
Magento Live
UK | 2017

New Magento 2.2 Deployment Capabilities & Patterns

© 2017 Magento, Inc.

Alan Kent

Magento Chief Architect

Magento Commerce, ~~USA~~ Aussie

© 2017 Magento, Inc.

Coming in 2.2

New features to help deployments

- Build step no longer needs database access
- Faster compilation and asset generation
- Compact web assets – reduces bundle and extract time
- Read-only directories in production

Improved support for multiple environments

- Environment variables
- Per environment configuration file (env.php)
- Shared configuration file (config.php)
- New “sensitive” settings (passwords, PII)

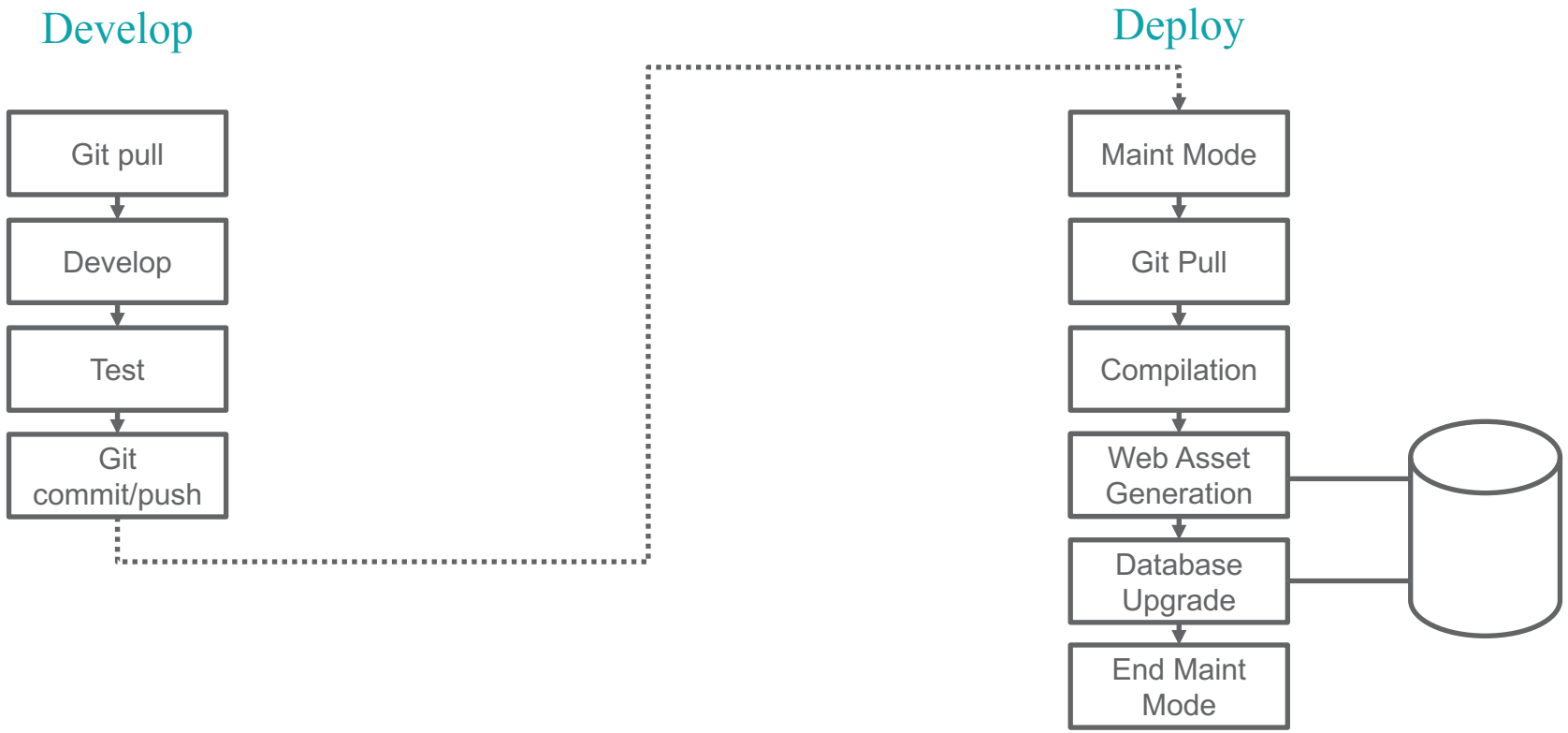
https://12factor.net/

I. Codebase in version control
III. Store config in environment
V. Build, release, run
X. Dev/prod parity

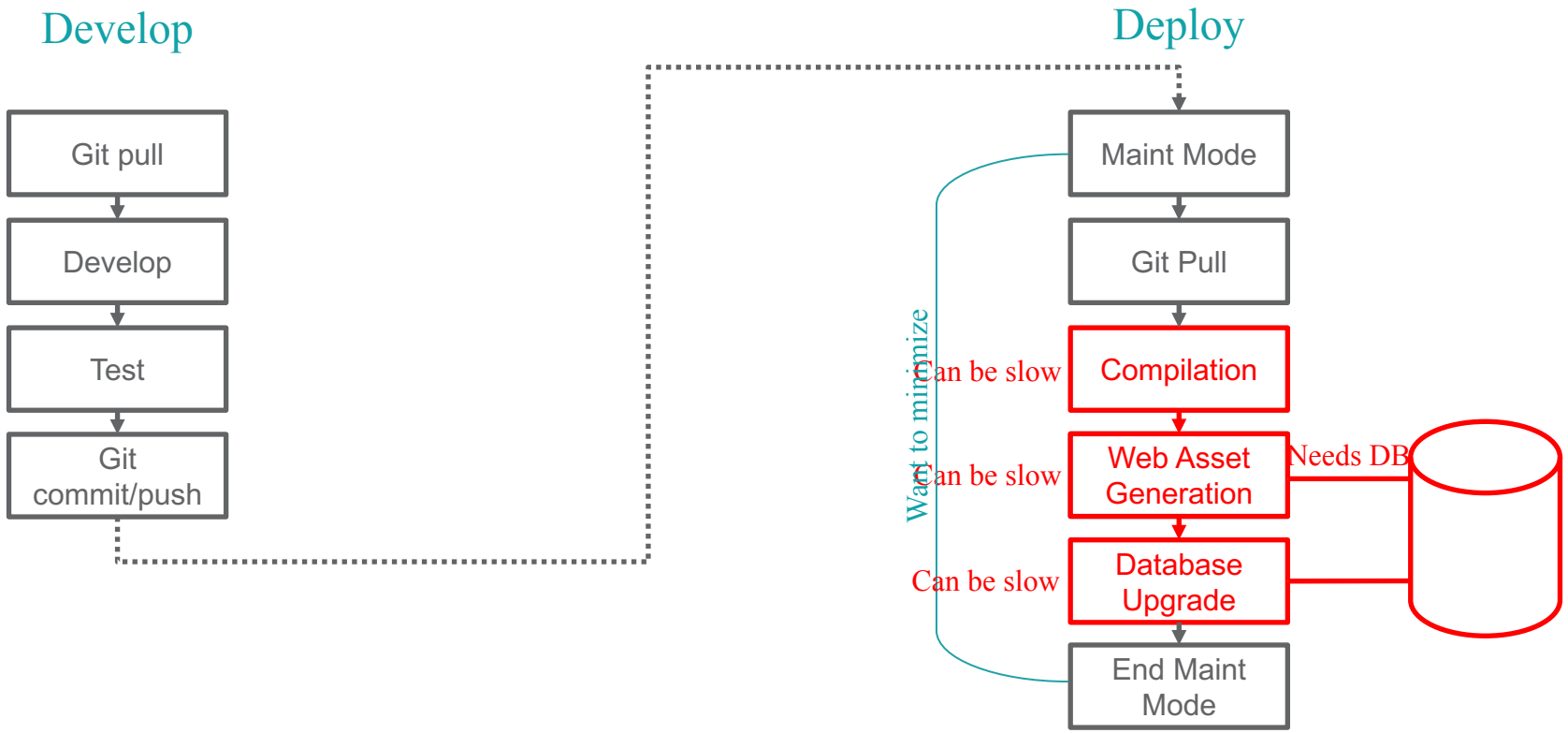
© 2017 Magento, Inc.

Deployment Flow

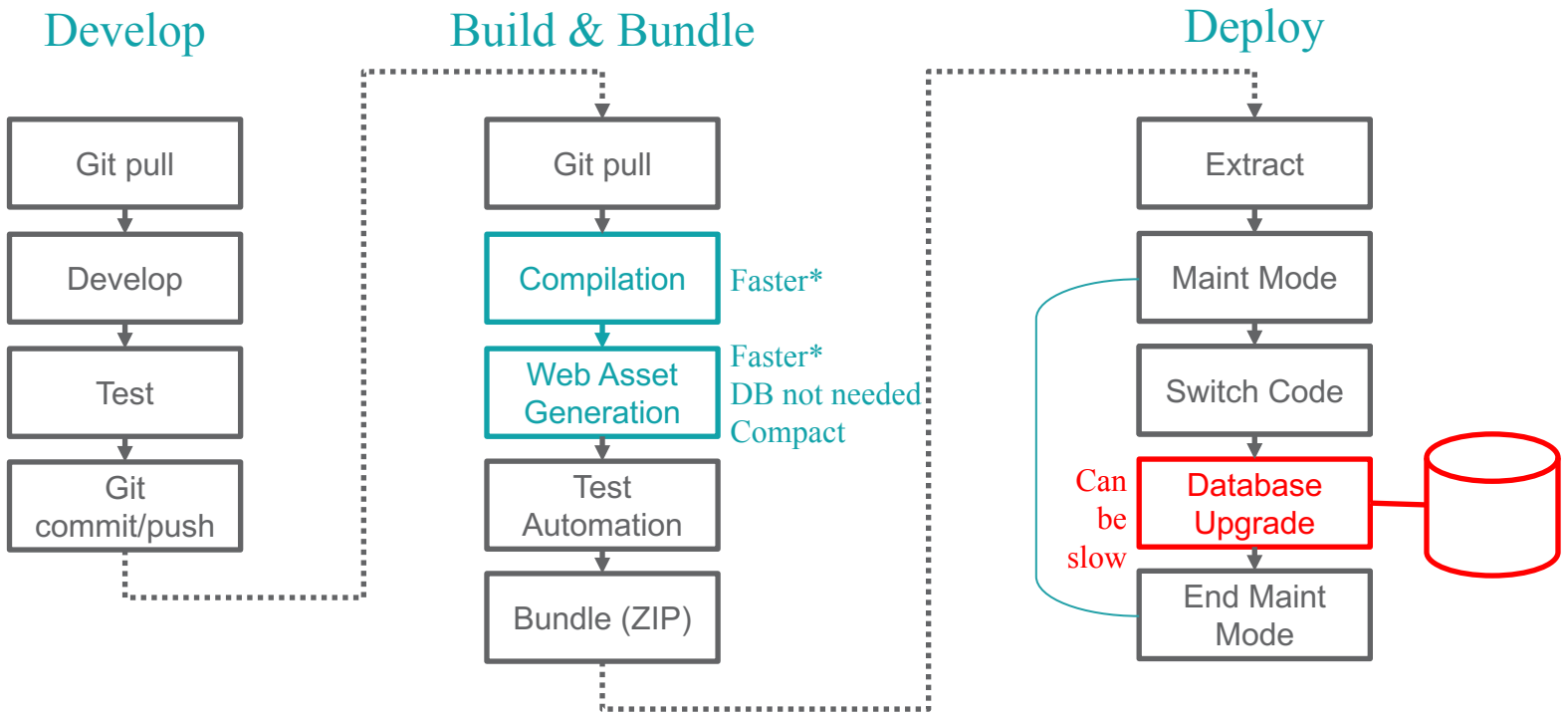
A Simple Development/Deployment Lifecycle



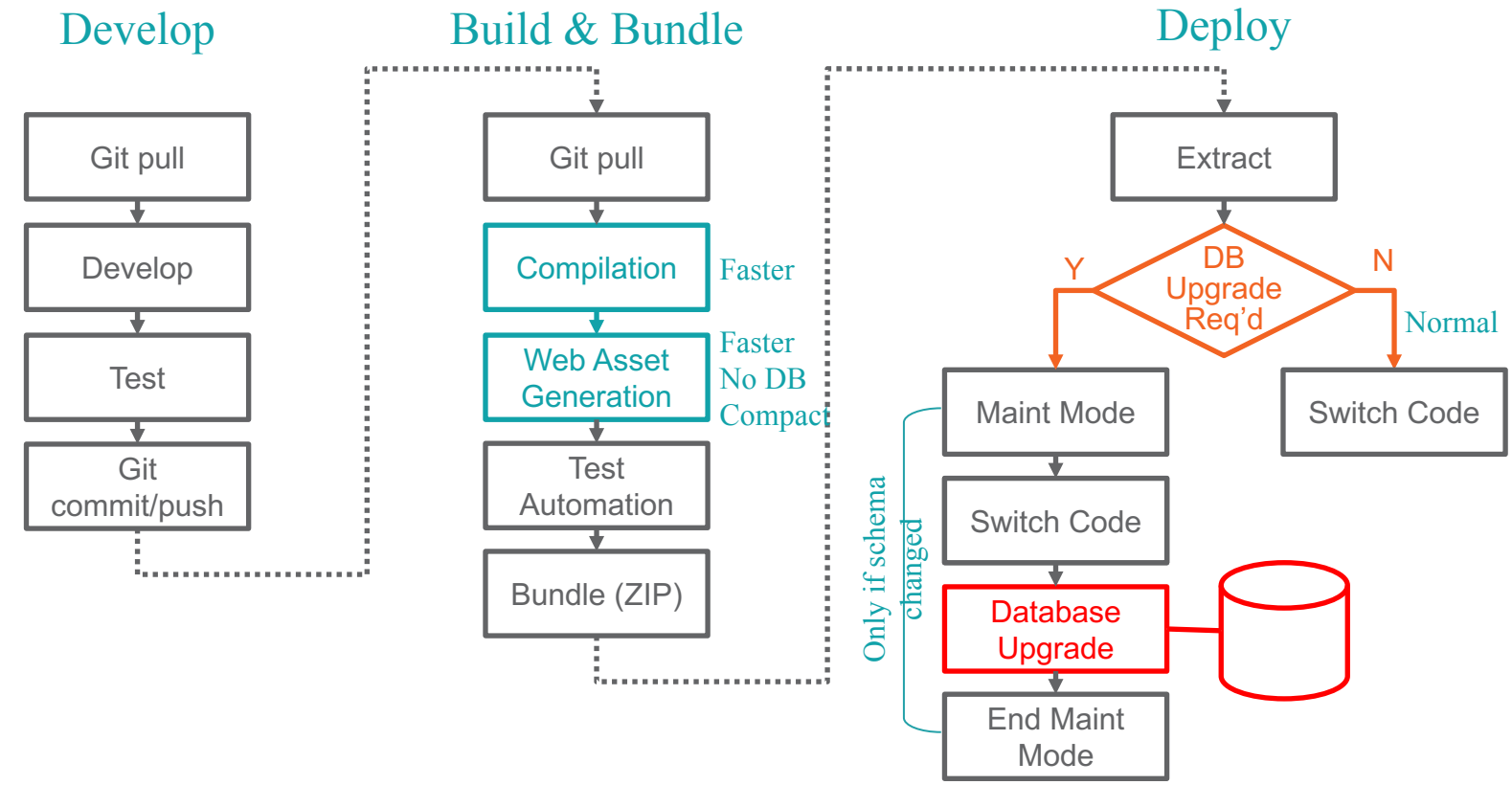
A Simple Development Deployment Lifecycle



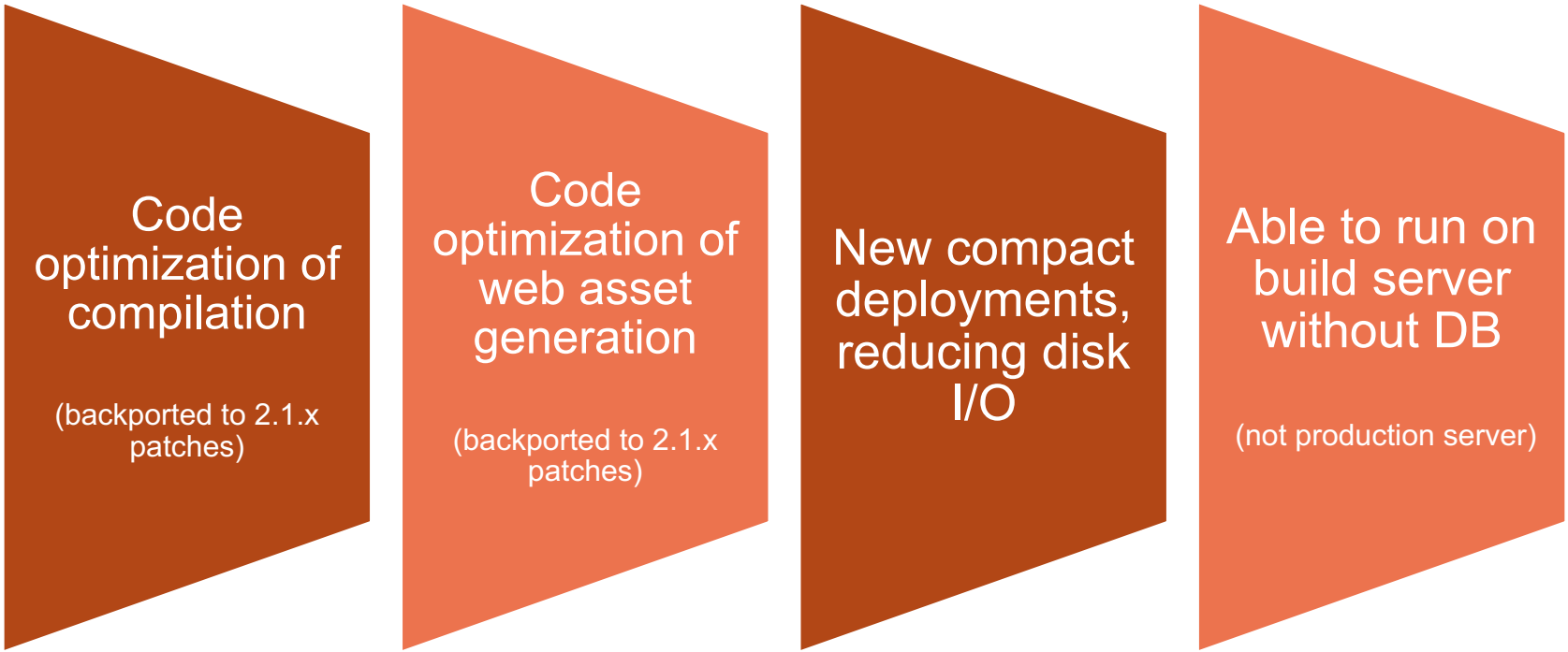
Adding a Build/Package Phase to Build Pipeline



Future: Zero Downtime Deployments (if no DB change)



Summary of Performance Boosts



© 2017 Magento, Inc.

Compact Mode for Static Assets

3 locales x2 faster; 15 locales x10 faster – due to less disk I/O

```
$ magento setup:static-content:deploy --strategy compact ...
```

Compact mode reduces duplication when multiple themes or locales

- Creates map.php & requirejs-map.js per area specifying real file locations
- PHP & JS functions supplied to map URLs to shared resources

Flexibility in what to do in build vs deploy phases, what is in git, …

```
$ magento setup:upgrade --keep-generated ...
```

© 2017 Magento, Inc.

Configuration as Code

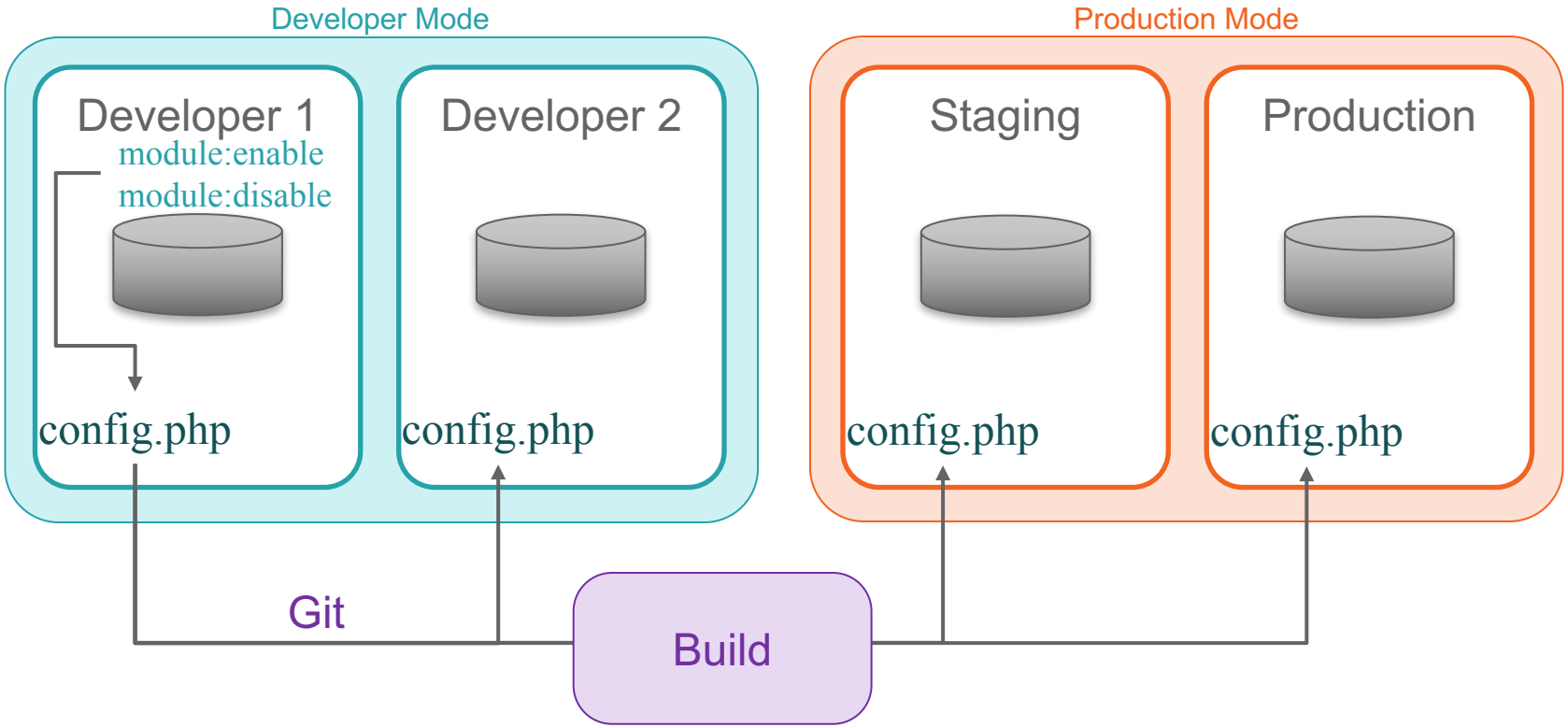
To move configuration between environments, 12 Factor App says “configuration is code” (under version control, pushed with code, …)

Sections of config.php

- Enabled modules
- Scopes (websites, stores, store views) and Themes
- Store configuration settings

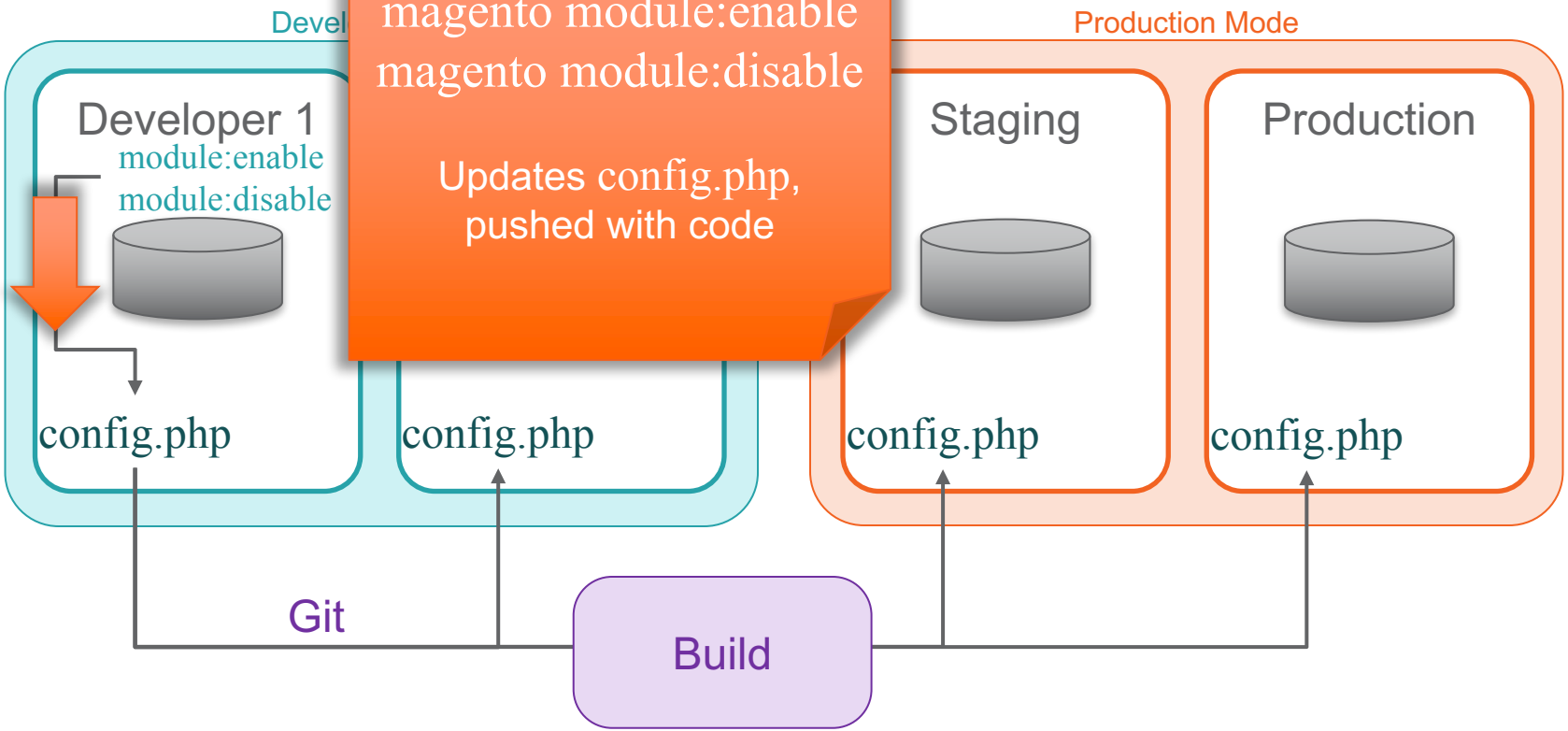
© 2017 Magento, Inc.

Enabled Modules Section



© 2017 Magento, Inc.

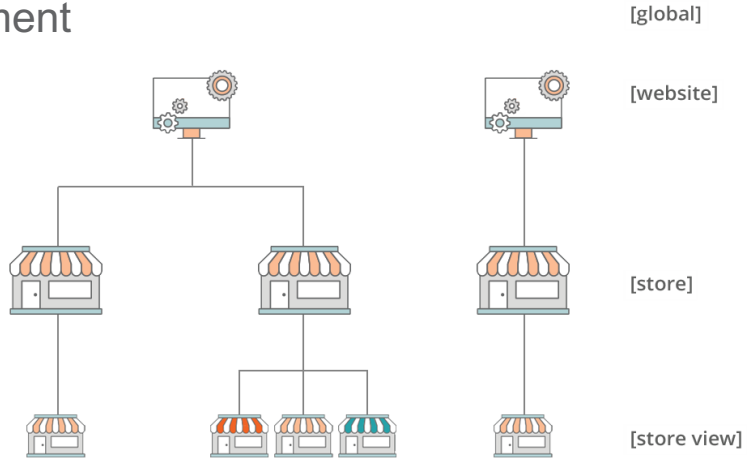
Enabled Modu



© 2017 Magento, Inc.

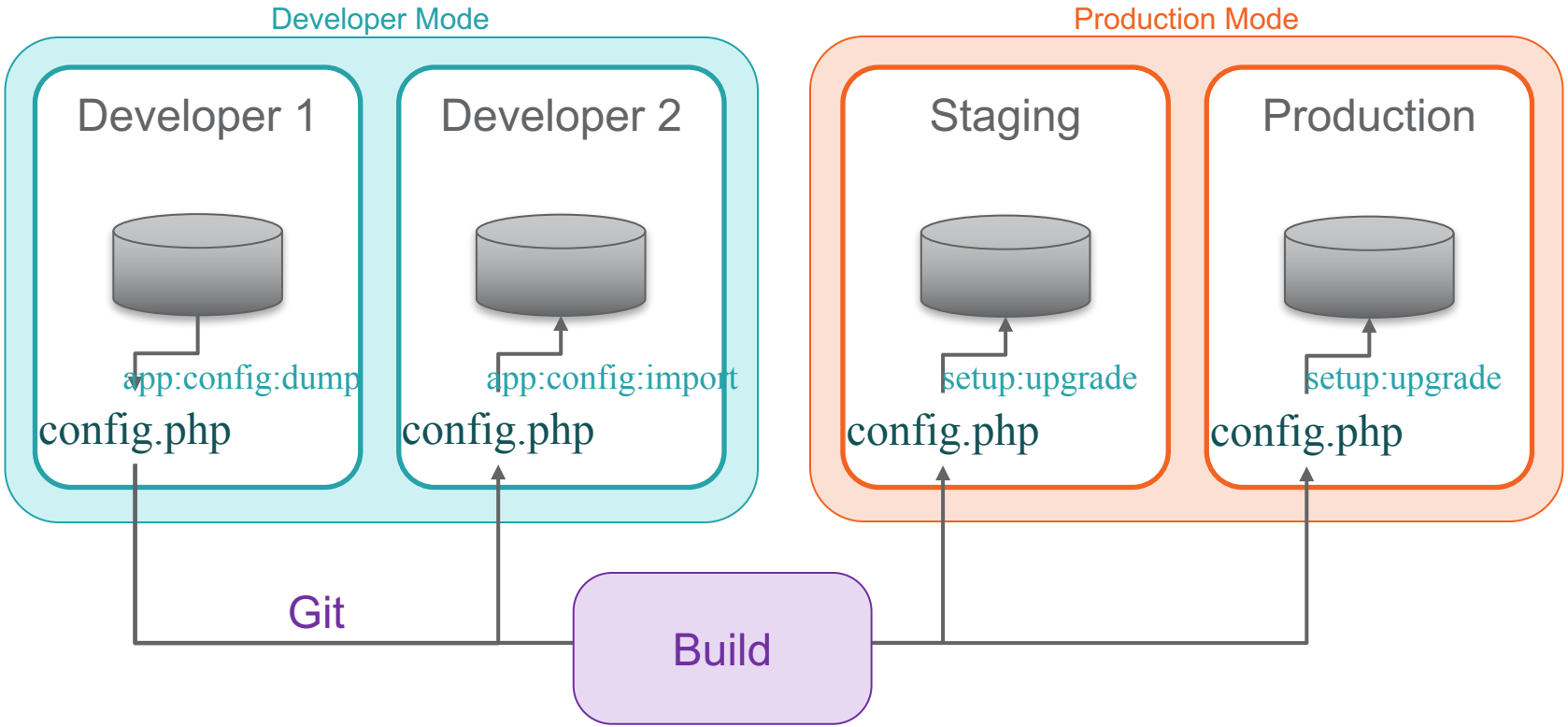
Scopes and Themes

- Websites, stores, and store views defined via Admin
  - In database for foreign key validation
- List of available themes also in database
- Needed by build phrase
  - Export database content to file for deployment



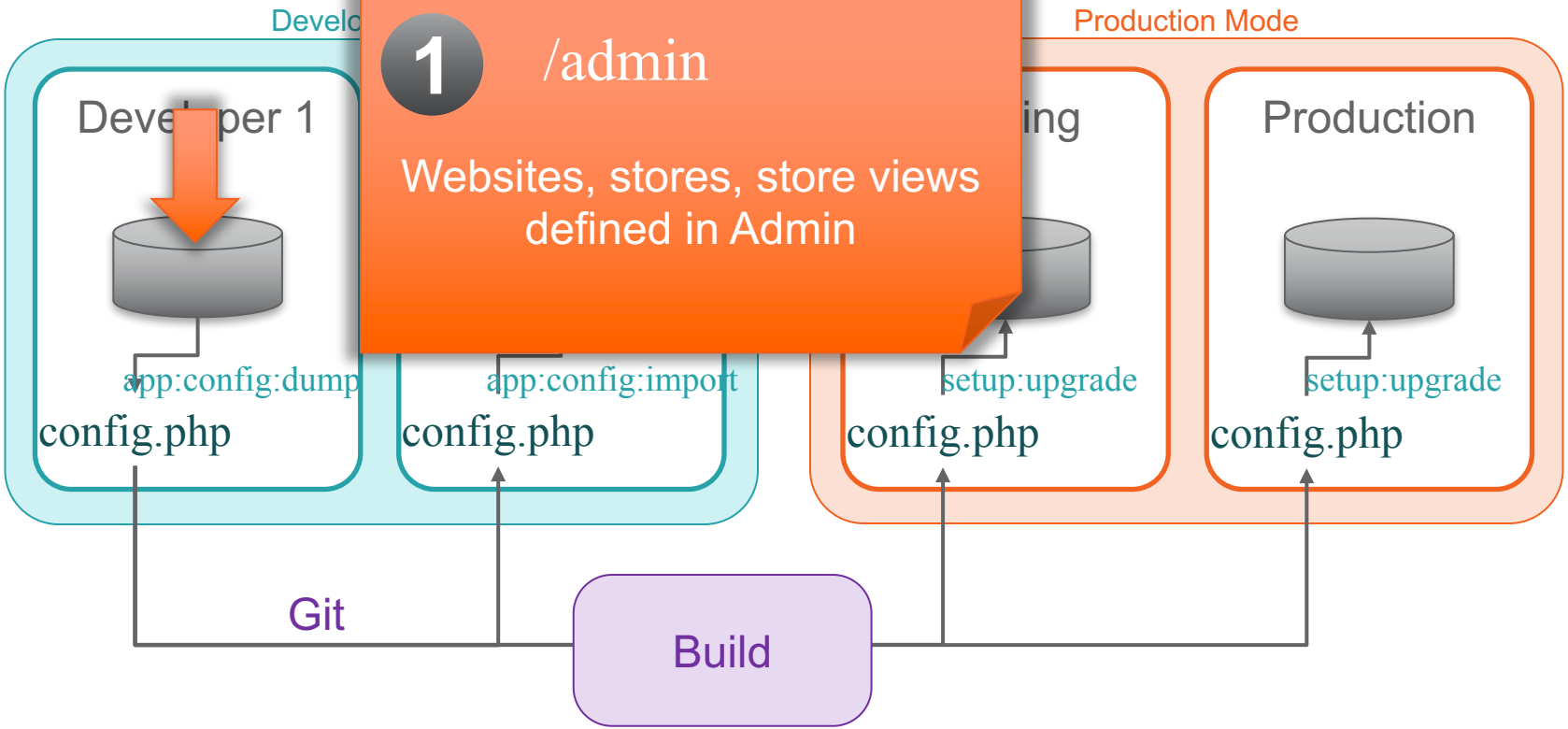
© 2017 Magento, Inc.

Scopes and Themes Sections



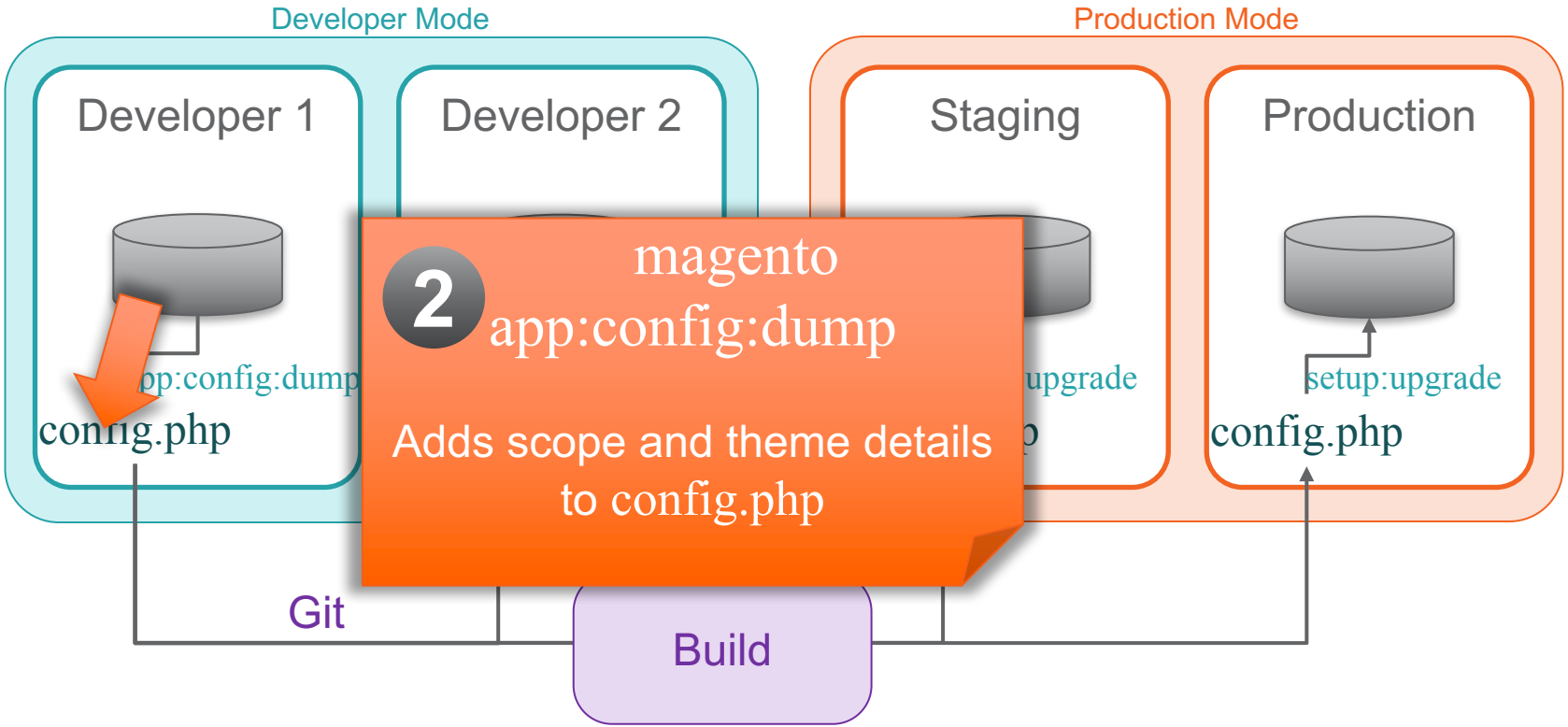
© 2017 Magento, Inc.

Scopes and Themes Sections



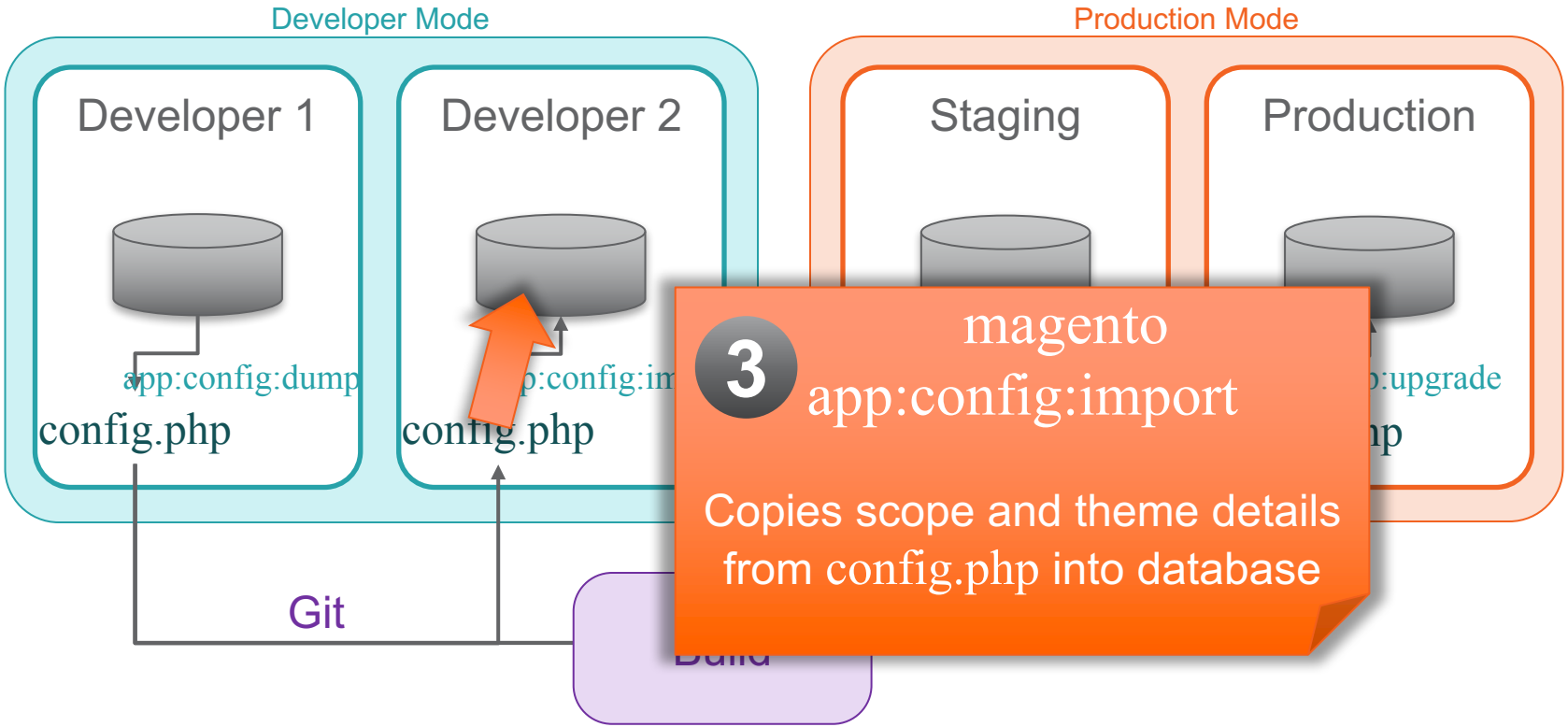
© 2017 Magento, Inc.

Scopes and Themes Sections



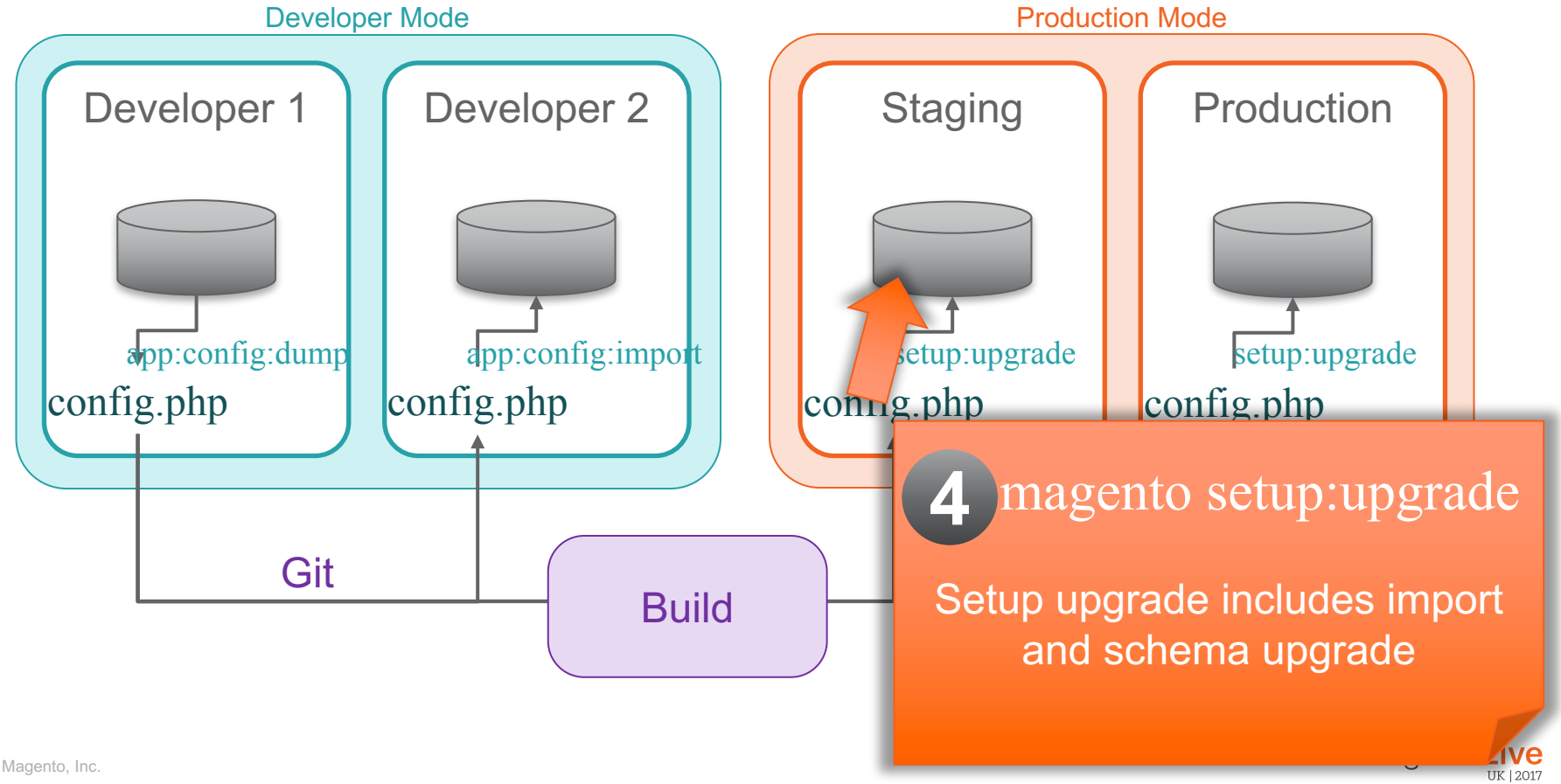
© 2017 Magento, Inc.

Scopes and Themes Sections



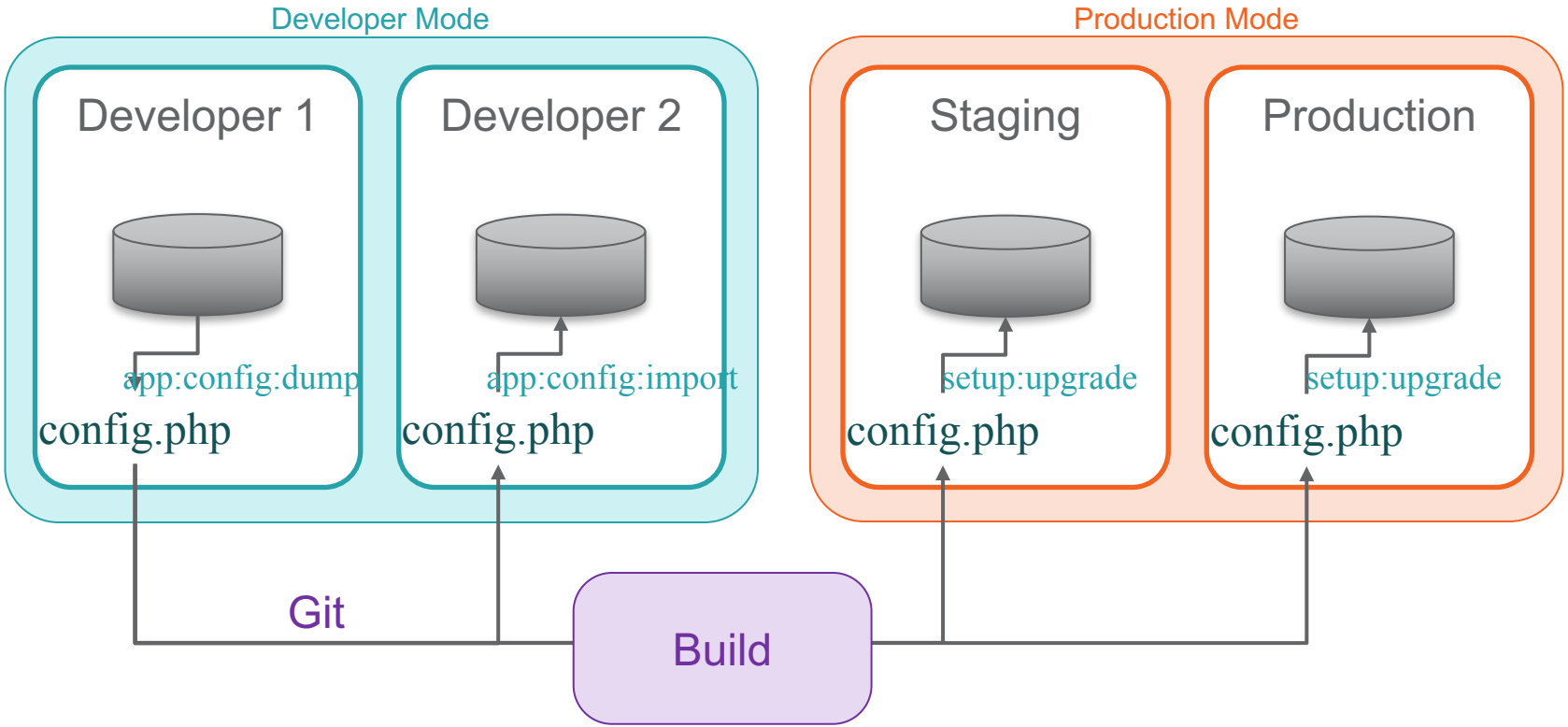
© 2017 Magento, Inc.

Scopes and Themes Sections



© 2017 Magento, Inc.

Scopes and Themes Sections



© 2017 Magento, Inc.

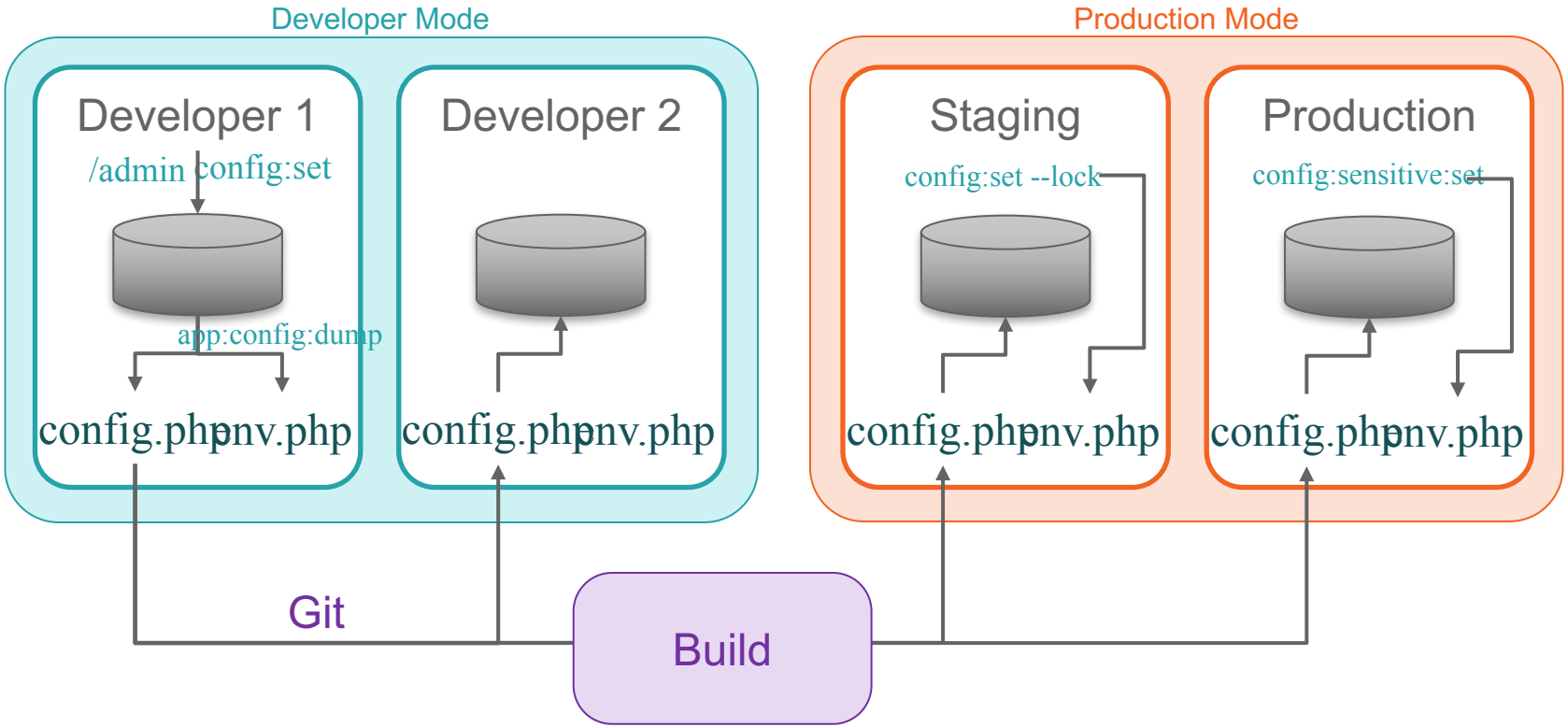
System Configuration Settings Fallback

1. Configuration values in environment variables
2. Configuration values in environment-specific config file (env.php)
3. Configuration values in shared config file (config.php)
4. Configuration values in DB
5. Configuration default value (config.xml)

Settings in config.php and env.php are "locked" - cannot be changed via Admin

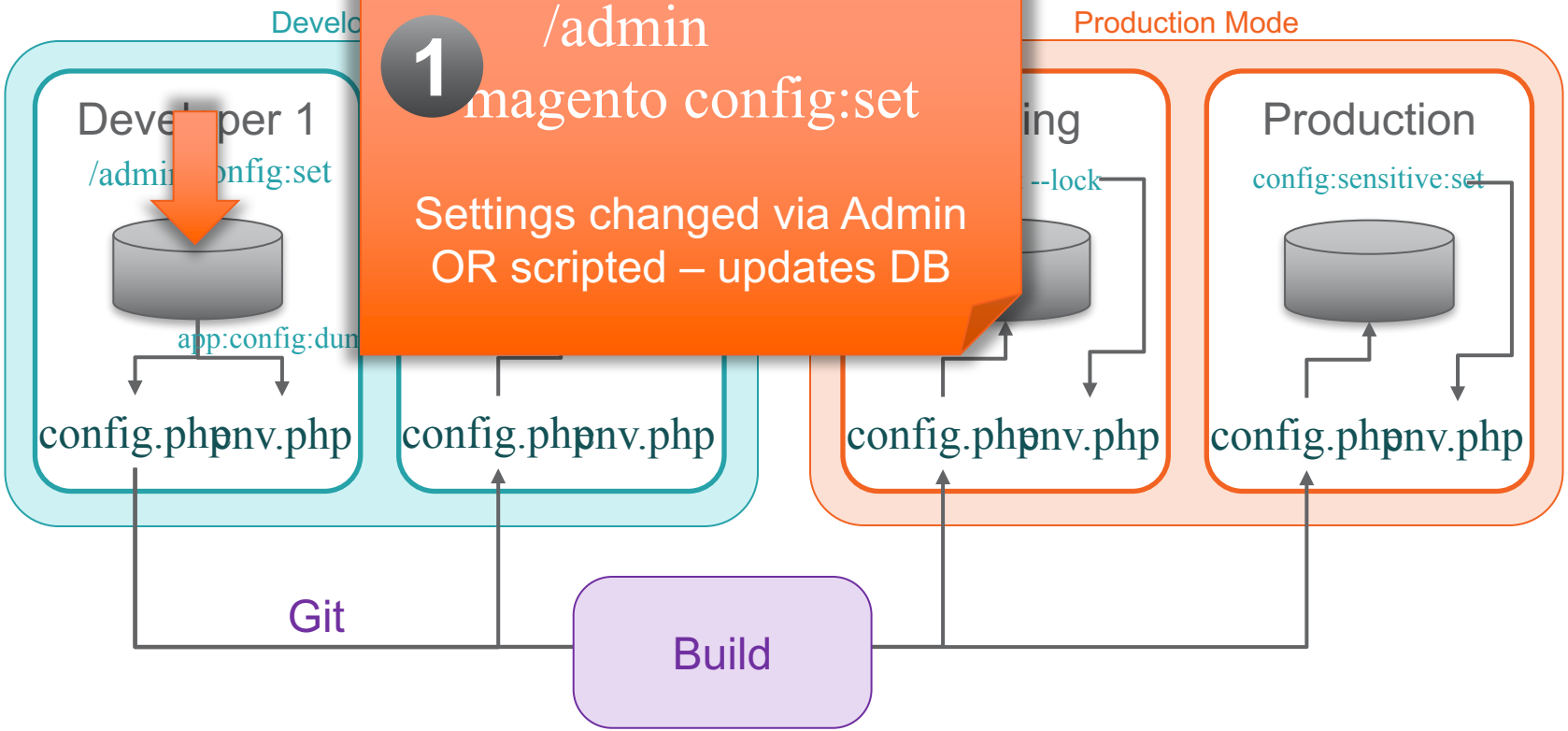
© 2017 Magento, Inc.

System Configuration Settings



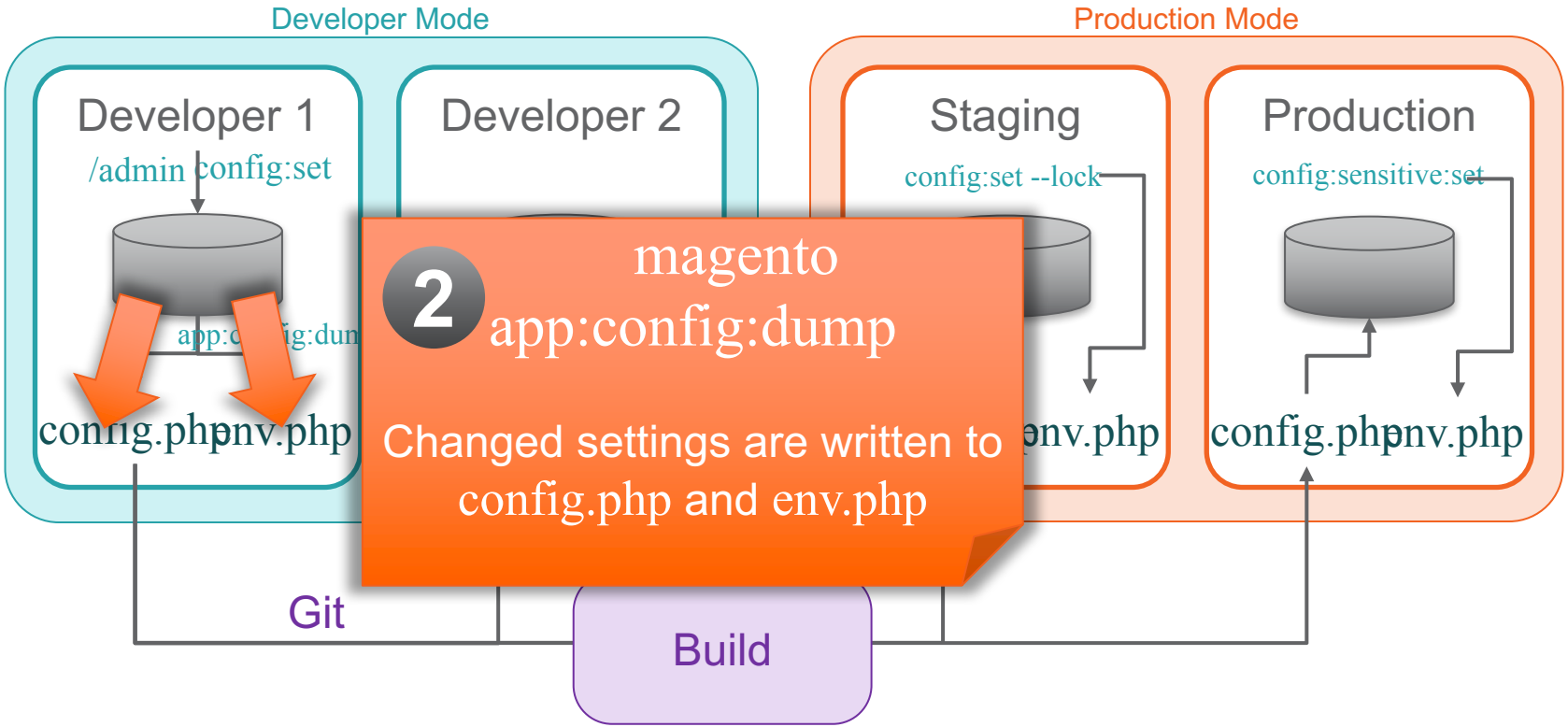
© 2017 Magento, Inc.

System Configuration Settings



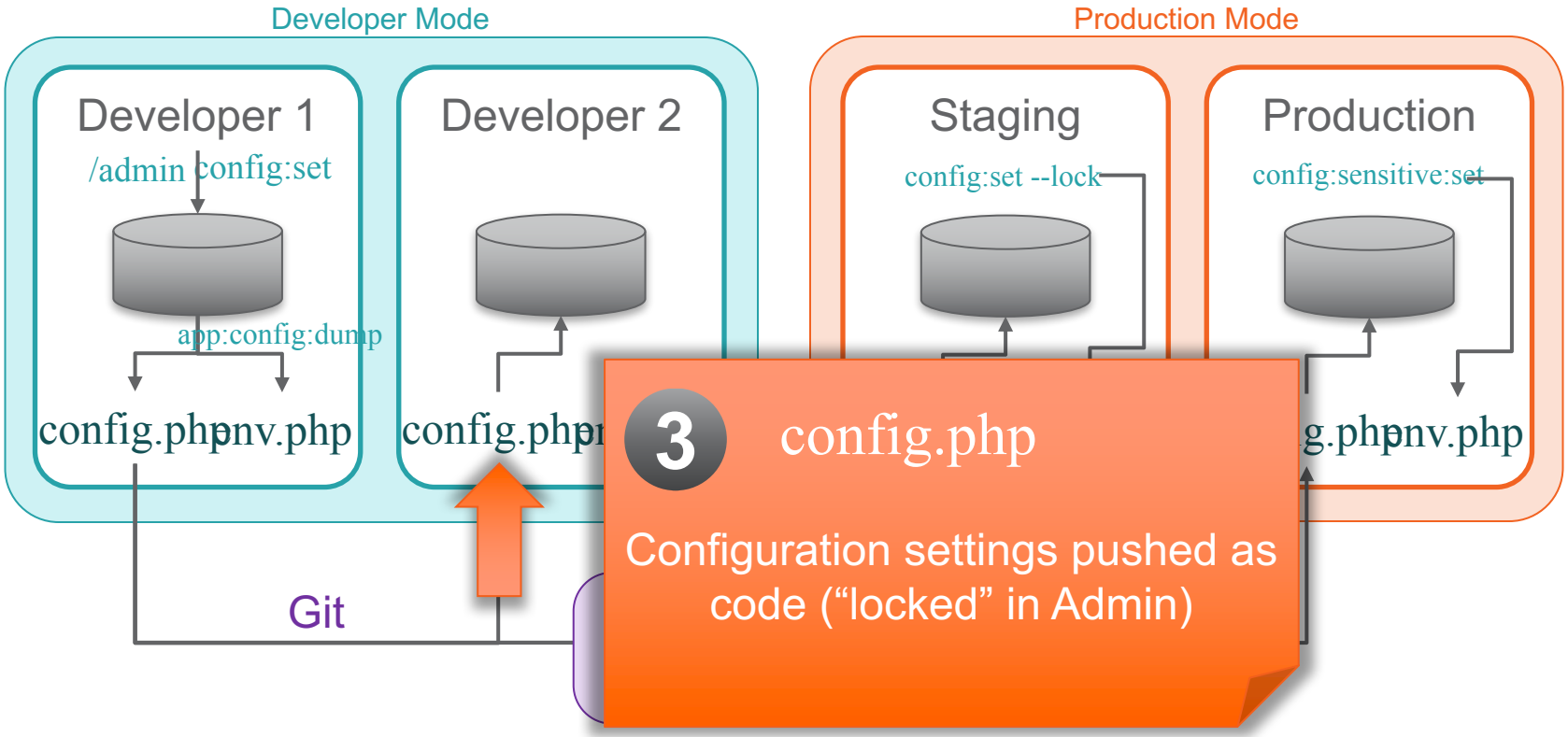
© 2017 Magento, Inc.

System Configuration Settings



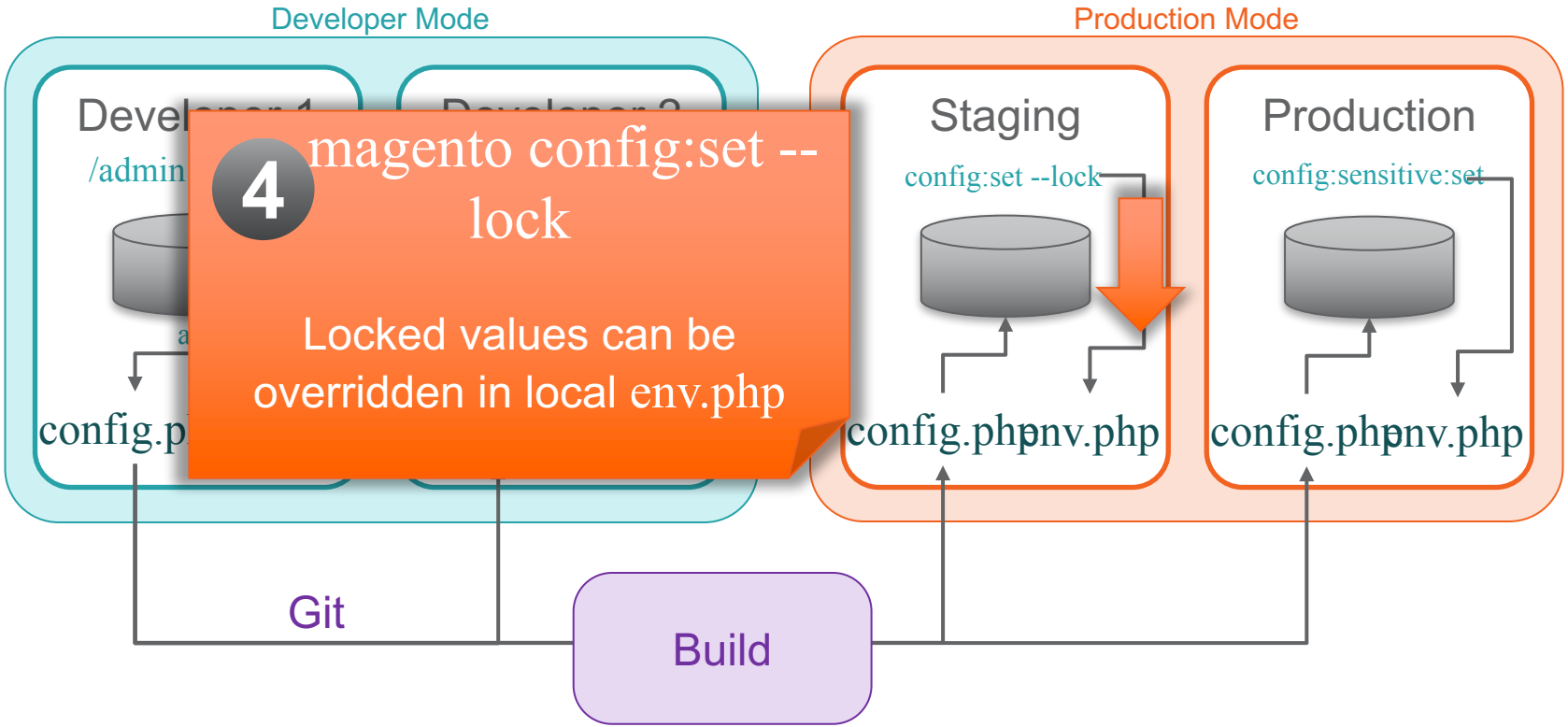
© 2017 Magento, Inc.

System Configuration Settings



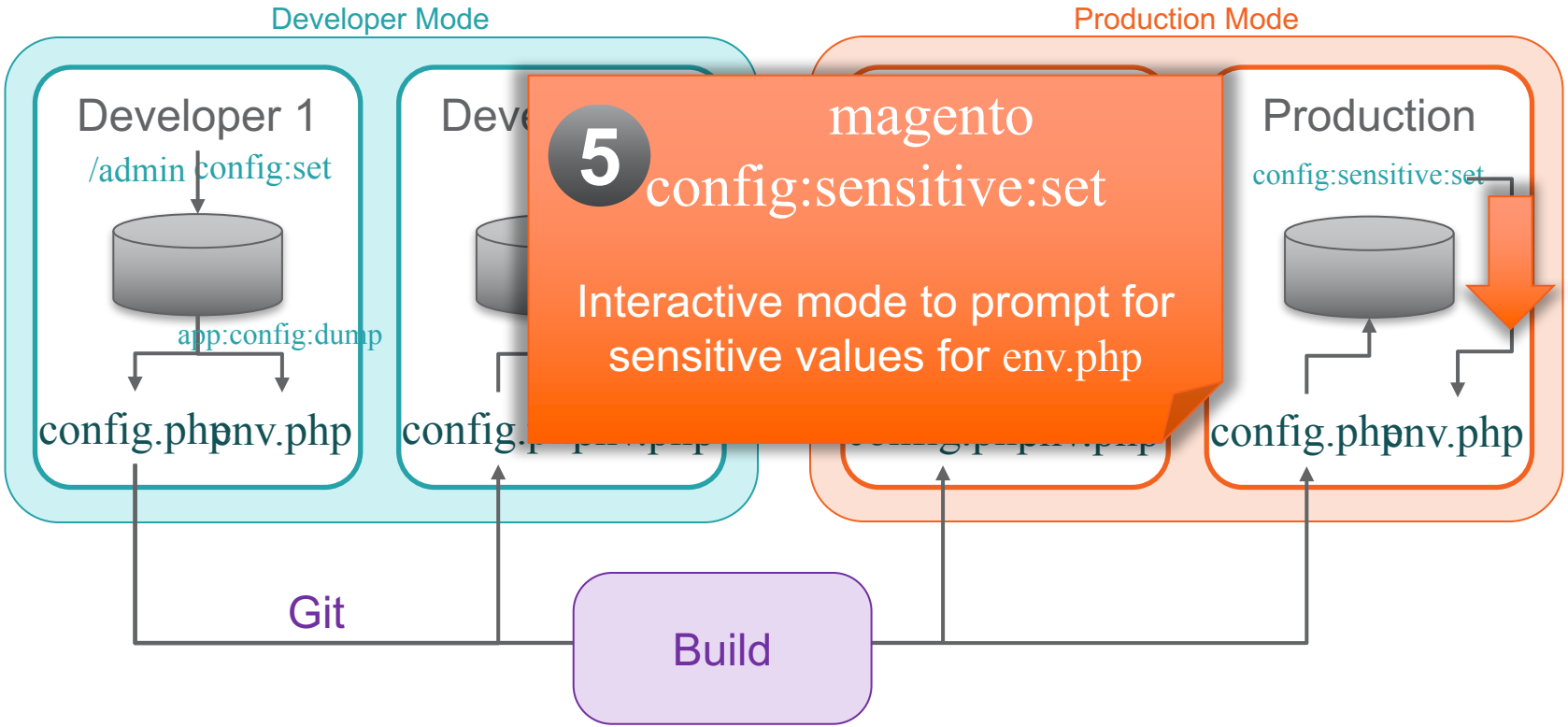
© 2017 Magento, Inc.

System Configuration Settings



© 2017 Magento, Inc.

System Configuration Settings



© 2017 Magento, Inc.

Other Recommendations

Source Code Management Recommendations

Marketplace

composer require ...

Git Repository

composer.json

composer.lock

vendor

Project

Build Process

Production

Git Repository

Modules

Alternative: publish to a private packagist repo ***https://packagist.com/***

composer install --prefer-source

VS

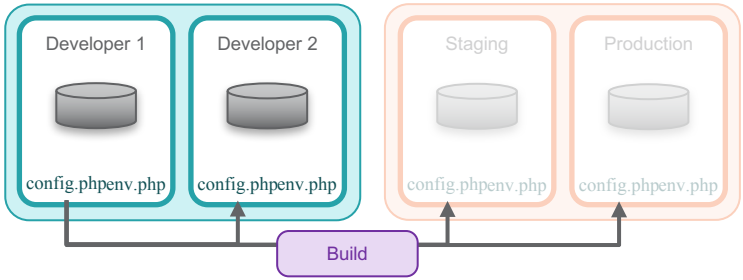
```
"config": {
    "preferred-install": {
        "alankent/*": "source",
        "*": "dist"
    }
}
```

Caches well

© 2017 Magento, Inc.

Build & Bundle Recommendations

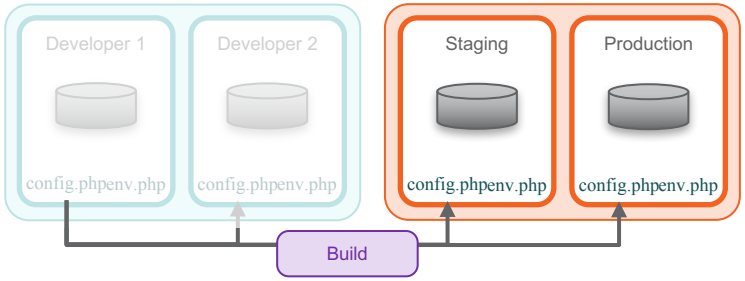
- Build process
  - Think about how to update tool chain, or customize per project
  - Git clone/pull project repo to get the code – needs repo keys
  - composer install – keep cache between runs for speed/traffic
  - Decide if test files should be in deployment bundle
- Test Automation – run functional tests on the production deployment build
- Separate Integration Test & Production environments?
  - Want environments close to each other
  - Minimize env.php (keep environments as close as possible)



© 2017 Magento, Inc.

Deployment Recommendations

- Read-only file system (/var and /generated now separate)
- Plan out your configuration settings per environment
- Remember config.php and env.php are executable code
- Have build phase create common bundle to share amongst environments
- Swap files in – use symlink, change config file, blue/green servers
- Consider serving stale content from cache during cache flush
- Decide where to deploy
  - Magento Cloud
  - Zerolag, Nexcess, Rackspace, Byte, Simple Helix - more done for you
  - Docker / AWS / Azure - set it up yourself



© 2017 Magento, Inc.

Conclusions

Workflow Review – Developer Environment

1. Use Admin to define websites, stores, store views in database
2. Enable/disable modules via config.php
3. Developer uses Admin or config:set to set store configuration settings
4. Developer runs app:config:dump to create config.php and env.php
   - Shared → config.php
   - Environment specific (sandbox vs live endpoints, …) → env.php
   - Sensitive (passwords, PII, …) setting names → config.php, values -> env.php
5. The config.php file is committed to git for sharing

© 2017 Magento, Inc.

Workflow Review – Target Environments

1. Get updated copy of config.php along with rest of latest code
2. Run app:config:import to update scopes and themes in database
3. config:sensitive:set can be used to set any missing sensitive settings
4. config:set will not override “locked” values already in config.php/env.php
5. config:set --lock will add value to env.php even if “locked”
6. Update database schema, if needed (with store in maintenance mode)
7. Flush caches, go live

© 2017 Magento, Inc.

Future

- Command to detect if schema changed for optimized deployments
- Declarative schemas to make schema changes easier
- Further standardization of
  - Development environment (DevBox still in beta)
  - Build pipeline commands
  - Test automation invocation (and other improvements)
  - Hosting partner specific deployment processes

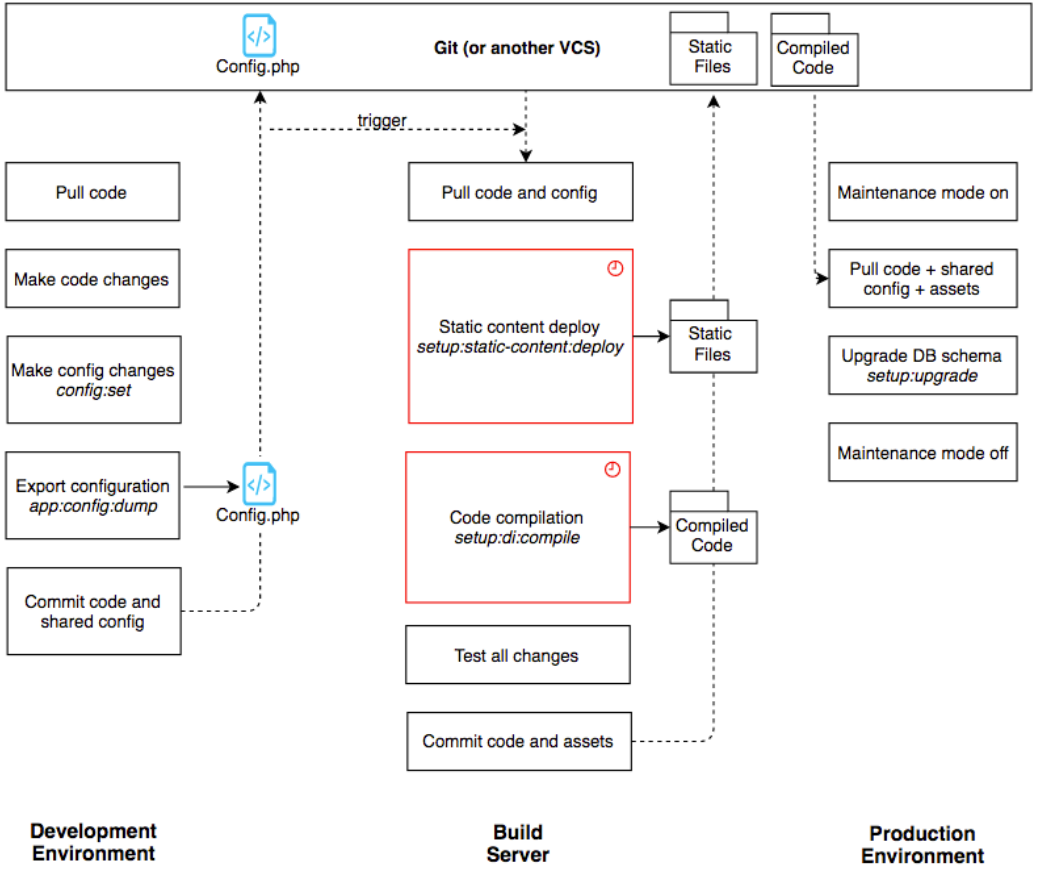
© 2017 Magento, Inc.

Q&A

Alan Kent
@akent99

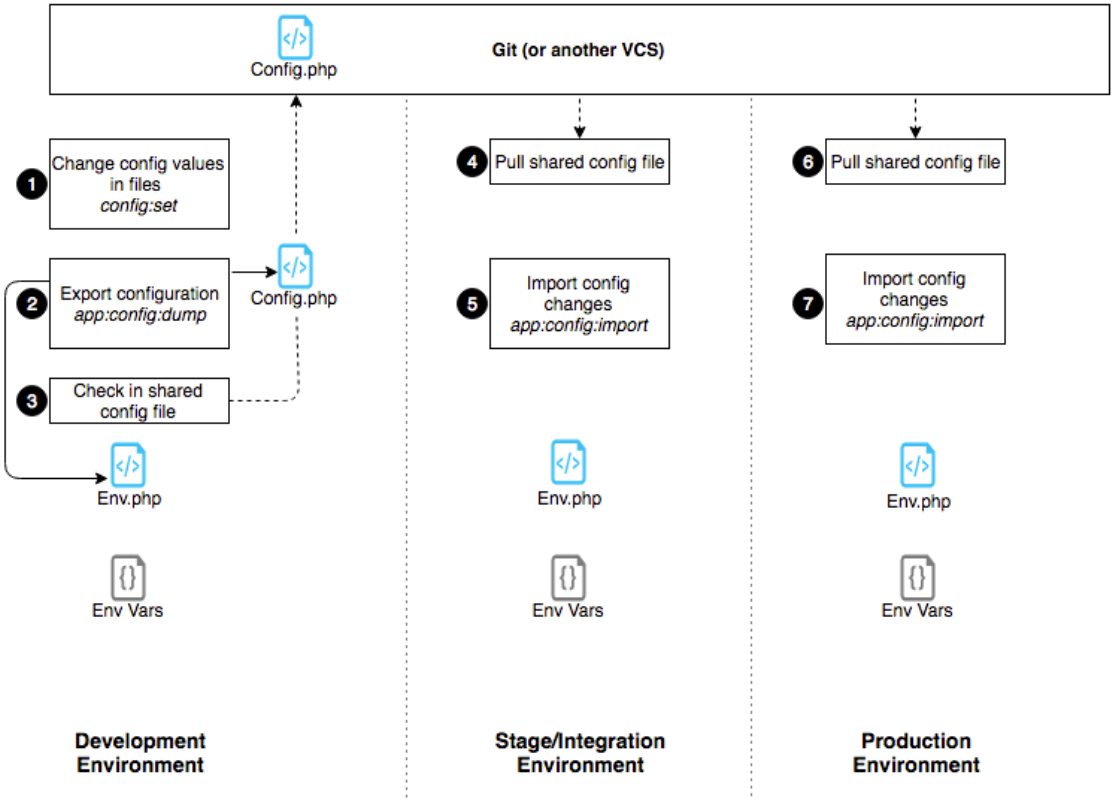
Pipeline Deployment

Deployment flow

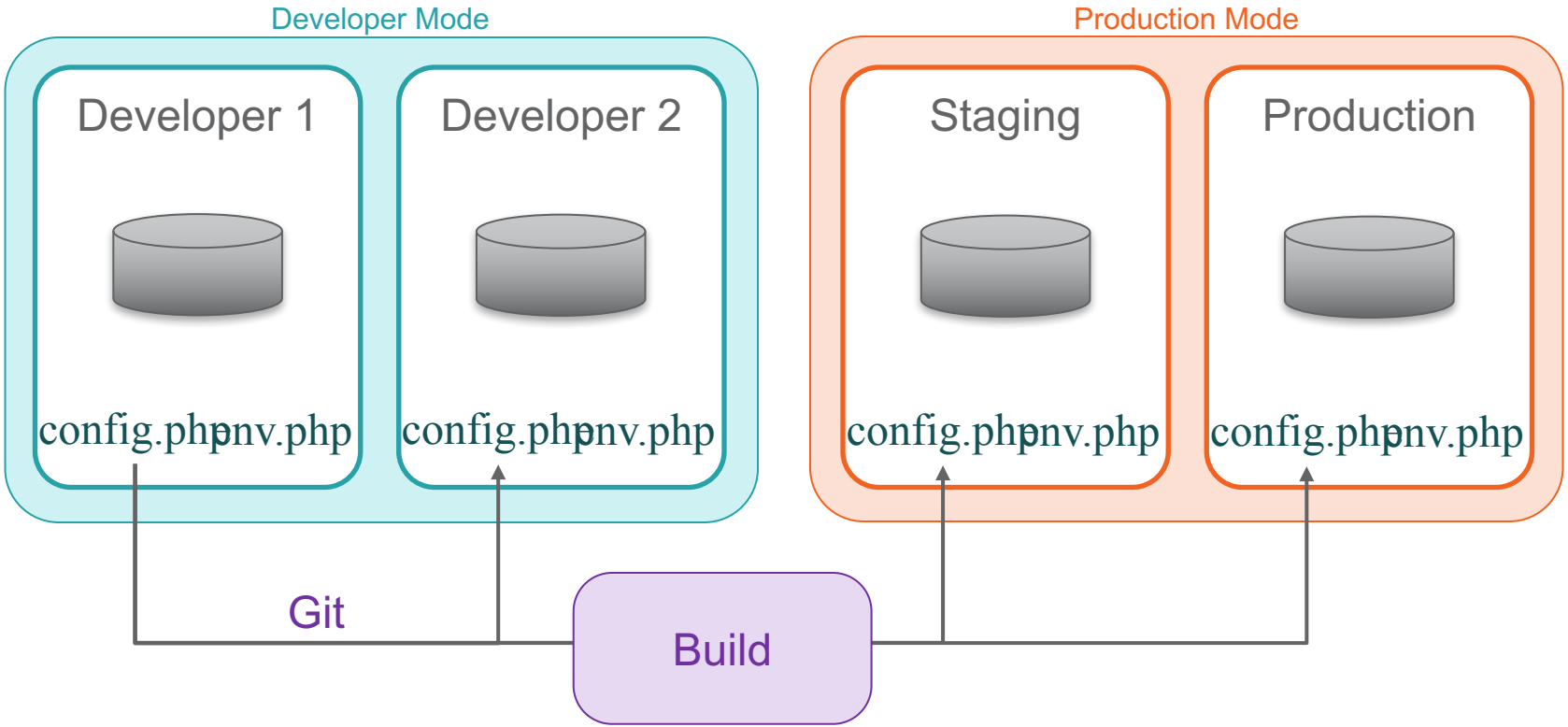


Pipeline Deployment

Configuration Management



Sample Developer and Production Environments



© 2017 Magento, Inc.

Overview of Steps

TODO: CONFUSING SLIDE – REDO?
DEFINE LOCKING EARLIER

Development Mode Settings

Develop

Identify per environment settings (including those used by extensions)
Commit shared configuration (config.php) to git

Production Mode Settings

Build & Bundle

Depends on production shared production mode settings
(Must not depend on dev or specific prod environment settings)

Deploy to Test/Staging

Environment variables, env.php, (config.php), database settings

Deploy to Prod

Environment variables, env.php, (config.php), database settings

© 2017 Magento, Inc.

Overview of Steps

Development Mode Settings

Develop

Identify per environment settings (including those used by extensions)
Commit shared configuration (config.php) to git

Production Mode Settings

Build & Bundle

Depends on production shared production mode settings
(Must not depend on dev or specific prod environment settings)

Deploy to Test/Staging

Environment variables, env.php, (config.php), database settings

Deploy to Prod

Environment variables, env.php, (config.php), database settings

© 2017 Magento, Inc.

Overview of Steps

Development Mode Settings

Develop

Identify per env … those used by extensions)

Commit shared … git

Produ… Mode Settin…

… ction shared product…

… n dev or specific pro…

Developer controlled
Shared settings

Developer controlled
Per environment settings

Merchant controlled
Per environment settings

Deploy to Test/Staging

Environment variables, env.php, (config.php), database settings

Deploy to Prod

Environment variables, env.php, (config.php), database settings

© 2017 Magento, Inc.


Magento Live
UK | 2017