链滴

Swarm 系列 2-- 独立 Swarm

作者：james

原文链接：https://ld246.com/article/1520329989222

来源网站：链滴

<h2 id="Docker-Swarm">Docker Swarm</h2>

参考：https://docs.docker.com/swarm/
独立 Docker Swarm 是 Docker 1.12 版本之前的遗产。Docker 公司当前不会反对用户使用它但是最好使用最新的 swarm 模式。

<h3 id="1--Docker-Swarm-简介">1. Docker Swarm 简介</h3>

Docker Swarm 是 docker 的本地集群。将一些 docker 主机组成一个大的虚拟 docker 主机。swarm 支持 Docker 自身的 API 来操作，以下都可以实现 swarm 操作：

Dokku
Docker Compose
Docker Machine
Jenkins

像别的 Docker 项目一样，swarm 项目也遵循“替换，插入，运行”的原则。作为基础开发原，API 是向后兼容的。意味着你可以替换你想要的 swarm 的调度模块。swarm 的可替换设计提供了滑的使用便利性。也允许大规模生产环境上的部署，使用更强大的后端调度，如 Mesos。

<h4 id="1-1-明白swarm集群的创建">1.1 明白 swarm 集群的创建</h4>

创建 swarm 集群的第一步是拉取 swarm 镜像 Docker Swarm image，然后使用 docker 来配置warm 的管理节点，使所有节点来运行 swarm。这要求：

在每个节点上开放 TCP 端口来与 swarm 的管理节点通信
在每个节点安装 docker
创建管理 TLS 证书来加密通信。

如果你想手动安装或，有志于贡献，参考：Build a swarm cluster for production

<h4 id="1-2-服务发现">1.2 服务发现</h4>

再 docker swarm 后端你需要配置服务发现服务。参考文档： Discovery service

<h4 id="1-3-高级调度">1.3 高级调度</h4>

Advanced scheduling 参考：策略 strategies 和过滤 filter 文档

<h4 id="1-4-SwarmAPI">1.4 SwarmAPI</h4>

SwarmAPI 于 dockerAPI 相一

，在一些地方有扩展。

<h3 id="2--开始使用Docker-Swarm">2. 开始使用 Docker Swarm</h3>

你可以使用 swarm 可运行镜像来创建容器，或者使用可运行 swarm 二进制文件来创建 swarm 群。本节将介绍两种方式，并讨论它们的利弊（discusses their pros and cons）

<h4 id="2-1-使用交互容器创建一个swarm集群">2.1 使用交互容器创建一个 swarm 集群</h4>

你可以使用 swarm 官方镜像来创建一个 swarm 集群。该镜像由 Docker 官方维护并定时自动新。使用 docker run 来运行镜像，并且有很多选项和子命令来创建和管理集群。
和其它容器镜像一样，你可以下载 swarm:latest 镜像或其它版本。

<pre><code class="highlight-chroma">james@james-CW65:~ > docker run swarm
Unable to find image 'swarm:latest' locally
latest: Pulling from library/swarm
dd72058debf0: Pull complete
cb543654edaf: Pull complete
44212202dc6d: Pull complete
Digest: sha256:c9a27b020ae4439432c842769d8e731661d5987962e33004114e4aba9d03b4c
Status: Downloaded newer image for swarm:latest
Usage: swarm [OPTIONS] COMMAND [arg...]

A Docker-native clustering system

Version: 1.2.8 (4886b1)

Options:
  --debug   debug mode [$DEBUG]
  --log-level, -l "info"   Log level (options: debug, info, warn, error, fatal, panic)
  --experimental   enable experimental features
  --help, -h   show help
  --version, -v   print the version

Commands:
  create, c   Create a cluster
  list, l   List nodes

</code></pre>

in a cluster
  manage, m   Manage a docker cluster
  join, j   Join a docker cluster
  help   Shows a list of commands or help for one command

Run 'swarm COMMAND --help' for more information on a command.
james@james-C65:~ >
</code></pre>
<h5 id="2-1-1-从容器运行swarm镜像">2.1.1 从容器运行 swarm 镜像</h5>

ssh 到一个节点
使用 docker run 运行 swarm 镜像。最简单的命令是 docker run swarm --help
上面运行过后，镜像会退出。

<pre><code class="highlight-chroma">james@james-CW65:~ > docker ps -a
CONTAINER ID IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
eeaf17dc3096 swarm   "/swarm --help"   7 minutes ago   Exited (0) 7 minutes ago   gallant_lamarr
</code></pre>
<h5 id="2-1-2-为什么使用镜像-">2.1.2 为什么使用镜像？</h5>

使用容器运行镜像有三个优势：

不需要安装二进制文件运行镜像
每次都可以用 docker run 来运行新镜像
容器使 swarm 环境与宿主机隔离。不需要维护环境。

因此运行镜像容器 是一个推荐的使用 swarm 集群的方式。所有以下文档都使用这种方式。

<h4 id="2-2-使用二进制运行Swarm">2.2 使用二进制运行 Swarm</h4>

再你在宿主机上运行 swarm 之前，你需要从源代码编译出可执行文件或从别的地方获取可靠的件。参考： ONTRIBUTING.md 来获取源代码

<h5 id="2-2-1-为什么使用可执行文件">2.2.1 为什么使用可执行文件</h5>

使用二进制可执行文件有个唯一的优点：如果你是开发者修改 swarm project 的代码，那你就需要容器化后再测试运行，可以直接运行。
有以下缺点：

源代码编译是个负担
可执行文件没有容器化运行的优点，如隔离。

大部分文档都是使用容器运行的方法。

最后由于没有 docker 引擎，一些工具无法再集群节点上运行，如命令行工具。

<h3 id="3--创建swarm集群">3. 创建 swarm 集群</h3>
<h4 id="1--生成发现token-运行swarm-create-来生成token-">1. 生成发现 token，运行 swarm create 来生成 token：</h4>
<pre><code class="highlight-chroma">james@james-CW65:~ > docker run swarm create
6e56a11eb2ac3efa7e467a99fa683b5
Token based discovery is now deprecated and might be removed in the future.
It will be replaced by a default discovery backed by Docker Swarm Mode.
Other mechanisms such as consul and etcd will continue to work as expected.
</code></pre>

注意该命令依赖于 swarm 自身的服务发现机制。要使用别的服务发现参考：discovery backends.

<h4 id="2--启动swarm管理节点">2. 启动 swarm 管理节点</h4>

集群中的管理节点负责调度集群中的容器。管理节点管理一系列的客户端 agents（也叫节点）
swarm 的 agent 用于负责承载容器。它们通常是 docker daemon，你可以使用 docker API 与交互。
本节创建 1 个管理节点，2 个工作节点

在 virtual Box 下创建一个 swarm 管理节点：

<pre><code class="highlight-chroma">--swarm   Configure Machine to join a Swarm cluster
--swarm-addr   addr to advertise for Swarm (default: detect and use the machine IP)
--swarm-discovery   Discovery service to use with Swarm
--swarm-experimental   Enable Swarm experimental features
--swarm-host "tcp://0.0.0.0:3376"   ip/socket to listen on for Swarm master
--swarm-image "warm:latest"   Specify Docker image to use for Swarm [$MACHINE_SWARM_IMAGE]
--swarm-join-opt --swarm-join-opt option --swarm-join-opt option] Define arbitrary flags for Swarm join
--swarm-master   Configure Machine to be a Swarm master
--swarm-opt [--s
</code></pre>

```
arm-opt option --swarm-opt option]                    Define arbitrary flags for Swarm master
</span></span><span class="highlight-line"><span class="highlight-cl">--swarm-strategy
spread"                                 Define a default scheduling strategy for Swarm
</span></span><span class="highlight-line"><span class="highlight-cl">james@james-C
65:~/.docker/machine/cache &gt; docker-machine create -d virtualbox --swarm --swarm-mas
er --swarm-discovery token://6e56a11eb2ac3ef3a7e467a99fa683b5 swarm-master
</span></span><span class="highlight-line"><span class="highlight-cl">Running pre-creat
 checks...
</span></span><span class="highlight-line"><span class="highlight-cl">Creating machine..

</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-master) C
pying /home/james/.docker/machine/cache/boot2docker.iso to /home/james/.docker/machi
e/machines/swarm-master/boot2docker.iso...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-master) C
eating VirtualBox VM...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-master) C
eating SSH key...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-master) S
arting the VM...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-master) C
eck network to re-create if needed...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-master)
aiting for an IP...
</span></span><span class="highlight-line"><span class="highlight-cl">Waiting for machi
e to be running, this may take a few minutes...
</span></span><span class="highlight-line"><span class="highlight-cl">Detecting operati
g system of created instance...
</span></span><span class="highlight-line"><span class="highlight-cl">Waiting for SSH to
be available...
</span></span><span class="highlight-line"><span class="highlight-cl">Detecting the prov
sioner...
</span></span><span class="highlight-line"><span class="highlight-cl">Provisioning with
oot2docker...
</span></span><span class="highlight-line"><span class="highlight-cl">Copying certs to t
e local machine directory...
</span></span><span class="highlight-line"><span class="highlight-cl">Copying certs to t
e remote machine...
</span></span><span class="highlight-line"><span class="highlight-cl">Setting Docker co
figuration on the remote daemon...
</span></span><span class="highlight-line"><span class="highlight-cl">Configuring swar
...
</span></span><span class="highlight-line"><span class="highlight-cl">Checking connect
on to Docker...
</span></span><span class="highlight-line"><span class="highlight-cl">Docker is up and
unning!
</span></span><span class="highlight-line"><span class="highlight-cl">To see how to co
nect your Docker Client to the Docker Engine running on this virtual machine, run: docker-ma
hine env swarm-master
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">james@james-C
65:~/.docker/machine/cache &gt; docker-machine ls
</span></span><span class="highlight-line"><span class="highlight-cl">NAME          ACTI
E   DRIVER      STATE     URL                     SWARM               DOCKER      ERRORS
</span></span><span class="highlight-line"><span class="highlight-cl">default       -
```

```
irtualbox   Running   tcp://192.168.99.100:2376                    v17.12.0-ce
</span></span><span class="highlight-line"><span class="highlight-cl">swarm-master   -
    virtualbox   Running   tcp://192.168.99.101:2376   swarm-master (master)   v17.12.1-ce
</span></span><span class="highlight-line"><span class="highlight-cl">//可以看的在swar
-master中运行了2个swarm容器，分别是join和manage
</span></span><span class="highlight-line"><span class="highlight-cl">docker@swarm-
aster:~$ docker ps
</span></span><span class="highlight-line"><span class="highlight-cl">CONTAINER ID
  IMAGE               COMMAND                  CREATED             STATUS              PORTS
       NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">85edb0c36a44
 swarm:latest        "/swarm join --adver…"   6 minutes ago       Up 6 minutes        2375/tcp
              swarm-agent
</span></span><span class="highlight-line"><span class="highlight-cl">8ef6de0b0ea1
swarm:latest        "/swarm manage --tls…"   6 minutes ago       Up 6 minutes        2375/tcp, 0.0
0.0:3376-&gt;3376/tcp   swarm-agent-master
</span></span></code></pre>
<ol start="2">
<li>创建一个工作节点</li>
</ol>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">james@james-CW65:~/.docker/machine/cache &gt; docker-machine create -d virtualbox
-swarm  --swarm-discovery token://6e56a11eb2ac3ef3a7e467a99fa683b5 swarm-agent-00
</span></span><span class="highlight-line"><span class="highlight-cl">Running pre-creat
 checks...
</span></span><span class="highlight-line"><span class="highlight-cl">Creating machine..

</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-agent-00)
Copying /home/james/.docker/machine/cache/boot2docker.iso to /home/james/.docker/ma
hine/machines/swarm-agent-00/boot2docker.iso...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-agent-00)
Creating VirtualBox VM...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-agent-00)
Creating SSH key...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-agent-00)
Starting the VM...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-agent-00)
Check network to re-create if needed...
</span></span><span class="highlight-line"><span class="highlight-cl">(swarm-agent-00)
Waiting for an IP...
</span></span><span class="highlight-line"><span class="highlight-cl">Waiting for machi
e to be running, this may take a few minutes...
</span></span><span class="highlight-line"><span class="highlight-cl">Detecting operati
g system of created instance...
</span></span><span class="highlight-line"><span class="highlight-cl">Waiting for SSH to
be available...
</span></span><span class="highlight-line"><span class="highlight-cl">Detecting the prov
sioner...
</span></span><span class="highlight-line"><span class="highlight-cl">Provisioning with
oot2docker...
</span></span><span class="highlight-line"><span class="highlight-cl">Copying certs to t
e local machine directory...
</span></span><span class="highlight-line"><span class="highlight-cl">Copying certs to t
e remote machine...
```

Setting Docker configuration on the remote daemon...
Configuring swarm...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env swarm-agent-00
james@james-C65:~/.docker/machine/cache > docker-machine ls
NAME ACTIVE DRIVER STATE URL SWARM DOCKER ERRORS
default - virtualbox Running tcp://192.168.99.100:2376 v17.12.0-ce
swarm-agent-00 virtualbox Running tcp://192.168.99.102:2376 swarm-master v17.12.1-ce
swarm-master - virtualbox Running tcp://192.168.99.101:2376 swarm-master (master) v17.12.1-ce
//
docker@swarm-agent-00:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2886d0aa3906 swarm:latest "/swarm join --adver…" About a minute ago Up About a minute 2375/tcp swarm-agent
</code></pre>

<ol start="3">
可以再添加几个工作节点。

<h4 id="3--管理swarn集群">3. 管理 swarn 集群</h4>

切换环境到 swarm-master：docker-machine env swarm-master
使用 docker info 获取集群信息

<pre><code class="highlight-chroma">由于使用了docker swarm模式，这里的输出被覆盖了。
</code></pre>

可以查看出每个节点都暴露了 2376 端口，每个节点都运行了 swarm agent 的容器。
不建议在生产环节中在 master 上运行 manager 和 agent，可能会导致 agent 失败引发问题。

<ol start="3">
检查运行在集群中的容器：

<pre><code class="highlight-chroma">docker@swarm-master:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
</code></pre>

```
        NAMES
</span></span><span class="highlight-line"><span class="highlight-cl">85edb0c36a44
 swarm:latest        "/swarm join --adver…"    33 minutes ago      Up 33 minutes       2375/tcp
                    swarm-agent
</span></span><span class="highlight-line"><span class="highlight-cl">8ef6de0b0ea1
swarm:latest        "/swarm manage --tls…"    34 minutes ago      Up 34 minutes       2375/tcp, 0.
.0.0:3376-&gt;3376/tcp    swarm-agent-master
</span></span></code></pre>
```