

What's new in Docker 1.13?

Michael Irwin - February 8, 2017

Restructured cli

- Number of commands at top level was becoming quite large
- Effort being made to better group and organize commands

```
docker images -> docker image list
docker rmi      -> docker image rm
docker create  -> docker container create
docker ps      -> docker container ls
```

- Use `DOCKER_HIDE_LEGACY_COMMANDS=1` to only show new command structure
- All legacy commands are still supported in 1.13. Not sure when removed yet...

Experimental built-in

- Experimental builds use to be separate
- Made it hard to play with experimental features
- Now, add `--experimental` to dockerd settings
- Setting available directly from Docker for Mac/Windows GUI
- Find out if experimental is enabled using `docker system info`

```
> docker system info
...
Experimental: true
...
```

CLI Backwards Compatibility

Ever get hit with this?

```
> docker ps  
Error response from daemon: client is newer than server
```

- Docker for Mac/Windows are always using the latest client. Servers may not be. Commands would then cause the error.

CLI Backwards Compatibility

Ever get hit with this?

```
> docker ps  
Error response from daemon: client is newer than server
```

- Docker for Mac/Windows are always using the latest client. Servers may not be. Commands would then cause the error.
- No longer an issue!
- CLI round-trips with daemon to determine version and available commands

Swarm Encrypts at Rest

- Docker Swarm's RAFT database keeps data encrypted at rest with decryption keys on filesystem
- With new `--auto-lock` flag, user has to provide key to unlock manager

```
$ docker swarm init --autolock
```

Swarm initialized: current node (mhlh7vx93t1i0digtdenvqjx1) is now a

To add a worker to this swarm, run the following **command**:

```
docker swarm join \  
--token SWMTKN-1-32lrgk2aa5ubngy1u3tb85j3nh9gy1-072qmokabz6cqma5  
172.31.10.227:2377
```

To add a manager to this swarm, run '**docker swarm join-token manager**

To unlock a swarm manager after it restarts, run the **docker swarm un**
command and provide the following key:

```
SWMKEY-1-CV8QI2JwrGwpGcP9m/cQ+lteVIX12ZjvWV3ol2reNMU
```

Please remember to store this key **in** a password manager, since without it you will not be able to restart the manager.

...

```
sudo shutdown -r now
```

```
docker swarm unlock < ./swarm-key
```

Compose to Swarm

- New V3 Compose syntax released
 - Removes non-portable options (volume-driver, volume-from, and a few others)
 - Added Swarm options (replicas, mode, etc.)

```
docker stack deploy --compose-file=docker-compose-stack.yml my-app
docker stack list
docker stack rm my-app
```


Sample V3 Compose file

```
version: '3'

services:
  app:
    image: mikesir87/cats:1.0
    ports:
      - 5000:5000
    deploy:
      replicas: 2
      update_config:
        parallelism: 1
      delay: 10s
```

Data Management Commands

- New subcommands under `docker system`...
- `docker system df` will show Docker's disk usage

```
bash-3.2$ docker system df
```

TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	120	17	10.27 GB	5.131 GB (49%)
Containers	35	4	149.3 MB	9.963 MB (6%)
Local Volumes	6	4	295.1 MB	4.98 kB (0%)

- `docker system prune` will remove all inactive images/containers/volumes
 - Inactive images have no containers based from image
 - Inactive containers are non-running containers
 - Inactive volumes are unused volumes

Secret Management

- Ability to create secrets for Swarm cluster

```
docker secret create app-properties app.properties  
curl http://example.com/properties | docker secret create app-pr
```

- Add a secret to a Swarm cluster...

```
docker service create --secret app-properties --name=app -p 80:8
```

- Secrets are available in `/run/secrets/`
 - This example would have a file named `/run/secrets/app-properties`

Other Orchestration Updates

- Pin image by digest
 - When running image by tag, Swarm uses digest to ensure all nodes running same version
- Service updates can now rollback, instead of continue or abort
- Added failure thresholds to rollouts

Experimental features

- `docker build --squash` - produces only one layer for a build
- `docker service logs` - aggregates logs for a service across all nodes

That's it!