ORACLE

OCI – Function: Quick Start

Setup your OCI environment to start using OCI Function

March 2023, Version 1.0

1 Table of Contents

2 Create a Virtual Cloud Network

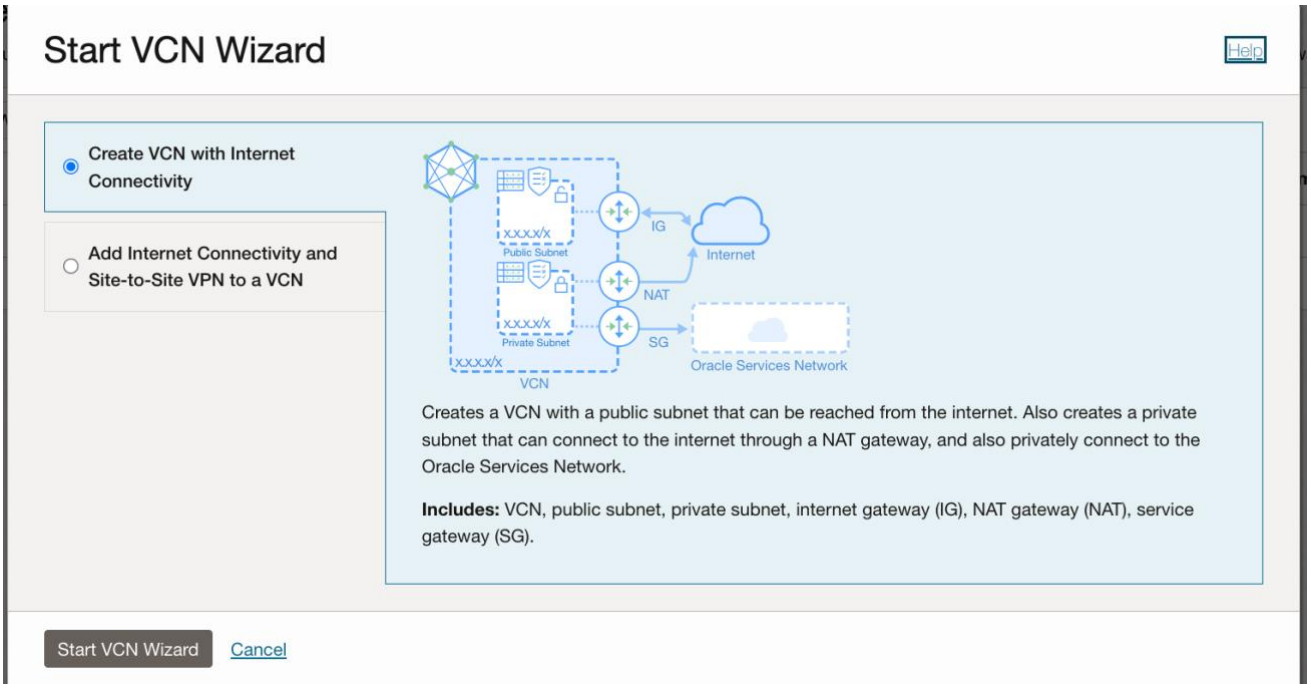
In this scenario I create a dedicated VCN because my function doesn't need to communicate with my other applications.

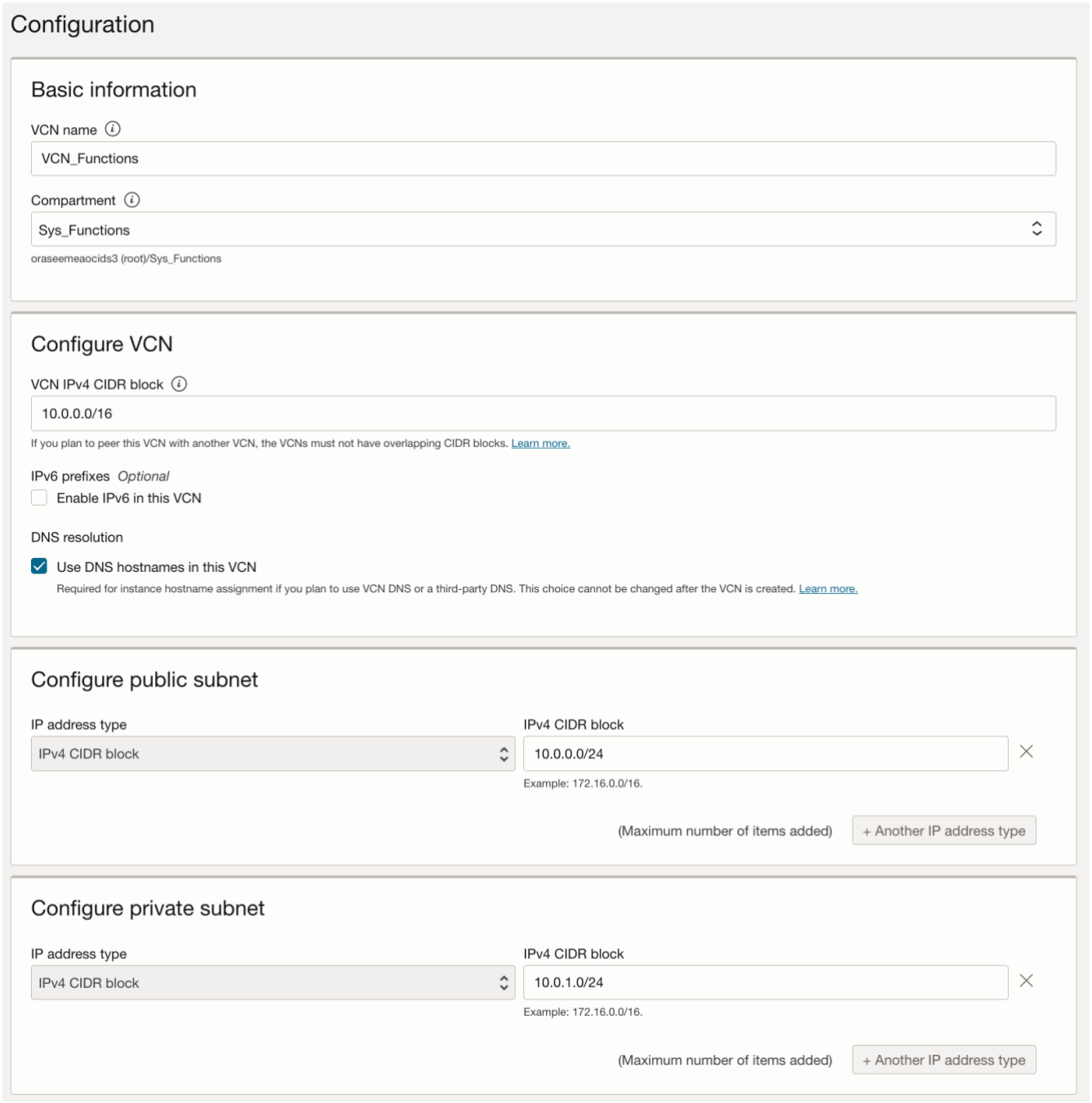
This VCN will host two components:

- An OCI instance (Virtual Machine) that will be used to build my applications
- My OCI Functions

Start VCN Wizard

- Networking > Virtual Cloud Network
- Select your hosting compartment (here Sys_Functions)

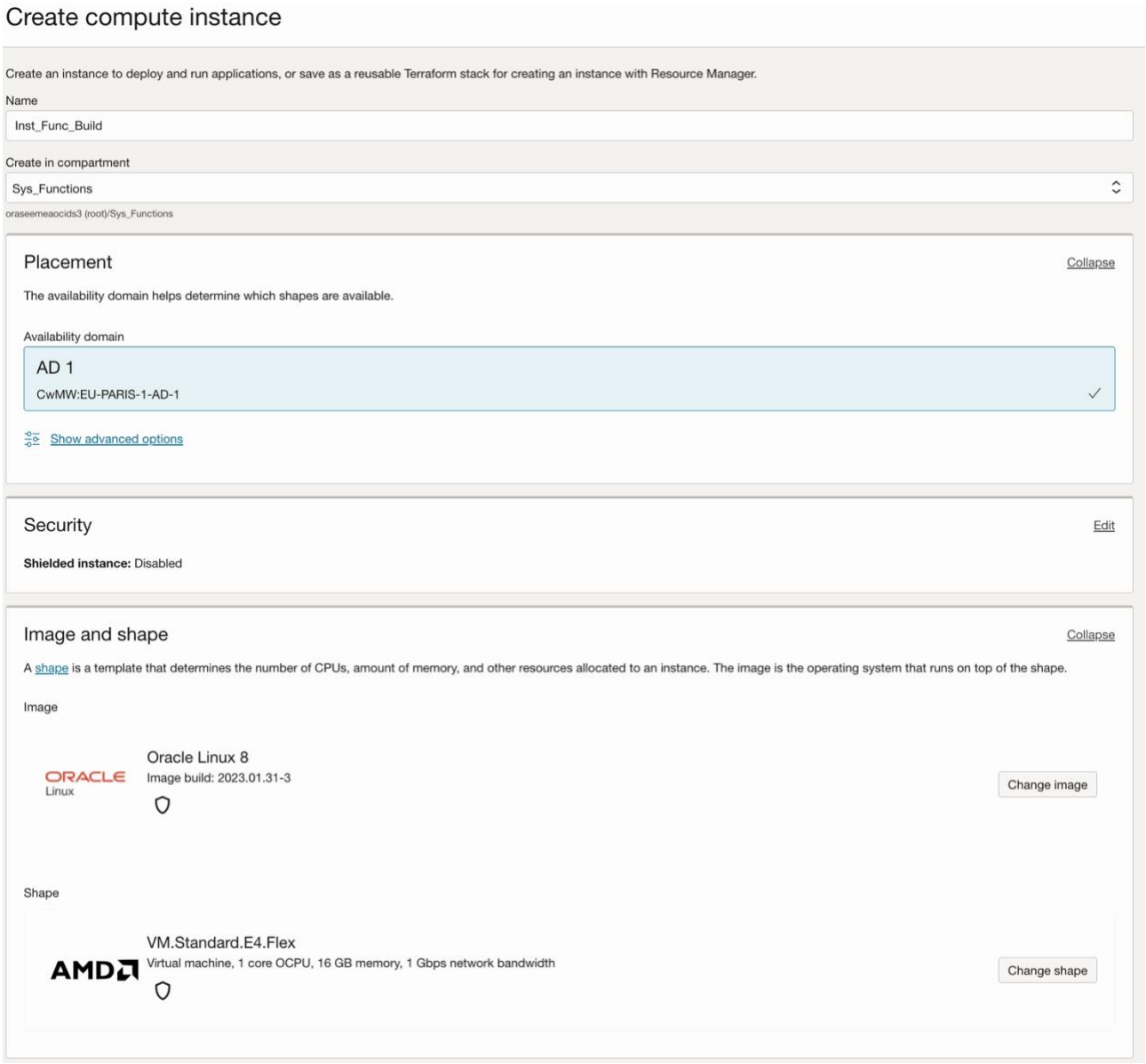


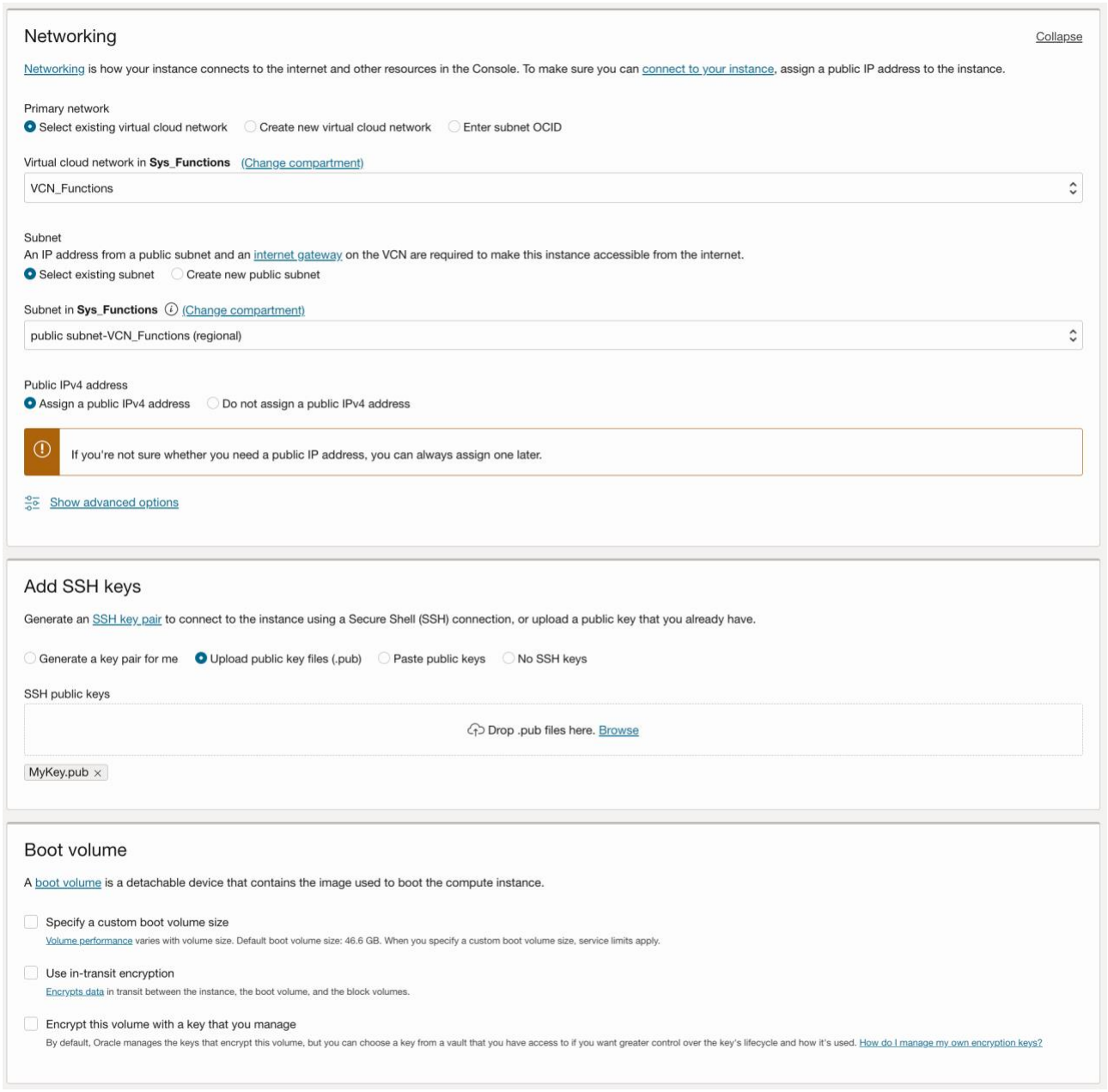

Configuration
Basic information
VCN name
VCN_Functions
Compartment
Sys_Functions
oraseemeaocids3 (root)/Sys_Functions
Configure VCN
VCN IPv4 CIDR block
10.0.0.0/16
If you plan to peer this VCN with another VCN, the VCNs must not have overlapping CIDR blocks. Learn more.
IPv6 prefixes Optional
Enable IPv6 in this VCN
DNS resolution
Use DNS hostnames in this VCN
Required for instance hostname assignment if you plan to use VCN DNS or a third-party DNS. This choice cannot be changed after the VCN is created. Learn more.
Configure public subnet
IP address type
IPv4 CIDR block
IPv4 CIDR block
10.0.0.0/24
Example: 172.16.0.0/16.
(Maximum number of items added)
+ Another IP address type
Configure private subnet
IP address type
IPv4 CIDR block
IPv4 CIDR block
10.0.1.0/24
Example: 172.16.0.0/16.
(Maximum number of items added)
+ Another IP address type

3 Create the building instance

Start compute instance creation

- Compute > Instances
- Select your hosting compartment (here Sys_Functions)
- Use an Oracle Linux 8 image
- You can select any shape available
- Connect the instance to the public subnet of the VCN created previously




Networking
Collapse
Networking is how your instance connects to the internet and other resources in the Console. To make sure you can connect to your instance, assign a public IP address to the instance.
Primary network
Select existing virtual cloud network
Create new virtual cloud network
Enter subnet OCID
Virtual cloud network in Sys_Functions (Change compartment)
VCN_Functions
Subnet
An IP address from a public subnet and an internet gateway on the VCN are required to make this instance accessible from the internet.
Select existing subnet
Create new public subnet
Subnet in Sys_Functions (Change compartment)
public subnet-VCN_Functions (regional)
Public IPv4 address
Assign a public IPv4 address
Do not assign a public IPv4 address
If you're not sure whether you need a public IP address, you can always assign one later.
Show advanced options
Add SSH keys
Generate an SSH key pair to connect to the instance using a Secure Shell (SSH) connection, or upload a public key that you already have.
Generate a key pair for me
Upload public key files (.pub)
Paste public keys
No SSH keys
SSH public keys
Drop .pub files here. Browse
MyKey.pub
Boot volume
A boot volume is a detachable device that contains the image used to boot the compute instance.
Specify a custom boot volume size
Volume performance varies with volume size. Default boot volume size: 46.6 GB. When you specify a custom boot volume size, service limits apply.
Use in-transit encryption
Encrypts data in transit between the instance, the boot volume, and the block volumes.
Encrypt this volume with a key that you manage
By default, Oracle manages the keys that encrypt this volume, but you can choose a key from a vault that you have access to if you want greater control over the key's lifecycle and how it's used. How do I manage my own encryption keys?

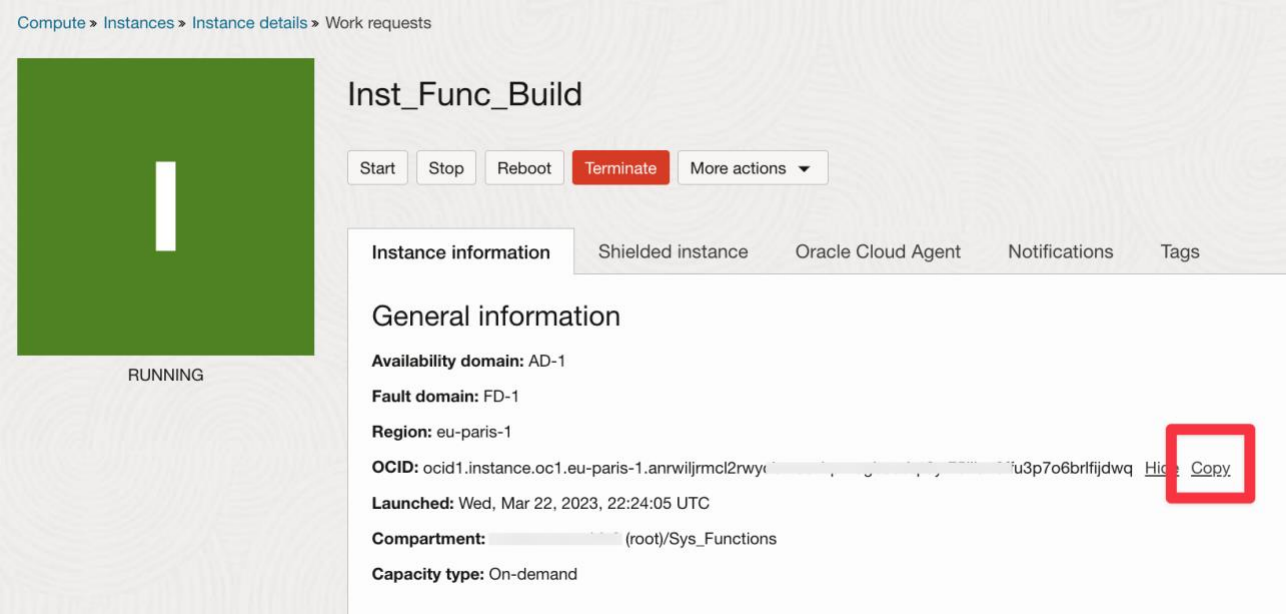
4 Create a Dynamic Group

Create a new dynamic group that includes the compute instance.

If you don't create a dynamic group and the appropriate policy, you must then create an API-key attached to your user account and configure your compute instance using

"oci setup config" command.

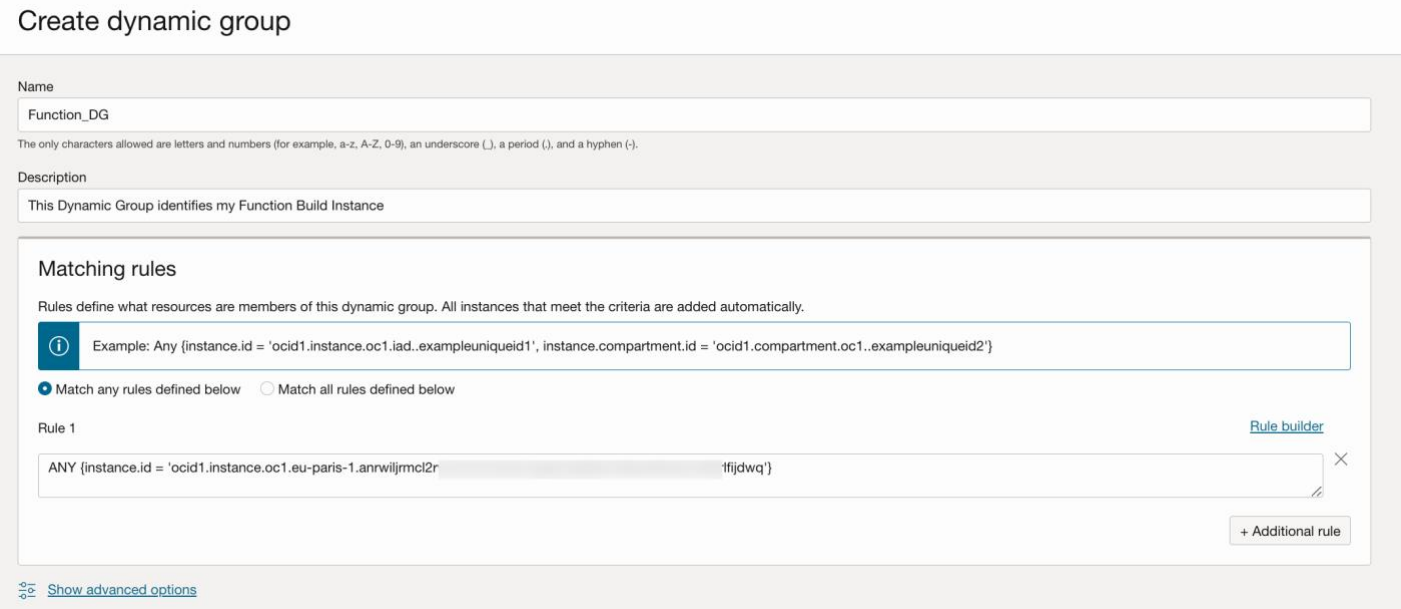
4.1 Get Instance OCID



4.2 Create a Dynamic Group

- Identity & Security > Identity > Dynamic Groups
- Add the following rule **using the OCID of your compute instance**

```
ANY {instance.id = 'ocid1.instance.oc1.eu-paris-1.anxxxxxxxxxxxxxxxxxwq'}
```

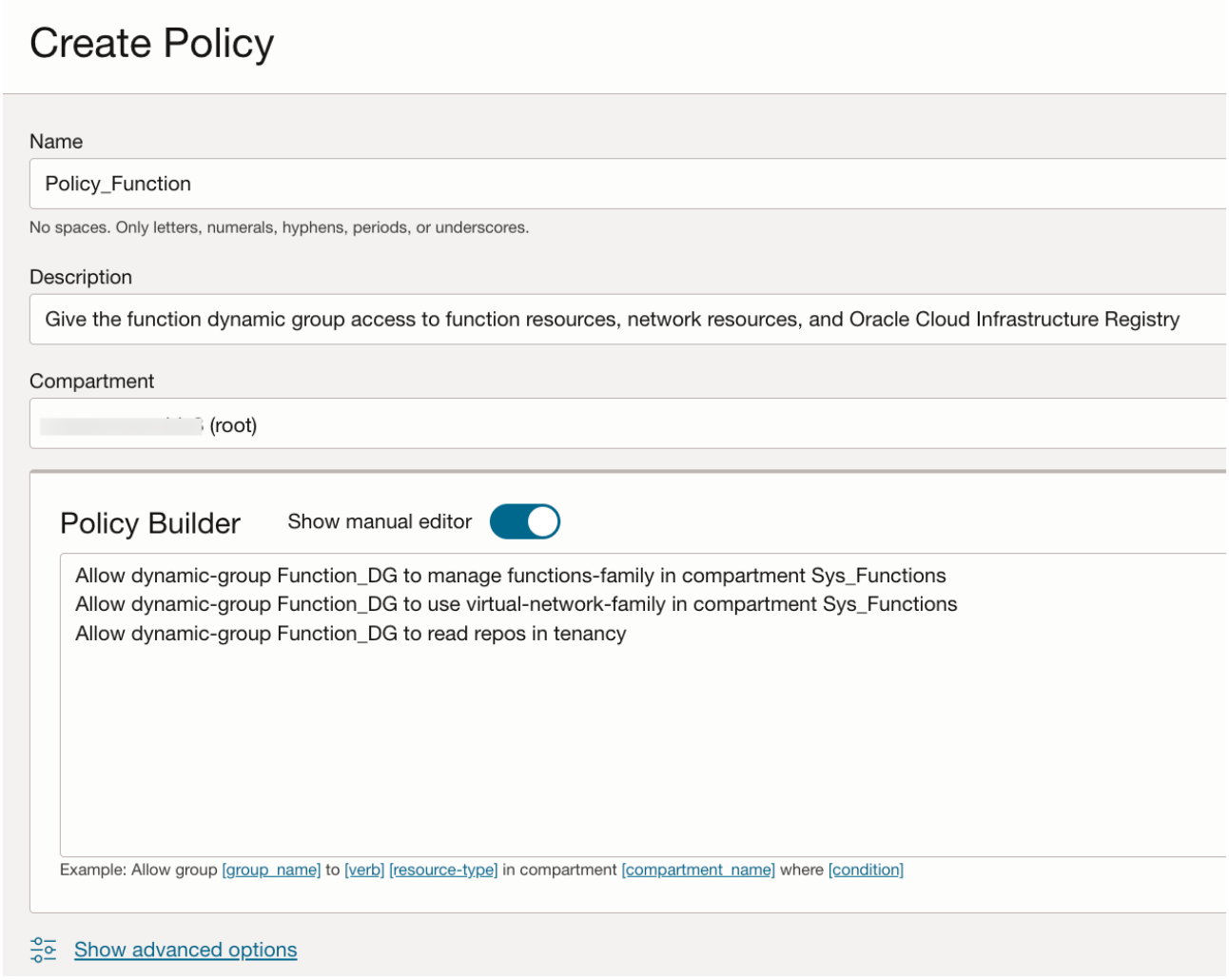


5 Create a policy

Create a policy **IN THE ROOT COMPARTMENT** to give the new dynamic group access to function resources, network resources, and Oracle Cloud Infrastructure Registry (OCIR).

- Identity & Security > Identity > Policies

```
Allow dynamic-group Function_DG to manage functions-family in compartment Sys_Functions
Allow dynamic-group Function_DG to use virtual-network-family in compartment Sys_Functions
Allow dynamic-group Function_DG to read repos in tenancy
```



6 Configure your Instance for building functions

These steps will set up your instance with all the components required to build your Dev environment.
Connect to your instance using ssh and its public IP.

6.1 Install components

You can do it manually running commands below or you can run the install script running :

Automatic Installation :

```
curl https://gist.githubusercontent.com/Olygo/814a4ee6451ac4ac821bcd55641e58a5/raw | sh
```

Manual installation:

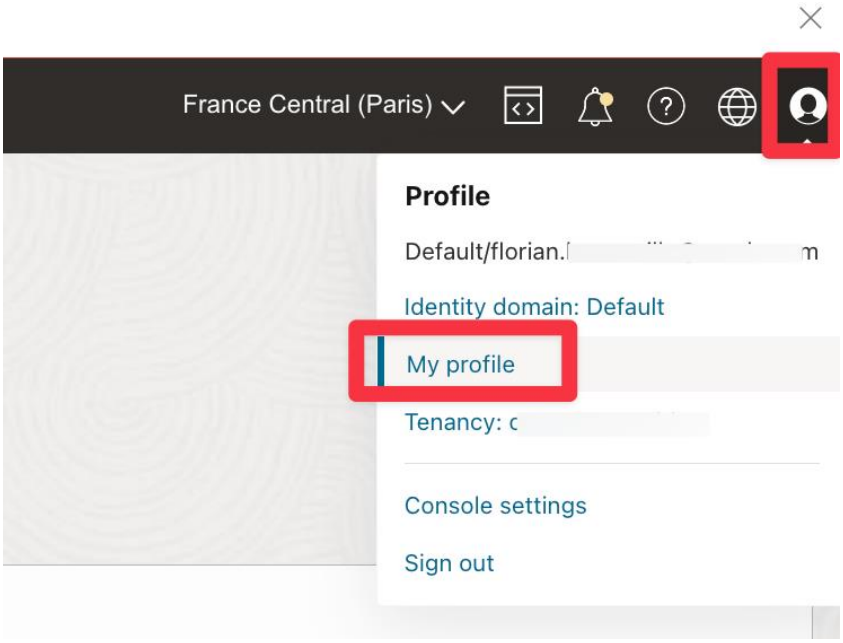
```
sudo dnf update -y
sudo dnf install git -y
git clone https://github.com/Olygo/OCI-FN_TagCompute_FF.git
python3 -m pip install pip --upgrade --user
python3 -m pip install wheel --upgrade --user
python3 -m pip install oci --upgrade --user
python3 -m pip show oci --version | grep Version
python3 -m pip install oci-cli --upgrade --user
python3 -m pip show oci-cli --version | grep Version
sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo -y
sudo dnf install -y docker-ce --nobest -y
docker version
sudo systemctl enable docker.service
sudo systemctl start docker.service
sudo usermod -a -G docker opc
curl -LSs https://raw.githubusercontent.com/fnproject/cli/master/install | sh
fn version
shutdown -r now
```

7 Create an Auth Token

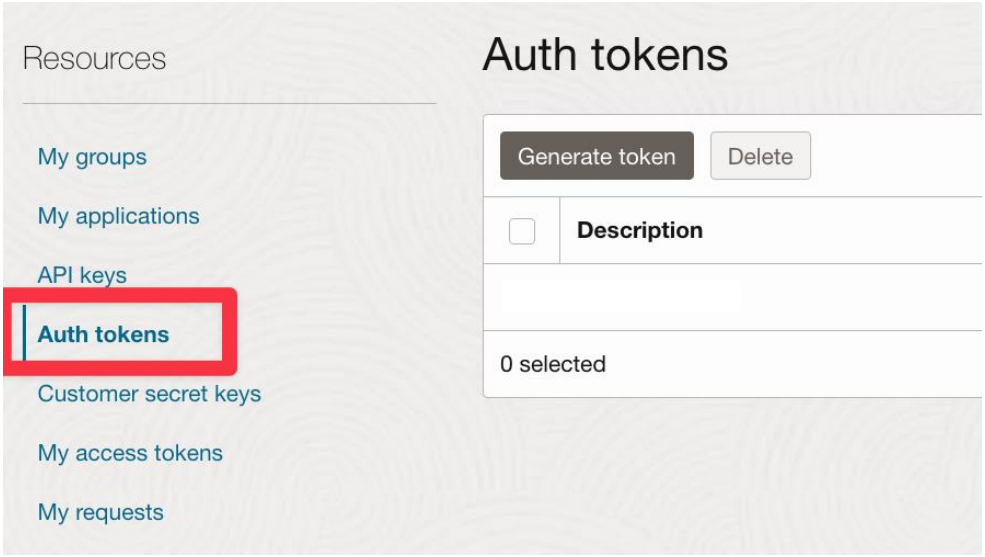
The Auth Token will be used to connect to your Container Registry (OCIR) that will host your docker container. This token will be required later for the docker login command.

7.1 Go to your account profile

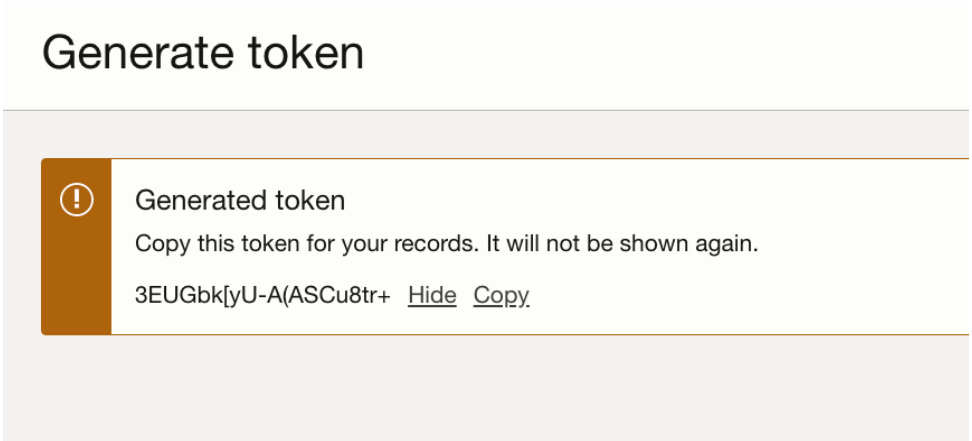
Click Profile (top right) > My profile



7.2 Generate a new Auth Token



7.3 Save your Auth Token

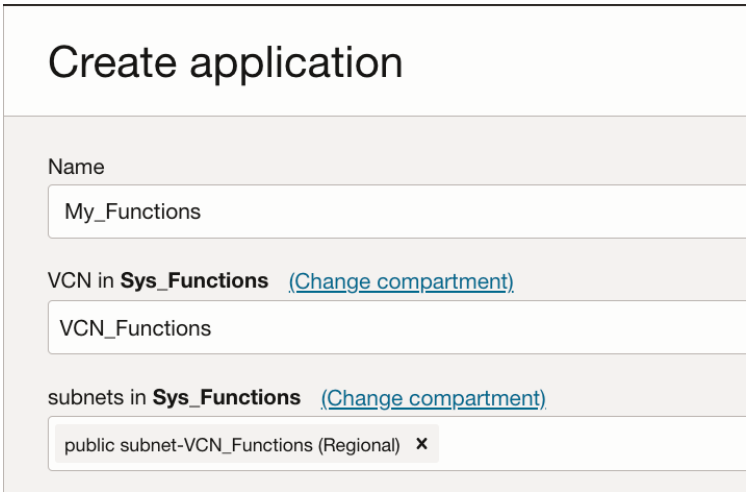


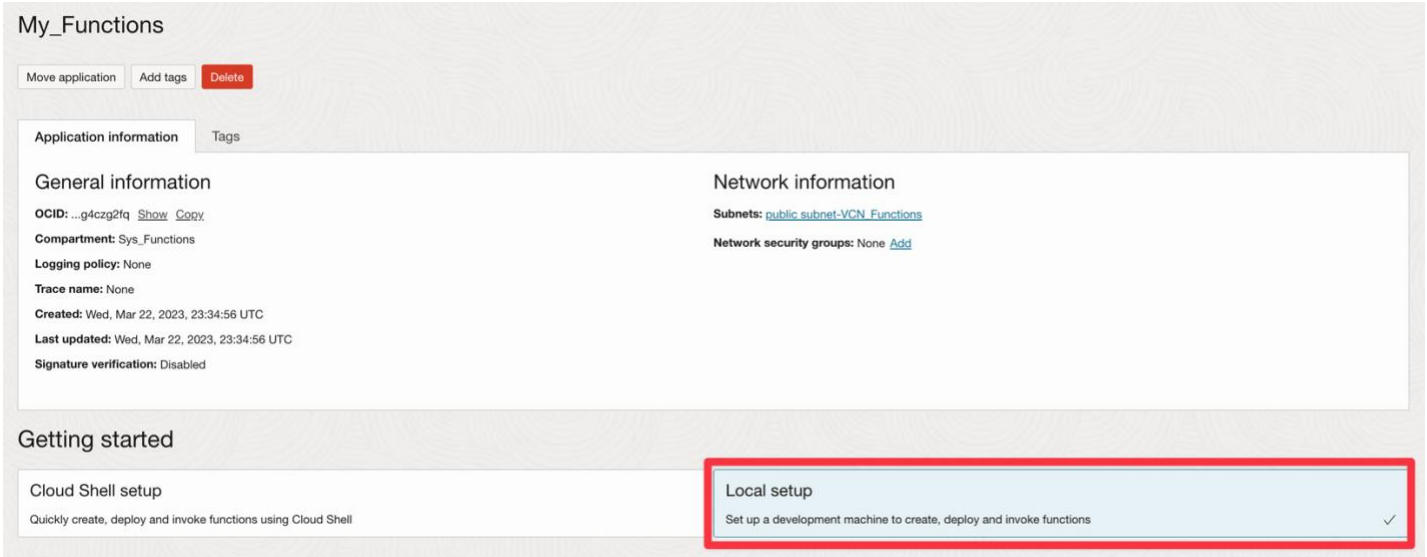
Save your auth token in your notes or personal vault for later use.

8 Create and deploy your function application

8.1 Create application from the OCI Console

- `Developer Services > Functions > Applications`





8.2 Configure your compute instance using “Local setup”

Steps 1 & 2 are not required here as we have already downloaded our function’s code through git clone above.

Jump into the function folder downloaded:

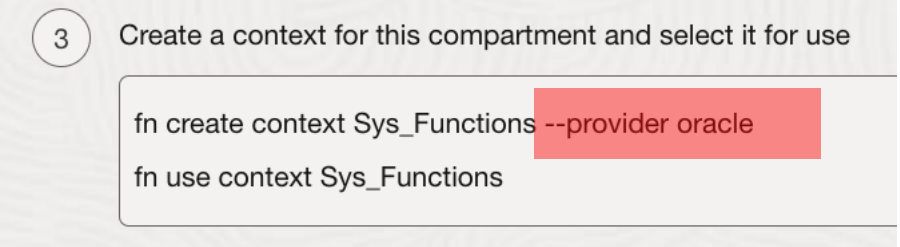
```
cd ./OCI-FN_TagCompute_FF
```

Now run function setup commands from the **Local Setup** section:

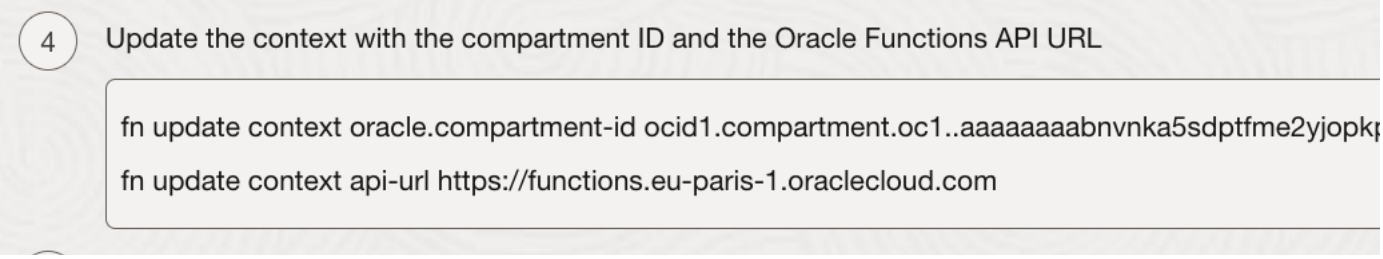
Step 3: Context name can be anything :

```
/!\ DO NOT USE: --provider oracle
       USE    : --provider oracle-ip
```

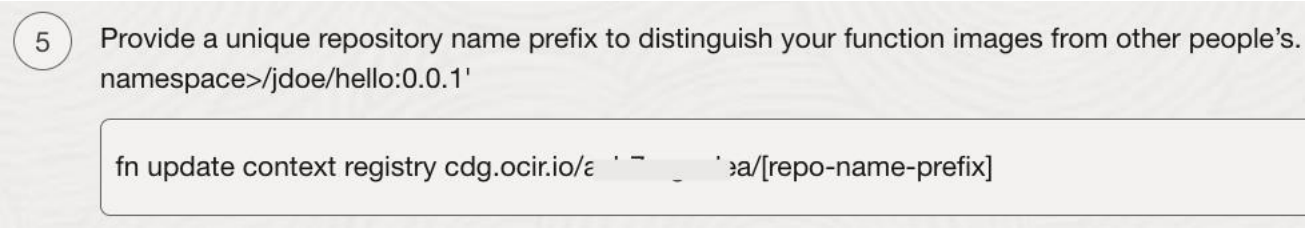
```
oracle: authenticates a user with a local oci-cli config file
oracle-ip: authenticates an instance with instance_principals (dynamic-groups)
```



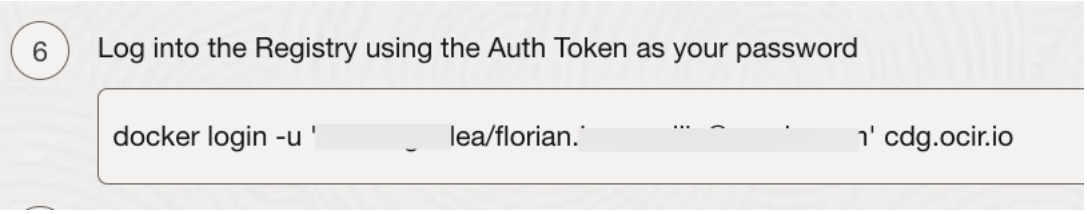
Step 4: Compartment hosting your function, API endpoint reflects your OCI region :



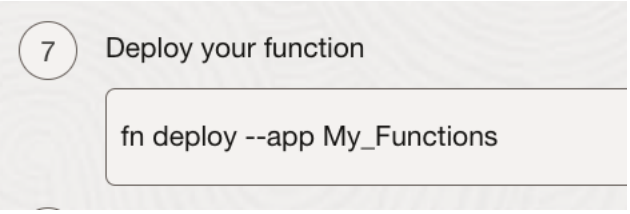
Step 5: Choose a name for the container registry repository (OCIR), it can be anything in lower cases without spaces : [repo-name-prefix]



Step 6: log into OCIR using docker login and your **Auth Token** created previously :



Step 7: Push the local function code to your OCI Application and OCIR repository:



8.3 Output from your compute instance :

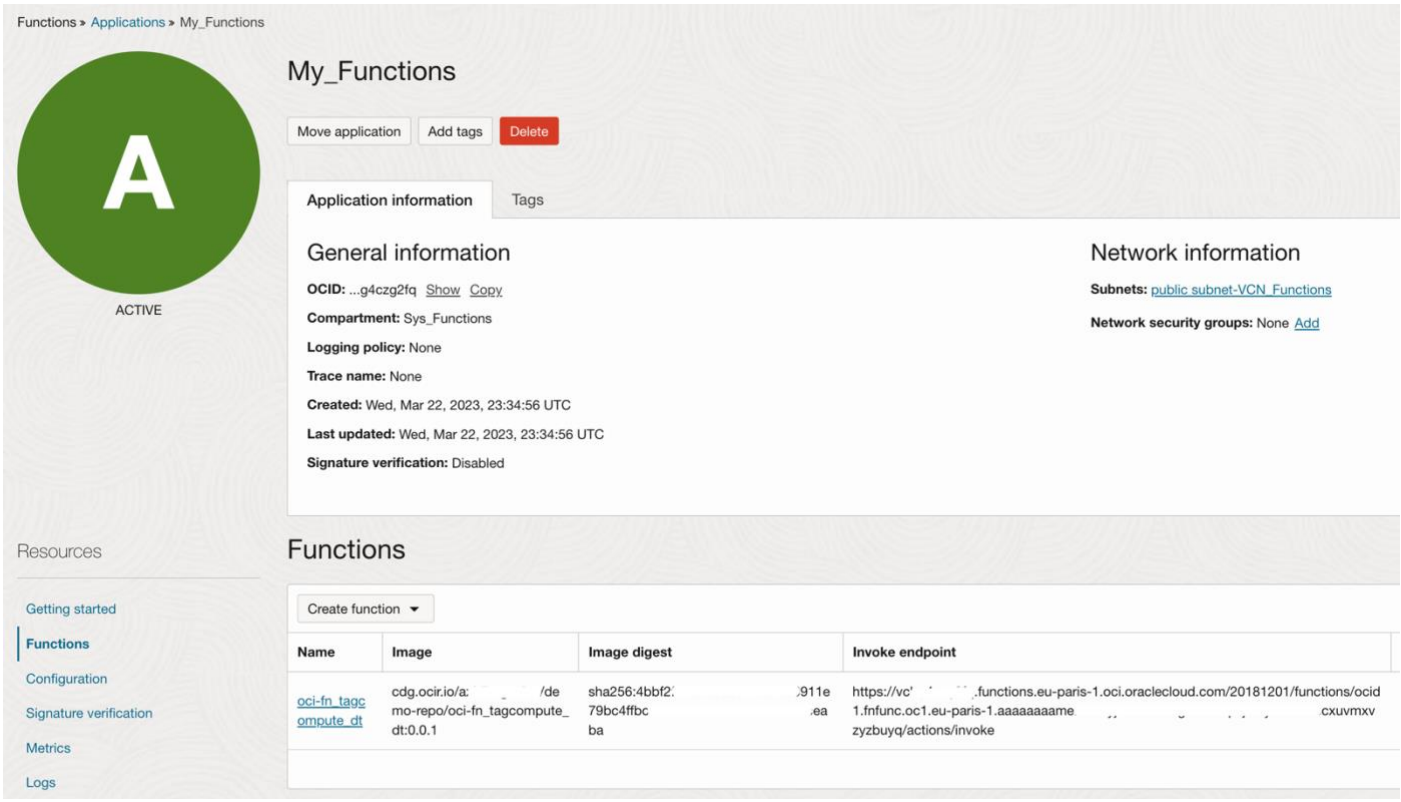
```
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ fn create context Sys_Functions --provider oracle-ip
Successfully created context: Sys_Functions
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ fn use context Sys_Functions
Now using context: Sys_Functions
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ fn update context oracle.compartment-id ocid1.compartment.oc1..aaaaaaaabn

Current context updated oracle.compartment-id with ocid1.compartment.oc1..aaaaaaaabnvnka5s
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ fn update context api-url https://functions.eu-paris-1.oraclecloud.com
Current context updated api-url with https://functions.eu-paris-1.oraclecloud.com
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ fn update context registry cdg.ocir.io/a      a/demo-repo
Current context updated registry with cdg.ocir.io/a      :a/demo-repo
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ docker login -u 'a      a/florian                    .com' cdg.ocir.io
Password:
WARNING! Your password will be stored unencrypted in /home/opc/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[opc@inst-func-build-142296 OCI-FN_TagCompute_DT]$ fn deploy --app My_Functions
Deploying oci-fn_tagcompute_dt to app: My_Functions
Bumped to version 0.0.1
Using Container engine docker
Building image cdg.ocir.io/a      a/demo-repo/oci-fn_tagcompute_dt:0.0.1 .....
Parts:  [cdg.ocir.io a      a demo-repo oci-fn_tagcompute_dt:0.0.1]
Using Container engine docker to push
Pushing cdg.ocir.io/a          a/demo-repo/oci-fn_tagcompute_dt:0.0.1 to docker registry...The push refers to repository [cd
pute_dt]
753aa253c2e1: Pushed
1466c8241698: Pushed
eb5011488c70: Pushed
64297c36ab1d: Pushed
0059976a0ea6: Pushed
e76ab7648bea: Pushed
f46abc7ec396: Pushed
0.0.1: digest: sha256:4bbf2fff28d8da9df79e70                    32c3ceaba size: 1782
Updating function oci-fn_tagcompute_dt using image cdg.ocir.io/a      a/demo-repo/oci-fn_tagcompute_dt:0.0.1...
Successfully created function: oci-fn_tagcompute_dt with cdg.ocir.io/a      a/demo-repo/oci-fn_tagcompute_dt:0.0.1
```

8.4 Output from your OCI console :

Your function has been published to your Application:



Your container function has been stored into your container registry:

By default, the repository is created in the root compartment, you can move it to another compartment using: Action > Move compartment.