Setting up DuckDB on a Digital Ocean Droplet

J. Christopher Westland

2025-03-13

Overview

- You can set up a DuckDB database with R and Plumber on a Digital Ocean droplet by following specific steps, though some details may require manual configuration.
- PlumberDeploy can help deploy the API, but you'll need to ensure the database file is included and the firewall is set up.
- You will need to populate the database locally before deployment, as plumberDeploy focuses on API deployment, not database population.

Setting Up DuckDB, R, and Plumber on Digital Ocean

To get started, you'll first need to set up DuckDB locally in R, populate it with your data, and create a Plumber API that connects to this database. Then, use plumberDeploy to upload everything to a Digital Ocean droplet. Here's how:

Local Setup

- Install the duckdb package in R using install.packages("duckdb").
- Create and populate your DuckDB database file (e.g., "mydatabase.duckdb") with data, like this:

```r
library(duckdb)
con <- dbConnect(duckdb(), dbfile = "mydatabase.duckdb")
# Example: Write a data frame to the database
my_data_frame <- data.frame(id = 1:5, value = rnorm(5))
dbWriteTable(con, "mytable", my_data_frame)
dbDisconnect(con)
```

- Write your Plumber API in an R file (e.g., "api.R"), ensuring it uses a relative path to connect to "mydatabase.duckdb". For example:

```r
#* @get /data
function() {
  library(duckdb)
  con <- dbConnect(duckdb(), dbfile = "mydatabase.duckdb")
  data <- dbGetQuery(con, "SELECT * FROM mytable")
  dbDisconnect(con)
  return(data)
}
```

Deployment to Digital Ocean

- Set up a Digital Ocean account, generate an API key, and add an SSH key to your account.
- Install plumberDeploy and analogsea packages: install.packages(c("plumberDeploy", "analogsea")).
- Validate your Digital Ocean account with analogsea::account().
- Deploy your API and database by placing both "api.R" and "mydatabase.duckdb" in a directory (e.g., "myapi"), then run:

```r
pr <- plumber::plumb("api.R")
pr$run(port = 8000)
```

Access it at http://localhost:8000/data to ensure it works before deployment.

2. Preparing for Deployment

Create a directory (e.g., "myapi") and place both "api.R" and "mydatabase.duckdb" inside it. This directory will be the source for deployment, and plumberDeploy will copy all files within it to the droplet.

3. Setting Up Digital Ocean Account

To deploy, you'll need a Digital Ocean account:

- Sign up at Digital Ocean's website and generate an API key from the API section of your account settings.
- Set up an SSH key for secure access. You can use analogsea::key_create() or follow Digital Ocean's guide at Digital Ocean SSH Keys.
- Install the necessary R packages: install.packages(c("plumberDeploy", "analogsea")).
- Validate your account with analogsea::account() to ensure connectivity.

4. Deploying with plumberDeploy

Use plumberDeploy to deploy your API:

- Run the deployment command:

```
plumberDeploy::do_deploy_api(name = "myapi", path = "path/to/myapi", port = 8000, install_pa
```

- name: The name of your API, which will also name the droplet if a new one is created.
- path: The path to your "myapi" directory containing "api.R" and "mydatabase.duckdb".

```
-   port: The port for the API (default is 8000, but ensure it's open on the droplet).

-   install_packages: Specifies "duckdb" and "plumber" to ensure they're installed on the dr
```

plumberDeploy will:

- Create a new droplet if one doesn't exist with the given name.
- Install R and the specified packages on the droplet.
- Copy all files from the "path" directory (including "mydatabase.duckdb") to "/var/plumber/[name]" on the droplet.
- Start the Plumber API service, running it from "/var/plumber/[name]".

5. Post-Deployment Configuration

After deployment, you need to ensure the API is accessible:

- Digital Ocean droplets have a default firewall that allows SSH (port 22) and HTTP (port 80), but port 8000 (or your chosen port) may not be open. Log into your droplet via SSH and add a firewall rule:
  - Use ufw allow 8000 to allow traffic on port 8000, or configure via the Digital Ocean control panel.
- Your API should now be accessible at http://[droplet_ip]:8000/data, where [droplet_ip] is the IP address of your droplet, visible in the Digital Ocean dashboard.

6. Ensuring Persistence and Accessibility

The Plumber API service is set up to run persistently, and plumberDeploy handles starting it. However, ensure it restarts on server reboot by checking the service configuration (typically managed by systemd on Ubuntu droplets). You can verify with:

- sudo systemctl status plumber-myapi (replace "myapi" with your API name).

Technical Considerations

- Database File Location: The Plumber API code uses a relative path ("my-database.duckdb") to connect to the database. On the droplet, this file is in "/var/plumber/[name]", and the working directory is set to this location during API execution, ensuring the relative path works.
- DuckDB Dependencies: DuckDB is self-contained within the R package, but ensure the R version on the droplet (installed by plumberDeploy from CRAN) is compatible with your duckdb package version.
- Firewall and Security: Opening ports manually is necessary for accessibility, but consider security implications, such as restricting access to specific IP ranges if needed.

Table: Deployment Checklist

Step	Action	Notes
Local Setup	Install duckdb, create and populate database, write Plumber API	Test locally first

Step	Action	Notes
Prepare Directory	Place API code and database file in "myapi" directory	Ensure all files are included
Digital Ocean Setup	Create account, API key, and SSH key	Validate with analogsea::account()
Deploy API	Use plumberDeploy::do_deploy_api with install_packages = c("duckdb")	Specify port (e.g., 8000)
Post-Deployment	Add firewall rule for the port (e.g., ufw allow 8000)	Check accessibility at droplet IP:port
Verify Service	Ensure Plumber service is running and persistent	Use systemctl for status checks

Addressing the "Populate with plumberDeploy" Query

The phrase "populate it with plumberDeploy" is likely a misunderstanding, as plumberDeploy is for deployment, not database population. It's recommended to populate the DuckDB database locally before deployment, as shown in Step 1. If you need to populate the database via the API after deployment, you can add POST endpoints to insert data, but that's a separate step not handled by plumberDeploy.

Detailed Guide for Deploying DuckDB, R, and Plumber on Digital Ocean with SSH Setup

This section provides a comprehensive guide for deploying a DuckDB database with R and Plumber onto a Digital Ocean droplet, addressing all aspects of the process as outlined in the user's query, including the setup of Digital Ocean with SSH. The focus is on ensuring the database is populated and the API is deployed correctly, with attention to technical details and potential challenges, especially around SSH configuration.

Understanding the Components

To begin, let's clarify the tools involved:

- DuckDB is an in-memory analytical database management system that can also use on-disk storage, accessible in R via the duckdb package (DuckDB R Package).
- R is a programming language for statistical computing, widely used for data analysis.
- Plumber is an R package that turns R functions into web APIs using special comments, making it easy to expose your code as HTTP endpoints (Plumber Package).

- plumberDeploy is a companion package to Plumber, designed to automate the deployment of Plumber APIs to cloud servers like Digital Ocean, focusing on ease of use for R developers (plumberDeploy Package).
- Digital Ocean droplet is a virtual machine provided by Digital Ocean, a cloud computing platform, where you'll host your setup.

The user's query mentions "populate it with plumberDeploy," which initially seems ambiguous. Given the documentation, plumberDeploy is primarily for deploying APIs, not populating databases. It's likely the user intends to populate the DuckDB database locally before deployment and then upload both the R-Plumber code and the populated database to the droplet.

Step-by-Step Deployment Process

1. Local Setup of DuckDB and Plumber API

First, set up DuckDB in your local R environment:

- Install the duckdb package: install.packages("duckdb").
- Create a DuckDB database file and populate it with data. For example:

```
library(duckdb)
con <- dbConnect(duckdb(), dbfile = "mydatabase.duckdb")
# Example: Write a data frame to the database
my_data_frame <- data.frame(id = 1:5, value = rnorm(5))
dbWriteTable(con, "mytable", my_data_frame)
dbDisconnect(con)
```

This creates a file named "mydatabase.duckdb" in your working directory, populated with sample data.

Next, create your Plumber API:

- Install Plumber if not already installed: install.packages("plumber").
- Write your API in an R file, ensuring it connects to the DuckDB database using a relative path. For instance, create "api.R" with:

```r
#* @get /data
function() {
  library(duckdb)
  con <- dbConnect(duckdb(), dbfile = "mydatabase.duckdb")
  data <- dbGetQuery(con, "SELECT * FROM mytable")
  dbDisconnect(con)
  return(data)
}
```

This example defines a GET endpoint "/data" that returns all rows from "mytable". Test it locally using:

```r
pr <- plumber::plumb("api.R")
pr$run(port = 8000)
```

Access it at http://localhost:8000/data to ensure it works before deployment.

2. Preparing for Deployment and Setting Up Digital Ocean with SSH

Create a directory (e.g., "myapi") and place both "api.R" and "mydatabase.duckdb" inside it. This directory will be the source for deployment, and plumberDeploy will copy all files within it to the droplet.

To set up Digital Ocean with SSH:

- Sign up at Digital Ocean and generate an API key from the API section of your account settings.
- Set up an SSH key for secure access. You can use analogsea::key_create() or follow Digital Ocean's guide at Digital Ocean SSH Keys. Add the public key to your Digital Ocean account via the control panel.
- Install the necessary R packages: install.packages(c("plumberDeploy", "analogsea")).
- Validate your Digital Ocean account with analogsea::account() to ensure connectivity.

3. Deploying with plumberDeploy

Use plumberDeploy to deploy your API:

- Run the deployment command:

```r
pr <- plumber::plumb("api.R")
pr$run(port = 8000)
```

- name: The name of your API, which will also name the droplet if a new one is created.

- path: The path to your “myapi” directory containing “api.R” and “mydatabase.duckdb”.

```
-   port: The port for the API (default is 8000, but ensure it's open on the droplet).

-   install_packages: Specifies "duckdb" and "plumber" to ensure they're installed on the dr
```

plumberDeploy will:

- Create a new droplet if one doesn't exist with the given name.
- Install R and the specified packages on the droplet.
- Copy all files from the “path” directory (including “mydatabase.duckdb”) to “/var/plumber/[name]” on the droplet.
- Start the Plumber API service, running it from “/var/plumber/[name]”.

4. Post-Deployment Configuration

After deployment, you need to ensure the API is accessible:

- Digital Ocean droplets have a default firewall that allows SSH (port 22) and HTTP (port 80), but port 8000 (or your chosen port) may not be open. Log into your droplet via SSH and add a firewall rule:
- Use ufw allow 8000 to allow traffic on port 8000, or configure via the Digital Ocean control panel at Digital Ocean Firewalls.
- Your API should now be accessible at http://[droplet_ip]:8000/data, where [droplet_ip] is the IP address of your droplet, visible in the Digital Ocean dashboard.

5. Ensuring Persistence and Accessibility

The Plumber API service is set up to run persistently, and plumberDeploy handles starting it. However, ensure it restarts on server reboot by checking the service configuration (typically managed by systemd on Ubuntu droplets). You can verify with:

- sudo systemctl status plumber-myapi (replace “myapi” with your API name).

Technical Considerations

- Database File Location: The Plumber API code uses a relative path (“mydatabase.duckdb”) to connect to the database. On the droplet, this file is in “/var/plumber/[name]”, and the working directory is set to this location during API execution, ensuring the relative path works.

- DuckDB Dependencies: DuckDB is self-contained within the R package, but ensure the R version on the droplet (installed by plumberDeploy from CRAN) is compatible with your duckdb package version.
- Firewall and Security: Opening ports manually is necessary for accessibility, but consider security implications, such as restricting access to specific IP ranges if needed.
- SSH Access: Ensure your SSH key is properly configured for secure access to the droplet, which is crucial for post-deployment tasks like firewall configuration.

Table 2: **Deployment Checklist**

Step	**Action**	**Notes**
Local Setup	Install duckdb, create and populate database, write Plumber API	Test locally first
Prepare Directory	Place API code and database file in "myapi" directory	Ensure all files are included
Digital Ocean Setup	Create account, API key, and SSH key	Validate with analogsea::account()
Deploy API	Use plumberDeploy::do_deploy_api with install_packages = c("duckdb")	Specify port (e.g., 8000)
Post-Deployment	Add firewall rule for the port (e.g., ufw allow 8000)	Check accessibility at droplet IP:port
Verify Service	Ensure Plumber service is running and persistent	Use systemctl for status checks

Addressing the "Populate with plumberDeploy" Query

The phrase "populate it with plumberDeploy" is likely a misunderstanding, as plumberDeploy is for deployment, not database population. It's recommended to populate the DuckDB database locally before deployment, as shown in Step 1. If you need to populate the database via the API after deployment, you can add POST endpoints to insert data, but that's a separate step not handled by plumberDeploy.

Unexpected Detail: Manual Firewall Configuration

You might not expect that after deploying with plumberDeploy, you'll need to manually configure the firewall to allow traffic on your chosen port (e.g., 8000), as this isn't automated. This step is crucial for external accessibility and reflects Digital Ocean's security-first approach.

Final Thoughts

By following these steps, you can successfully place your DuckDB database, R code, and Plumber API on a Digital Ocean droplet, ensuring the database is pre-populated and the API is accessible. Remember to test locally first and handle post-deployment configurations like firewall rules for a smooth experience, especially ensuring SSH access is set up correctly for secure management.

References

- DuckDB R Package GitHub repository
- Plumber Package official documentation
- plumberDeploy Package GitHub repository
- Digital Ocean SSH Keys
- Digital Ocean Firewalls