iRODS®

# iRODS Development and Testing Environments (v8)

Alan King
Senior Software Developer
iRODS Consortium

July 5-8, 2022
iRODS User Group Meeting 2022
Leuven, Belgium

# Overview

- Motivation
- History
- Limitations of the past
- Development Environment
- Testing Environment
- Future Work

# Motivation

Passing tests build confidence in the changes developers make to software, asserting the correctness of the changes and of the entire system.

Therefore, building and testing iRODS should be easy and consistent for everybody.

# History of Build-and-Test Systems at iRODS

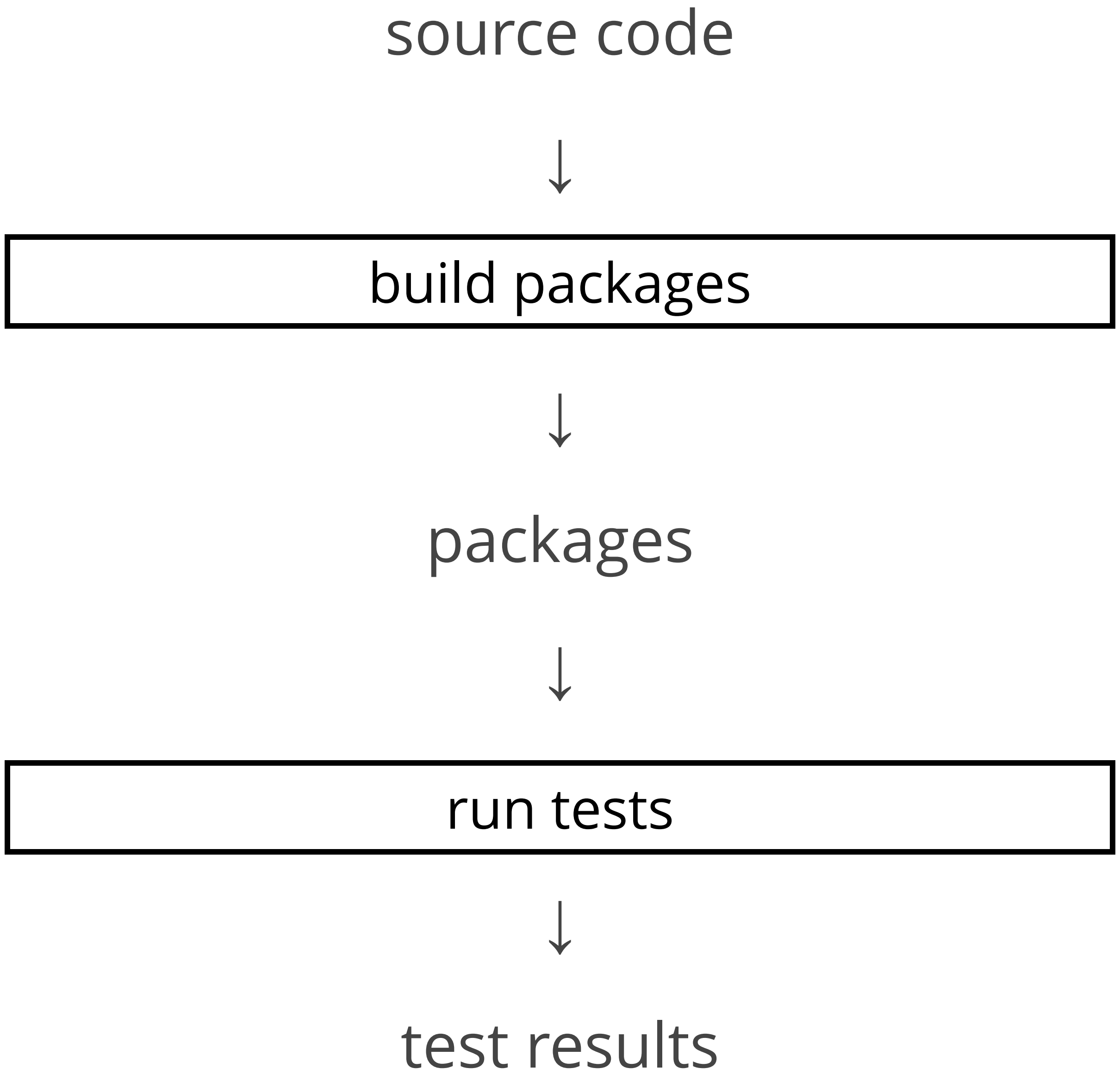
- **v1 - July 2011:** Python → Node.js → RabbitMQ → Celery → Eucalyptus
- **v2 - October 2012:** Python → Node.js → ssh → OpenStack
- **v3 - January 2013:** Hudson → Python → OpenStack
- **v4 - October 2013:** Hudson → Python → vSphere long-running VMs
- **v5 - Spring 2015:** Jenkins → Python → Ansible → zone_bundles → vSphere dynamic VMs
- **v6 - Spring 2017:** Jenkins → Python → vSphere dynamic VMs → build/test hooks
- **v7 - Summer 2019:** Docker → Jenkins → Python → Docker → build/test hooks

# Limitations of the past

- Tied to Jenkins
  - Everybody is required to be an administrator
  - Relatively inflexible
  - Difficult to maintain
- Difficult to manage test results and built package output
- Docker image explosion (one tag per test run(!))

Bottom line: Our needs as developers and project maintainers were not being met satisfactorily.

# Basic Build-and-Test Workflow



source code

↓

build packages

↓

packages

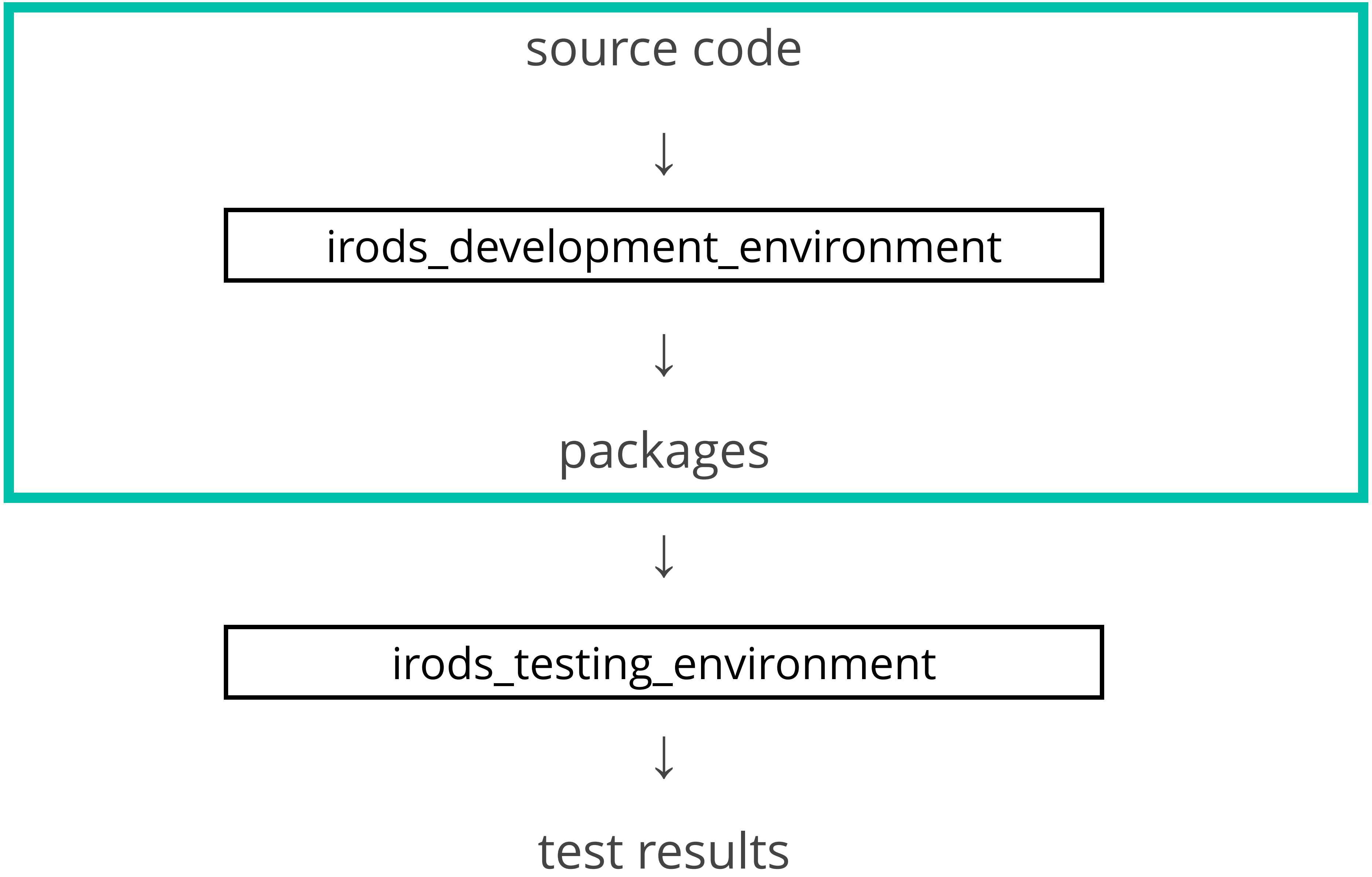
↓

run tests

↓

test results

# v8 - 2021/2022



source code

↓

irods_development_environment

↓

packages

↓

irods_testing_environment

↓

test results

# iRODS Development Environment

https://github.com/irods/irods_development_environment

Containerized package builders for iRODS, plugins, and externals for all supported OS's, plus debugging tools.

- Builds local code and produces local packages
- Runs the standard build process(es) in containers

Advantages:

- No repeated cloning or copying repositories required
  - Developer makes edits with preferred local tools
- Build cache allows for faster iteration
- Consistent process for development and release

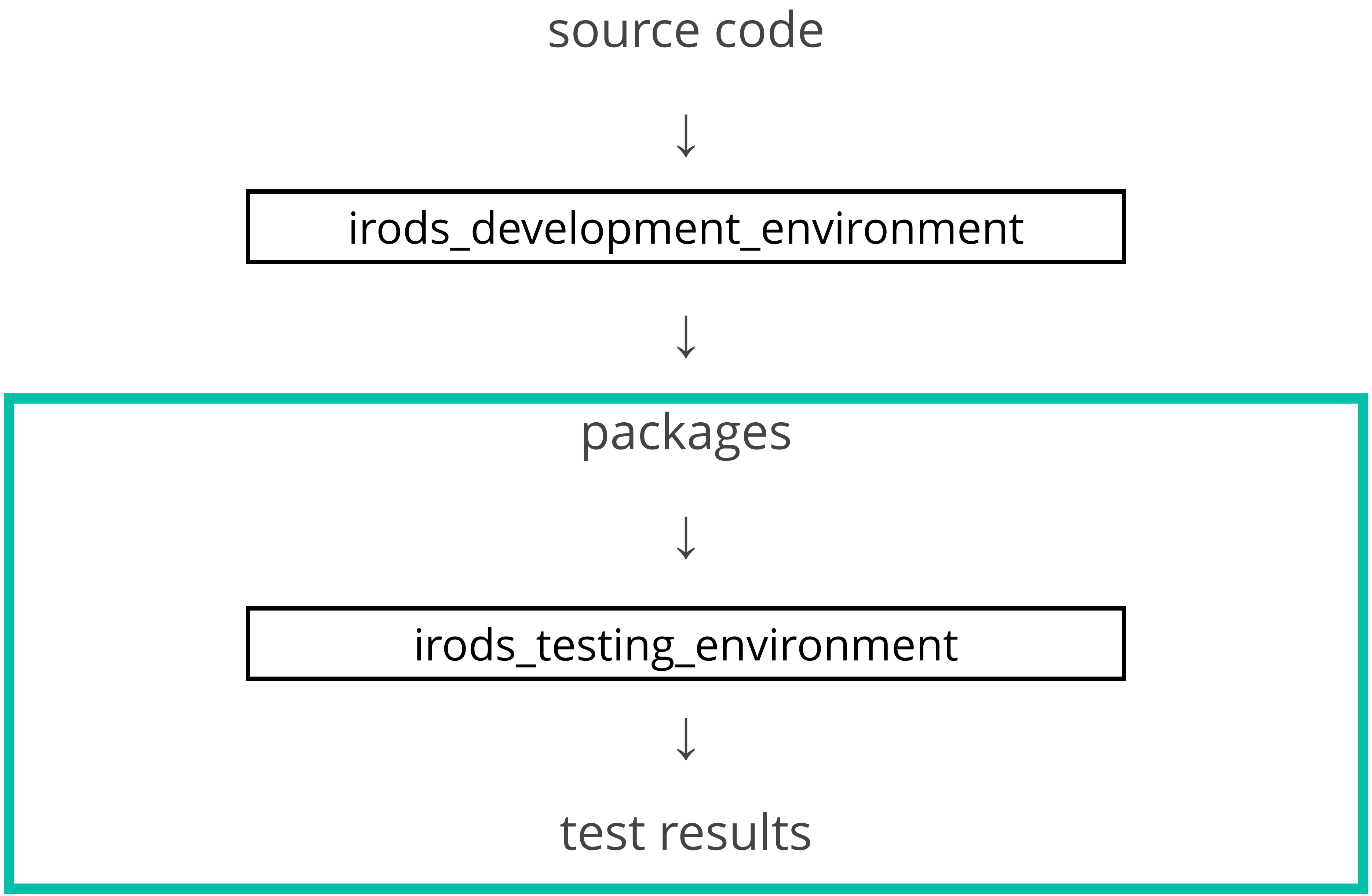
# iRODS Development Environment: Example Usage

```
$ docker run --rm \
           -v ${irods_sourcedir}:/irods_source:ro \
           -v ${irods_builddir}:/irods_build \
           -v ${icommands_sourcedir}:/icommands_source:ro \
           -v ${icommands_builddir}:/icommands_build \
           -v ${irods_packagedir}:/irods_packages \
           -v ${externals_packagedir}:/irods_externals_packages \
           irods-core-builder-m:${PLATFORM}-${VERSION}
```

## Run a Docker container that builds packages

- Volume mounts for **source code** (input)
- Volume mount for build cache and built **packages** (output)
- Use separate Docker image tags for each OS/version
- Optionally use custom iRODS externals packages

# v8 - 2021/2022



source code

↓

irods_development_environment

↓

packages

↓

irods_testing_environment

↓

test results

# iRODS Testing Environment

https://github.com/irods/irods_testing_environment

Scripts built on python library which uses Docker Compose to stand up iRODS zones in various configurations and run tests.

- Local scripts execute commands in long-running containers
- Uses local or released packages and create local test results
- Scripts can run tests or just stand up iRODS zone(s)

Advantages:

- Precision controls for running tests in parallel
- Convenient way to reproduce issues
- Consistent process for bench and automated testing

# iRODS Testing Environment: Capabilities

- Stand up long-running, topological iRODS zone(s)
  - stand_it_up.py - stand up a zone
  - federate.py - stand up and federate multiple zones
- Run iRODS tests in various configurations
  - run_core_tests.py - run iRODS server tests
  - run_unit_tests.py - run unit tests for iRODS libraries
  - run_topology_tests.py - run tests on multi-server zone
  - run_federation_tests.py - run tests in federated zones
  - run_plugin_tests.py - run tests for iRODS plugins

# iRODS Testing Environment: Running Tests

```
$ python run_core_tests.py \
  --project-directory projects/ubuntu-20.04/ubuntu-20.04-postgres-10.12 \
  --irods-package-directory ~/hdd/builds/irods_packages/4-3-stable/ubuntu-20.04 \
  --concurrent-test-executor-count 4
```

- Stand up 4 iRODS zones using locally built packages
- Distribute the test suite between the zones and run each set in parallel
- Copy test results and log files to host machine
- Tear down the 4 zones and remove containers

# iRODS Testing Environment: Running Tests

```
-----
results for [ubuntu-2004-postgres-1012_irods-catalog-provider_2]
        passed tests:
                [[   3.1020]s]  [test_auth]
                [[ 386.7651]s]  [test_delay_queue]
                [[  13.6752]s]  [test_icd]
                [[ 176.7343]s]  [test_icp]
                [[   4.6180]s]  [test_ihelp]
                [[  27.8215]s]  [test_imeta_admin_mode]
                [[ 269.8607]s]  [test_imeta_set]
                [[  20.6078]s]  [test_itrim]
                [[  12.2143]s]  [test_load_balanced_suite]
                [[   2.7831]s]  [test_prep_genquery_iterator]
                [[1109.4858]s]  [test_resource_types.Test_Resource_Compound]
                [[1018.1528]s]  [test_resource_types.Test_Resource_MultiLayered]
                [[1009.2253]s]  [test_resource_types.Test_Resource_RandomWithinRandom]
                [[1542.5102]s]  [test_resource_types.Test_Resource_ReplicationToTwoCompoundResourcesWithPreferArchive]
                [[  69.4207]s]  [test_rule_engine_plugin_framework]
                [[  35.5109]s]  [test_ssl]
        skipped tests:
        failed tests:
        return code:[0]
        time elapsed: [6.723e+03]seconds ([   1]hours [    52.04]minutes)
-----
```

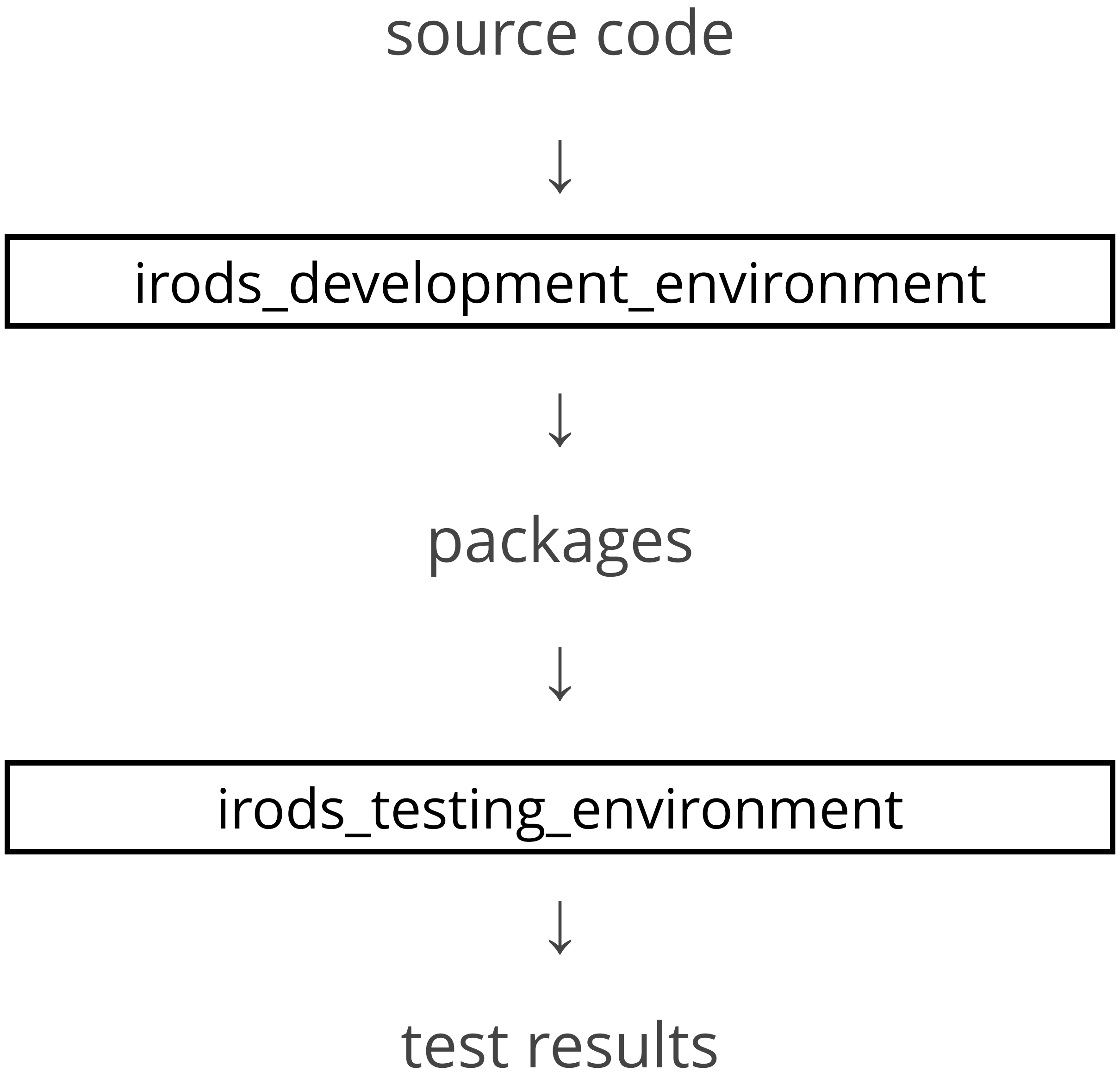
# iRODS Testing Environment: Running Tests

```
-----
results for [ubuntu-2004-postgres-1012_irods-catalog-provider_4]
        passed tests:
                [[  30.0808]s]  [test_collection_mtime]
                [[ 837.2263]s]  [test_iadmin]
                [[  59.8457]s]  [test_ichmod]
                [[  13.3225]s]  [test_ifsck]
                [[  58.9188]s]  [test_ils]
                [[   8.9275]s]  [test_imeta_help]
                [[  34.9601]s]  [test_imv]
<snip>
                [[  28.2190]s]  [test_quotas]
                [[1081.7692]s]  [test_resource_types.Test_Resource_CompoundWithUnivmss]
                [[ 808.2622]s]  [test_resource_types.Test_Resource_Passthru]
                [[2091.9287]s]  [test_resource_types.Test_Resource_Replication]
                [[ 857.6486]s]  [test_resource_types.Test_Resource_Unixfilesystem]
                [[ 917.5663]s]  [test_rulebase]
                [[  78.1721]s]  [test_symlink_operations]
        skipped tests:
        failed tests:
        return code:[0]
        time elapsed: [7.345e+03]seconds ([   2]hours [    2.424]minutes)
-----
```

# iRODS Testing Environment: Running Tests

```
All tests passed! :)
time elapsed: [10955.3559]seconds ([   3]hours [ 2.5893]minutes)
==== end of test run results ====

2022-07-04 20:57:00,726 ERROR - collecting logs [/tmp/ubuntu-2004-postgres-
10123x3tjb2r/ubuntu-2004-postgres-1012_be703715-7901-4a34-affa-10ebea651ff4]
```

# Basic Build-and-Test Workflow



source code

↓

irods_development_environment

↓

packages

↓

irods_testing_environment

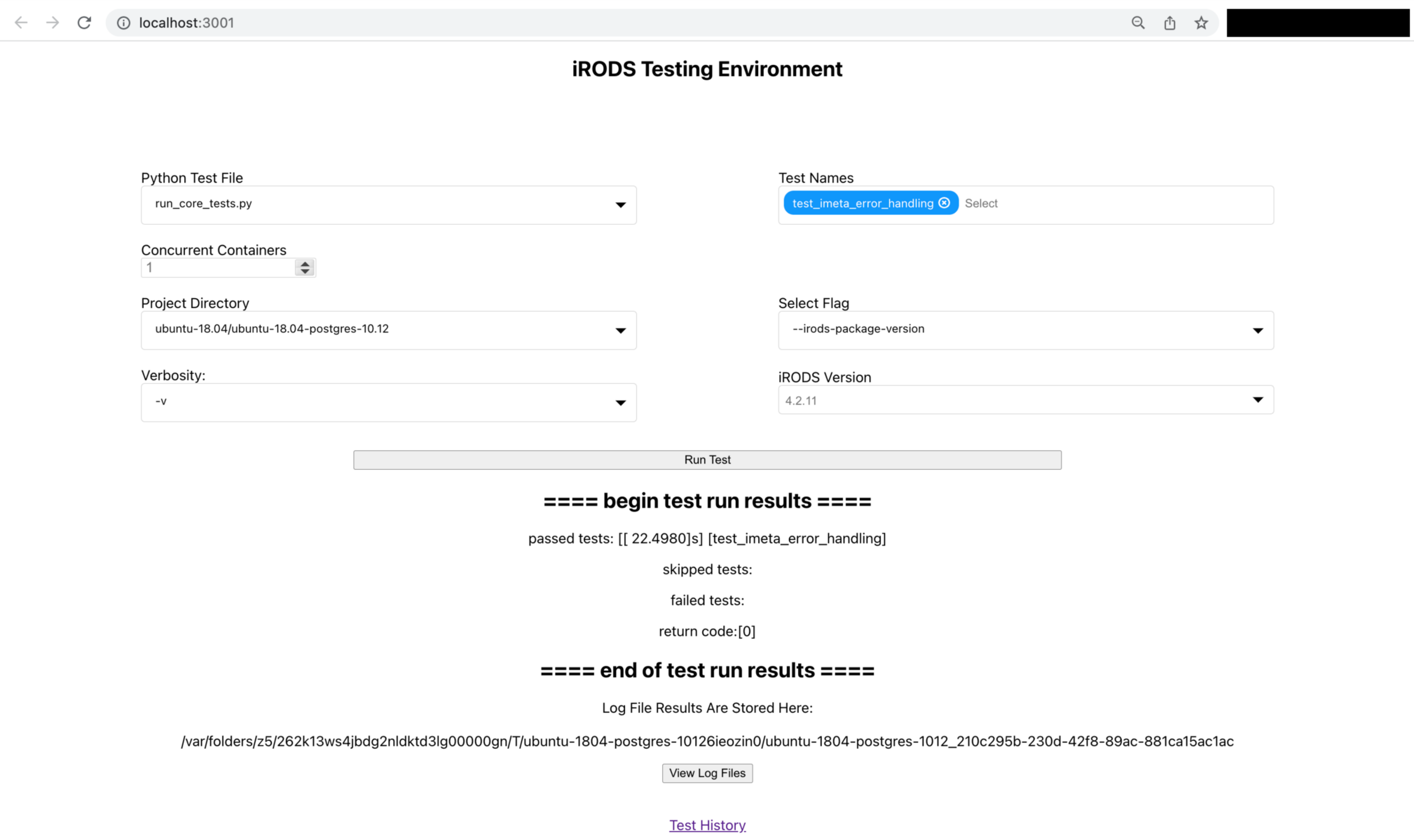
↓

test results

# Future Work and Dreams

- Web application
  - Automation/CI
- Client build/test
- Environment reproduction (zone report)
- Execution environment agnosticism (Docker, podman, VMs)
- Orchestration

# The Build-and-Test Monitor



localhost:3001 [redacted]

## iRODS Testing Environment

Python Test File: run_core_tests.py

Test Names: test_imeta_error_handling ⊗ Select

Concurrent Containers: 1

Project Directory: ubuntu-18.04/ubuntu-18.04-postgres-10.12

Select Flag: --irods-package-version

Verbosity: -v

iRODS Version: 4.2.11

### ==== begin test run results ====

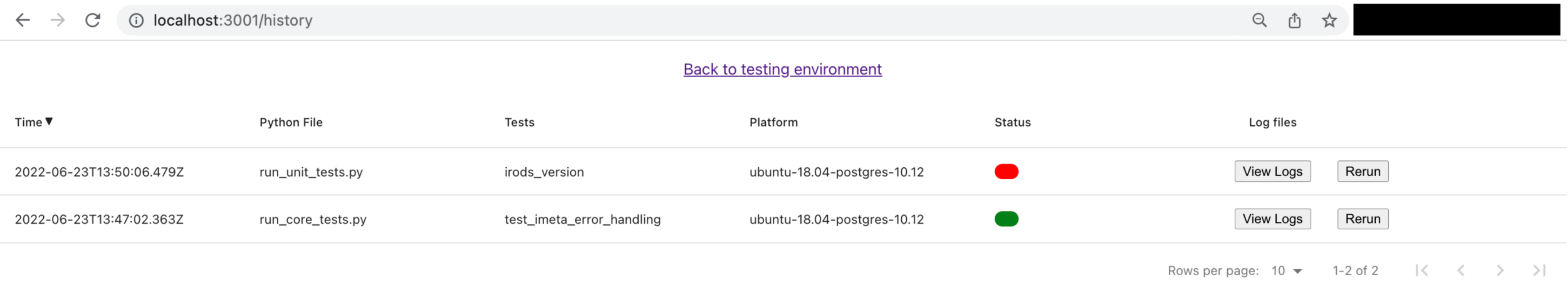
```
passed tests: [[ 22.4980]s] [test_imeta_error_handling]

skipped tests:

failed tests:

return code:[0]
```

### ==== end of test run results ====

Log File Results Are Stored Here:

/var/folders/z5/262k13ws4jbdg2nldktd3lg00000gn/T/ubuntu-1804-postgres-10126ieozin0/ubuntu-1804-postgres-1012_210c295b-230d-42f8-89ac-881ca15ac1ac

# The Build-and-Test Monitor



localhost:3001/history [redacted]

Back to testing environment

| Time ▼ | Python File | Tests | Platform | Status | Log files |
|---|---|---|---|---|---|
| 2022-06-23T13:50:06.479Z | run_unit_tests.py | irods_version | ubuntu-18.04-postgres-10.12 | 🟥 | View Logs Rerun |
| 2022-06-23T13:47:02.363Z | run_core_tests.py | test_imeta_error_handling | ubuntu-18.04-postgres-10.12 | 🟩 | View Logs Rerun |

Rows per page: 10 ▾ 1-2 of 2

We are seeking community participation!

Please give it a whirl and let us know how it goes.

Thanks for listening