



Gestión de Contenedores - Astillero

Por Ivan Krizsan | 10 de julio de 2016

0 Comentario

Contenido [[show](#)]

Estoy buscando algo que me ayude a administrar varios hosts Docker y las aplicaciones que se ejecutan en esos hosts. Una vez más, voy a utilizar mi [laboratorio virtual](#) , para empezar a explorar las alternativas. En este primer artículo voy a ver [Shipyard](#) , que es una aplicación de [código abierto](#) para la gestión de un cluster Docker.

Lista de deseos

Antes de mirar cualquier cosa creo que debo enumerar algunas de las cualidades que quiero en el tipo de aplicación de gestión que estoy buscando.

Recuerde que estas son mis preferencias personales y nada más.

1. Capaz de manejar múltiples hosts Docker.
2. Requiere un mínimo de software adicional en un host que no contenga el software de administración.
Obviamente Docker necesita ser instalado en un nodo, pero cuanto menos software adicional requiera mejor en mi opinión.
3. Proporcionar una API para la administración.
No veo una necesidad inmediata, pero sospecho que en algún momento en el tiempo voy a querer automatizar las tareas.
4. Proporcionar una interfaz gráfica de usuario para la gestión.
Mientras que una interfaz de línea de comandos está bien y dandy, una buena interfaz gráfica de usuario hace más fácil comprender lo que está pasando y hará más fácil de aprender.
Además, una buena GUI será capaz de dar una visión general, así como centrarse en los detalles cuando sea necesario.
5. Gestión automática de aplicaciones / contenedores.
Ejemplos: Escalar una aplicación dependiendo de su carga. Retroceso automático en caso de fallo al implementar una nueva versión de un contenedor.
6. Facilitar la gestión de aplicaciones que consisten en múltiples contenedores Docker.
Lo ideal sería que sería capaz de agrupar los contenedores no sólo dependiendo de cómo se van a desplegar, escalado, etc, que es el tipo regular de agrupación logrado con Docker Compose, sino también los contenedores de grupo en función de metadatos en el contenedor.

7. Gratis, opcionalmente con soporte comercial.

Gratis para que yo pueda probar todas las características y desarrollar sin tener que usar una versión de prueba o similar. Apoyo comercial porque da a gerentes y jefes una sensación de seguridad cuando hay una factura a pagar, en caso de que quiera introducirla en una organización de clientes.

8. Control de acceso.

No quiero que nadie pueda hacer cambios en mis aplicaciones, así que me gustaría tener algún tipo de control de acceso donde pudiera limitar el acceso de otras personas a las aplicaciones que configuro. Sólo las cosas tradicionales con los usuarios, grupos y, opcional pero muy agradable tener, un registro de auditoría.

9. Disponibilidad.

¿Qué sucede si un nodo del clúster o un contenedor se desconecta, ya sea un nodo de administrador o un nodo de trabajo? Si se trata de un nodo gestor, debe haber algún tipo de mecanismo de conmutación por error para que otro nodo asuma las responsabilidades del gestor. Si un contenedor se desconecta, debería iniciarse una nueva instancia de la misma.

10. Documentación.

Buena documentación es siempre agradable.

11. Facilidad de instalación.

Sospecho que habrá adiciones a esta lista mientras el viaje se desenreda y volveré a hacer cambios.

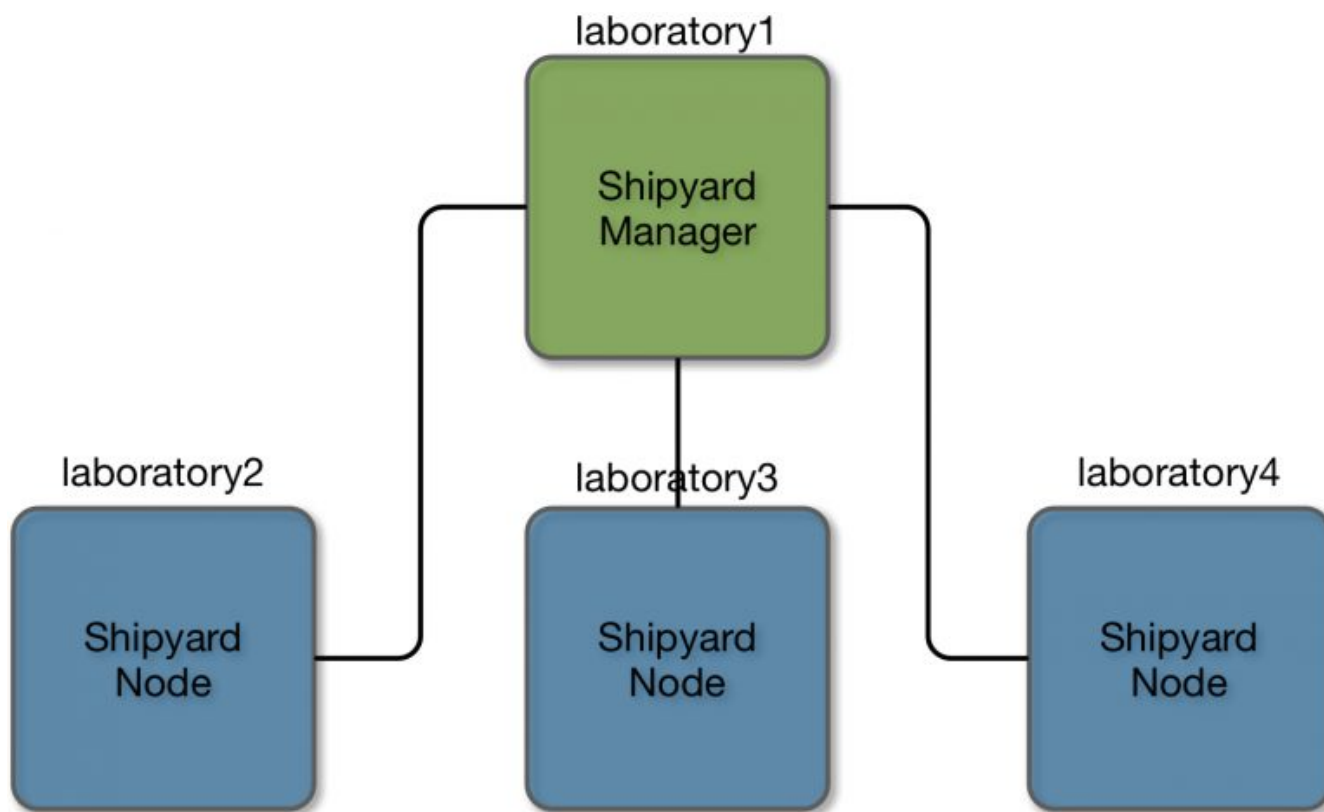
Soy consciente de Docker Swarm e incluso de sus [mejoras en Docker 1.12](#), pero mi deseo es encontrar algo más.

Astillero

Como antes, este primer post se centra en [Shipyard](#), que es una aplicación para gestionar imágenes y contenedores de Docker. Shipyard utiliza Docker Swarm en su núcleo y tiene una GUI en forma de una aplicación web.

Instalación

En primer lugar estableceré astillero en mi laboratorio virtual que, como antes, consiste en cuatro máquinas virtuales que ejecutan Ubuntu 16.04 con un gestor y tres nodos.



Desplegando Astillero en mi laboratorio virtual.

Como se muestra en la figura anterior, ejecutaré el gerente del astillero en el nodo laboratorio1 y los nodos laboratorio2, laboratorio3 y laboratorio4 serán nodos astilleros regulares. Utilizaré estos nombres de nodos en todo el ejemplo.

Instalación del administrador

Para que sea fácil para mí durante toda la instalación, prueba y reinstalación, he creado un script que instala e inicia el administrador del astillero. Realice los siguientes pasos en el laboratorio1:

- Cree un directorio llamado "astillero".

```
1 mkdir shipyard
2 cd shipyard
```

- Cree un archivo llamado "start-manager.sh" con el siguiente contenido:

```
1 #!/bin/sh
2 # Must have root to run Shipyard.
3 if [[ $EUID -ne 0 ]]
4 then
5     echo "This script must be run as root."
6     exit 1
7 fi
8
9 # Download Shipyard deployment script if it isn't already downloaded.
10 if [ ! -f deploy ]
11 then
12     wget https://shipyard-project.com/deploy
13     chmod +x deploy
14 fi
15
16 # Start Shipyard manager.
```

```
17 ACTION=deploy
18 . deploy
```

- Hacer que el script de inicio-administrador sea ejecutable.

```
1 chmod +x start-manager.sh
```

- Inicie el administrador del astillero.

```
1 sudo su
2 # Enter password.
3 ./start-manager.sh
```

La salida del registro debe terminar con algo a lo largo de estas líneas:

```
1 Waiting for Shipyard on 192.168.1.180:8080
2
3 Shipyard available at http://192.168.1.180:8080
4 Username: admin Password: shipyard
```

- En un navegador, abra la URL mostrada en el registro e inicie sesión usando el nombre de usuario y la contraseña que también se muestran en el registro.

Debería ver la lista de contenedores que en mi caso contiene siete contenedores.

shipyard								
CONTAINERS								
IMAGES								
NODES								
REGISTRIES								
ACCOUNTS								
EVENTS								
ADMIN								
Refresh								
Deploy Container								
Search containers...								
		Id	Node	Name	Image	Status	Created	Actions
<input type="checkbox"/>		349c1f27a45e	laboratory1	shipyard-controller	shipyard/shipyard:latest	Up About a minute	2016-06-29 20:26:55 +0200	
<input type="checkbox"/>		19977c4323e1	laboratory1	shipyard-swarm-agent	swarm:latest	Up About a minute	2016-06-29 20:26:30 +0200	
<input type="checkbox"/>		e563159bb037	laboratory1	shipyard-swarm-manager	swarm:latest	Up About a minute	2016-06-29 20:26:30 +0200	
<input type="checkbox"/>		05085154634a	laboratory1	shipyard-proxy	shipyard/docker-proxy:latest	Up 2 minutes	2016-06-29 20:26:15 +0200	
<input type="checkbox"/>		25f8c92f10a9	laboratory1	shipyard-certs	alpine	Up 2 minutes	2016-06-29 20:26:06 +0200	
<input type="checkbox"/>		8102f37c016f	laboratory1	shipyard-discovery	microbox/etcd:latest	Up 2 minutes	2016-06-29 20:26:06 +0200	
<input type="checkbox"/>		1558913f22ca	laboratory1	shipyard-rethinkdb	rethinkdb	Up 2 minutes	2016-06-29 20:25:55 +0200	

Lista de contenedores en la aplicación del administrador del astillero.

- Haga clic en la pestaña Nodos.

A continuación, debe ver la lista de nodos en el grupo del astillero.

shipyard					
CONTAINERS					
IMAGES					
NODES					
REGISTRIES					
ACCOUNTS					
EVENTS					
ADMIN					
Refresh					
Search nodes...					
Name	Address	Containers	Reserved CPUs	Reserved Memory	Labels
laboratory1	192.168.1.180:2375	7	0 / 2	0 B / 5.086 GiB	executiondriver=, kernelversion=4.4.0-28-generic, operatingsystem=Ubuntu 16.04 LTS, storagedriver=aufs

Lista de nodos en la aplicación Administrador del astillero.

Ahora hemos instalado y puesto en marcha correctamente el administrador del astillero. Le sugiero que mantenga abierta la aplicación del administrador del astillero, mostrando la lista de nodos, ya que facilitará la confirmación de la adición exitosa de un nodo al clúster del astillero.

Instalación del nodo

El siguiente procedimiento para instalar e iniciar un nodo debe repetirse para el número de máquinas virtuales adicionales en su biblioteca - en mi caso tres.

- Cree un directorio llamado "astillero".

```
1 mkdir shipyard
2 cd shipyard
```

- Cree un archivo llamado "start-node.sh" con los siguientes contenidos:

Tenga en cuenta que la dirección IP de la línea resaltada debe modificarse a la dirección IP del equipo (virtual) donde se ejecuta su administrador del astillero.

```
1 #!/bin/bash
2 # Exit upon errors.
3 set -e
4
5 # Must have root to run Shipyard.
6 if [[ $EUID -ne 0 ]]
7 then
8     echo "This script must be run as root."
9     exit 1
10 fi
11
12 # Download Shipyard deployment script if it isn't already downloaded.
13 if [ ! -f deploy ]
14 then
15     wget https://shipyard-project.com/deploy
16     chmod +x deploy
17 fi
18
19 # Start Shipyard node.
20 ACTION=node
21 # IMPORTANT! Change this IP address to the IP address where your Shipyard manager runs
22 DISCOVERY=etcd://192.168.1.180:4001
23 . deploy
```

- Hacer que la secuencia de comandos de nodo de inicio sea ejecutable.

```
1 chmod +x start-node.sh
```

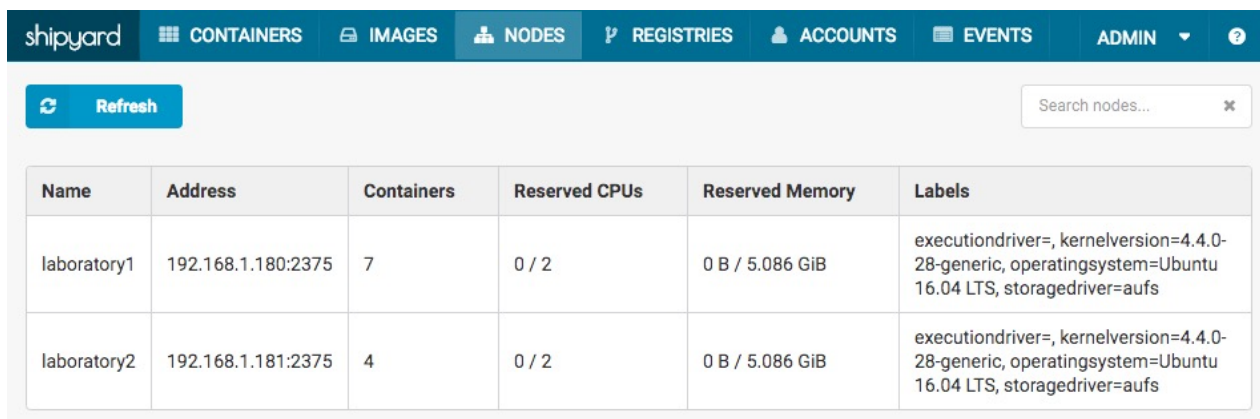
- Inicie el nodo Shipyard.

```
1 sudo su
2 # Enter password.
3 ./start-node.sh
```

La salida de registro debe terminar con algo similar a esto:

```
1 Status: Downloaded newer image for swarm:latest
2 -> Starting Swarm Agent
3 Node added to Swarm: 192.168.1.181
```

- En la pestaña Nodos de la aplicación Administrador del astillero, haga clic en el botón Actualizar. El nuevo nodo debe aparecer en la lista de nodos.



Name	Address	Containers	Reserved CPUs	Reserved Memory	Labels
laboratory1	192.168.1.180:2375	7	0 / 2	0 B / 5.086 GiB	executiondriver=, kernelversion=4.4.0-28-generic, operatingsystem=Ubuntu 16.04 LTS, storagedriver=aufs
laboratory2	192.168.1.181:2375	4	0 / 2	0 B / 5.086 GiB	executiondriver=, kernelversion=4.4.0-28-generic, operatingsystem=Ubuntu 16.04 LTS, storagedriver=aufs

Lista de nodos en la aplicación Administrador del astillero con un nodo recién agregado.

Repita este proceso hasta que haya agregado el número deseado de nodos a su clúster del astillero.

Examinando Astillero

Antes de examinar qué tan bien astillero coincide con mis necesidades, voy a enumerar muy brevemente algunas de las características de Shipyard.

- Gestión de contenedores Docker.
Iniciar, detener, pausar, retirar contenedores. Examinar estadísticas del contenedor (CPU, memoria, red). Examine los registros. Ejecute comandos en una "ventana de terminal" en la aplicación web.
- Gestión de imágenes Docker.
Tire de las imágenes. Eliminar imágenes.
- Gestión de nodos.
Como ya hemos visto, puede listar los nodos actuales en el enjambre.
- Gestión de registros de Docker.
Agregar y eliminar registros.
- Gestión de usuarios.
Agregar y eliminar usuarios. Asignar derechos de acceso a los usuarios.
- Ver eventos.
Enumera todas las operaciones realizadas en Astillero.
- Reinicio automático.
Los contenedores del astillero se inician automáticamente cuando vuelvo a arrancar una computadora en la que el astillero (administrador o nodo) se estaba ejecutando al apagarse.

Así que, ¿qué tan bien el astillero está a la altura de los deseos que he enumerado anteriormente?

Administrar varios hosts Docker

Comprobar. No hay mucho que discutir aquí.

Requiere un mínimo de software adicional

Los siguientes contenedores de Docker se ejecutan en un nodo que no es administrador:

1	CONTAINER ID	IMAGE	COMMAND	CREATED
2	65d5af34868f	swarm:latest	"/swarm j --addr 192."	11 minutes ago
3	c1e02c99694f	swarm:latest	"/swarm m --replicati"	11 minutes ago
4	1bbbf393c4dd	shipyard/docker-proxy:latest	"/usr/local/bin/run"	12 minutes ago
5	5e1a1fa921fe	alpine	"sh"	12 minutes ago

Al listar las imágenes de Docker necesarias para ejecutar un nodo que no sea administrador, podemos ver que son todas muy pequeñas.

1	REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
2	alpine	latest	4e38e38c8ce0	10 days ago	4.799
3	swarm	latest	a180b24e38ed	3 weeks ago	19.34
4	shipyard/docker-proxy	latest	cfee14e5d6f2	6 months ago	9.468

Observando los contenedores que se ejecutan en un nodo gestor:

1	CONTAINER ID	IMAGE	COMMAND	CREATED
2	349c1f27a45e	shipyard/shipyard:latest	"/bin/controller --de"	4 days ago
3	19977c4323e1	swarm:latest	"/swarm j --addr 192."	4 days ago
4	e563159bb037	swarm:latest	"/swarm m --replicati"	4 days ago
5	05085154634a	shipyard/docker-proxy:latest	"/usr/local/bin/run"	4 days ago
6	25f8c92f10a9	alpine	"sh"	4 days ago
7	8102f37c016f	microbox/etcd:latest	"/bin/etcd -addr 192."	4 days ago
8	1558913f22ca	rethinkdb	"rethinkdb --bind all"	4 days ago

Echa un vistazo a las imágenes de Docker en el nodo administrador:

1	REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
2	alpine	latest	4e38e38c8ce0	10 days ago	4.799
3	swarm	latest	a180b24e38ed	3 weeks ago	19.34
4	rethinkdb	latest	70aa872c4c97	3 weeks ago	184.2
5	shipyard/shipyard	latest	ab940e847950	4 weeks ago	58.92
6	shipyard/docker-proxy	latest	cfee14e5d6f2	6 months ago	9.468
7	ehazlett/curl	latest	de1763b15312	10 months ago	8.751
8	microbox/etcd	latest	6aef84b9ec5a	11 months ago	17.87

Estoy menos preocupado por la cantidad de software adicional necesario para ejecutar un nodo administrador. Por supuesto, quiero tener varios nodos de gestión para la conmutación por error en un entorno de producción, pero sin embargo, sospecho que habrá un número fijo de nodos de gestión.

Con Shipyard, todavía se necesita algún software adicional para los nodos que no son gestores aunque parezca un mínimo.

Tanto el gestor como los nodos no gestores utilizan la imagen de Docker Swarm, que va a ser obsoleta una vez que se libere Docker 1.12.

Conclusión:

Compruebe. Shipyard tiene, en mi opinión, una cantidad razonable de software adicional requerido para ejecutar nodos no gestores.

API para la gestión

Comprobar.

Además de la funcionalidad proporcionada por la aplicación web Shipyard, la API también admite:

Política de Privacidad y Cookies

- Acceso a la [API de Docker Remote](#) .

En el momento de escribir este informe, Shipyard admite hasta la versión 1.20 de la API Docker Remote.

Consulte el [código del controlador del astillero](#) para obtener más detalles.

- Gestión de las claves webhook utilizadas para la implementación en [Docker Hub](#) .

La documentación del Shipyard API está disponible [aquí](#) .

GUI para la gestión

Ya hemos visto vislumbres de la GUI del astillero anteriormente en este artículo, así que: Compruebe.

Hay funcionalidad de filtrado simple disponible en las diferentes páginas de la aplicación web. Por ejemplo, en la página que lista todos los contenedores en ejecución, puede ingresar algún texto y sólo los contenedores que contengan entradas en la lista, aunque en cualquier columna, aparecerán en la lista.

La GUI es un poco simplista, pero eso no puede ser una mala cosa. Con cuatro nodos en mi enjambre, no tengo problemas para encontrar mi camino alrededor de los diferentes nodos. No he intentado 400 nodos todavía, así que no puedo comentar sobre eso.

Gestión automática de aplicaciones / contenedores

Aparte de la funcionalidad básica de Docker para configurar una política de reinicio al crear un nuevo contenedor de Docker, no pude encontrar ningún soporte para la administración automática de aplicaciones / contenedores.

Gestión de Aplicaciones que Consisten de Contenedores Múltiples

La gestión de aplicaciones que consisten en múltiples contenedores, estilo Docker Compose, no está disponible en Shipyard en este momento. Hay una [solicitud para esta función en el proyecto Shipyard GitHub](#) .

Opción gratuita con soporte

Astillero es libre, por lo que he podido decir, pero lamentablemente no parece haber ninguna opción de soporte disponible. Daré a Shipyard un cheque parcial.

Control de acceso

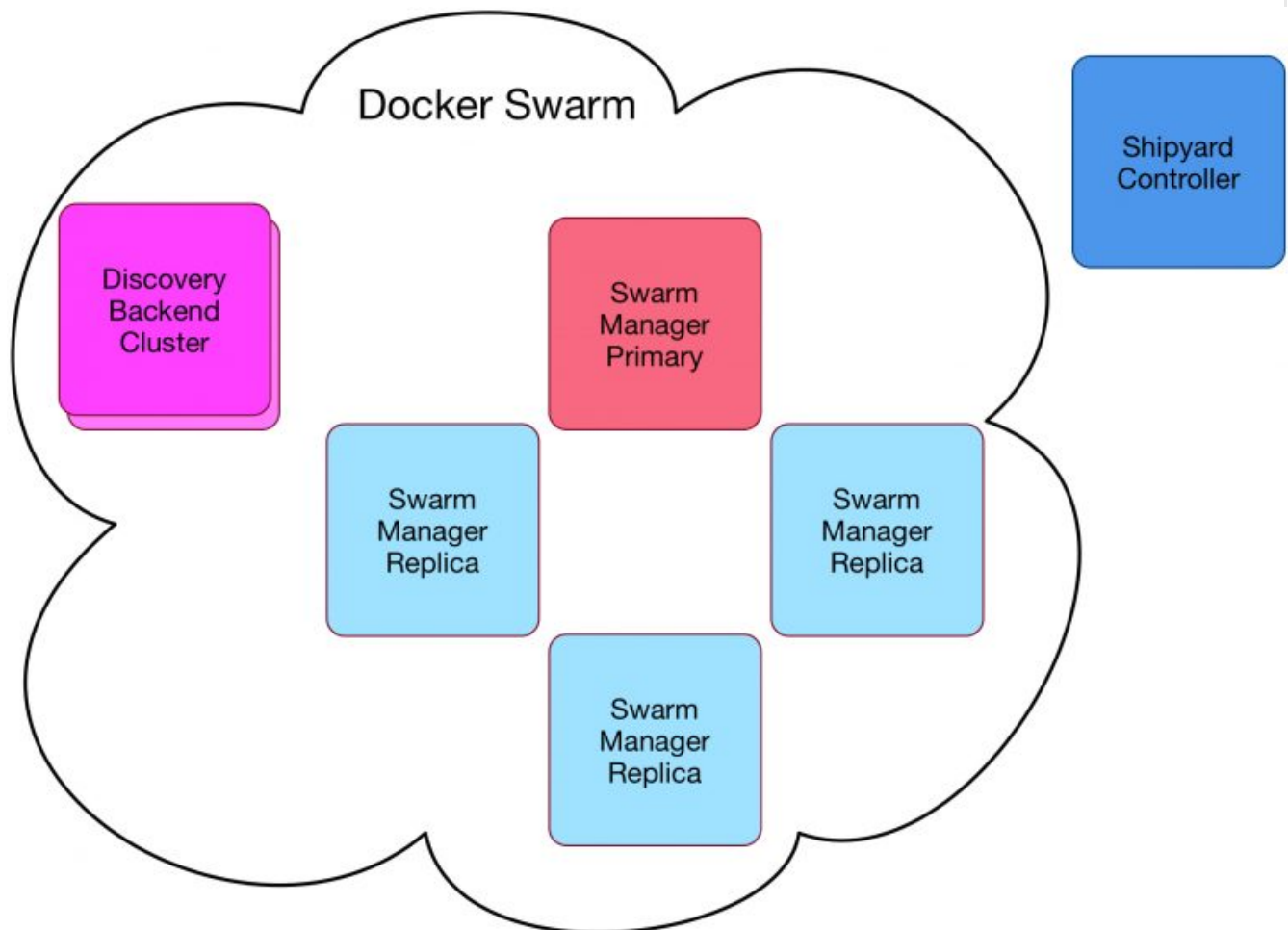
Como se mencionó anteriormente, Shipyard implementa la gestión de usuarios con control de acceso. La documentación incluso reclama soporte para LDAP, aunque no he probado esto.

La lista de eventos en el astillero también se puede utilizar como registro de auditoría, para lo cual quiero dar a Shipyard créditos adicionales. Así que le doy a Shipyard un doble control, en lo que respecta al control de acceso.

Availability

Shipyard acts as a front-end for Docker Swarm, which has support for high availability, and thus does not need to implement these features itself. For those not familiar with Docker Swarm, I will briefly mention that it allows for failover both as far as managers and also as far as containers are concerned. I'll give Shipyard a check.

If I were to set up a cluster of computers on which to run Docker containers and use Shipyard as the manager front-end, I would first set up a properly configured Docker Swarm and then add the Shipyard controller.



Highly available Docker Swarm cluster with a Shipyard controller.

There is a section in the Shipyard documentation describing how to [manually deploy Shipyard](#) if you want to attempt something like this.

Documentation

Los desarrolladores del astillero ha hecho un esfuerzo, en lo que respecta a la documentación. Hay documentación no sólo para la interfaz gráfica del astillero, sino también para la API. Aunque cuando probé el API tuve que echar un vistazo al controlador REST antes de hacerlo bien. ¡Comprobar!

Facilidad de instalación

Como se ve en la sección sobre la instalación anterior, astillero es bastante fácil de instalar - sólo ejecute los scripts con algunos parámetros. Hay características adicionales en el script de instalación del astillero, como

una opción de desinstalación.

Conclusión

Shipyard parece una solución de gestor de clúster Docker agradable y ligera, construida en la parte superior de Docker Swarm. Tiene la mayoría de las características que estoy buscando, pero nada que, en mi opinión, se destaca. El depósito de GitHub del astillero tiene, en el momento de escribir, alrededor de 40 contribuyentes con dos contribuyentes principales.

Desde Shipyard es el primer producto que miro en esta área, no puedo compararlo realmente a cualquier otra cosa.

Aquí hay una tabla que resume mi examen de Astillero:

Lista de artículos	Estado
1. Administrar varios hosts Docker.	Sí
2. Mínimo de software adicional.	Sí
3. API de gestión.	Sí
4. GUI de administración.	Sí
5. Gestión automática de la aplicación / contenedor.	No
6. Administrar grupos de contenedores.	No
7. Libre, con opción para el apoyo comercial.	Gratis: Sí. Apoyo comercial: No
8. Control de acceso.	Sí, con registro de auditoría.
9. Disponibilidad.	Sí, implementado por Docker Swarm.
10. Documentación.	Sí
11. Facilidad de instalación	Sí

Dado que Docker Swarm se incorporará en el Docker Engine 1.12, esperaré el soporte de esta versión en Shipyard hasta que lo instale para nada más que para la experimentación.

Codificación feliz!

Categoría: Operaciones Etiquetas: gestión de contenedores , muelle , astillero