

Deploying on Render

Login to docker hub

On your terminal

\$ docker login

< it will ask username/password>

Go to your BE repo

\$ cd be/app (there is Dockerfile)

\$ docker build -t docker.io/giddlyimages/giddly-be:<tag name>

Example:

\$ docker build -t docker.io/giddlyimages/giddly-be:1.0.43.biz .

Push to docker hub

\$ docker push docker.io/giddlyimages/giddly-be:1.0.43.biz .

Login to Render : <https://render.com/>

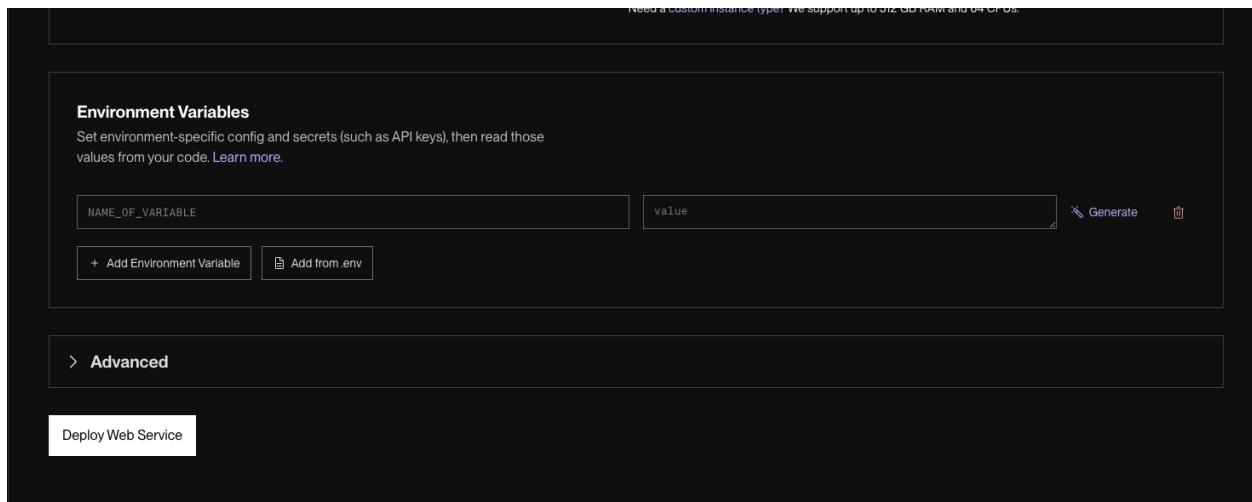
Goto Dashboard

Service Name	Status	Runtime	Region	Updated
giddly-be-business	✓ Deployed	Image	Singapore	30min
giddly-be:1.0	✓ Deployed	Image	Singapore	6h
biz-auth-service:1.0	✓ Deployed	Image	Singapore	4d
admin:1.0.0	✓ Deployed	Image	Singapore	10d
search-api:0.0.1	✓ Deployed	Image	Singapore	2mo

If you are deploying a new service , click “New” on top right corner.

Select “Web Service” -> Existing Image , place your docker image url you build up. Add credentials, select (Norway).

Add any environment variables using “Add Environment Variable” button .
Once ready “Deploy Web Service”



----- Redeploying an existing running App-----
If you want to update on an existing service . From Dashboard select the service , select
“Settings” on left nav bar.

The screenshot shows the Giddly dashboard with the "giddly-be-business" service selected. The left sidebar shows navigation options like Dashboard, Events, Settings (which is highlighted), and others. The main area is titled "Deploy" and contains sections for "Image" and "Docker Command". Under "Image", there's a "Image URL" field with the value "docker.io/giddlybeimages/giddly-be:1.043.biz" and a "Credential" field with the value "docker". Under "Docker Command", there's a text input field containing "\$" and a "Pre-Deploy Command" section with the note "Optional Render runs this command before the start command. Useful for". On the right side, there's a vertical sidebar with links to "General", "Deploy", "Custom Domains", "Edge Caching", "Notifications", "Health Checks", "Maintenance Mode", and "Delete or suspend".

Update Image in “Image URL” with your latest tag. Save it

The screenshot shows a modal dialog titled "Update Image". In the "Image URL" section, the input field contains "docker.io/giddlyimages/giddly-be:1.0.43.blz". Below this, a message states "We were able to authorize access to this image with the selected credential." At the bottom, there are "Cancel" and "Save Changes" buttons.

Top right corner “Manual Deploy” -> “Deploy Latest References”

The screenshot shows the Giddly platform interface. On the left, a sidebar includes "Dashboard", "giddly-be-business" (which is selected and highlighted in purple), "Events", "Settings" (which is also highlighted in purple), "MONITOR", "Logs", and "Metrics". The main area displays "WEB SERVICE" details for "giddly-be-business": "Service ID: srv-d4em0l2l9vc73avfsg", "Image: docker.io/giddlyimages/giddly-be:1.0.43.blz", and "business-be-dev.giddly.com". To the right, there are "Connect" and "Manual Deploy" buttons, with "Manual Deploy" being highlighted in purple. A dropdown menu under "Manual Deploy" lists "Deploy latest reference" and "Restart service". At the bottom, a "Settings" section is shown with a "General" tab selected.

Once deployed , you can start seeing logs in the log editor.

The screenshot shows the Giddly application dashboard for the project 'giddly-be-business'. The left sidebar contains navigation links for Dashboard, Events, Settings, MONITOR, Logs (which is selected and highlighted in purple), Metrics, MANAGE, Environment, Shell, Scaling, Previews, Disk, Jobs, Changelog, Invite a friend, and Contact support. The main area is titled 'Logs' and displays deployment logs for February 14, 2026. The logs show the application starting, finding configuration files, establishing database connections, and initializing worker dependencies. A note at the top states: 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.' A search bar and a filter for the last hour are also present.

```
Feb 14
04:09:54 PM =>> Starting service...
04:10:04 PM [qr2hv] Config file not found or unreadable: Config File "app" Not Found in "[/app]"
04:10:04 PM [qr2hv] 2026/02/14 21:10:04 firebase-service-account.json not found in /app directory; will check env var
04:10:04 PM [qr2hv] 2026/02/14 21:10:04 Using GOOGLE_APPLICATION_CREDENTIALS environment variable for Firebase credentials
04:10:04 PM [qr2hv] 2026/02/14 21:10:04 Firebase credentials resolved successfully: [REDACTED]giddlytest.firebaseio-adminsdk-fbsvc-c971aa6993.js
04:10:04 PM [qr2hv] 2026/02/14 21:10:04 Redis connection established successfully to redis-18666.cl.ap-southeast-1-1.ec2.cloud.redislabs.com:18666
04:10:04 PM [qr2hv] 2026/02/14 21:10:04 Asynq client connected to Redis (shared connection)
04:10:04 PM [qr2hv] 2026/02/14 21:10:04 Asynq client initialized successfully
04:10:04 PM [qr2hv] PostgreSQL connected successfully...
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 Worker dependencies initialized successfully
04:10:05 PM [qr2hv] [GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 CORS configured with origins: [https://admin-1-0-0-jx60.onrender.com http://localhost:5173 https://giddly-be-business-webapp.onrender.com https://business-fe-dev.giddly.com]
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 Detected DER format (44 bytes), extracting 32-byte raw key
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 Admin public key initialized successfully (length: 32 bytes)
04:10:05 PM [qr2hv]
04:10:05 PM [qr2hv] [GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
04:10:05 PM [qr2hv] - using env: export GIN_MODE=release
04:10:05 PM [qr2hv] - using code: gin.SetMode(gin.ReleaseMode)
04:10:05 PM [qr2hv]
04:10:05 PM [qr2hv] [BADGER] 2026/02/14 21:10:05 BadgerDB initialized successfully
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 Initializing SES rate limit buckets in Redis...
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 ✓ SES rate limit buckets initialized (rate: 14/s, capacity: 50, daily: 10M)
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 ✓ Asynq worker server started
04:10:05 PM [qr2hv] 2026/02/14 21:10:05 Starting Asynq worker server
```