# Environment

* Ubuntu 14.04.4
* 16GM RAM with 250HDD
* PostgreSQL 9.5
* Java 1.8
* iRODS 4.1.8
* ServiceMix 6.1.0
* Mounted Globus ISILON Drive - /mnt/IRODsTest
* Mounted Cache resource drive
* Assign full access to service account on mounted drives

# Setup

Make service account “ncif-hpcdm-svc” to have following access rights:

* /opt/apache-servicemix-6.1.1 - read, write and execute
* /etc/irods/ - read, write
* Java - Execute
* /var/lib/irods - read, write and execute
* /mnt/IRODsTest - read, write and execute
* /opt/HPC\_Data\_Management - read, write and execute
* /opt/DICE - read, write and execute
* svn – Exceute
* mvn - Execute

Assign read/write permission on “/opt” folder to the user account used for deploying the application

Assign sudo su to “ncif-hpcdm-svc” account for the user account used for deploying the application

Assign sudo su to “irods” account for the user account used for deploying the application

iadmin modzone tempZone name ncifHpcZone

iadmin modresc demoResc name ncifHpcResc

update .irods/irods\_environment.json

update /etc/irods/server-config.json

# Postgresql

CREATE USER ncif\_hpcdm\_db WITH PASSWORD '@)!^';

GRANT ALL PRIVILEGES ON DATABASE "ICAT" to "ncif\_hpcdm\_db";

ALTER ROLE ncif\_hpcdm\_db SUPERUSER;

Run scripts from \hpc-prototype-dev\src\hpc\hpc-server\hpc-dao-impl\src\main\scripts

INSERT INTO "HPC\_USER"("USER\_ID", "FIRST\_NAME", "LAST\_NAME", "IRODS\_USERNAME", "CREATED", "LAST\_UPDATED", "IRODS\_PASSWORD", "DOC") VALUES ('konkapv', 'Prasad', 'Konka', 'konkapv', '07-05-2016', '07-05-2016', null, 'FNLCR');

# Register system account

Register service accounts for Cleversafe, iRODS, Globus

Assign permission to /FNL\_SF\_Archive to admin user

# Update properties

hpc.ws.rs.host=fr-s-hpcdm-uat-p.ncifcrf.gov

hpc.service.dataTransfer.downloadDirectory=/mnt/IRODsScratch/data/S3

# iRODS.

hpc.integration.irods.host=fr-s-hpcdm-uat-p.ncifcrf.gov

hpc.integration.irods.port=1247

hpc.integration.irods.zone=ncifHpcZone

hpc.integration.irods.resource=ncifHpcResc

hpc.integration.irods.basePath=/ncifHpcZone/home/FNL\_SF\_Archive

hpc.integration.globus.archive.endpoint=nihfnlcr#gridftp1

hpc.integration.globus.archive.path=/mnt/IRODsTest/FNL\_SF\_Archive/ServiceFolderFor2Hop

hpc.integration.globus.archive.directory=/mnt/IRODsTest/FNL\_SF\_Archive/ServiceFolderFor2Hop

hpc.integration.globus.archive.type=TEMPORARY\_ARCHIVE

hpc.integration.globus.download.endpoint=nihfnlcr#gridftp1

hpc.integration.globus.download.path=/mnt/IRODsTest/FNL\_SF\_Archive/ServiceFolderFor2Hop

hpc.integration.globus.download.directory=/mnt/IRODsTest/FNL\_SF\_Archive/ServiceFolderFor2Hop

# Cleversafe.

hpc.integration.cleversafe.URL=http://fr-s-clvrsf-01.ncifcrf.gov

hpc.integration.cleversafe.archive.vault=DSE-TestVault2

hpc.integration.cleversafe.archive.objectId=FNL\_SF\_Archive

hpc.integration.cleversafe.archive.type=ARCHIVE

hpc.dao.postgresql.dbName=ICAT

hpc.dao.postgresql.username=ncif\_hpcdm\_db

hpc.dao.postgresql.password=hpc

# INSTALLING iRODS

<https://docs.irods.org/master/manual/installation/>

Before installing iRODS, we first need to satisfy assumptions about hostnames that iRODS

relies on, and then install and configure a database.

**Hostnames**

iRODS networking is built on top of hostnames. A hostname is a label that identifies a device in a computer network. The hostname of a computer can be determined many different ways, including:

• the command-line program hostname

• the C function gethostname()

• the Python function socket.gethostname()

**FYI:** To learn more about hostnames, domain names, and IP addresses, visit MIT’s “IP Addresses, Host Names and Domain Names” page: https://ist.mit.edu/network/ip

iRODS makes three assumptions about all of the servers in a zone (both the iCAT server and any resource servers):

• Each server has a unique hostname.

• Each server is able to resolve the hostname of all other servers (i.e., find the IP address of a server, given its hostname).

• Each server is able to communicate with all other servers using the resolved IP

addresses.

Therefore, before installing iRODS, we must make sure these assumptions are satisfied.

***Setting the Hostname***

The iRODS zone we will be creating and using will consist of only an iCAT server—no resource servers. Therefore, to satisfy the preceding networking assumptions we only need to set an appropriate hostname on the iCAT server and make sure that the iCAT server knows its own hostname. The hostname we will be using for the iCAT server is learner-vb.example.org. This hostname is already set. To verify this, execute:

$ hostname

The command should print learner-vb.example.org to the terminal.

If the hostname were not set, you could set the hostname by executing:

$ sudo hostname learner-vb.example.org

To make the hostname change permanent across computer restarts, we need to edit the contents of the file /etc/hostname so that the file contains only learner- vb.example.org. We will use the editor *nano*. To edit /etc/hostname, execute:

$ sudo nano /etc/hostname

Then delete the current contents, enter the new hostname, and save and close the file.

***Resolving (or Mapping) the Hostname to an IP Address***

Computers in production network environments will be able to rely on an existing DNS (Domain Name System) to resolve the hostnames of all the iRODS servers. Our test setup does not have a DNS; however the iRODS team has already configured the training VM with its hostname.

Should you wish to do this yourself in a later installation, you would need to edit the file

/etc/hosts. Execute:

$ sudo nano /etc/hosts

Each line in /etc/hosts consists of a leading IP address followed by a list of white-space- separated hostnames that we want to resolve to that IP address. Find the line that starts with

127.0.0.1. It will look something like the following:

127.0.0.1 localhost

This IP address corresponds to a special loopback device that lets the computer send messages to itself. Add the desired hostname learner-vb.example.org to the front of the list of hostnames, which should resolve to 127.0.0.1.

The updated line should look something like this:

127.0.0.1 learner-vb.example.org localhost

After making this change, save and close the file.

To check that we can now resolve the hostname, execute:

$ ping -c 3 learner-vb.example.org

The program should print something similar to the message below. The ping rates will differ.

PING learner-vb.example.org (127.0.0.1) 56(84) bytes of data.

64 bytes from learner-vb.example.org (127.0.0.1): icmp\_seq=1 ttl=64 time=0.027 ms

64 bytes from learner-vb.example.org (127.0.0.1): icmp\_seq=2 ttl=64 time=0.037 ms

64 bytes from learner-vb.example.org (127.0.0.1): icmp\_seq=3 ttl=64 time=0.038 ms

—- learner-vb.example.org ping statistics —-

3 packets transmitted, 3 received, 0% packet loss, time 1998ms rtt min/avg/max/mdev = 0.027/0.034/0.038/0.005 ms

If instead the output is

ping: unknown host learner-vb.example.org

the /etc/hosts file has not been configured correctly. Review the edits to /etc/hosts to identify any errors that might have been made.

**Ports**

iRODS servers use a number of ports for network communication. By default, these are:

• 1247 and 1248 for normal operation

• 20000 - 20199 for transmitting large files

***Note!*** The default Ubuntu 14 installation does not have a firewall, so iRODS will be able to use these ports without any additional action.

**The iCAT Database**

iRODS stores most of its information (e.g. user names, file names and locations, metadata) in the iCAT. iRODS assumes this database is created and managed by a third party. Therefore, before installing iRODS, we have to create and configure the database iRODS will be using.

For this training, we will be using PostgreSQL for our iRODS database. First let’s update

Ubuntu’s apt repository.

$ sudo apt-get update

Then let’s install the PostgreSQL server software.

$ sudo apt-get install postgresql

Next, we will switch user to the Linux user account—postgres—that controls the PostgreSQL

server software so that we can create the iCAT database:

$ sudo su - postgres

Start the PostgreSQL command console:

$ psql

Now we are in PostgreSQL, so we will switch to database query language.

***Note!*** Because we are now using database query language, be sure to use semi-colons (;)

to end statements.

Let’s create the database to be used by iRODS:

> CREATE DATABASE "ICAT";

Create the PostgreSQL user account to be used by iRODS:

> CREATE USER irods WITH PASSWORD 'testpassword';

Give the iRODS PostgreSQL user account permission to use the database:

> GRANT ALL PRIVILEGES ON DATABASE "ICAT" to irods;

Log out of the PostgreSQL command console:

> \q

Log out of the Linux user account—postgres—that controls the PostgreSQL server software:

$ exit

You are now once again learner.

**Installing iRODS Software Packages**

iRODS is split into two packages:

• the core server software

• the database plugin specific to the type of database used (PostgreSQL in our case)

To download the core server software execute:

$ wget ftp://ftp.renci.org/pub/irods/releases/4.1.8/ubuntu14/irods-icat-4.1.8-ubuntu14-x86\_64.deb

To download the PostgreSQL database plugin, execute:

$ wget ftp://ftp.renci.org/pub/irods/releases/4.1.8/ubuntu14/irods-database-plugin-postgres-1.8-ubuntu14-x86\_64.deb

After doing so, there should be three new files in your current directory:

• irods-icat-4.1.8-ubuntu14-x86\_64.deb

• irods-database-plugin-postgres-1.8-ubuntu14-x86\_64.deb

Install the downloaded packages by executing:

$ sudo dpkg -i irods-icat-4.1.8-ubuntu14-x86\_64.deb irods-database-plugin-postgres-1.8-ubuntu14-x86\_64.deb

The install command will warn you about missing package dependencies with a message similar to:

dpkg: error processing package irods-database-plugin-postgres (—install):

dependency problems - leaving unconfigured Processing triggers for man-db (2.6.7.1-1) ... Processing triggers for ureadahead (0.100.0-16) ... Processing triggers for libc-bin (2.19-0ubuntu6) ... Errors were encountered while processing:

irods-icat

irods-database-plugin-postgres

Finish the installation of the iRODS packages by installing the required dependencies:

$ sudo apt-get -f install

Press Enter when the installer asks if you would like to continue.

Then you will be presented with, among other things, two messages to the screen:

======================================================================= Welcome to iRODS.

This installation of an iCAT server is currently incomplete and needs a database plugin to be installed and configured before

it can be started and used.

and

Please consult the manual for further instructions.

=======================================================================

=======================================================================

iRODS Postgres Database Plugin installation was successful.

To configure this plugin, the following prerequisites need to be met:

- an existing database user (to be used by the iRODS server)

- an existing database (to be used as the iCAT catalog)

- permissions for existing user on existing database

Then run the following setup script:

sudo /var/lib/irods/packaging/setup\_irods.sh

=======================================================================

The final installation step is running the setup script:

$ sudo /var/lib/irods/packaging/setup\_irods.sh

The setup script will prompt for a number of pieces of information. Some prompts provide a default value. The default value will be at the end of the prompt in square brackets.

***Note!*** To select the default value, press the Enter key without typing any information. For this installation, we will use the default value for each prompt that provides one. (See **Appendix C: Installation Prompts** for a worksheet so you can plan your responses to the

prompts in future installations.)

iRODS service account name [irods]:

The Linux account that will run the iRODS server software. The account will be created if it does not already exist.

iRODS service group name [irods]:

The primary group of the Linux account that will run the iRODS server software.

iRODS server's zone [tempZone]:

The name of the iRODS zone.

iRODS server's port [1247]:

The main iRODS port.

iRODS port range (begin) [20000]:

The beginning of the port range used when transferring large files.

iRODS port range (end) [20199]:

The end of the port range used when transferring large files.

iRODS Vault directory [/var/lib/irods/iRODS/Vault]:

The ***Vault*** (i.e., storage) location of the default unixfilesystem resource created during installation.

iRODS server's zone\_key [TEMPORARY\_zone\_key]:

A secret key used in server-to-server communication.

iRODS server's negotiation\_key [TEMPORARY\_32byte\_negotiation\_key]:

A secret key used in server-to-server communication.

Control Plane port [1248]:

The port used for the control plane. The control plane receives status updates from all servers, and issues commands to servers to pause, resume, shut down, etc.

Control Plane key [TEMPORARY\_\_32byte\_ctrl\_plane\_key]:

A secret key shared by all servers.

Schema Validation Base URI (or 'off') [https://schemas.irods.org/configuration]:

The location of the schema files used to validate the server's configuration files.

iRODS server's administrator username [rods]:

The name of the iRODS administration account that will be created during setup.

iRODS server's administrator password:

There is no default value for the iRODS administration account password. For the purposes of this class, use rods. In the future, however, you will want to use more complex passwords.

Please confirm these settings [yes]:

Review the summary of your chosen settings. If you need to change them, type no to go through the prompts again. Otherwise, press Enter to accept the settings and continue.

Database server's hostname or IP address:

There is no default value. Enter localhost.

Database server's port [5432]:

The database server listens for notifications from other applications on this port. The default value, 5432, is correct for default PostgreSQL installations.

Database name [ICAT]:

This is the name of the database that we created in PostgreSQL during the iCAT

database installation.

Database username [irods]:

Enter irods. This must match the irods Linux account name to authenticate into

Postgres without changing Postgres settings.

Database password:

There is no default value. Enter testpassword. This is the same password we set for during the iCAT database installation.

Please confirm these settings [yes]:

Review the summary of your chosen settings. If you need to change them, type no to go through the prompts again. Otherwise, press Enter to accept the settings and continue.

Once the script has received all of its input, it will complete the setup. A successful setup will end with the following text:

---------------

Running update\_catalog\_schema.py... Updating to Catalog Schema... 2

Updating to Catalog Schema... 3

Updating to Catalog Schema... 4

Done.

**Installation Checklist**

For future installations, this checklist may be helpful.

|  |  |  |
| --- | --- | --- |
| Before Installation | | |
|  | Hostnames | |
|  | Set and confirm hostname to learner-vb.example.org. |
|  | Set post-restart hostname to learner-vb.example.org. |
|  | Add hostname to /etc/hosts. |
| iCAT Database | |
|  | Install PostgreSQL database server software. |
|  | Create ICAT database. |
|  | Create irods PostgreSQL user. |
|  | Grant irods PostgreSQL user permissions on ICAT database. |
| iRODS Software Installation | | |
|  |  | Download iRODS packages. |
|  | Install iRODS packages. |
|  | Install missing iRODS dependencies. |
|  | Run iRODS setup script. |

# Remove an existing iRODS installation

<https://groups.google.com/forum/#!topic/irod-chat/AEoiYvAWoS8>

You can "sudo rm -rf" to delete "/var/lib/irods", "/tmp/irods", and "/etc/irods". Then

sudo userdel irods

"sudo dpkg --purge --force-all irods-icat irods-database-plugin-postgres".

apt-get remove irods-icat

sudo apt-get purge irods-icat

For 4.1.x versions

sudo mkdir /var/lib/irods/packaging

sudo touch /var/lib/irods/packaging/postinstall.sh /var/lib/irods/packaging/preremove.sh

sudo chmod +x /var/lib/irods/packaging/postinstall.sh /var/lib/irods/packaging/preremove.sh

sudo dpkg -P irods-resource

sudo dpkg -i irods-resource-4.1.0-ubuntu14-x86\_64.deb

# Setting up iRODS Resource Server

Resource Server

wget <ftp://ftp.renci.org/pub/irods/releases/4.1.8/ubuntu14/irods-resource-4.1.8-ubuntu14-x86_64.deb>

sudo dpkg -i irods-resource-4.1.8-ubuntu14-x86\_64.deb

sudo apt-get -f install

sudo /var/lib/irods/packaging/setup\_irods.sh

The setup\_irods.sh script will ask for the following fifteen pieces of information about the existing Zone that the iRODS resource server will need in order to stand up and then connect to its configured iCAT Zone:

1. Service Account Name
2. Service Account Group
3. iCAT Port
4. Parallel Port Range (Begin)
5. Parallel Port Range (End)
6. Vault Directory
7. zone\_key
8. negotiation\_key
9. Control Plane Port
10. Control Plane Key
11. Schema Validation Base URI
12. iRODS Administrator Username
13. iCAT Host
14. iCAT Zone
15. iRODS Administrator Password

## Default Environment

Once a server is up and running, the default environment can be shown:

irods@hostname:~/ $ ienv

NOTICE: Release Version = rods4.1.8, API Version = d

NOTICE: irods\_session\_environment\_file - /var/lib/irods/.irods/irods\_environment.json.19345

NOTICE: irods\_user\_name - rods

NOTICE: irods\_host - hostname

NOTICE: xmsg\_host is not defined

NOTICE: irods\_home - /tempZone/home/rods

NOTICE: irods\_cwd - /tempZone/home/rods

NOTICE: irods\_authentication\_scheme is not defined

NOTICE: irods\_port - 1247

NOTICE: xmsg\_port is not defined

NOTICE: irods\_default\_resource - demoResc

NOTICE: irods\_zone\_name - tempZone

NOTICE: irods\_client\_server\_policy - CS\_NEG\_REFUSE

NOTICE: irods\_client\_server\_negotiation - request\_server\_negotiation

NOTICE: irods\_encryption\_key\_size - 32

NOTICE: irods\_encryption\_salt\_size - 8

NOTICE: irods\_encryption\_num\_hash\_rounds - 16

NOTICE: irods\_encryption\_algorithm - AES-256-CBC

NOTICE: irods\_default\_hash\_scheme - SHA256

NOTICE: irods\_match\_hash\_policy - compatible

NOTICE: irods\_gsi\_server\_dn is not defined

NOTICE: irods\_debug is not defined

NOTICE: irods\_log\_level is not defined

NOTICE: irods\_authentication\_file is not defined

NOTICE: irods\_ssl\_ca\_certificate\_path is not defined

NOTICE: irods\_ssl\_ca\_certificate\_file is not defined

NOTICE: irods\_ssl\_verify\_server is not defined

NOTICE: irods\_ssl\_certificate\_chain\_file is not defined

NOTICE: irods\_ssl\_certificate\_key\_file is not defined

NOTICE: irods\_ssl\_dh\_params\_file is not defined

NOTICE: irods\_server\_control\_plane\_key - TEMPORARY\_\_32byte\_ctrl\_plane\_key

NOTICE: irods\_server\_control\_plane\_encryption\_num\_hash\_rounds - 16

NOTICE: irods\_server\_control\_plane\_encryption\_algorithm - AES-256-CBC

NOTICE: irods\_server\_control\_plane\_port - 1248

NOTICE: irods\_maximum\_size\_for\_single\_buffer\_in\_megabytes - 32

NOTICE: irods\_default\_number\_of\_transfer\_threads - 4

NOTICE: irods\_transfer\_buffer\_size\_for\_parallel\_transfer\_in\_megabytes - 4

NOTICE: irods\_plugins\_home is not defined

# iRODS Resource Server with S3 plugin

<https://github.com/irods/irods_resource_plugin_s3>

wget <ftp://ftp.renci.org/pub/irods/plugins/irods_resource_plugin_s3/1.3/irods-resource-plugin-s3-1.3-ubuntu14-x86_64.deb>

sudo dpkg -i irods-resource-plugin-s3-1.3-ubuntu14-x86\_64.deb

sudo apt-get -f install

iadmin mkresc s3CompResc compound

iadmin mkresc cacheResc unixfilesystem ec2-52-87-208-221.compute-1.amazonaws.com:/var/lib/irods/iRODS/Vault

iadmin modresc archiveResc s3 ec2-52-87-208-221.compute-1.amazonaws.com:/hpcdms-irods-s3-bucket1/irods/Vault "S3\_DEFAULT\_HOSTNAME=s3.amazonaws.com;S3\_AUTH\_FILE=/home/ubuntu/s3.key;S3\_RETRY\_COUNT=10;S3\_WAIT\_TIME\_SEC=30;S3\_PROTO=HTTP"

iadmin addchildtoresc s3CompResc cacheResc cache

iadmin addchildtoresc s3CompResc archiveResc archive

# setup SSL & PAM Authentication

<https://docs.irods.org/4.1.7/manual/authentication/>

# CentOS installation

sudo rpm -i irods-resource-4.1.0-centos6-x86\_64.rpm

fuse-libs is needed by irods-resource-4.1.0-0.x86\_64

libfuse.so.2()(64bit) is needed by irods-resource-4.1.0-0.x86\_64

libfuse.so.2(FUSE\_2.6)(64bit) is needed by irods-resource-4.1.0-0.x86\_64

perl(JSON) is needed by irods-resource-4.1.0-0.x86\_64

perl-JSON is needed by irods-resource-4.1.0-0.x86\_64

python-jsonschema is needed by irods-resource-4.1.0-0.x86\_64

python-psutil is needed by irods-resource-4.1.0-0.x86\_64

python-requests is needed by irods-resource-4.1.0-0.x86\_64

sudo /var/lib/irods/packaging/setup\_irods.sh

-------------------------------------------

iRODS Port: 1247

Range (Begin): 20000

Range (End): 20199

Vault Directory: /var/lib/irods/iRODS/Vault

zone\_key: TEMPORARY\_zone\_key

negotiation\_key: TEMPORARY\_32byte\_negotiation\_key

Control Plane Port: 1248

Control Plane Key: TEMPORARY\_\_32byte\_ctrl\_plane\_key

Schema Validation Base URI: https://schemas.irods.org/configuration

Administrator Username: rods