

---

Article

[Dzmitry Rabotkin](#) · Dec 3, 2022 2m read

[Open Exchange](#)

## Guide how to run and use IRIS for Health docker image in GCloud

Hi, I would like to tell you how easy it is to spin up IRIS for Health docker container in compute engine(VPS) in google cloud.

I know that to run IRIS for Health in AWS is pretty simple and straightforward, but I wanted to try if it's the same easy in GCP environment.

Create vm instance. 2GB RAM is more than enough.

Name \*

iris-for-health-docker



Labels ?

+ ADD LABELS

Region \*

us-central1 (Iowa)



Region is permanent

Zone \*

us-central1-a



Zone is permanent

## Machine configuration

Machine family

GENERAL-PURPOSE

COMPUTE-OPTIMIZED

MEMORY-OPTIMIZED

GPU

Machine types for common workloads, optimized for cost and flexibility

Series

E2



CPU platform selection based on availability

Machine type

e2-small (2 vCPU, 2 GB memory)



vCPU

0.5-2 vCPU (1 shared core)

Memory

2 GB

✓ CPU PLATFORM AND GPU


Display device

Enable to use screen capturing and recording tools.

I used Debian 11 as Linux distro.

Standart persistent disk is cheaper.

## Boot disk ?

Name	iris-for-health-docker
Type	New standard persistent disk
Size	10 GB
License type ?	Free
Image	 Debian GNU/Linux 11 (bullseye)

CHANGE

## Identity and API access ?

### Service accounts ?

Service account

Compute Engine default service account



Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. [Learn more](#)

### Access scopes ?

- ☒ Allow default access
- ☐ Allow full access to all Cloud APIs
- ☐ Set access for each API


## Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet


- ☒ Allow HTTP traffic
- ☒ Allow HTTPS traffic

Don ' t forget to allow http, https traffic


Last thing for setting up virtual machine is allow external ports in firewall rules.




VPC network




VPC networks



IP addresses



Bring your own IP



Firewall

Firewall

CREATE FIREWALL POLICY

CREATE FIREWALL RULE

Filter

Enter property name or value

	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs
<input checked="" type="checkbox"/>	<a href="#">default-allow-http</a>	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80, 3000, 5000, 8080, 52773	Allow	1000	<a href="#">default</a>	Off
<input checked="" type="checkbox"/>	<a href="#">default-allow-https</a>	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443, 3000, 52773	Allow	1000	<a href="#">default</a>	Off

I allow here only 52773 which is for web user interface. My Kotlin app will run on same vm and use 1972 internally, so I didn ' t allow it too.

Let ' s Install docker-compose then run IRIS:

```
sudo apt install docker-compose
```

```
sudo docker run --name iris -d --publish 1972:1972 --publish 52773:52773
containers.intersystems.com/intersystems/iris-community:2022.1.0.209.0 --check-caps false
```

I found that command in InterSystems documentation pages here:  
<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=ACLOUD>

I only added 2 things:

1. “ -d ” after name of container “ iris ” which means run in detached mode. So container will still run on background after you close SSH shell.

2. “ - -check-caps false ” at the end. I immediately found it in this article which was very helpful.

<https://community.intersystems.com/post/using-intersystems-iris-containers-docker-201014>

```
@iris-for-health-docker:~$ sudo docker run --name iris -d --publish 1972:1972 --publish 52773:52773 c
ontainers.intersystems.com/intersystems/iris-community:2022.1.0.209.0 --check-caps false
Unable to find image 'containers.intersystems.com/intersystems/iris-community:2022.1.0.209.0' locally
2022.1.0.209.0: Pulling from intersystems/iris-community
08c01a0ec47e: Pull complete
f88269600f2d: Downloading [=====>] 212.5MB/229MB
e2e720c7d229: Download complete
9575f8200941: Downloading [=====>] 72.98MB/437.7MB
7c9053886ee1: Download complete
2bc3734ecca3: Download complete
a9cc57efa4e0: Download complete
```

```
sudo docker ps
```

Container is up and running just with 1 command:

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
4be4d26b01b	containers.intersystems.com/intersystems/iris-community:2022.1.0.209.0	iris	"/tini -- /iris-main..."	8 minutes ago	Up 8 minutes (healthy)	0.0.0.0:1972->1972/tcp, 2188/tcp, 5


Now check UI in browser:

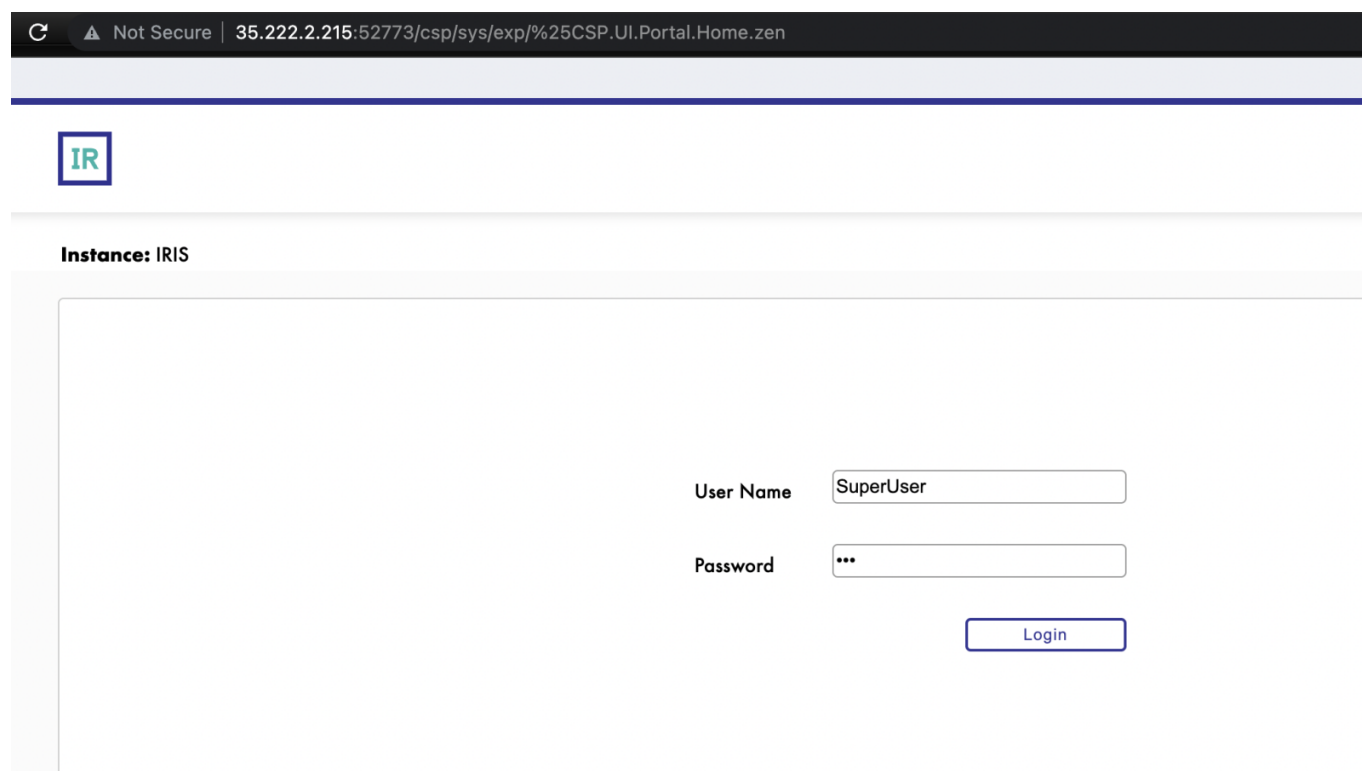
This link I found at Intersystems documentation pages:

<https://docs.intersystems.com/irisforhealthlatest/csp/docbook/DocBook.UI.Page.cls?KEY=GSAUSINGPORTALhttp://35.222.2.215:52773/csp/sys/UtilHome.csp>

35.222.2.215 is external ip of my Virtual machine.

52773 port we exposed with docker run command.

		<a href="#">iris-for-health-docker</a>	us-central1-a	10.128.0.5 ( <a href="#">nic0</a> )	<a href="#">35.222.2.215</a> ( <a href="#">nic0</a> )	SSH ▾	⋮
---	---	--	---------------	-------------------------------------	---	-------	---



Not Secure | 35.222.2.215:52773/csp/sys/exp/%25CSP.UI.Portal.Home.zen

**IR**

**Instance: IRIS**

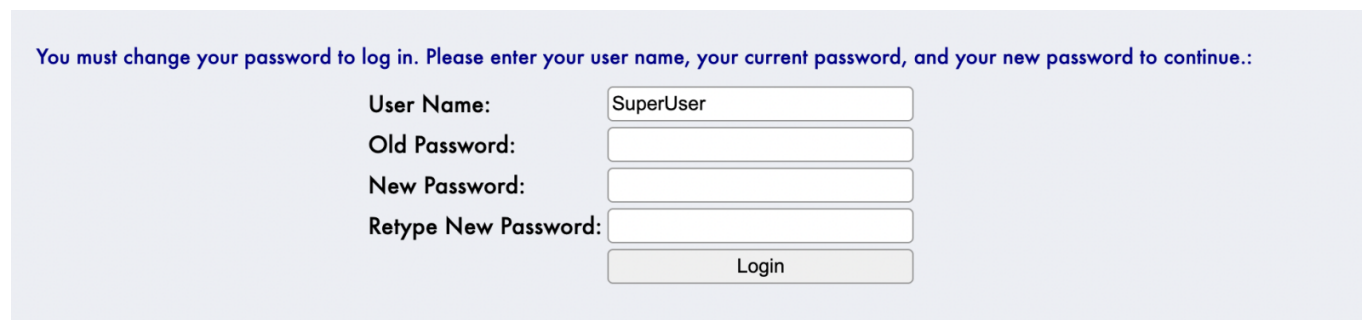
User Name

Password

Default Login: SuperUser

Password: SYS

Now you have to change default password:



You must change your password to log in. Please enter your user name, your current password, and your new password to continue.:

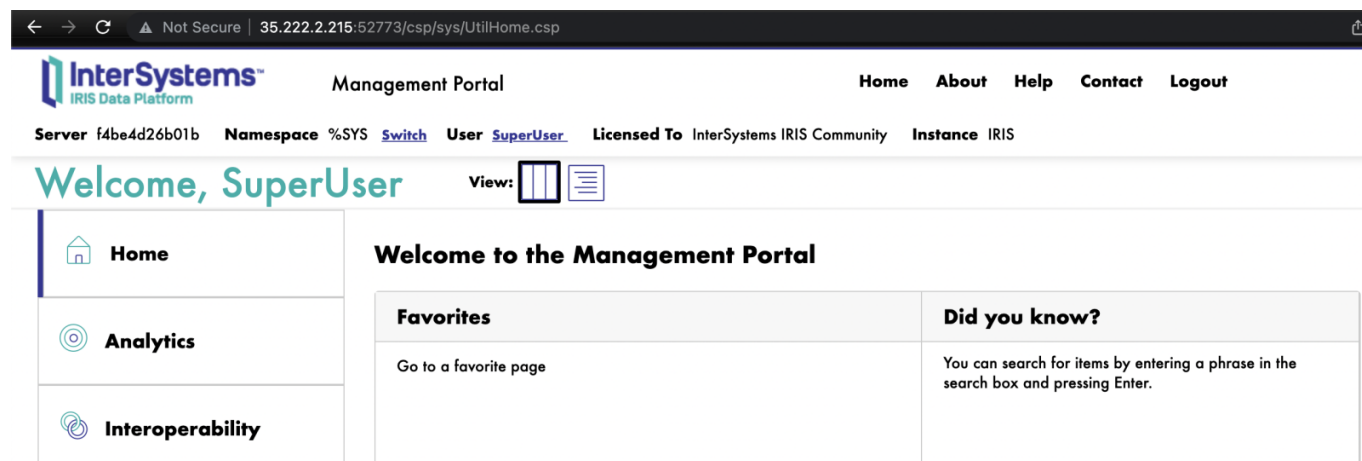
User Name:

Old Password:

New Password:

Retype New Password:

Congratulations, you are logged in:



ATTENTION: don't forget to delete VM instance, so you lose money on hosting.

Conclusion:

In this article I only covered how to run IRIS for Health community edition docker container in google cloud linux environment. Other part of my Kotlin/InterSystems journey will be in description of our project Dia Bro App. Please check it up and vote if you like the idea.

[#Docker #InterSystems IRIS for Health](#)  
[Check the related application on InterSystems Open Exchange](#)

---

Source URL: <https://community.intersystems.com/post/guide-how-run-and-use-iris-health-docker-image-gcloud>