Creating Releases using Docker

Justin Menga
FULL STACK TECHNOLOGIST
@jmenga pseudo.co.de

Introduction

Release Stage

- Creating application release settings
- Building a release image
- Creating a release environment
- Running acceptance tests

Continuous Delivery Workflow

Test

Build

Release

Deploy

Release Workflow using Docker

Create Release Environment

Create Release Settings
Create Release Image

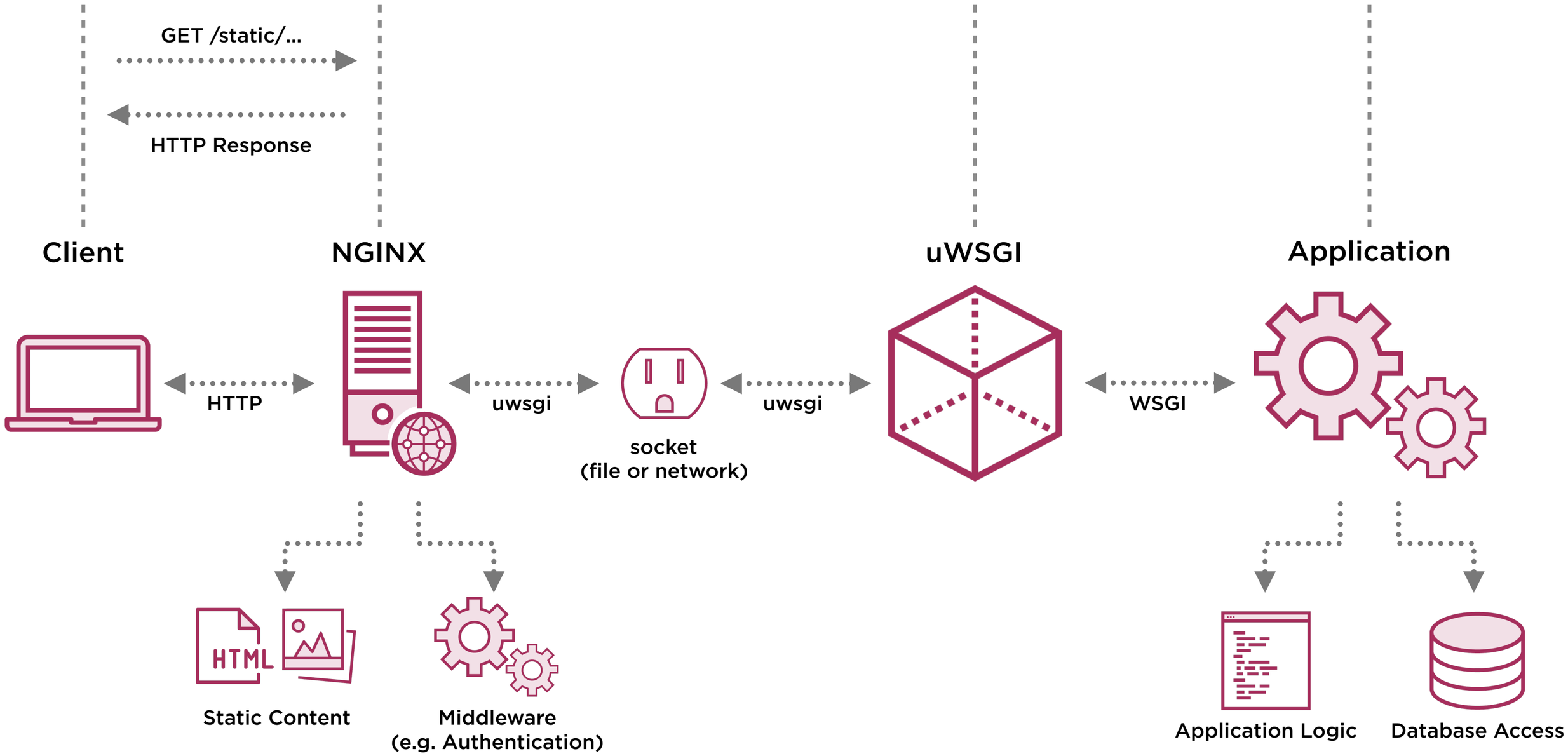
Bootstrap Release Environment

Prepare Environment
Start Application

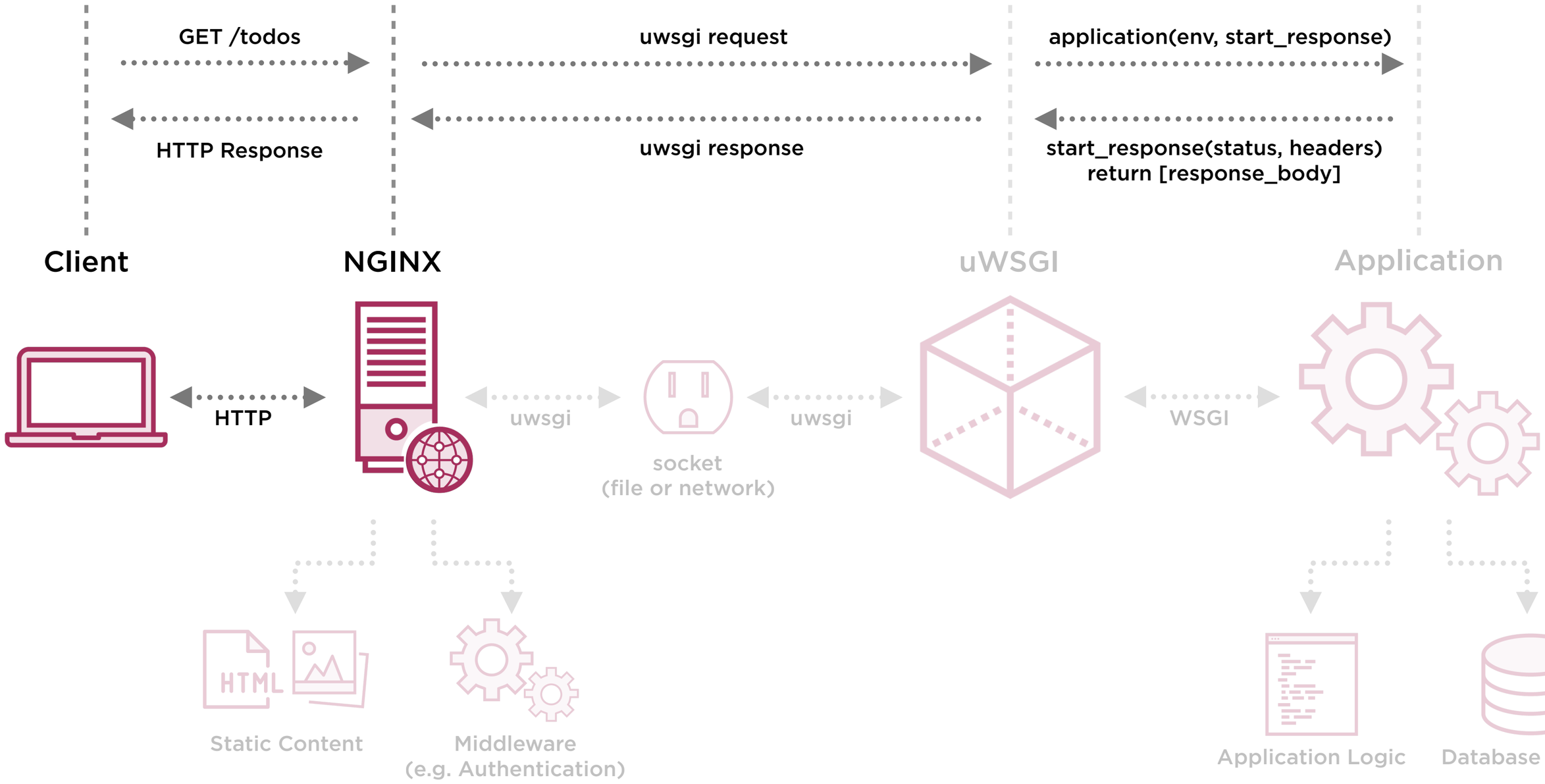
Acceptance Testing

Run Acceptance Tests
Publish Release Image

Serving the Application



Serving the Application



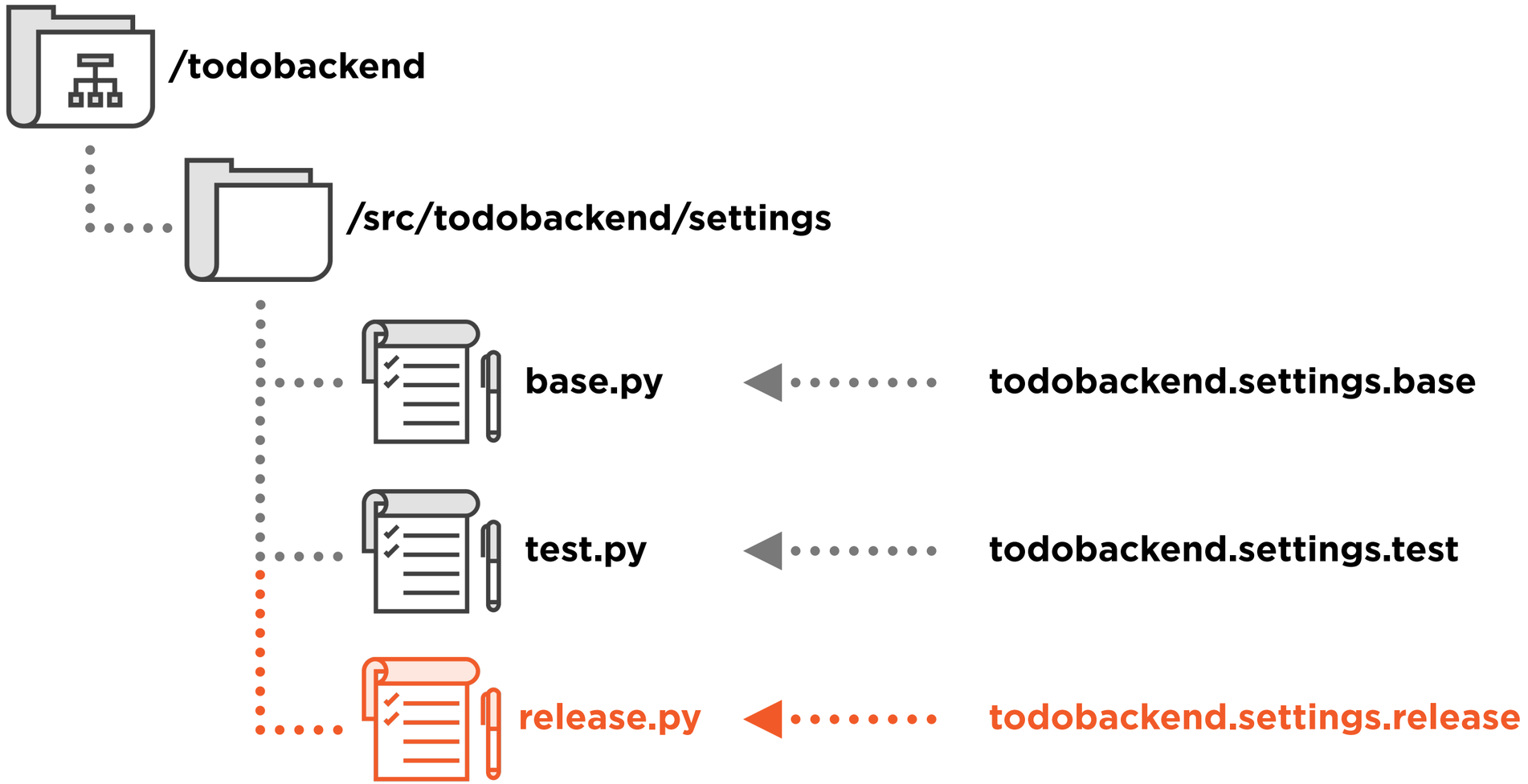
Demo

Creating the Release Environment

- Creating application release settings
- Creating the release image
- Describing the release environment
- Testing the release image

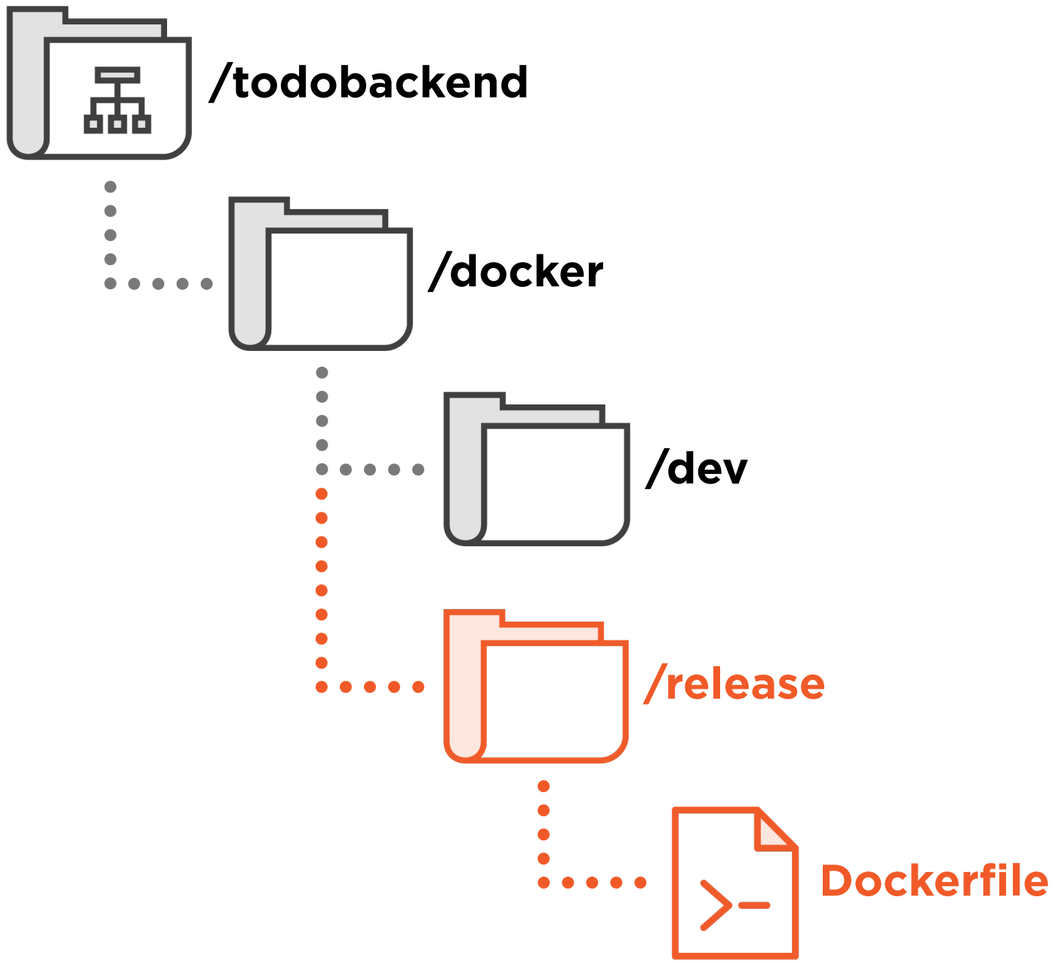
Creating Application Release Settings

Application Configuration Settings



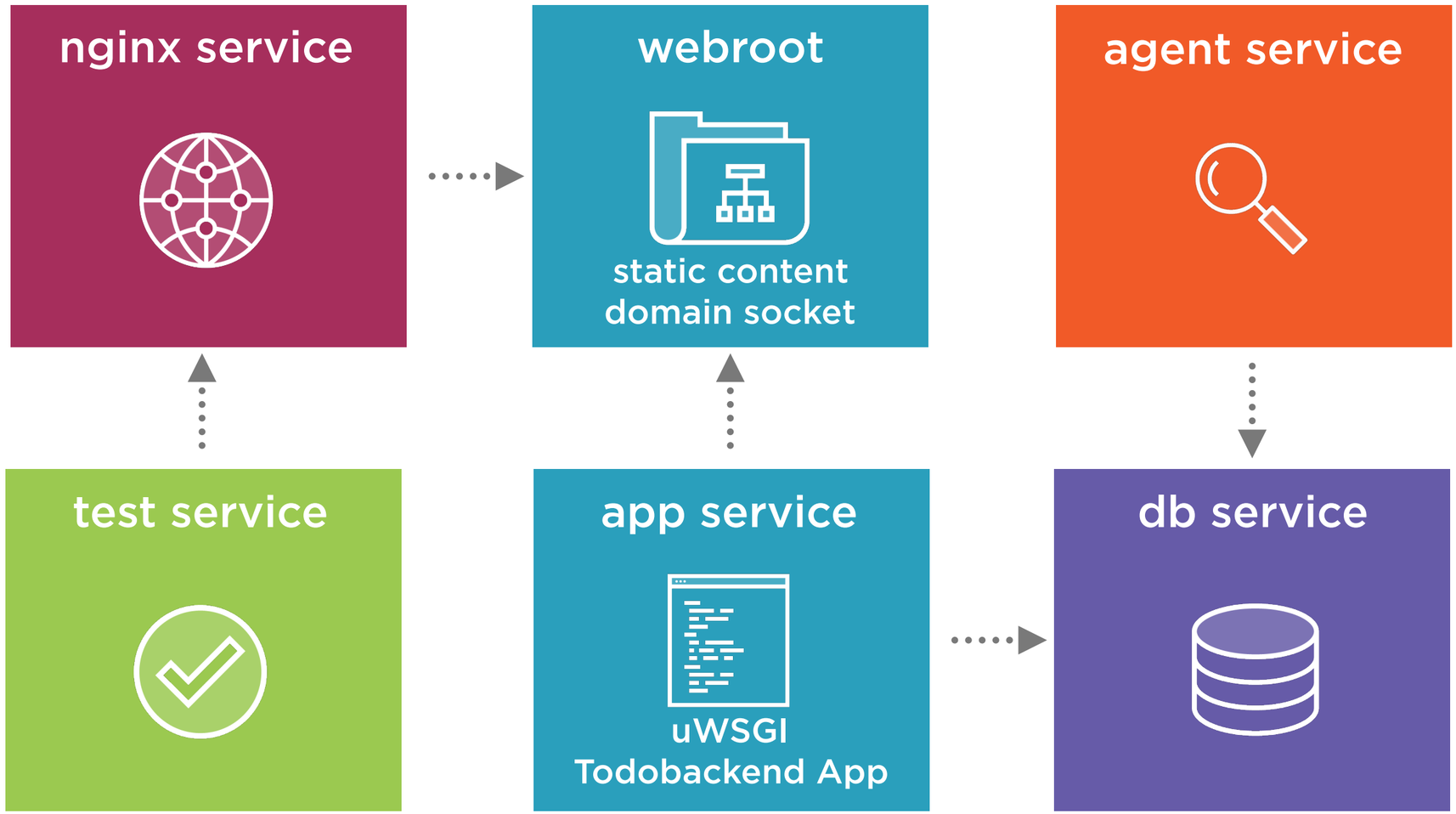
Creating the Release Image

Release Environment Folder Structure

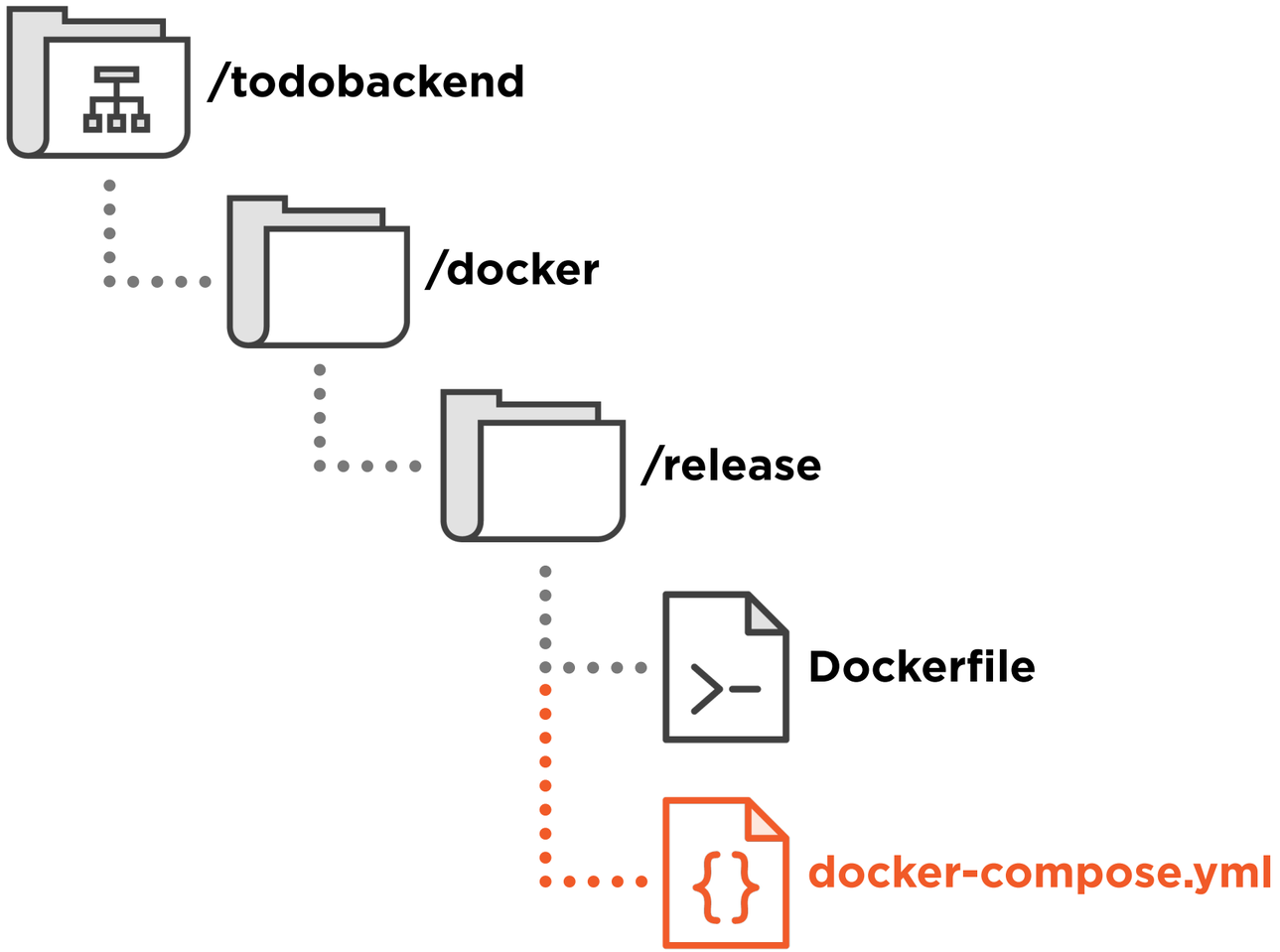


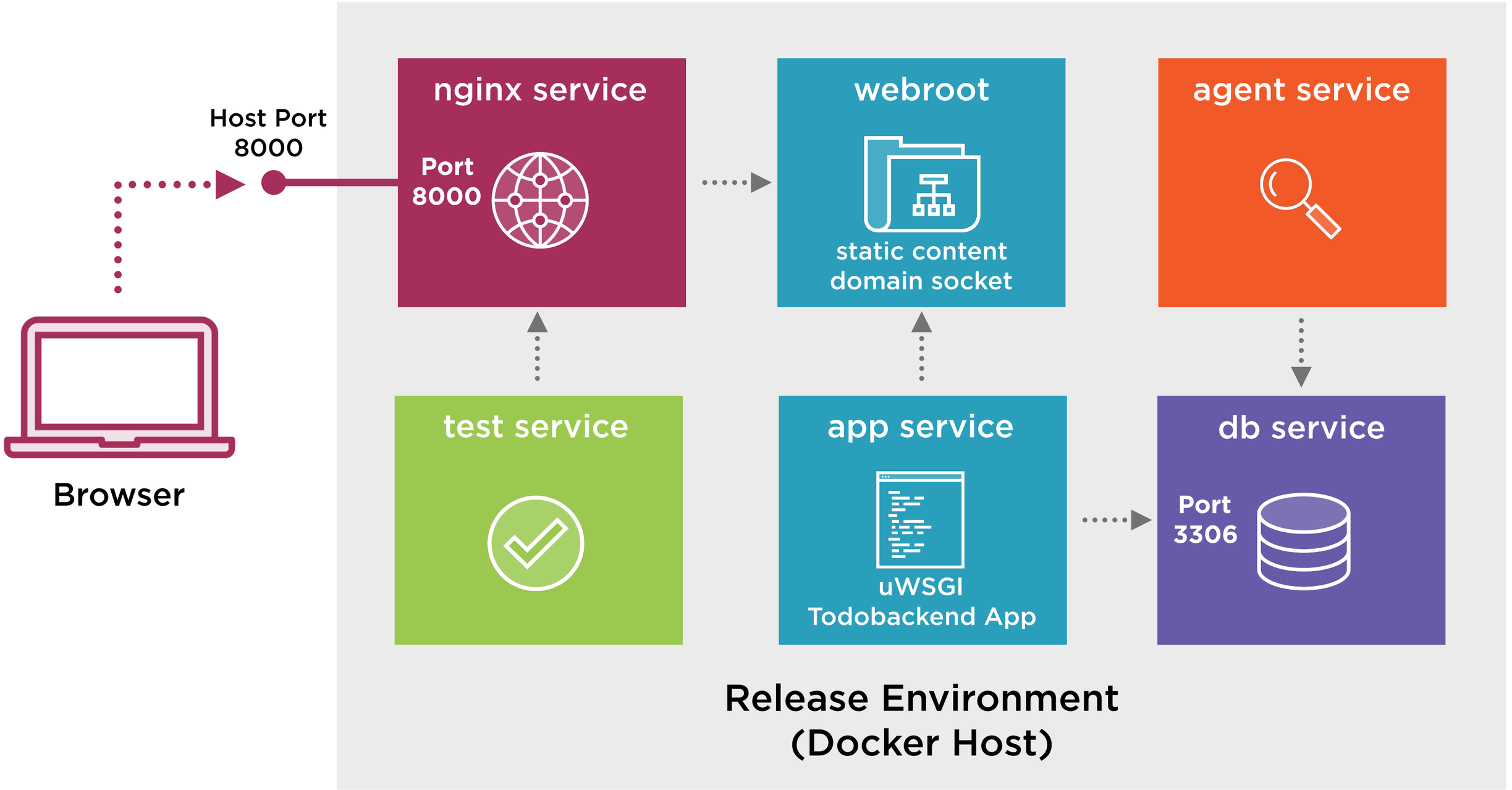
Describing the Release Environment

Release Environment Design



Release Environment Folder Structure




Host Port
8000
Browser
nginx service
Port
8000
webroot
static content
domain socket
agent service
test service
app service
uWSGI
Todobackend App
db service
Port
3306
Release Environment
(Docker Host)

Testing the Release Image

Release Environment Issues

Runtime Environment

Missing requirements from base image

Encountered upon initial setup

Application container fails to start

Release environment misconfiguration

Bootstrap Tasks

Establish initial state

Required for each release stage iteration

Typically present as errors to user

E.g. Setup database schema and data

Demo

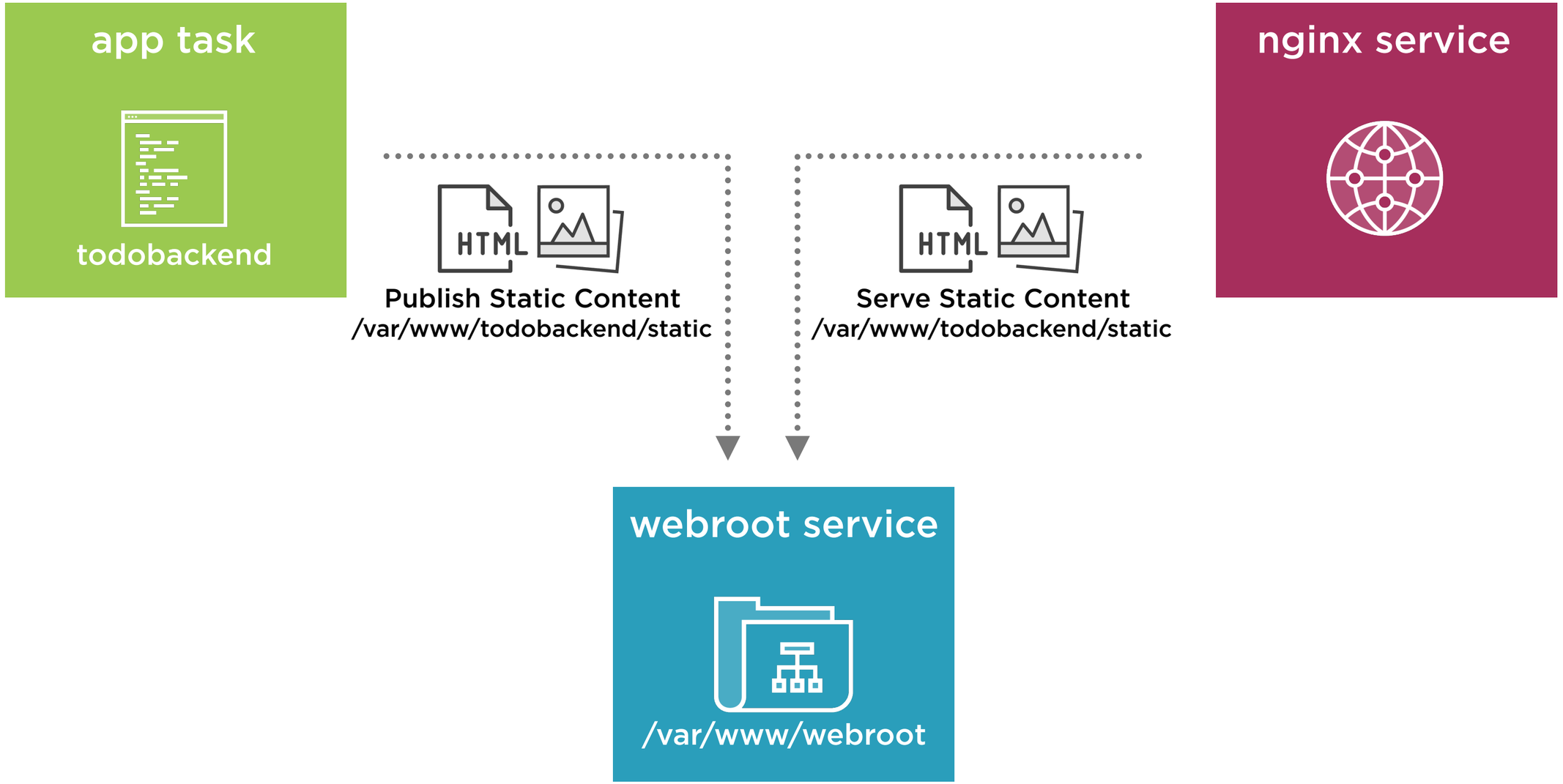
Bootstrapping the Application

- Initialisation tasks
- Starting the application

Initialisation Tasks

Collecting Static Files

docker-compose run manage.py collectstatic



Bootstrapping the Application Review

```
docker-compose build
```

- Creates release environment from latest build artifacts

```
docker-compose up agent
```

- Starts db service and ensures it is running

```
docker-compose run --rm app
manage.py collectstatic
--noinput
```

- Creates static content in the webroot volume container

```
docker-compose run --rm app
manage.py migrate --noinput
```

- Creates schema and tables in application database

```
docker-compose up nginx
```

- Starts linked app service and then starts nginx front end

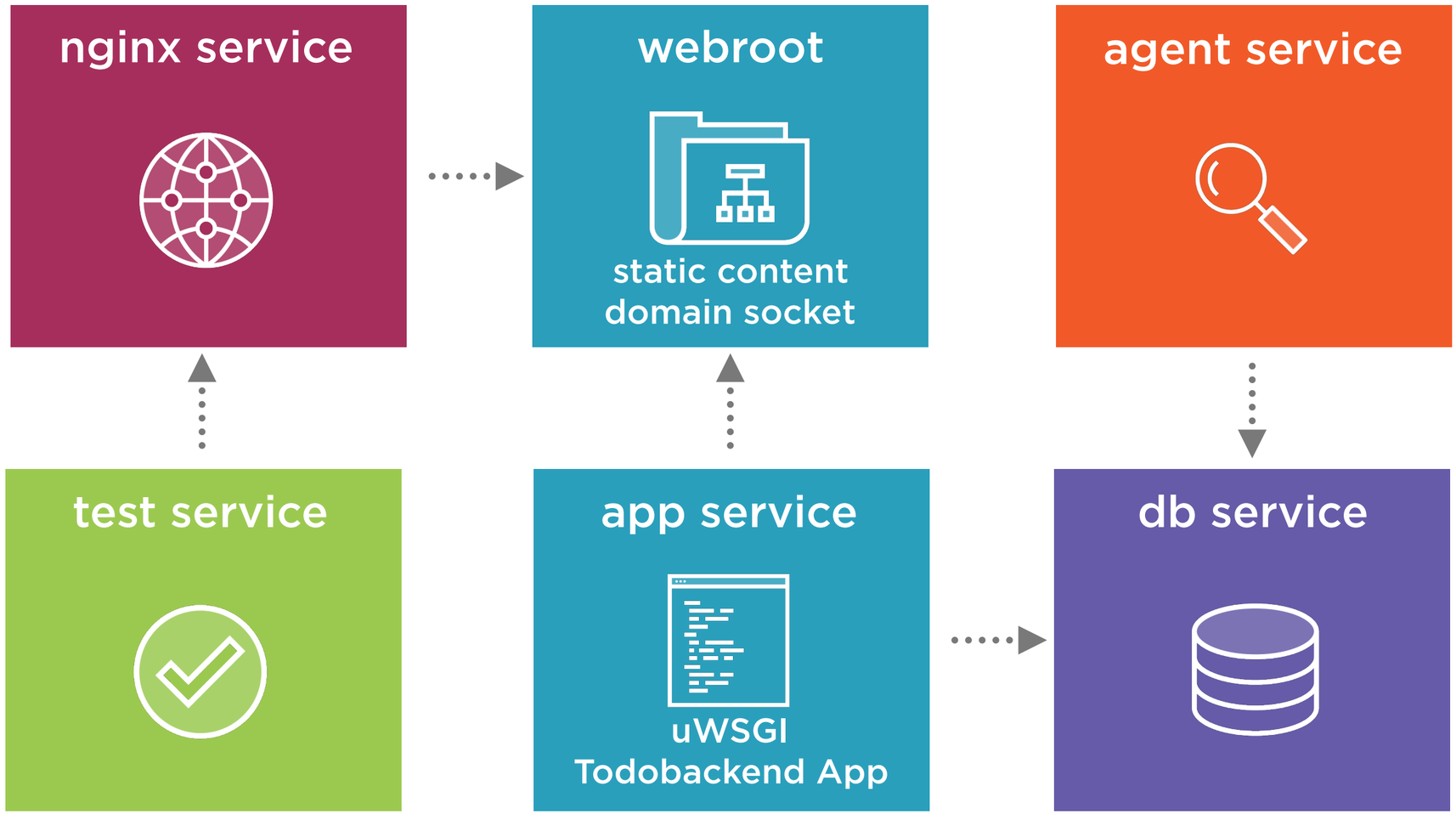
Demo

Acceptance Testing

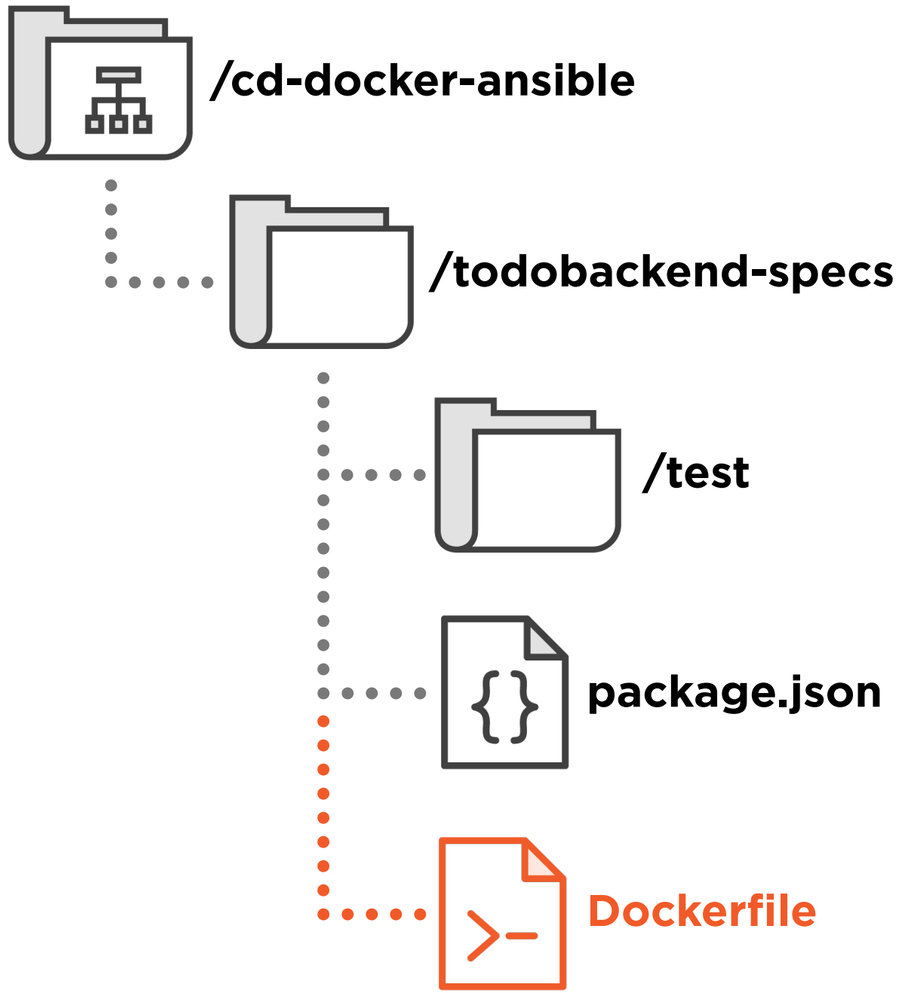
- Creating a test service
- Running acceptance tests
- Publishing the release image

Creating the Test Service

Release Environment Design



Acceptance Tests Folder Structure



Running Acceptance Tests

Continuous Delivery Milestones

Unit and Integration Tests

Application Artifact and Docker Release Image

Acceptance Tests

Publish Application Artifact and Docker Release Image

Continuous Delivery Milestones

Unit and Integration Tests

Application Artifact and Docker Release Image

Acceptance Tests

Publish Application Artifact and Docker Release Image

Continuous Delivery Workflow Review

Test Environment

1. docker-compose build

Development Image

3. docker-compose up test

test service

cache service

/cache

4. docker-compose up builder

builder service

db service

agent service

2. docker-compose up agent

target -> /wheelhouse

Release Environment

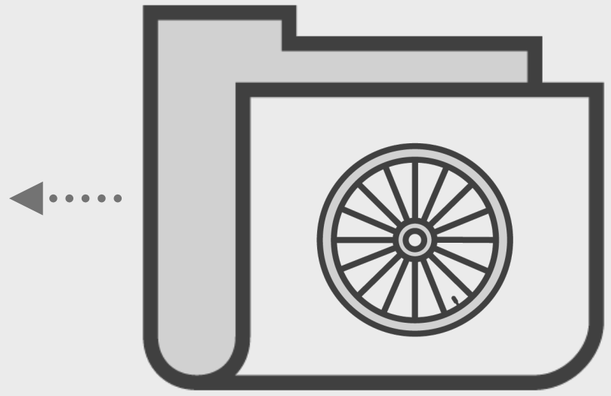
5. docker-compose build

Release
Image

6. docker-compose up agent

app service

Application

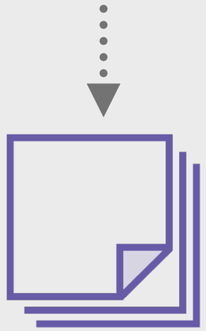


target -> /wheelhouse

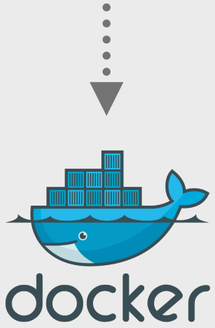
Release Environment

5. docker-compose build

Release Image

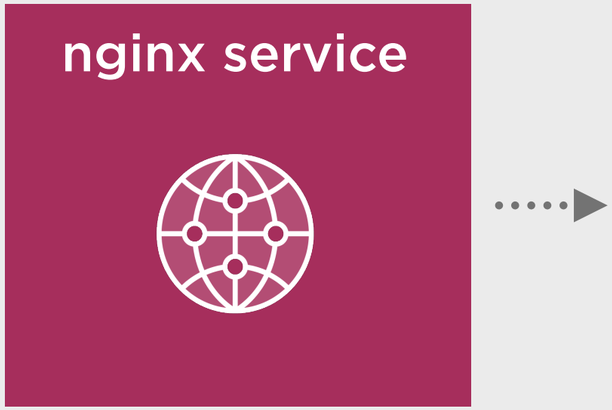


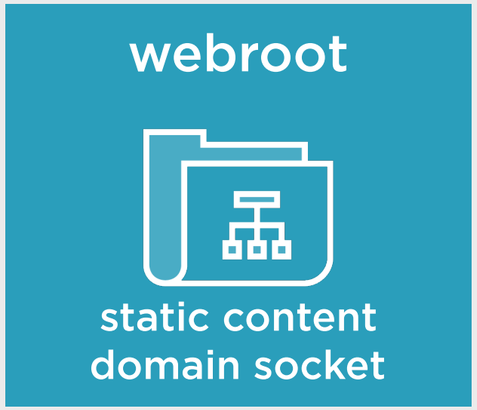
Release Candidate

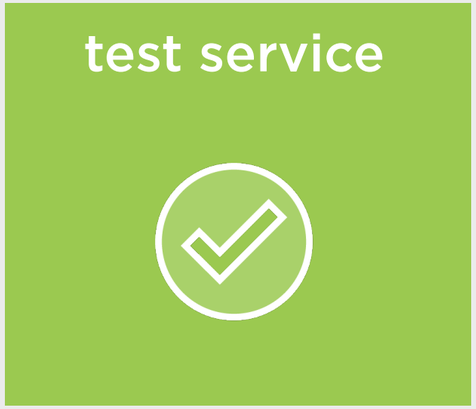


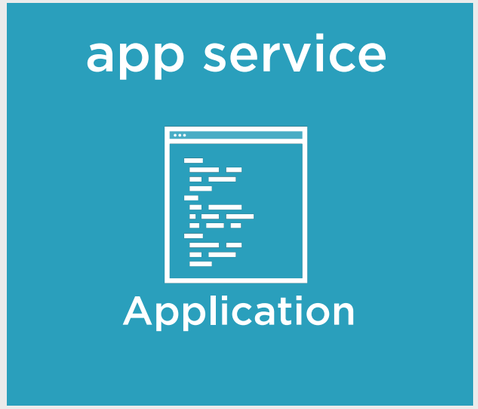
7. docker-compose run app manage.py collectstatic

6. docker-compose up agent









9. docker-compose up test

8. docker-compose run app manage.py migrate

Summary

Release Stage

- Creates release image from application build artifacts
- Verifies application passes acceptance tests in a release environment
- Runs from a release environment
- Requires initialization tasks
- Release image ready to be published and deployed if all tests pass