

DevOps Environment for an OFBiz Agile project

Nathan Boudou



APACHECON
EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
– OCTOBER 1-3, 2015 –



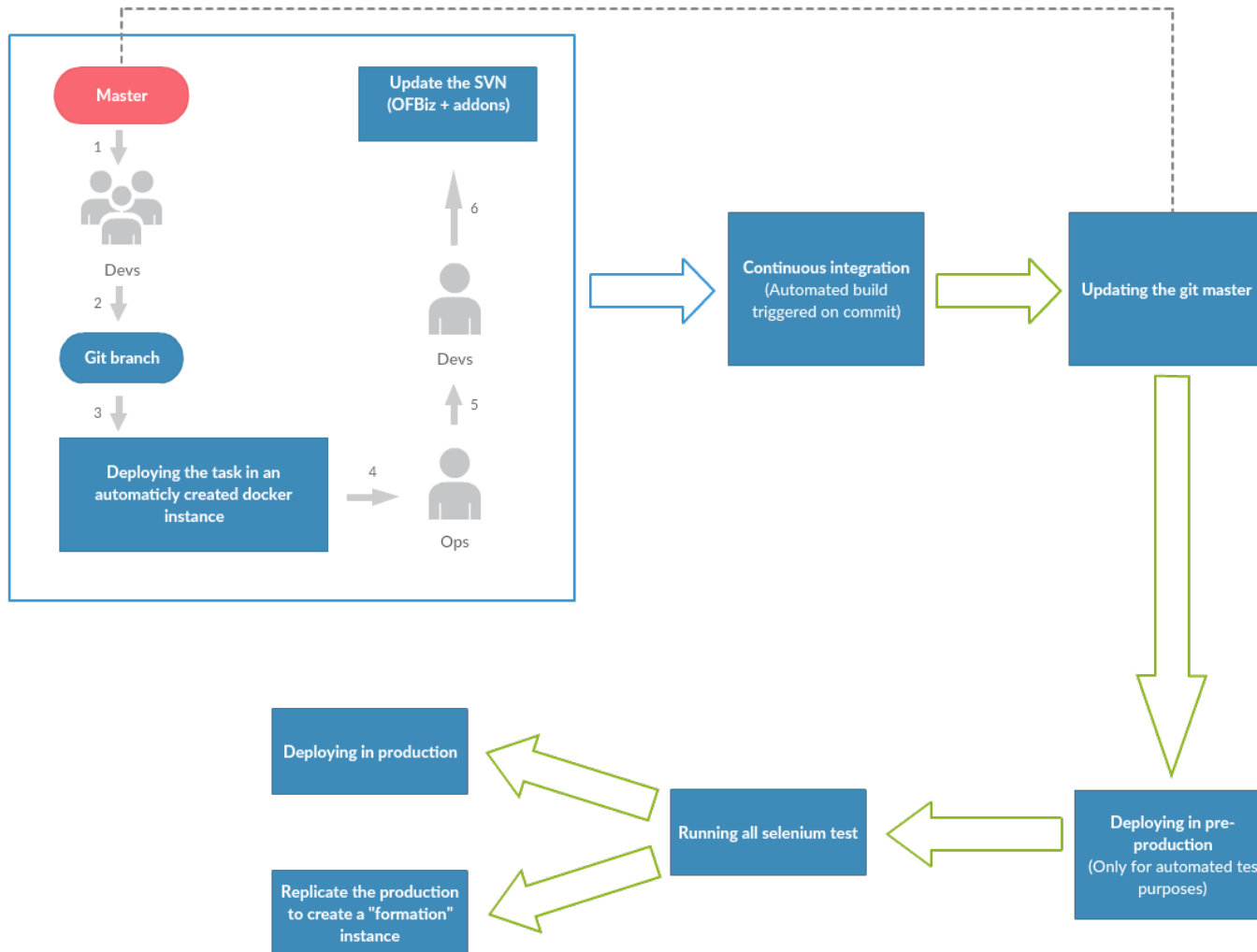
Outline

- Introduction
- Presentation of our flow
- Softwares used
- Increase your amount of releases
- Communication
- Conclusion

- 
- Presentation of our flow



Presentation of our flow





• Software used



SVN / Git

- In our case both of them
- Taking advantage of the hooks
 - Launching hooks on commits (build for example)



Jenkins

- Task automation (Seleniums, build, deploying apps...)
- Updating the master branch if the app build correctly
- Using commit hooks to start jobs
- Creating / Deleting docker instances



Docker

- Container created on the go
- For testing new functionalities
- Everything is tested on docker with a production database before going further
- Really simple to create and destroy



Selenium

- Every functionality need to have his selenium script
- Can't be released without the ability to be tested automatically
- Are launched on every new master version
- If seleniums are good, we can move to production

- 
- Increase your amount of releases



- Increase your amount of releases

- Every new functionality need to be tested quickly
- Better communication between Devs and Ops
- With the ability to have quick instances of functionalities, we can communicate faster with more efficiency



- Increase your amount of releases

- Ops validate the functionality
- Main commiter review the code (functionality + selenium) before accepting it



- Increase your amount of releases

- Enjoy automation
- New master created
- Selenium are running
- Soon in production
- More concentration on the functionalities



- Selenium is good for you

- Detect any regression for you
- Allow you to concentrate only on the current functionality in development



• Communication



- Before anything

- Use all the time saved by your automated tasks to have a better preparation of the development of new functionalities



- When working remotely.. or not

- You need to have a bunch of tools to allow a good communication between the member of your team



• Conclusion



• Conclusion

- There is no setup better than the one you need
- Experiment things and involve your team to improve your DevOps environment
- Use automation, a lot, really



• Thank you