

Merlin Workflow Tools

RabbitMQ and Redis

Sarah Mings
Zeke Morton
Eliana Neurohr

Mentors: Dave Fox, Jason Shortino

August 15, 2019



LLNL-PRES-XXXXXX

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Merlin Team



Sarah Mings

California State University, East Bay



**CAL STATE
EAST BAY**

Zeke Morton

University of California, Davis



Eliana Neurohr

Colorado College



Overview

- What is Merlin?
- Objectives
- What is RabbitMq, Celery, Redis?
- Puppet Manifest
- Docker Containers
- SSL Certificates
- Challenges
- What's Next?

What is Merlin?

- Open source workflow management tool for scientists to submit simulations to the HPCs
 - <https://github.com/LLNL/merlin>
- Our tools
 - Message brokers: RabbitMQ and Redis
 - Task queue: Celery
 - Configuration management tool: Puppet
 - Docker



Merlin

Objectives

- Install and test RabbitMQ, Redis, and Celery
- Puppetize the install of RabbitMQ and Redis
- Dockerize RabbitMQ and Redis
- Add security to RabbitMQ and Redis
 - Passwords and SSL certificates

What is RabbitMQ?

- Message broker that makes distributed systems development easy
- A message broker is to take incoming messages from applications and deliver to other applications

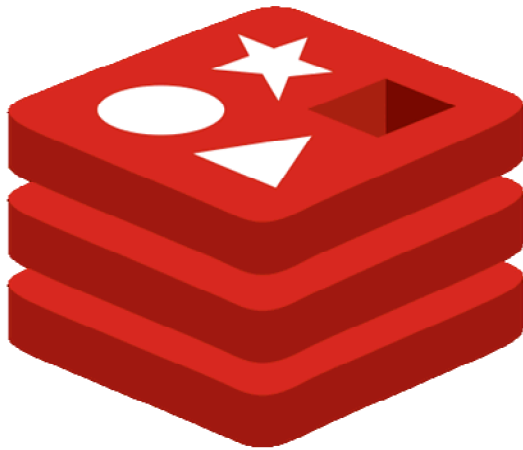
Testing RabbitMQ

```
[morton30@radon3 rabbit]$ python send.py
[x] Sent 'Hello World!'
[morton30@radon3 rabbit]$ python rec.py
[*] Waiting for messages. To exit press CTRL+C
[x] Received 'Hello World!'
```

- Used the Pika Package in a virtual environment and a pip install

What is Redis?

- It's is an in-memory, key-value database, commonly referred to as a data structure server.
- Unlike simplistic key-value data stores that offer limited data structures, Redis has a vast variety of data structures to meet your application needs.

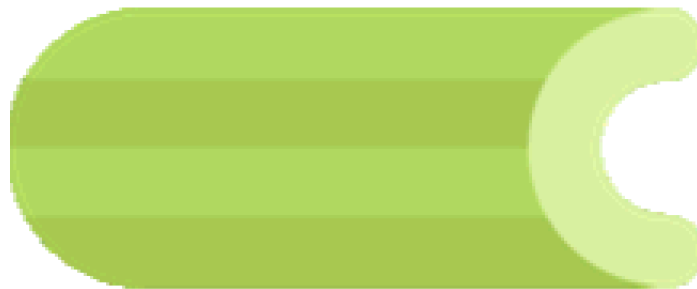


redis

https://en.wikipedia.org/wiki/Redis#/media/File:Redis_Logo.svg

What is Celery?

- It's a task queue with batteries included.
- Task queues let applications perform work, called tasks, asynchronously outside of a user request. If an app needs to execute work in the background, it adds tasks to task queues. The tasks are executed by worker processes.



[https://en.wikipedia.org/wiki/Celery_\(software\)#/media/File:Celery_logo.png](https://en.wikipedia.org/wiki/Celery_(software)#/media/File:Celery_logo.png)

Install Celery & Test Celery

- \$ pip install Celery
- Make task.py

```
from celery import Celery
```

```
BROKER_URL = 'amqp://Rabbit:passw0rd@localhost//Rabbit'
```

```
BACKEND_URL = 'redis://@localhost'
```

```
app = Celery('tasks', broker=BROKER_URL,  
            backend=BACKEND_URL)
```

```
@app.task
```

```
def add(x, y):
```

```
    return x + y
```



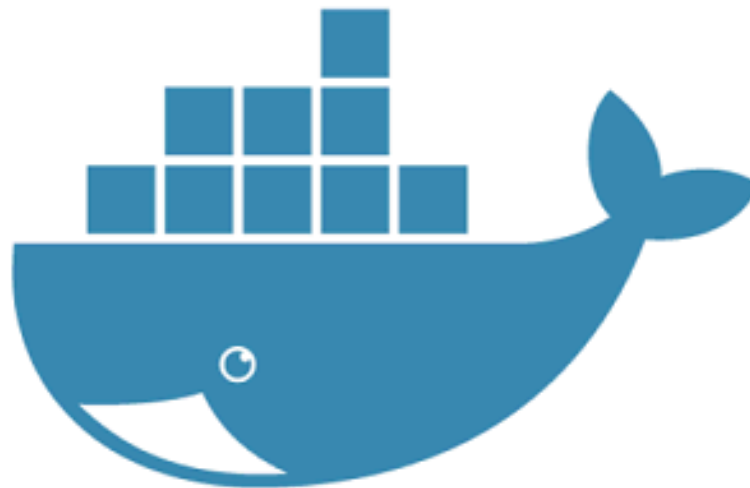
- Configuration management tool
- Best for downloading packages, placing files, and starting and enabling services
- `$ puppet resource <type> <item> >> manifest.pp`
- `$ puppet apply manifest.pp`

Puppet Manifest

```
package { 'Celery':  
  ensure   => 'installed',  
  provider => 'pip',  
}  
  
exec { 'certs':  
  command => "sh ssl.sh",  
  path    => '/sbin:/bin:/usr/sbin:/usr/bin',  
}  
  
service { ['redis', 'rabbitmq-server']:  
  ensure => running,  
  enable => true,  
}  
  
file { ['/etc/rabbitmq/rabbitmq.config':  
  ensure   => 'file',  
  group    => 0,  
  mode     => '0777',  
  owner    => 0,  
  seltype  => 'usr_t',  
  seluser  => 'unconfined_u',  
  source   => '/tmp/rabbitmq.config',  
}]
```

Docker Containers

- Docker container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.



<https://codeburst.io/basics-of-docker-c1416b02d03c>

SSL Certificates Generation

- `tls-gen` is an open source tool originally used for RabbitMQ
- `tls-gen` generates a self-signed Certificate Authority (CA) certificate and two or more pairs of keys: client and server, all with a single command.
- Used basic profile that used a Elliptic Curve Cryptography(ECC) 256bit type
- <https://github.com/michaelklishin/tls-gen>

SSL Certificates RabbitMQ with Docker

- Used self sign certificates in environment variables
- Edit the docker-compose.yml

```
version: '3'

services:
  my-rabbit:
    hostname: my-rabbit
    image: rabbitmq:3
    ports:
      - 5671:5671
    environment:
      - SSL="true"
      - RABBITMQ_SSL_CERTFILE=/tmp/ssl/server_certificate.pem
      - RABBITMQ_SSL_KEYFILE=/tmp/ssl/server_key.pem
      - RABBITMQ_SSL_CACERTFILE=/tmp/ssl/ca_certificate.pem
      - RABBITMQ_DEFAULT_USER=Rabbit
      - RABBITMQ_DEFAULT_PASS=password
      - RABBITMQ_DEFAULT_VHOST=/Rabbit
    volumes:
      - /tmp/ssl:/tmp/ssl
```

Password for Redis

- Set up password in Redis configuration file
- Only can set up ONE password!
- Merlin team found work around by encrypting all data

```
version: '3'

services:
  some-redis:
    image: redis
    command: redis-server --requirepass foobared
    ports:
      - '6379:6379'
```

SSL Certificates RabbitMQ

- Used self sign certificates from RabbitMQ documentation
- Edit /etc/rabbitmq/rabbitmq.config

```
{ssl_listeners, [5671]},
```

```
{ssl_options, [{cacertfile,      "/tmp/ssl/ca_certificate.pem"},  
               {certfile,       "/tmp/ssl/server_certificate.pem"},  
               {keyfile,        "/tmp/ssl/server_key.pem"},  
               {verify,         verify_none},  
               {fail_if_no_peer_cert, false}]}}
```

Challenges

- RabbitMQ Manual Install
- Managing all the software dependencies
- Puppet Manifest
- Add security
- SE Linux

What's Next

- Possible security enhancements for Redis
- Integration and testing it with Merlin
- Testing with other Linux distributions

Special Thanks

Dave
Jason
Thomas
Rigo
Bryan
ean



Thanks!

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.