

Launching RStudio Server with CondorBox

Kyuhan Kim

2025-01-29

Introduction

This guide provides a step-by-step approach to launching an **RStudio Server** within a Docker container using the **CondorBox** package. By containerizing RStudio Server, you ensure a **consistent development environment** across different users and projects, which is crucial for **reproducibility** in fisheries stock assessment workflows.

The launch script is available in the **examples** folder as `launch_rserver.R` or can be generated using the `CondorBox::generate_rserver_template()` function. For more information, visit the CondorBox GitHub repository at <https://github.com/PacificCommunity/ofp-sam-CondorBox>.

Prerequisites

Before you begin, ensure that the following prerequisites are satisfied:

- **Docker:** Installed and running on your machine.
- **RStudio Server:** Installed and accessible.
- **CondorBox Package:** Installed in R.

Installing and Loading CondorBox

Run the following R code to install and load the **CondorBox** package:

```
# Install remotes if not already installed
if (!requireNamespace("remotes", quietly = TRUE)) {
  install.packages("remotes")
}
```

```
# Install CondorBox from GitHub
remotes::install_github("PacificCommunity/ofp-sam-CondorBox")

# Load the CondorBox package
library(CondorBox)
```

Starting the RStudio Server

To launch the RStudio Server, use the `manage_rstudio_server()` function. This function handles the initiation of the server within a Docker container.

```
manage_rstudio_server(
  action = "start",
  image = "rocker/rstudio",      # Default RStudio Docker image
  host_port = 9999,             # Port to access RStudio in the browser
  password = "FAME"             # Login password for RStudio Server
)
```

Once the server is up and running, access it via your web browser by navigating to:

<http://localhost:9999>

Login Credentials:

- Username: rstudio
- Password: FAME

Stopping the RStudio Server

To stop the RStudio Server and release system resources, execute the following command:

```
manage_rstudio_server(action = "stop")
```

Cleaning Docker Resources

If you encounter issues such as being unable to log in, it's advisable to clean up Docker resources. Use the interactive cleaning function as shown below:

```
# Clean Docker resources interactively
clean_docker_resources_interactive()
```

Setting Up GitHub SSH Keys

For seamless interaction with GitHub repositories—such as cloning and committing directly from the RStudio Server—you need to configure SSH keys. This ensures secure and uninterrupted communication between your server environment and GitHub.

Steps to Configure SSH Keys:

- Access RStudio Server: Open your web browser and navigate to <http://localhost:9999>.
- Log in using the credentials provided earlier.
- Run the SSH Setup Function:

```
# Set up GitHub SSH keys
setup_github_ssh(
  email = "your_email@example.com",           # Replace with your GitHub email
  github_username = "your_github_username",   # Replace with your GitHub username
  github_token = "your_github_token_here"     # Replace with your GitHub Personal Access Token
)
```

- `email`: Your GitHub-associated email address.
- `github_username`: Your GitHub username.
- `github_token`: Your GitHub Personal Access Token. Ensure it has the necessary scopes for repository access.

Verify SSH Configuration:

After running the setup function, verify that SSH keys are correctly configured by attempting to clone a repository or by running `ssh -T git@github.com` in the RStudio terminal.

Troubleshooting

If you encounter any issues during the setup or operation of the RStudio Server, consider the following troubleshooting steps:

- Docker Not Running: Ensure Docker is active. Restart Docker if necessary.
- Port Conflicts: If port 9999 is in use, choose a different `host_port` when starting the server.
- Authentication Issues: Verify that the password is correctly set and that SSH keys are properly configured.
- Resource Limitations: Ensure your machine has sufficient resources (CPU, RAM) allocated for Docker containers.

Remarks

By following this guide, you have successfully launched an RStudio Server within a Docker container using the CondorBox package. This setup not only streamlines your development workflow but also enhances reproducibility and collaboration across your projects.