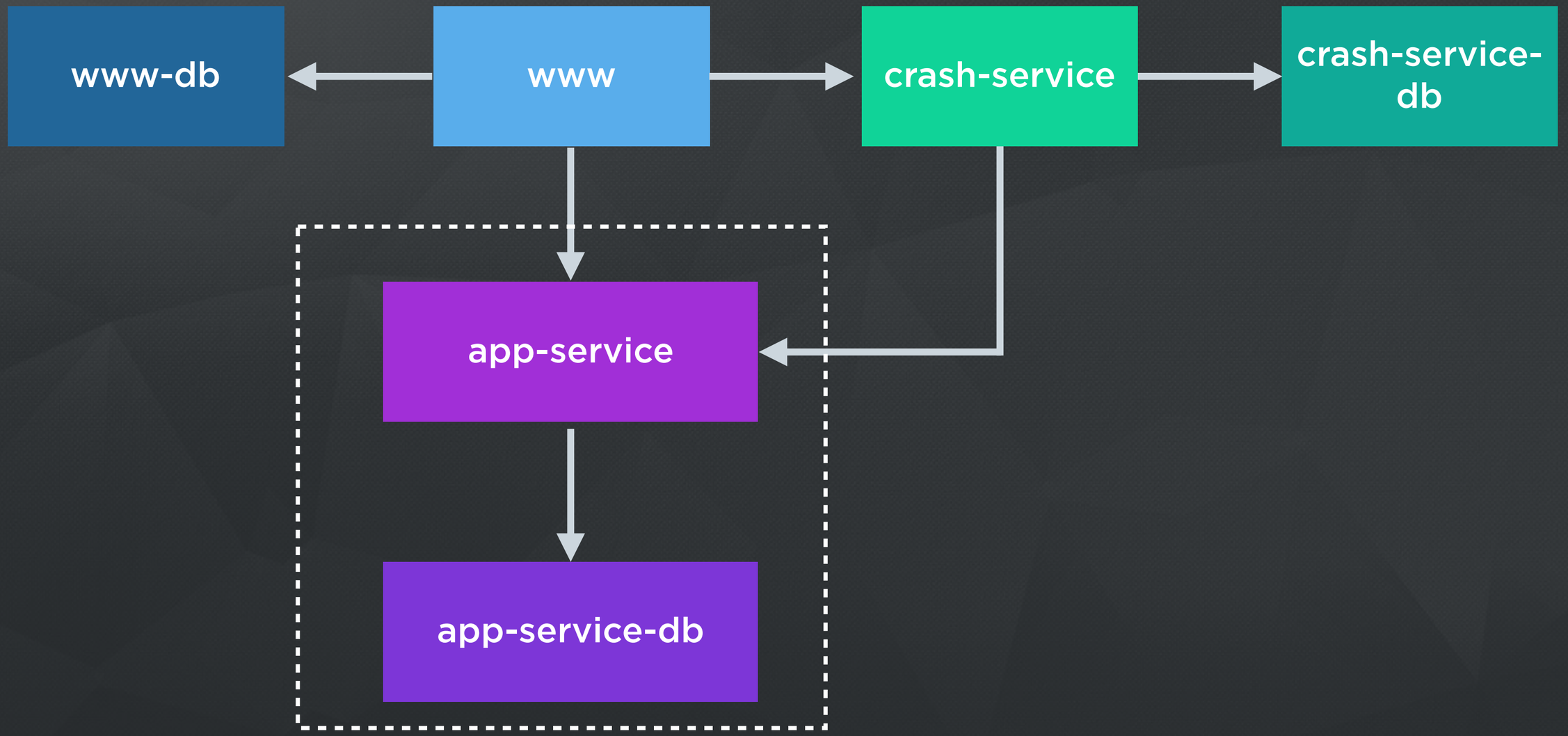
$ galley run www.dev

```
www-db: Checking… not found. Creating… done! Starting… done!
app-service-db: Checking… not found. Creating… done! Starting… done!
app—service: Checking… not found. Creating… done! Starting… done!
www: Checking… not found. Creating… done! Starting… done!
Resolving dependencies......
Using rake (10.3.2)
Using libxml-ruby (2.6.0)
Using i18n (0.6.9)
…
>> Thin web server (v1.5.1 codename Straight Razor)
>> Maximum connections set to 1024
>> Listening on 0.0.0.0:3000, CTRL+C to stop
```

NodeJS CLI

Galleyfile.js describes your architecture

flight

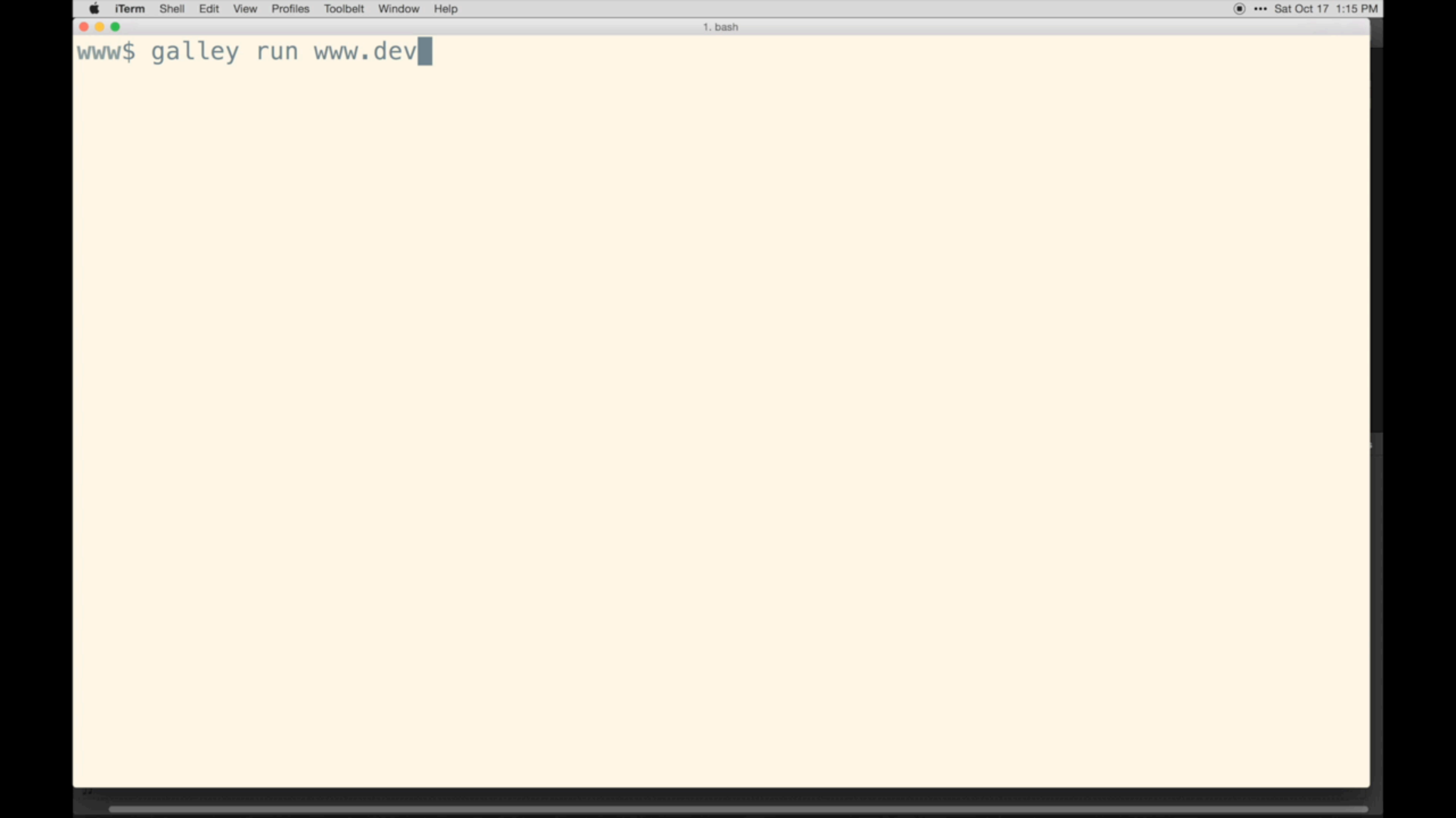

www-db
www
crash-service
crash-service-db
app-service
app-service-db

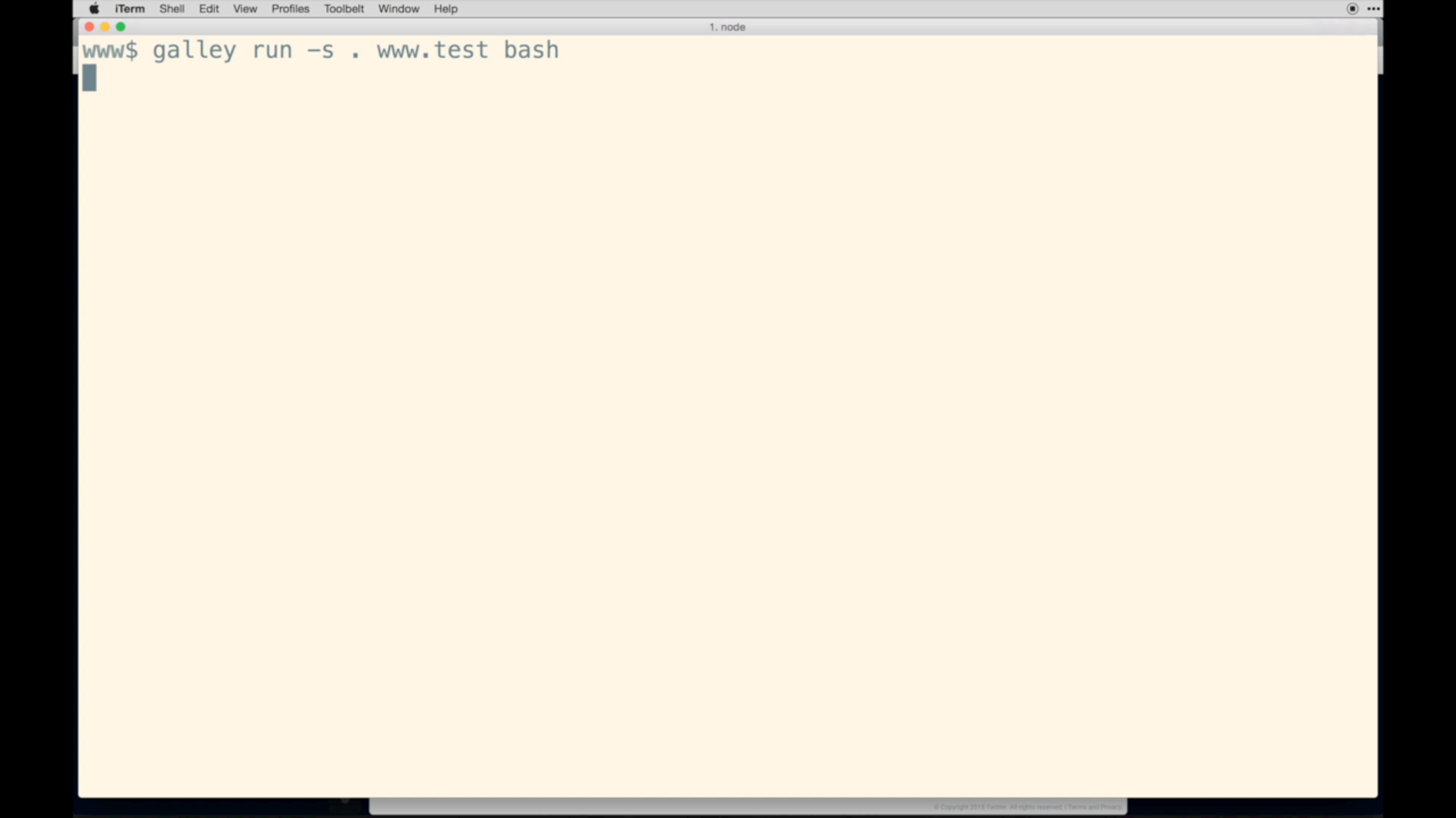
flight

```
module.exports = {
  'app-service': {
    'source': '/srv/app-service',
    'ports': ['3200:3000'],
    'links': ['app-service-db'],
    'env': {
      'NODE_ENV': DEV_TEST_ENV
    }
  },
  'app-service-db': {
    'image': 'mongo',
    'stateful': true
  }
  …
}
```

flight

See it in Action


iTerm Shell Edit View Profiles Toolbelt Window Help
Sat Oct 17 1:15 PM
1. bash
www$ galley run www.dev


iTerm Shell Edit View Profiles Toolbelt Window Help
1. node
www$ galley run -s . www.test bash

Galley Commands

- `$ galley run`
- `$ galley pull`
- `$ galley stop-env`
- `$ galley cleanup`

Fun Galley Bonuses

REAL-WORLD ADAPTATIONS

flight

Source For A Service Has A Fixed Directory

Before:

```
$ docker run --link app-service—db:app-service-db \
    --port 3100:3000
    --volume /Users/code/app-service:/srv/app-service \
    app-service
```

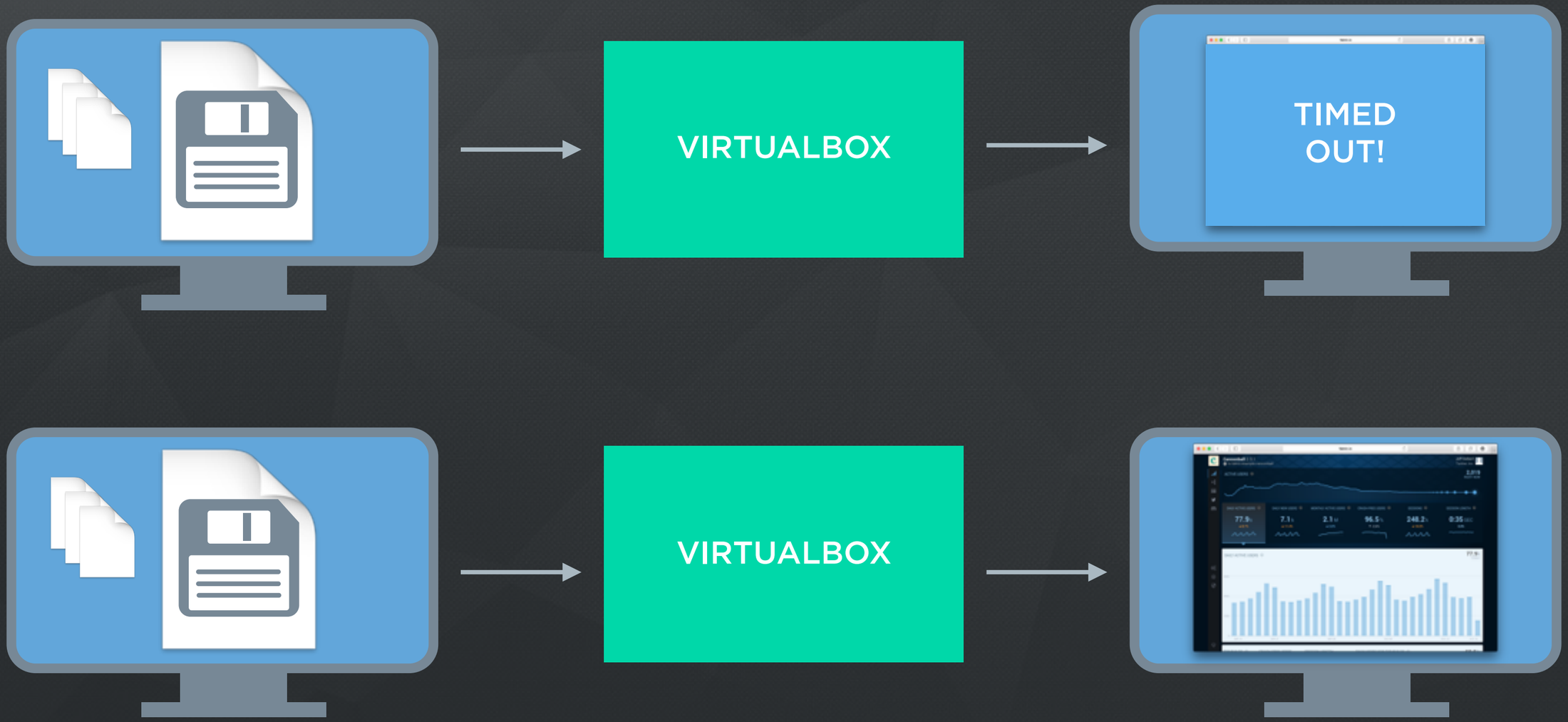
With Galley:

```
'app-service': {
  'source': '/srv/app-service',
  'links': ['app-service-db'],
  'volumesFrom': ['srv-config'],
  'ports': ['3200:3000'],
  'env': DEV_TEST_ENV
}
```

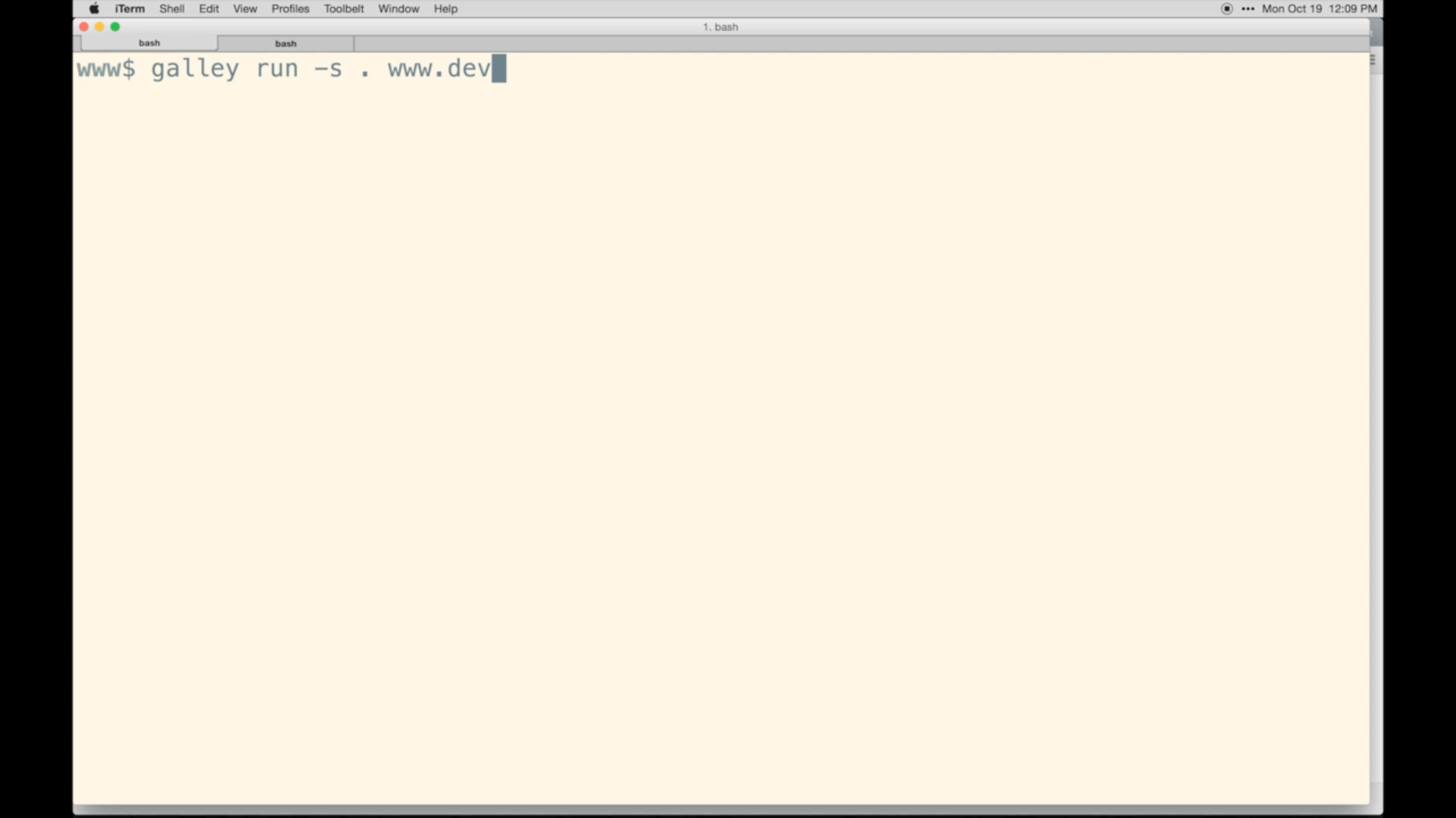
```
$ pwd
/Users/code/app-service

$ galley run -s . app-service
```

flight

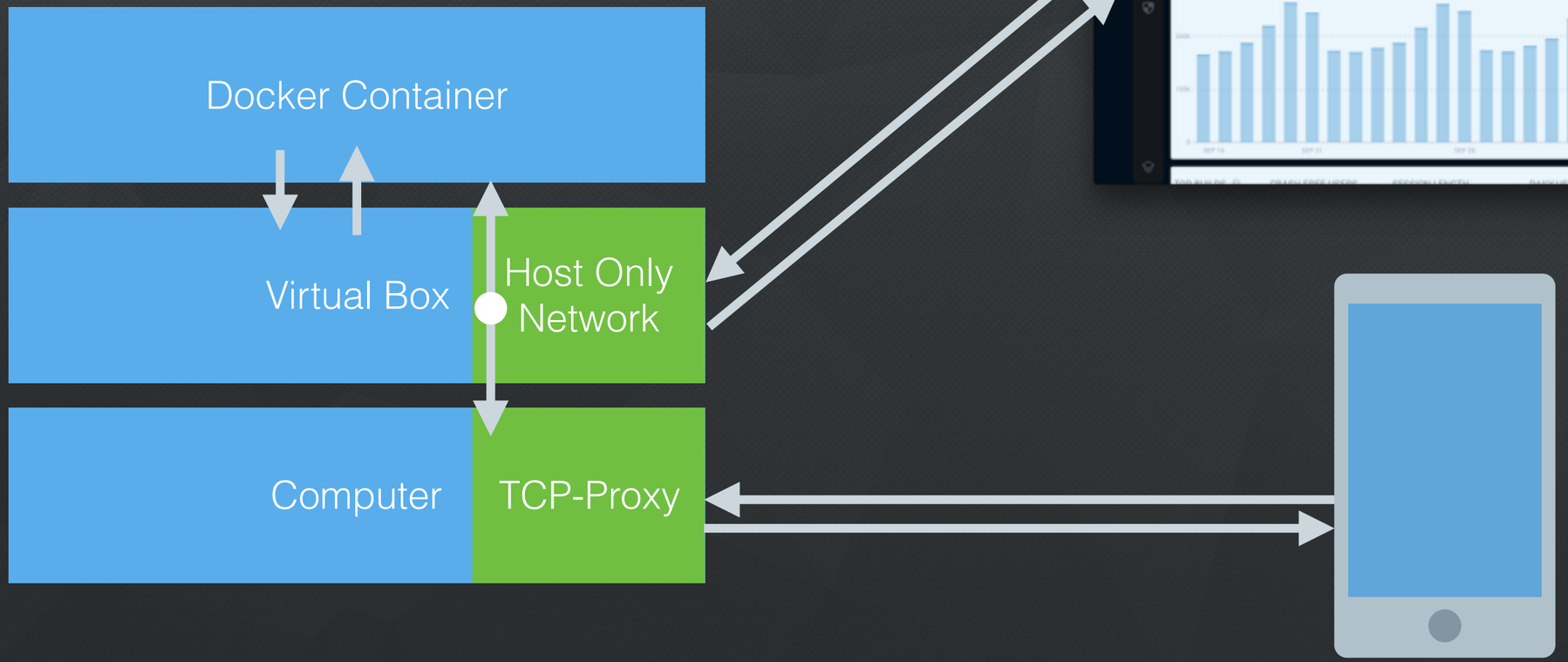
Development On A Mac?



flight

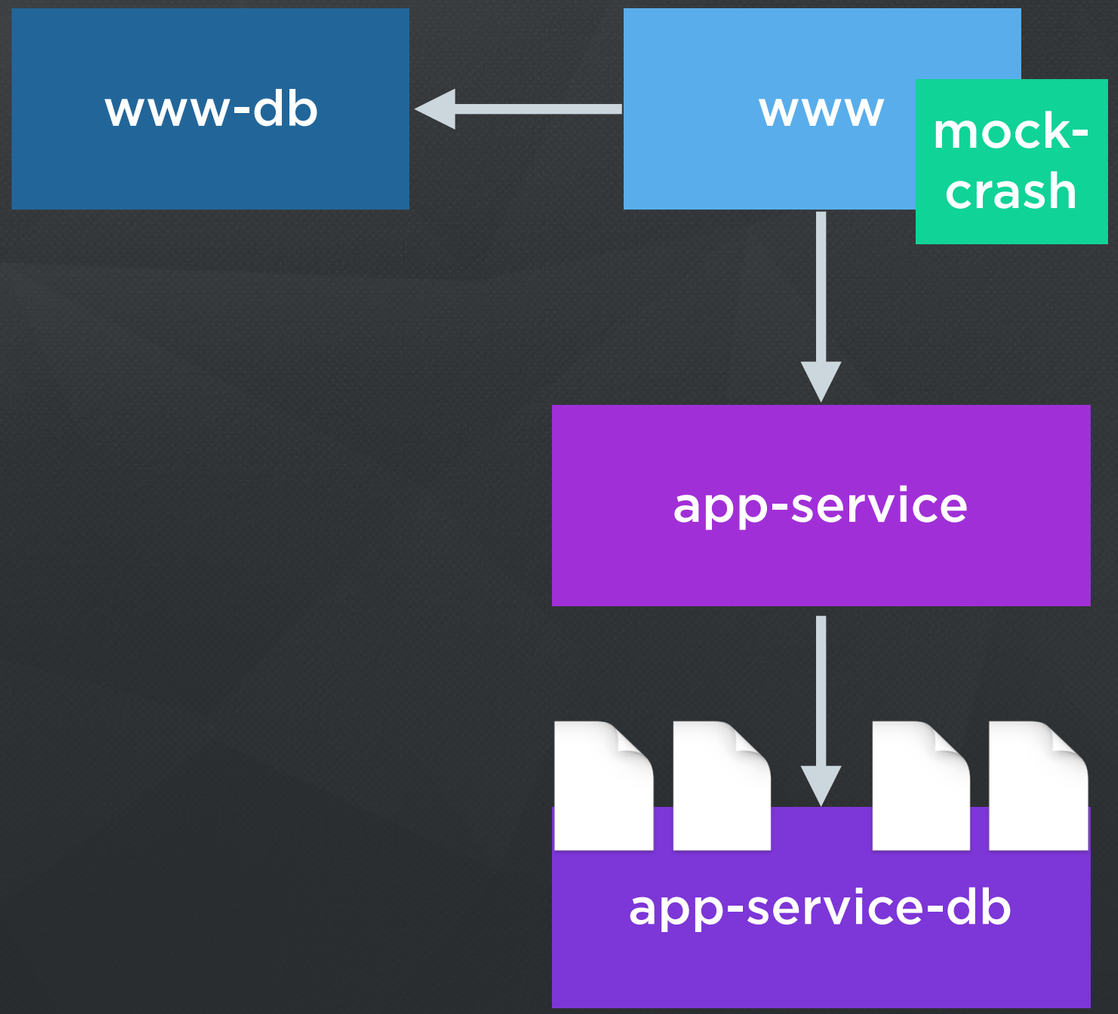

iTerm Shell Edit View Profiles Toolbelt Window Help
Mon Oct 19 12:09 PM
1. bash
bash
bash
www$ galley run -s . www.dev

Testing On Your Phone

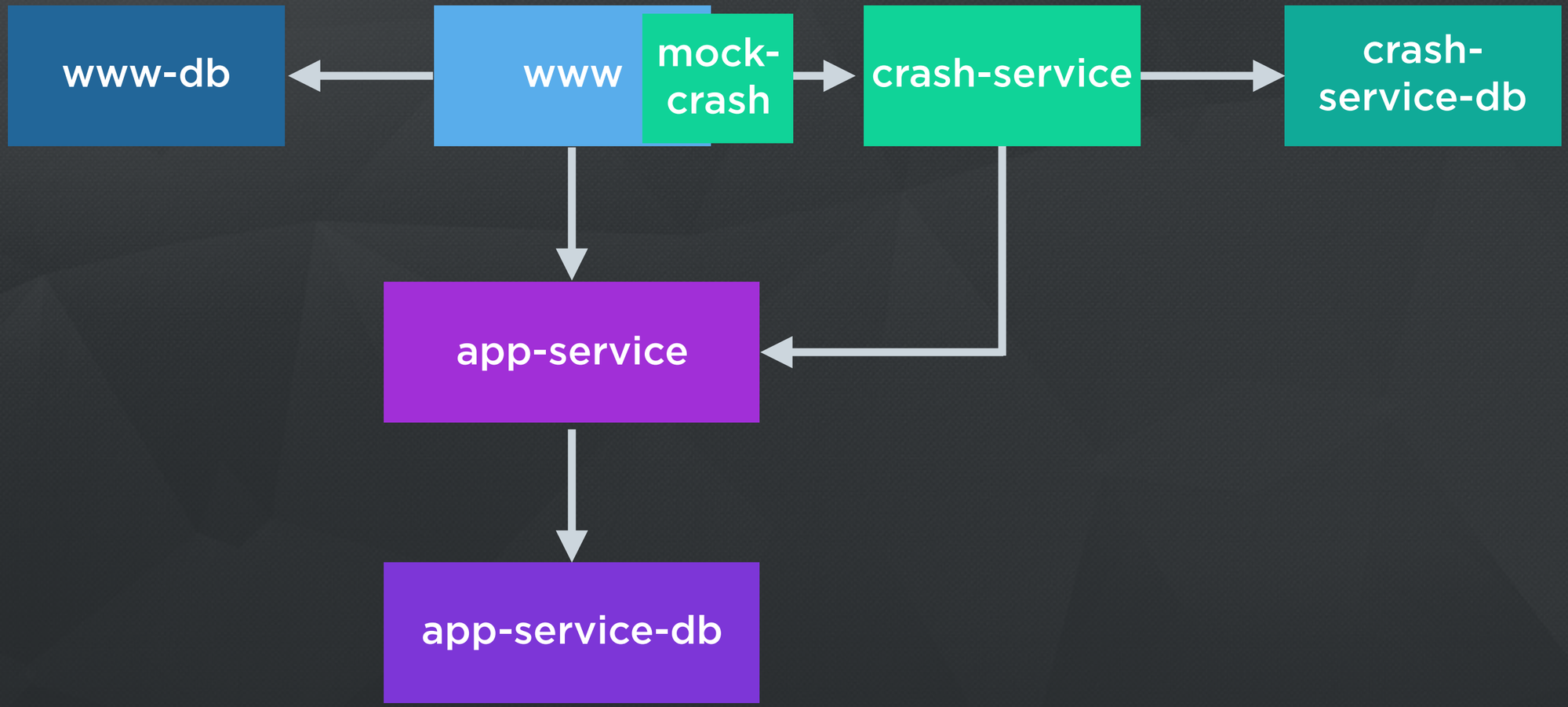


flight

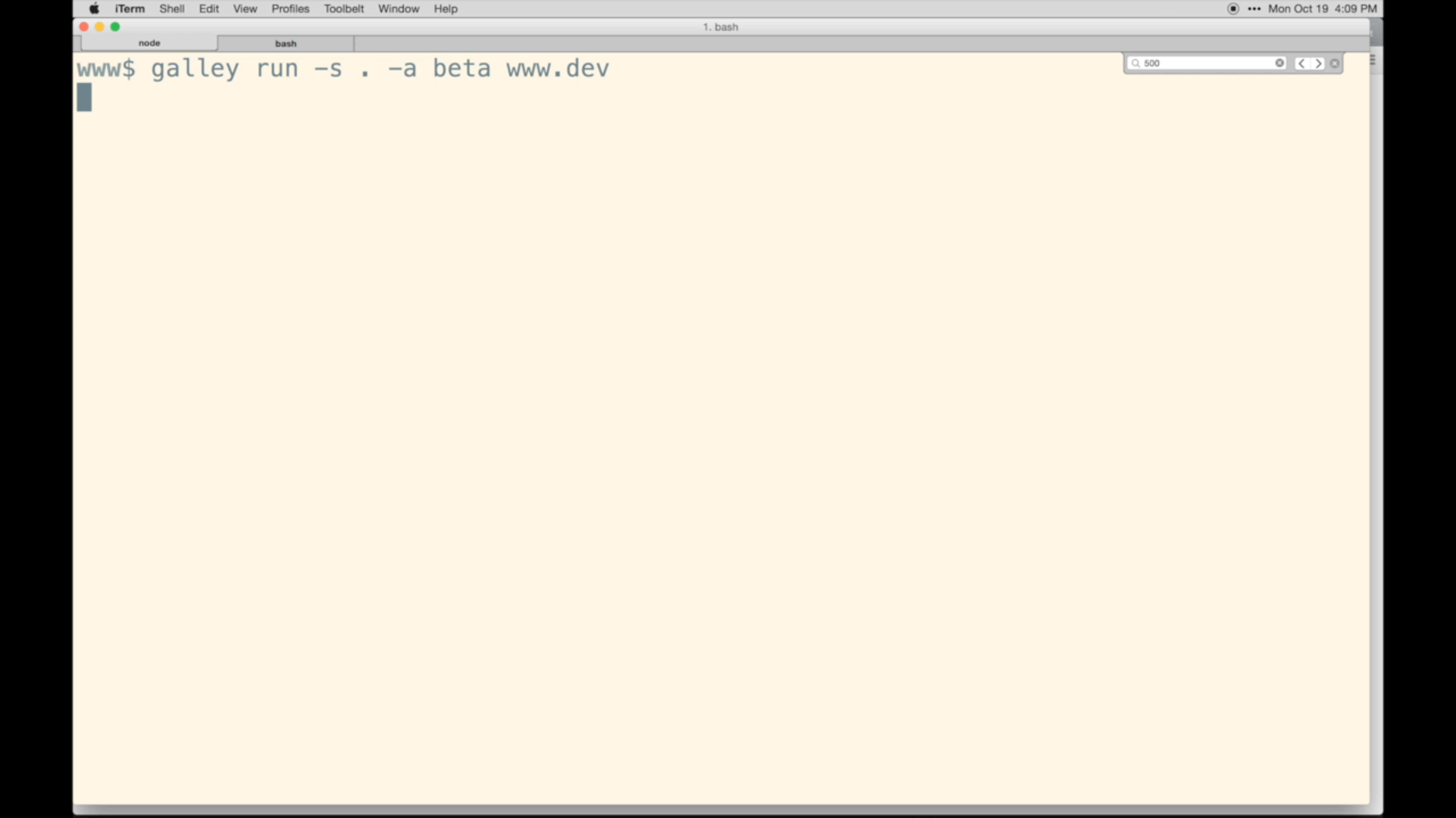
Stateful Containers



Addons



flight


iTerm Shell Edit View Profiles Toolbelt Window Help
Mon Oct 19 4:09 PM
1. bash
node
bash
500
www$ galley run -s . -a beta www.dev

Open Source

HTTPS://T.CO/GALLEY

flight

How To Get Started

- Install Galley

  ```
  $ npm install -g galley-cli
  ```

- Clone the template repo

  ```
  $ git clone git@github.com:twitter-fabric/galley-template.git
  ```

- Start describing your architecture

Thank You

@joans