]init[
Digitale Kommunikation

Streamline your drupal development workflow in a 3-tier-environment

- A story about *drush make* and *drush aliases*

thomas.bussmeyer@init.de
Berlin, 18.09.2011

1. Who we are
2. Scenario
3. Solution
4. Notes

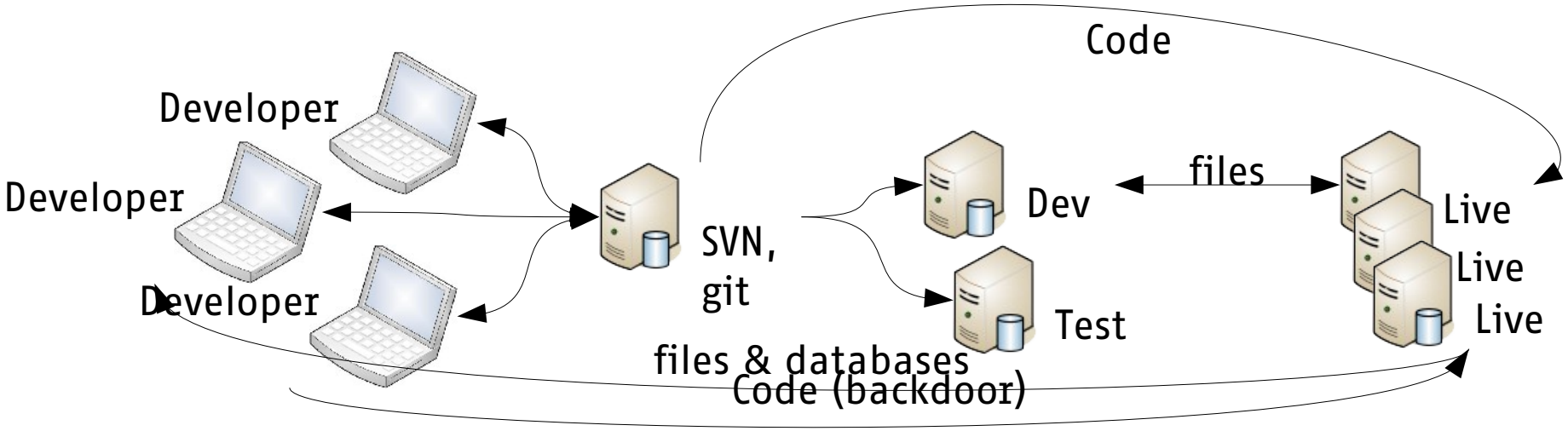
Who we are

Have a look at

http://www.init.de

Scenario

A typical set-up



Trunk? Maybe.
Database: from july
Files: some files missing

Code? Trunk.
Database: an old one. I don't know.
Plenty of files missing

Code version? Don't know
Current database
Current files

Scenario

Summary

→ The set-up is quite well.

(Version control is a good idea. Not to hack live, too. Teamwork with developers is fun.)

→ But it's so slow and too much manual work.

(You won't do it every time)

→ And a controlled workflow and automated documentation is missing

(“I know what version is on all my boxes. They all have the current database and files on it.”)

Scenario

Two questions (especially after launch)

- How do I get my new developments quickly and repeatedly to the development and test servers, finally to the live system?

  (Automated, every time in the same way, and fast. The project managers and customers are curious. Me too.)

- How do I ensure that I develop and test under realistic conditions on the development and test system?

  (I would like to test my new code with the „live system“.)

Scenario

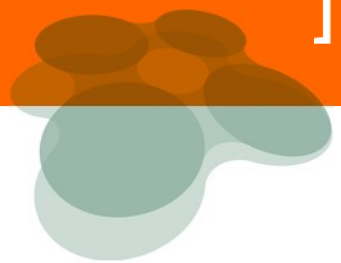
and two ~~presumptions~~ rules

- Configuration: Everything lives in code and code goes up (Local->SVN, SVN->Dev->Test->Live)!

  (So get used to features, strongarm and the drupal api)

- Database and files go down (Live->Test, Live->Dev, Live->Local).

  (Don't touch the live files and database after launch! You won't migrate content.)



Solutions

Solution? Again two things

→ Introducing: *drush make* and *drush aliases*

(and some little helper shell scripts)

→ Examples:

```
$ drush make project.make build_20110918-01
$ drush sql-sync  @live [@local, @dev, @test]
```

Solutions

drush make – the answer to no. 1

- drush make is an extension to drush that can create a ready-to-use drupal sites

  (You might set up a special build server for the build tasks.)

- It pulls sources from various locations. You can describe it as a build tool for drupal.

  (Only custom code in you repo. For all the other bricks and pieces you use the official repos.)

Solutions

drush make – what does it look like

```
; Drush Make file

core = 7.x

api = 2

; MODULES

projects[admin_menu][version] = "3.x-dev"

projects[admin_menu][subdir] = adm

projects[advanced_help][version] = "1.x-dev"

projects[advanced_help][subdir] = dev


projects[devel][version] = "1.x-dev"

projects[devel][subdir] = dev

[...]
```

Solutions

drush make – it goes even further

```
; custom svn, git or libraries or patches

; Features

projects[atrium_features][type] = "module"
projects[atrium_features][download][type] = "git"
projects[atrium_features][download][url] = "http://github.com/phase2/a
projects[atrium_features][download][tag] = "6.x-1.0"

; Libraries

libraries[jquery_ui][download][type] = "get"
libraries[jquery_ui][download][url] = "http://jquery-ui.googlecode.com
libraries[jquery_ui][directory_name] = "jquery.ui"
libraries[jquery_ui][destination] = "modules/contrib/jquery_ui"

; and patches, great!
; To apply a patch to a project, use the `patch` attribute and pass in
; of the patch.

projects[admin_menu][patch][] = "http://drupal.org/files/issues/admin_

[...]
```

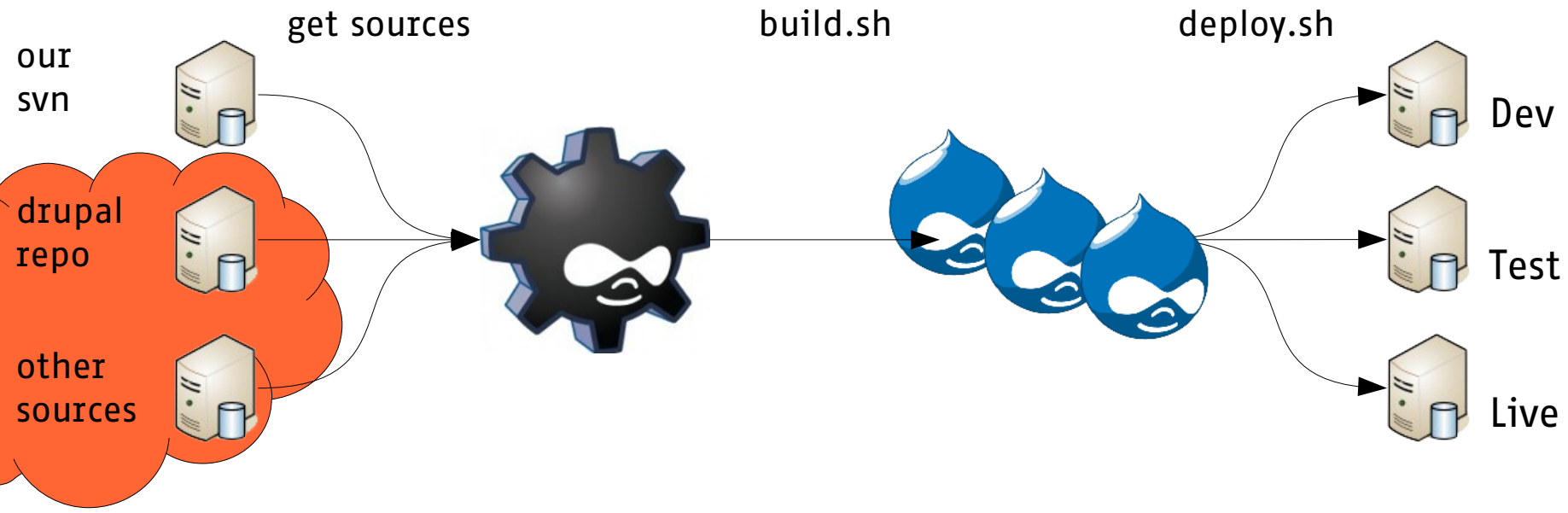
Solutions

Our repo is a drupal profile

- projectdir/
  - config
  - libraries (only custom)
  - modules (only custom module and features)
  - themes (only custom)
  - *project.make*
  - *project.profile*

Solutions

Code upstream – the big picture



Solutions

And some little helper scripts

- build.sh, on build server
  - Run drush make
  - Copy setting files
  - Delete unnecessary files
  - Make a build, e.g. *build_20110918-01.tar.gz*
- deploy.sh, on dev, test, live server
  - Backup database and current code
  - Get e.g. *build_20110918-01* from build server and untar it
  - Symlink files in new build
  - Switch symlink to new build

Scenario

Reminder: The two questions

- How do I get my new developments quickly and repeatedly to the development and test servers?

  (Automated every time in the same way and fast. The project managers and customers are curious. Me too.)

- How do I ensure that I develop and test under realistic conditions on the development and test system?

  (I would like to test my new code with the „live system“.)

Scenario

Reminder: The ~~two~~ questions

- ~~How do I get my new developments quickly and repeatedly to the development and test servers?~~

  ~~(Automated every time in the same way and fast. The project managers and customers are curious. Me too.)~~

- How do I ensure that I develop and test under realistic conditions on the development and test system?

  (I would like to test my new code with the „live system“.)

Solutions

drush aliases – the answer to no. 2

- → Define all your hosts in your enviroment (even the remote ones)
- → Move *files* and *databases* around with a single drush command.

Solutions

drush aliases – what does it look like

```
$aliases['dev'] = array(
  'uri' => 'dev.local',
  'root' => '/Sites/dev.local',
  'path-aliases' => array(
    '%files' => 'sites/example.com/files',
    '%dump' => '/tmp/drush/sql-sync-dev-local.sql',
  ),
  'command-specific' => array (
    'sql-sync' => array (
      'simulate' => '0',
      'structure-tables' => array(
        'custom' => array(
          'cache', 'cache_filter', 'cache_menu', 'cache_page', 'histor
        ),
       ),
    ),
    'rsync' => array (
      'simulate' => '0',
    ),
  ),
);
```

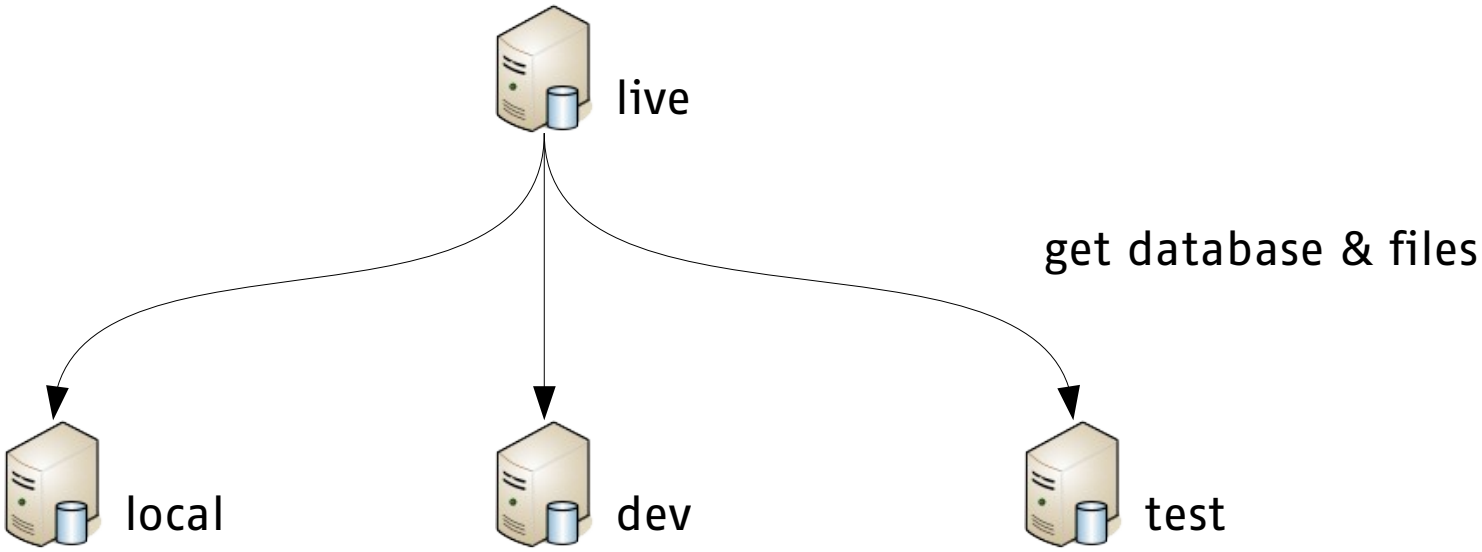
Solutions

drush aliases – what does it look like

```
$aliases['prod'] = array(
  'uri' => 'prod.local',
  'root' => '/Sites/prod.local',
  'path-aliases' => array(
    '%files' => 'sites/example.com/files',
    '%dump' => '/tmp/drush/sql-sync-prod-local.sql',
  ),
  'command-specific' => array (
    'sql-sync' => array (
      'simulate' => '1',
    ),
    'rsync' => array (
      'simulate' => '1',
    ),
  ),
);
```

Solutions

Data & files downstream – the big picture




```
$ drush rsync @live [@local, @dev, @test]
$ drush sql-sync  @live [@local, @dev, @test]
```

Notes

Final notes

- Get used to features, strongarm, installation profiles
- Use drush and drush make to build your application
- Use drush aliases capabilities to sync your boxes

- Drush http://drupal.org/project/drush
- Drush make http://drupal.org/project/drush_make
- http://developmentseed.org/blog/2010/jul/27/drush-make-files-production-drupal-sites/
- http://drupal.org/node/1006620

]init[
Digitale Kommunikation

Thank You

for your attention!

thomas.bussmeyer@init.de
Berlin, 18.09.2011