

# Setting Up Your Repository

In the [deployment-configurations](#) repository you can centrally manage configurations for deployments on the GlueOps Platform. This guide will walk you through the steps to set up your repository, providing a detailed explanation of its structure and how to customize it to suit your organization's needs.

## ! INFO

Here is the [deployment-configurations](#) template you use to ease setup overhead.

## Directory Structure

The [deployment-configurations](#) repository is organized into three core directories:

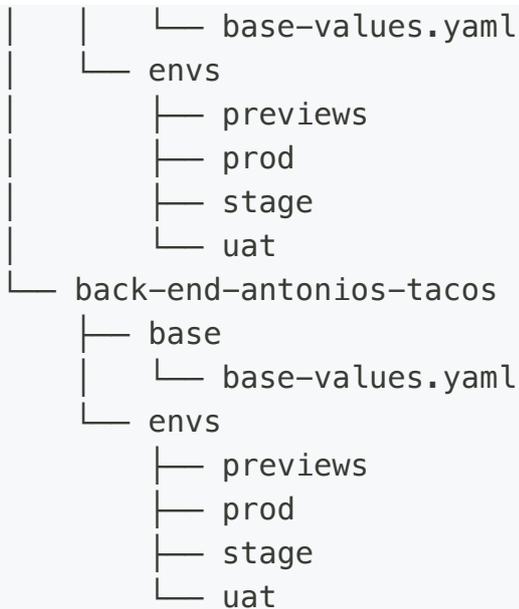
```
deployment-configurations
├── apps
├── common
├── env-overlays
```

- **apps:** Contains configurations for each application deployed on the GlueOps Platform.
- **common:** Holds configurations shared across all deployments on the platform. It allows you to avoid repetition of configuration files for more efficient setups.
- **env-overlays:** Contains configurations applied to groups of environments, like "prod" and "nonprod."

## apps Directory

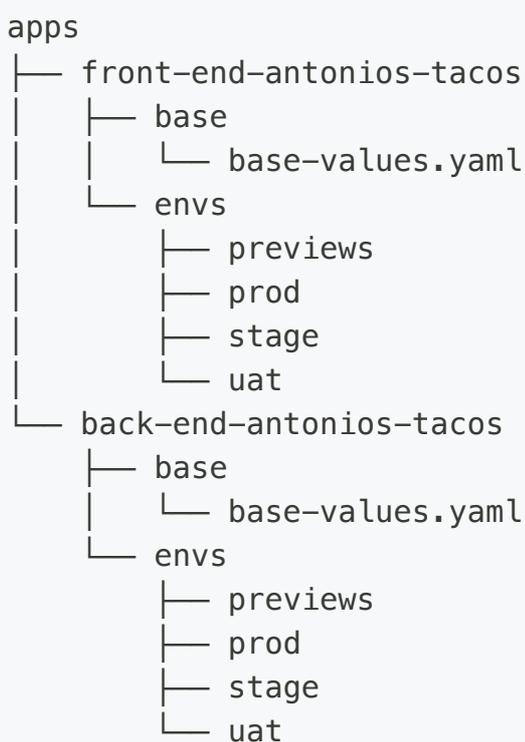
The `apps` directory is where you define the applications deployed on the GlueOps Platform and the environments for each application. Here's is how it's structured:

```
apps
├── front-end-antonios-tacos
│   └── base
```



In the directory, we have two applications deployed on the platform: `front-end-antonios-tacos` and `back-end-antonios-tacos`.

- The `base-values.yaml` file in the `base` directory contains configurations shared among all environments, such as the image repository and image pull policy.
- The `values.yaml` file is placed under each specific environment directory (`prod`, `uat`, `stage`) within the `envs` directory this contains environment-specific configurations like image tags, secrets, and ports.
- The `previews` file within the `envs` directory allows you to configure ephemeral deployments based on open pull requests.



# env-overlays Directory

The `env-overlays` directory contains directories representing groups of environments. Here, you can configure settings common to specific environments. Here's the structure:

```
env-overlays
├── nonprod
│   └── env-values.yaml
└── prod
    └── env-values.yaml
```

Each directory contains an `env-values.yaml` file with these configurations.

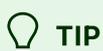
- The `nonprod` directory contains configurations applied to all non-production environments, like `stage` and `uat`.
- The `prod` directory contains configurations that apply to the production environment.

## Creating Your deployment-configurations Repository

To set up your `deployment-configurations` repository:

1. Create a new repository using the provided [template](#) in your GitHub organization.
2. In the `app` directory, duplicate one of the example demo apps and rename it to your **repository name**. For example, if your repository name is ***neptune*** then replace your application name to ***neptune***.

```
├── neptune
│   ├── base
│   │   └── base-values.yaml
│   └── envs
│       ├── previews
│       ├── prod
│       ├── stage
│       └── uat
```



TIP

You can add new directories for each additional application you want to deploy.

3. In the `base-values.yaml` file inside the `base` directory, update the information to fit your application.
4. Update the `values.yaml` file in the environments (e.g., `stage`, `prod`, `uat`) you want to deploy. Change the image tag, hostnames, and other necessary details to match your application and GlueOps configuration.

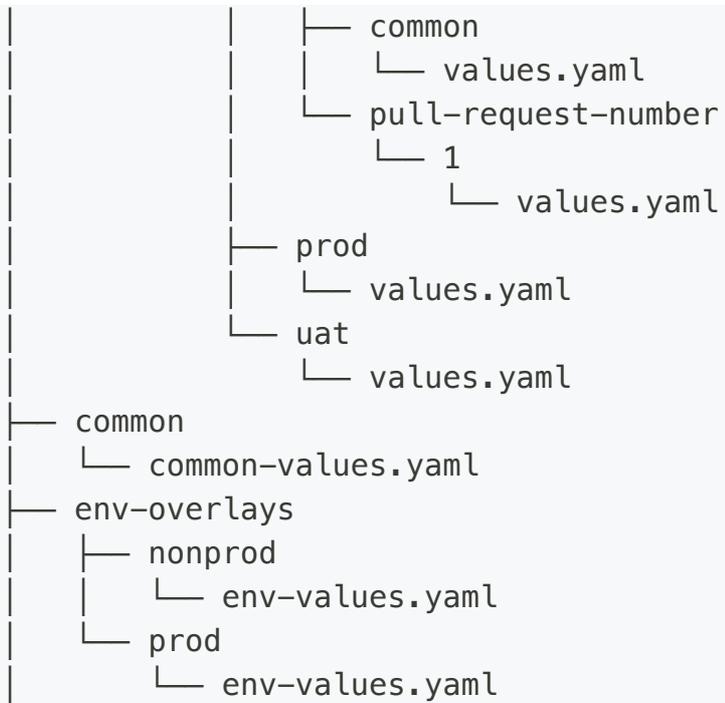
### ! INFO

There is no maximum number of directories per environment. You can choose to have environments like `stage`, `prod`, and `uat`, or just `stage` and `prod` based on your needs.

5. Repeat the above steps for each application and environment you want to manage in your repository.

For example, if you want to deploy the applications `data-api` in `stage` and `prod` environments and `commerce-front-end` in `uat` and `prod` environments, the resulting directory structure would look like this:

```
deployment-configurations
├── apps
│   ├── data-api
│   │   ├── base
│   │   │   └── base-values.yaml
│   │   └── envs
│   │       ├── previews
│   │       │   ├── common
│   │       │   │   └── values.yaml
│   │       │   └── pull-request-number
│   │       │       └── 1
│   │       │           └── values.yaml
│   │       ├── prod
│   │       │   └── values.yaml
│   │       └── stage
│   │           └── values.yaml
│   └── commerce-front-end
│       ├── base
│       │   └── base-values.yaml
│       └── envs
│           └── previews
```



By following these steps, you can efficiently manage and organize your deployment configurations on the GlueOps Platform, providing better control and consistency across your applications and environments.