Technical Document

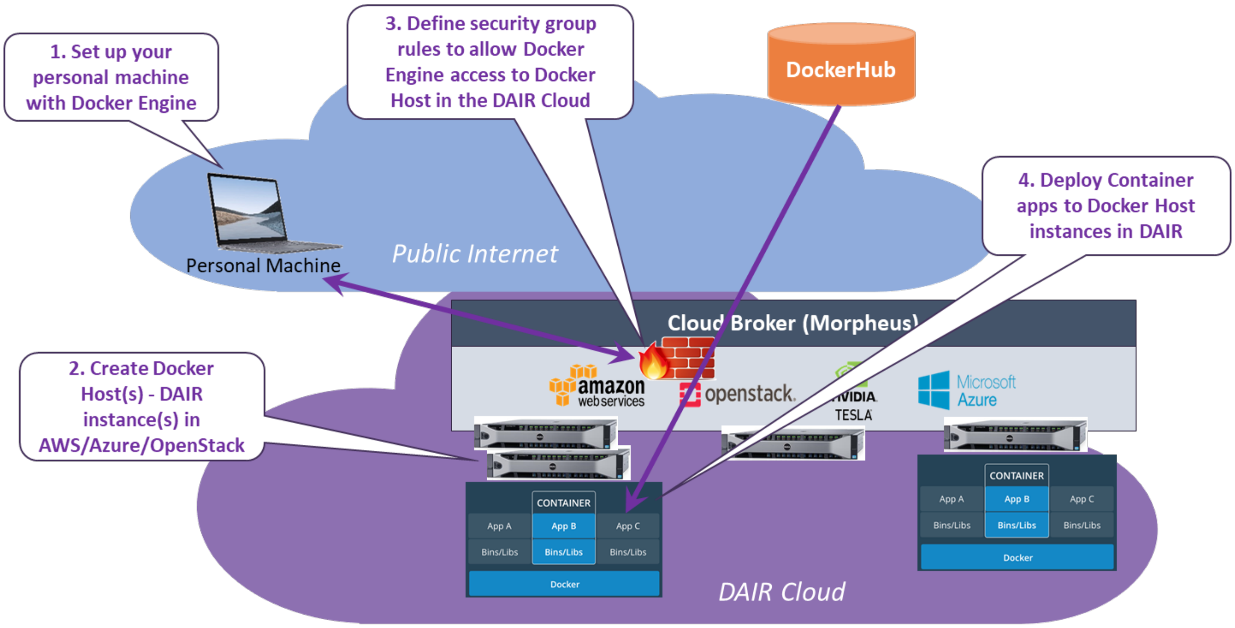
How to Deploy Docker in the DAIR Cloud

Author: Don McCullough
Solution Architect
dair.admin@canarie.ca
canarie.ca | @canarie_inc

Table of Contents

Overview

This guide provides step-by-step instructions to set up a fully functional Docker environment by creating a Docker host in the DAIR Cloud, which you will manage from your PC running Docker Engine. The figure below shows the solution created by following the steps in this guide, with the ultimate goal of being able to quickly deploy application containers to your Docker host in DAIR.


1. Set up your personal machine with Docker Engine
3. Define security group rules to allow Docker Engine access to Docker Host in the DAIR Cloud
DockerHub
4. Deploy Container apps to Docker Host instances in DAIR
Personal Machine
Public Internet
Cloud Broker (Morpheus)
amazon webservices
openstack.
NVIDIA TESLA
Microsoft Azure
2. Create Docker Host(s) - DAIR instance(s) in AWS/Azure/OpenStack
CONTAINER
App A
App B
App C
Bins/Libs
Docker
DAIR Cloud

1. Setup your personal machine with Docker Engine	3. Define security group rules to allow Docker Engine access to Docker Hos in the DAIR Cloud	Docker Hub	4. Deploy Container apps to Docker Host instances in DAIR
Personal Machine	Public Internet	Cloud Broker (Morpheus)	
2. Create Docker Host(s) – DAIR instance(s) in AWS/Azure/OpenStack		DAIR Cloud	

Resource Considerations

Before you begin, think about the number of containers you want to run on your host and calculate the resources (# of cores, memory, and storage space) required. Containers are exceptionally “light” when compared to non-containerized apps running on VMs, and allow you to run more apps with fewer resources. It is generally a good idea during development and testing to use multiple small instances rather than one large instance. This helps you to get a better understanding of scalable infrastructure as your application/service grows.

Security Considerations

It is always wise to plan your cloud infrastructure security group needs early on. Security best practices dictate that you provide only the minimum required access to your cloud resources.

In the DAIR Cloud, only open up ports 2376 & 22 between your Docker Engine (e.g. your local laptop for deployment of Docker containers) to your Docker Host in the cloud. At the end of this guide, we will load an Apache (web server) container, so you will also want to allow HTTP traffic on port 80 to 0.0.0.0/0 (i.e. "open to the world”).

Pre-Requisites

You will need to install Docker on your local laptop or a development machine of your choice beforehand. In doing so, you will have local control to manage what containers you choose to deploy and run on the Docker Host in DAIR. Please follow the installation instructions on the Docker website. This guide describes steps as though you are using Docker Community Edition on an Apple MacBook, but should be near identical if you are on a Windows laptop.

Tip: There is a Docker Slack channel if you need Docker support.

Step 1: Install Docker on Personal Machine (Mac Laptop)

Docker Hub: https://hub.docker.com

In Docker 2.2.0.0, Docker-machine was removed on Apple Macs but was reinstated in version 2.2.3.0. You may need to perform an upgrade or to use this command on your Mac:

```
$ brew install docker-machine
```

Step 2: Install Docker on a Linux Instance (Docker Host in DAIR)

In this step, you will install Docker on a Linux instance (named “docker-demo”) in the DAIR Cloud. If you are unsure how to create a Linux instance in DAIR, refer to the “Creating a Linux Instance” Section of the How to Use DAIR Cloud Resources – Technical Guide. This “docker-demo” instance will become your

Docker Host where you can deploy one or more Docker containers. Start by SSHing to your DAIR instance before running the following commands.

Connect via SSH to your DAIR Instance:

Use your **Username** and **SSH key** name and replace XXX.XXX.XXX.XXX with your instance EIP (external public or floating IP)

```
ssh  -i  ~/yourkey  yourusername@XXX.XXX.XXX.XXX
```

Install Docker Engine on Ubuntu [Reference Docker Documentation]

We are using Ubuntu for this example. There are documents for many other server options on the Docker Hub site, and Docker keeps these documents up to date. Alternatively, you can run these same commands from the Linux instance Console tab in the DAIR Cloud management interface.

Docker Install Process

```
$ sudo apt-get update
```

```
$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
```

```
$ sudo apt-key fingerprint 0EBFCD88
```

Expected output:

```

pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

```

$ sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
```

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Next, permit Sudo with no password for this Docker Host machine by editing the sudoers file shown below:

```
$ vi /etc/sudoers.d/90-cloud-init-users
replace yourusername ALL=(ALL) ALL with yourusername ALL=(ALL) NOPASSWD:ALL
:wq!
$ exit
```

Step 3: Set Security Rules for the Docker Host

In the DAIR Cloud, you will need open up ports 2376 & 22 between your Docker Engine (e.g. your local laptop for deployment of Docker containers) to your Docker Host in the cloud.
Tip: To determine the public IP address of your local machine, perform a Google search for “What’s my IP” from a browser on your laptop.

At the end of this guide, we will be loading an Apache (web server) container, so you will also want to create a rule allowing HTTP traffic on port 80 to 0.0.0.0/0 (i.e. "open to the world”). If you are unsure how to add a firewall rule to your default Security Group in DAIR, refer to the “Setup Security Groups / Firewalls” Section of the How to Use DAIR Cloud Resources – Technical Guide.

Step 4: Container Deployment (Apache)

Now that your Docker host is configured, you can go back to your Docker Engine (e.g. your Mac laptop). Run the following commands to complete the setup of your Docker environment and attach to your new Docker host in DAIR. Finally, you will deploy the Apache (HTTP server) container on your Docker host.

Add the host in DAIR to your local Docker environment

Use your **Username** and **SSH key** name/location and the **external IP** of your Docker Host machine (created in Step 3) as replacements for the highlighted arguments in the command below:

```
$ docker-machine create -d generic --generic-ip-address
XXX.XXX.XXX.XXX --generic-ssh-user yourusername --generic-ssh-key ~/
yourkey docker-demo
```

Example Output:

Expected output:

```
Running pre-create checks...
Creating machine...
(docker-demo) Importing SSH key...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with ubuntu(systemd)...
Installing Docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
```

To learn how to connect your Docker Client to the Docker Engine running on the host machine in DAIR, type this command:

```
$ docker-machine env docker-demo
```

Next, set your environment to your new machine and follow the instructions.

```
$ docker-machine env docker-demo
```

Expected output:

```
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://XXX.XXX.XXX.XXX:2376"
export
DOCKER_CERT_PATH="/Users/don.mccullough/.docker/machine/machines/docke
r-demo-azure"
export DOCKER_MACHINE_NAME="docker-demo"
# Run this command to configure your shell:
# eval $(docker-machine env docker-demo)
```

Run the recommend command (highlighted above) from the output of the last command:

```
$ eval $(docker-machine env docker-demo)
```

Now you can SSH into your Docker host directly from your local machine to verify everything is working.

```
$ docker-machine ssh docker-demo
```

Type “exit” to quit your SSH session on the Docker host.

Type the following command to verify your system status is “ACTIVE”:

```
$ docker-machine ls
```

.

Deploy the Apache Container

Finally, run a simple command to fetch the Apache container (from Docker Hub) and deploy/run it on your “docker-demo” host in DAIR:

```
$ docker run --name apache -d -it -p 80:80 httpd
```

To verify that the Apache container is running, point your browser to the external IP of your Docker host. You should see the default Apache web page.

That is all there is to it! You now have a Docker host and can easily deploy additional containers to it from your laptop.

Recommended Resources

1. A great companion to this document: Docker Commands for Daily Life

2. Recording of a previous DAIR webinar that shows all the steps in this guide as a video tutorial: How to Deploy Docker