New Traffic Server CI

Brian Neradt, Brian Olsen, Evan Zelkowitz

yahoo!

Agenda

1. New CI Team Members
2. Viewing PR CI job details.
3. Accessing build artifacts
    - AuTest Sandboxes
    - Rendered Sphinx docs
    - Code coverage
4. Using CI Docker Images
5. Branch builds

yahoo!

The New CI Team

yahoo!

New CI Team

- Brian Neradt
- Brian Olsen
- Bryan Call
- Evan Zelkowitz
- Leif Hedstrom

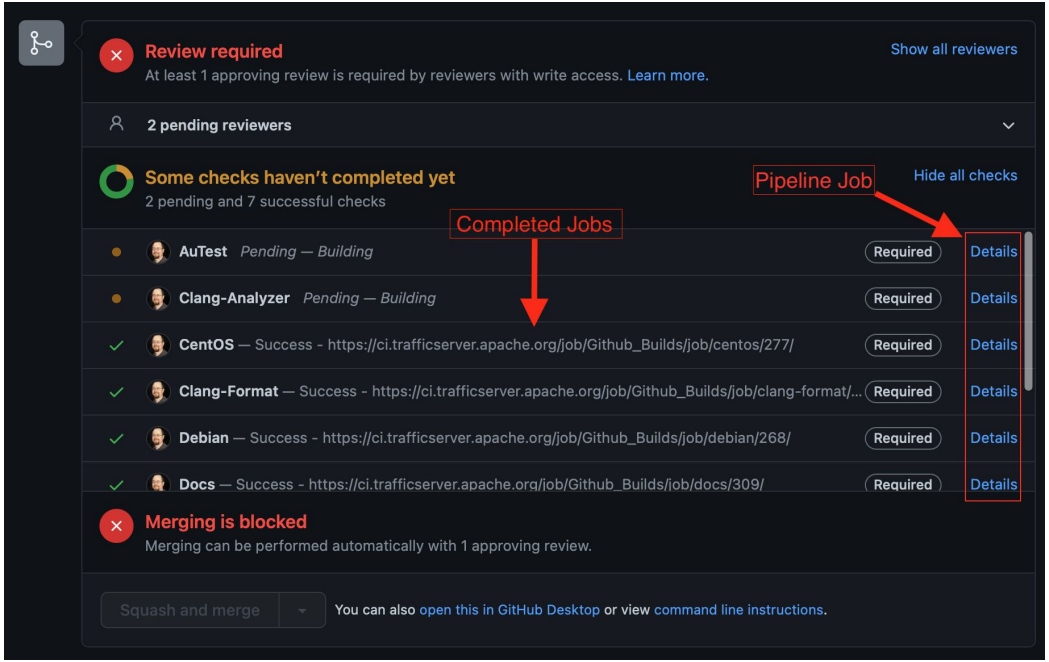
yahoo!

Viewing PR CI Job Details

yahoo!

PR View

- PR Jobs are built in two stages:
  - Docs, RAT, clang-format
  - OS builds, autest, clang-analyzer
- The Details link takes you to the PR master job which runs the various jobs for the PR.



Workflow:

- PR triggers pipeline job via a webhook.
- Pipeline job runs groovy script which triggers the individual jobs.
- Jobs post results back to github.

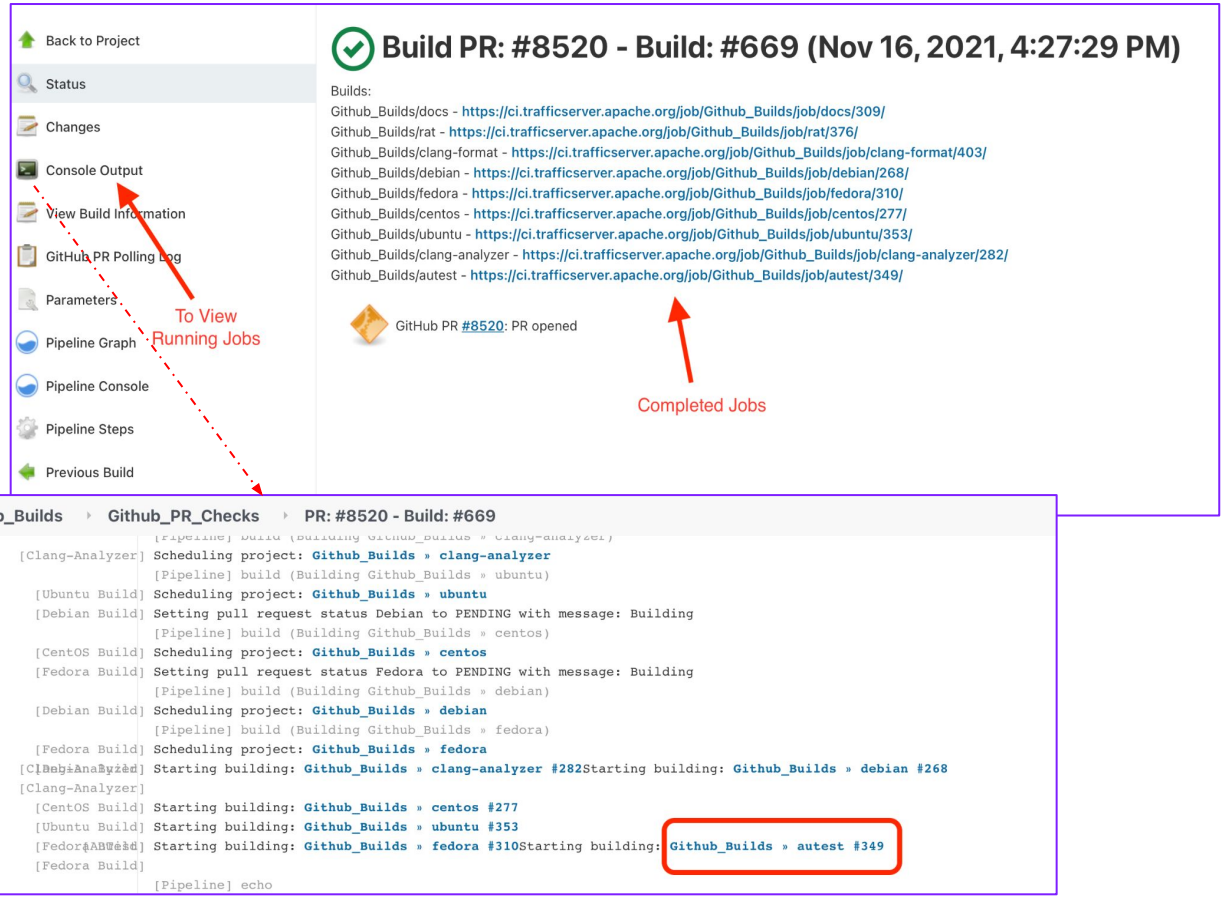
Groovy scripts: https://github.com/apache/trafficserver-ci/tree/main/jenkins/github

yahoo!

Details View

To view individual CI jobs:

1. Follow Details link.
2. Completed jobs with links are displayed in the description.
3. Running jobs can be accessed from the Console Output.



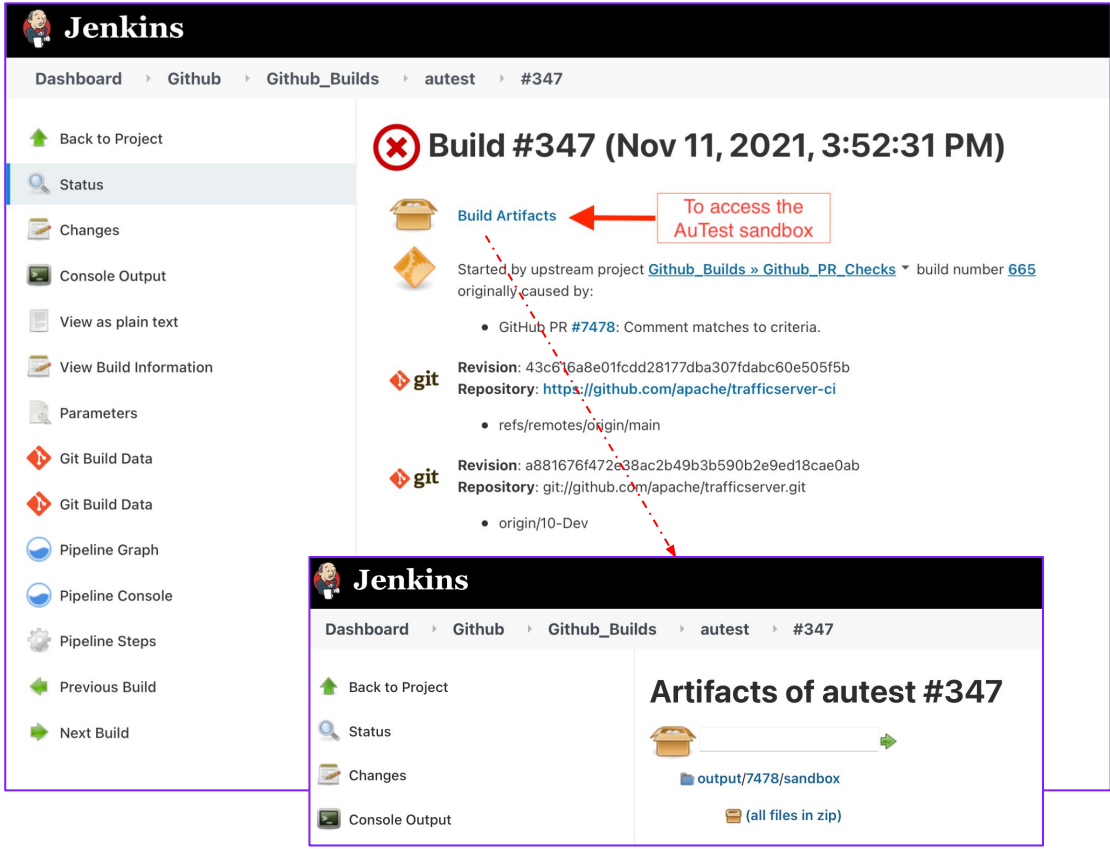
yahoo!

Accessing Build Artifacts

yahoo!

Access AuTest Sandbox

- Jenkins will export the AuTest sandbox for failed AuTest runs.
- Access it via the “Build Artifacts” link in the failed job.



yahoo!

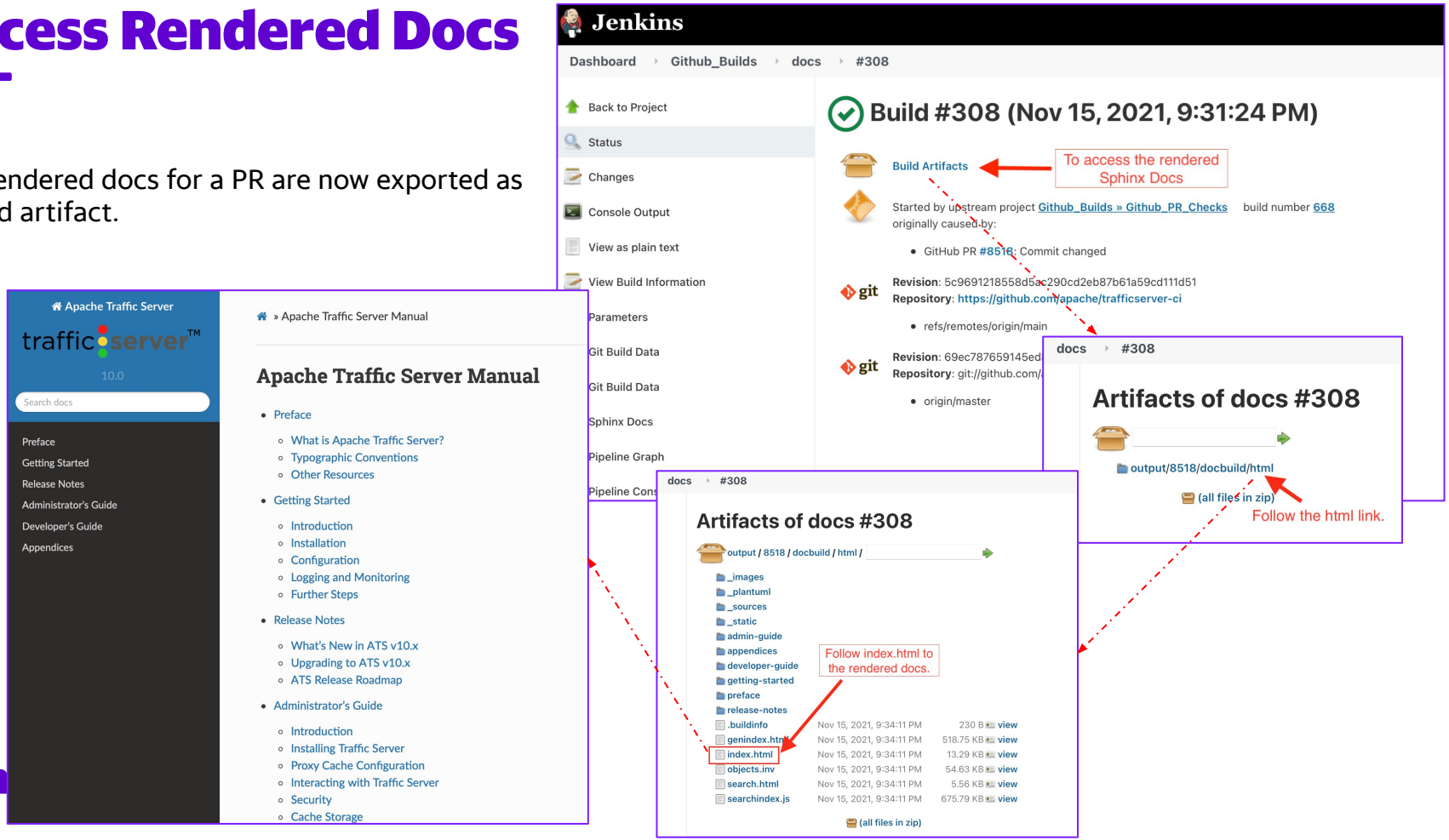
clang-analyzer

Failed clang-analyzer PR runs will also have an archive. It can be accessed in a similar way a failed AuTest's sandbox is accessed.

yahoo!

Access Rendered Docs

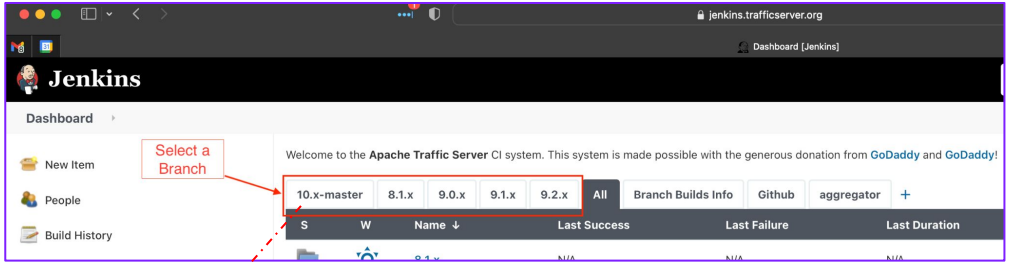
The rendered docs for a PR are now exported as a build artifact.

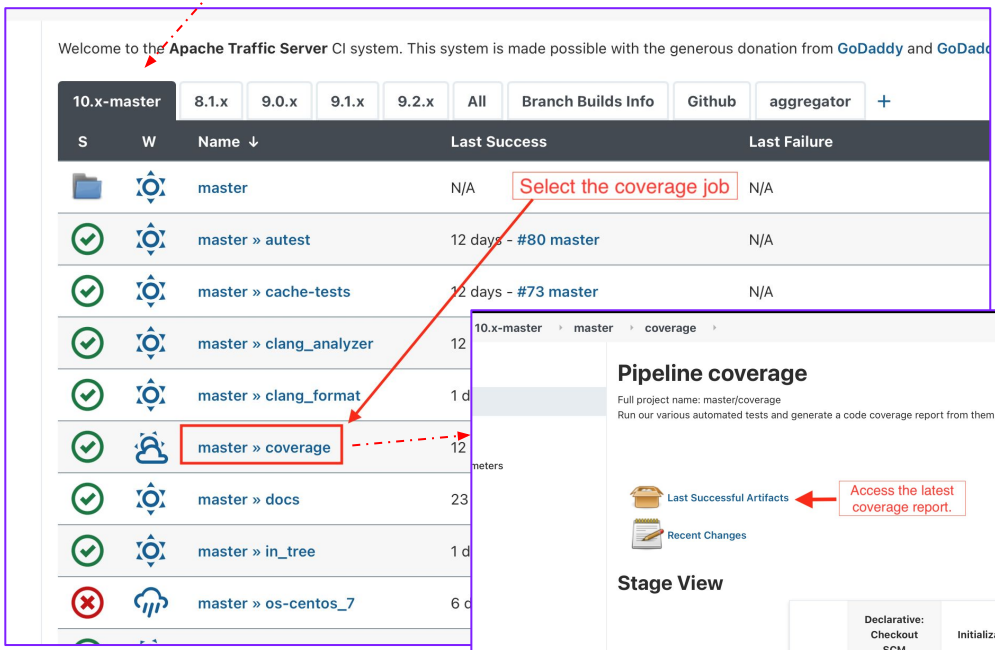


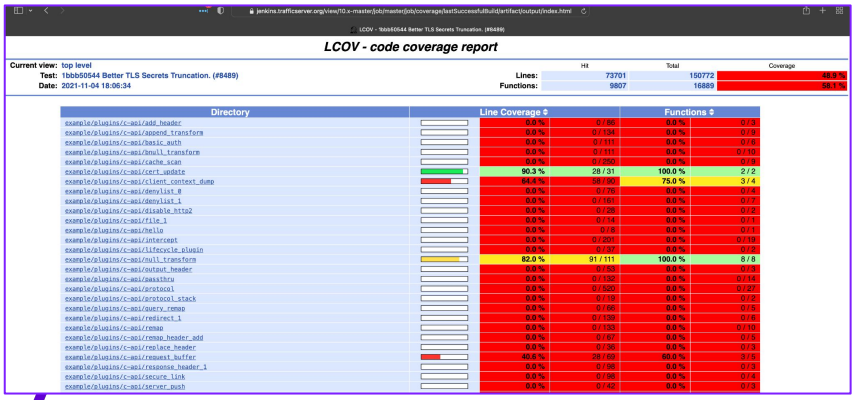
Access Branch Coverage

Test coverage reports are generated for branch builds:

1. https://jenkins.trafficserver.org
2. Select a branch
3. Select the “coverage” build.
4. Select “Last Successful Artifacts”
5. output > index.html







Local Builds with CI Docker Images

yahoo!

Using CI's Docker Images

The Docker images CI uses are publicly available. Using it for local builds and autest runs is easy:

```
# Install docker
docker pull controller.trafficserver.org/ats/centos:8
docker run -it -u 1200:1200 --init --cap-add=SYS_PTRACE --network=host \
    controller.trafficserver.org/ats/centos:8 /bin/bash
cd
git clone https://github.com/apache/trafficserver.git
cd trafficserver
autoreconf -fi && ./configure --with-openssl=/opt/openssl-quic
--enable-experimental-plugins --enable-example-plugins --prefix=/tmp/ats
--enable-werror --enable-debug --enable-wccp --enable-luajit && make -j4
install

cd tests
./autest.sh --ats-bin /tmp/ats/bin/ --sandbox /tmp/sandbox -f <some_test>
```

yahoo!

Using CI's Docker Images

Sample docker-compose.yml (docker-compose run builder)

```
version: "2.2"
services:
  builder:
     image: controller.trafficserver.org/ats/centos:8
     cap_add: [ SYS_PTRACE ]
     volumes:
     - ./:/home/jenkins/workspace/:Z
     environment:
     - TYPE=debug
     - GITHUB_BRANCH=master
     - COMPILER=gcc
     - WORKSPACE=/home/jenkins/workspace
     - FEATURES=asan
     network_mode: host
     working_dir: /home/jenkins/workspace
     command: bash
```

yahoo!

Branch Builds

yahoo!

Branch Builds

Branch builds more thoroughly check the status and health of ATS.

Build phases:

1. Build in tree, out of tree, RAT, clang-format
2. Multiple OS and compiler combinations (ie: debian 9 clang debug)
   a. Currently up to 36 of these builds per branch
3. clang analyzer, autests, docs, cache tests, coverage

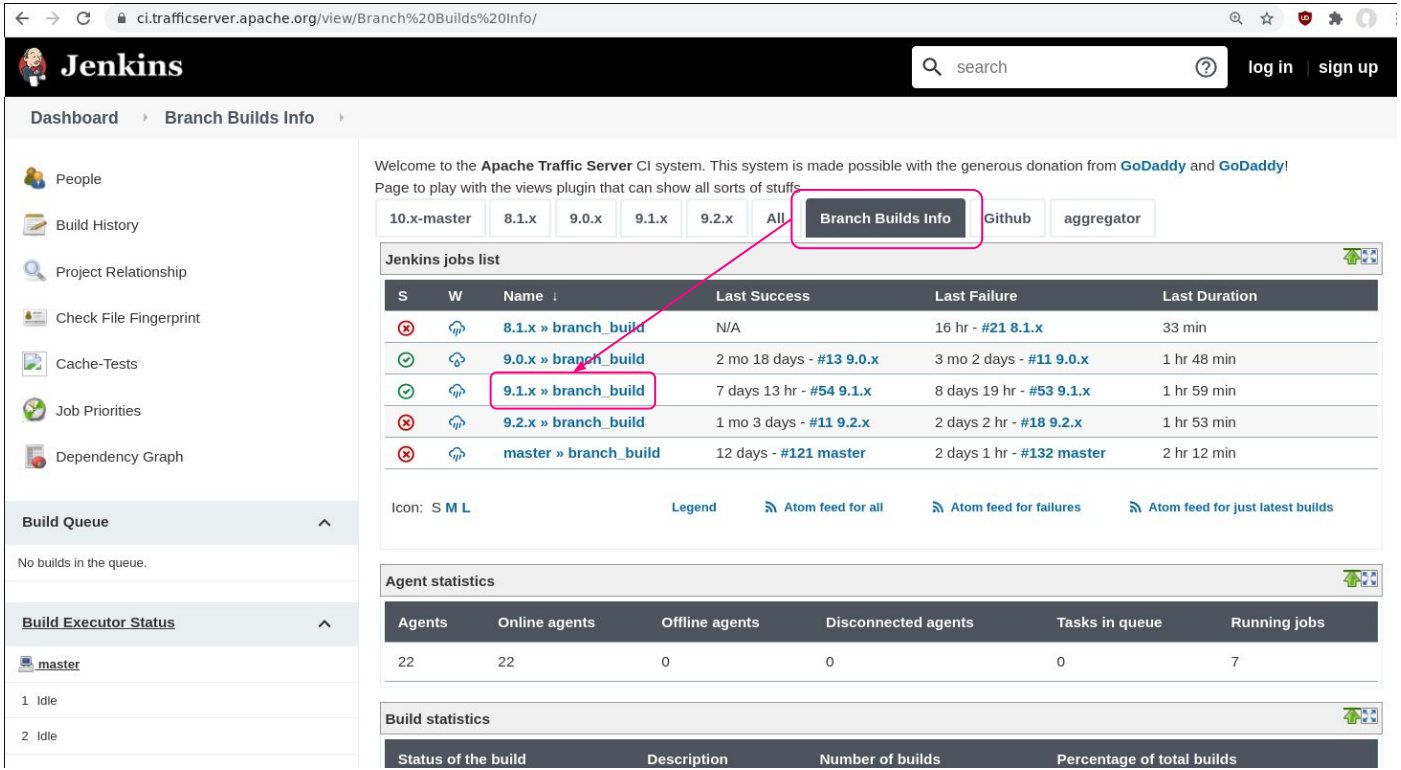
Stage View

	Initialization	Validation	In Tree	Out Of Tree	RAT	clang format	Snapshot	OS Builds	9.1.x/os-centos_7	9.1.x/os-centos_8	9.1.x/os-debian_10	9.1.x/os-debian_9	9.1.x/os-fedora_32	9.1.x/os-fedora_33	9.1.x/os-ubuntu_18.04	9.1.x/os-ubuntu_20.04	9.1.x/os-ubuntu_21.04	Verification	clang analyzer	autests	docs	cache_tests	coverage
Average stage times: (Average full run time: ~1h 59min)	1s	81ms	5min 46s	5min 44s	1min 27s	1min 52s	123ms	531ms	58min 9s	48min 54s	1h 13min	1h 15min	1h 18min	1h 11min	1h 20min	53min 3s	1h 10min	113ms	3min 3s	5min 54s	1min 10s	1min 1s	9min 5s
#54 9.1.x Nov 09 18:53 No Changes	1s	70ms	5min 29s	5min 31s	1min 16s	1min 16s	118ms	550ms	54min 29s	48min 6s	36min 5s	56min 8s	1h 8min	1h 3min	1h 5min	23min 26s	39min 22s	89ms	15min 17s	29min 29s	5min 51s	5min 6s	45min 29s
#53 9.1.x Nov 08 12:53 1 commit	1s	83ms	5min 36s	5min 43s	1min 14s	1min 42s	106ms	586ms	1h 46min failed	20min 24s failed	1h 59min failed	1h 42min failed	1h 41min	1h 57min failed	2h 1min	28min 30s	2h 3min	95ms	156ms failed	163ms failed	175ms failed	187ms failed	203ms failed
#52 Nov 04 No Changes		70ms	5min 34s	5min 42s	1min 14s	1min 48s	147ms	573ms	37min 40s	45min 44s	39min 32s	46min 38s	42min 5s	18min 40s	33min 44s	31min 32s	50min 6s	108ms	162ms	174ms	186ms	199ms	211ms

yahoo!

Branch Builds

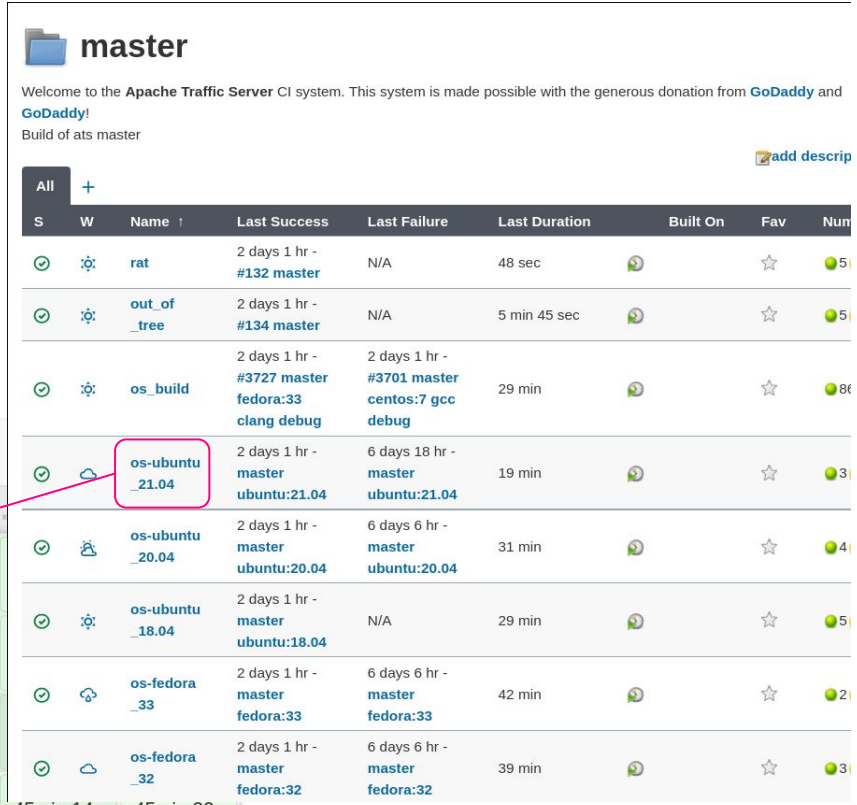
Quick status and dive



yahoo!

Branch Builds

- Each branch is checked twice daily for new PRs.
  - Potentially up to 344 OS builds daily.
- Default image used is centos 8.
- Each OS build just needs an appropriate docker image.



yahoo!

Questions?

yahoo!

Thank You

yahoo!