plumberDeploy

The `plumberDeploy` package separated the deployment and the `do_*` functions from `plumber`. The `plumberDeploy` package gives the ability to automatically deploy a plumber API from R functions on 'DigitalOcean' and other cloud-based servers.

Installation

You can install the released version of `plumberDeploy` from CRAN with (coming soon!):

```
install.packages("plumberDeploy")
```

And the development version from GitHub with:

```
# install.packages("remotes")
remotes::install_github("meztez/plumberDeploy")
```

Setup

If you're just getting started with hosting cloud servers, the DigitalOcean integration included in `plumberDeploy` will be the best way to get started. You'll be able to get a server hosting your custom API in just two R commands. Full documentation is available at https://www.rplumber.io/articles/hosting.html#digitalocean-1.

1. Create a DigitalOcean account
2. Install `plumberDeploy`. Validate your account with `analogsea::account()`.
3. Configure an ssh key for the Digital Ocean account before using methods included in this package. Use `analogsea::key_create` method or see https://www.digitalocean.com/docs/droplets/how-to/add-ssh-keys/to-account/.
4. Run a test command like `analogsea::droplets()` to confirm that it's able to connect to your DigitalOcean account.
5. Run `mydrop <- plumberDeploy::do_provision()`. This will start a virtual machine (or "droplet", as DigitalOcean calls them) and install Plumber and all the necessary prerequisite software. Once the provisioning is complete, you should be able to access port `8000` on your server's IP and see a response

should be able to access port `8000` on your server's IP and see a response from Plumber.

6. Install any R packages on the server that your API requires using `analogsea::install_r_package()`.
7. You can use `plumberDeploy::do_deploy_api()` to deploy or update your own custom APIs to a particular port on your server.
8. (Optional) Setup a domain name for your Plumber server so you can use www.myplumberserver.com instead of the server's IP address.
9. (Optional) Configure SSL

Getting everything connected the first time can be a bit of work, but once you have `analogsea` connected to your DigitalOcean account, you're now able to spin up new Plumber servers in DigitalOcean hosting your APIs with just a couple of R commands. You can even write scripts that provision an entire Plumber server with multiple APIs associated.

Your ssh key needs to be available on your local machine too. You can check this with `ssh::ssh_key_info()`. Validate that one of the public keys can be found in `lapply(analogsea::keys(), '[[', "public_key")`.

Deploy an api to a new droplet

`.api/plumber.R`

```
#* @get /
function() {
  Sys.Date()
}
```

Then run this code

```
id <- plumberDeploy::do_provision(example = FALSE)
# About 10 minutes
# STOP Make sure every packages the api depends on is available on the d
plumberDeploy::do_deploy_api(id, "date", "./api/", 8000, docs = TRUE)
```

Navigate to: `[[IPADDRESS]]/date/__docs__/`

Other useful commands

```
# Install package to your droplet
analogsea::install_r_package(droplet, c("readr", "remotes"))
# Install system dependencies to your droplet
```

```
# Install system dependencies of your droplet
analogsea::debian_apt_get_install(droplet, "libssl-dev","libsodium-dev",
```

R Packages installed on linux systems sometimes require system packages. Check console output carefully to see if a package was installed successfully.

Otherwise read the doc on functions, ask questions in the RStudio community or report issues to our github issue tracker.