Modelbox on Testdrive

This repo contains notes, instructions, scripts, and other artifacts you may need to run Modelbox containers on Testdrive machines.

Configure Podman (sysadmin tasks)

Due to high security risks of Docker we recommend using Podman as a container tool.

Install Podman

On RHEL systems the `podman` package is available in the OS repositories. So you should run:

```
$ sudo dnf install podman
```

In case you had Docker installed previously it will give you an error message telling which package you should uninstall to be able to install Podman.

Configure users

Podman maps UIDs and GIDs from normal numbers to much higher numbers. To configure this mapping run the following command for each user:

```
$ sudo usermod --add-subuids 200000-265535 --add-subgids 200000-265535 pavel.anni
```

The next user should change `2` to `3` in all four places. For additional users just increase this number.

Check the ranges that are already taken:

```
$ cat /etc/subuid
snuser1:100000:65536
pavel.anni:200000:65536
$ cat /etc/subgid
snuser1:100000:65536
pavel.anni:200000:65536
```

After that, run (as your normal user):

```
$ podman system migrate
```

Check if it was applied. In both commands you should see two lines in the output.

```
$ podman unshare cat /proc/self/uid_map
         0      61005          1
         1     200000      65536
$ podman unshare cat /proc/self/gid_map
         0  213314702          1
         1     200000      65536
```

Configure storage

By default Podman uses the user's home directory for container storage. If the system uses NFS for home directories then rootless Podman won't work with them. That means you have to use local drives for container storage.

For each user change their container storage configuration:

```
$ mkdir -p ~/.config/containers
$ mkdir -p /nvmedata/$USER/containers/storage
$ vim ~/.config/containers/storage.conf
```

and add the following to the `storage.conf` file:

```
[storage]
driver = "overlay"
runroot = "/run/user/61005"
graphroot = "/nvmedata/pavel.anni/containers/storage"
```

Change the 3rd and 4th line according to your uid and username.

Check if the user's directory exists under `/run/user` and is owned by the user:

```
# ls -l /run/user/
total 0
drwx------.  3 arvind.sujeeth 213314702  80 Jun 22 10:23 61004
drwx------. 14 pavel.anni     213314702 580 Jun 22 10:38 61005
```

Create it if it doesn't exist and change the owner and permissions accordingly:

```
# mkdir /run/user/61003
# chown sen:213314702 /run/user/61003
# chmod 0700 /run/user/61003
```

If SELinux is enabled (it should!) use this command to apply proper setting the that new storage directory. See here: https://github.com/containers/podman/blob/main/troubleshooting.md#11-changing-the-location-of-the-graphroot-leads-to-permission-denied

```
$ sudo semanage fcontext -a -e /var/lib/containers /nvmedata/pavel.anni/containers
$ restorecon -R -v /nvmedata/pavel.anni/containers
```

Start the socket service

To be able to use Podman from the script we need a socket. For each user start the following service as a normal user.

```
$ loginctl enable-linger sen
$ export XDG_RUNTIME_DIR=/run/user/$(id -u)
$ systemctl start --user podman.socket
```

The first two commands are taken from here: https://access.redhat.com/discussions/6029491

Check the status:

```
$ systemctl start --user podman.socket
```

And check if the socket exists:

```
$ ls -l /run/user/$UID/podman
total 0
srw-rw----. 1 pavel.anni 213314702 0 Jun 21 14:53 podman.sock
```

Use Podman (user tasks)

Check if you can run containers with Podman

Run this:

```
$ podman run -it --rm fedora
```

You should get a root prompt. If there are issues, check this Troubleshooting section in Podman documentation: https://github.com/containers/podman/blob/main/troubleshooting.md

Exit from the session by typing `exit` at the prompt. The container will be automatically deleted.

Activate Podman virtual environment

The `start_model_podman.py` script uses the `podman` package which is installed in a virtual environment. Activate the virtual environment:

```
$ source /opt/sambaflow/virtualenvs/podman/bin/activate
```

Start the model

Change to the directory where the script is located and run the script to start the modelbox:

```
$ cd /scratch/pavel.anni/modelbox_podman/scripts
$ python start_model_podman.py \
    --init-ckpt-path /nvmedata/shared/models/GPT13B_ITV2_HA/V2_Aligned \
    --pef /nvmedata/shared/pefs/modelbox_inference/gpt_13b_generative_inference.pef \
    host1:5000/gpt_13b_generative_inference:1.1.0-20230615
```

Run the client

Use the client script to send requests to the model. Open another terminal window or tab and run this command:

```
$ cd /scratch/pavel.anni/modelbox_podman/scripts
$ python run_model.py --predict-input "Tell me a story about Ford Model T."
```

Stop the model

Switch to the first terminal window where you ran the `start_model_podman.py` script. You will see messages like the one below repeating every 60 seconds:

```
status {
  elapsed_time: 178.59662866592407
  message: "model is serving"
}
```

Press Ctrl-C to stop the model. This should delete the container as well.

Make sure the container is not running:

```
$ podman ps
```

You should see the empty list of containers (unless you are running other containers using Podman).

```
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
```