
Article

[sween](#) · Dec 3, 2021 4m read

[Open Exchange](#)

Invite the FHIR Accelerator Service to your Kubernetes Microservice Party



Want a commercial grade FHIR® Implementation included in your micro service ecosystem and barely have enough time to fill out your Health plan elections?

Here is a quick way to invite the InterSystems® FHIR®Accelerator Service to your Kubernetes Microservice party for immediate use. The solution uses [Nginx](#) proxy ninja moves to get the job done. Though rustic and bound to create some technical arguments, I am pretty happy with the results until this community tells me otherwise, so FHIR® away as they say but it would be great if you heard me out first.

Disclaimer

You will see secrets in manifests, nodePort utilization and a tainted master node to get the point across, parental discretion is advised.

Prerequisites

You'll need a couple of things if you want to deploy this on your own. Its mainly positioned as a starting point or an approach for those wishing to plug FHIR® capability in quickly for development.

Environment

For this demonstration we are parked directly onto a Kubernetes Master Node tainted to put in work.

FHIR up the FHIR® Accelerator Service

- [X] [Create a account in the InterSystems® Cloud](#)
- [X] [Create an InterSystems FHIR® Accelerator Service Deployment](#)

- [] [Create an Api Key](#)

Record the Endpoint and API Key

This is all we need from the FHIR® Accelerator, we will interact with the service in our own way inside the Kubernetes Cluster.

Clone this Repo

For the "Rest of the README", park yourself on a Tainted Kubernetes Master Node.

```
git clone https://gitlab.com/isc_cloud/fhiraas-microservice-kubernetes.git
cd fhir-microservice-kubernetes
```

Deploy to Kubernetes Cluster

- [X] [Create a Namespace](#), this gives us a blast radius to play in for now.
- [X] [Create a ConfigMap](#), this is actually the reverse proxy madness implemented in the nginx.conf.

WE GOTTA EDIT SOMETHING HERE

Remember the key and endpoint we generated from the FHIR Accelerator Service?. We need to update them [here, in the deployment](#)

- [X] [Create a Deployment](#), here is the containers themselves, with a 3 pods for the deployment replica set.
- [X] [Create a Service](#), Expose it!, this is a simple NodePort service that binds a port on the Node to access the FHIR Accelerator Service. It exposes 30036 on the node, and forwards to the deployment pod on 52773.

```
cd fhir-microservice-kubernetes # should already be here, but just to be sure.
kubectl apply -f k8s/
```

Test It!

Lets do a quick test and see if FHIR is successfully getting served up through the NodePort.

- [] Node Port Socket Listening?
- [] Conformance Statement Responding?
- [] Query Patients?
- [] Get Patient?

Scale It!

We're on a single node and exposing the service to a node port so not sure this will increase our throughput, but go nuts and scale away anyway.

```
kubectl scale deployments/isc-fhiraas-deployment --replicas=30 -n isc-fhiraas
```

Throw some FHIR at it!

Included in this repo is a hostile, rustic, shell script for throwing some random patients into the Patient resource with a few bells and no whistles. It needs some environment variables or you can just edit the variables directly in the script. When running this script, ensure proper cpu fan operation and move objects from laptop area to prevent shuffling of objects in the event your laptop takes off.

```
bash bin/fhirbench.sh
```

Results

50 puts, and 50 searches in 13 seconds.

Blame

This stuff in the repo, Ron Sweeney (ron.sweeney@integrationrequired.com) of [Integration Required](#). These are the opinions of my employer.

[#Cloud](#) [#FHIR](#) [#integration-required](#) [#Kubernetes](#) [#InterSystems](#) [IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source

URL: <https://community.intersystems.com/post/invite-fhir-accelerator-service-your-kubernetes-microservice-party>