Installing using Docker/Podman compose

Installation · https://2yetis.net/docs/installing-using-docker-podman-compose/

Introduction

This article describes how to install uvuyo using docker/podman compose. Docker/podman compose will setup all containers needed to run uvuyo on a single machine. So it is very similar to the installation described in the article installing uvuyo in a docker/podman environment. The major difference is that compose will automate the creation of the pods, network and volumes.

Note
To get e better understanding about the architecture of uvuyo read the article about the uvuyo archtecture. This is not a prerequisite to do the installation, but might help you getting started faster.

The installation described in this article will like the installation using docker/podman install uvuyo on a single machine. Therefor it is recommended to use this installation type in development and test environments only, since the installation will not provide high availability and the performance is limited to the resources available on the machine you install uvuyo on. Nevertheless this is the easiest and fastest way on installing uvuyo.

Preparation

To install uvuyo in a docker/podman compose environment, you need to have installed docker or podman together with their appropriate compose components.

Besides the docker/podman environment you need to be able to download the images used. The images are loaded from dockerhub so ensure that you have access to the internet.

Note
Some organizations might limit the acces to the internet a prevent you from downloading the docker/podman images directly. In this case you might need to talk to you network administrators on how to access the containers needed.

Create the uvuyo Directory

The uvuyo node reads specific data like it's core configuration and the license file from the file system. In this step you will create the directory where uvuyo will look for these files. Choose a directory. This will be the uvuyo home directory. In the uvuyo home directory create a directory called etc.

On unix like systems it is a good choice to name the uvuyo home directory /opt/2yetis/uvuyo. On windows systems the directory name could be something like c:/apps/2yetis/uvuyo. It does not really matter what name you choose. You need to remember the name however, since it is used later on in the process.

Create the docker / podman compose file

The docker / podman compose file instructs docker / podman compose on which containers it should create which volumes should be used and it sets up a network beween the configured containers. Copy the following sample docker / podman compose file and save it on the machine where you want to install uvuyo. Name the file uvuyo-compose.yml.

```
# (c) 2022 - 2024 2Yetis Software UG, All rights reserved
# This docker compose file serves the common services used by uvuyo
#
version: "2"

services:
  zookeeper:
    image: docker.io/bitnami/zookeeper:latest
    container_name: "uvuyo-zookeeper"
    ports:
      - "2181:2181"
    volumes:
      - "zookeeper_data:/bitnami"
    environment:
      - ALLOW_ANONYMOUS_LOGIN=yes

  kafka:
    image: docker.io/bitnami/kafka:latest
    container_name: "uvuyo-kafka"
    ports:
      - "9092:9092"
```

```
    volumes:
      - "kafka_data:/bitnami"
    environment:
      - KAFKA_CFG_ZOOKEEPER_CONNECT=zookeeper:2181
      - KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://kafka:9092
      - ALLOW_PLAINTEXT_LISTENER=yes
    depends_on:
      - zookeeper

  elasticsearch:
    image: docker.io/bitnami/elasticsearch:latest
    container_name: "uvuyo-elasticsearch"
    environment:
      - xpack.security.enabled=false
      - discovery.type=single-node
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    cap_add:
      - IPC_LOCK
    volumes:
      - elasticsearch_data:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
      - 9300:9300

  uvuyo:
    image: docker.io/the2yetis/uvuyo:latest
    container_name: "uvuyo"
    environment:
      UVUYO_HOME: /app/uvuyo
      UVUYO_KAFKA_BOOTSTRAP_SERVERS: kafka:9092
      UVUYO_ELASTIC_URL: http://elasticsearch:9200
    ports:
      - "8443:8443"
      - "8162:162/udp"
      - "1828:1828"
    volumes:
      - c:\apps\uvuyo\etc:/app/uvuyo/etc
    depends_on:
```

```
      - kafka


volumes:
  zookeeper_data:
    driver: local

  kafka_data:
    driver: local

  elasticsearch_data:
    driver: local
```

Set the uvuyo home etc directory

After you created the docker / podman compose file. you need to make some changes to it. The first change is to tell the uvuyo container to use the uvuyo home etc directory you have created in the previous step. There for search the uvuyo-compose.yml file for the volume mapping /app/uvuyo/etc and change it to map to your directory. (In the sample file above you will find this in line 64. Change the first part before the colon to the uvuyo home etc directory you created.

Note
The docker compose file is a YAML file. YAML is very specific about indents. So make sure that you don't change the indent of the line else you might experiance problems later on when the file is read by docker / podman compose.

Set the image repository

As mentioned previously your organization might have setup their network in a way that you are not able to download the images directly from dockerhub. In this case you would need to change the lines where the images are defined appropriately. In the example above the prefix in the image docker.io indicates that the image is loaded from docker hub.

Creating the containers, volumes and network

After you have created the docker / podman compose file you are ready to create the container, volumes and images defined in the file. To do so run the following command:

- Docker
- Podman

```
docker-compose -f uvuyo-compose.yml create
```

```
podman compose -f uvuyo-compose.yml create
```

If you didn't save the compose file with the filename uvuyo-compose.yml make sure you edit the command appropriately by changing the filename after zthe -f option.

After the command was run successfully the resources should have been created. You can check this by running the appropriate docker / podman commands.

To check that the containers have been created run the following command:

- Docker
- Podman

```
docker ps --all
```

```
podman ps --all
```

The command should list all the containers configured in the system. Search for the four containers used by uvuyo. These are the uvuyo, the zookeeper, the kafka and the elasticsearch containers.

Note
Docker and podman are nearly 100% compatible. on unix like systems an alias is created so that you can use the docker commands starting with docker even when you are running in a podman environment.

Starting uvuyo

To start uvuyo run the following command:

- Docker

- Podman

```
docker-compose -f uvuyo-compose.yml start
```

```
podman compose -f uvuyo-compose.yml start
```

Stopping uvuyo

To stop uvuyo run the following command:

- Docker
- Podman

Add your content here...

```
docker-compose -f uvuyo-compose.yml stop
```

```
podman compose -f uvuyo-compose.yml stop
```

Remove uvuyo

To remove the uvuyo installation run the following command:

- Docker
- Podman

```
docker-compose -f uvuyo-compose.yml rm
```

```
podman compose -f uvuyo-compose.yml rm
```