Deploying with Docker

> Docker provides a way to run applications securely isolated in a container, packaged with all its dependencies and libraries.

Docker uses Linux containers to define a standard environment for an application to run within.

GitToken provides a Docker image and Dockerfile for deploying a GitToken server instance. The following sections provide an overview of using `docker-machine` and `docker-compose` command line tools with the GitToken Docker server image.

Machine

`docker-machine` is a command line interface (CLI) tool provided by Docker to deploy containers and images on remote servers and virtual machines.

Using `docker-machine` with `docker-compose` is the preferred method for setting up a GitToken server instance.

Configuring a Docker Machine

Configuring a Docker machine requires a remote server to deploy the GitToken server to. The following example demonstrates configuring a basic generic (host agnostic) server.

```
# Create a new docker machine
docker-machine create \
  # Specify the `generic` driver
  --driver generic \
  # Supply the public network IP address of the machine
  --generic-ip-address=___.___.___.___ \
  # Provide the path to the private SSH key of the
  # current user for authenticating into the machine
  --generic-ssh-key ~/.ssh/id_rsa \
  # Name the machine
  machine_name
```

Docker Image

Docker images are cached snapshots of built Docker containers. They provide a minimal amount of data, which may be composed of immutable layers of cached snapshot data from other images, to form a container environment runnable by docker machines.

GitToken provides public Docker images for services on Docker Hub.

Using an Image

GitToken provides a NodeJS Express application for handling GitHub web hook events and distributing tokens for git contributions.

From within a Docker machine, running the CLI `docker pull gittoken/express-server` will grab the GitToken `express-server` image for running a GitToken server instance.

The image is intended to be run with `docker-compose` where an environment variables (`.env`) file is provided to the `docker-compose.yml` `env_file` field.

Environment Variables File

The `.env` file customizes the values for the GitToken server instance.

```
# Example environment variables (.env) file

# NOTE Web3 Provider must be a public IP address,
# and not associated with localhost.
#
# i.e. The IP address must not be 127.0.0.1 or 0.0.0.0
#
# If a Docker container is running an Ethereum client on the machine,
# use the IP Address associated with that container's public network IP address
# or the bound port of the service on the machine.

WEB3_PROVIDER                  = "http://___.___.___.___:8545"

IS_GITHUB_WEBHOOK              = true

GITTOKEN_DIRECTORY_PATH        = "/gittoken-server"

GITTOKEN_KEYSTORE_FILENAME     = ".keystore"

GITTOKEN_CONTRACT_FILE         = "contract.json"

GITTOKEN_FAUCET_ACTIVE         = false

GITTOKEN_CONTRACT_OWNER        = "0x0000000000000000000000000000000000000000"

GITTOKEN_CONTRACT_OWNER_EMAIL  = "your_email@your_organization.website"

GITTOKEN_CONTRACT_ORGANIZATION = "Your Organization"

GITTOKEN_CONTRACT_SYMBOL       = "SYMBOL"

GITTOKEN_CONTRACT_DECIMALS     = 8

GITTOKEN_API_SESSION_SECRET    = 'SOMETHINGFANTASTIC'

GITHUB_API_ID                  = "YOUR_GITHUB_APPLICATION_CLIENT_ID"

GITHUB_API_SECRET              = "YOUR_GITHUB_APPLICATION_CLIENT_SECRET"

GITHUB_CALLBACK_URL            = "https://your_organization.website/auth/github/callback"
```

Compose

`docker-compose` is a command line interface (CLI) tool for composing docker service deployments.

Deploying a GitToken server instance is intended to be done using a `docker-compose.yml` file to define the configuration of the service.

Use the `gittoken/express-server:v1` Docker image to build the GitToken service instance from.

Configuration parameters of a GitToken server instance are passed to the container using an environment variables file.

GitToken server instance requires both ports 1324 and 1325 to be publicly exposed and available to the host machine. The host machine may map the default ports to different ports, according to the needs of the developer. For example, the developer may instead map ports `3000:1324` to bind the service to port 3000 on the host machine.

```
# Example docker-compose.yml configuration using the
# GitToken express-server Docker image

# Use docker-compose v3
version: '3.0'
# Define services
services:
  # Define gittoken service
  gittoken:
    # Use the GitToken Docker image to build from
    image: "gittoken/express-server:v1"
    # Define an environment variables file path
    # relative to the Docker machine file system
    env_file:
      - /gittoken-server/gittoken.env
    # Expose port 1324 for the Express Server
    # Expose port 1325 for the WebSocket Server
    ports:
      - 1324:1324
      - 1325:1325
    # Mount the local volume of the Docker machine
    # to the container volume
    volumes:
      - /gittoken-server/:/gittoken-server/
```

Dockerfile

```
# Example Dockerfile

# Use NodeJS Docker image
FROM node:6.11.0

# Install yarn to quickly install cached dependency files
RUN npm i -g yarn

# Run the following commands inside the gittoken-server directory
WORKDIR /gittoken-server

# Clone the git-token/express-server repository
RUN git clone https://github.com/git-token/express-server.git .

# Install dependencies
RUN yarn install

# Build source files
RUN yarn run build-src

# Start the GiToken Server
ENTRYPOINT yarn run start

# Expose port 1324 for the express application;
# Expose port 1325 for the web socket server.
EXPOSE 1324 1325
```