



## UMD DATA605 - Big Data Systems

### Lesson 3.3: Docker Compose

**Instructor:** Dr. GP Saggese, [gsaggese@umd.edu](mailto:gsaggese@umd.edu)

# Docker Compose

---

- **Manage multi-container apps running on a single node**
  - Describe app in a single *declarative* YAML file
    - Avoid long Docker command scripts
  - Compose interacts with Docker API to achieve the requested state
  - Examples:
    - Client app and Postgres DB
    - Microservices: Web front-end, Ordering, Back-end DB
  - In 2020, Docker Compose became an open standard for “code-to-cloud” process
- **Manage multi-container apps running on multiple hosts**
  - Docker Stacks / Swarm
  - Kubernetes

# Docker Compose: Tutorial Example

- Default name for a Compose file is `docker-compose.yml`
  - Specify `-f` for custom filenames
- **Top-level keys:**
  - `version:`
    - Mandatory first line for API version
    - Use latest version, typically 3 or higher
  - `services:`
    - Define microservices
  - `networks:`
    - Create new networks
    - Default bridge network connects containers on the same Docker host
  - `volumes:`
    - Create new volumes
- **Key in services** describes a “service” in terms of container
  - **Inner keys** specify Docker run command params

```
version: "3.8"
services:
  web-fe:
    build: .
    command: python app.py
    ports:
      - target: 5000
        published: 5001
    networks:
      - counter-net
    volumes:
      - type: volume
        source: counter-vol
        target: /code
  redis:
    image: "redis:alpine"
    networks:
      counter-net:
networks:
  counter-net:
volumes:
  counter-vol:
```

# Docker Compose: Commands

---

```
> docker compose --help
```

```
Usage: docker compose [OPTIONS] COMMAND
```

```
Options:
```

--env-file string	Specify an alternate environment file.
-f, --file stringArray	Compose configuration files
-p, --project-name string	Project name

```
Commands:
```

build	Build or rebuild services
convert	Converts the compose file to platform's canonical format
cp	Copy files/folders between a service container and the local filesystem
create	Creates containers for a service.
down	Stop and remove containers, networks
events	Receive real time events from containers.
exec	Execute a command in a running container.
images	List images used by the created containers
kill	Force stop service containers.
logs	View output from containers
ls	List running compose projects
pause	Pause services
port	Print the public port for a port binding.
ps	List containers
pull	Pull service images
push	Push service images
restart	Restart containers
rm	Removes stopped service containers
run	Run a one-off command on a service.
start	Start services
stop	Stop services
top	Display the running processes
unpause	Unpause services
up	Create and start containers
version	Show the Docker Compose version information

# Docker Compose: Commands

---

- Build the containers for the services  
`> docker compose build`
- Pull the needed images for the services  
`> docker compose pull`
- Show the running services  
`> docker compose ps`
- Show the status of the service  
`> docker compose ls`
- Bring up the entire service  
`> docker compose up`
- Rebuild after trying out some changes in dockerfile/compose file  
`> docker-compose up --build --force-recreate`

# Docker Compose: Commands

- Show the processes inside each container

```
> docker compose top
```

```
counter_app-redis-1
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
999	49590	49549	0	10:40	?	00:00:02	redis-server *:6379

```
counter_app-web-fe-1
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	49614	49574	0	10:40	?	00:00:00	python app.py
root	49734	49614	1	10:40	?	00:00:08	/usr/local/bin/python /

# Docker Compose: Commands

---

- Build the containers for the services

```
> docker compose down
[+] Running 3/2
Container counter_app-redis-1    Removed
Container counter_app-web-fe-1   Removed
Network counter_app_counter-net  Removed
```

- Shutdown service removing the volume (i.e., resetting state)

```
> docker-compose down -v
```

- Shutdown service removing images and volume

```
> docker-compose down -v --rmi all
```

# Docker Compose: Tutorial

---

- Example taken from <https://github.com/nigelpoulton/counter-app>
- [tutorial\\_docker\\_compose](#)

```
> cd tutorials/tutorial_docker_compose  
> vi tutorial_docker_compose.md
```