Cloud Computing Platform for High-level Physics Applications Development

Tong Zhang* and Dylan Maxwell

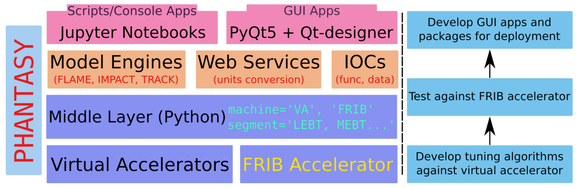
Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI 48824, USA
*E-mail: zhangt@frib.msu.edu

Abstract

To facilitate software development for the high-level applications on the particle accelerator, we proposed and prototyped a web-based computing platform, the so-called 'phyapps-cloud' [1]. Based on the technology stack composed by Python, JavaScript, Docker, and Web service, such a system could greatly decouple the deployment and development, that is the users only need to focus on the feature development by working on the infrastructure that served by 'phyapps-cloud', while the service provider could focus on the development of the infrastructure. In this contribution, the development details will be addressed, as well as the demonstration of developing Python scripts with physics tuning algorithm on this platform.

Background and Intentions

The high-level physics controls applications development at FRIB is based on a software solution based on Python programming language. The framework designed and implemented is named as PHANTASY, which is brief for Physics High-level Applications and Toolkit for Accelerator SYstem [7]. From the interactive scripting environment to GUI application development, PHANTASY solves the problem of building applications with friendly user interface from the physics tuning algorithms. The following figure shows the development workflow.



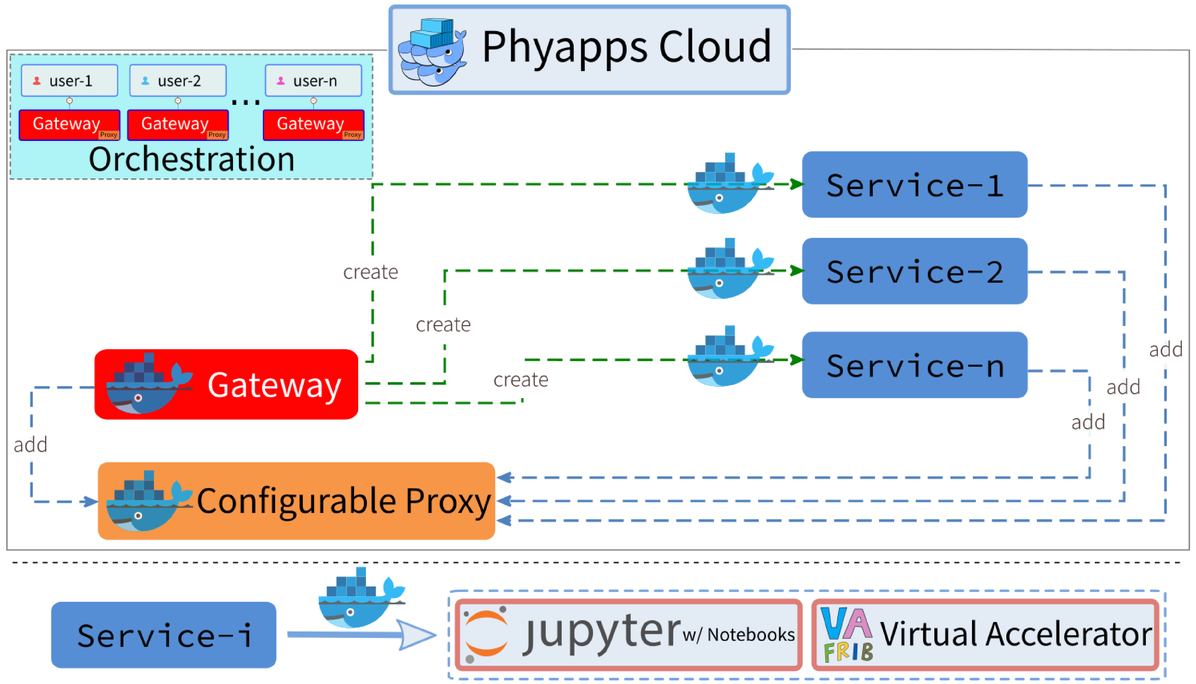
'phyapps-cloud' is the byproduct of PHANTASY project, it provides an alternative way to deploy all the development to a centralized web-based computing platform, the users can work with any web browser to develop the physics application in a private workspace.

The deployment is achieved by Docker Swarm [3], all the developed web applications and virtual accelerator physics application are packed as Docker container images. By incorporating mature development work from the opensource community, many other powerful functionality could be introduced to 'phyapps-cloud' Swarm stack.

The deployment of 'phyapps-cloud' is controlled by a docker-compose YAML file, and another Makefile is built to control all the system-depends environmental variables, e.g. TOKEN, IP, etc., which make it easy to deploy on different servers and even cloud service providers (AWS, Azure, GCP, etc.).

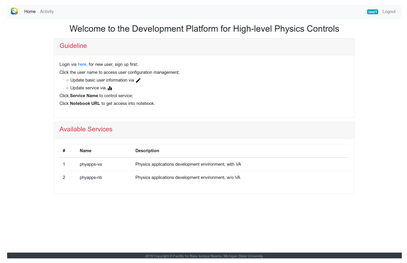
```
Start: $ SRV_IP = 10.20.30.40 make deploy
Stop : $ make stop
```

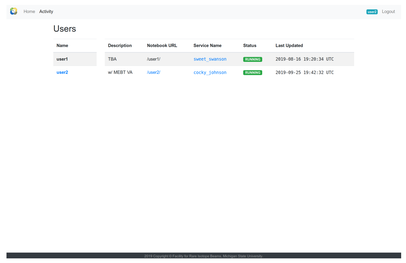
System Architecture

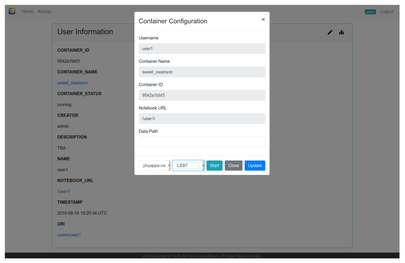


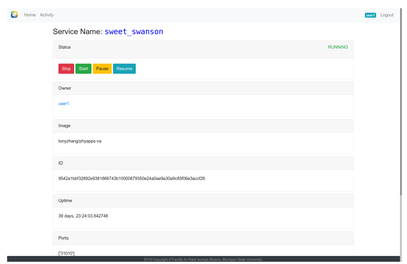
- `Gateway`: REST web service features the main entrance of 'phyapps-cloud', and the center for the users and services management
- `Configurable Proxy`: Web service features configurable proxy, which provides a way to update and manage a proxy table by REST API
- `Service-i`: Private service created by the users, currently, two different services are supported, both features with Jupyter-Notebook service and FRIB physics application development environment, but one also comes with FRIB virtual accelerator service, while the other is not

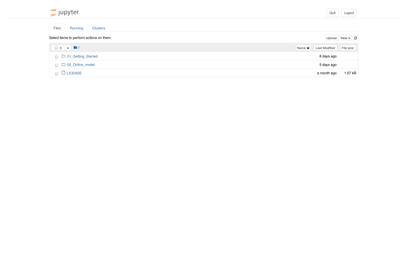
Screenshot Gallery

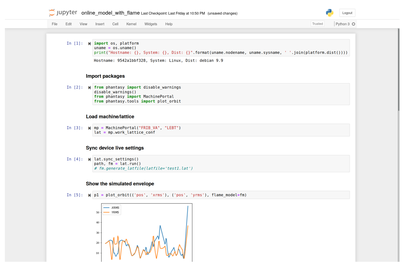












Compose YAML Snippet

```yaml
version: '3.7'
services:
  db:
    image: mysql:8.0.16
    command: --default-authentication-plugin=mysql_native_password
    volumes:
      - db-data:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
      - MYSQL_DATABASE=${DATABASE_NAME}
      - MYSQL_USER=${DATABASE_USER}
      - MYSQL_PASSWORD=${DATABASE_PASS}
    ports:
      - target: 3306
        published: 3307
        protocol: tcp
        mode: host
    deploy:
      restart_policy:
        condition: on-failure
  chp:
    image: "jupyterhub/configurable-http-proxy"
    environment:
      - CONFIGPROXY_AUTH_TOKEN=${TOKEN}
    command: --default-target http://127.0.0.1:5050 --ip ${SRV_IP} --port 8000
    networks:
      hostnet: {}
    deploy:
      restart_policy:
        condition: on-failure
  gateway:
    image: "tonyzhang/phyapps-gateway:latest"
    environment:
      - DB_NAME=${DATABASE_NAME}
      - DB_USER=${DATABASE_USER}
      - DB_PASS=${DATABASE_PASS}
      - PROXY_TOKEN=${TOKEN}
      - PROXY_BASE=http://127.0.0.1:8001/api/routes
      - DPATH=${PWD}/data
    networks:
      hostnet: {}
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    deploy:
      restart_policy:
        condition: on-failure
volumes:
  db-data:
networks:
  hostnet:
    external: true
    name: host
```

References

[1] https://github.com/archman/phyapps-cloud

[2] https://www.docker.com/

[3] https://docs.docker.com/engine/swarm/

[4] https://cloud.docker.com/repository/docker/tonyzhang/phyapps-gateway

[5] https://flask.palletsprojects.com/

[6] https://getbootstrap.com/docs/4.3

[7] T. Zhang *et al.*, "High-level Physics Controls Applications Development for FRIB", in this proceedings, paper TUCPR07.

Acknowledgements

Work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University.

FRIB MICHIGAN STATE UNIVERSITY U.S. DEPARTMENT OF ENERGY Office of Science