INTRODUCTION

This page provides examples to get started using the Openstack Container service known as Zun .

Setup

You must have an account and CLI access as described in Cloud Intro .

Running a Public Container

To run a container, first launch it on a specific *internal* network. See Networks section of the CLI guide for more information about networks. You can use the general purpose cloud network:

```
openstack appcontainer run --net network=cloud --wait nginx
```

The container will be assigned a *name* and an internal IP on the cloud network. To access the container from the outside world, first obtain a Floating IP address and then assign it to the container:

```
openstack floating ip create --subnet public37 campus37
openstack appcontainer add floating ip lambda-15-container 128.135.xx.xx #
use the IP address result from the previous command.
```

Now you can access the public IP address of the container for the default Nginx start page.

Try these commands to get started (using the UUID of your running container instances):

```
openstack appcontainer list
openstack appcontainer exec --interactive ${UUID} /bin/bash
```

Private Registry Images

Openstack automatically pulls from the Dockerhub registry, or images uploaded to Glance with

```
openstack image create --container-format docker ...
openstack image list
```

To start a container image that was uploaded to Glance, provide a reference to it in the appcontainer. All other options remain as usual.

```
openstack appcontainer run --image-driver glance ...
```

Glance has no user authentication, but will prevent access to your image from outside your project. To use a private docker registry instead of glance, first create a registry row in Zun. The following

example is part of a CI pipeline based on Git, Docker, Gitlab, and Openstack.

```
openstack appcontainer registry create --username gitlab+deploy-token-99 --
password sekret --domain vcs.cs.uchicago.edu --name VCS_LDAP
```

You can then create and run images from this registry, for example

```
openstack appcontainer create --registry VCS_LDAP --name ldap1 --wait --net
network=campus37 --image-driver docker
vcs.cs.uchicago.edu:5050/chudler/ldapslave
```

Substitute `vcs.cs.uchicago.edu:5050/…` for your own custom Docker Registry,

Modifying and Saving a Container

Importantly, the repository that you push the image to must have been setup in zun. See the previous section.

```
openstack appcontainer run --net network=cloud--wait nginx
openstack appcontainer exec --interactive ${UUID} sed -i -s 's/Welcome to
nginx/Hello world/' /usr/share/nginx/html/index.html
openstack appcontainer commit ${UUID} VCS_LDAPSLAVE:os_latest
```

Caveats

Private registries are not supported for the `openstack appcontainer image pull`, and related commands. Nonetheless, the images are automatically pulled to the compute host, or can be customized with

```
openstack appcontainer create --image-pull-policy ...
```

GPU

(currently, unavailable***)

Simply start your container on the designated host.

```
openstack appcontainer run --availability-zone gpu --environment
NVIDIA_VISIBLE_DEVICES=all --auto-remove --wait --interactive
nvidia/cuda:10.0-base nvidia-smi
```

From:
https://howto.cs.uchicago.edu/ - **How do I?**

Permanent link:
https://howto.cs.uchicago.edu/cloud:recipe:docker

Last update: **2020/11/18 16:28**