Package ‘buoyant’

January 19, 2026

Type Package

Title Deploy '_server.yml' Compliant Applications to 'DigitalOcean'

Version 0.1.0

Description Provides tools to deploy R web server applications that follow the '_server.yml' standard. This standard allows different R server frameworks ('plumber2', 'fiery', etc.) to be deployed using a common interface. The package supports deployment to 'DigitalOcean' and includes validation tools to ensure '_server.yml' files are correctly formatted.

License MIT + file LICENSE

URL https://posit-dev.github.io/buoyant/, https://github.com/posit-dev/buoyant

BugReports https://github.com/posit-dev/buoyant/issues

Depends R (>= 4.1.0)

Imports analogsea (>= 0.9.4), jsonlite, renv, ssh, withr, yaml

Suggests knitr, quarto, spelling, testthat (>= 3.1.0)

VignetteBuilder quarto

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

Language en-US

NeedsCompilation no

Author Barret Schloerke [aut, cre] (ORCID: <https://orcid.org/0000-0001-9986-114X>), Posit Software, PBC [cph, fnd] (ROR: <https://ror.org/03wc8by49>)

Maintainer Barret Schloerke <barret@posit.co>

Repository CRAN

Date/Publication 2026-01-19 18:50:06 UTC

Contents

`do_configure_https` *Add HTTPS to a buoyant Droplet*

Description

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

Usage

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

Arguments

`droplet` The droplet on which to act. It's expected that this droplet was provisioned using `do_provision()`. See `analogsea::droplet()` to obtain a reference to a running droplet.

`domain` The domain name associated with this instance. Used to obtain a TLS/SSL certificate.

`email` Your email address; given to letsencrypt for "urgent renewal and security notices".

`terms_of_service` Set to `TRUE` to agree to the letsencrypt subscriber agreement. At the time of writing, the current version is available here. Must be set to true to obtain a certificate through letsencrypt.

`force` If `FALSE`, will error if the given domain name does not appear to be registered for this droplet according to DigitalOcean's Metadata service. If `TRUE`, will ignore any discrepancy and attempt to register anyway.

`...` additional arguments to pass to `analogsea::droplet_ssh()`, such as `keyfile`.

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 on Google Domains or Amazon Route53.

When sourcing a domain name, check if your registrar allows you to manage your own DNS records. If not, consider a service like CloudFlare to manage your DNS. DigitalOcean also offers DNS management.

Value

The DigitalOcean droplet

Examples

```
## Not run:
  droplet <- analogsea::droplet(123456)

  # Add HTTPS support with Let's Encrypt
  do_configure_https(
    droplet,
    domain = "myapp.example.com",
    email = "admin@example.com",
    terms_of_service = TRUE
  )

## End(Not run)
```

`do_deploy_server` *Deploy or Update a _server.yml Application*

Description

Deploys a _server.yml-based application from your local machine to make it available on the remote server.

Usage

```
do_deploy_server(
  droplet,
  path,
  local_path,
  port,
  forward = FALSE,
  overwrite = FALSE,
  ...,
  keyfile = do_keyfile(),
  r_packages = NULL
)
```

Arguments

`droplet` The droplet on which to act. It's expected that this droplet was provisioned using `do_provision()`. See `analogsea::droplet()` to obtain a reference to a running droplet.

`path` The remote path/name of the application

`local_path` The local directory path containing the `_server.yml` file. The entire directory will be deployed.

`port` The internal port on which this service should run. This will not be visible to visitors, 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 `do_forward()` function for more details.

`overwrite` if an application is already running for this `path` name, and `overwrite = TRUE`, then `do_remove_server` will be run.

`...` additional arguments to pass to `analogsea::droplet_ssh()` or `analogsea::droplet_upload()`. Cannot contain `remote`, `local`, `keyfile` as named arguments.

`keyfile` Path to private key for authentication. By default, uses the key for "digitalocean.com" from `ssh::ssh_key_info()`.

`r_packages` A character vector of R packages to install via `{pak}` on the server. When `NULL` (default), all dependencies found via `{renv}` will be installed.

Value

The DigitalOcean droplet

`do_forward` *Forward root requests to an application*

Description

Configures nginx to forward requests from the root path (/) to a specific application.

Usage

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

Arguments

droplet The droplet on which to act.

path The application path to forward root requests to.

... additional arguments to pass to analogsea::droplet_ssh().

Value

The DigitalOcean droplet

Examples

```
## Not run:
  droplet <- analogsea::droplet(123456)

  # Forward root URL to an application
  do_forward(droplet, "myapp")

  # Now visiting http://your-ip/ will redirect to http://your-ip/myapp

## End(Not run)
```

do_install_server_deps

Install server dependencies on a droplet

Description

Installs R and common dependencies needed for running _server.yml applications. This is called automatically by do_provision() but can be called separately if needed.

Usage

```
do_install_server_deps(droplet, keyfile = do_keyfile())
```

Arguments

droplet The DigitalOcean droplet that you want to provision (see analogsea::droplet()). If empty, a new DigitalOcean server will be created.

keyfile Path to private key for authentication. By default, uses the key for "digitalocean.com" from ssh::ssh_key_info().

Value

Invisibly returns NULL. Called for side effects.

Examples

```
## Not run:
  # Reinstall or update server dependencies on an existing droplet
  droplet <- analogsea::droplet(123456)
  do_install_server_deps(droplet)

## End(Not run)
```

do_ip *Get the URL to a deployed application*

Description

Returns the URL to access a deployed application or the droplet's IP address.

Usage

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

Arguments

droplet	The droplet on which to act.
path	Optional path to append to the IP address. If not provided, just returns the IP address.

Value

A character string with the URL or IP address.

do_keyfile *Get the default DigitalOcean SSH keyfile path*

Description

Returns the path to the SSH private key for "digitalocean.com" from `ssh::ssh_key_info()`. This is used as the default keyfile for all buoyant functions that interact with DigitalOcean droplets.

Usage

```
do_keyfile()
```

Value

A character string with the path to the SSH private key, or NULL if no key is found.

Examples

```
## Not run:
  # Get the default keyfile path
  do_keyfile()

## End(Not run)
```

`do_provision` *Provision a DigitalOcean server for _server.yml applications*

Description

Create (if required), install the necessary prerequisites, and deploy a _server.yml-based R server 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, ..., keyfile = do_keyfile())
```

Arguments

`droplet`	The DigitalOcean droplet that you want to provision (see `analogsea::droplet()`). If empty, a new DigitalOcean server will be created.
`...`	Arguments passed into the `analogsea::droplet_create()` function.
`keyfile`	Path to private key for authentication. By default, uses the key for "digitalocean.com" from `ssh::ssh_key_info()`.

Details

Provisions a Ubuntu 24.04-x64 droplet with the following customizations:

- A recent version of R installed
- Common server dependencies installed
- Directory structure at `/var/server-apps` for deployed applications
- The `nginx` web server installed to route web traffic from port 80 (HTTP)
- `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

Note

Please see https://github.com/pachadotdev/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 <- analogsea::do_oauth()

  analogsea::droplets()
  droplet <- do_provision(region = "sfo3")
  analogsea::droplets()

  # Deploy a _server.yml application
  do_deploy_server(
    droplet,
    "myapp",
    "local/path/to/app/",
    port=8000,
    forward=TRUE
  )
  if (interactive()) {
    utils::browseURL(do_ip(droplet, "/myapp"))
  }

  analogsea::droplet_delete(droplet)

## End(Not run)
```

---

do_remove_forward *Remove root forwarding*

Description

Removes any root path forwarding configuration.

Usage

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

Arguments

droplet	The droplet on which to act.
...	additional arguments to pass to analogsea::droplet_ssh().

Value

This function currently stops with an error. Not yet implemented.

Examples

```
## Not run:
  droplet <- analogsea::droplet(123456)

  # This function is not yet implemented
  do_remove_forward(droplet)

## End(Not run)
```

`do_remove_server` *Remove a deployed application*

Description

Removes a deployed _server.yml application from the server.

Usage

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

Arguments

`droplet` The droplet on which to act.

`path` The path/name of the application to remove.

`delete` If `TRUE`, also deletes the application files. If `FALSE`, just stops and disables the service.

`...` additional arguments to pass to `analogsea::droplet_ssh()`.

Value

The DigitalOcean droplet

Examples

```
## Not run:
  droplet <- analogsea::droplet(123456)

  # Stop the service but keep files
  do_remove_server(droplet, "myapp", delete = FALSE)

  # Remove the service and delete all files
  do_remove_server(droplet, "myapp", delete = TRUE)

## End(Not run)
```

read_server_yml *Read _server.yml configuration*

Description

Reads and parses a `_server.yml` file, returning the configuration as a list.

Usage

```
read_server_yml(path)
```

Arguments

path	Path to the directory containing `_server.yml` or path to the `_server.yml` file itself.

Value

A list containing the parsed YAML configuration.

Examples

```
## Not run:
config <- read_server_yml("path/to/api")
print(config$engine)

## End(Not run)
```

validate_server_yml *Validate a _server.yml file*

Description

Checks that a `_server.yml` file is properly formatted according to the _server.yml standard. This includes verifying that the engine field exists and that the specified engine package has a `launch_server()` function.

Usage

```
validate_server_yml(path, check_engine = FALSE, verbose = TRUE)
```

Arguments

`path` Path to the directory containing `_server.yml` or path to the `_server.yml` file itself.

`check_engine` Logical. If `TRUE`, checks that the engine package is installed and has a `launch_server()` function. Default is `FALSE` since the engine may not be installed locally but will be on the deployment target.

`verbose` Logical. If `TRUE`, prints validation progress messages.

Details

The validation checks:

- The `_server.yml` file exists
- The file is valid YAML
- The required `engine` field is present and is a character string
- If `check_engine = TRUE`, verifies the engine package is installed and has a `launch_server()` function

Value

Invisibly returns `TRUE` if validation passes. Throws an error if validation fails.

Examples

```
## Not run:
# Validate a directory containing _server.yml
validate_server_yml("path/to/api")

# Validate and check that the engine is installed
validate_server_yml("path/to/api", check_engine = TRUE)

## End(Not run)
```

Index