DivSeek Canada Portal Documentation

Lacey-Anne Sanderson

Oct 11, 2020

GUIDES:

The DivSeek Canada Portal is a web-based platform to implement association genetics workflows supporting plant breeding and crop research focusing on large scale plant genetic resources / crop genotype-phenotype data sets whose access is brokered / managed by the project.

CHAPTER

ONE

CREATE DIVSEEK CANADA PORTAL GUIDE

This guide will walk you through how to create a copy of the DivSeek Portal. The DivSeek portal has been designed to encapsulate the functionality in a collection of Docker containers so that you can **easily create your own portal with the same functionality to house your data**.

1.1 Finding a place to Host your Portal

The DivSeek Canada Portal is being designed to run within a Docker Compose deployed system of Docker containers when the application is run on a Linux server or virtual machine.

Platform Requirements:

- An operating system able to install docker and run instances; Linux preferred.
- The ability to expose ports to the outside world.
- **Enough physical space to house the website containing your data. You can choose to keep all of the following on a single p**
  - Root partition: at least 20Gb
  - Docker partition: at least 200Gb for docker
  - Each Crop partition: at least 500Gb
- Enough processing power to quickly deliver pages (at least 4 core, 6 GB RAM) with additional processing power required for analysis completed via Galaxy.

1.1.1 ComputeCanada OpenStack Cloud setup

We have chosen to host the main DivSeek Canada portal on the ComputeCanada Cloud. To do so for your portal, first request an allocation which based on the platform requirements listed above. The following steps indicate how to prepare your allocation for your own portal.

Create the Cloud Instance

We start by creating a persistent p4-6gb (4 core, 6 GB RAM) flavour of compute instance. The security group should open up the TCP/IP ports exposed by the various docker instances, as specified in the project's docker-compose.yml file.

Note: It is important to ensure that the "root" partition gets a sufficiently large disk volume size - typically, at least 20 gigabytes - generally larger than the default image size - which is sometimes as small as 2 gigabytes. This root volume size must be explicitly set during cloud instance launch since it is not generally changeable after the instance has been launched.

Docker Image and Volume Storage

By default, the Docker image/volume cache (and other metadata) resides under /var/lib/docker which will end up being hosted on the root volume of a cloud image, which may be relatively modest in size. To avoid "out of file storage" messages, which related to limits in inode and actual byte storage, it is advised that you remap (and copy the default contents of) the /var/lib/docker directory onto an extra mounted storage volume (which should be configured to be automounted by fstab configuration).

In effect, it is generally useful to host the entire portal and its associated docker storage volumes on such an extra mounted volume. We generally use the /opt subdirectory as the target of the mount, then directly install various code and related subdirectories there, including the physical target of a symbolic link to the /var/lib/docker subdirectory. You will generally wish to set this latter symbolic link first before installing Docker itself (here we assume that docker has not yet been installed (let alone running).

In Compute Canada, using the OpenStack dashboard, a cloud "Volume" can be created and attached to a running DivSeek Canada Portal cloud server instance. We suggest creating a volume at least 200 GB in size (to allow for significant genomic data storage). After attaching the volume to the instance, the volume is initialized and mounted from within an SSH terminal session, as follows (where '$' is the Linux Bash CLI terminal prompt):

```
# Before starting, make sure that the new volume (here, 'vdb') is visible (should be!)
# NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
# vda     254:0    0  2.2G  0 disk
# ├─vda1  254:1    0  2.1G  0 part /
# ├─vda14 254:14   0    4M  0 part
# └─vda15 254:15   0  106M  0 part /boot/efi
# vdb     254:16   0  200G  0 disk

# First, initialize the filing system on the new, empty, raw volume (assumed here to␣
↪be on /dev/vdb)
sudo mkfs -t ext4 /dev/vdb

# Mount the new volume in its place (we assume that the folder '/opt' already exists)
sudo mount /dev/vdb /opt

# Provide a symbolic link to the future home of the docker storage subdirectories
sudo mkdir /opt/docker
sudo chmod go-r /opt/docker

# It is assumed that /var/lib/docker doesn't already exist.
# Otherwise, you'll need to delete it first, then create the symlink
sudo ln -s /opt/docker /var/lib
```

It is also recommended that a separate additional volume for each crop dataset deployed be created, following the above instructions. A 500 GB volume is likely to be needed for this given the genomic large data sets involved. Also,

you can attach volumes containing the raw input data (e.g. VCF files) as you need (Note: you should obviously not run the mkfs command afresh on any volumes which already have data!) It is recommended first that you create a subdirectory /opt/divseekcanada then mount these additional volumes in that subdirectory, namely something like the following:

```
# Create a master folder for the DivSeek Canada code
sudo mkdir -p /opt/divseekcanada
# ensuring easy $USER access to these resources
sudo chown ubuntu:ubuntu /opt/divseekcanada
# Add the data volumes
sudo mkdir -p /opt/divseekcanada/data/downy-mildew
sudo mount /dev/vdc /opt/divseekcanada/data/downy-mildew
sudo mkdir -p /opt/divseekcanada/Sunflower
sudo mount /dev/vdd /opt/divseekcanada/Sunflower

After completing the above steps, you should configure /etc/fstab file for system␣
↪boot up mounting of the new volumes:

# These volumes need to be auto mounted upon each reboot of the system
# so you should (carefully) add them to the Linux /etc/fstab file
# of the server, something like the following text entries (customize for your crop):
/dev/vdb        /opt    ext4    rw,relatime     0       0
/dev/vdc        /opt/divseekcanada/data/downy-mildew    ext4    rw,relatime     0     ␣
↪  0
/dev/vdd        /opt/divseekcanada/Sunflower    ext4    rw,relatime     0       0

# test the fstab mount with a 'fake' mounting
sudo mount -vf
```

Now, you can proceed to install Docker and Docker Compose.

1.2 Install Pre-requisites

1.2.1 Installation of Docker

To run Docker, you'll obviously need to install Docker first in your target Linux operating environment (bare metal server or virtual machine running Linux).

For our installations, we typically use Ubuntu Linux, for which there is an Ubuntu-specific docker installation using the repository. Note that you should have 'curl' installed first before installing Docker:

```
sudo apt install curl
```

For other installations, please find instructions specific to your choice of Linux variant, on the Docker site.

Testing Docker

In order to ensure that Docker is working correctly, run the following command:

```
sudo docker run hello-world
```

This should result in something akin to the following output:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:be0cd392e45be79ffeffa6b05338b98ebb16c87b255f48e297ec7f98e123905c
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

  docker run -it ubuntu bash

  Share images, automate workflows, and more with a free Docker ID:
    https://cloud.docker.com/

  For more examples and ideas, visit:
    https://docs.docker.com/engine/userguide/
```

1.2.2 Installing Docker Compose

You will then also need to install Docker Compose alongside Docker on your target Linux operating environment.

Note: Note that under Ubuntu, you likely need to do a bit more preparation to avoid having to run docker (and docker-compose) as 'sudo'. See here for details on how to fix this.

Testing Docker Compose

In order to ensure Docker Compose is working correctly, issue the following command:

```
docker-compose --version
docker-compose version 1.22.0, build f46880f
```

Note: Note that your particular version and build number may be different than what is shown here. We don't currently expect that docker-compose version differences should have a significant impact on the build, but if in doubt,

refer to the release notes of the docker-compose site for advice.

1.2.3 Installing the DivSeek Canada Portal code base

This project resides in this Github project repository.

First, ensure that you have the git client installed (here again, we assume Ubuntu; '$' is the bash CLI prompt):

```
sudo apt update
sudo apt install git
```

Next, you should configure git with your Git repository metadata and, perhaps, activate credential management (we use 'cache' mode here to avoid storing credentials in plain text on disk)

```
git config --global user.name "your-git-account"
git config --global user.email "your-email"
git config --global credential.helper cache
```

Then, you can clone the project. A convenient location for the code is in a folder under /opt/divseekcanada:

```
cd /opt/divseekcanada
git clone https://github.com/DivSeek-Canada/divseek-canada-portal
```

1.3 Deployment of the Portal System

the DivSeek Canada Portal customizes a git fork of the Galaxy Genome Annotation "Dockerized GMOD" code base. The core of the customization is in the Docker Compose build file (docker-compose.yml) on the divseek-canada-build branch. Prior to running the build, however, some configuration tasks need to be completed.

1.3.1 Docker Compose Parameter Setting

The docker-compose.yml is parameterized for (crop) site specific site deployment using environment variables defined in a .env file, derived from the available template.env file, which needs to be copied into .env then customized to point to your actual public host particulars. For example, you can change the admin account particulars, i.e.

```
DC_ADMIN_USER=divseek_admin
DC_ADMIN_EMAIL=admin@divseekcanada.ca
```

or perhaps, the site crop, title, hostname, http protocol and Tripal path:

```
DC_CROP=Sunflower
DC_SITE_NAME="DivSeek Canada"
DC_SITE_BASE_HOSTNAME=sunflower.divseekcanada.ca
DC_BASE_URL_PROTO=https://
```

1.3.2 Docker Compose Preliminaries

The general project launch steps noted in the original GMOD deployment project README are otherwise followed, albeit with the divseek-canada-build customized docker-compose.yml file. To start off which, we can pre-load our Docker system with the required pre-built images, as follows:

```
docker-compose pull  # Pulls in the required service Docker images
```

1.3.3 NGINX Proxy Configuration

The original dockerized-gmod-deployment includes a 'proxy' service that runs the NGINX web server software in a container. To configure this package, the project specifies an NGINX configuration under a subfolder nginx. Unfortunately, most realistic site deployments (e.g. with https:// SSL configuration, particular hostnames, etc.) generally necessitates the creation of a customized NGINX file which, although taking the docker compose system into account, needs to also include additional elements, the composition of which this project cannot foresee and hard code (nor parameterize directly, since NGINX doesn't allow for that). The compromise we've taken here, in the divseek-canada-build branch, is to convert the default NGINX into a template, then provide some suggestions here on how to customize and properly deploy your copy of the template for use in the system. The following protocol is simply one that worked for us; those of you with deeper knowledge can likely converge on your own solution to the NGINX configuration.

1. In the divseek-canada-portal (a.k.a. dockerized-gmod-deployment) project, go into the nginx project subfolder and copy over the nginx/default.conf-template into nginx/default.conf.
2. Editing the nginx/default.conf file, rename the name my-divseek-portal-server of the server_name parameter and everywhere else that it is found inside the server block, to the DC_SITE_BASE_HOSTNAME hostname which you set in your .env file (e.g. sunflower.divseekcanada.ca). You should make sure that your DNS is properly set up to point to your cloud server IP address (usually with an A record) before proceeding to the next (certbot) step of the configuration. You may need to wait a short while for your DNS entry to propagate through the internet before attempting to run the configuration.
3. Configure https:// SSL certificate configuration. Using the certbot tool of the free certificate LetsEncrypt initiative is a nice way forward here, but you need to be a bit clever to achieve this since certbot generally requires that you specify the web server and operating system you are using so it can make reasonable assumptions about where things should go. This task is facilitated somewhat by using a Docker image for Certbox plus available instructions on how to set things up by a clever developer named 'Philipp'. We've partly applied the required certbot customizations to the docker-compose.yml file, but you'll need to apply the NGINX configuration edits and run the indicated procedure for the initial generation of SSL certificates for insertion into the configuration. For convenience, Philipp's init-letsencrypt.sh has been customized (to read the host name set in your .env file) and embedded in our project. If you have set your .env file correctly, then You may therefore run it as follows:

```
sudo ./init-letsencrypt.sh
```

4. You should now go back into the nginx/default.conf file and uncomment the directive include ./services.conf to enable inclusion of the full set of NGINX service proxy redirections.

Now we are set to build the system and fire it up.

1.3.4 Running the Docker-Compose Build

It is recommended to first start the databases (first making sure that you are in the root project directory for the code):

```
cd /opt/divseekcanada/divseek-canada-portal
docker-compose up -d apollo_db chado # Launches the database containers
```

In a new terminal, in the same folder, you can run docker-compose logs -f in order to follow the build process and decypher errors.

```
# Wait for tripal to come up and install Chado.
docker-compose up -d --build tripal

# It takes a few minutes until you see an apache start-up notification.
# Then, run a non-specific compose build to bring up the rest of the services.
docker-compose up -d
```

1.4 Troubleshooting

1.4.1 ElasticSearch

During the creation of the ElasticSearch indexing container in the Docker Tripal system, one may run up against another resource limit, reported by the following error message:

```
max virtual memory areas vm.max_map_count [65530] is too low, increase to at least␣
↪[262144]
```

This solution to this is to add this line:

```
vm.max_map_count=262144
```

in your /etc/sysctl.conf file on the host system and run

```
sudo sysctl -p
```

to reload configuration with new value.

CHAPTER

TWO

GENOME CANADA PILOT PROJECT

The first iteration of the platform is funded under a Genome Canada Project with co-funding from other partners.

> Growing populations, a changing climate and increasing constraints on land, water and fertilizer together translate into increased risks to global food security and pressure to dramatically expand agricultural productivity in Canada – and quickly. This can't happen, though, without accelerated plant breeding programs to develop high-yielding, climate-friendly and "earth-friendly" plant varieties. Further, Canada is required by the terms of international treaties and agreements to develop mechanisms for sharing these plants and the genetic information underlying them.
>
> DivSeek Canada is a project that will offer a way forward on both fronts. DivSeek Canada is the Canadian arm of an international initiative, DivSeek – a community driven effort involving a diverse set of partners who have voluntarily come together to unlock the potential of crop diversity to enhance the productivity, sustainability and resilience of crops and agricultural systems. This new Canadian-based project will accelerate plant breeding by leveraging the genetic diversity in the world's live collections and seed banks to create a unified, coordinated and cohesive information management platform. Canadian stakeholders will be consulted to guide the development of the platform, establish it on Compute Canada infrastructure and populate it with genomic information for three Canadian crops (lentils, flax and sunflower). It will also make mapping, breeding and visualization tools available on the platform for plant breeders, develop training resources and develop a long-term plan for its continued enhancement, as well as sustainable hosting, outreach and stakeholder support in Canada.
>
> The platform not only provides an expandable database for Canadian crop information, but also offers a model for the DivSeek initiative globally. It is expected to galvanize the use of genomic information by plant breeders to accelerate crop breeding in Canada, particularly in small to medium-sized crop communities who have not previously had the financial resources or bioinformatics skill set to exploit the genomic information available.
>
> —DivSeek Canada: Harnessing Genomics to Accelerate Crop Improvement in Canada