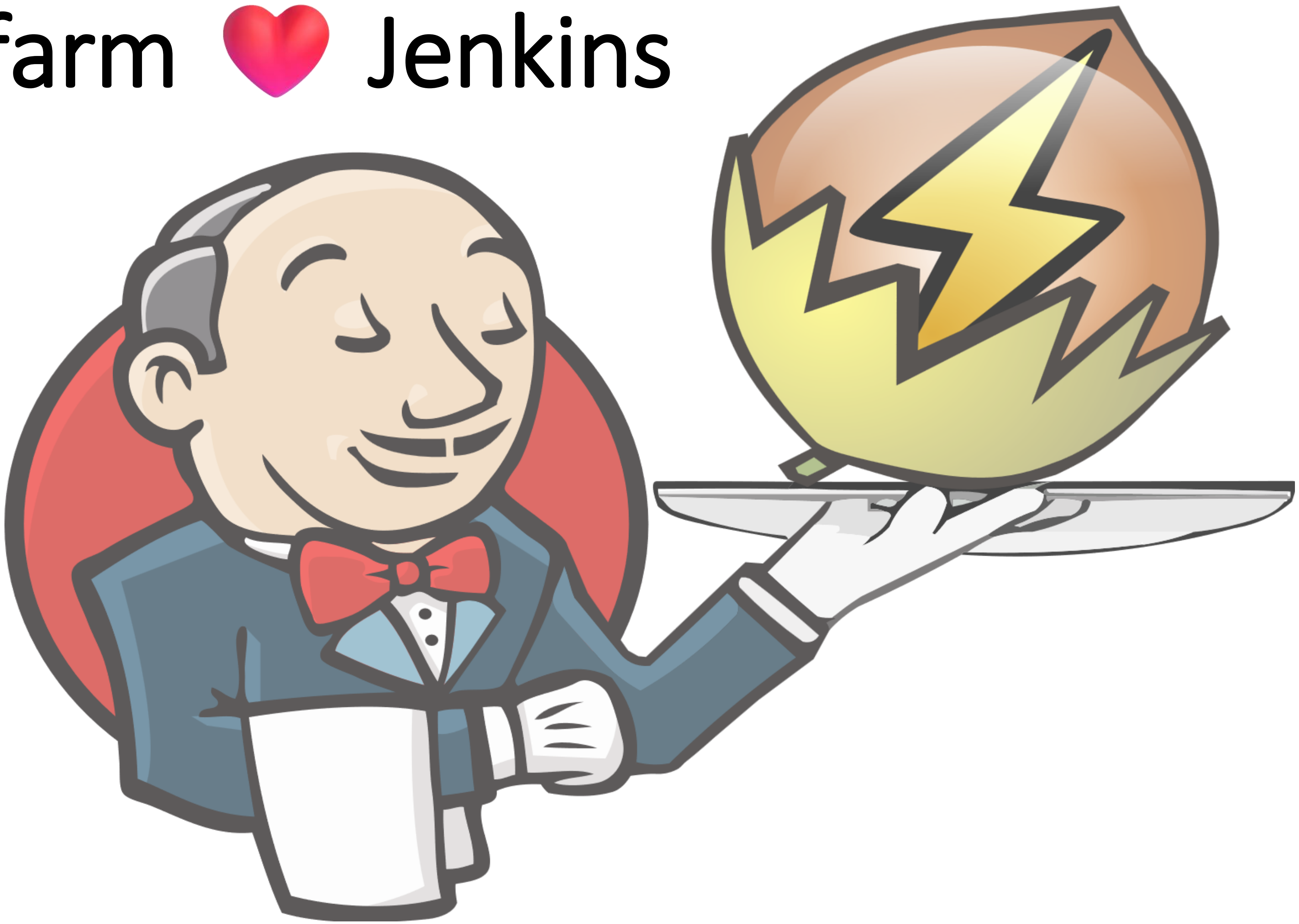
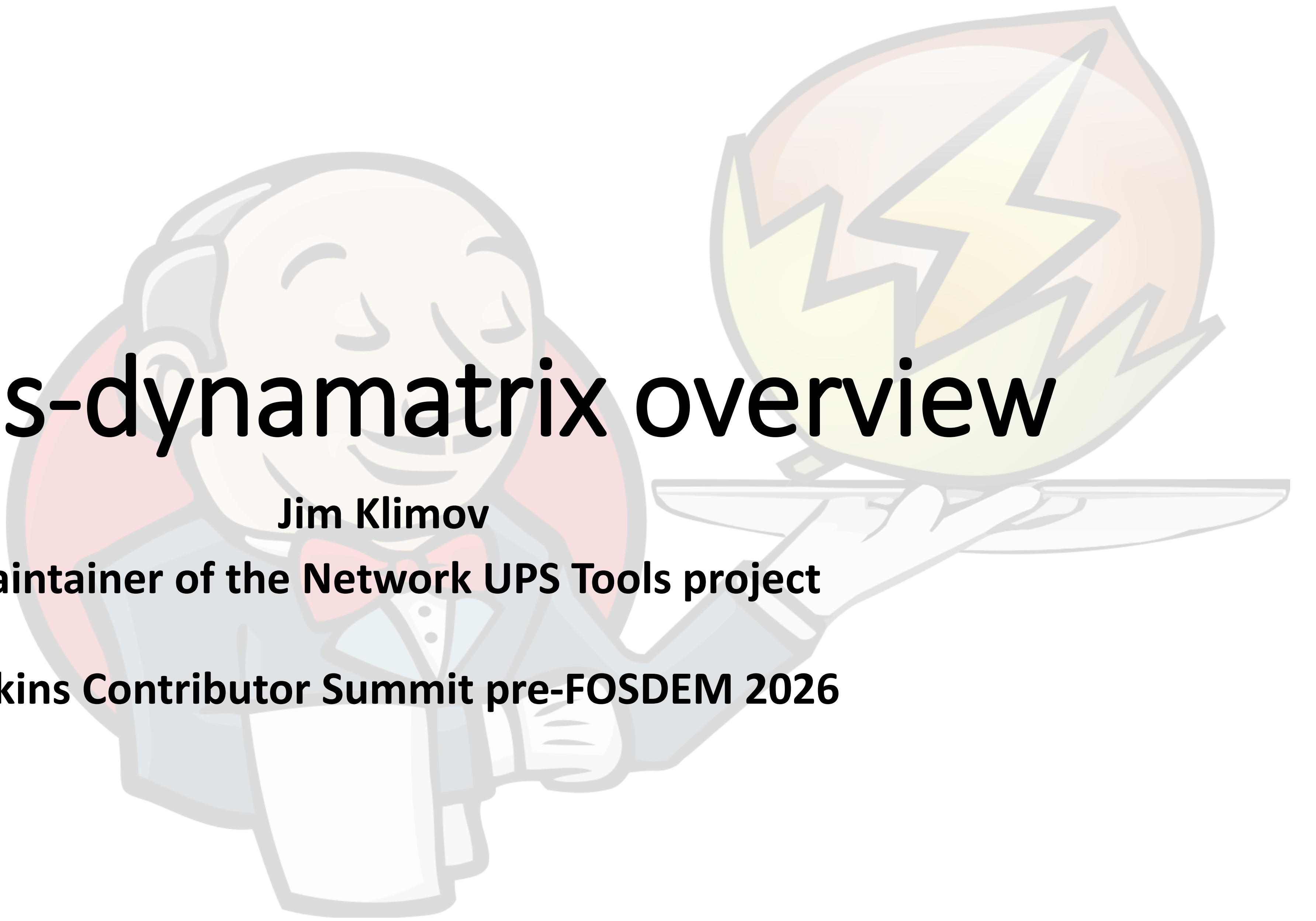
NUT CI farm ❤️ Jenkins





Jenkins-dynamatrix overview

Jim Klimov

Maintainer of the Network UPS Tools project

Jenkins Contributor Summit pre-FOSDEM 2026

Why

- Our modus operandi: Any system that was power-protected with NUT, and is still alive, deserves to be protected with modern NUT versions. Their vendors may not be around… for decades now.
- Ensure portability across 25+ years of distros (and toolkit versions and implementations, and language revisions, and dependencies…)
- Not all tools, dependencies, features are ubiquitous, or behaved consistently over the years
- Not a single-vendor ecosystem like Java, NodeJS, Perl or Python – rather a lot of loosely similar programs doing their thing in different ways, expecting and complaining variably

Why

- Had a practical problem with existing matrix approaches (in various CI offerings), where the matrix dictates the build agents which must exist for the CI build, and changes to matrix must be part of codebase
- Most free cloud CI platforms offer a few Linux/Windows(/MacOS) releases as task runners, with one or two CPU architectures
    - This represents the reality of market majority, but not the world FOSS lives in
- Loved Jenkins since before pipelines
- Wanted to learn JSL programming when they appeared

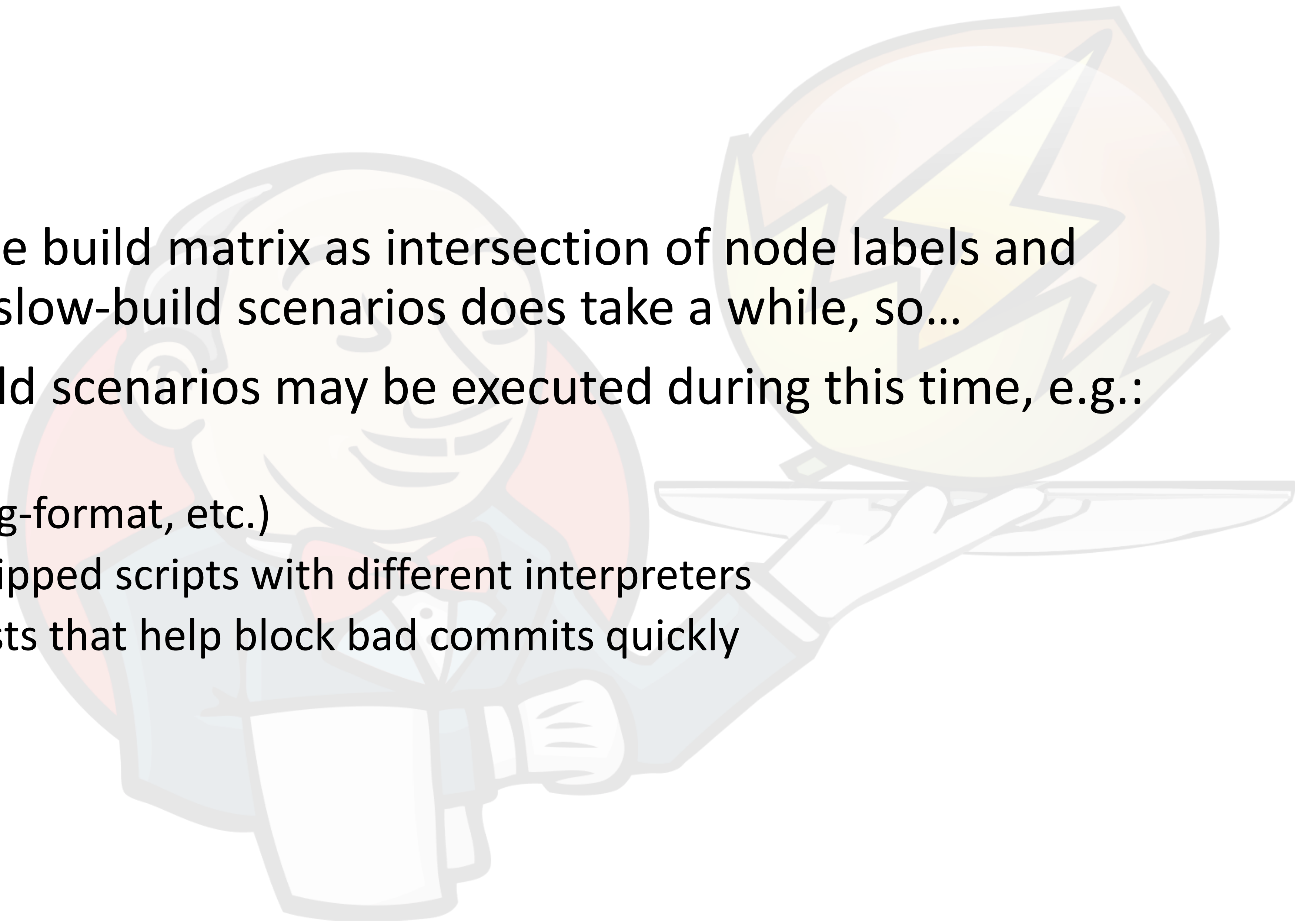
More Why

- Low-level projects in mind (C/C++ on different platforms), although variants of Python, JDK, etc. suffer similarly as languages evolve radically
- Shadow compilations improve quality and help bug hunts by different takes on static analysis, warnings => Jenkins Warnings-NG etc.
- Keep history of (main-branch) builds and their analysis and other artifacts
- Community can provide runners for machines they are interested in

Some How

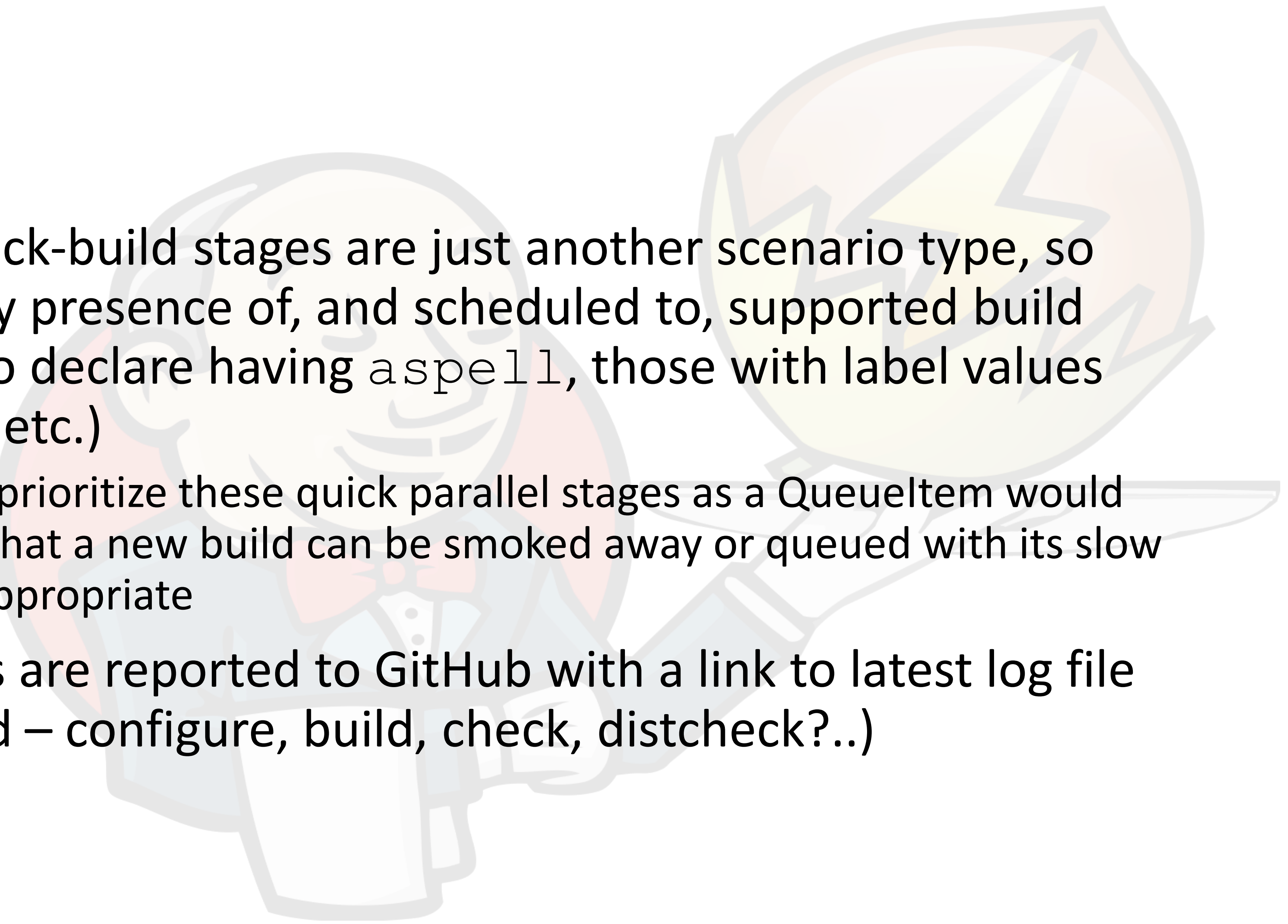
- A single Jenkins Shared Library with the logical backend of the Dynamatrix itself, dissemination of tested code base to builders (DynamarixStash), work with build agents/nodes and their announced labels, supported toolkits (plain GNU autotools, ZeroMQ-inspired `ci_build.sh`), knowledge of supported C and C++ standards by certain GCC and CLANG version ranges, warning analysis summary, GH notification reporting of issues, Badge plugin progress reporting, etc.
- A Jenkinsfile-dynamatrix in a project declares the label-based axes of interest (e.g. `OS_DISTRO`, `COMPILER`, `${COMPILER}VER`…), possibly different for some “slow build” scenario groups and calls a step from the JSL to construct and run the parallel stages for work

Some How



- “Discovery” of the build matrix as intersection of node labels and requirements of slow-build scenarios does take a while, so…
- …a few quick-build scenarios may be executed during this time, e.g.:
  - Spellcheck
  - Stylecheck (clang-format, etc.)
  - Shellcheck of shipped scripts with different interpreters
  - Other smoke tests that help block bad commits quickly

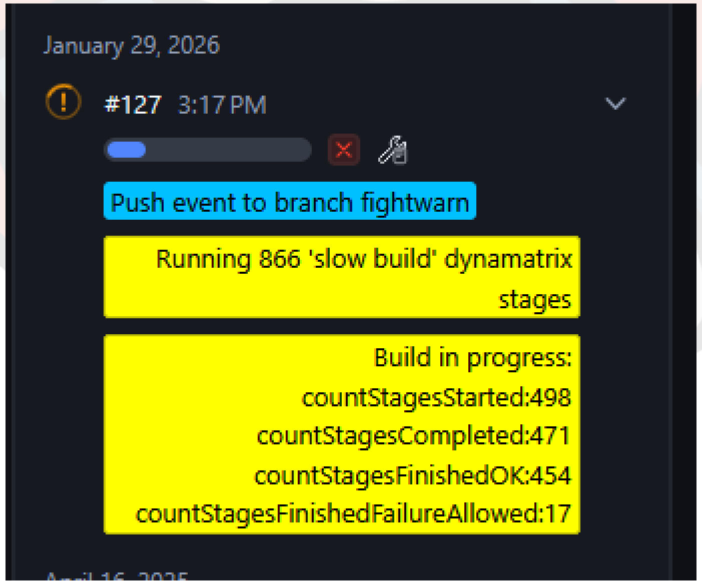
Some How



- The (optional) quick-build stages are just another scenario type, so also are chosen by presence of, and scheduled to, supported build agents (those who declare having `aspell`, those with label values for `SHELL=xxx`, etc.)
  - FIXME: A way to prioritize these quick parallel stages as a QueueItem would be welcome, so that a new build can be smoked away or queued with its slow build matrix as appropriate
- Problematic cases are reported to GitHub with a link to latest log file (wherever it failed – configure, build, check, distcheck?..)

Some How

- Examples from actual CI server?
  - Blue Ocean view
  - Agent configurations
  - GitHub results view
  - Badges



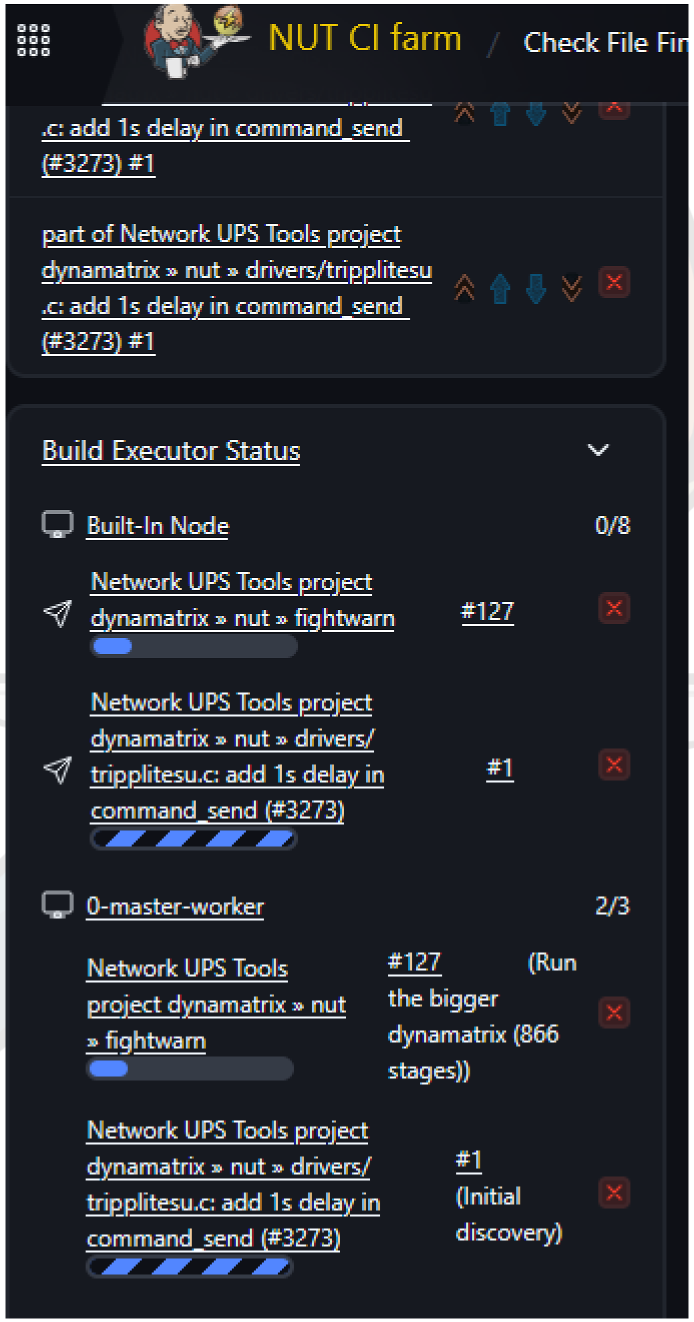
January 29, 2026

#127 3:17 PM

Push event to branch fightwarn

Running 866 'slow build' dynamatrix stages

Build in progress:
countStagesStarted:498
countStagesCompleted:471
countStagesFinishedOK:454
countStagesFinishedFailureAllowed:17

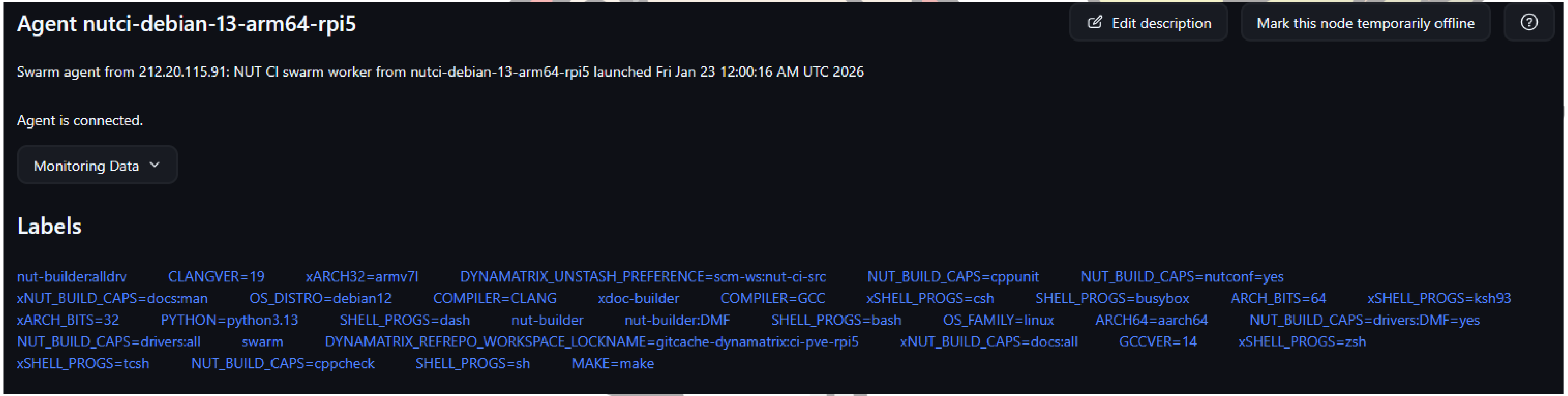


.c: add 1s delay in command_send (#3273) #1

part of Network UPS Tools project dynamatrix » nut » drivers/tripplitesu.c: add 1s delay in command_send (#3273) #1

Build Executor Status

Built-In Node		0/8
Network UPS Tools project dynamatrix » nut » fightwarn	#127	
Network UPS Tools project dynamatrix » nut » drivers/tripplitesu.c: add 1s delay in command_send (#3273)	#1	
0-master-worker		2/3
Network UPS Tools project dynamatrix » nut » fightwarn	#127 (Run the bigger dynamatrix (866 stages))	
Network UPS Tools project dynamatrix » nut » drivers/tripplitesu.c: add 1s delay in command_send (#3273)	#1 (Initial discovery)	



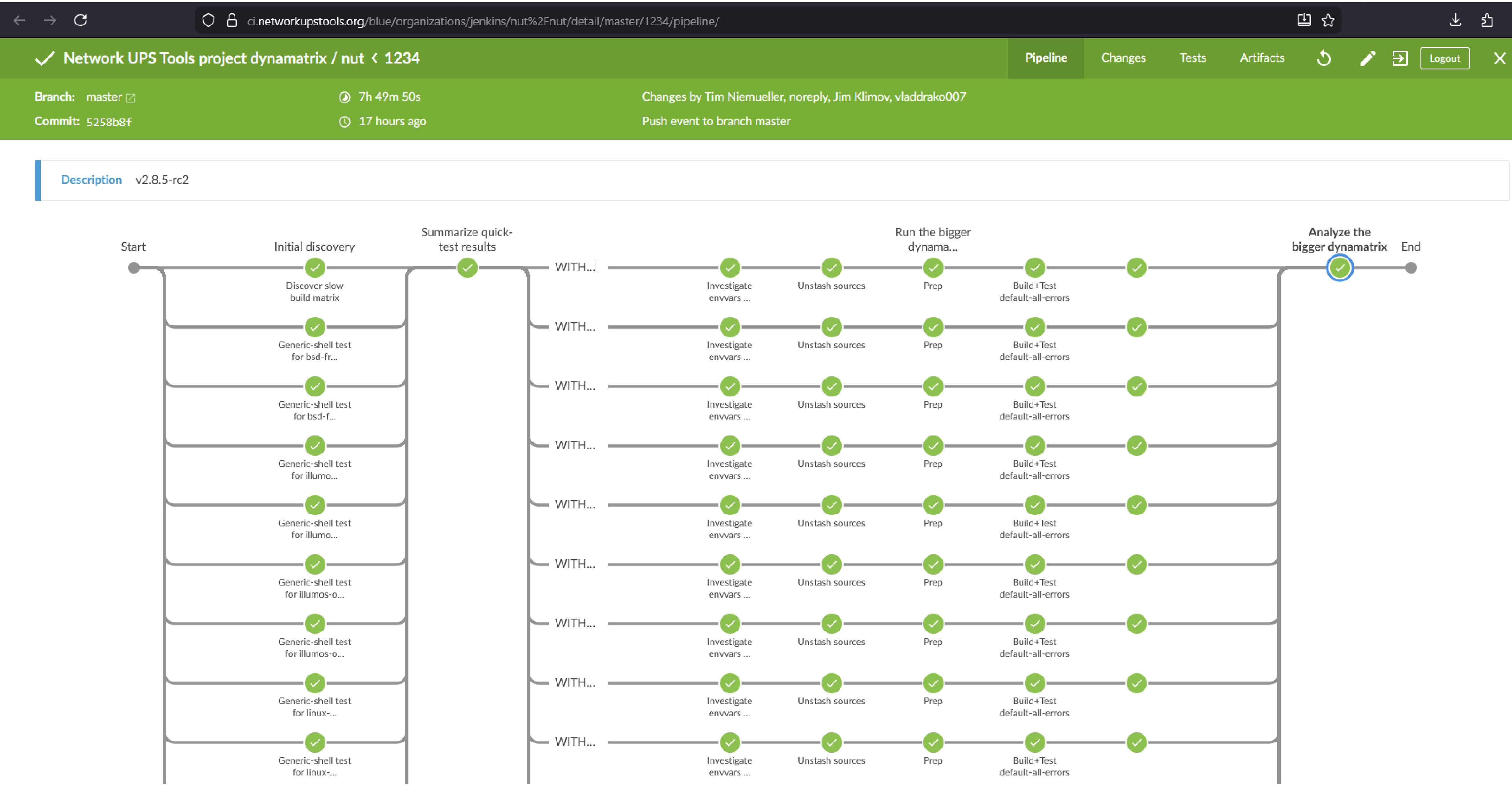
Agent nutci-debian-13-arm64-rpi5

Swarm agent from 212.20.115.91: NUT CI swarm worker from nutci-debian-13-arm64-rpi5 launched Fri Jan 23 12:00:16 AM UTC 2026

Agent is connected.

Labels

nut-builder:alldrv CLANGVER=19 xARCH32=armv7l DYNAMATRIX_UNSTASH_PREFERENCE=scm-ws:nut-ci-src NUT_BUILD_CAPS=cppunit NUT_BUILD_CAPS=nutconf=yes xNUT_BUILD_CAPS=docs:man OS_DISTRO=debian12 COMPILER=CLANG xdoc-builder COMPILER=GCC xSHELL_PROGS=csh SHELL_PROGS=busybox ARCH_BITS=64 xSHELL_PROGS=ksh93 xARCH_BITS=32 PYTHON=python3.13 SHELL_PROGS=dash nut-builder nut-builder:DMF SHELL_PROGS=bash OS_FAMILY=linux ARCH64=aarch64 NUT_BUILD_CAPS=drivers:DMF=yes NUT_BUILD_CAPS=drivers:all swarm DYNAMATRIX_REFREPO_WORKSPACE_LOCKNAME=gitcache-dynamatrix:ci-pve-rpi5 xNUT_BUILD_CAPS=docs:all GCCVER=14 xSHELL_PROGS=zsh xSHELL_PROGS=tcsh NUT_BUILD_CAPS=cppcheck SHELL_PROGS=sh MAKE=make


ci.networkupstools.org/blue/organizations/jenkins/nut%2Fnut/detail/master/1234/pipeline/
Pipeline
Changes
Tests
Artifacts
Logout
Start
Initial discovery
Summarize quick-test results
Run the bigger dynama...
Analyze the bigger dynamatrix
End
Discover slow build matrix
Generic-shell test for bsd-fr...
Generic-shell test for bsd-f...
Generic-shell test for illumo...
Generic-shell test for illumo...
Generic-shell test for illumos-o...
Generic-shell test for illumos-o...
Generic-shell test for linux-...
Generic-shell test for linux-...
WITH...
Investigate envvars ...
Unstash sources
Prep
Build+Test default-all-errors

Network UPS Tools project dynamatrix / nut < 1234

Branch: master

Commit: 5258b8f

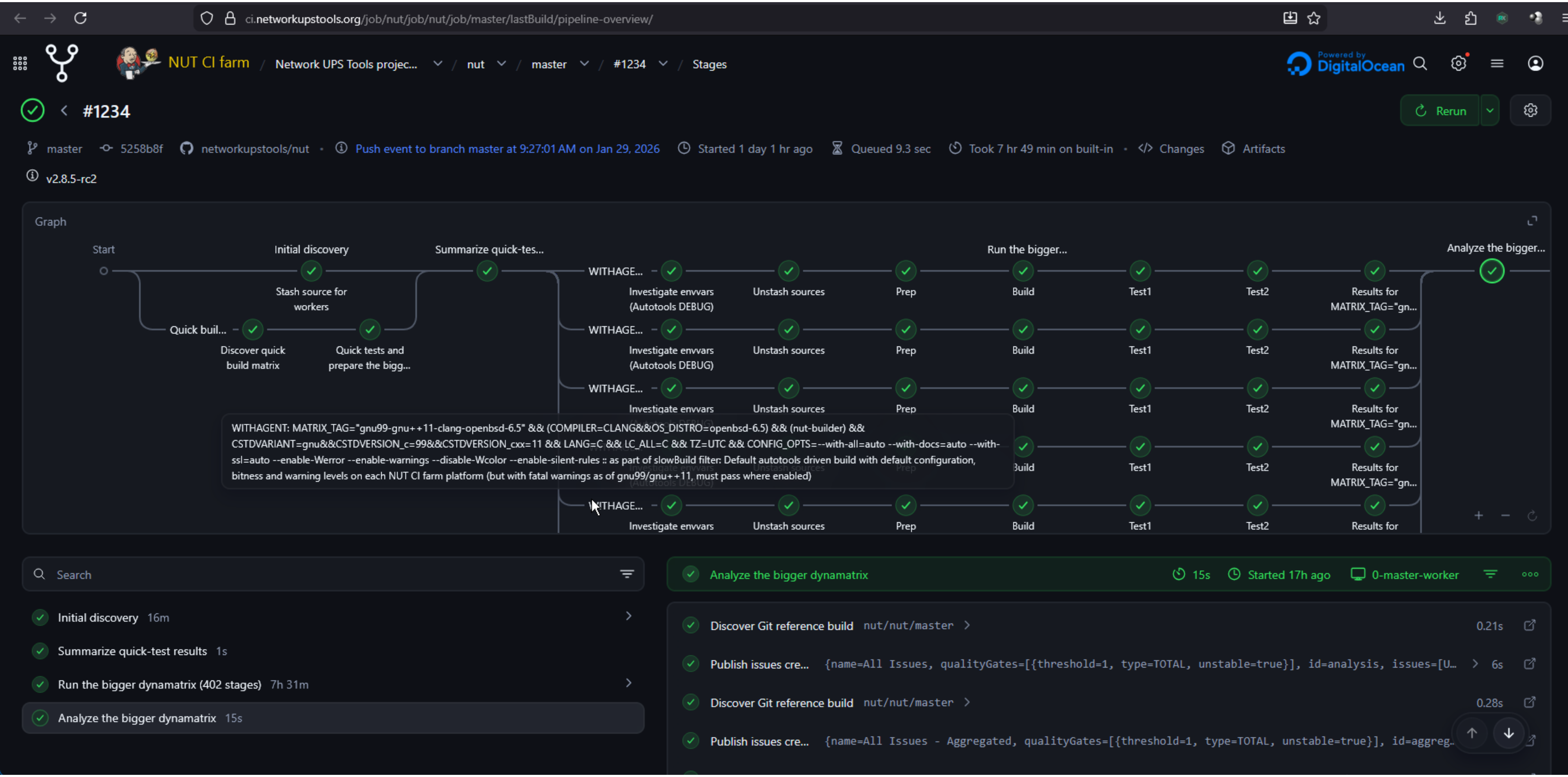
7h 49m 50s

17 hours ago

Changes by Tim Niemueller, noreply, Jim Klimov, vladdrako007

Push event to branch master

Description v2.8.5-rc2



ci.networkupstools.org/job/nut/job/nut/job/master/lastBuild/pipeline-overview/

NUT CI farm / Network UPS Tools projec... / nut / master / #1234 / Stages

Powered by DigitalOcean

#1234

Rerun

master · 5258b8f · networkupstools/nut · Push event to branch master at 9:27:01 AM on Jan 29, 2026 · Started 1 day 1 hr ago · Queued 9.3 sec · Took 7 hr 49 min on built-in · Changes · Artifacts

v2.8.5-rc2

Graph

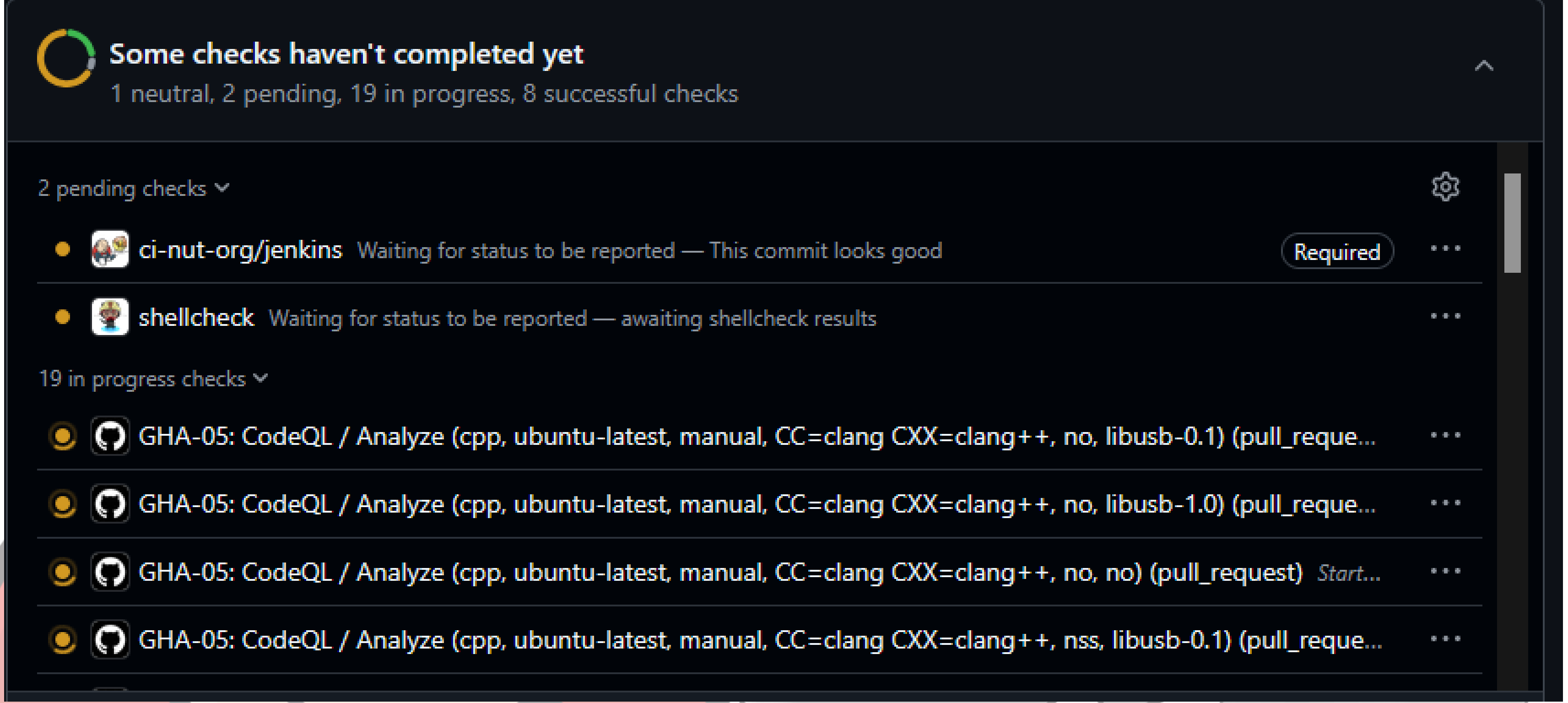
WITHAGENT: MATRIX_TAG="gnu99-gnu++11-clang-openbsd-6.5" && (COMPILER=CLANG&&OS_DISTRO=openbsd-6.5) && (nut-builder) && CSTDVARIANT=gnu&&CSTDVERSION_c=99&&CSTDVERSION_cxx=11 && LANG=C && LC_ALL=C && TZ=UTC && CONFIG_OPTS=--with-all=auto --with-docs=auto --with-ssl=auto --enable-Werror --enable-warnings --disable-Wcolor --enable-silent-rules :: as part of slowBuild filter: Default autotools driven build with default configuration, bitness and warning levels on each NUT CI farm platform (but with fatal warnings as of gnu99/gnu++11, must pass where enabled)

Search

- ✓ Initial discovery 16m
- ✓ Summarize quick-test results 1s
- ✓ Run the bigger dynamatrix (402 stages) 7h 31m
- ✓ Analyze the bigger dynamatrix 15s

✓ Analyze the bigger dynamatrix · 15s · Started 17h ago · 0-master-worker

✓ Discover Git reference build	`nut/nut/master`	0.21s
✓ Publish issues cre...	`{name=All Issues, qualityGates=[{threshold=1, type=TOTAL, unstable=true}], id=analysis, issues=[U...`	6s
✓ Discover Git reference build	`nut/nut/master`	0.28s
✓ Publish issues cre...	`{name=All Issues - Aggregated, qualityGates=[{threshold=1, type=TOTAL, unstable=true}], id=aggreg...`	



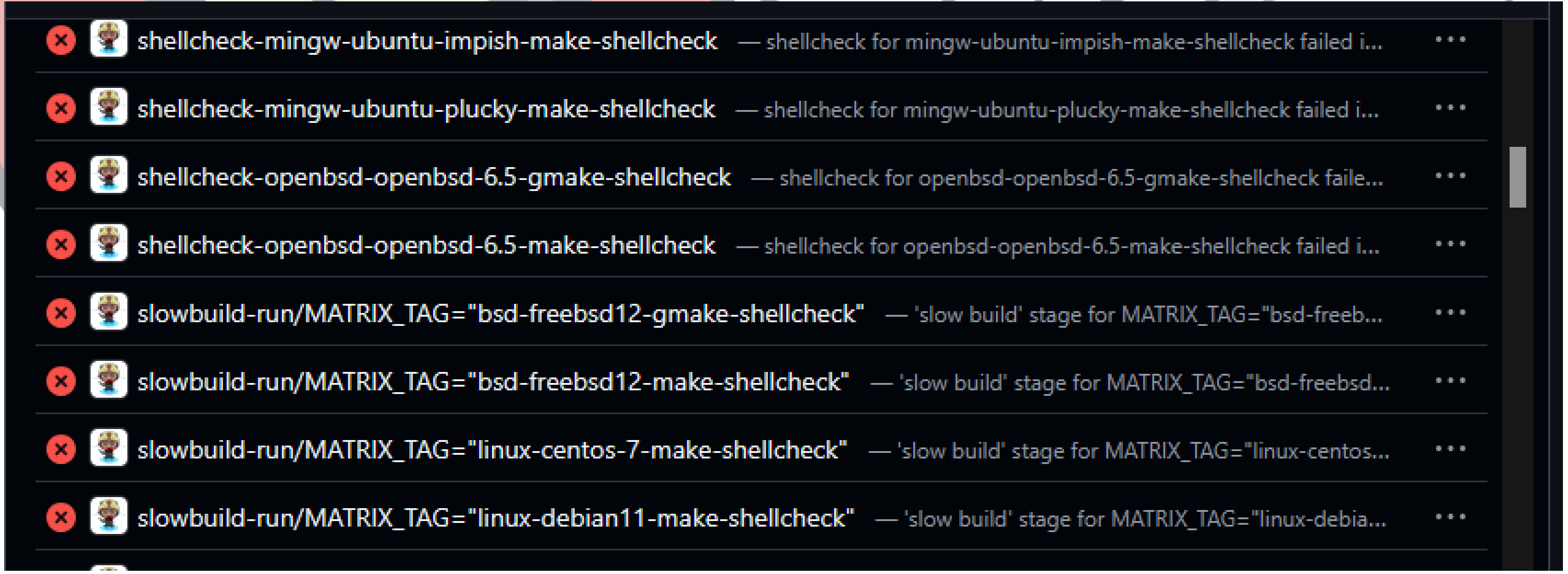
Some checks haven't completed yet
1 neutral, 2 pending, 19 in progress, 8 successful checks

2 pending checks

- ci-nut-org/jenkins Waiting for status to be reported — This commit looks good Required
- shellcheck Waiting for status to be reported — awaiting shellcheck results

19 in progress checks

- GHA-05: CodeQL / Analyze (cpp, ubuntu-latest, manual, CC=clang CXX=clang++, no, libusb-0.1) (pull_reque...
- GHA-05: CodeQL / Analyze (cpp, ubuntu-latest, manual, CC=clang CXX=clang++, no, libusb-1.0) (pull_reque...
- GHA-05: CodeQL / Analyze (cpp, ubuntu-latest, manual, CC=clang CXX=clang++, no, no) (pull_request) Start...
- GHA-05: CodeQL / Analyze (cpp, ubuntu-latest, manual, CC=clang CXX=clang++, nss, libusb-0.1) (pull_reque...



- shellcheck-mingw-ubuntu-impish-make-shellcheck — shellcheck for mingw-ubuntu-impish-make-shellcheck failed i...
- shellcheck-mingw-ubuntu-plucky-make-shellcheck — shellcheck for mingw-ubuntu-plucky-make-shellcheck failed i...
- shellcheck-openbsd-openbsd-6.5-gmake-shellcheck — shellcheck for openbsd-openbsd-6.5-gmake-shellcheck faile...
- shellcheck-openbsd-openbsd-6.5-make-shellcheck — shellcheck for openbsd-openbsd-6.5-make-shellcheck failed i...
- slowbuild-run/MATRIX_TAG="bsd-freebsd12-gmake-shellcheck" — 'slow build' stage for MATRIX_TAG="bsd-freeb...
- slowbuild-run/MATRIX_TAG="bsd-freebsd12-make-shellcheck" — 'slow build' stage for MATRIX_TAG="bsd-freebsd...
- slowbuild-run/MATRIX_TAG="linux-centos-7-make-shellcheck" — 'slow build' stage for MATRIX_TAG="linux-centos...
- slowbuild-run/MATRIX_TAG="linux-debian11-make-shellcheck" — 'slow build' stage for MATRIX_TAG="linux-debia...

Stress(-test)ing Jenkins itself

- ~300-700 dynamatrix scenarios (queue items) per NUT code iteration
- Some run more dependency-based combos internally
- Burning CPUs across the globe for 7+ hours per job
- Scripted transparent retries of queue items that failed due to loss of agent (networking, reboots, out of disk space…)
- Updating build stats as “short text”/”badges” (box in left column)
  - Cross-browser portable wrapping of long word-like tokens is a PITA
- While asking on Gitter about issues I’ve hit with the NUT CI farm, they rang a bell to people running Jenkins infra. Too few projects of similar massively-parallel scale (doing this on constrained resources)?

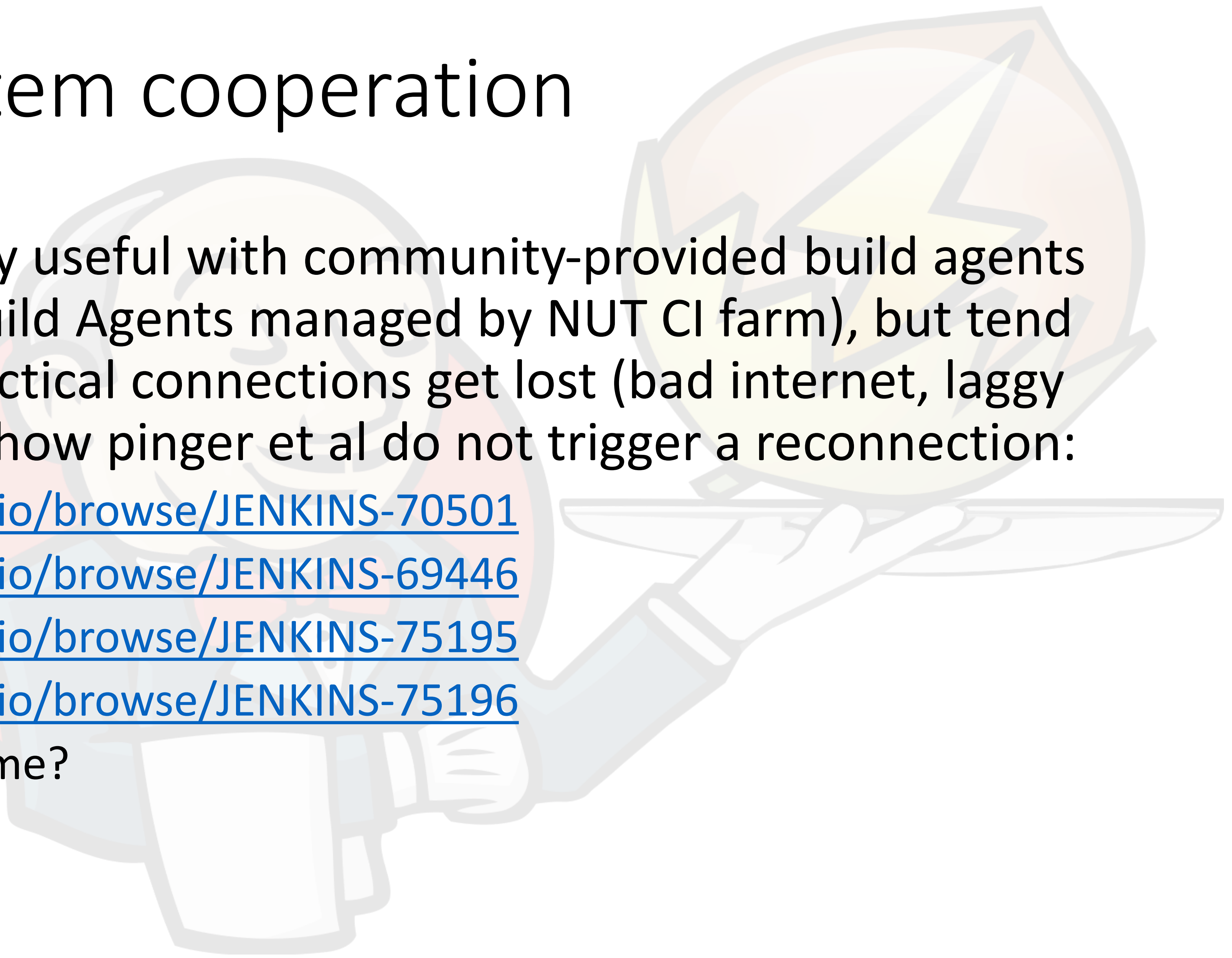
Stress(-test)ing Jenkins itself

- Infra-inflicted hiccups or failed builds are a pain, so some Jenkins issues were identified, reported, addressed…:
  - Race conditions in serialization (several plugins) => https://issues.jenkins.io/browse/JENKINS-76294
  - Agent loss/reconnection (especially swarm) => https://issues.jenkins.io/browse/JENKINS-75196 https://issues.jenkins.io/browse/JENKINS-75195
  - Native OS interaction (thread name setting) => https://github.com/jenkinsci/workflow-support-plugin/pull/345
  - Pipeline graph updates crashing Jenkins controller => https://github.com/jenkinsci/pipeline-graph-view-plugin/issues/862 https://github.com/jenkinsci/pipeline-graph-view-plugin/pull/887 https://github.com/jenkinsci/pipeline-graph-view-plugin/issues/888

Further ecosystem cooperation

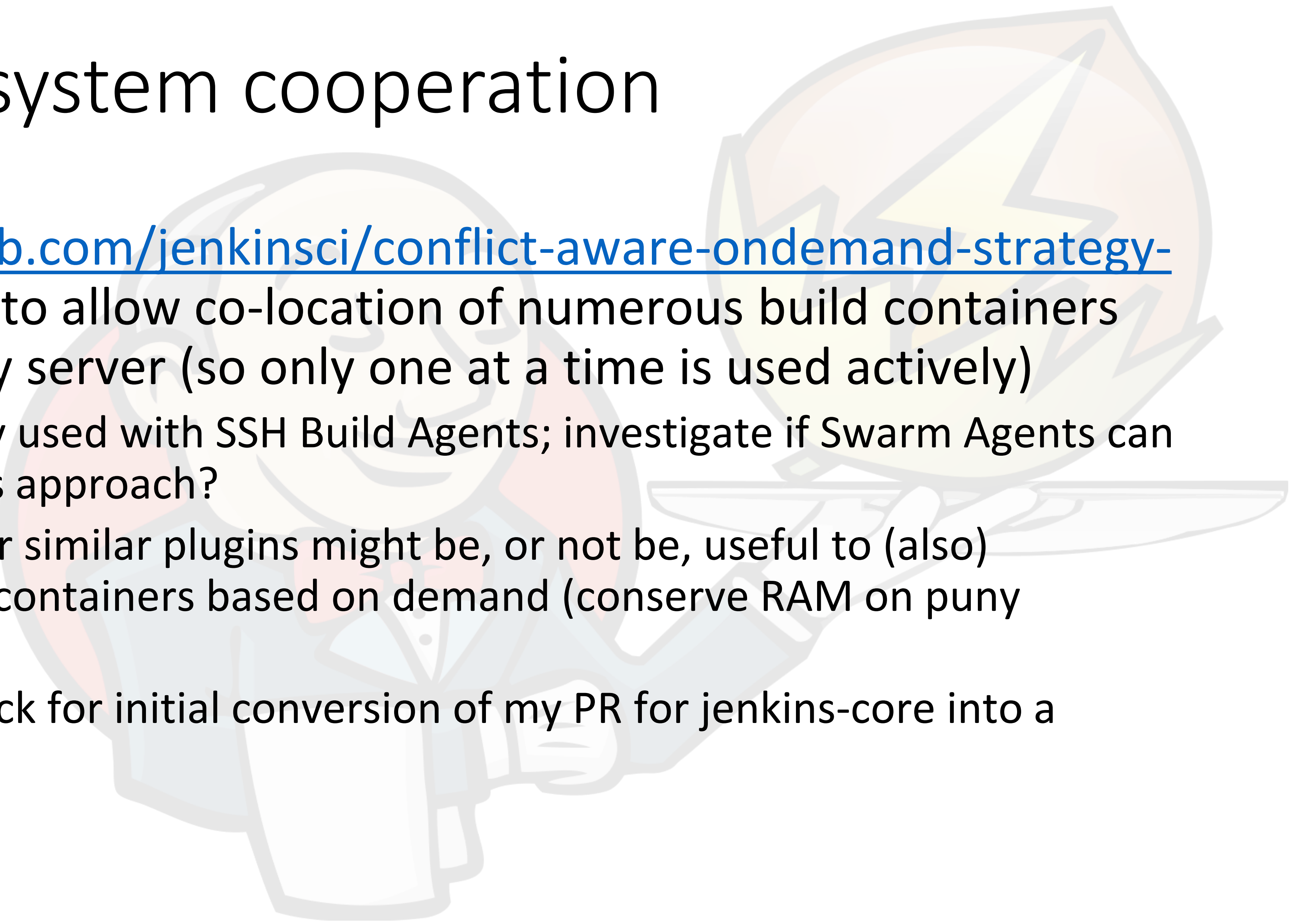
- Support for `@Library('libname@${BRANCH_NAME}')` helps with branches like staging/stable/fightwarn whose code can be identical but respective JSL variants might differ
  => https://github.com/jenkinsci/pipeline-groovy-lib-plugin/pull/19
- Builds of git-client with "Fanned-out Git refrepo" can help faster checkouts directly; some lessons learned in that PR are used in DynamatrixStash logic (e.g. `.gitcache-dynamatrix` directories)
  => https://github.com/jenkinsci/git-client-plugin/pull/644

Further ecosystem cooperation

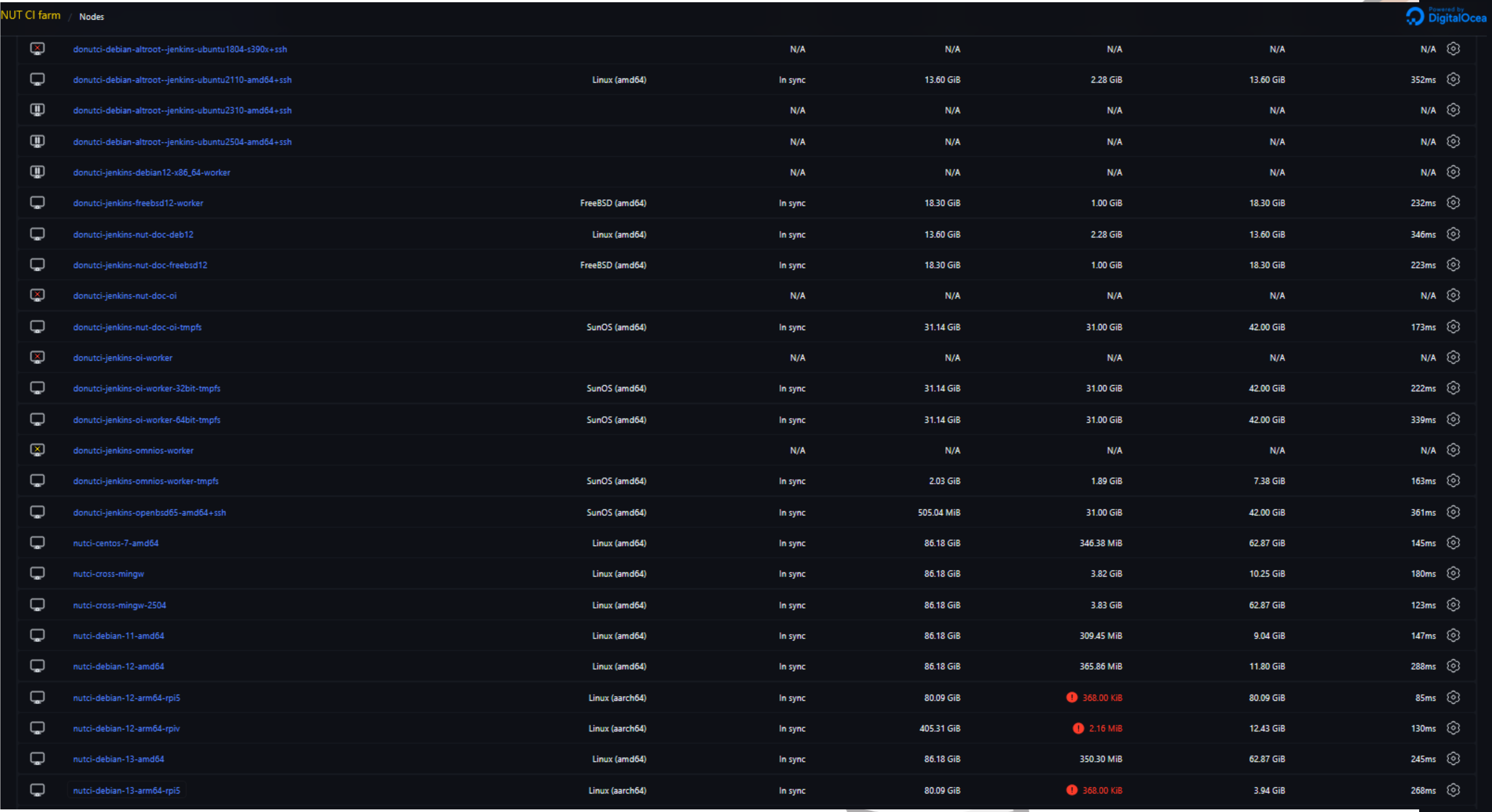


- Swarm Agents are very useful with community-provided build agents (as opposed to SSH Build Agents managed by NUT CI farm), but tend to be fragile when practical connections get lost (bad internet, laggy controller) - but somehow pinger et al do not trigger a reconnection:
  - https://issues.jenkins.io/browse/JENKINS-70501
  - https://issues.jenkins.io/browse/JENKINS-69446
  - https://issues.jenkins.io/browse/JENKINS-75195
  - https://issues.jenkins.io/browse/JENKINS-75196
  - Maybe I’ve missed some?

Further ecosystem cooperation

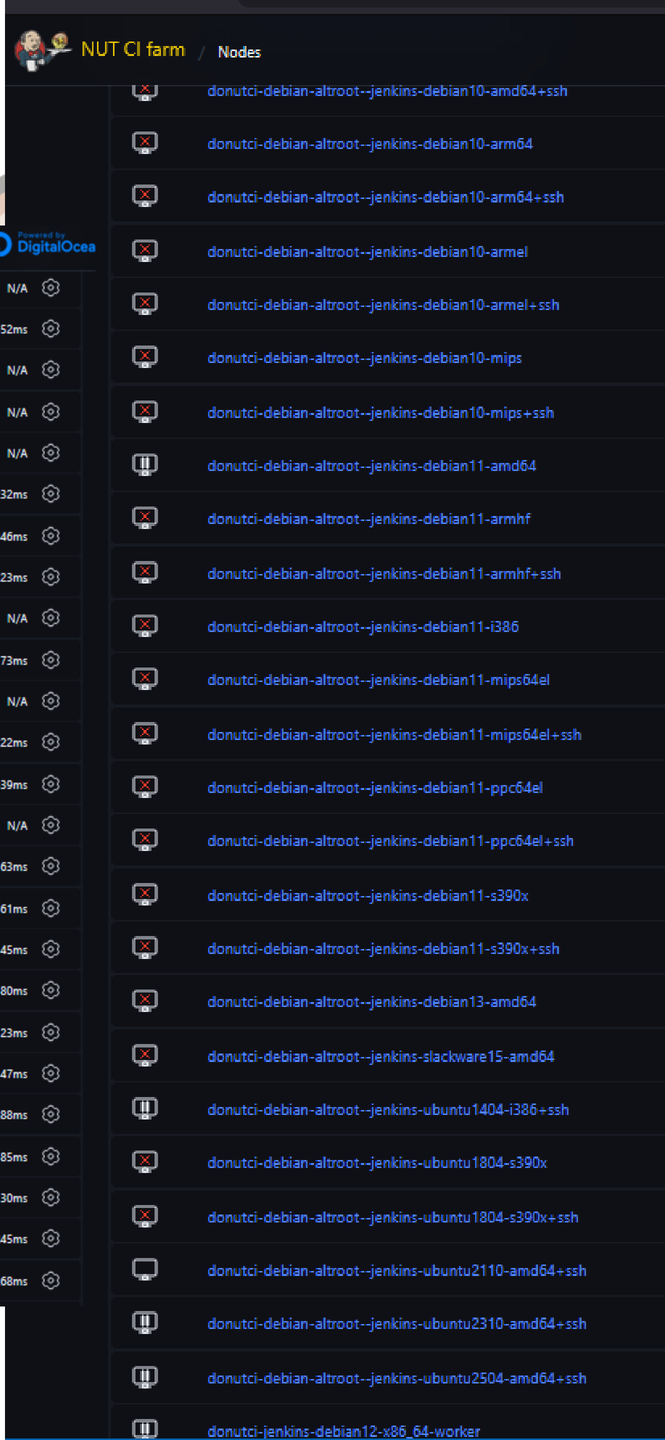


- The https://github.com/jenkinsci/conflict-aware-ondemand-strategy-plugin was made to allow co-location of numerous build containers on the same puny server (so only one at a time is used actively)
  - FIXME: Currently used with SSH Build Agents; investigate if Swarm Agents can benefit from this approach?
  - Scripted Cloud or similar plugins might be, or not be, useful to (also) start/shutdown containers based on demand (conserve RAM on puny servers)
  - Thanks to @dbeck for initial conversion of my PR for jenkins-core into a plugin ☺



NUT CI farm / Nodes

donutci-debian-altroot--jenkins-ubuntu1804-s390x+ssh		N/A	N/A	N/A	N/A	N/A
donutci-debian-altroot--jenkins-ubuntu2110-amd64+ssh	Linux (amd64)	In sync	13.60 GiB	2.28 GiB	13.60 GiB	352ms
donutci-debian-altroot--jenkins-ubuntu2310-amd64+ssh		N/A	N/A	N/A	N/A	N/A
donutci-debian-altroot--jenkins-ubuntu2504-amd64+ssh		N/A	N/A	N/A	N/A	N/A
donutci-jenkins-debian12-x86_64-worker		N/A	N/A	N/A	N/A	N/A
donutci-jenkins-freebsd12-worker	FreeBSD (amd64)	In sync	18.30 GiB	1.00 GiB	18.30 GiB	232ms
donutci-jenkins-nut-doc-deb12	Linux (amd64)	In sync	13.60 GiB	2.28 GiB	13.60 GiB	346ms
donutci-jenkins-nut-doc-freebsd12	FreeBSD (amd64)	In sync	18.30 GiB	1.00 GiB	18.30 GiB	223ms
donutci-jenkins-nut-doc-oi		N/A	N/A	N/A	N/A	N/A
donutci-jenkins-nut-doc-oi-tmpfs	SunOS (amd64)	In sync	31.14 GiB	31.00 GiB	42.00 GiB	173ms
donutci-jenkins-oi-worker		N/A	N/A	N/A	N/A	N/A
donutci-jenkins-oi-worker-32bit-tmpfs	SunOS (amd64)	In sync	31.14 GiB	31.00 GiB	42.00 GiB	222ms
donutci-jenkins-oi-worker-64bit-tmpfs	SunOS (amd64)	In sync	31.14 GiB	31.00 GiB	42.00 GiB	339ms
donutci-jenkins-omnios-worker		N/A	N/A	N/A	N/A	N/A
donutci-jenkins-omnios-worker-tmpfs	SunOS (amd64)	In sync	2.03 GiB	1.89 GiB	7.38 GiB	163ms
donutci-jenkins-openbsd65-amd64+ssh	SunOS (amd64)	In sync	505.04 MiB	31.00 GiB	42.00 GiB	361ms
nutci-centos-7-amd64	Linux (amd64)	In sync	86.18 GiB	346.38 MiB	62.87 GiB	145ms
nutci-cross-mingw	Linux (amd64)	In sync	86.18 GiB	3.82 GiB	10.25 GiB	180ms
nutci-cross-mingw-2504	Linux (amd64)	In sync	86.18 GiB	3.83 GiB	62.87 GiB	123ms
nutci-debian-11-amd64	Linux (amd64)	In sync	86.18 GiB	309.45 MiB	9.04 GiB	147ms
nutci-debian-12-amd64	Linux (amd64)	In sync	86.18 GiB	365.86 MiB	11.80 GiB	288ms
nutci-debian-12-arm64-rpi5	Linux (aarch64)	In sync	80.09 GiB	! 368.00 KiB	80.09 GiB	85ms
nutci-debian-12-arm64-rpiv	Linux (aarch64)	In sync	405.31 GiB	! 2.16 MiB	12.43 GiB	130ms
nutci-debian-13-amd64	Linux (amd64)	In sync	86.18 GiB	350.30 MiB	62.87 GiB	245ms
nutci-debian-13-arm64-rpi5	Linux (aarch64)	In sync	80.09 GiB	! 368.00 KiB	3.94 GiB	268ms



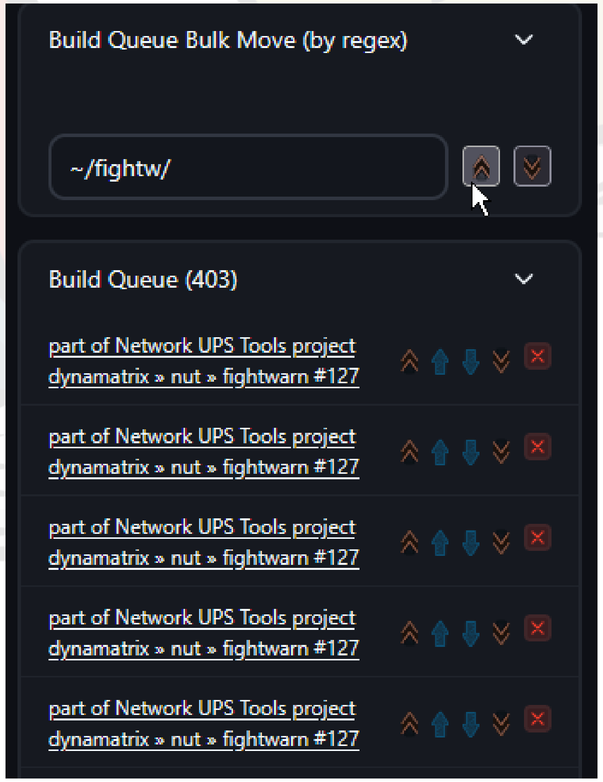
NUT CI farm / Nodes

- donutci-debian-altroot--jenkins-debian10-amd64+ssh
- donutci-debian-altroot--jenkins-debian10-arm64
- donutci-debian-altroot--jenkins-debian10-arm64+ssh
- donutci-debian-altroot--jenkins-debian10-armel
- donutci-debian-altroot--jenkins-debian10-armel+ssh
- donutci-debian-altroot--jenkins-debian10-mips
- donutci-debian-altroot--jenkins-debian10-mips+ssh
- donutci-debian-altroot--jenkins-debian11-amd64
- donutci-debian-altroot--jenkins-debian11-armhf
- donutci-debian-altroot--jenkins-debian11-armhf+ssh
- donutci-debian-altroot--jenkins-debian11-i386
- donutci-debian-altroot--jenkins-debian11-mips64el
- donutci-debian-altroot--jenkins-debian11-mips64el+ssh
- donutci-debian-altroot--jenkins-debian11-ppc64el
- donutci-debian-altroot--jenkins-debian11-ppc64el+ssh
- donutci-debian-altroot--jenkins-debian11-s390x
- donutci-debian-altroot--jenkins-debian11-s390x+ssh
- donutci-debian-altroot--jenkins-debian13-amd64
- donutci-debian-altroot--jenkins-slackware15-amd64
- donutci-debian-altroot--jenkins-ubuntu1404-i386+ssh
- donutci-debian-altroot--jenkins-ubuntu1804-s390x
- donutci-debian-altroot--jenkins-ubuntu1804-s390x+ssh
- donutci-debian-altroot--jenkins-ubuntu2110-amd64+ssh
- donutci-debian-altroot--jenkins-ubuntu2310-amd64+ssh
- donutci-debian-altroot--jenkins-ubuntu2504-amd64+ssh
- donutci-jenkins-debian12-x86_64-worker

Further ecosystem cooperation

Queue management

- https://github.com/jenkinsci/simple-queue-plugin/pull/17 and nearby PRs for that plugin
- Need obscure views like checkFingerprints nowadays to see and manage the queue



Build Queue Bulk Move (by regex)

~/fightw/

Build Queue (403)

part of Network UPS Tools project dynamatrix » nut » fightwarn #127

part of Network UPS Tools project dynamatrix » nut » fightwarn #127

part of Network UPS Tools project dynamatrix » nut » fightwarn #127

part of Network UPS Tools project dynamatrix » nut » fightwarn #127

part of Network UPS Tools project dynamatrix » nut » fightwarn #127

Lessons learned: Tunneled SSH agents

- In some cases, the tested operating system is too old to have a modern JDK available in any manner. In other cases, `qemu-static` virtualization of foreign CPU architectures is not working for a JVM.
- It helps to run an SSH agent to a system with modern JDK (controller localhost, hypervisor with legacy/emulated Linux containers, even a neighboring VM with shared NFS) which then uses carefully crafted prefix/suffix for command lines with further SSH hop, that in effect run them in the older operating environment.

Lessons learned: Tunneled SSH agents

- As far as Jenkins Remoting is concerned, it works with a local file system. So same path names to same files must exist on both the “proxy” and actual builder.
- SSH server on worker must allow passing envvars (`AcceptEnv *`)
- One of the ways to set this up is documented at https://networkupstools.org/docs/user-manual.chunked/_connecting_jenkins_to_the_containers.html#_where_to_run_agent_jar => “Using Jenkins SSH Build Agents” section

Lessons learned: Tunneled SSH agents

- Example of OpenBSD 6.5 and Solaris 8 with NFS to younger neighbors:
  - Remote root directory (as seen on both actual worker and intermediate host): `/export/home/abuild/jenkins-nutci-openbsd-65-amd64`
  - Prefix Start Agent Command: `echo PING > /dev/tcp/nutci-openbsd-65-amd64/22 && test -w /var/shm/jenkins-nutci-openbsd-65-amd64/ &&`
  - `CI_WRAP_SH=ssh -o SendEnv='*' "nutci-openbsd-65-amd64" /bin/sh -xe`
- Similar for Linux containers with host filesystem visibility

Lessons learned: TMPFS builds

- Many cases automated by jenkins-swarm-nutci
- Some set up manually with SSH Build agents
  - Remote root directory: `/tmp/jenkins-nut-32bit`
  - Prefix Start Agent Command: `rm -rf /tmp/jenkins-nut-32bit/workspace ; mkdir -p /tmp/jenkins-nut-32bit/workspace && /usr/gnu/bin/ln -frs /export/home/abuild/.gitcache-dynamatrix{,@tmp} /tmp/jenkins-nut-32bit/workspace/ &&`

Some links

- https://stories.jenkins.io/user-story/jenkins-is-the-way-for-networkupstools/ - “**Jenkins is the way to build multi-platform NUT, and jenkinsfile-dynamatrix is the way to find what can be built today**”
- https://github.com/networkupstools/jenkins-dynamatrix - project repository for the Jenkins Shared Library
- https://github.com/networkupstools/jenkins-swarm-nutci - scripting for Jenkins Swarm agents (as a service under systemd, SMF, launchd, upstart, rc.d, init…) to dial into the project’s controller and offer themselves by labels as builders for the platform and toolkits there

Some links

- https://github.com/networkupstools/nut/blob/master/Jenkinsfile-dynamatrix - practical example of the pipeline preparing it for a large project (NUT)
- https://github.com/networkupstools/nut/blob/master/ci_build.sh - practical adaptation of feature-driven CI runs under numerous engines and developer workstations alike for the autotools based C/C++ project (NUT)
- https://github.com/networkupstools/libmodbus/blob/rtu_usb_jf/Jenkinsfile-dynamatrix - a much smaller example pipeline for one autotools setup of a simpler project