Continuous Delivery

road to 10+ deploys per day

@sn3d
linkedin.com/in/zdenkovrabel

DevOps ?

DevOps

The Three Ways

DevOps

Flow

(Business) (Customer)

dev ops

DevOps

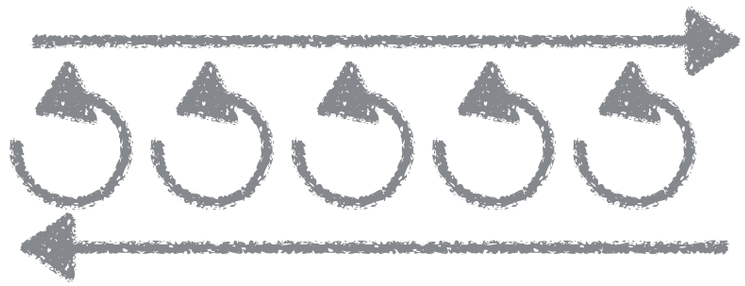
Feedback

(Business) (Customer)

DevOps

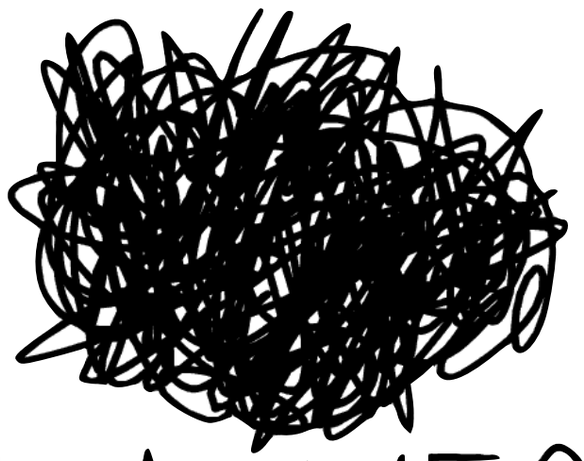
Experimentation

(Business) (Customer)

dev  ops

at the beginning there was...

OH YES



I'M A MESS

Theory of Constraints

Theory of Constraints

Every process has a constraint (**bottleneck**) and focusing improvement efforts on that constraint is the fastest and most effective path to improved profitability

Theory of Constraints

know your bottlenecks…

Theory of Constraints

measure it, monitor it

Rule nr.1

Every green build is potential release

Rule nr.2

Everything fails! That’s life.

bottleneck #1

Jenkins pipeline

bottleneck #1

Jenkins pipeline

Hermetic Build

bottleneck #1

Jenkins pipeline

"Just try to port your build pipeline from one company to another. All the hidden connections that make it efficient also make it harder to adapt."

- Release It!

bottleneck #1

Jenkins pipeline

Hermetic Build (by SRE)

Our builds are hermetic, meaning that they are insensitive to the libraries and other software installed on the build machine.

bottleneck #1

Jenkins pipeline

Hermetic Build (by SRE)

If two people attempt to build the same product at the same revision number in the source code repository on different machines, we expect identical results

bottleneck #1

Jenkins pipeline

shell scripts...

bottleneck #1

Jenkins pipeline

… Liqiubase with pure XMLs

bottleneck #1

Jenkins pipeline

Tests

- super-fast and ready for parallel running
- don't share data
- BDD

bottleneck #1

Jenkins pipeline

Tests

- ReadyAPI

bottleneck #1

Jenkins pipeline

Tests

- ~~ReadyAPI~~
- Java BDD (Cucumber + Serenity)

bottleneck #1

Jenkins pipeline

Tests

- ~~ReadyAPI~~
- ~~Java BDD (Cucumber + Serenity)~~
- Python (conda + behave)

bottleneck #1

Jenkins pipeline

Configuration

- command line arguments
- property file
- environment variables

bottleneck #1

Jenkins pipeline

cat ./envs/dev

export APP_JDBC_URI_SERVER=...
export APP_JDBC_USER=...
export APP_JDBC_PASSWORD=...

bottleneck #1

Jenkins pipeline

source ./envs/dev

bottleneck #1

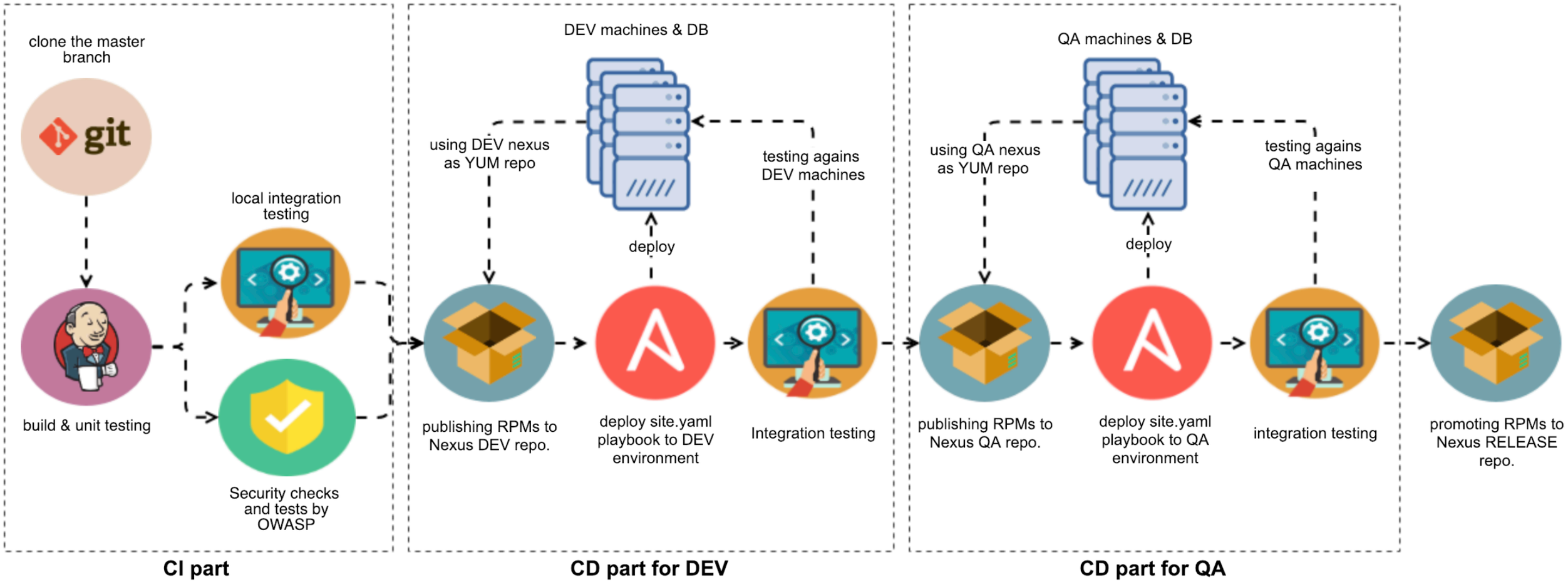
Jenkins pipeline

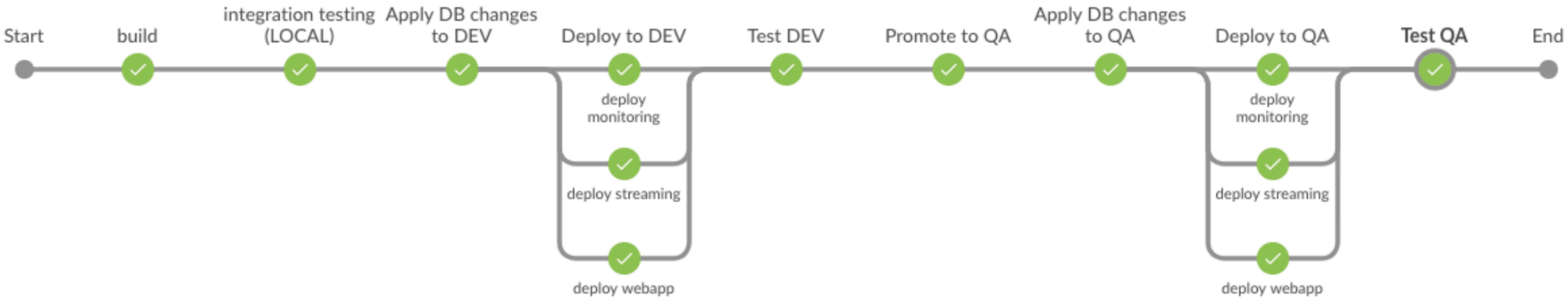
one simple declarative pipeline

bottleneck #1

Jenkins pipeline

```
pipeline {
  stages {
    ...
    stage("integration testing") {
      steps {
        sh "./bin/install"
        sh "./bin/create_db"
        sh "./bin/start"
        sh "./bin/integ_testing"
        sh "./bin/stop"
      }
    }
```


clone the master branch
git
local integration testing
build & unit testing
Security checks and tests by OWASP
CI part
DEV machines & DB
using DEV nexus as YUM repo
deploy
testing agains DEV machines
publishing RPMs to Nexus DEV repo.
deploy site.yaml playbook to DEV environment
Integration testing
CD part for DEV
QA machines & DB
using QA nexus as YUM repo
deploy
testing agains QA machines
publishing RPMs to Nexus QA repo.
deploy site.yaml playbook to QA environment
integration testing
promoting RPMs to Nexus RELEASE repo.
CD part for QA


Start
build
integration testing (LOCAL)
Apply DB changes to DEV
Deploy to DEV
deploy monitoring
deploy streaming
deploy webapp
Test DEV
Promote to QA
Apply DB changes to QA
Deploy to QA
deploy monitoring
deploy streaming
deploy webapp
Test QA
End

bottleneck #2

Deployment

bottleneck #2

Deployment

talk to your operations

bottleneck #2

Deployment

prepare distribution package (RPM) properly

bottleneck #2

Deployment

Liquibase as RPM

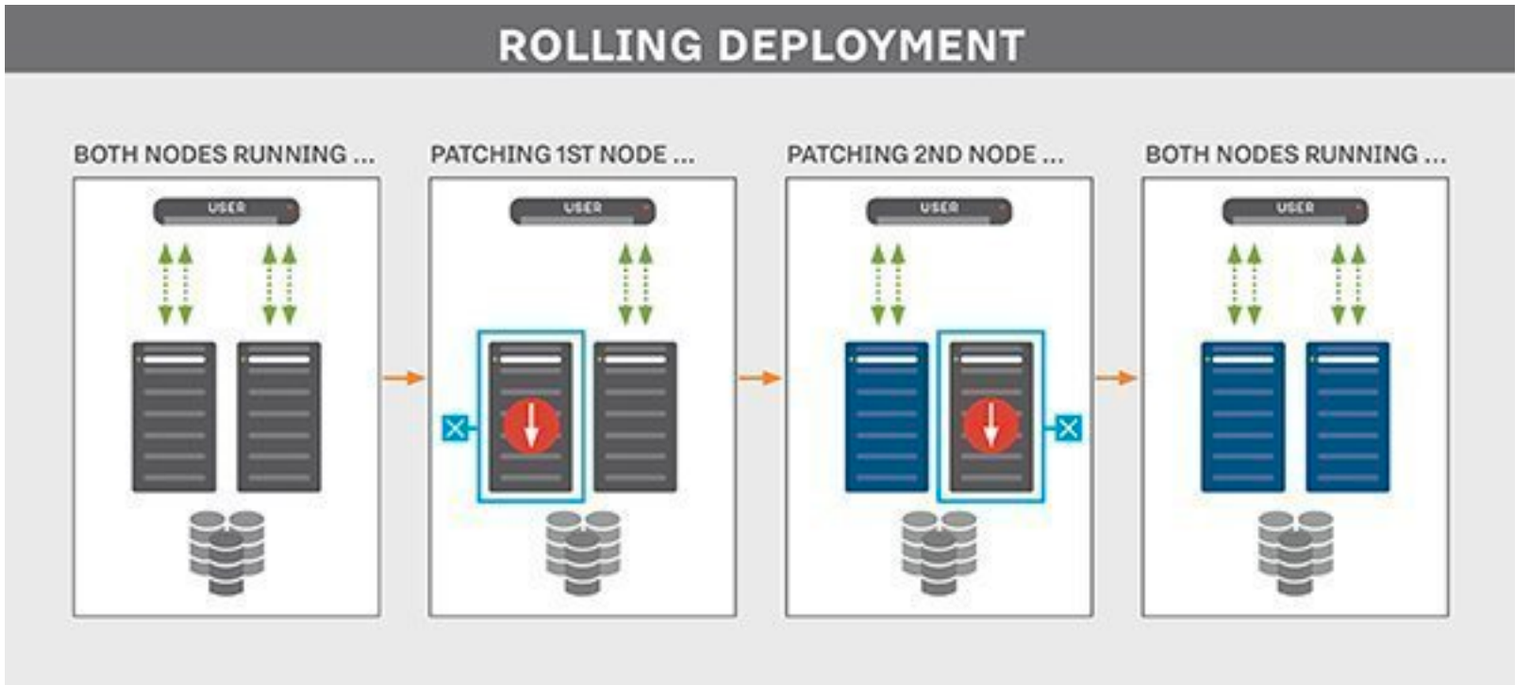
bottleneck #2

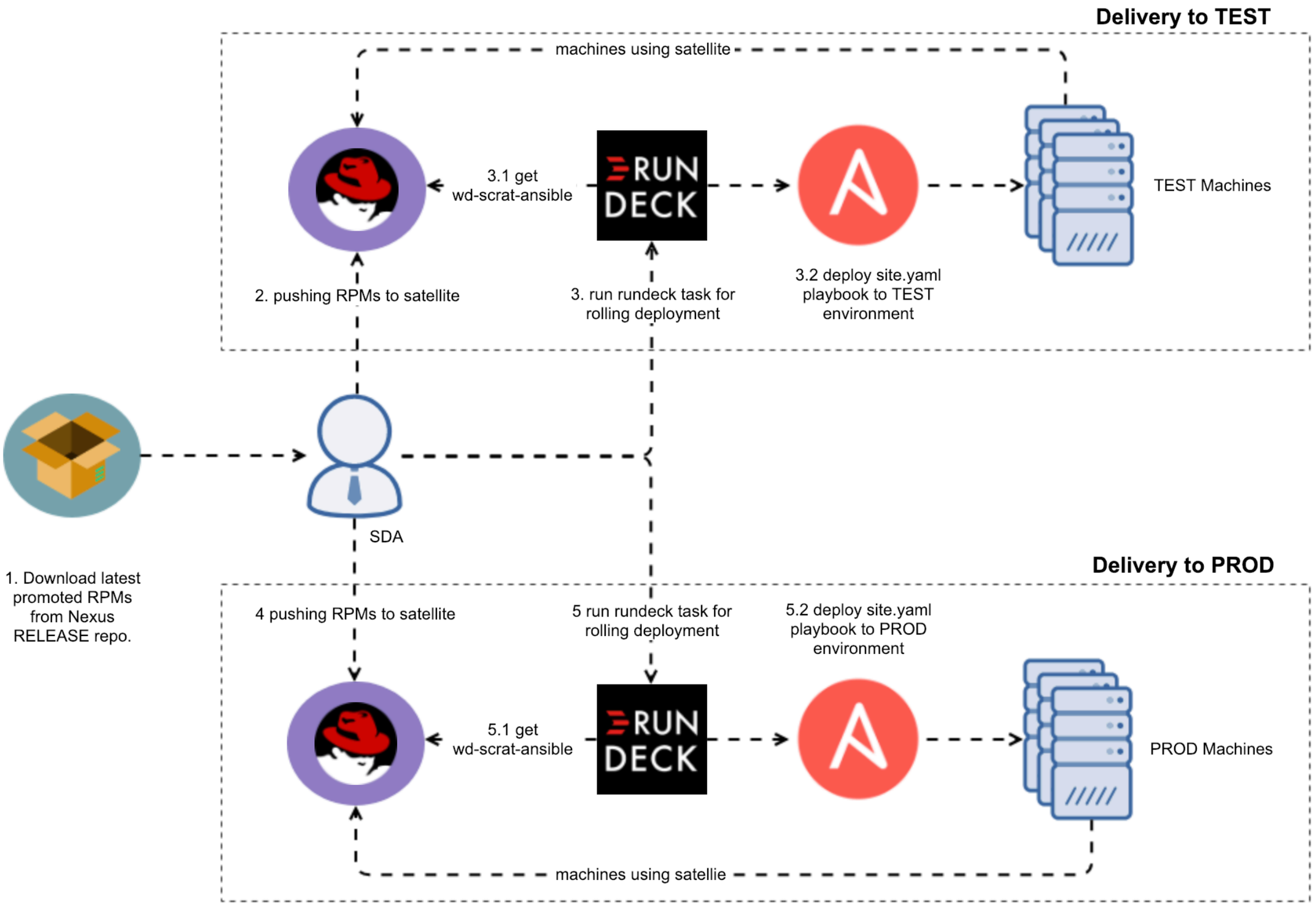
Deployment

Ansible scripts as RPM

bottleneck #2

Deployment




Delivery to TEST
machines using satellite
3.1 get
wd-scrat-ansible
RUN
DECK
TEST Machines
2. pushing RPMs to satellite
3. run rundeck task for
rolling deployment
3.2 deploy site.yaml
playbook to TEST
environment
SDA
1. Download latest
promoted RPMs
from Nexus
RELEASE repo.
Delivery to PROD
4 pushing RPMs to satellite
5 run rundeck task for
rolling deployment
5.2 deploy site.yaml
playbook to PROD
environment
5.1 get
wd-scrat-ansible
RUN
DECK
PROD Machines
machines using satellie

bottleneck #2

Deployment

Lame Duck (by SRE)

The backend task is listening on its port and can serve, but is explicitly asking clients to stop sending requests.

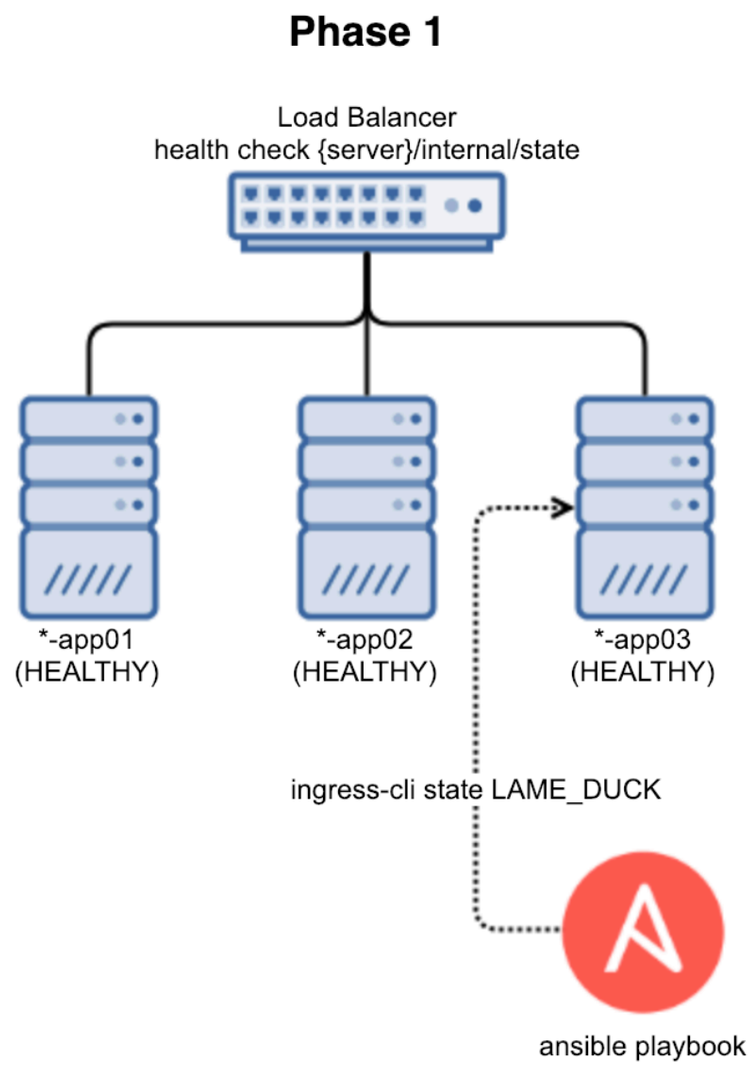
bottleneck #2

Deployment

```
mkfifo /tmp/app.in
```

bottleneck #2

Deployment



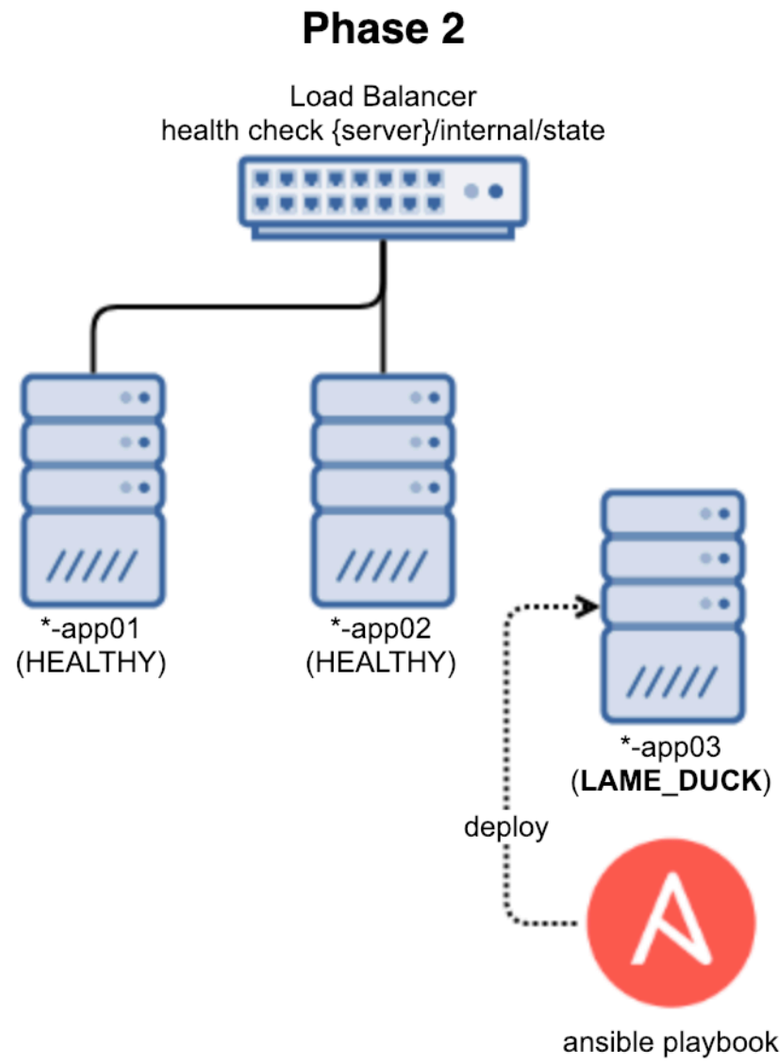
bottleneck #2

Deployment

```
echo “STATE LAMEDUCK” > /tmp/app.in
```

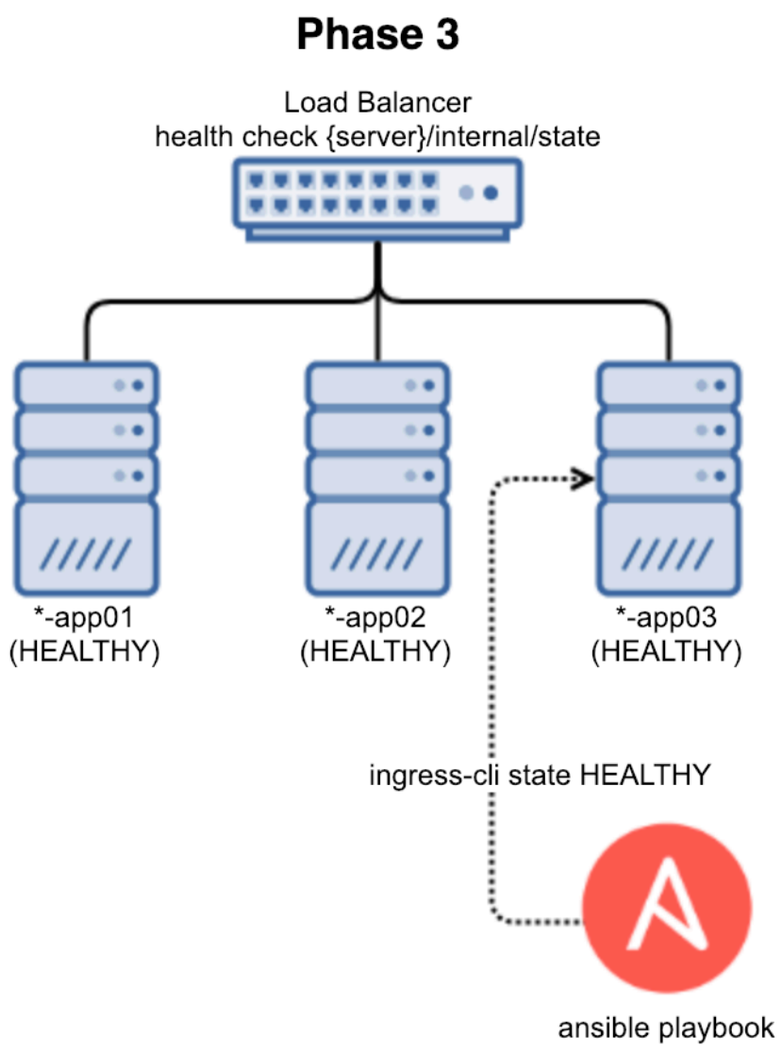
bottleneck #2

Deployment



bottleneck #2

Deployment



Release & Versions

Release and Versions

Snapshots?

Release and Versions

Maven release plugin?

Release and Versions

versions driven by tags

Release and Versions

```
$ git tag -a app-1.4
```

Release and Versions

```
$ git describe --tags --match 'app-*'
app-1.4-27-gde93a33
```

Release and Versions

```
$ git describe —tags —match ‘app-*’ | cut
    —delimiter=‘-‘ —fields=2
```

```
$ git describe —tags —match ‘app-*’ | cut
    —delimiter=‘-‘ —fields=3
```

Release and Versions

```
$ gradle build -Pversion=${BASE}.${PATCH}
```

bottleneck #3

Stages (DEV,INTE,QA,TEST)

bottleneck #3

Stages

There's no "right number" of QA environments. Virtualize them so every team can create its own on-demand QA environment.

- Release It!

bottleneck #4

Multiple Repositories

bottleneck #4

Multiple Repositories

Monorepository

bottleneck #4

Multiple Repositories

Monorepository

- one repo for devs, testers, ops

bottleneck #4

Multiple Repositories

Monorepository

- one repo for devs, testers, ops
- composite build (gradle)

bottleneck #4

Multiple Repositories

app/build.gradle

```
dependencies {
    implementation "org.my:some-lib:1.0"
}
```

bottleneck #4

Multiple Repositories

app/settings.gradle

```
includeBuild '../libs/some-lib'
```

bottleneck #5

GitFlow

bottleneck #5

GitFlow

Best merge?

bottleneck #5

GitFlow

Best merge?

No merge.

bottleneck #5

GitFlow

TrunkBasedDevelopment.com

- Paul Hammant

bottleneck #5

GitFlow

Engineers thought trunk-based development would never work, but once they started, they couldn't imagine ever going back

- Gary Gruver (HP)

bottleneck #5

GitFlow

Trunk-based development and continuous integration goes hand-in-hand, and one would not work effectively without the other.

- Jon Arild Tørresdal

bottleneck #5

GitFlow

The idea that developers should work in small batches off master or trunk rather than on long-lived feature branches is still one of the most controversial ideas in the Agile canon, despite the fact it is the norm in high-performing organizations such as Google. Indeed, many practitioners express surprise that this practice is in fact implied by continuous integration, but it is: The clue is in the word “integration.”

- state of DevOps 2016

bottleneck #5

GitFlow

Dark Launch

bottleneck #5

GitFlow

Feature Toggles

bottleneck #5

GitFlow

Feature Toggles

- build better abstractions

bottleneck #5

GitFlow

Feature Toggles

- build better abstractions
- testing is not easy

bottleneck #5

GitFlow

Feature Toggles

- build better abstractions
- testing is not easy
- have a plan to remove it

bottleneck #5

GitFlow

Feature Toggles

- build better abstractions
- testing is not easy
- have a plan to remove it
- always set default value: disabled

bottleneck #5

GitFlow

Feature Toggles

- build better abstractions
- testing is not easy
- have a plan to remove it
- always set default value: disabled
- config is probably not the right place

bottleneck #6

Disconnected Business and Engineering

bottleneck #6

Disconnected Business and Engineering

Confluence/Jira integration

bottleneck #6

Disconnected Business and Engineering

SCRUM & Sprints

bottleneck #6

Disconnected Business and Engineering

human intervention only when something goes wrong

What’s next?


REAL MEN...
TEST IN PRODUCTION
memegenerator.net

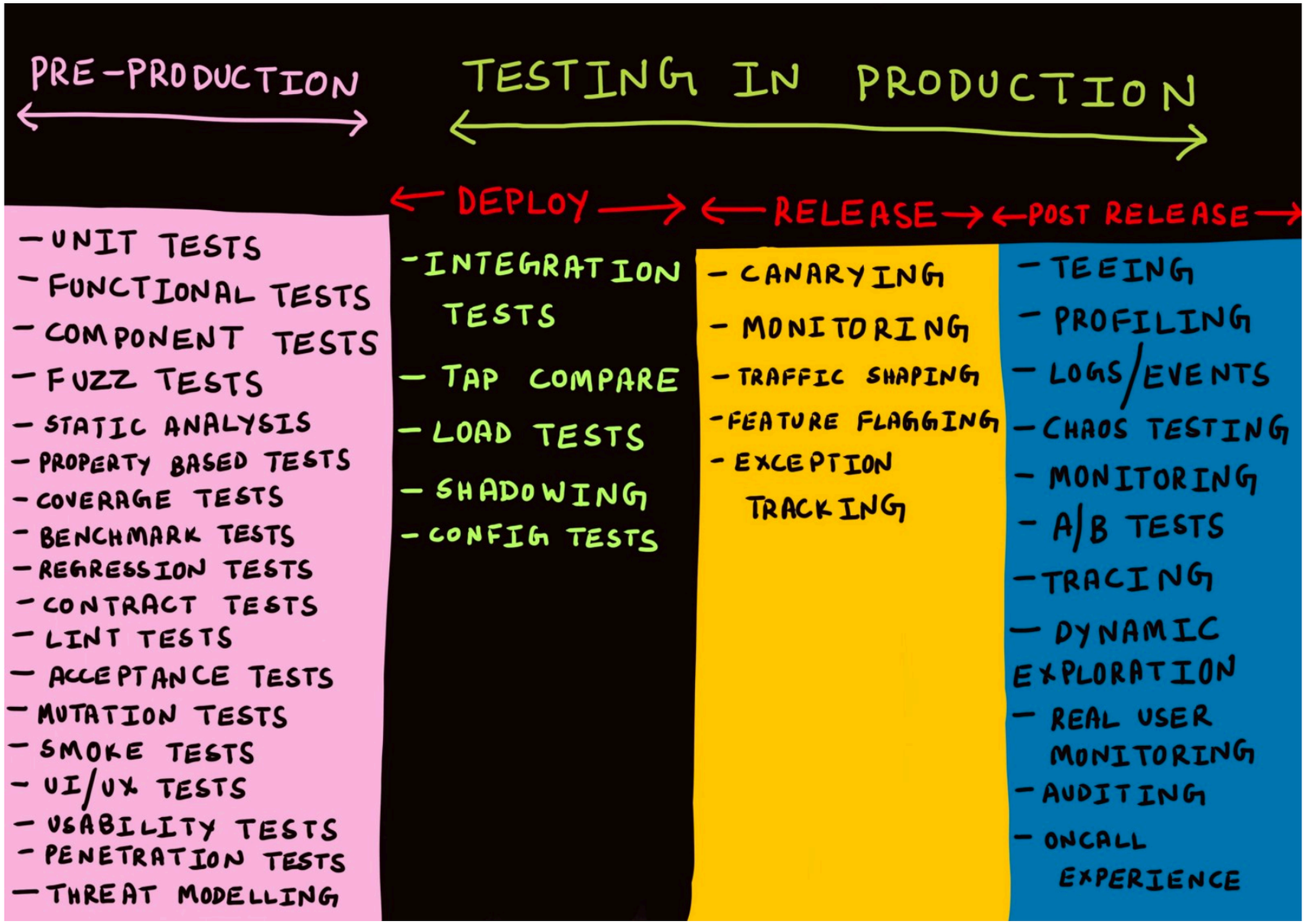
Observable Systems

Canary Releases

Traffic Mirroring

Chaos Engineering

Cindy Sridharan: Testing in Production, the safe way



(c) Cindy Sridharan

That's all Folks!