FULLY AUTOMATED

ZERO DOWNTIME

DEPLOYMENTS IN MAGENTO 2.2

Stämpfli

- @Jalogut
- Senior developer at Stämpfli AG
- +6 years working on Magento
- 2x certified

AGENDA

- Project Setup
- Wrong deployment
- Right deployment
- Zero Downtime
- Build Pipeline
- CI/CD

AGENDA

- **Project Setup**
- Wrong deployment
- Right deployment
- Zero Downtime
- Build Pipeline
- CI/CD

- git (or another VCS)
- composer
- composer.lock in VCS
- etc/config.php in VCS
- ~~etc/env.php~~ **NOT** in VCS
- ~~vendor~~ **NOT** in VCS

TUTORIAL

Proper Magento 2 Composer Setup:
https://blog.hauri.me/

EXAMPLE

https://github.com/jalogut/magento-2.2-demo

AGENDA

- Project Setup
- **Wrong deployment**
- Right deployment
- Zero Downtime
- Build Pipeline
- CI/CD

MANUALLY

[Demo]

CONS

- Waste of time
- Easy to make mistakes
- High Downtime: 15min - 30min

SIMPLE AUTOMATION

[Demo]

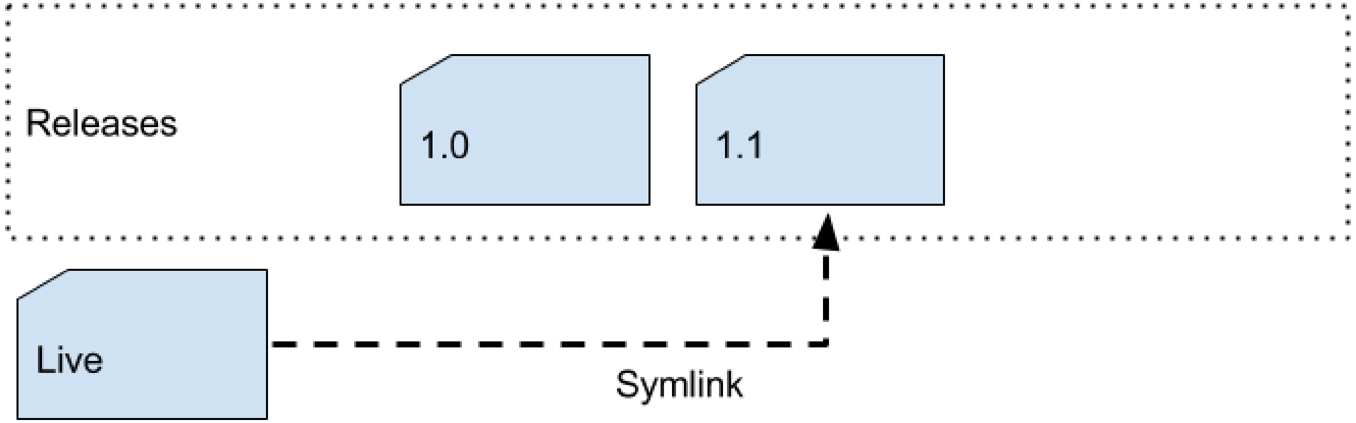
CONS

- Not reliable
- High Downtime: 15min - 30min

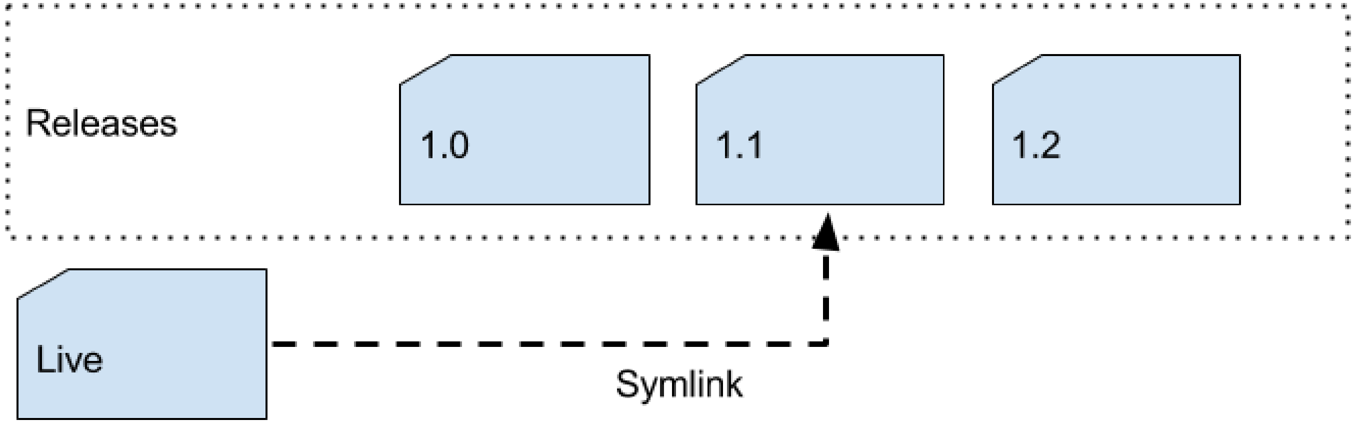
AGENDA

- Project Setup
- Wrong deployment
- **Right deployment**
- Zero Downtime
- Build Pipeline
- CI/CD

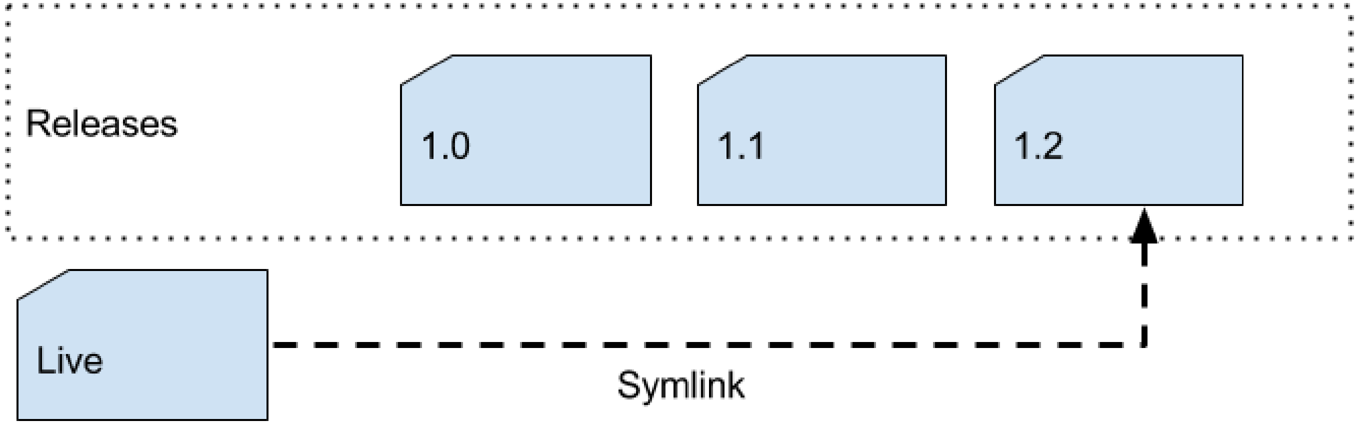
CURRENT LIVE



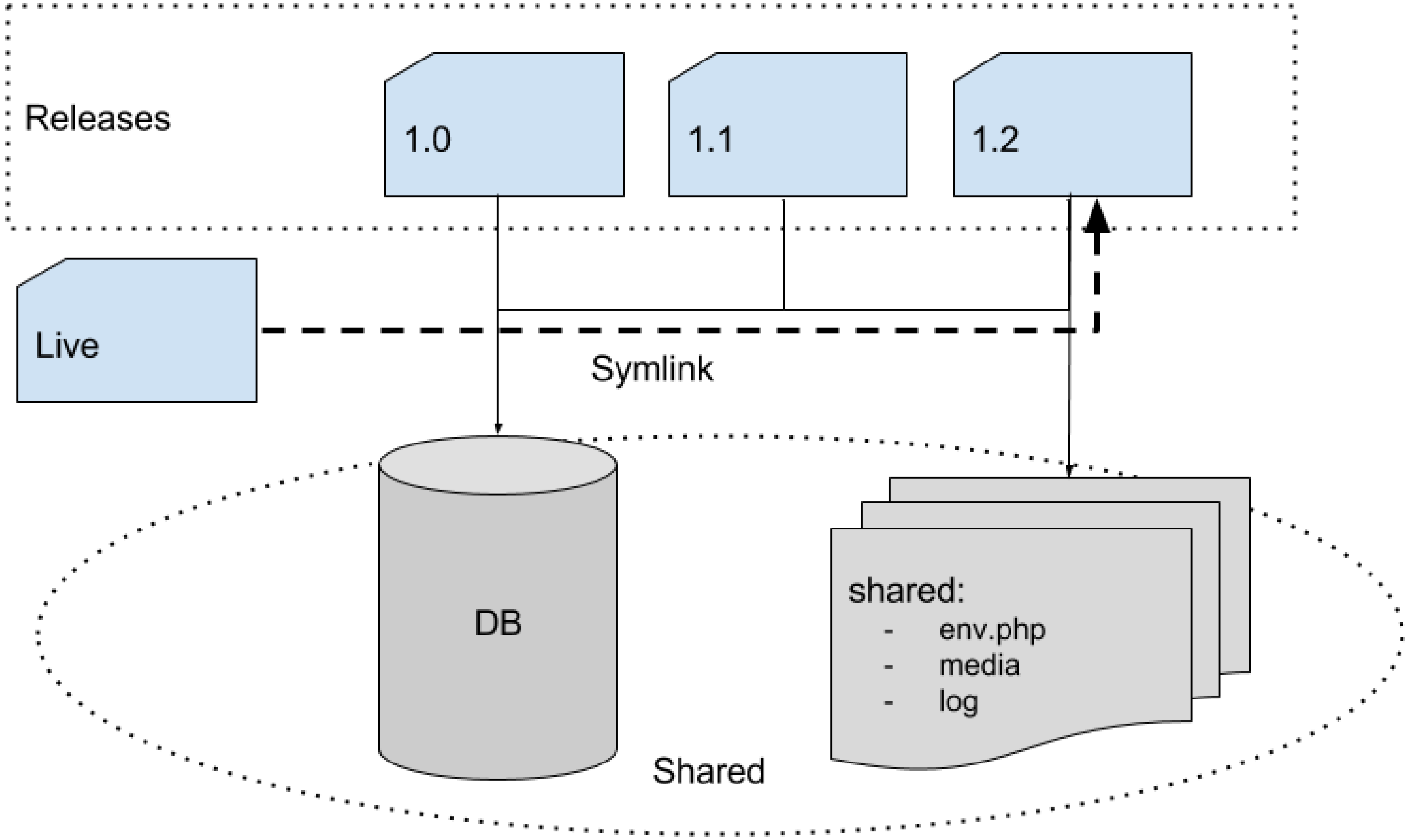
DEPLOYING NEW VERSION



SWITCH SYMLINK



SHARED DATA


Releases
1.0
1.1
1.2
Live
Symlink
DB
shared:
- env.php
- media
- log
Shared

TOOLS

- Deployer (PHP)
- Capistrano (Ruby)
- Shipit-deploy (Node-js)

[Demo]

PROS

- Fully automated
- Reliable
- Low Downtime: ~20sec

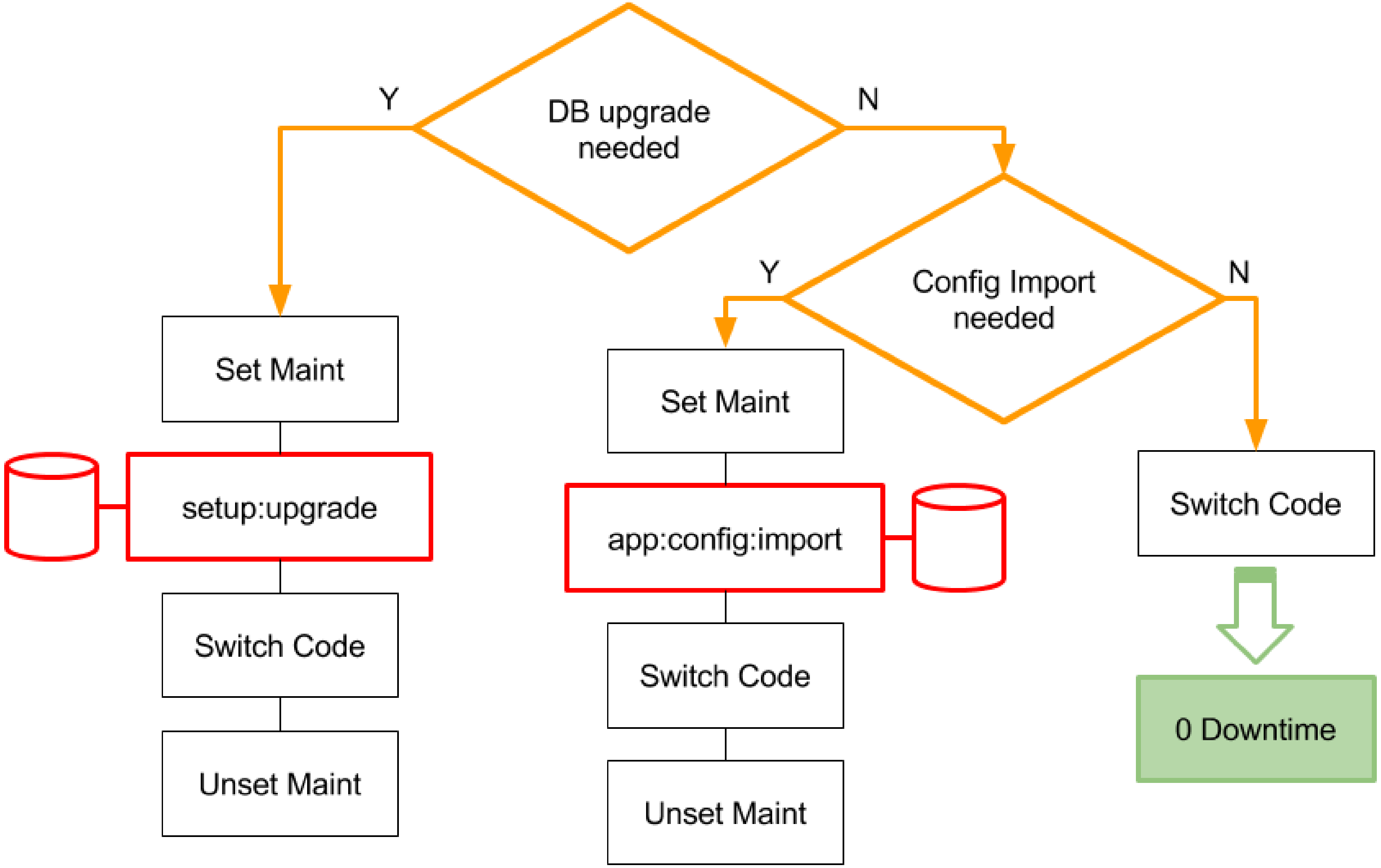
DISCLAIMER

HEREAFTER ONLY VERSION >= 2.2

AGENDA

- Project Setup
- Wrong deployment
- Right deployment
- **Zero Downtime**
- Build Pipeline
- CI/CD

SKIP MAINTENANCE



COMMANDS

- setup:db:status
- config:import:status (>= 2.2.3)

WORKAROUND (CONFIG:IMPORT:STATUS)

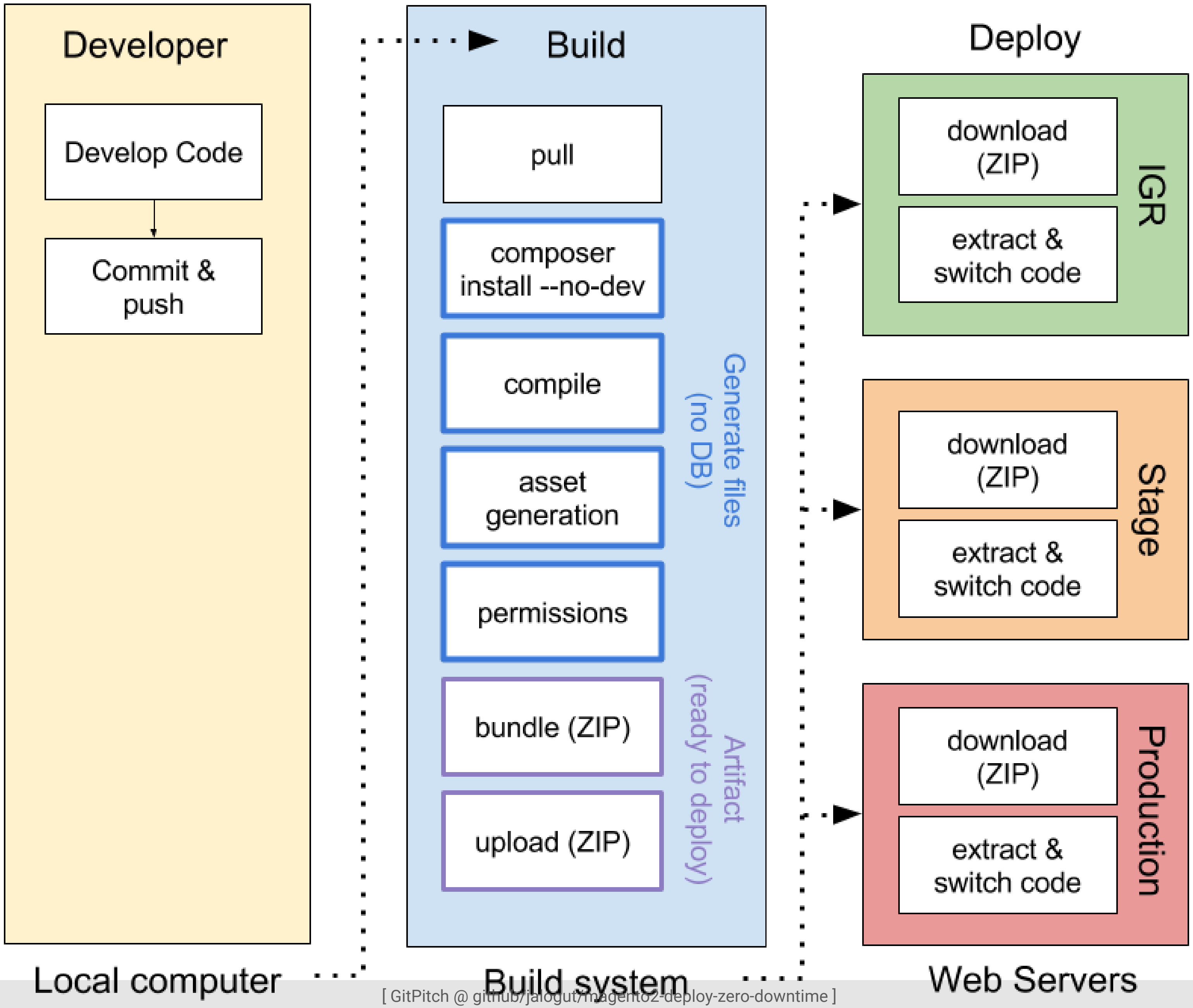
```
bin/magento config:set workaround/check/config_status 1
# Check output == "please run app:config:import"
```

[Demo]

ZERO DOWNTIME ACCOMPLISHED!

AGENDA

- Project Setup
- Wrong deployment
- Right deployment
- Zero Downtime
- **Build Pipeline**
- CI/CD

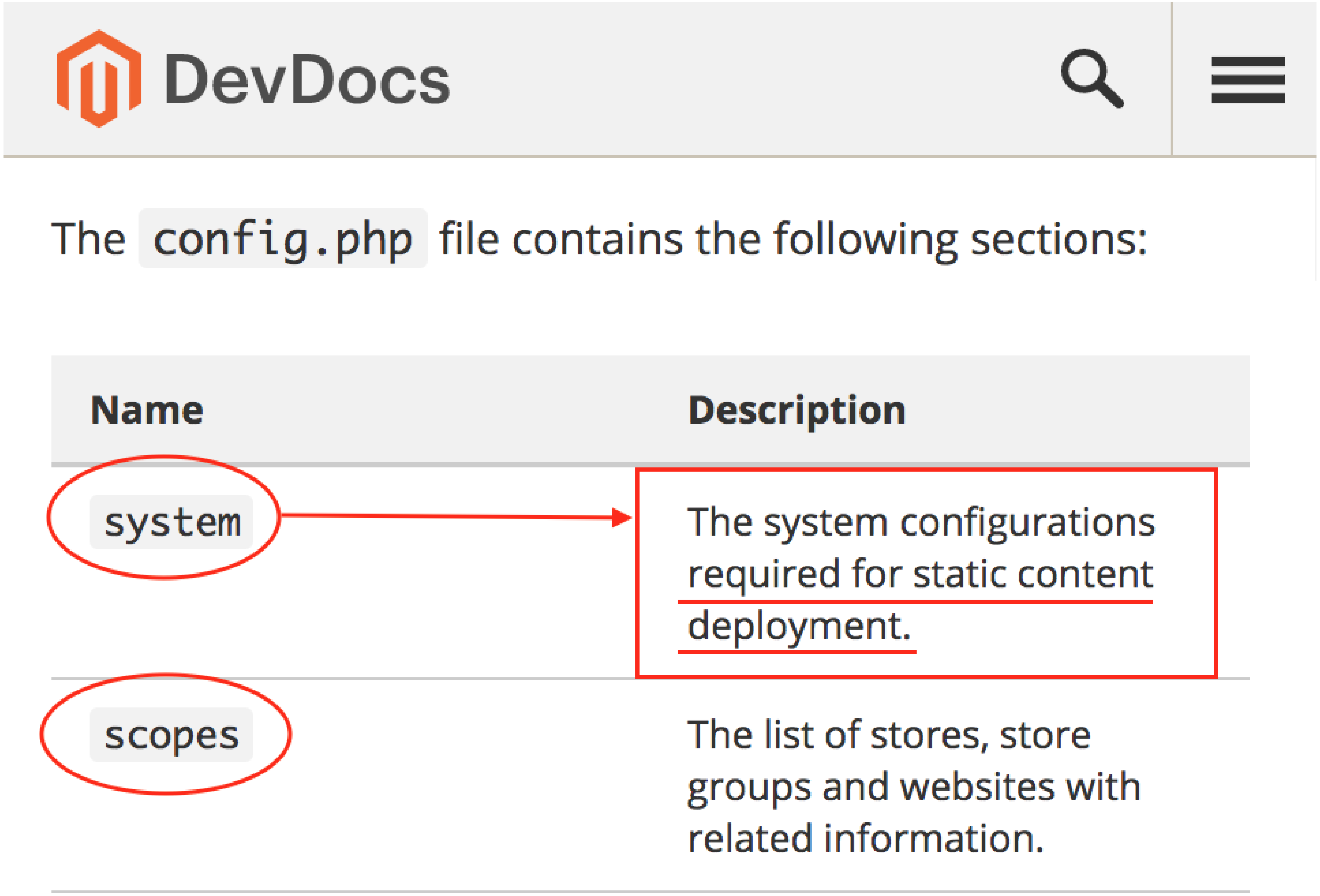

Developer
Develop Code
Commit & push
Local computer
Build
pull
composer install --no-dev
compile
asset generation
permissions
Generate files (no DB)
bundle (ZIP)
upload (ZIP)
Artifact (ready to deploy)
Build system
Deploy
download (ZIP)
extract & switch code
IGR
download (ZIP)
extract & switch code
Stage
download (ZIP)
extract & switch code
Production
Web Servers

WHAT DO WE NEED?

Config in Build === Config in Servers

HOW?

Config propagation in files



The `config.php` file contains the following sections:

Name	Description
`system`	The system configurations required for static content deployment.
`scopes`	The list of stores, store groups and websites with related information.

app:config:dump

UP-VOTE PR!

- Dump only static: PR #12410

WORKAROUND:

magento2-config-dump-skip-system.xml:
https://gist.github.com/jalogut/

[Demo]

WHY?

- Save CPU during deployment
- Identify files generation issues right away
- Common bundle to share amongst Servers
- History for auditing or back in time checks

AGENDA

- Project Setup
- Wrong deployment
- Right deployment
- Zero Downtime
- Build Pipeline
- **CI/CD**

AUTOMATE

tests → builds → deploys

JENKINS TUTORIAL

Setup Continuos Integration/Delivery system: https://dev.to/jalogut

FILES

- build.sh → Jenkins
- deploy.sh → Web Servers
- properties → git

CONTINUOUS DEPLOYMENT

Unique hash

```
# deploy.sh
if [[ ${VERSION} = "develop" ]]; then
    TARGET=${TARGET}-$(date +%s)
fi
RELEASE=${WORKING_DIR}/${TARGET}
```

Source: Jenkinsfile

```
node {
    DEV_SERVER="<user>@<host>"
    STAGE_SERVER="<user>@<host>"
    PROD_SERVER="<user>@<host>"

     // Clean workspace before doing anything
    deleteDir()

    try {
        stage ('Clone') {
            checkout scm
```

CI/CD

VIDEO DISABLED

STAGE/PROD Deploy

VIDEO DISABLED

TIPS!

↓

FLUSH ALL CACHES

OPcache, Varnish, ...

SPEED UP COMPOSER

Better caching

```
composer install --no-dev --prefer-dist --optimize-autoloa
```

Parallel Downloads

```
composer global require hirak/prestissimo
```

UPDATE CRONTAB AUTOMATICALLY

cron:install --force

```
<type name="Magento\Framework\Crontab\TasksProviderInterfa
    <arguments>
        <argument name="tasks" xsi:type="array">
            <item name="CUSTOM_CRON" xsi:type="array">
                <item name="expression" xsi:type="string">
                <item name="command" xsi:type="string">{ma
            </item>
        </argument>
    </arguments>
</type>
```

FASTER STATIC FILES DEPLOYMENT

Compact Strategy (up to 10x faster)

```
setup:static-content:deploy -f --strategy compact
```

Warning: I didn't test it yet!

DB BACKUP

```
# deploy.sh
if [[ 1 == ${UPGRADE_NEEDED} ]]; then
      bin/magento maintenance:enable
      n98-magerun2 db:dump --compression='gzip' ${WORKING_
      bin/magento setup:upgrade --keep-generated
fi
```

ISSUES

↓

APP:CONFIG:DUMP

- Avoid dumping everything: PR #12410
- https://gist.github.com/jalogut/ → magento2-config-dump-skip-system.xml

SHARE DEV CONFIGURATION

- Cumbersome using *$_ENV* variables: Dev docs
- Allow configuration by Env: PR #12361

JS TRANSLATIONS

- Still not fixed: #10673
- `setup:static-content:deploy` once per language

STATIC DEPLOY OPTIONS IGNORED

Doesn't exclude the theme

```
setup:static-content:deploy -f --exclude-theme=Magento/bla
```

OUT-OF-THE-BOX TOOLS

- staempfli/magento2-builder-tool
- staempfli/magento2-deployment-tool

Usage Example: jalogut/magento-22-mg2-builder

TAKE AWAYS

- Start small: zero downtime possible w/o build system
- Build pipeline allows new deployment strategies
- Pipeline issues: create a PR and be patient

RESOURCES

↓

MAGENTO 2:

- Alan Kent - Magento 2.2 deployment:
  - Video
  - Slides
- MageDeploy2:
  - https://dev98.de → Introducing MageDeploy2

DEPLOYMENTS IN GENERAL

https://serversforhackers.com/ → DeployPHP

≡

THIS PRESENTATION

- Slides: https://gitpitch.com/jalogut/magento2-deploy-zero-downtime
- Scripts: https://github.com/jalogut/magento2-deploy-zero-downtime/tree/master/scripts
- Project Example: https://github.com/jalogut/magento-2.2-demo

≡

THANKS

Feedback & Questions

@jalogut

≡