How it works

The system polls Dart every 5 minutes looking for recently published packages. When it finds a new package or version, the system downloads and extracts the packages, inserts it into a Docker container, executes `pub install` and runs your unit tests.

Getting Started

Every Dart package is automatically tested. There is no signup, no setup and no configuration. We do ask that you follow certain best-practices regarding project structure and naming conventions, discussed below.

Detecting Tests

Unit tests should be located in the `/test` directory in the root of your project, and all unit tests should have the suffix `_test.dart`:

```
test/*_test.dart
```

Alternatively, you can place a `run.sh` script in the test directory for a bit more control. If the `run.sh` shell script is detected it will take precedence.

```
test/run.sh
```

Content Shell Tests

Content Shell tests are identified as `_test.dart` files that have a matching `_test.html` file. Running these tests is currently disabled, but should be available in a few weeks.

Viewing Results

You can see the results of your tests at http://pub.drone.io

Advanced

In some cases, you may need more control over your build commands and build environment. For example, if your package is a Redis driver you'll probably want a running instance of Redis available.

This can be achieved by adding a `.drone.yml` to the root of your project. A `.drone.yml` file will always take precedence when attempting to configure your build environment and determine

which commands to execute. Here is an example file:

```
image: dart
services:
  - redis
script:
  - bash test/load_data.sh
  - dart test/postgres_test.dart
```

Each service is started in its very own container. Traffic is proxied between the build container and the service container(s). This means you can connect to services via `localhost` and the standard port, even though the service is running in a separate container.

When running unit tests, we recommend connecting to services via TCP and not Unix Sockets. Because services run in separate containers, you will not be able to access the Unix Socket.

Available Services

Here is a full list of available services:

```
services:
  - cassandra
  - couchdb
  - elasticsearch
  - neo4j
  - mongodb
  - mysql
  - postgres
  - rabbitmq
  - redis
  - riak
  - zookeeper
```