

# Package ‘plumberDeploy’

May 9, 2026

**Type** Package

**Title** Plumber Deployment

**Version** 0.2.1

**Depends** R (>= 3.0.0)

**Description** Gives the ability to automatically deploy a plumber API from R functions on 'DigitalOcean' and other cloud-based servers.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** analogsea (>= 0.9.4), ssh, jsonlite, lifecycle, utils

**BugReports** <https://github.com/meztez/plumberDeploy/issues>

**URL** <https://github.com/meztez/plumberDeploy>

**SystemRequirements** libssh >= 0.6.0 (the original, not libssh2)

**NeedsCompilation** no

**Author** Bruno Tremblay [cre, ctb],  
Jeff Allen [aut, ccp],  
John Muschelli [ctb] (ORCID: <<https://orcid.org/0000-0001-6469-1750>>),  
Barret Schloerke [ctb],  
Trestle Technology, LLC [cph, fnd]

**Maintainer** Bruno Tremblay <[cran@neoxone.com](mailto:cran@neoxone.com)>

**Repository** CRAN

**Date/Publication** 2021-03-22 09:40:05 UTC

## Contents

do_configure_https . . . . .	2
do_deploy_api . . . . .	3
do_forward . . . . .	4

do_provision . . . . .	5
do_remove_api . . . . .	6
do_remove_forward . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

do_configure_https	<i>Add HTTPS to a plumber Droplet</i>
--------------------	---------------------------------------

---

## Description

Adds TLS/SSL (HTTPS) to a droplet created using `do_provision()`.

## Usage

```
do_configure_https(  
    droplet,  
    domain,  
    email,  
    termsOfService = FALSE,  
    force = FALSE,  
    ...  
)
```

## Arguments

droplet	The droplet on which to act. See <code>analogsea::droplet()</code> .
domain	The domain name associated with this instance. Used to obtain a TLS/SSL certificate.
email	Your email address; given only to letsencrypt when requesting a certificate to enable them to contact you about issues with renewal or security.
termsOfService	Set to TRUE to agree to the letsencrypt subscriber agreement. At the time of writing, the current version is available <a href="#">here</a> . Must be set to true to obtain a certificate through letsencrypt.
force	If FALSE, will abort if it believes that the given domain name is not yet pointing at the appropriate IP address for this droplet. If TRUE, will ignore this check and attempt to proceed regardless.
...	additional arguments to pass to <code>analogsea::droplet_ssh()</code>

## Details

In order to get a TLS/SSL certificate, you need to point a domain name to the IP address associated with your droplet. If you don't already have a domain name, you can register one [here](#). Point a (sub)domain to the IP address associated with your plumber droplet before calling this function. These changes may take a few minutes or hours to propagate around the Internet, but once complete you can then execute this function with the given domain to be granted a TLS/SSL certificate for that domain.

Obtains a free TLS/SSL certificate from [letsencrypt](#) and installs it in nginx. It also configures nginx to route all unencrypted HTTP traffic (port 80) to HTTPS. Your TLS certificate will be automatically renewed and deployed. It also opens port 443 in the firewall to allow incoming HTTPS traffic.

Historically, HTTPS certificates required payment in advance. If you appreciate this service, consider [donating to the letsencrypt project](#).

## Value

The DigitalOcean droplet

---

do_deploy_api	<i>Deploy or Update an API</i>
---------------	--------------------------------

---

## Description

Deploys an API from your local machine to make it available on the remote plumber server.

## Usage

```
do_deploy_api(
  droplet,
  path,
  localPath,
  port,
  forward = FALSE,
  docs = FALSE,
  preflight,
  overwrite = FALSE,
  ...
)
```

## Arguments

droplet	The droplet on which to act. It's expected that this droplet was provisioned using <a href="#">do_provision()</a> . See <a href="#">analogsea::droplet()</a> to obtain a reference to a running droplet.
path	The remote path/name of the application
localPath	The local path to the API that you want to deploy. The entire directory referenced will be deployed, and the <code>plumber.R</code> file inside of that directory will be used as the root plumber file. The directory <b>MUST</b> contain a <code>plumber.R</code> file.
port	The internal port on which this service should run. This will not be user visible, but must be unique and point to a port that is available on your server. If unsure, try a number around 8000.
forward	If TRUE, will setup requests targeting the root URL on the server to point to this application. See the <a href="#">do_forward()</a> function for more details.

docs	If TRUE, will enable the documentation interface for the remotely deployed API. By default, the interface is disabled.
preflight	R commands to run after <code>plumb()</code> ing the <code>plumber.R</code> file, but before <code>run()</code> ing the plumber service. This is an opportunity to e.g. add new filters. If you need to specify multiple commands, they should be semi-colon-delimited.
overwrite	if an application is already running for this path name, and <code>overwrite = TRUE</code> , then <code>do_remove_api</code> will be run.
...	additional arguments to pass to <code>analogsea::droplet_ssh()</code> or <code>analogsea::droplet_upload()</code> , such as <code>keyfile</code> . Cannot contain <code>remote</code> , <code>local</code> as named arguments.

**Value**

The DigitalOcean droplet, but called for side effects

---

do_forward	<i>Forward Root Requests to an API</i>
------------	----------------------------------------

---

**Description**

Forward Root Requests to an API

**Usage**

```
do_forward(droplet, path, ...)
```

**Arguments**

droplet	The droplet on which to act. It's expected that this droplet was provisioned using <code>do_provision()</code> .
path	The path to which root requests should be forwarded
...	additional arguments to pass to <code>analogsea::droplet_upload()</code>

**Value**

The DigitalOcean droplet

---

do_provision	<i>Provision a DigitalOcean plumber server</i>
--------------	------------------------------------------------

---

### Description

Create (if required), install the necessary prerequisites, and deploy a sample plumber application on a DigitalOcean virtual machine. You may sign up for a Digital Ocean account [here](#). You should configure an account ssh key with `analogsea::key_create()` prior to using this method. This command is idempotent, so feel free to run it on a single server multiple times.

### Usage

```
do_provision(droplet, unstable = FALSE, example = TRUE, ...)
```

```
do_install_plumber(droplet, unstable, ...)
```

```
do_ip(droplet, path)
```

### Arguments

droplet	The DigitalOcean droplet that you want to provision (see <code>analogsea::droplet()</code> ).
unstable	If FALSE, will install plumber from CRAN. If TRUE, will install the unstable version of plumber from GitHub.
example	If TRUE, will deploy an example API named hello to the server on port 8000.
...	Arguments passed into the <code>analogsea::droplet_create()</code> function.
path	path to attach to the IP address before browsing. Should likely start with a / or : (as in <code>:8080</code> ), otherwise / will be added.

### Details

Provisions a Ubuntu 20.04-x64 droplet with the following customizations:

- A recent version of R installed
- plumber installed globally in the system library
- An example plumber API deployed at `/var/plumber`
- A systemd definition for the above plumber API which will ensure that the plumber API is started on machine boot and respawned if the R process ever crashes. On the server you can use commands like `systemctl restart plumber` to manage your API, or `journalctl -u plumber` to see the logs associated with your plumber process.
- The ‘nginx’ web server installed to route web traffic from port 80 (HTTP) to your plumber process.
- ufw installed as a firewall to restrict access on the server. By default it only allows incoming traffic on port 22 (SSH) and port 80 (HTTP).
- A 4GB swap file is created to ensure that machines with little RAM (the default) are able to get through the necessary R package compilations.

**Value**

The DigitalOcean droplet  
The URL to be browsed

**Note**

Please see <https://github.com/sckott/analogsea/issues/205> in case of an error by default do\_provision and an error of "Error: Size is not available in this region."

**Examples**

```
## Not run:
auth = try(analogsea::do_oauth())
if (!inherits(auth, "try-error") &&
    inherits(auth, "request")) {
  analogsea::droplets()
  droplet = do_provision(region = "sfo3", example = FALSE)
  analogsea::droplets()
  analogsea::install_r_package(droplet, c("readr", "remotes"))
  do_deploy_api(droplet, "hello",
                system.file("plumber", "10-welcome", package = "plumber"),
                port=8000, forward=TRUE)
  if (interactive()) {
    utils::browseURL(do_ip(droplet, "/hello"))
  }
  analogsea::droplet_delete(droplet)
}

## End(Not run)
```

---

do_remove_api	<i>Remove an API from the server</i>
---------------	--------------------------------------

---

**Description**

Removes all services and routing rules associated with a particular service. Optionally purges the associated API directory from disk.

**Usage**

```
do_remove_api(droplet, path, delete = FALSE, ...)
```

**Arguments**

droplet	The droplet on which to act. It's expected that this droplet was provisioned using <code>do_provision()</code> . See <code>analogsea::droplet()</code> to obtain a reference to a running droplet.
path	The path/name of the plumber service

delete	If TRUE, will also delete the associated directory (/var/plumber/whatever) from the server.
...	additional arguments to pass to <code>analogsea::droplet_ssh()</code> or <code>analogsea::droplet_upload()</code> , such as <code>keyfile</code> . Cannot contain <code>remote</code> , <code>local</code> as named arguments.

**Value**

The DigitalOcean droplet, but called for side effects

---

do\_remove\_forward      *Remove the forwarding rule*

---

**Description**

Removes the forwarding rule from the root path on the server. The server will no longer forward requests for / to an application.

**Usage**

```
do_remove_forward(droplet, ...)
```

**Arguments**

droplet	The droplet on which to act. It's expected that this droplet was provisioned using <code>do_provision()</code> . See <code>analogsea::droplet()</code> to obtain a reference to a running droplet.
...	additional arguments to pass to <code>analogsea::droplet_ssh()</code>

**Value**

The DigitalOcean droplet, but called for side effects

# Index

analogsea::droplet(), 2, 3, 5–7  
analogsea::droplet\_create(), 5  
analogsea::droplet\_ssh(), 2, 4, 7  
analogsea::droplet\_upload(), 4, 7  
analogsea::key\_create(), 5  
  
do\_configure\_https, 2  
do\_deploy\_api, 3  
do\_forward, 4  
do\_forward(), 3  
do\_install\_plumber (do\_provision), 5  
do\_ip (do\_provision), 5  
do\_provision, 5  
do\_provision(), 2–4, 6, 7  
do\_remove\_api, 6  
do\_remove\_forward, 7