Professional Docker Compose Reference

Configuration: `configs/docker-compose-starter.yml`

Command Synopsis

Command

```
docker compose -f configs/docker-compose-starter.yml up -d zookeeper kafka
↪ mysql connect-ui
```

1. Objective

Initialize and run a targeted subset of services defined in a custom Docker Compose file, executing them in detached (background) mode. Ideal for bootstrapping a streaming data pipeline environment: ZooKeeper, Kafka, MySQL, and Kafka Connect UI.

2. Component Breakdown

`docker compose`
: Invokes Docker Compose v2, the orchestration tool for multi-container applications.

`-f <file>`
: Specifies the path to the Compose file: `configs/docker-compose-starter.yml`.

`up`
: Builds (if needed), recreates, and starts the defined containers.

`-d`
: Detached mode: runs containers in the background.

`<services>`
: Explicit list of services to start: `zookeeper`, `kafka`, `mysql`, `connect-ui`.

Service	Role in Architecture
`zookeeper`	Coordination and metadata management for Kafka cluster.
`kafka`	Distributed event streaming broker for high-throughput data.
`mysql`	Relational database acting as source or sink in data pipelines.
`connect-ui`	Web-based interface to configure monitor Kafka Connect.

3. Service Roles

4. Prerequisites

- Docker Engine and Docker Compose v2+ installed and running.
- Compose file `configs/docker-compose-starter.yml` present and valid.
- Any referenced volumes, networks, and environment variables accessible.

5. Typical Workflow

1. Verify installations:
   - `docker --version`
   - `docker compose version`
2. Navigate to project root (or use absolute path).
3. Execute:

```
docker compose -f configs/docker-compose-starter.yml up -d zookeeper
↪ kafka mysql connect-ui
```

4. Monitor logs if necessary:

```
docker compose logs -f kafka connect-ui
```

5. Access Kafka Connect UI at the port defined in the Compose file.