



Docker Compose CLI for ECS

14 April 2021 - AWS User Group Greece

Who am I?

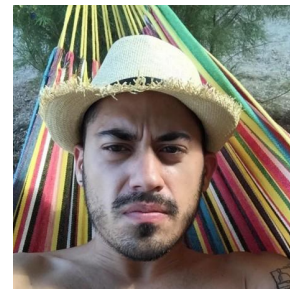
Paris Kasidiaris // [@pariskasid](https://twitter.com/pariskasid)

Founder at LOGIC – <https://withlogic.co/>

Co-founder at SourceLair – <https://www.sourcelair.com>

Co-host of Mikri Kouventa Podcast – <http://mikrikouventa.fm>

Co-organizer of Docker Athens User Group – <http://docker.gr>



Intro to Docker Compose CLI



Docker Compose CLI

- A new **uniform** CLI to manage applications
- Describe your application stack in a simple industry standard file
- Develop, build and publish whole application stacks



docker compose vs docker-compose

- Subcommand of docker
- Written in Go
- Multiple backends (local, ECS, ACI, Kube)
- Completely separate CLI tool
- Written in Python
- Local backend only



Why Docker Compose CLI

- Straightforward cloud native application development
- Use industry standard Compose specification
- Use the same commands local development on cloud providers (AWS) as well

Develop a web application with Docker Compose CLI



Prerequisites

- Latest Docker Desktop on macOS or Windows
- Latest Compose CLI on Linux
- Your editor of choice



Develop a web application with Docker Compose CLI

1. Create a Dockerfile for our app
2. Create docker-compose.yml baseline
3. Create docker-compose.override.yml for development
4. Write some code



Web application specs

- Python 3.8
- Flask
- Return hostname of process
- Public Docker image on Docker Hub

Let's get our hands dirty.



Docker Compose CLI for ECS



Docker Compose CLI for ECS

- Deploy to ECS using the same Docker Compose configuration
- Built in load balancing
- Built in scaling and auto scaling
- Built in secret management
- Built in service discovery

Deploy on ECS via Docker Compose CLI



Prerequisites

- An active AWS account
- AWS credentials configured locally
- Latest Docker Desktop installed on macOS or Windows
- Latest Compose CLI installed on Linux



Under the hood

Compose CLI is a sophisticated **client** that transforms a Compose Spec file into AWS resources.

Eventually everything gets deployed as a **CloudFormation stack**. There is no black box.

If you have to, you can also **access and modify** the CloudFormation stack before deploying.



Deploy on ECS via Docker Compose CLI

1. Create an ECS context in Docker client
2. Create a `docker-compose.ecs.yml` for ECS sprinkles
3. Run `docker compose up`

Really... that's it.



Scale on ECS via Docker Compose CLI

- Use `deploy.resources.limits` in Compose file to scale container size
- Use `deploy.replicas` in Compose file to scale container replicas



Auto scale on ECS via Docker Compose CLI

- Docker Compose CLI supports CPU-based container replica auto scaling on ECS
- Uses the `x-aws-autoscaling` vendor attribute in Compose File



Known issues



Known issues

- [#1214](#) – Compose CLI likes `us-east-1` a bit too much
- [#670](#) – No `docker compose exec` yet

That's all folks!



A few useful links

- <https://docs.docker.com/cloud/ecs-integration/>
- <https://github.com/docker/compose-cli>
- <https://github.com/compose-spec/compose-spec>
- <https://docs.docker.com/compose/compose-file/compose-file-v3/>

Questions?



We can help with DevOps!

Do you plan to double down on Docker, Infrastructure and all the sweet DevOps?

Do you want to build your next project on lean and automated processes?

We are here to help: paris@withlogic.co



Thanks!

Find the code at <https://github.com/parisk/aws-user-group-greece-compose-cli>

Follow me at <https://twitter.com/pariskasid>