BuildBot

And Continuous Integration RegCM4 experiences

S.Cozzini/A.Messina/G.Giuliani

Warning: Some slides/ideas stolen by Willie <willie@issdu.com.tw>

Agenda

- How do we use BuildBot here?
- What is BuildBot?
- Practices of Continuous Integration
- Example for RegCM code

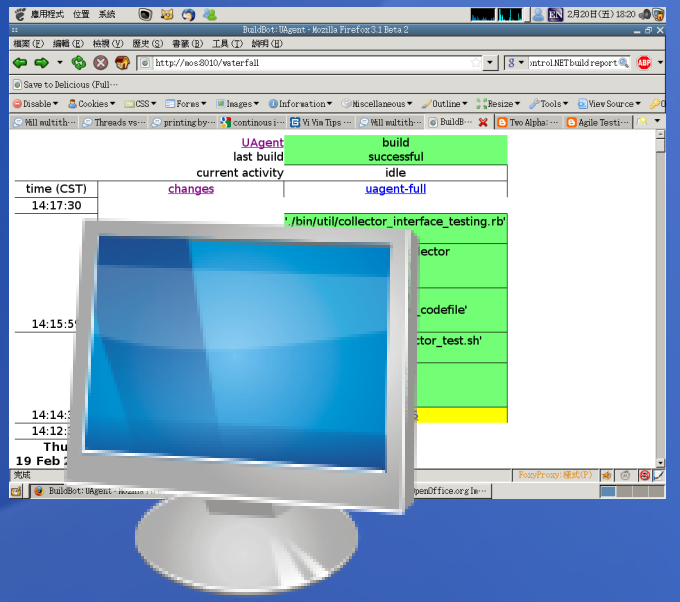
How do we use BuildBot?

2. Commit

3. Trigger

4. Wait & See

Email If Fail



1. make a change on RegCM

Subject: buildbot failure in ICTP BuildBot on
regcm-conf=debug|mod=openmpi:1.4.3:intel:2011
Date: Tue, 21 Feb 2012 22:20:39 +0100
From: buildmaster@gforge.ictp.it
To: amessina@ictp.it, ggiulian@ictp.it
The Buildbot has detected a failed build on builder
regcm-conf=debug|mod=openmpi:1.4.3:intel:2011 while building ICTP
BuildBot.
Full details are available at:

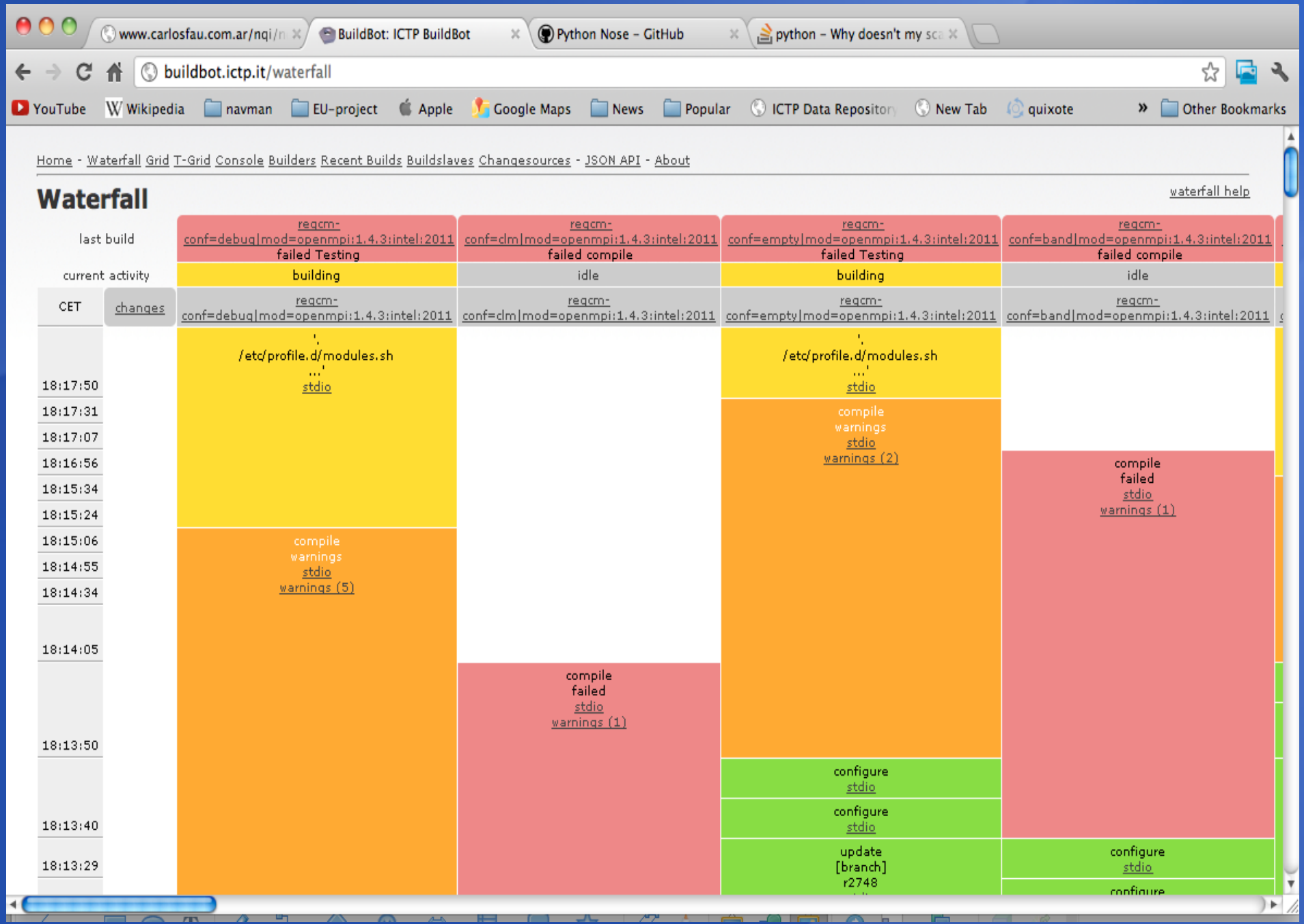
http://buildbot.ictp.it/builders/regcm-conf%3Ddebug%7Cmod%3Dopenmpi%3A1.4.3%3Aintel%3A2
Buildbot URL: http://buildbot.ictp.it/
Buildslave for this Build: argo-buildbot01
Build Reason: The web-page 'force build' button was pressed by '':
Build Source Stamp: 2750
Blamelist:
BUILD FAILED: failed compile
sincerely,
 -The Buildbot

What is BuildBot?

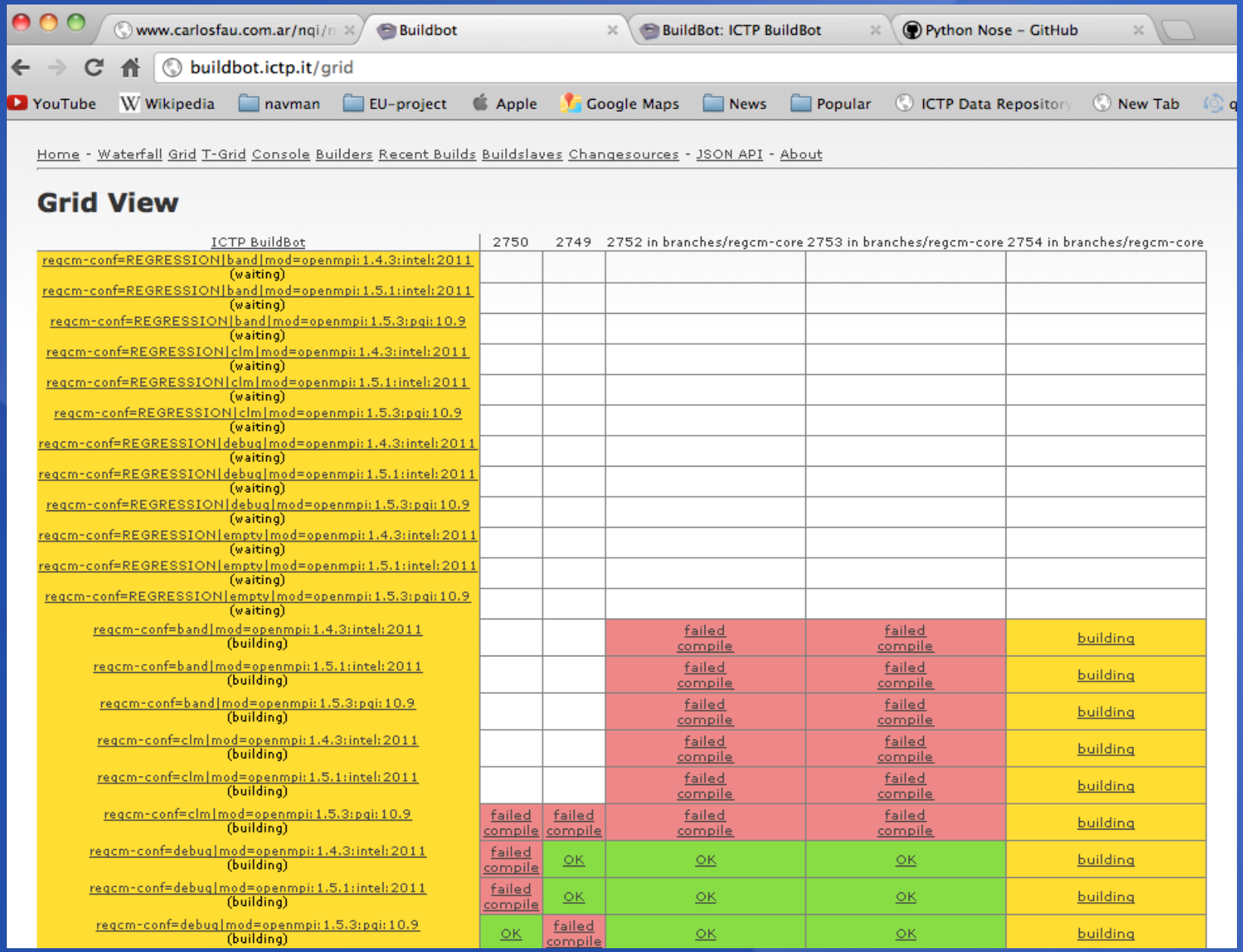
BuildBot

- A continuous integration system
- Help to automate the compile/test cycle
- Written in Python

Screenshot


www.carlosfau.com.ar/nqi/r
BuildBot: ICTP BuildBot
Python Nose - GitHub
python - Why doesn't my sca
buildbot.ictp.it/waterfall
YouTube
Wikipedia
navman
EU-project
Apple
Google Maps
News
Popular
ICTP Data Repositor
New Tab
quixote
Other Bookmarks
Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources - JSON API - About
Waterfall
waterfall help
last build
current activity
CET
changes
reqcm-conf=debug|mod=openmpi:1.4.3:intel:2011
failed Testing
reqcm-conf=clm|mod=openmpi:1.4.3:intel:2011
failed compile
reqcm-conf=empty|mod=openmpi:1.4.3:intel:2011
failed Testing
reqcm-conf=band|mod=openmpi:1.4.3:intel:2011
failed compile
building
idle
building
idle
18:17:50
18:17:31
18:17:07
18:16:56
18:15:34
18:15:24
18:15:06
18:14:55
18:14:34
18:14:05
18:13:50
18:13:40
18:13:29
/etc/profile.d/modules.sh
stdio
compile
warnings
stdio
warnings (5)
compile
warnings
stdio
warnings (2)
compile
failed
stdio
warnings (1)
configure
stdio
update
[branch]
r2748

Screenshot



www.carlosfau.com.ar/nqi/n | Buildbot | BuildBot: ICTP BuildBot | Python Nose – GitHub

buildbot.ictp.it/grid

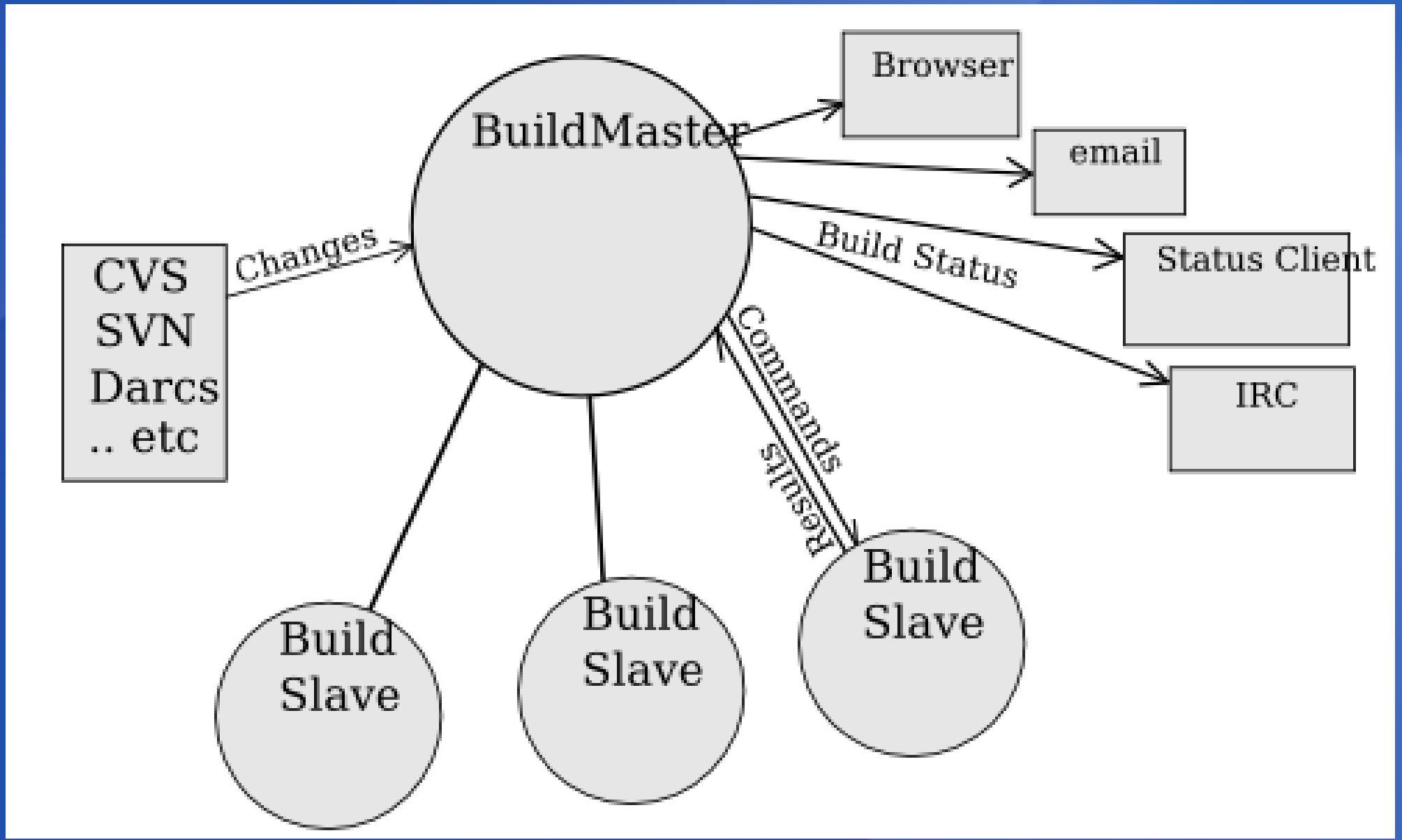
YouTube | Wikipedia | navman | EU-project | Apple | Google Maps | News | Popular | ICTP Data Repository | New Tab

Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources - JSON API - About

Grid View

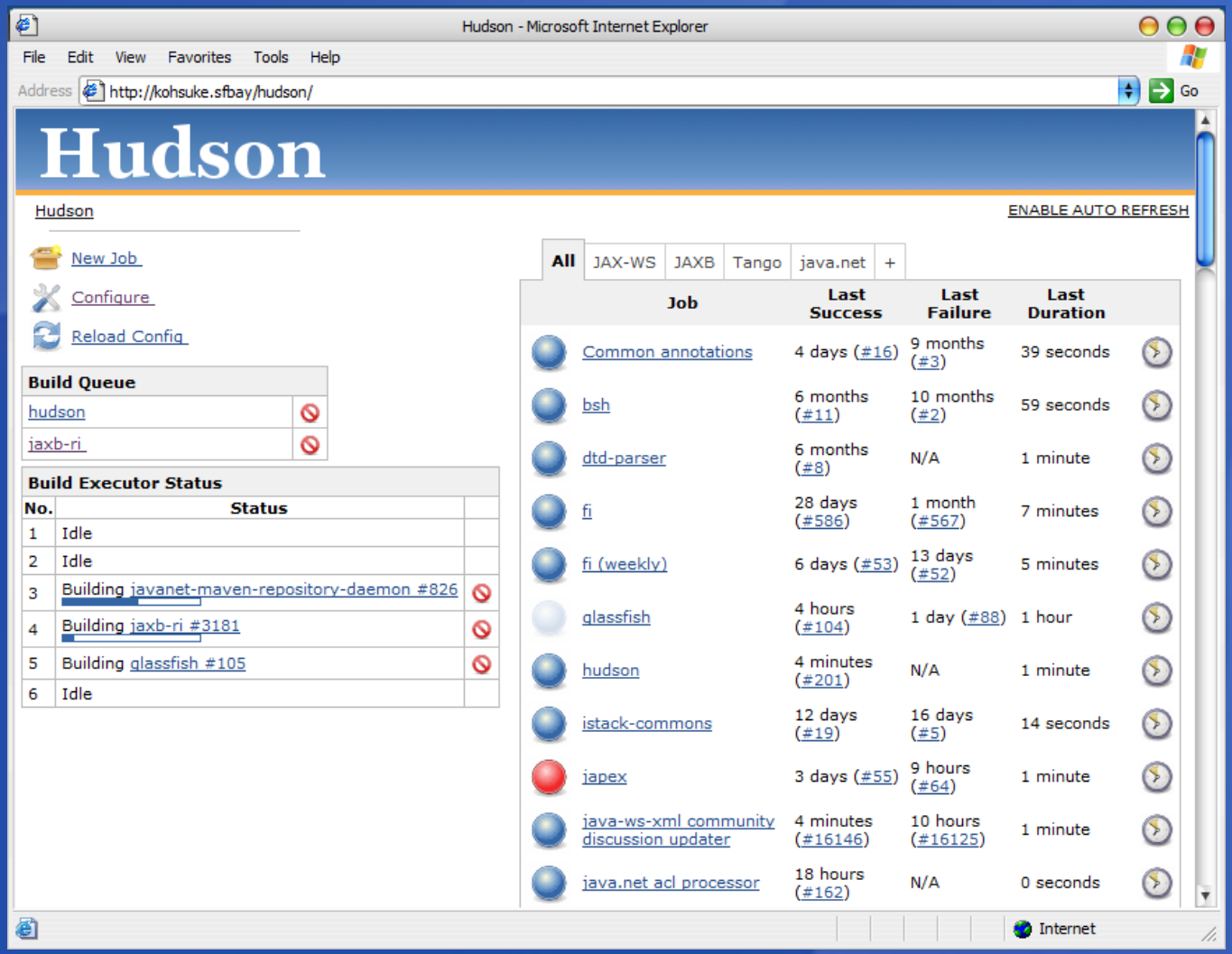
ICTP BuildBot	2750	2749	2752 in branches/regcm-core	2753 in branches/regcm-core	2754 in branches/regcm-core
regcm-conf=REGRESSION|band|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|band|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|band|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=REGRESSION|clm|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|clm|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|clm|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=REGRESSION|debug|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|debug|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|debug|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=REGRESSION|empty|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|empty|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|empty|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=band|mod=openmpi:1.4.3:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=band|mod=openmpi:1.5.1:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=band|mod=openmpi:1.5.3:pgi:10.9 (building)			failed compile	failed compile	building
regcm-conf=clm|mod=openmpi:1.4.3:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=clm|mod=openmpi:1.5.1:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=clm|mod=openmpi:1.5.3:pgi:10.9 (building)	failed compile	failed compile	failed compile	failed compile	building
regcm-conf=debug|mod=openmpi:1.4.3:intel:2011 (building)	failed compile	OK	OK	OK	building
regcm-conf=debug|mod=openmpi:1.5.1:intel:2011 (building)	failed compile	OK	OK	OK	building
regcm-conf=debug|mod=openmpi:1.5.3:pgi:10.9 (building)	OK	failed compile	OK	OK	building

System architecture



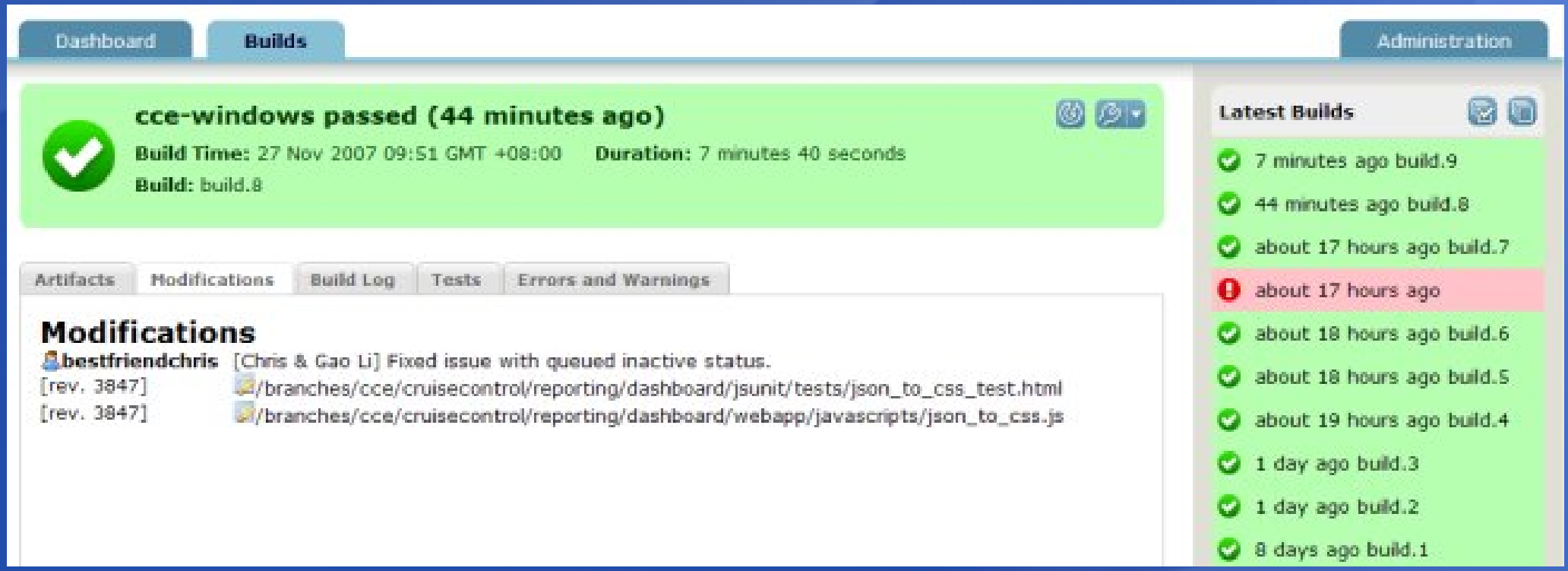
Similar projects

- Hudson (JAVA)



Similar projects

- CruiseControl (JAVA)



BuildBot Demo

Practices of Continuous Integration

What is CI

- A software development practice (requires no particular tooling)
- From Extreme Programming development process

Wikipedia definition

In software engineering, continuous integration (CI) implements continuous processes of applying quality contol on small pieces of effort, applied frequently. Continuous integration aims to improve the quality of software, and to reduce the time taken to deliver it, by replacing the traditional practice of applying quality control after completing all development.

Why use CI system

- Detect integration errors ASAP
  - Fix bugs made in last week: painful
  - Fix bugs made in last month: very painful!!!

People thought...

- Before
  - It can't work (here).
  - Doing it won't make much difference.
- After
  - Yes we do that – how could you live without it?

How to use CI system

1. Checkout source code from VCS
2. Modify code
3. Run self-testing code (ex. unit test)
4. Update and commit
5. Run self-testing code on an integration machine

How to use CI system (cont.)

- Step 5. can be done automatically by CI system
- Errors is detected rapidly
- A good team should have many correct builds a day
- Bad builds do occur from time to time, but should be quickly fixed

Continuos integration

- Enhance portability:
  - Different Build slaves for different HW/SW architectures
- Enhance/check numerical stability:
  - Different build slaves for different compiler options
- Enhance your own ”...ility”

Practices of CI

- Maintain a single source repository
  - Put everything required for a build
- Automate the build
  - Build and launch in a single command
  - Need a virgin machine

Practices of CI (cont.)

- Make your build self-testing
  - Testing can catch a lot of bugs – enough to be useful
- Everyone commits every day
  - Conflicts that stay undetected for weeks can be very hard to resolve

Practices of CI (cont.)

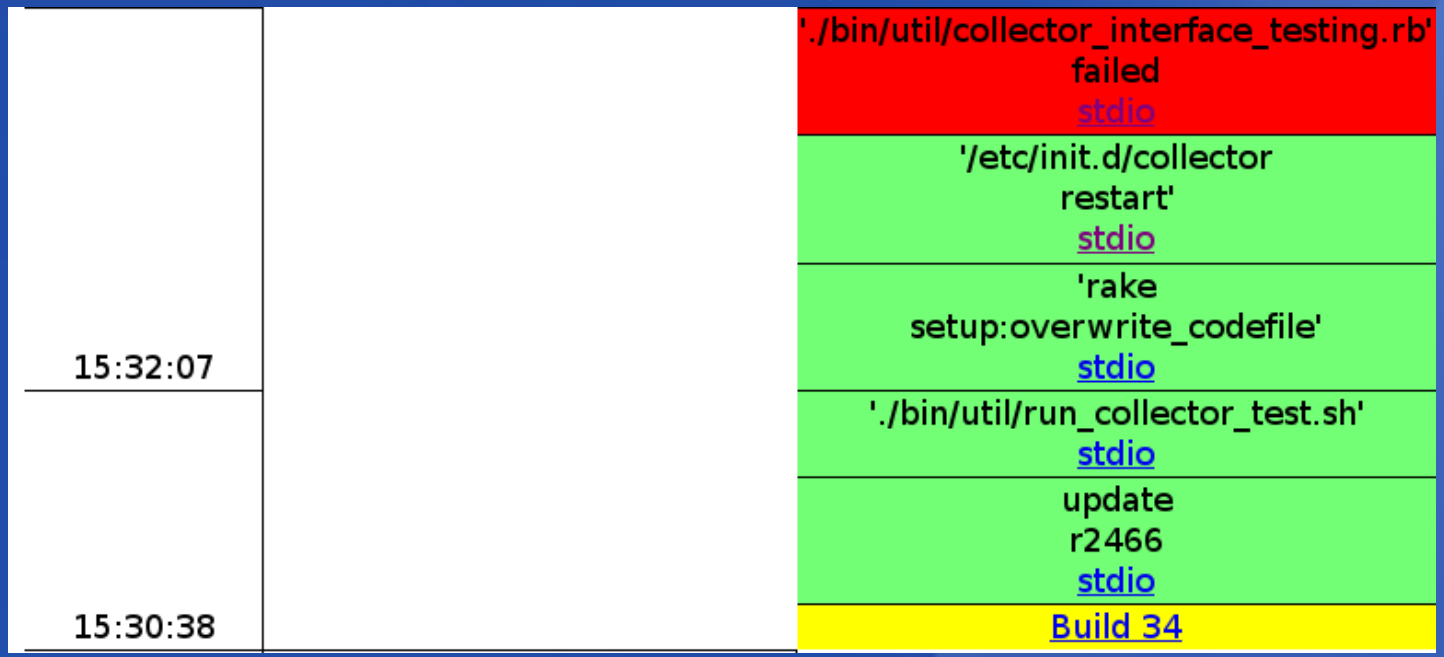
- Every commit should build the mainline on an integration machine
  - Why: things still go wrong
    - Developers forget to run build before commit
    - Environmental differences
  - Manual build & CI system

Practices of CI (cont.)

- Keep the build fast
  - The whole point of CI is to provide rapid feedback
  - 10 minutes build

Practices of CI (cont.)

- Keep the build fast (cont.)
  - Staged build (2nd build doesn't have the same 'stop everthing' quality)



Practices of CI (cont.)

- Keep the build fast (cont.)
  - 2nd build fails: add another test into commit test suit
    - MySQL story
- Test in a clone of the production environment
  - Real tests without mocks

Practices of CI (cont.)

- Make it easy for anyone to get the latest executable
  - Nightly build
- Everyone can see what's happening
  - BuildBot demo (countdown)
- Automate deployment
  - Also needs automated rollback

Benefits of CI

- Reduce risk
  - Integration is a long and unpredictable process
- There's no long integration anymore
- Easier to find bugs and remove them
  - Quality of your test suits

Where to start

1. Get the build automated
   - An automated nightly build is a fine step
2. Introduce some automated testing into your build
   - Try to identify the major areas

Where to start (cont.)

3. Try to speed up the commit build
   - Magic 10 mins

Buildbot for RegCM (1)

- Continuous integration:
  - Test all the compilers/libraries available on the ICTP HPC facility
  - Test all the option available:
    - ./configure –enable-clm
    - ./configure –enable-band
    - ./configure –enable-debug
  - Run some regression tests on any change...

Buildbot for RegCM(2)

- Nightly bots to run a set of basic examples
  - RegCM_root/Testing
- Aim:
  - Check if last modification broke numerical results
- Other things to check/do
  - Enable bots on branches
  - Enable bots on different HW

BuildBot

And Continuous Integration
RegCM4 experiences

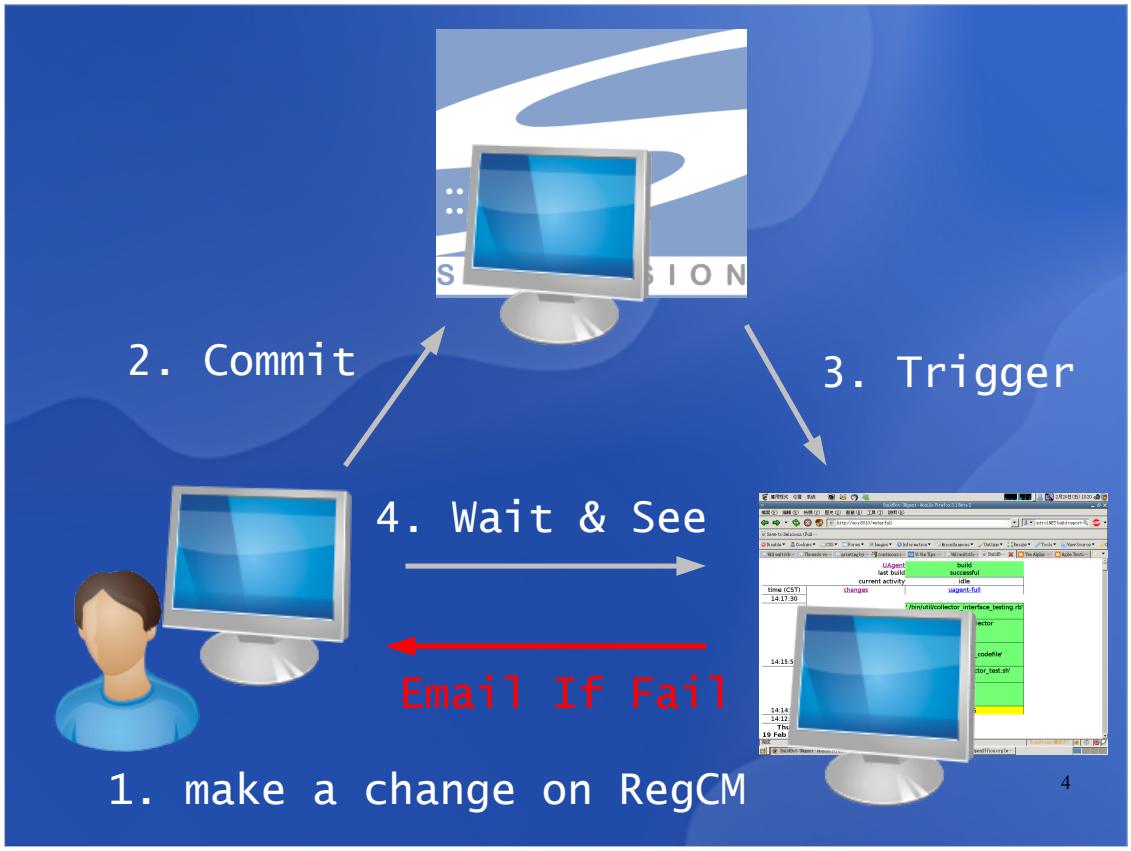
S.Cozzini/A.Messina/G.Giuliani

Warning: Some slides/ideas stolen by Willie <willie@issdu.com.tw>

Agenda

- How do we use BuildBot here?
- What is BuildBot?
- Practices of Continuous Integration
- Example for RegCM code

How do we use BuildBot?


2. Commit
3. Trigger
4. Wait & See
Email If Fail
1. make a change on RegCM

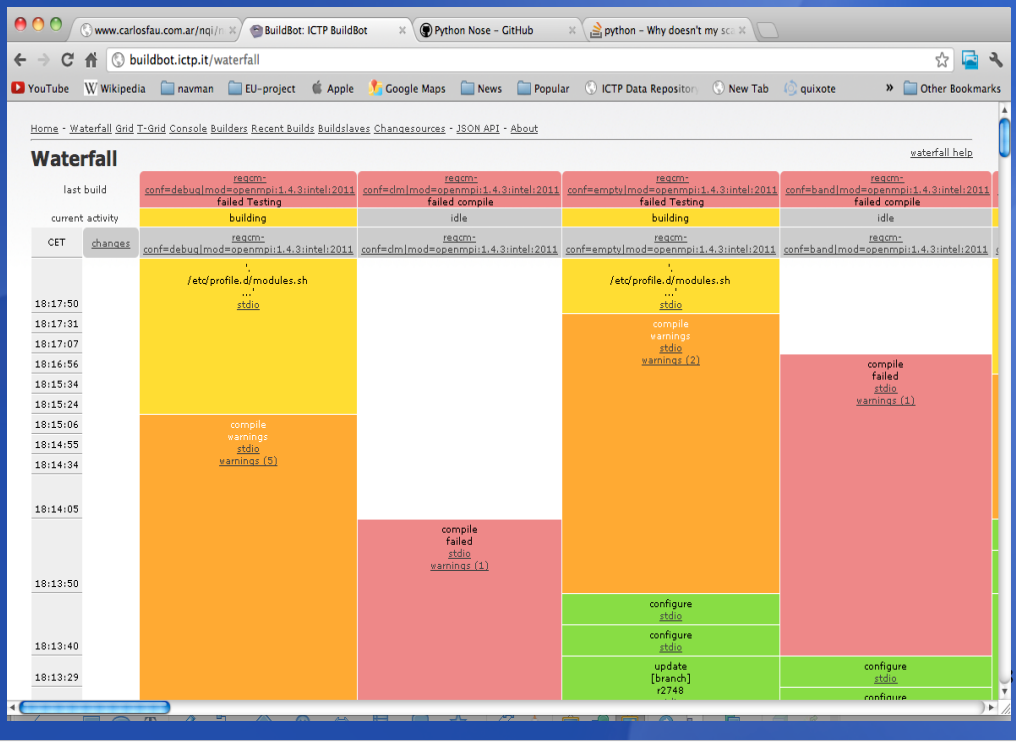
```
Subject: buildbot failure in ICTP BuildBot on
regcm-conf=debug|mod=openmpi:1.4.3:intel:2011
Date: Tue, 21 Feb 2012 22:20:39 +0100
From: buildmaster@gforge.ictp.it
To: amessina@ictp.it, ggiulian@ictp.it
The Buildbot has detected a failed build on builder
regcm-conf=debug|mod=openmpi:1.4.3:intel:2011 while building ICTP
BuildBot.
Full details are available at:

http://buildbot.ictp.it/builders/regcm-conf%3Ddebug%7Cmod%3Dopenmpi%3A1.4.3%3Aintel%3A20
Buildbot URL: http://buildbot.ictp.it/
Buildslave for this Build: argo-buildbot01
Build Reason: The web-page 'force build' button was pressed by '':
Build Source Stamp: 2750
Blamelist:
BUILD FAILED: failed compile
sincerely,
 -The Buildbot
```

What is BuildBot?

BuildBot

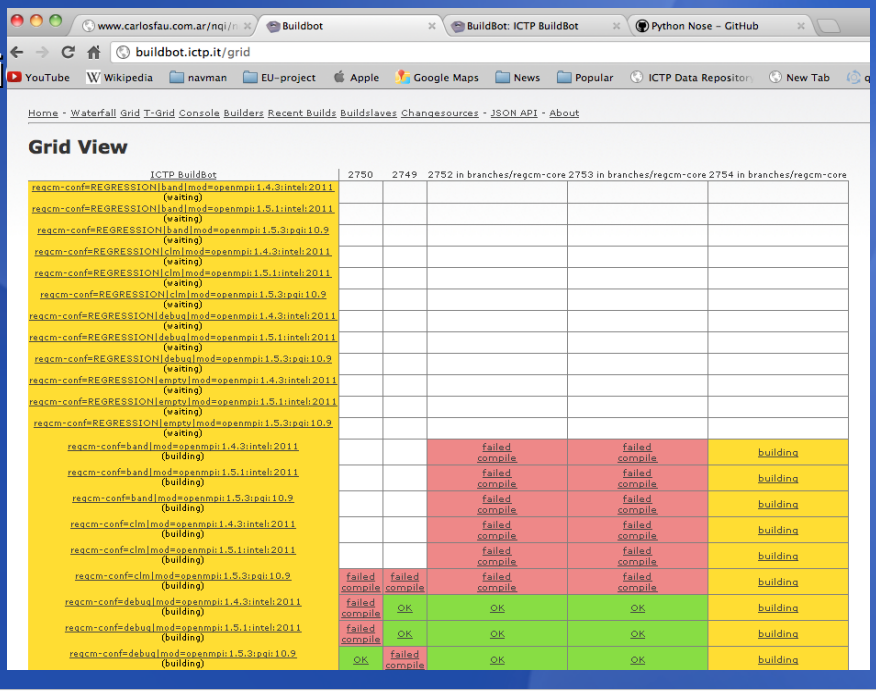
- A continuous integration system
- Help to automate the compile/test cycle
- Written in Python

Screenshot


www.carlosfau.com.ar/nqi/
BuildBot: ICTP BuildBot
Python Nose – GitHub
python – Why doesn't my sc
buildbot.ictp.it/waterfall
YouTube
Wikipedia
navman
EU-project
Apple
Google Maps
News
Popular
ICTP Data Repositor
New Tab
quixote
Other Bookmarks
Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources - JSON API - About
Waterfall
waterfall help
last build
current activity
CET
changes
reqcm-conf=debug|mod=openmpi:1.4.3:intel:2011
failed Testing
reqcm-conf=dm|mod=openmpi:1.4.3:intel:2011
failed compile
reqcm-conf=empty|mod=openmpi:1.4.3:intel:2011
failed Testing
reqcm-conf=band|mod=openmpi:1.4.3:intel:2011
failed compile
building
idle
building
idle
'./etc/profile.d/modules.sh ...'
stdio
compile
warnings
stdio
warnings (5)
compile
failed
stdio
warnings (1)
compile
warnings
stdio
warnings (2)
compile
failed
stdio
warnings (1)
configure
stdio
configure
stdio
update
[branch]
r2748
configure
stdio
18:17:50
18:17:31
18:17:07
18:16:56
18:15:34
18:15:24
18:15:06
18:14:55
18:14:34
18:14:05
18:13:50
18:13:40
18:13:29

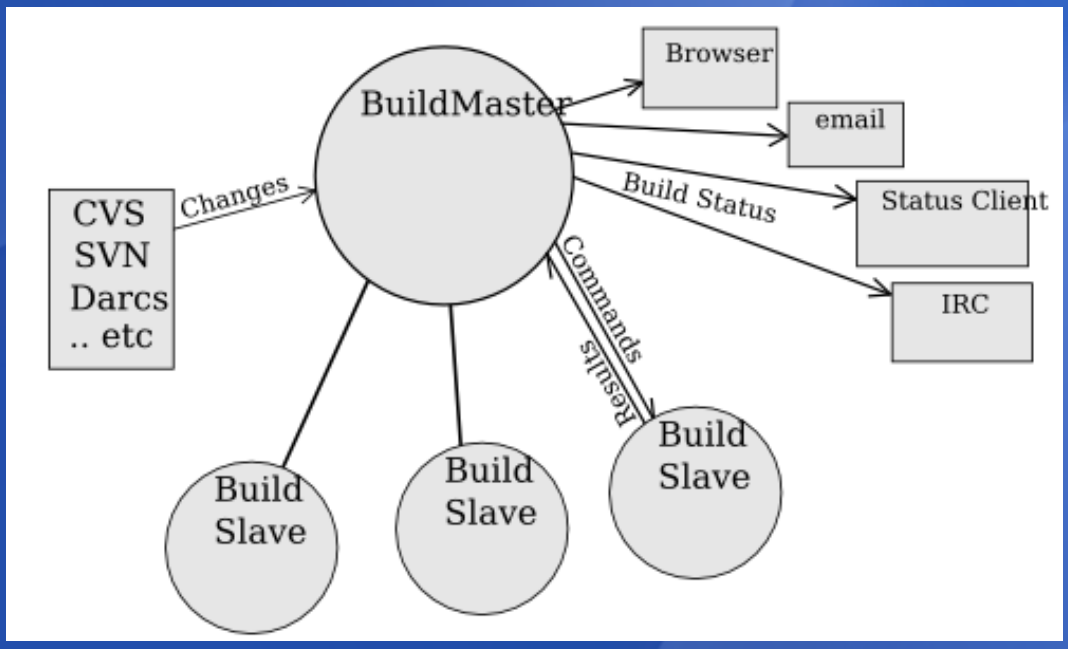
Screenshot

- Cli



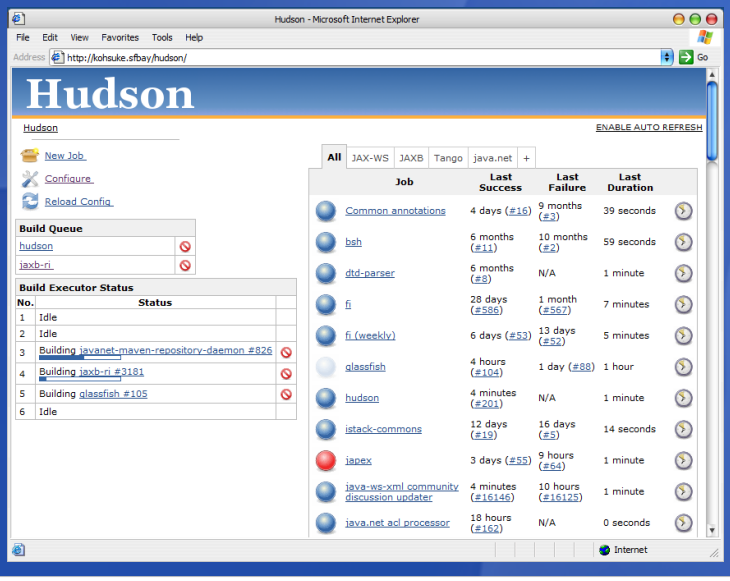
ICTP BuildBot	2750	2749	2752 in branches/regcm-core	2753 in branches/regcm-core	2754 in branches/regcm-core
regcm-conf=REGRESSION|band|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|band|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|band|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=REGRESSION|clm|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|clm|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|clm|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=REGRESSION|debug|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|debug|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|debug|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=REGRESSION|empty|mod=openmpi:1.4.3:intel:2011 (waiting)					
regcm-conf=REGRESSION|empty|mod=openmpi:1.5.1:intel:2011 (waiting)					
regcm-conf=REGRESSION|empty|mod=openmpi:1.5.3:pgi:10.9 (waiting)					
regcm-conf=band|mod=openmpi:1.4.3:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=band|mod=openmpi:1.5.1:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=band|mod=openmpi:1.5.3:pgi:10.9 (building)			failed compile	failed compile	building
regcm-conf=clm|mod=openmpi:1.4.3:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=clm|mod=openmpi:1.5.1:intel:2011 (building)			failed compile	failed compile	building
regcm-conf=clm|mod=openmpi:1.5.3:pgi:10.9 (building)	failed compile	failed compile	failed compile	failed compile	building
regcm-conf=debug|mod=openmpi:1.4.3:intel:2011 (building)	failed compile	OK	OK	OK	building
regcm-conf=debug|mod=openmpi:1.5.1:intel:2011 (building)	failed compile	OK	OK	OK	building
regcm-conf=debug|mod=openmpi:1.5.3:pgi:10.9 (building)	OK	failed compile	OK	OK	building

System architecture



Similar projects

- Hudson (JAVA)



Similar projects

- CruiseControl (JAVA)

BuildBot Demo

Practices of Continuous Integration

What is CI

- A software development practice (requires no particular tooling)
- From Extreme Programming development process

Wikipedia definition

In software engineering, continuous integration (CI) implements continuous processes of applying quality contol on small pieces of effort, applied frequently. Continuous integration aims to improve the quality of software, and to reduce the time taken to deliver it, by replacing the traditional practice of applying quality control after completing all development.

Why use CI system

- Detect integration errors ASAP
  - Fix bugs made in last week: painful
  - Fix bugs made in last month: very painful!!!

People thought...

- Before
  - It can't work (here).
  - Doing it won't make much difference.
- After
  - Yes we do that – how could you live without it?

How to use CI system

1. Checkout source code from VCS
2. Modify code
3. Run self-testing code (ex. unit test)
4. Update and commit
5. Run self-testing code on an integration machine

How to use CI system (cont.)

- Step 5. can be done automatically by CI system
- Errors is detected rapidly
- A good team should have many correct builds a day
- Bad builds do occur from time to time, but should be quickly fixed

Continuos integration

- Enhance portability:
  - Different Build slaves for different HW/SW architectures
- Enhance/check numerical stability:
  - Different build slaves for different compiler options
- Enhance your own ”...ility”

Practices of CI

- Maintain a single source repository
  - Put everything required for a build
- Automate the build
  - Build and launch in a single command
  - Need a virgin machine

Practices of CI (cont.)

- Make your build self-testing
  - Testing can catch a lot of bugs – enough to be useful
- Everyone commits every day
  - Conflicts that stay undetected for weeks can be very hard to resolve

Practices of CI (cont.)

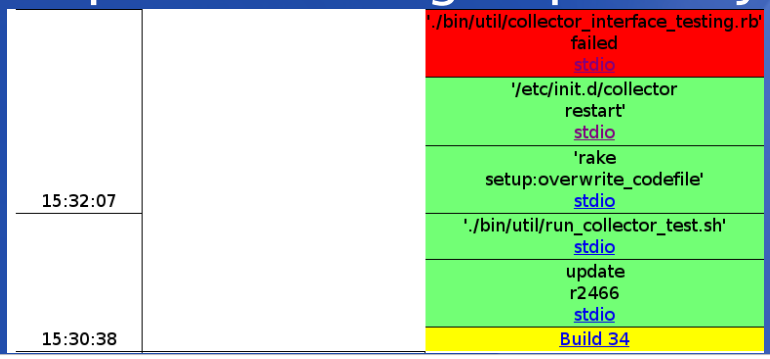
- Every commit should build the mainline on an integration machine
  - Why: things still go wrong
    - Developers forget to run build before commit
    - Environmental differences
  - Manual build & CI system

Practices of CI (cont.)

- Keep the build fast
  - The whole point of CI is to provide rapid feedback
  - 10 minutes build

Practices of CI (cont.)

- Keep the build fast (cont.)
  - Staged build (2nd build doesn't have the same 'stop everthing' quality)



Practices of CI (cont.)

- Keep the build fast (cont.)
  - 2nd build fails: add another test into commit test suit
    - MySQL story
- Test in a clone of the production environment
  - Real tests without mocks

Practices of CI (cont.)

- Make it easy for anyone to get the latest executable
  - Nightly build
- Everyone can see what's happening
  - BuildBot demo (countdown)
- Automate deployment
  - Also needs automated rollback

Benefits of CI

- Reduce risk
  - Integration is a long and unpredictable process
- There's no long integration anymore
- Easier to find bugs and remove them
  - Quality of your test suits

Where to start

1. Get the build automated
   - An automated nightly build is a fine step
2. Introduce some automated testing into your build
   - Try to identify the major areas

Where to start (cont.)

3. Try to speed up the commit build
   - Magic 10 mins

Buildbot for RegCM (1)

- Continuous integration:
  - Test all the compilers/libraries available on the ICTP HPC facility
  - Test all the option available:
    - ./configure –enable-clm
    - ./configure –enable-band
    - ./configure –enable-debug
  - Run some regression tests on any change...

Buildbot for RegCM(2)

- Nightly bots to run a set of basic examples
  - RegCM_root/Testing
- Aim:
  - Check if last modification broke numerical results
- Other things to check/do
  - Enable bots on branches
  - Enable bots on different HW