👋 Welcome!

Deploying a Service on EDITO Datalab

Learn how to turn your script into a containerized web service and launch it on the EDITO platform.

By **Samuel Fooks**
Flanders Marine Institute (VLIZ)

🎯 What You'll Learn

- ✅ Dockerize a script (R or Python)
- ✅ Push the image to a public Docker registry
- ✅ Configure a Helm chart
- ✅ Deploy the service on the EDITO playground
- ✅ Publish to production via Merge Request

🧱 Application/Service

view_parquet_service

view_parquet.Rmd

The `view_parquet.Rmd` script provides an interactive tool to load, filter, and visualize Parquet datasets. It includes:

- **Interactive Table**: View and filter data using a searchable, paginated table.
- **Map Visualization**: Display geospatial data (e.g., points, polygons) on an interactive map using `leaflet`.
- **Download Filtered Data**: Export selected data as a CSV file.
- **Metadata Schema**: Display the schema of the loaded Parquet dataset.

Is not instructional (Tutorial), and doesn't only perform a specific calculation/run model (Process).

Dockerfile Example

```
FROM rocker/shiny:4.5.0

RUN apt-get update && apt-get install -y \
    libcurl4-openssl-dev \
    libssl-dev \
    libxml2-dev \
    libudunits2-dev \
    libgdal-dev \
    libgeos-dev \
    libproj-dev \
    libfontconfig1-dev \
    libharfbuzz-dev \
    libfribidi-dev \
    libfreetype6-dev \
    libpng-dev \
    libtiff5-dev \
    && rm -rf /var/lib/apt/lists/*

RUN R -e "install.packages(c('shiny', 'arrow', 'leaflet', 'DT', 'dplyr', 'sf', 'leaflet.extras', 'shinythemes'))"

COPY view_parquet.Rmd /srv/shiny-server/view_parquet.Rmd

EXPOSE 3838

CMD ["R", "-e", "rmarkdown::run('/srv/shiny-server/view_parquet.Rmd', shiny_args = list(host = '0.0.0.0', port = 3838))"]
```

Make a container registry token

Working with container registry

You need your container registry token

Build and Push Docker Image

Build and version your container using semantic versioning docs

Not technically required, but if your new version fails, roll back easily.

```
docker build -t ghcr.io/yourusername/view_parquet:1.0.1 .

export CR_PAT = mycontainerregistrytoken

echo $CR_PAT | docker login ghcr.io -u yourusername --password-stdin

docker push ghcr.io/yourusername/view_parquet:1.0.1
```

Test your public image

```
docker run -p 3838:3838 ghcr.io/yourusername/view_parquet:1.0.1
```

Open your browser and navigate to:

```
http://localhost:3838
```

Your working app version is now usable by anyone, anywhere with Docker and an internet connection

📦 Clone the service playground, and add your service

How to add your service, README.md

```
#clone the repo
git clone https://gitlab.mercator-ocean.fr/pub/edito-infra/service-playground.git
cd service-playground

# make your own branch
git checkout -b parquet_viewer_r
git push origin parquet_viewer_r

## Here we use the  terria-map-viewer as a basis for our service
## instead of making from scratch
cp -r terria-map-viewer parquet_viewer_r
```

Helm and Kubernetes Overview

Kubernetes

- **Pods**: Smallest deployable units in Kubernetes, running one or more containers.
- **Cluster**: A group of nodes (machines) managed by Kubernetes.
- **Service**: Exposes your application to the network, enabling communication. (Not to be confused with the predefined datalab deployment services)

Helm

- **Helm Charts**: Pre-configured Kubernetes resources packaged together.
- **Templates**: YAML files with placeholders for dynamic values.
- **Values**: Configuration file (`values.yaml`) to customize deployments.

Basic Helm Template Example

Chart.yaml

```
name: my-service
version: 1.0.0
description: A sample Helm chart
```

values.yaml

```
image:
  repository: my-docker-repo/my-service
  tag: "1.0.0"
service:
  type: ClusterIP
  port: 8080
```

Let's edit our Chart.yaml

Edit `Chart.yaml`:

```yaml
name: view-parquet
description: An interactive Parquet viewer on EDITO
home: https://github.com/yourusername/view_parquet
icon: https://your.icon.url/icon.png
keywords: [shiny, r, parquet, viewer]
version: 1.0.0
appVersion: "1.0.0"
dependencies:
  - name: library-chart
    version: 1.5.16
    repository: https://inseefrlab.github.io/helm-charts-interactive-services
```

🛠️ Let's update values.yaml

`values.yaml`

```
service:
  image:
    version: "ghcr.io/yourusername/view-parquet:1.0.1"

networking:
  service:
    port: 3838
```

The additional values.schema.json

Inputs from the user interface go into here, and this goes with the Helm chart deployment into the cluster.

- app version
- resources

Ex. Let users select different versions of your app

`values.schema.json`

```
"listEnum": [
    "ghcr.io/yourusername/view-parquet:1.0.1",
    "ghcr.io/yourusername/view-parquet:1.0.0"
],
"default": "ghcr.io/yourusername/view-parquet:1.0.1"
```

Update templates/NOTES.txt

Can show the link where the service is deployed, link to sample dataset, etc.

This will be displayed in the pop-up to the user while the service is being deployed.

`templates/NOTES.txt`

```
Your Parquet Viewer in R is being deployed!

It will be available on this [link](http{{ if $.Values.ingress.tls }}s{{ end }}://{{ .Values.ingress.hostname }}).
```

🌐 Enable Ingress (Optional)

In `values.schema.json`, allow user-defined ingress:

```
"x-onyxia": {
  "overwriteDefaultWith": "{{project.id}}-{{k8s.randomSubdomain}}-0.{{k8s.domain}}"
}
// Remove "hidden": true line
```

For more details, refer to the Kubernetes Ingress documentation.

🔒 Add S3 or Marine Service Secrets (Optional)

Add to `values.schema.json`:

```
"s3": {
  "x-onyxia": { "overwriteSchemaWith": "ide/s3.json" }
}
```

Enable secret in templates:

```
envFrom:
- secretRef:
    name: {{ include "library-chart.secretNameS3" . }}
```

For more details, refer to the Kubernetes Secrets documentation.

Commit your changes

First install pre-commit

Run 'make check-format' and it will make sure the formatting is ok

```
make check-format
```

Commit your changes

```
# Stage all changes
git add .
# Commit the changes with a descriptive message
git commit -m "Added my awesome service"
# Push the changes to your branch
git push origin parquet_viewer_r
```

🚀 Launch in Playground

- Check your commit in the [pipelines] (https://gitlab.mercator-ocean.fr/pub/edito-infra/service-playground/-/pipelines)
- If successful, Wait for 5–10 min
- If it fails, check the pipeline logs
- Launch from EDITO Datalab and open the 'link' to your awesome App!

✅ Production Release, out of the playground

Once tested and matured:

- Add yourself to `Chart.yaml` as maintainer
- Submit a Merge Request
- Ping `@pub/edito-infra/codeowners`

🙌 Done!

🎉 Your service is live on EDITO!

🧩 You now know how to go from script → container → Helm → Datalab.

Questions?

📧 edito-infra-dev@mercator-ocean.eu

Docs

- Service Playground README.md
- EDITO docs