



iRODS Build and Test v9. Automation via GitHub and Kubernetes

A presentation by:

Phil Owen

POwen@RENCI.org

iRODS User Group Meeting 2024
May 28-31 - Amsterdam, Netherlands



Scan to follow along or for
review later

iRODS Testing - What to expect in this discussion

Presentation outline:

- iRODS testing - Provide some context and our motivation.
- Solution - Using Kubernetes and GitHub to craft a novel way to test iRODS.
- Impact - How our work has had a positive effect.
- Future vision - Where do we go from here.

iRODS Testing - Motivation

Improve iRODS testing performance and ease of use.

Testing iRODS presents a number of challenges. Creating and maintaining complex test environments can be time-consuming.

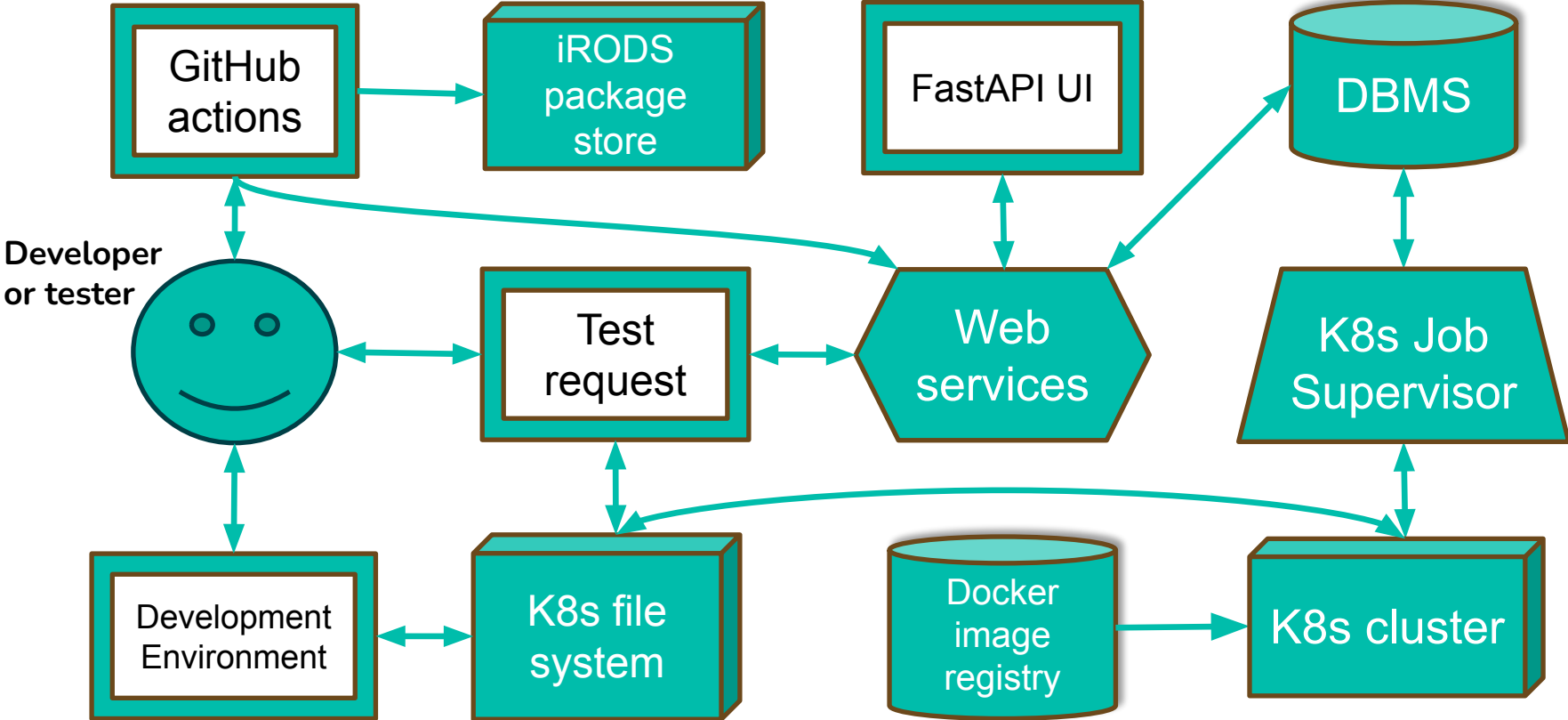
We came up with this feature wish list:

- Scale to accommodate a growing development audience.
- Automatically provision iRODS provider/consumer topologies.
- Support automation and integration with other services.
- Support standardized testing environments and strategies.
- Support the timely distribution of evolving testing paradigms.
- Provide an automated forensic analysis of test results.
- Provide automatically generated code certification documentation.

iRODS Testing - Infrastructure overview

- **GitHub** - Source code repository. Automated iRODS package build/push and testing requests are spawned by GitHub Actions.
- **Docker container image registry** - Storage for Docker images of operating systems, databases and components.
- **Kubernetes cluster** - On-prem with two namespaces assigned to iRODS (development and production).
- **Web-based interfaces** - Two user interfaces and 17 web-services implemented for this application (FastAPI and React).
- **Application supporting services** - Application services and components are implemented as Kubernetes deployments and cronjobs.
- **File systems** - Storage for iRODS package builds, testing results and forensics.

iRODS Testing – Solution landscape (high-level)



What is Kubernetes (K8s)

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

The name Kubernetes originates from Greek, meaning helmsman or pilot. K8s as an abbreviation results from counting the eight letters between the "K" and the "s".

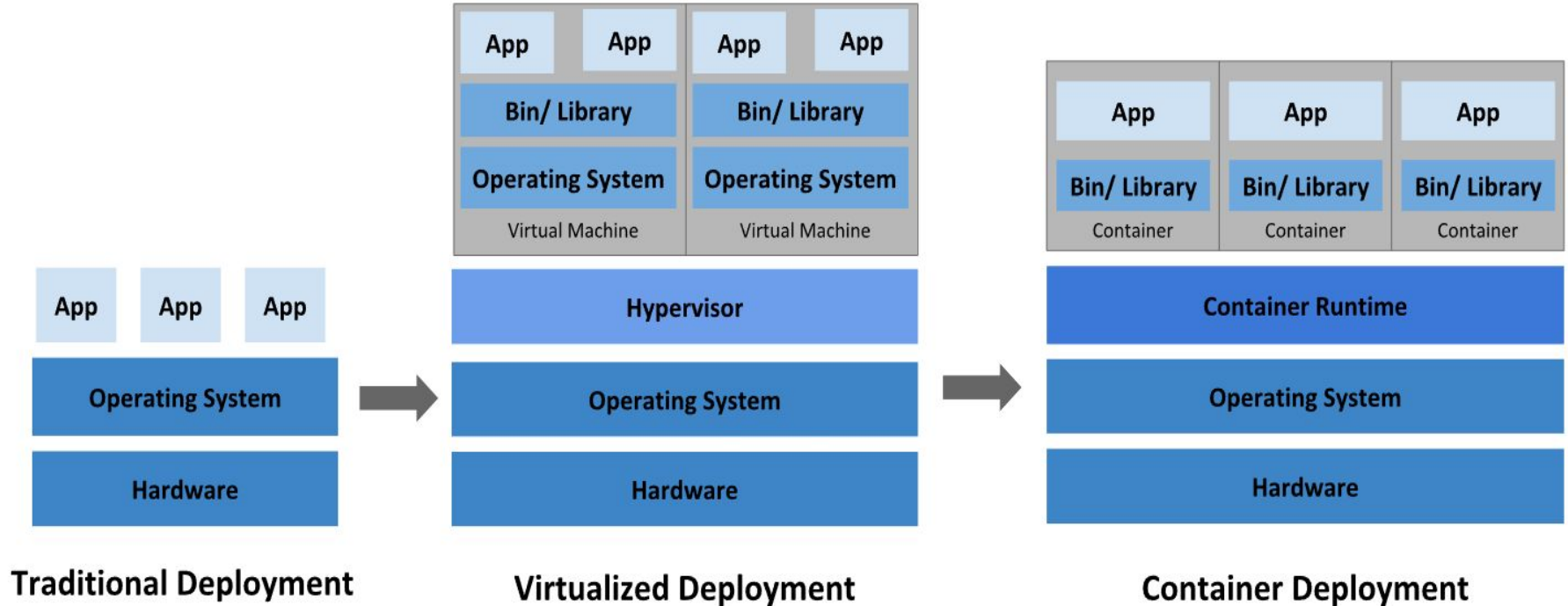
Google open-sourced the Kubernetes project in 2014.



kubernetes

Source: <https://kubernetes.io/docs/concepts/overview/>

Why Docker containers and Kubernetes?



Source: <https://kubernetes.io/docs/concepts/overview/>

iRODS Testing - Application services on K8s

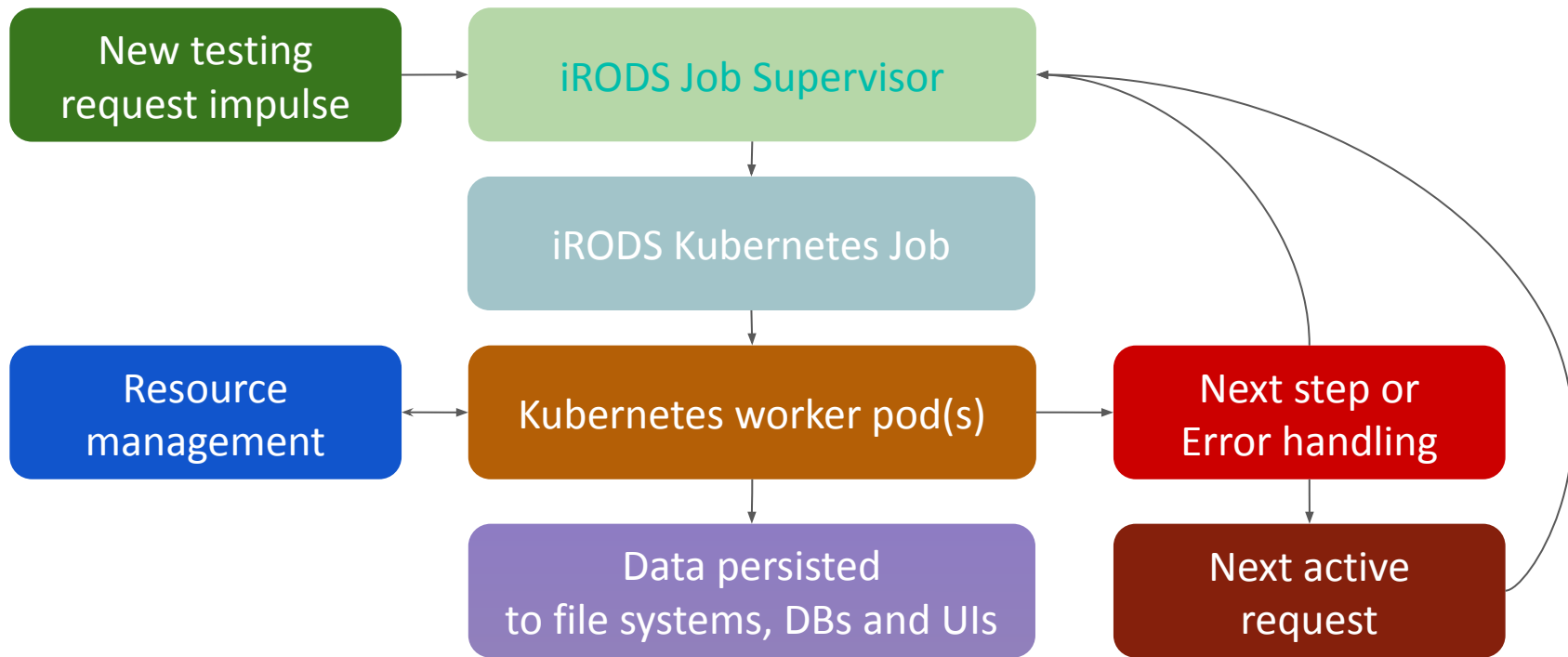
- **K8s Job Supervisor** - Workflow manager that creates and monitors K8s jobs for each data processing component.
- **Job Supervisor settings UI** - FastAPI application used to provide web service interfaces that provide access to configurations and data.
- **Job Supervisor database** - PostgreSQL DBMS to store the Job Supervisor configurations and test requests.
- **iRODS Test request UI** - A web interface (React.js) where a developer/tester can manually make and/or monitor a testing request.
- **Cron Jobs** - Scheduled processes launched for data archival and database backups.

iRODS Testing - K8s Job supervisor features

The iRODS Job supervisor is a workflow manager that has been implemented on a K8s cluster to provide:

- **Programmatically interfaces** with K8s using the K8s API.
- **Creation and sequencing** of K8s jobs in order (or in parallel).
- **Monitors** each K8s job from start to completion.
- **Removes** job and resource allocations on completion.
- **Error handling** including pod retries and resource waiting.
- **Job characteristics** are stored in a database which are adjustable via web services or UI (FastAPI).

iRODS Testing - Job Supervisor process flow



iRODS Testing - Job Supervisor workflow features

Each job supervisor workflow type equates to an iRODS topology. A workflow is a linked list of process step with configuration definitions that include:

- A Docker image.
- Run-time resource allocation (memory, cpu, ephemeral space, etc.).
- NFS and K8s file system volume mounts.
- Parameterized component command line(s).
- Multiple containers per pod and parallel jobs are supported.
- K8s cluster node affinity is supported.
- K8s Pod failure policies are supported.

iRODS Testing - Workflow components

- **Initial data staging** - A process that provisions the run with a file system data store and pod initialization scripts into the K8s working environment.
- **iRODS Database** - A database whose type and version is specified by the test requestor.
- **iRODS Provider** - A iRODS provider deployment that is provisioned with various iRODS packages and tests to perform.
- **iRODS Consumer** - A iRODS consumer deployment that is provisioned with various iRODS packages and tests to perform.
- **Test result forensics** - A process that collects test results and processes them for return back to the requestor.
- **Final data staging** - A process that compresses and moves finalized data, updates databases, and removes intermediate data.

iRODS Testing - The Positive effect

Our proof-of-concept of automating the building and testing iRODS using GitHub and Kubernetes services, along with the K8s Job Supervisor, has been achieved.

Automation is everything. GitHub code pushes initiate the building of iRODS packages and can make an iRODS test request. The Job Supervisor creates an iRODS environment on Kubernetes where test suites are executed. The Kubernetes implementation has improved the overall testing time by supporting numerous parallel runs.

The iRODS Development teams will certainly appreciate the reduced time from 'development to test results' and quicker iRODS environment deployments that this infrastructure provides.

iRODS Testing - Future vision

Things on the horizon

- Make this solution publicly available ASAP.
- Incorporate and support additional complex topologies.
- Support for more operating systems and databases.
- Increased level of forensic analysis captured.
- Performance improvements.
- Test suites revisited to support group or individual test selection.
- Address shortcomings that came along with K8s deployments.

Acknowledgements, Questions?

iRODS Testing Team

- Kory Draughn
- Alan King
- Phil Owen
- Terrell Russell

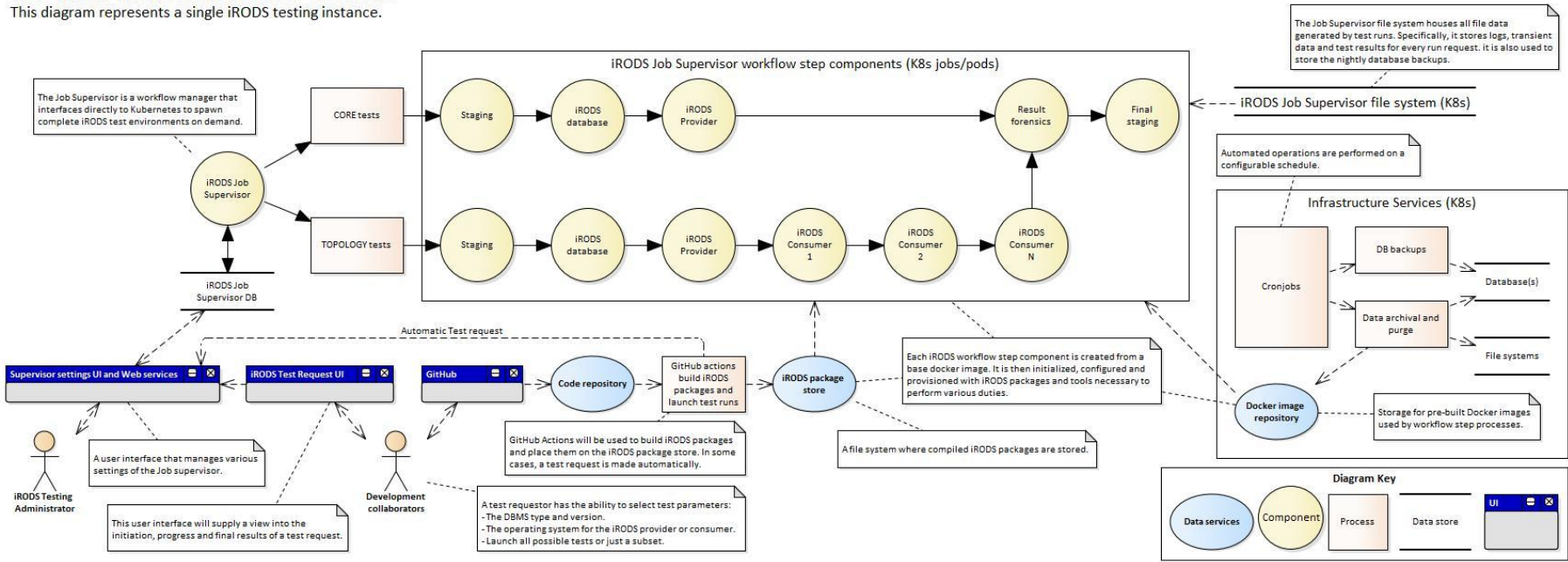


Scan to access this
presentation

Appendix A - Solution data flow diagram

dfd iRODS Testing - Data Flow diagram

iRODS Testing - Data flow diagram, Revised - 02/15/2024
 This diagram represents a single iRODS testing instance.



Appendix B - UI Screenshots

iRODS. Test Request

Create a test run View test run progress

Request name Test environment

Package directory ✓ Test executor type

Operating system Database type

Tests

- test_all_rules.Test_AllRules
- test_all_rules.Test_JSON_microservices
- test_all_rules.Test_msiDataObjRepl_checksum_keywords
- test_auth.Test_Auth
- test_auth.test_iinit
- test_catalog
- test_collection_mtime
- test_configuration_reload
- test_control_plane
- test_delay_queue.Test_Delay_Queue
- test_delay_queue.Test_Execution_Frequency
- test_dynamic_peps
- test_iadmin_set_grid_configuration.test_get_grid_configuration
- test_iadmin_set_grid_configuration.test_set_grid_configuration
- test_iadmin.Test_iadmin

Select All/Clear

Submit request

Debug mode

Submit your request when ready

Version: v0.0.24 Copyright © iRODS Consortium 2024 [About](#)

iRODS. Test Request

Create a test run View test run progress

Test request name

"all-tests" testing complete...

```
{
  "Request name": "all-tests",
  "Testing Jobs": {
    "Total": 23,
    "Complete": 23
  },
  "Jobs": [
    {
      "ID": 32,
      "Status": "New run accepted for all-tests, staging running, staging complete, database running, database configuring, provider running, forensics running, provider configuring, forensics complete, final-staging running, final-staging complete, removing any remaining services, run complete in 111 minutes and 17 seconds",
      "Results": {
        "irods.test.test_all_rules.Test_AllRules-20240410181707": {
          "name": "irods.test.test_all_rules.Test_AllRules-20240410181707",
          "tests": "125",
          "file": "irods/test/test_all_rules.py",
          "time": "4048.057",
          "timestamp": "2024-04-10T19:24:35",
          "failures": "0",
          "errors": "1",
          "skipped": "5",
          "error_details": [

```

Version: v0.0.24 Copyright © iRODS Consortium 2024 [About](#)

Appendix B - UI Screenshots (cont.)

iRODS. Test Request

Create a test run View test run progress

This application is the user interface to the iRODS Test request system.

The iRODS Test request system represents an integrated environment that spans numerous technologies and services:

- GitHub code repositories and actions.
- Docker container image registry.
- Kubernetes cluster.
- Web-based interfaces.
- Supporting application services.

Motivation: Improve iRODS testing performance and ease of use.

Design Goals - Primary

- Speed up test runs by running existing iRODS tests on 'more' machines in parallel.
- iRODS Development team gets 'hands-free' testing automatically or upon request.
- Team visibility of ongoing testing and progress.


Design Goals - Secondary


- GitHub Action integration for per pull-request and support merged test runs.
- External visibility of ongoing testing progress.
- Automated builds in GitHub.

Technology list of the iRODS Test request system:

- DBMS
 - PostgreSQL v15.4
- Development environment
 - PyLint v3.1.0
 - PyTest v8.1.1
 - Python v3.12.2
- Docker
 - Engine v25.0.3
 - API v1.44
- Kubernetes
 - API v28.1.0
 - Helm v3.12.1
 - Client v1.28.7
 - Server v1.27.10
- User interface
 - Node.js v20.11.1
 - Npm v10.5.0
 - React v18.2.0
 - ReactStrap v9.2.2
- Web-services
 - FASTAPI v0.110.0

Version: v0.0.24

 Copyright © iRODS Consortium 2024



Appendix C - Service Screenshots

iRODS-K8s Settings v0.0.26 OAS 3.1

Authorize

default

- GET /get_sv_component_versions Get Sv Component Versions
- GET /get_environment_type_names Get Environment Type Names
- GET /get_test_names Get Test Names
- GET /get_dbms_image_names Get Dbms Image Names
- GET /get_os_image_names Get Os Image Names
- GET /get_test_request_names Get Os Request Names
- GET /get_run_status/ Get Run Status
- GET /get_job_order/{workflow_type_name} Display Job Order
- GET /reset_job_order/{workflow_type_name} Reset Job Order
- GET /get_job_defs Display Job Definitions
- GET /get_log_file_list Get The Log File List
- GET /get_log_file/ Get The Log File
- GET /get_test_result_file Get Test Results File
- PUT /superv_workflow_request/{workflow_type}/run_status/{run_status} Superv Workflow Request
- PUT /run_id/{run_id}/status/{status} Set The Run Status
- PUT /workflow_type_name/{workflow_type_name}/job_type_name/{job_type_name}/next_job_type/{next_job_type_name} Set The Supervisor Job Order

Overview **Pods** Deployments Daemon Sets Stateful Sets Replica Sets Replication Controllers

51 Items Namespace: irods-dev

Name	Co...	Status	Restarts	Age	CPU	Memory
forensics-60-f2zxc		Running	0	2m16s	0.000	12.3MiB
final-staging-61-n85x8		Succeeded	0	2m11s	0.000	0
provider-62-p69tk		Running	0	2m6s	0.425	151.7MiB
forensics-62-42px9		Running	0	2m3s	0.000	12.5MiB
provider-63-bn5q6		Running	0	117s	0.201	204.7MiB
forensics-63-dq2k7		Running	0	113s	0.000	12.6MiB
provider-64-hvpfj		Running	0	106s	0.075	207.4MiB
forensics-64-7nxgj		Running	0	102s	0.000	12.9MiB
provider-65-7hxwx		Running	0	95s	0.087	213.6MiB
forensics-65-nzilt		Running	0	91s	0.000	13.1MiB
provider-66-sgmcp		Running	0	83s	0.240	189.3MiB
forensics-66-4d4dv		Running	0	79s	0.000	13.3MiB
provider-67-5bw9x		Running	0	70s	0.916	165.7MiB
forensics-67-7j7fv		Running	0	66s	0.000	13.4MiB
provider-68-dwf9q		Running	0	56s	1.008	179.6MiB
forensics-68-db7vn		Running	0	52s	0.000	13.6MiB
provider-69-4jpph		Running	0	42s	0.987	154.2MiB
forensics-69-bfmvl		Running	0	37s	0.000	14.7MiB
provider-70-gzrzg		Running	0	27s	0.000	0
forensics-70-76zcx		Running	0	22s	0.000	0
provider-71-4pl5d		Pending	0	11s	0.000	0

Appendix D - Various Links

- **iRODS K8s/Helm deployments:** https://github.com/irods/irods_k8s
- **iRODS Job Supervisor code:** <https://github.com/irods-contrib/iRODS-K8s-Supervisor>
- **iRODS Job Supervisor Settings UI code:** <https://github.com/irods-contrib/iRODS-K8s-Settings>
- **iRODS Data Staging code:** <https://github.com/irods-contrib/iRODS-K8s-Staging>
- **iRODS Test result forensics code:** <https://github.com/irods-contrib/iRODS-K8s-Forensics>
- **iRODS Test Request UI code:** <https://github.com/PhillipsOwen/irods-testrequest-ui>
- **iRODS Data flow diagram:** https://drive.google.com/file/d/1-I9qFQKvz5u_k3LyRSz66wjbJHUid2AH/view
- **iRODS Job supervisor diagram:** https://drive.google.com/file/d/1xQQsOI3iuJFVnul66b-bp4xS_IXYA6hG/view