

Fragments

Table of contents

- [Example 1](#)
- [Example 2](#)
- [Example 3](#)
- [Example 4](#)

With Compose, you can use built-in [YAML](#) features to make your Compose file neater and more efficient. Anchors and aliases let you create re-usable blocks. This is useful if you start to find common configurations that span multiple services. Having re-usable blocks minimizes potential mistakes.

Anchors are created using the `&` sign. The sign is followed by an alias name. You can use this alias with the `*` sign later to reference the value following the anchor. Make sure there is no space between the `&` and the `*` characters and the following alias name.

You can use more than one anchor and alias in a single Compose file.

[Example 1](#)

```
volumes:
  db-data: &default-volume
    driver: default
  metrics: *default-volume
```

In the example above, a `default-volume` anchor is created based on the `db-data` volume. It is later reused by the alias `*default-volume` to define the `metrics` volume.

Anchor resolution takes place before [variables interpolation](#), so variables can't be used to set anchors or aliases.

[Example 2](#)

```
services:
  first:
    image: my-image:latest
    environment: &env
      - CONFIG_KEY
      - EXAMPLE_KEY
      - DEMO_VAR
  second:
    image: another-image:latest
    environment: *env
```

If you have an anchor that you want to use in more than one service, use it in conjunction with an [extension](#) to make your Compose file easier to maintain.

[Example 3](#)

You may want to partially override values. Compose follows the rule outlined by [YAML merge type](#).

In the following example, `metrics` volume specification uses alias to avoid repetition but overrides `name` attribute:

```
services:
  backend:
    image: example/database
    volumes:
      - db-data
      - metrics
volumes:
  db-data: &default-volume
    driver: default
    name: "data"
  metrics:
    <<: *default-volume
    name: "metrics"
```

[Example 4](#)

You can also extend the anchor to add additional values.

```
services:
  first:
    image: my-image:latest
    environment: &env
    FOO: BAR
    ZOT: QUIX
  second:
    image: another-image:latest
    environment:
      <<: *env
      YET_ANOTHER: VARIABLE
```

Note

[YAML merge](#) only applies to mappings, and can't be used with sequences.

In example above, the environment variables must be declared using the `FOO: BAR` mapping syntax, while the sequence syntax – `FOO=BAR` is only valid when no fragments are involved.