Everwell

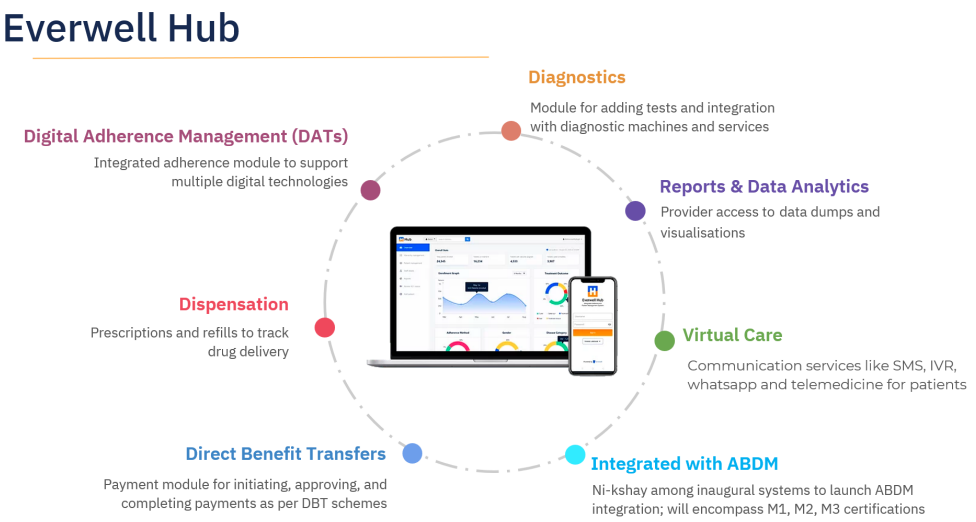
Everwell Hub V2

Documentation of Everwell Hub V2 application

Contents

EVERWELL HUB

The Everwell Hub is a comprehensive, integrated platform for case management. Everwell Hub has supported more than 30 million patients across 17 countries spanning TB, HIV and mental health. Interfaces exist for all stakeholders - program managers, field staff and end beneficiaries. The platform can be used in a ready to use SaaS (Software as a Service) model or can be customized and deployed in a dedicated data center. In India, Everwell Hub is the underlying platform powering Ni-kshay - India's national platform for managing every single Tuberculosis patient.



SALIENT FEATURES

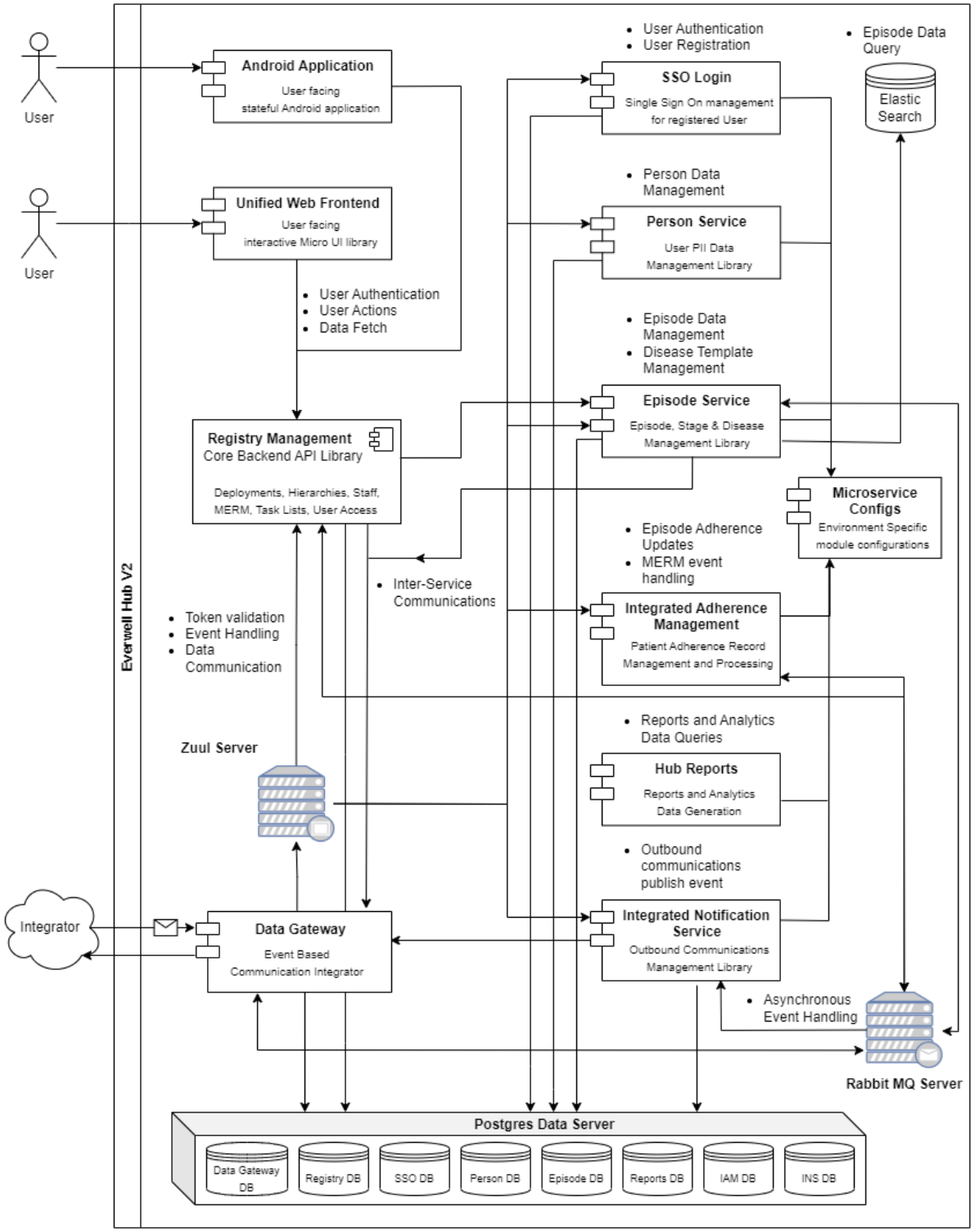
- Enables end to end management of patients, starting from symptomatic to long term post treatment follow up.
- Integrated with a cafeteria of Digital Adherence Technologies.
- Event based integration with third party vendors for integrated data processing.
- Case management functions such as transfer and system assisted deduplication helps in improved tracking, record keeping and more intuitive case management.
- The various automated outputs are generated which are used extensively for program reviews at all program levels.
- Extensive reports and dashboards views based on latest data allow for tracking and assessment of response.
- Automated/manually initiated patient communication via SMS and phone/WhatsApp, from the mobile app.
- Intelligent Task Lists for staff enabling timely follow-up and prioritizing care on a need basis.
- Configurable features allowing for smarter releases and flexible product adoption.

EVERWELL HUB ARCHITECTURE

The Everwell Hub V1 (first generation of Everwell Hub product) was designed on a Monolithic architecture where the Frontend module interacts with a monolithic backend that internally handles all the requested operations for the user.

The Everwell Hub V2 (current generation of Everwell Hub product) has been designed on a Micro UI - Microservices integrated architecture where the individual micro level components of the frontend module interact with individual microservices for specific requested operations for the user. The integration between front end modules and backend microservices is governed by Single Sign On Authentication (SSO). The current architecture provides performance oriented, dynamically configurable, individual as well as integrated functionality supporting easy to maintain modules which greatly enhance the usability and application of the product as a whole.

ARCHITECTURE DIAGRAM



OPEN SOURCE REPOSITORY

https://gitlab.com/everwell/hub-foss

APPLICATION MODULES

DATA GATEWAY

Data Gateway is an integration hub which allows for data exchange with external and internal systems. The Gateway has the capability to have -

- Automated Vendor Registration and Subscription for data events from the FLMP system
- Failure emails to alert on error-prone scenarios.
- Data exchange between in-system micro-services.
- Audit log management of the data requests.
- Archival and management policy for the data requests.

Tech Stack - 

Repository - /data-gateway

UNIFIED WEB FRONTEND

Unified Web Frontend is a modular frontend platform built to provide frontend components to users which can be used for seamless responsive UI rendering. The features of Unified Web Frontend are -

- Responsive UI.
- Modular UI components for smart rendering.
- Dynamic form rendering to support backend driven UI behavior.
- Event tracking support for analytics.
- Localization support.
- Remote Procedure Calls governed lazy loading of data.

Tech Stack - 

Repository - /unified-web-frontend

ANDROID APPLICATION

Everwell Mobile application provides similar functionality to the Unified Web Frontend based website. Available to download via Play Store the application provides the ability to -

- Enroll/ Update a patient.

- Monitor patient adherence.
- Access key statistics for each particular center or region.
- Event tracking support for analytics.
- Localization support.
- Remote Procedure Calls governed lazy loading of data.

Tech Stack - 

Repository - /99dots-android

REGISTRY MANAGEMENT

Registry Management is a digital library of core APIs of Everwell Hub which provides extensive capabilities including but not limited to -

- UI request handling.
- Deployment Management.
- User Management.
- Facility Management.
- Staff Management.
- MERM Management.
- Inter-Mapping of sub modules.
- Integration with in-system services for enabling pluggable functionalities.
- Geospatial and Demographic data recording.
- Support for configurations to control workflows at facility level.
- Support for associating entities with facilities.
- Customizable Outbound events configuration.
- User Permissions to control user accessibility for workflows.
- Role based access control governed Sidebar permissions to manage module accessibility.
- Job Runr support for configuring event or schedule based jobs.

Tech Stack - 

Repository - /registryservice

SSO LOGIN

Single Sign On (SSO) based user management module which serves as the single source of truth for Registered Users and provides the capability of -

- Logged in web user session management.
- Android device user session management.
- Secure SSO authorization mechanism for session management.

Tech Stack - 

Repository - /ssologin

PERSON SERVICE

Person service is responsible for management of Personal Identifiable Information (PII) of registered users. The module provides the capability to -

- Store PII data of the registered user.
- Map multiple email-ids with registered users.
- Map multiple phone numbers with registered users.

Tech Stack - 

Repository - /UserService

EPISODE SERVICE

Episode service is an integral component of Everwell Hub which is responsible for handling all supported episode/patient level operations. The features of this module include -

- Configuration of deployment specific disease support.
- Storage of episode specific data.
- Episode Stage transition and management.
- Rapid retrieval of Episode data through ElasticSearch.
- Episode level feature configuration.
- Rule based configuration for automated data updates.
- Job Runr support for configuring event or schedule based jobs.

Tech Stack - 

Repository - /transition-service

INTEGRATED ADHERENCE MANAGEMENT

Integrated adherence management is the centralized record keeping service designed to handle adherence logging, tracking and mappings. The module provides the capabilities to -

- Map episodes with adherence technologies.
- Digitally update Adherence.
- Manage Adherence schedules.
- Manage Episode Adherence records.
- Configure outbound events for adherence triggers.

Tech Stack - 

Repository - /iam-backend

INTEGRATED NOTIFICATIONS SERVICE

Integrated notifications service is responsible for handling all outbound notifications to users. The module provides the capability to -

- Configure vendor support for provisioning communications.
- Publish-Subscribe messaging support for pushing notifications.
- Configurable event triggers.
- Configurable multi-lingual messaging templates.
- Outbound message logging support for reconciliation.

Tech Stack - 

Repository - /ins

HUB REPORTS

Hub Reports is an integrated analytical module which is built for provisioning customizable reports as per user requirements. The Reports module provides the capability to -

- Setup common report templates.

- Configure dynamic report formats.
- Customize reports data filtering.

Tech Stack -

Repository - /hub-reporting-microservice

TROUBLESHOOTING

The Everwell Hub is deployed using separate CI-CD pipelines for each of its components across different deployment clusters and employs a variety of tools for monitoring application health and runtime statistics.

MONITORING

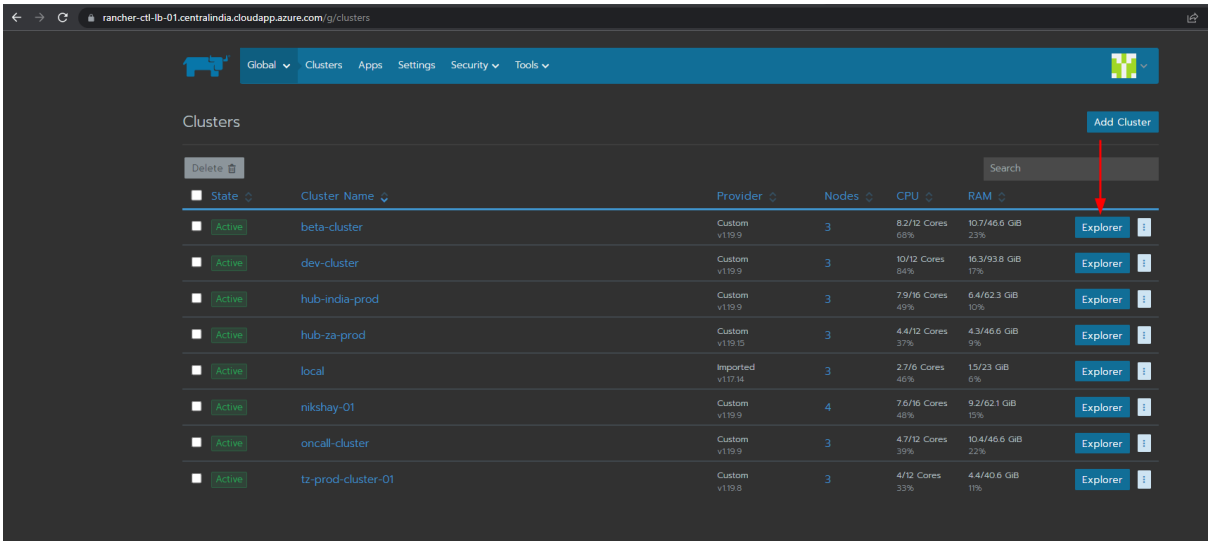
The primary tool for monitoring application and individual component health is Rancher. Rancher provides a consolidated view of the cluster and the individual components deployed in that cluster.

Below mentioned are the steps for application health monitoring using Rancher

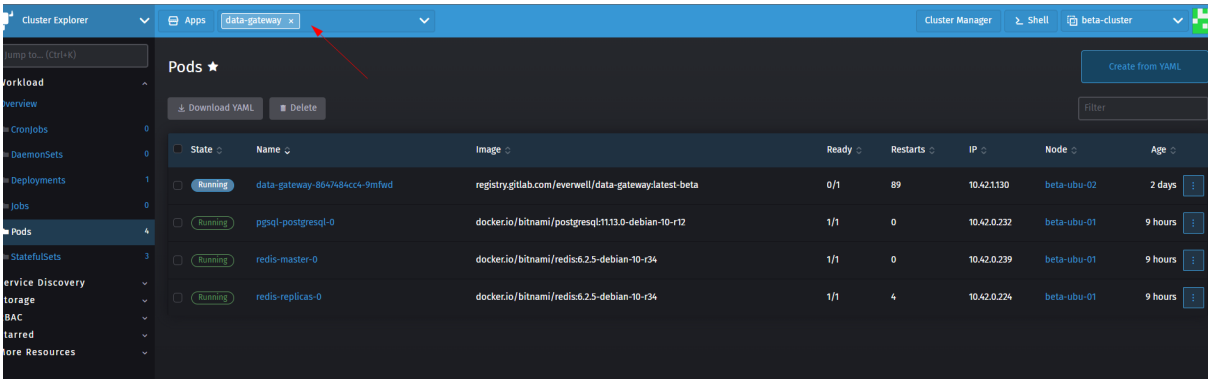
1. Log in to the Rancher server using the below URL.

   https://rancher.everwell.org/dashboard/c/c-m-p96vnqzp/explorer/apps.deployment

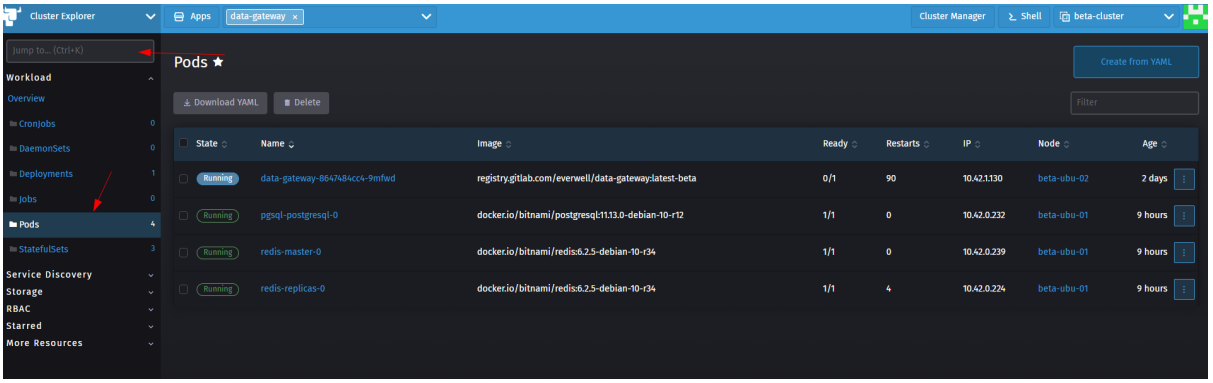
2. Select the cluster where the pod is running



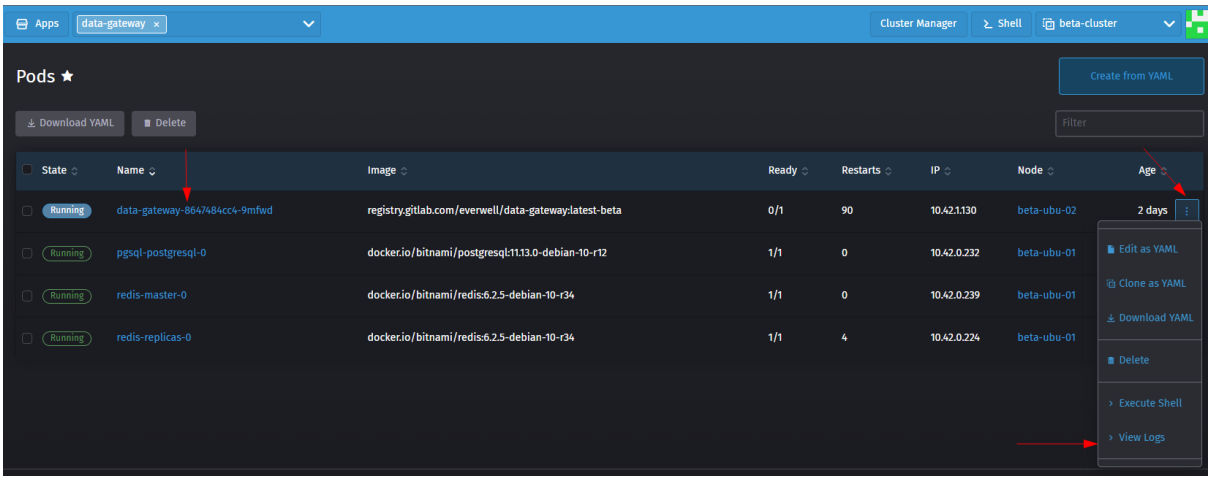
3. Select the namespace where the pod is running.



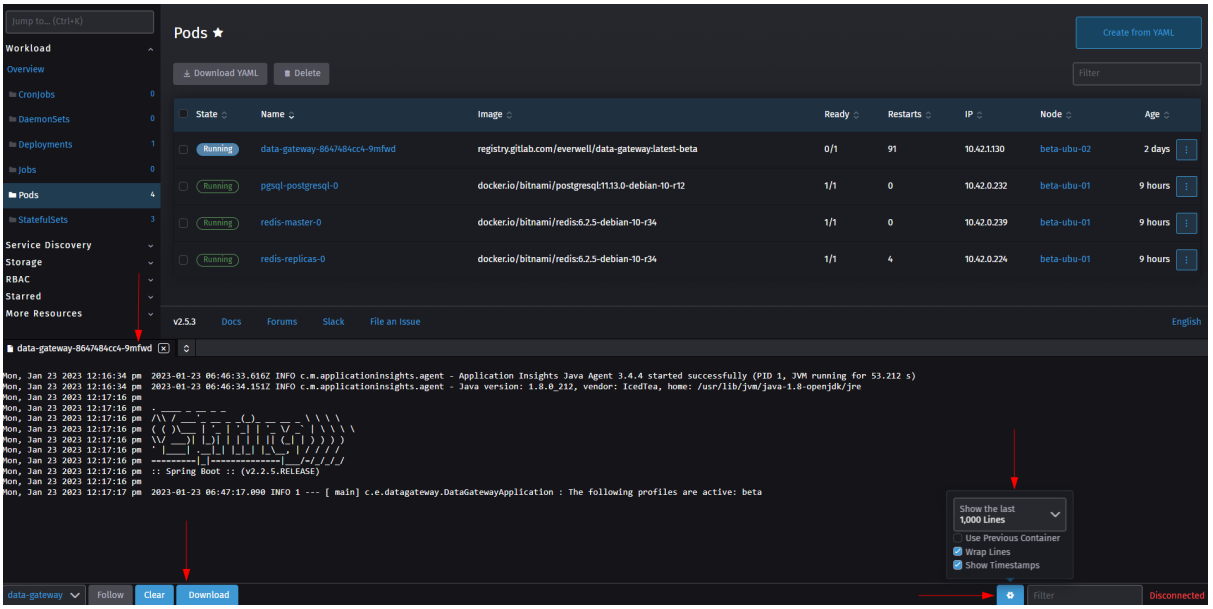
4. Click on the "Workloads" tab from the left-side menu.



5. Click on the 3 dots of the desired pod



6. You can select the number of lines that should be visible

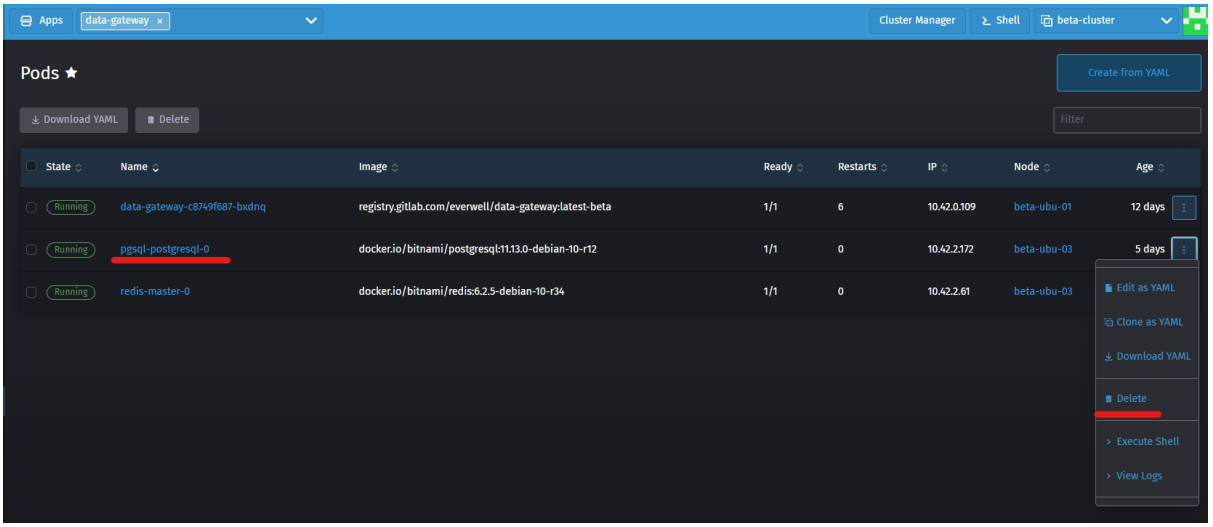


How-To Resolve Read-only file system errors

Check the application pod for the errors, if the Read-only file system error is in postgres pod then navigate to the pod, click on 3 dots and delete. Re-creating the pod fixes the issue

Example:

Caused by: org.postgresql.util.PSQLException: FATAL: could not open file "base/16605/2601": Read-only file system



Repeat the same steps if there are issues in redis or rmq pods

DEBUGGING

Local Setup

The local setup for each of the application components can be done following instructions present in the Readme.md file of each component.

Authentication

The Everwell Hub supports two types of authentication models for authorizing communication between different application components.

1. **OAuth Bearer Token** - The OAuth bearer token is used to authenticate communication between two microservices, the token can be generated by calling the Get Client API of target microservice using appropriate Client signature.
2. **SSO Authorization Cookie** - The SSO Authorization Cookie is used to authenticate communication between User facing components and receiver Microservice. The SSO Authorization Cookie generation and expiration is controlled by the SSO Login upon user login and logout/closure actions.

Documentation

All the microservice components of Everwell Hub application support Swagger/Open-API documentations which can be accessed at the following URL -

<Application Context URL>/swagger-ui.html

Please note that Swagger documentation is not visible for production environments as part of security measures.

The documentations can be referred to for understanding individual API behavior and configurations.

GENERAL INFORMATION

- The environment specific application properties for each of the micro-services is configured in the microservice-configs library which is centrally hosted for Everwell Hub application.
- Redis Server is used for data caching wherever required in the application components.
- Netflix Zuul server is used for facilitating inter-service communications via Data Gateway.
- RabbitMQ server is used for publishing and subscribing asynchronous events in the system.