Title: Recommended way to run a Docker Compose stack in production?

Post Body:

I have a couple of compose files (docker-compose.yml) describing a simple Django application (five containers, three images).

I want to run this stack in production - to have the whole stack begin on boot, and for containers to restart or be recreated if they crash. There aren't any volumes I care about and the containers won't hold any important state and can be recycled at will.

I haven't found much information on using specifically docker-compose in production in such a way. The documentation is helpful but doesn't mention anything about starting on boot, and I am using Amazon Linux so don't (currently) have access to Docker Machine. I'm used to using supervisord to babysit processes and ensure they start on boot up, but I don't think this is the way to do it with Docker containers, as they end up being ultimately supervised by the Docker daemon?

As a simple start I am thinking to just put `restart: always` on all my services and make an init script to do `docker-compose up -d` on boot. Is there a recommended way to manage a docker-compose stack in production in a robust way?

EDIT: I'm looking for a 'simple' way to run the equivalent of `docker-compose up` for my container stack in a robust way. I know upfront that all the containers declared in the stack can reside on the same machine; in this case I don't have need to orchestrate containers from the same stack across multiple instances, but that would be helpful to know as well.

Accepted Answer:

Compose is a client tool, but when you run `docker-compose up -d` all the container options are sent to the Engine and stored. If you specify `restart` as `always` (or preferably `unless-stopped` to give you more flexibility) then you don't need run `docker-compose up` every time your host boots.

When the host starts, provided you have configured the Docker daemon to start on boot, Docker will start all the containers that are flagged to be restarted. So you only need to run `docker-compose up -d` **once** and Docker takes care of the rest.

As to orchestrating containers across multiple nodes in a Swarm - the preferred approach will be to use Distributed Application Bundles, but that's currently (as of Docker 1.12) experimental. You'll basically create a bundle from a local Compose file which represents your distributed system, and then deploy that remotely to a Swarm. Docker moves fast, so I would expect that functionality to be available soon.

Highest Rated Answer:

You can find in their documentation more information about using docker-compose in production. But, as they mention, `compose` is primarily aimed at development and testing environments.

If you want to use your containers in production, I would suggest you to use a suitable tool to orchestrate containers, as Kubernetes.