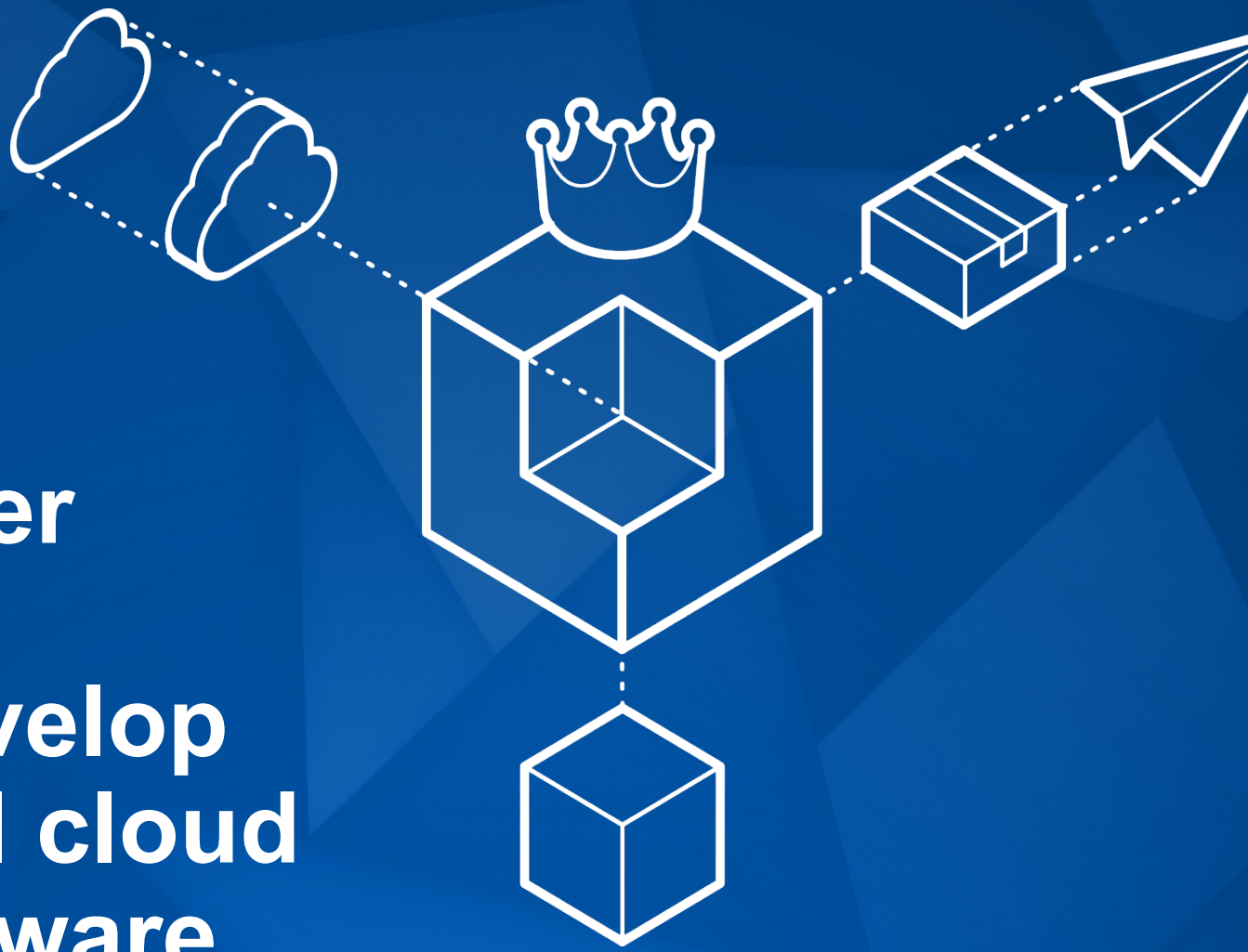


API billioner

How to develop successful cloud native software

Potr Bochyński

SAP

(v)

Agenda

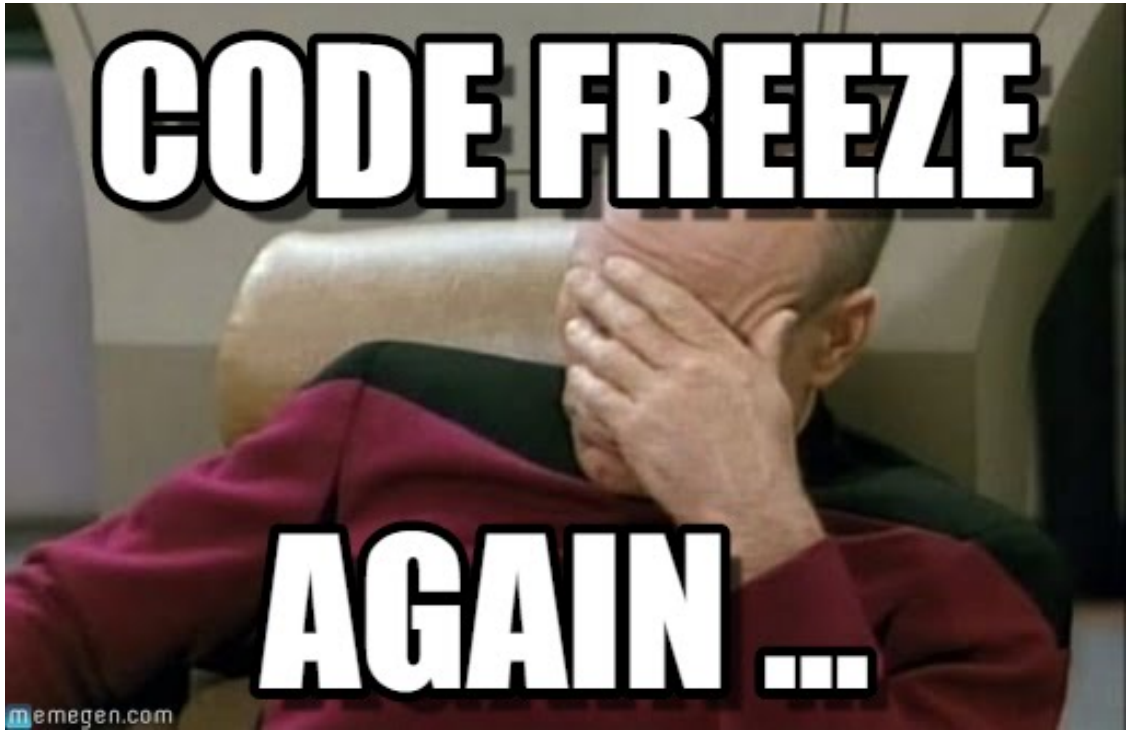
- DevOps - what is that?
- infrastructure as a code
- continuous integration, continuous delivery, continuous deployment
- health check and monitoring
- Zero Downtime deployment

Fear of Change

Function freeze

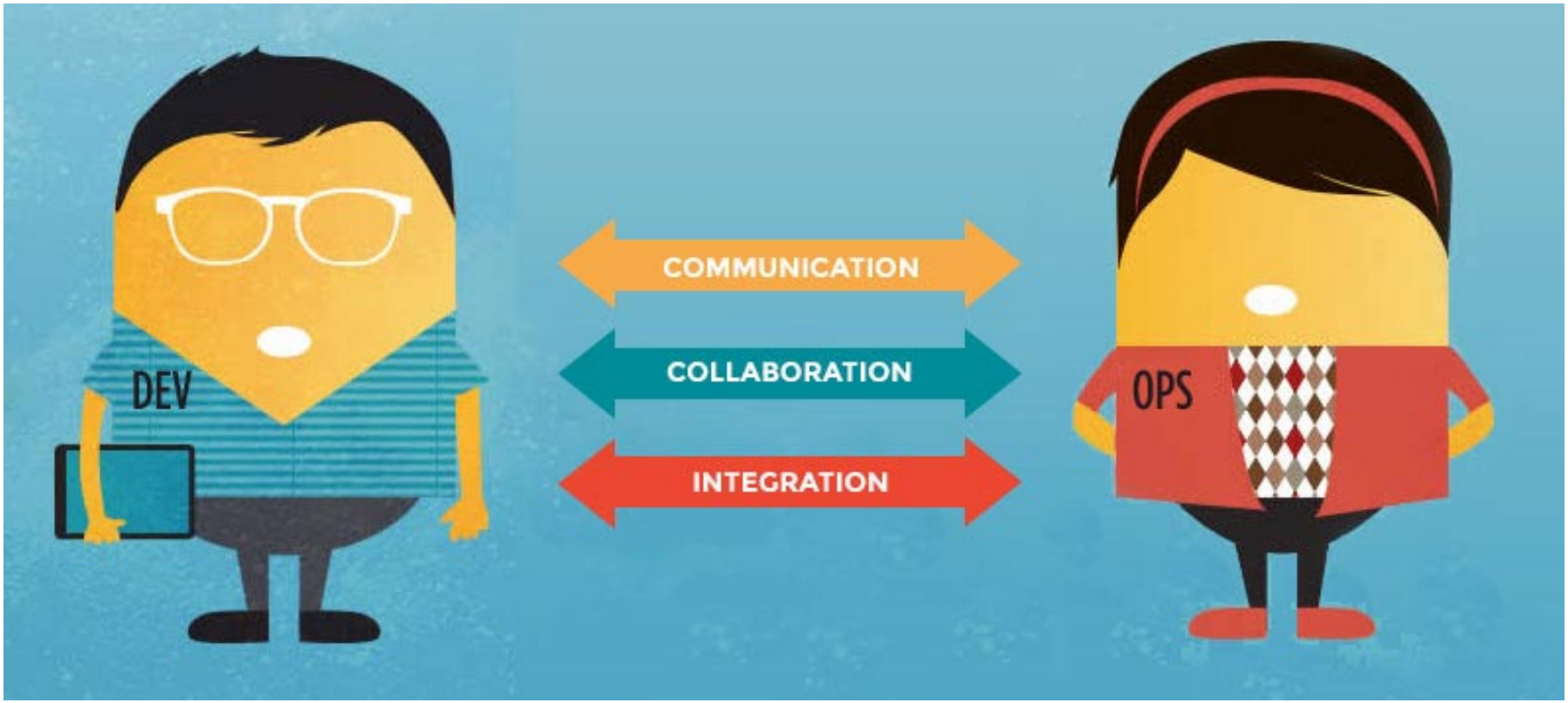
Code freeze

Do not touch this!



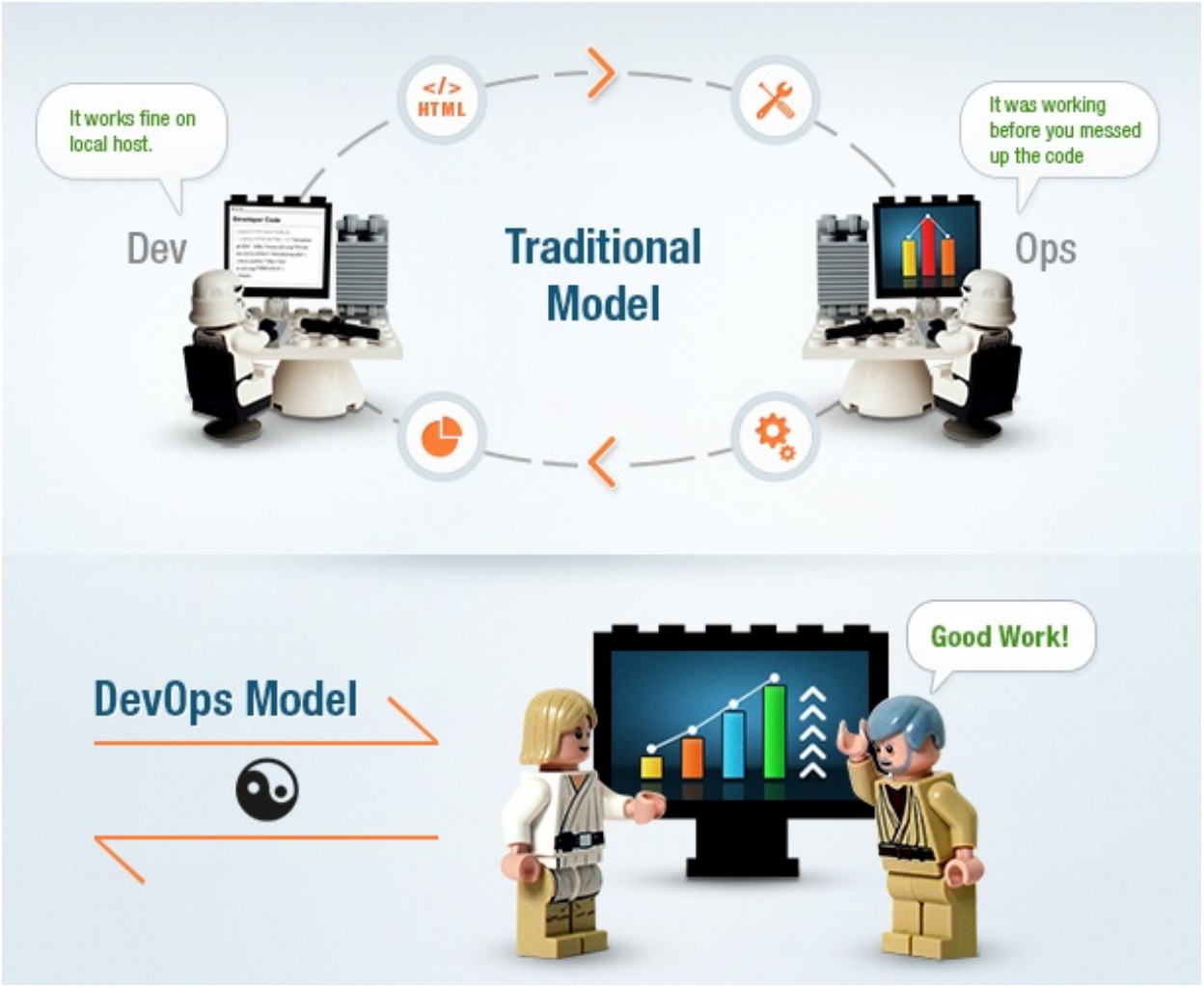
Risky Deployments

DevOps

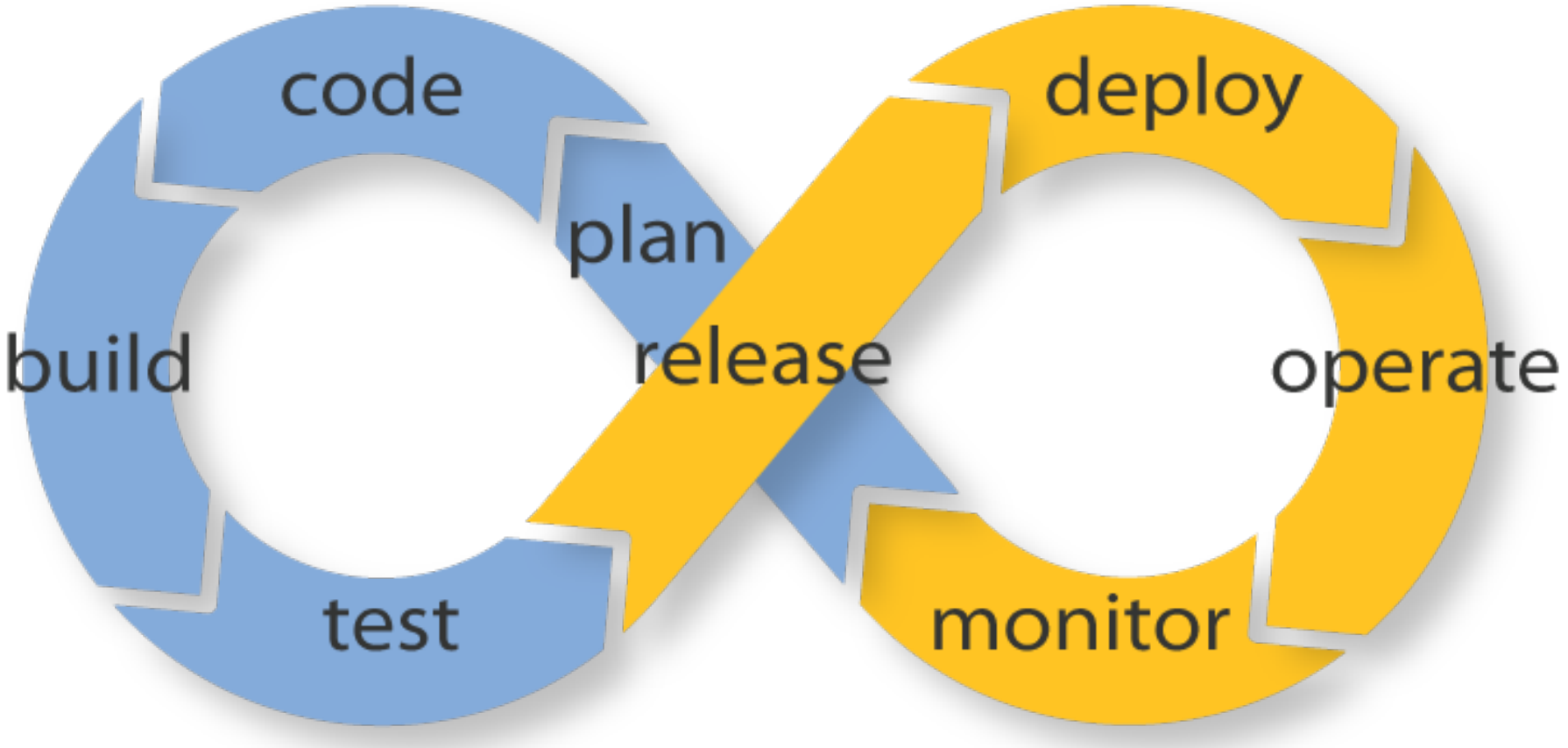


source: http://newrelic.com/devops

DevOps


It works fine on local host.
</> HTML
It was working before you messed up the code
Dev
Traditional Model
Ops
DevOps Model
Good Work!

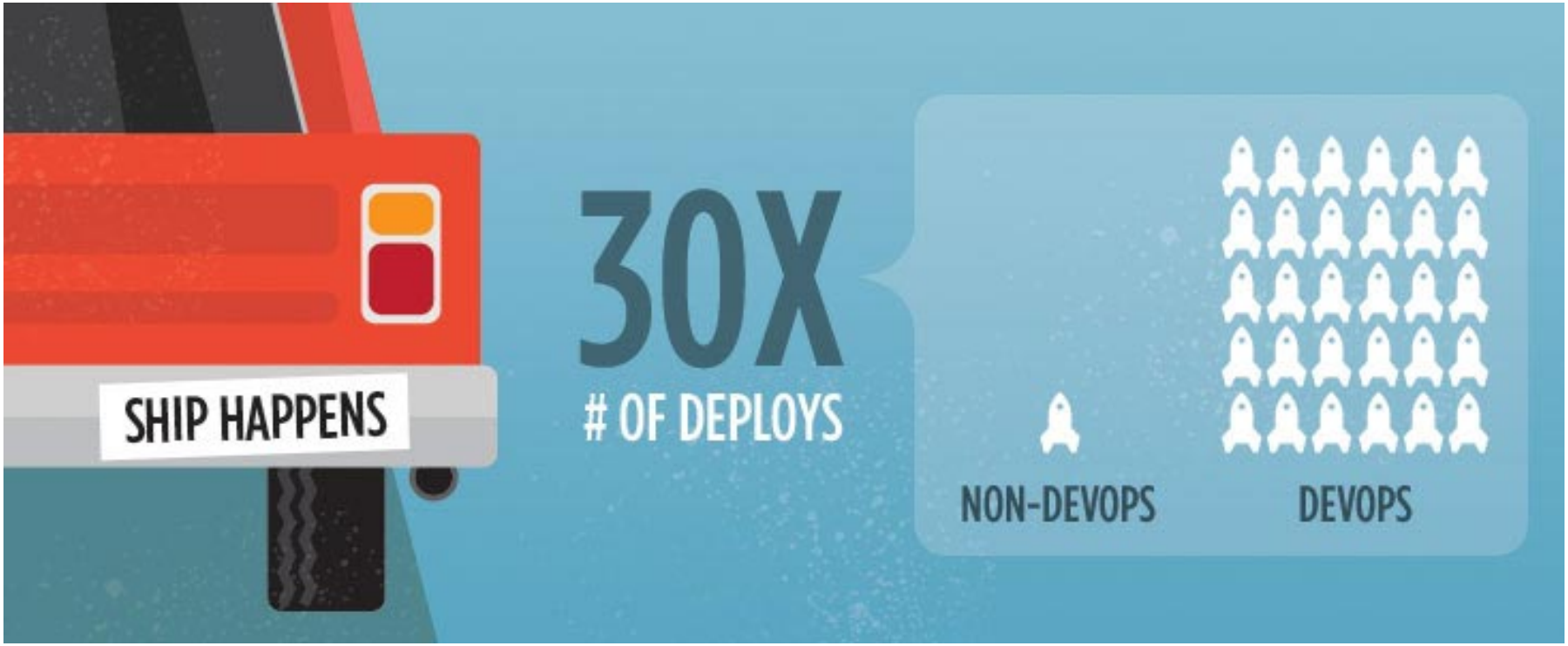
DevOps



Endless Possibilities: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

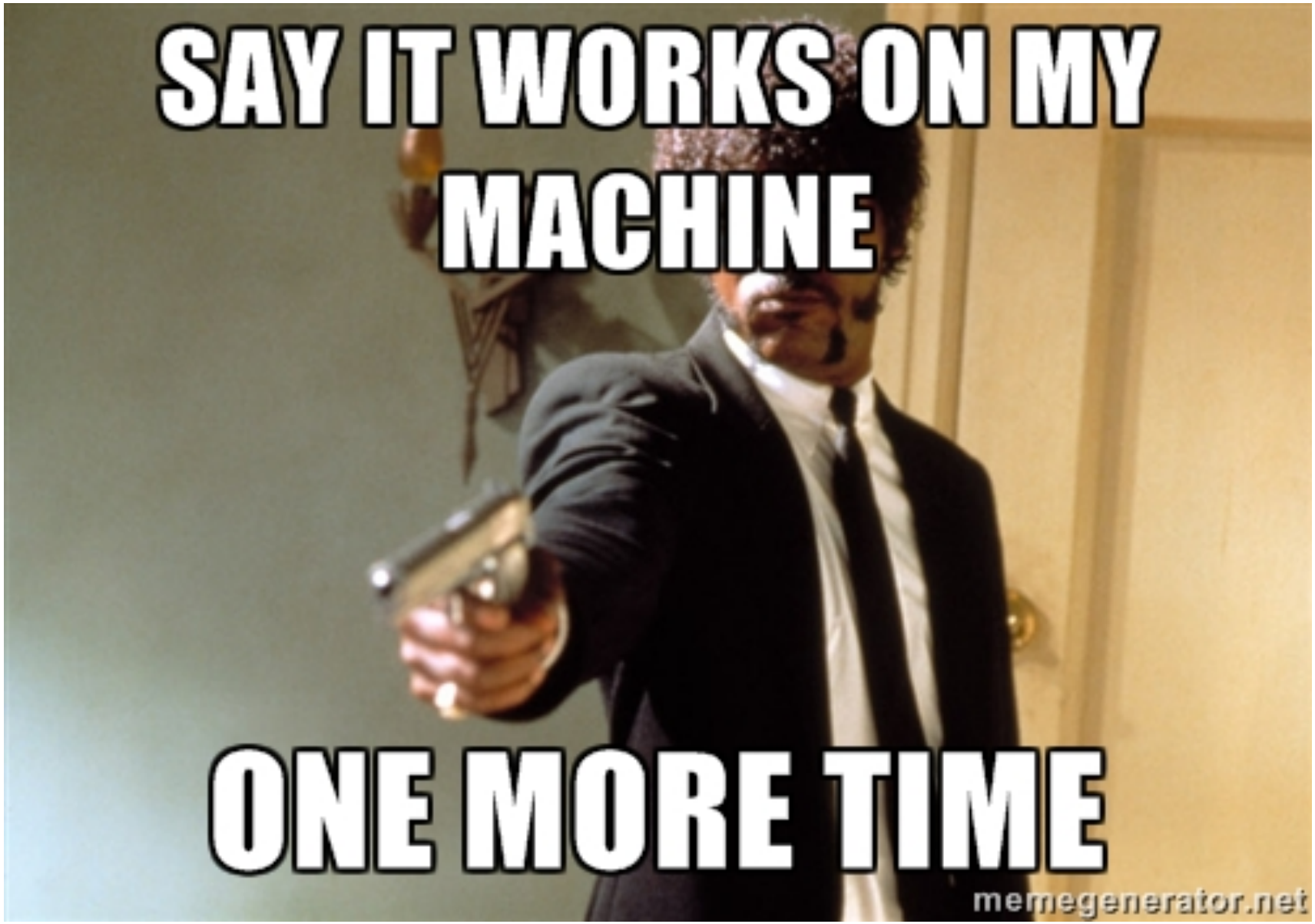
DevOps benefits

More deploys means faster time-to-market and continual improvement.



Frequent releases

DevOps



Project Setup

Database (MongoDB, Postgress, Cassandra)

Cache (Redis)

Message Queue (RabbitMQ, Kafka)

Service Directory (Zookeeper, etcd)

Application servers (Tomcat, JBoss)

CI server (Jenkins, TeamCity)

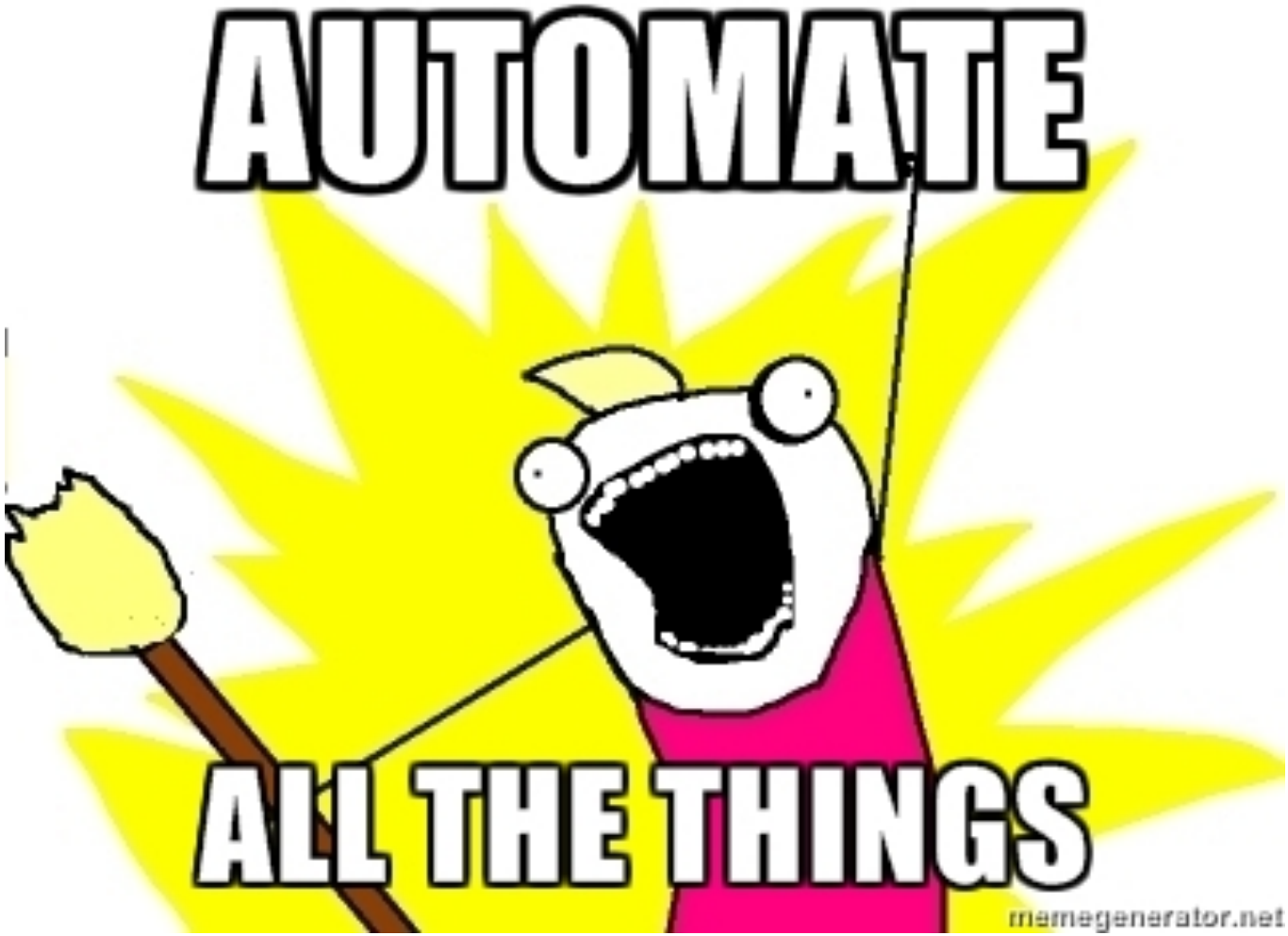
etc….

Infrastructure as a Code

Setup of machines is:

- **time consuming**
- **error prone**
- **time consuming**
- **repetitive**
- **did I mention time consuming?**
- **and BORING!**

Infrastructure as a Code



Infrastructure as a Code

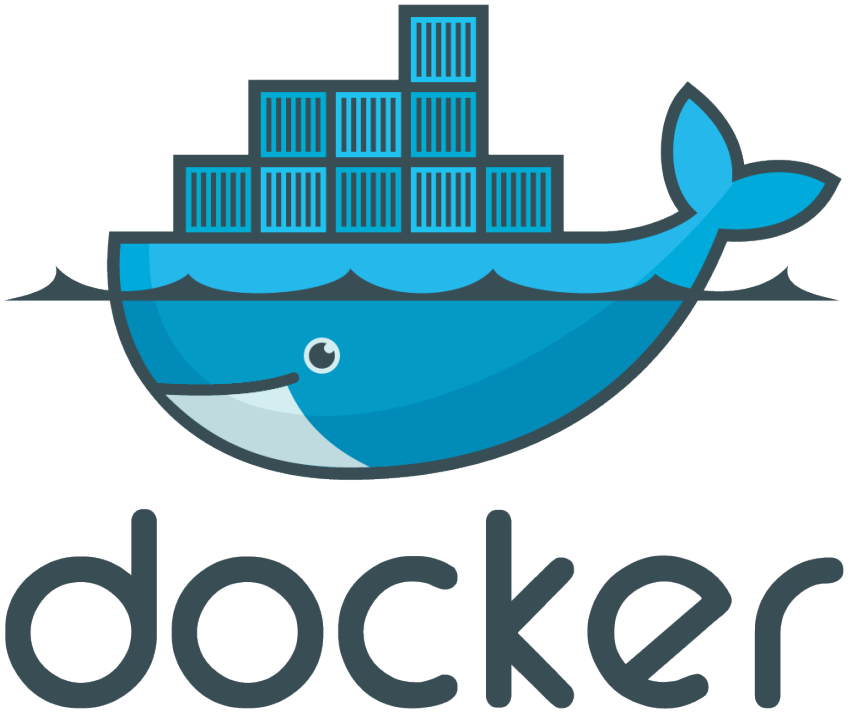
EXAMPLES:

- **ANSIBLE**
- **VAGRANT**
- **CHEF**
- **PUPPET**
- **DOCKER**

SAP

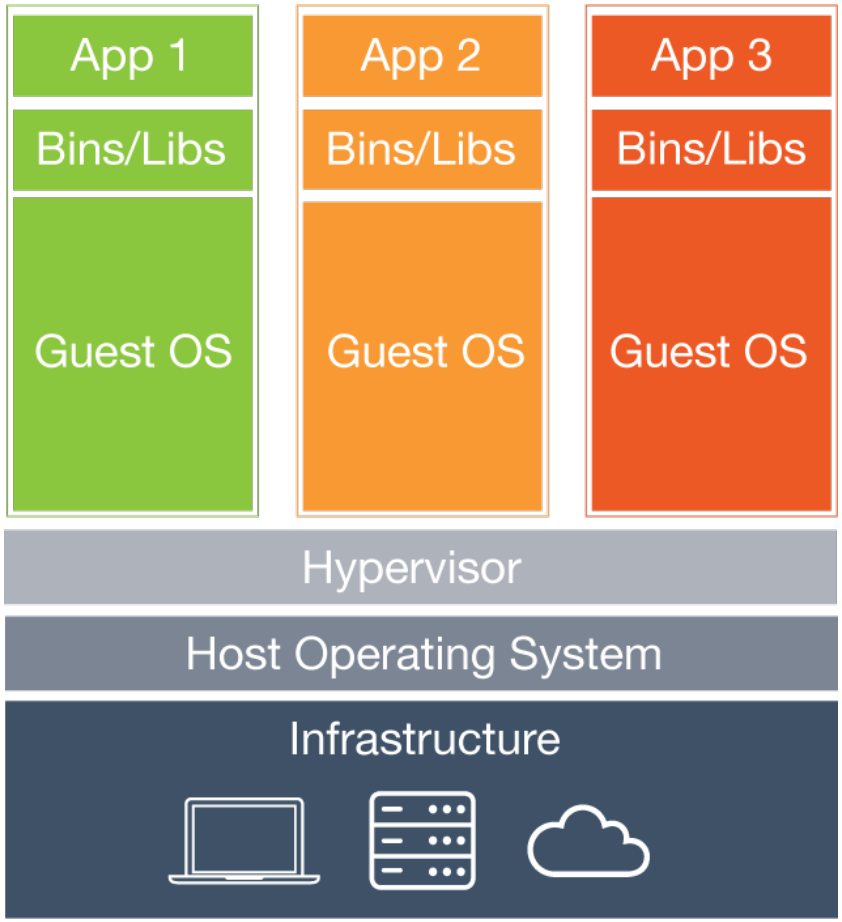
Infrastructure as a Code - example

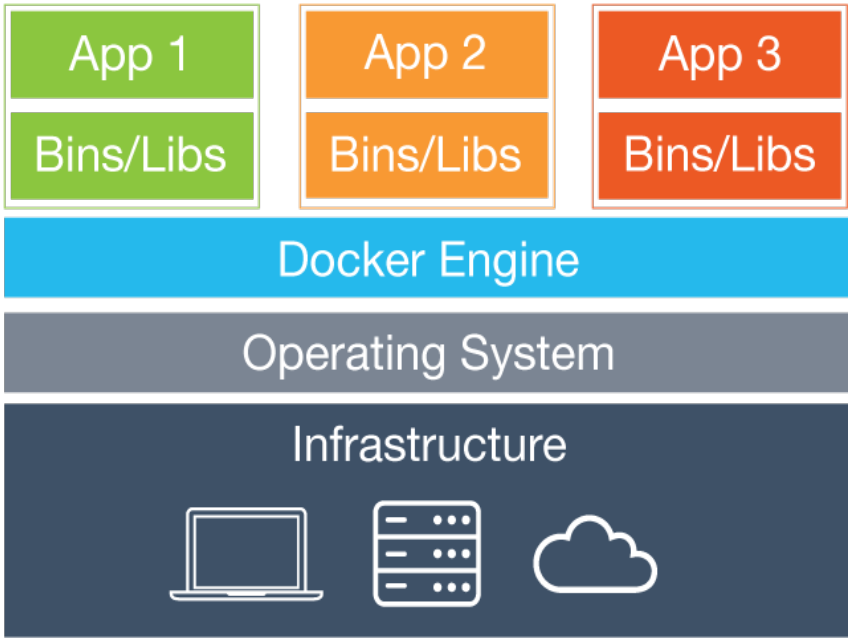
```
---
- hosts: server
  sudo: yes
  sudo_user: root

  tasks:

  - name: install mysql-server
    apt: name=mysql-server state=present update_cache=yes

  - name: install ansible dependencies
    apt: name=python-mysqldb state=present

  - name: Ensure mysql is running
    service: name=mysql state=started

  - name: Create user with the password and all previleges
    mysql_user: login_user=root login_password="" name={{ mysql_user }} password={{

  - name: Delete test database
    mysql_db: name=test state=absent

  - name: Create ansible_example database
    mysql_db: name=ansible_example state=present
```


docker

VM vs Docker

lightweight linux containers

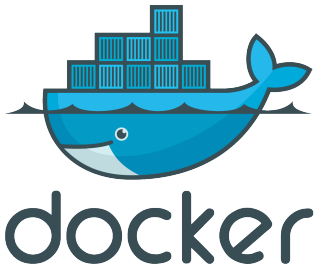




MongoDB setup

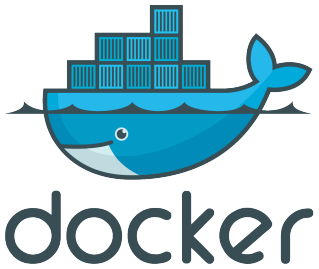
EXAMPLE

```
docker run --name workshop -d mongo

docker run -it --link some-mongo:mongo --rm
mongo sh -c 'exec mongo
"$MONGO_PORT_27017_TCP_ADDR:
$MONGO_PORT_27017_TCP_PORT/test"'
```



Dockerize nodejs app

https://docs.docker.com/engine/examples/nodejs_web_app/



Continuous Integration

How do you know that your code is ok?

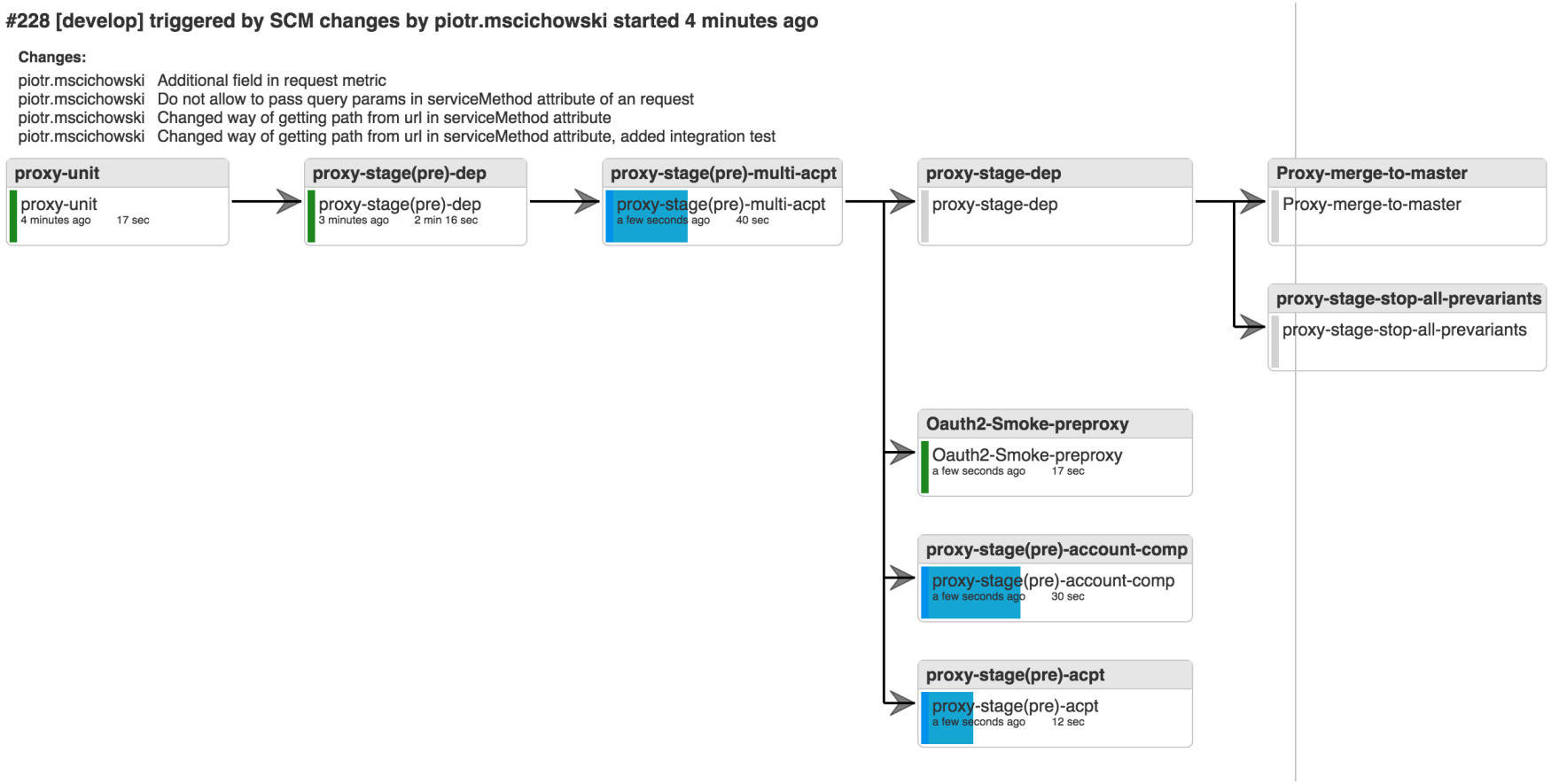
How do you know that it is not working only on your machine?

What about different OS, language versions, browsers, etc?

Continuous Integration for free - example

Travis CI

Jenkins Pipeline - sample



Continuous Integration

It is not only passing a test…

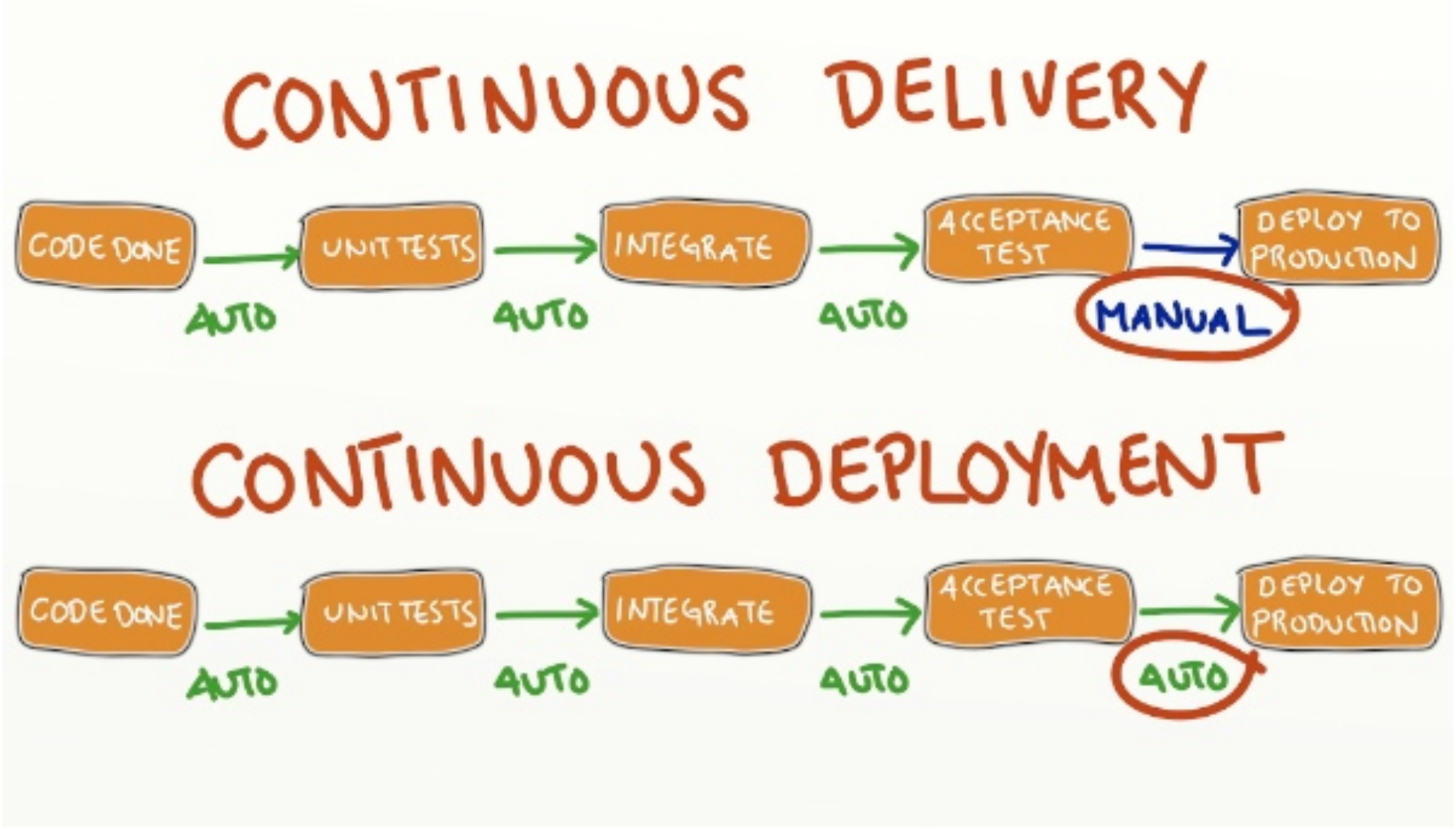
How it works with other components?

Environments:

STAGE

PROD

Continuous Deployment



Continuous Deployment for free - example

Monitoring

How do I know that it really works?

Acceptance tests

Smoke tests

Health check / uptime

Health check for free - example

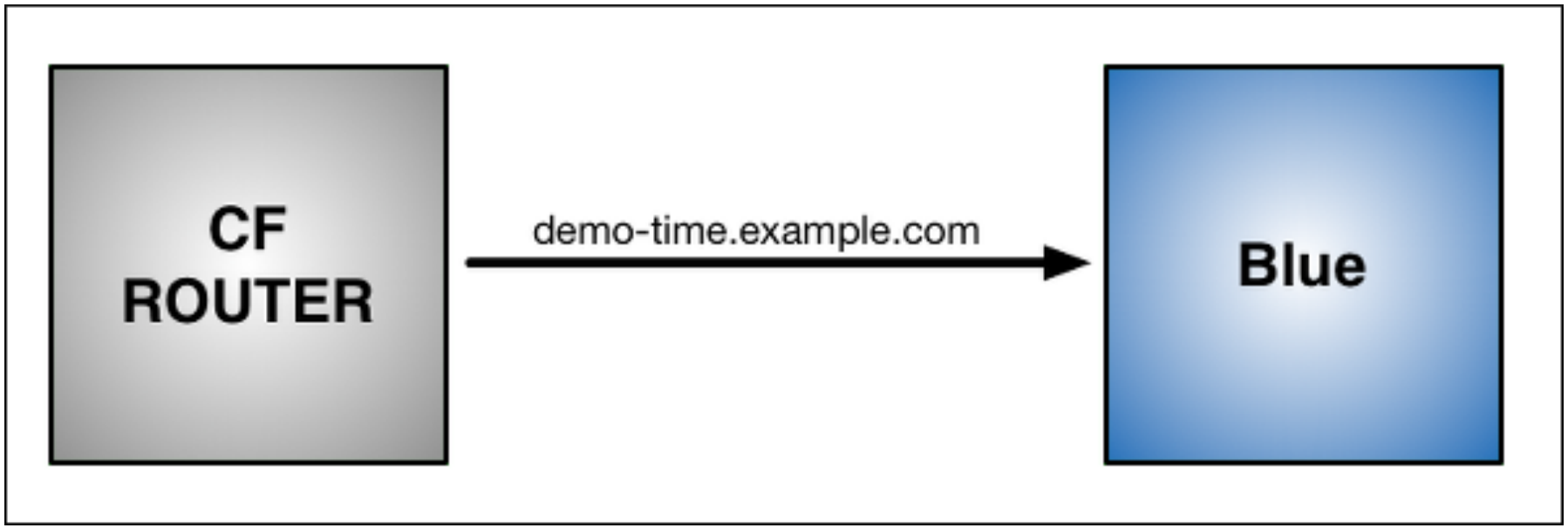
Monitoring

If your app is down it is too late ☹

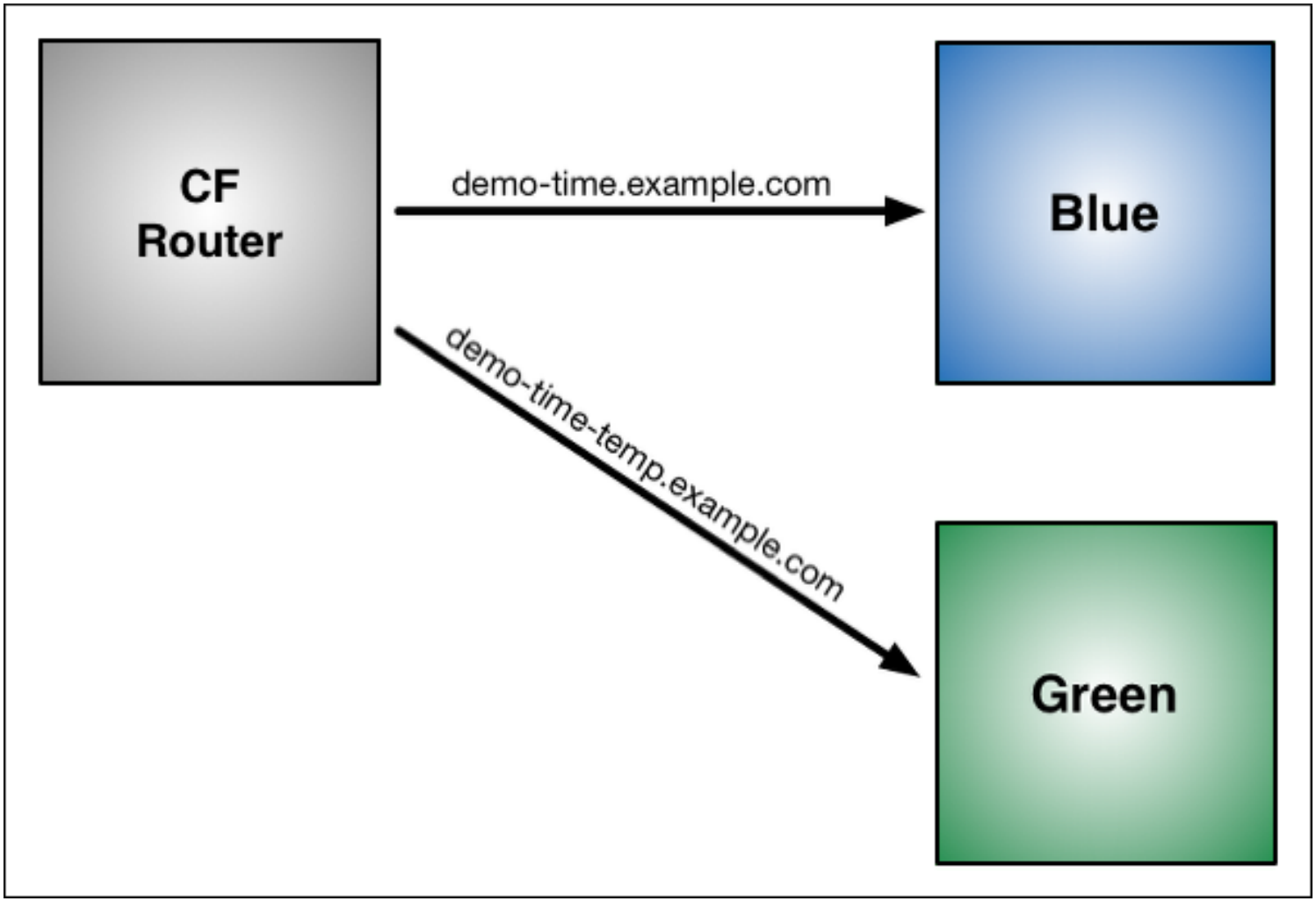
You should know BEFORE that you will have a problem

EXAMPLE:

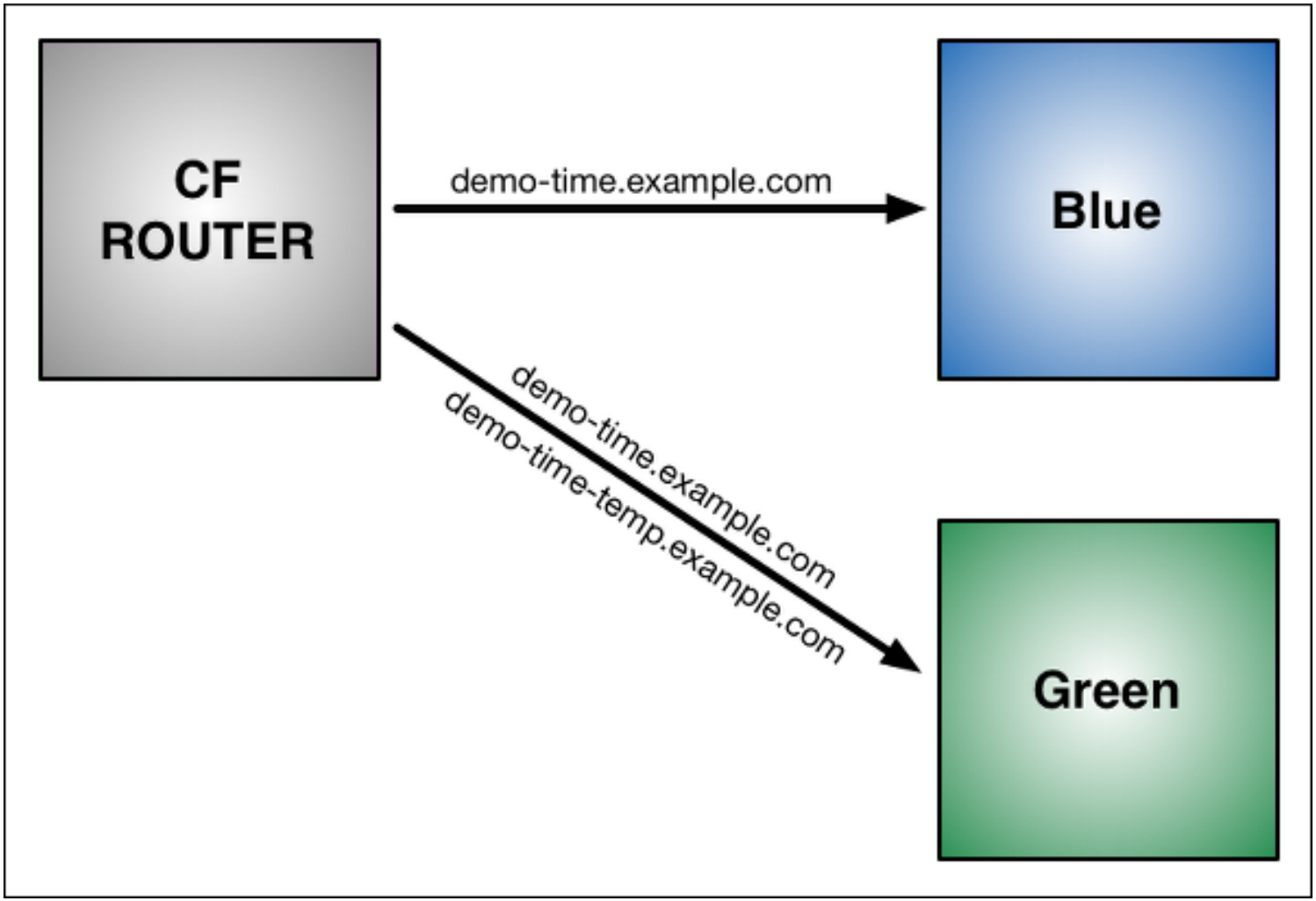
Blue Green Deployment – Cloud Foundry



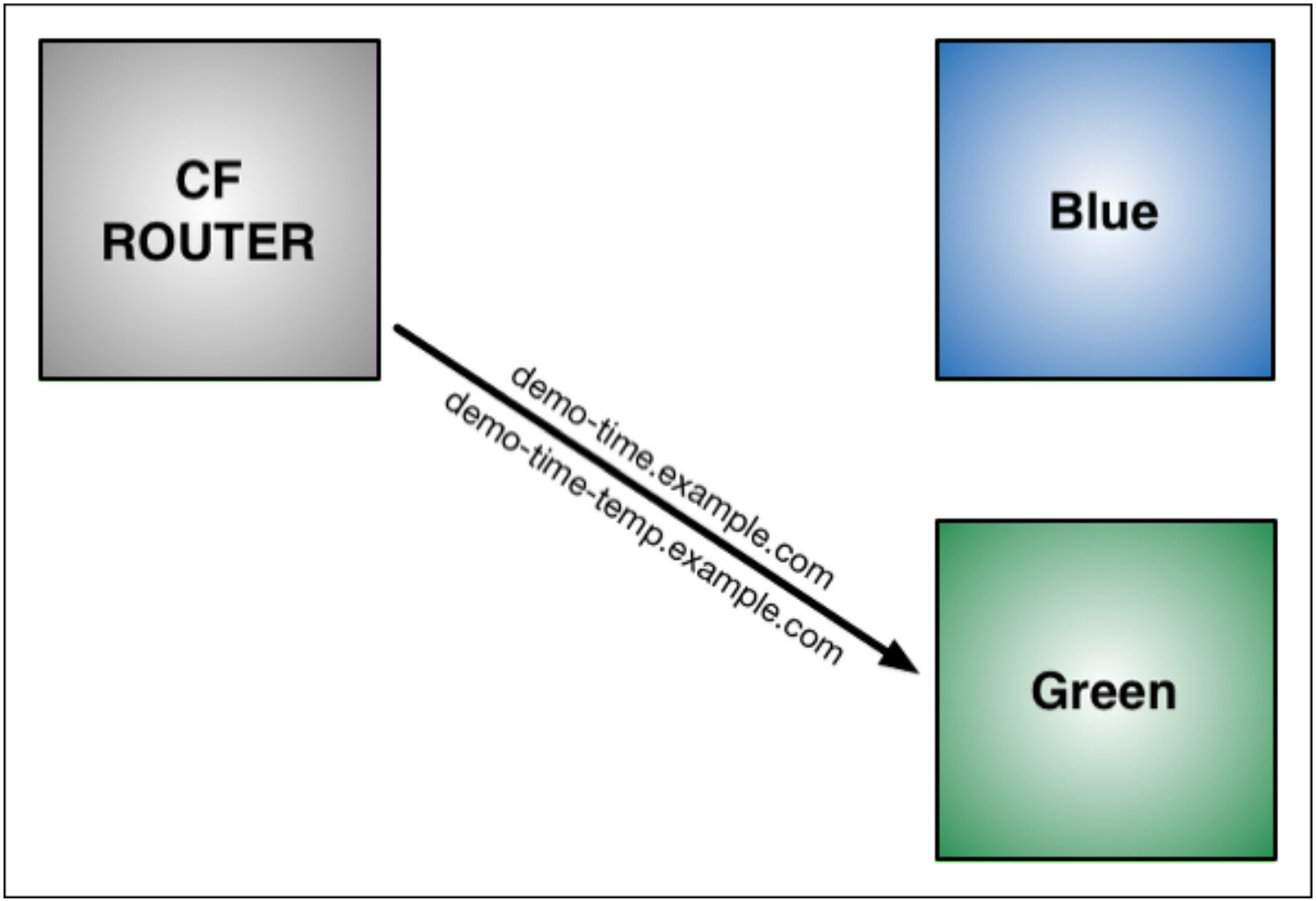
Blue Green Deployment – Cloud Foundry



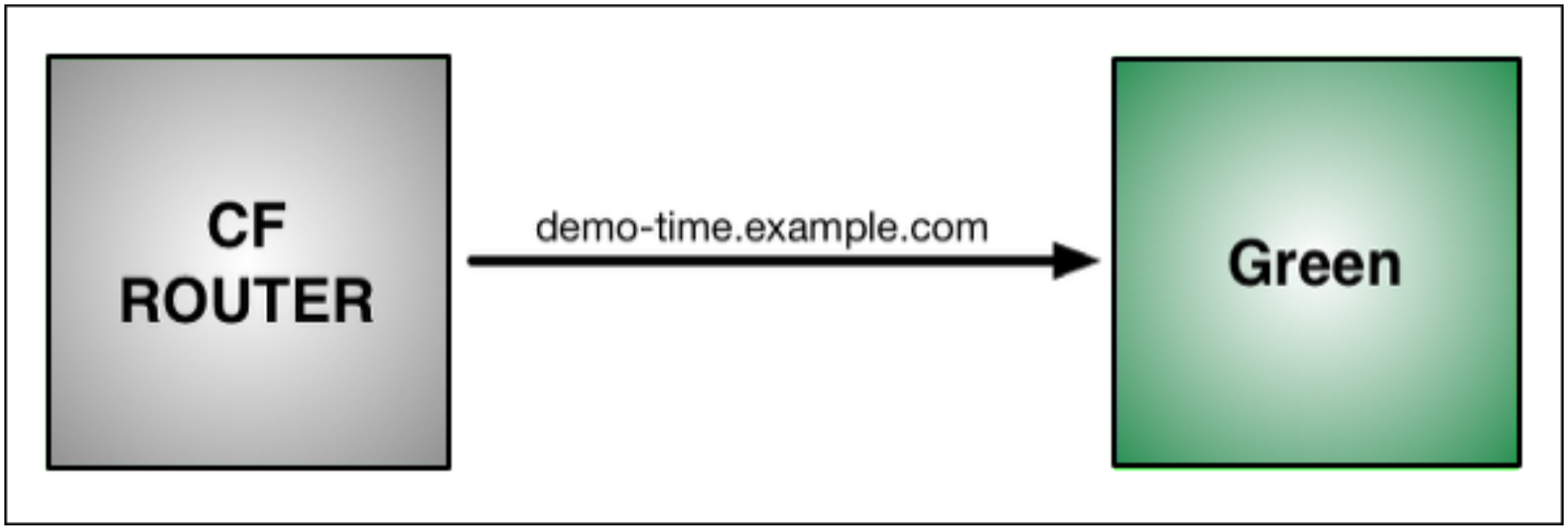
Blue Green Deployment – Cloud Foundry



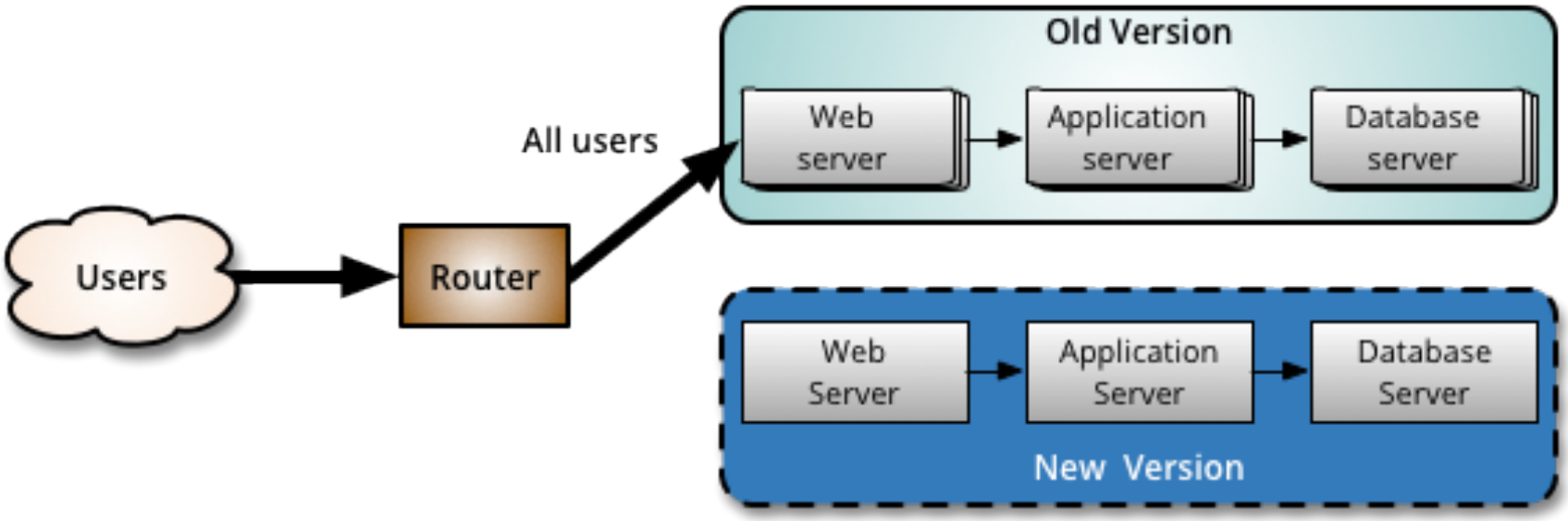
Blue Green Deployment – Cloud Foundry



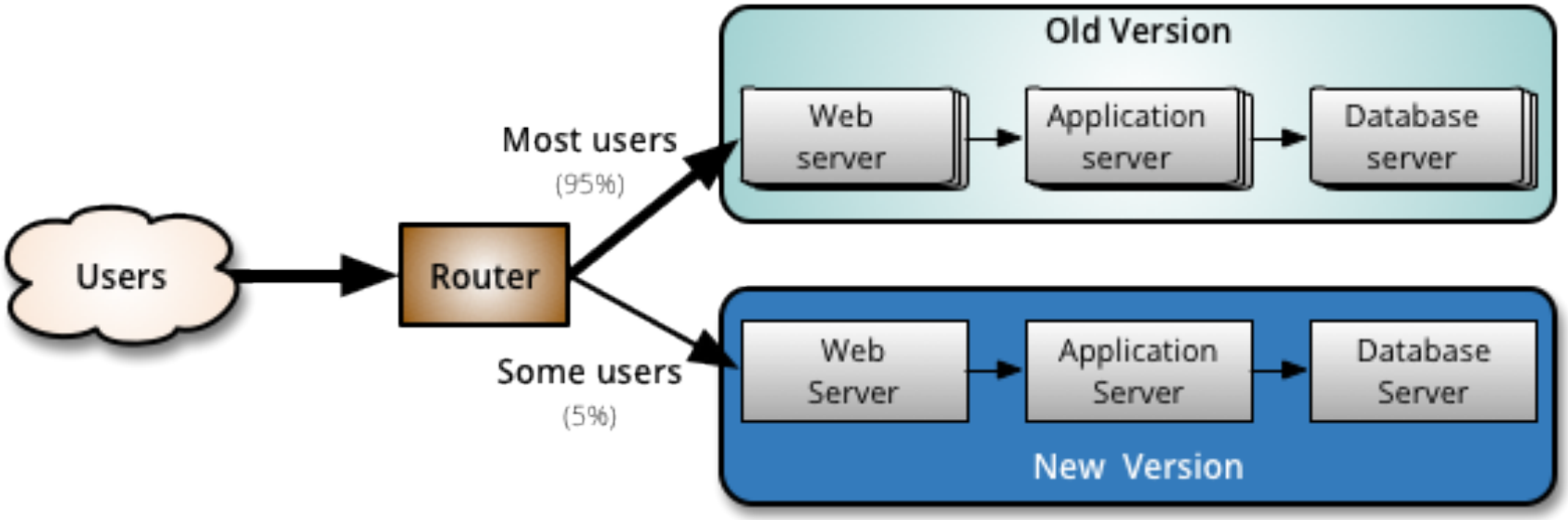
Blue Green Deployment – Cloud Foundry



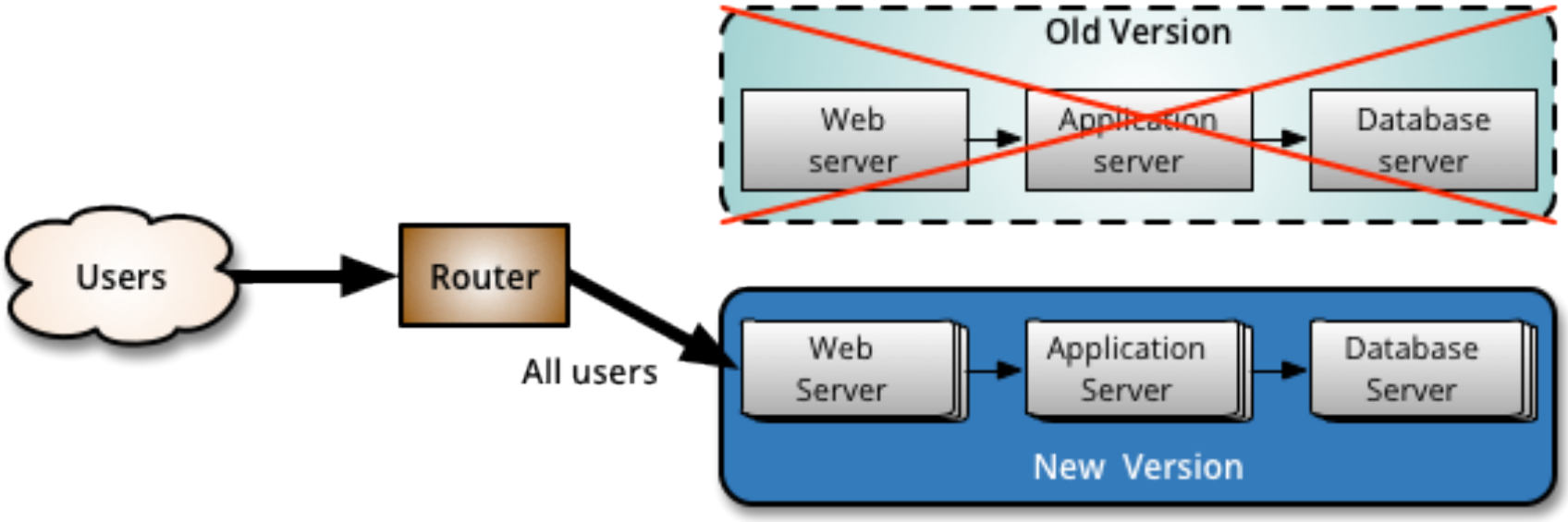
Canary Release



source: http://martinfowler.com/bliki/CanaryRelease.html



source: http://martinfowler.com/bliki/CanaryRelease.html



source: http://martinfowler.com/bliki/CanaryRelease.html