Article
Ron Sweeney · Sep 1, 2021 5m read

Hands Off HealthShare Deployment Workflow with Gitlab Runners

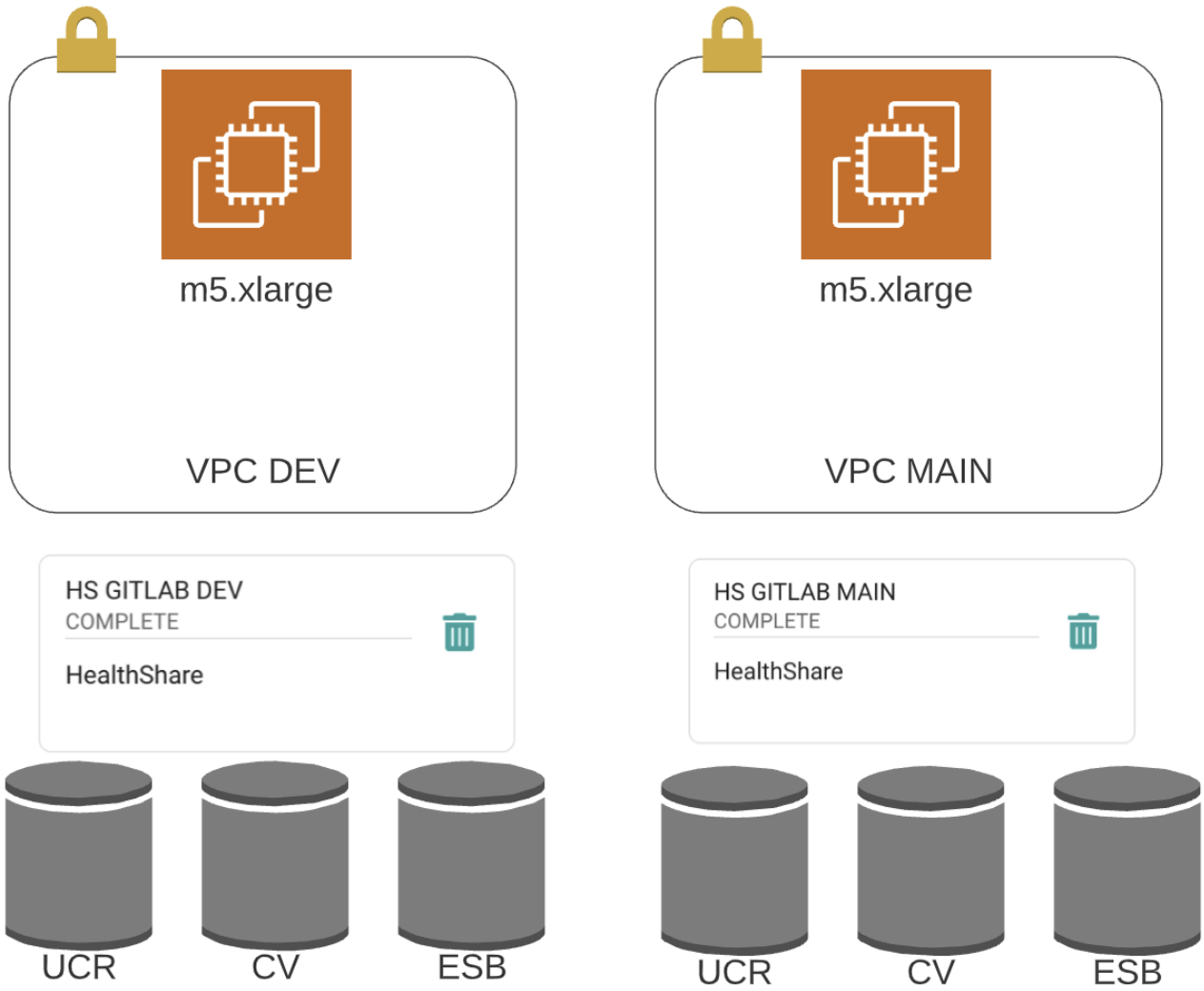
Deploying InterSystems HealthShare code, supporting lookups and artifacts like ssl certs, keys etc is relatively straight forward using Gitlab Runners. Not only does this approach enable managing the code base and deploying with git type workflows, but it also lends to a speedy recovery and repeatable environments for some implementations.

For those of you with HealthShare specific experience, I think we could agree that is an oversimplification of the process with all the moving actors of a HealthShare implementation, but this does provide some insight on how to get things deployed with a simplistic example.

For this example, lets paint the picture of what the minimal deployment actually looks like, then demo the code promotion.

HealthShare

We have, two seperate HealthShare environments/servers in their own demarcation zone (VPC). Each is fairly identical, running 3 instances on each, ESB (HealthConnect), CV (Clinical Viewer) and UCR (Universal Care Record).



For the population of folks that can benefit from showing the running order of the instances on the back end of each of the servers, it looks like this:

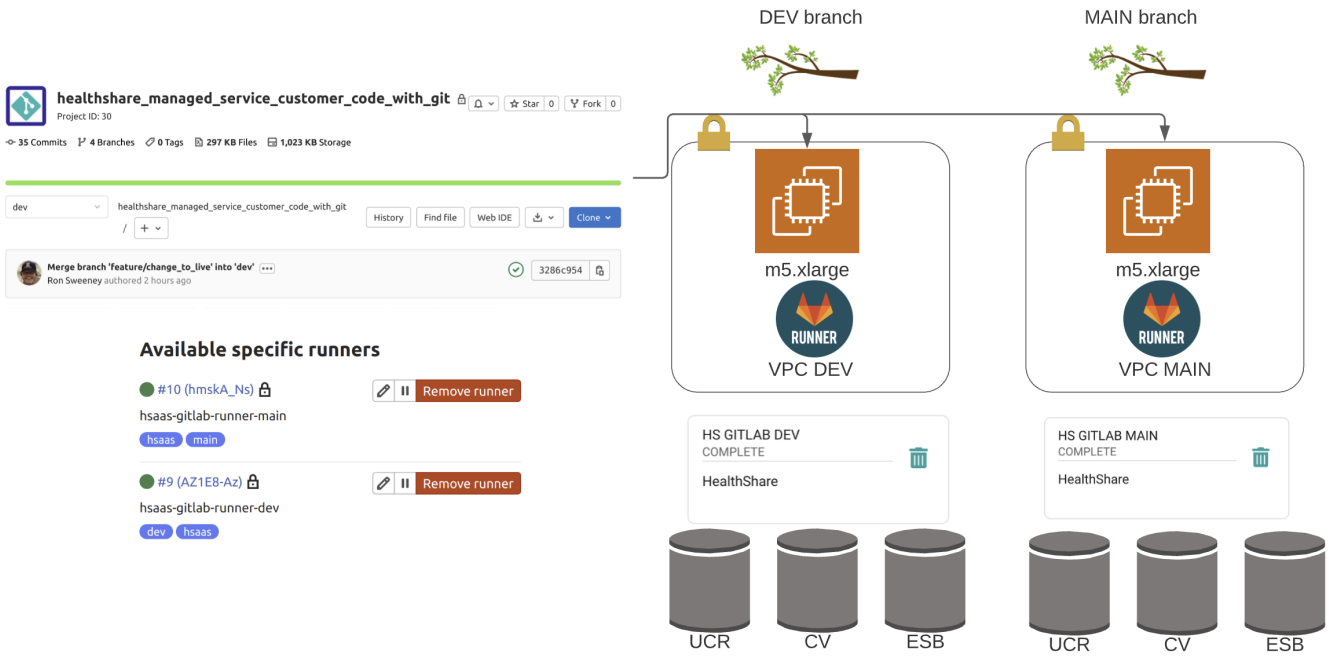
```
ec2-user@ip-172-31-0-100:~
[ec2-user@ip-172-31-0-100 ~]$ iris list

Configuration 'CV'
        directory:    /data/sys/cv
        versionid:    2021.1.0HSCV.1005.0
        datadir:      /data/sys/cv
        conf file:    iris.cpf  (SuperServer port = 1974, WebServer = 57774)
        status:       running, since Tue Jun 22 14:59:11 2021
        state:        ok
        product:      InterSystems HealthShare

Configuration 'ESB'   (default)
        directory:    /data/sys/esb
        versionid:    2021.1.0.215.0
        datadir:      /data/sys/esb
        conf file:    iris.cpf  (SuperServer port = 1972, WebServer = 57772)
        status:       running, since Tue Jun 22 14:59:06 2021
        state:        ok
        product:      InterSystems HealthConnect

Configuration 'UCR'
        directory:    /data/sys/ucr
        versionid:    2021.1.0HS.1009.0
        datadir:      /data/sys/ucr
        conf file:    iris.cpf  (SuperServer port = 1973, WebServer = 57773)
        status:       running, since Tue Jun 22 14:59:20 2021
        state:        ok
        product:      InterSystems HealthShare
[ec2-user@ip-172-31-0-100 ~]$
```

With the Two HealthShare boxes and spinning, we delegate one of them to DEV and one of them to LIVE, and label them as such.

Gitlab

This sets us up to create in Gitlab:

- 1 Gitlab repository
- 2 Branches, dev and main
- 2 Project specific runners, one tagged "dev" and the other tagged "main" and installed on their perspective servers.



The idea here is that we are setting up our runners to execute deployment actions on merges to "dev" where we can iterate on the DEV branch/DEV environment and the after testing is complete, we can merge to "main" in a process, with the ultimate goal of a Hands Off! deployment to Production where humans make mistakes.

Hands Off Live Environment Demo

Pipeline Code

The pipeline code in the demo is available as a gist here. You can see below that the example has two stages, one for dev and one for main, that are exclusively locked to branches. The main point I am trying to drive home with this pipeline is you can pretty much remotely control the implementation of your HealthShare completely through code through the use of Installers, or just by importing and having the ability to run cos and or operating system commands using the gitlab runner.

Spoiler

#Deployment #Git #integration-required #HealthShare

Source URL:https://community.intersystems.com/post/hands-healthshare-deployment-workflow-gitlab-runners