

When talking about languages to use in Production in data science, R is usually not part of the conversation and if it is, it's referenced as a secondary language. One of the main reasons this occurs is because R it's commonly associated with being more suitable for statistical analysis and languages like Python and JavaScript, more suitable for doing other tasks such as creating web applications or implementing machine learning models. However, one realm where R's capabilities haven't been explored to the maximum is Systems Administration.

At Jumping Rivers we make use of R as our main tool for doing tasks related to Systems Administration. The main way in which we implement our solutions is by dividing one package per service and then developing the specific functions to manage it. One of these packages that we have developed is named jrDroplet.

jrDroplet

jrDroplet is a package designed specifically to manage Virtual Machines in Digital Ocean for our training courses. The idea is that with a single line we are able to create a Digital Ocean droplet with the packages installed for our courses, hiding all of the background complexities related to infrastructure. Below is an overview of our `create_droplet` function, reduced slightly for simplicity:

```
create_droplet <- function(client_name,
                           droplet_name,
                           vm_size,
                           ssh_keys,
                           image_base,
                           region,
                           sub_domain,
                           dns_root)
{
  image <- get_latest_training_snapshot(region = region,
                                       base = image_base)[[1]]

  message(paste0("Using image ", image$name))

  analogsea::droplet_create(
    name = droplet_name,
    region = region,
    ssh_keys = ssh_keys,
    size = vm_size,
    image = image$id)

  droplets <- analogsea::droplets()

  message('Waiting for IP address to be assigned to VM')

  ip_address <- droplets[[droplet_name]]$networks$v4[[1]]$ip_address

  dr <- analogsea::domain_record_create(
    domain = dns_root,
    type = 'A',
    name = sub_domain,
    data = ip_address
  )
}
```

This function takes a set of given arguments and proceeds to do a number of steps that would be required to be done manually in the Digital Ocean interface. I will explain below what is happening in the code and what would be the equivalent in the interface.

```
image <- get_latest_training_snapshot(region = region,
                                     base = image_base)[[1]]
```

In this code chunk we are obtaining the latest snapshot created in the Jumping River's Digital Ocean organization, searching by base image, meaning searching if the R image or the Python image. These base images are built using a tool named Packer, however implementation details of this process will come in a future post. The equivalent of this code chunk would be when creating a droplet, to select the Snapshots tab and manually pick the training snapshot.

```
analogsea::droplet_create(
  name = droplet_name,
  region = region,
  ssh_keys = ssh_keys,
  size = vm_size,
  image = image$id)
```

In this code chunk we are using the package `analogsea` which is the backbone of our `jrDroplet` package. `analogsea` is a package to manage Digital Ocean infrastructure through the API and following Open Source principles, we are building on it for our specific use case. In this case, we are using the `droplet_create()` function to create the Droplet for our training VM with the desired parameters, and based on the latest training snapshot.

```
droplets <- analogsea::droplets()

ip_address <- droplets[[droplet_name]]$networks$v4[[1]]$ip_address

dr <- analogsea::domain_record_create(
  domain = dns_root,
  type = 'A',
  name = sub_domain,
  data = ip_address
)
```

This is the final code chunk we are going to discuss in this post. What we are doing here is first listing all of the available droplets to then search for the IP address of the droplet created. We need this IP address for the function `domain_record_create`. Very briefly, a Domain Record is a record connecting a specific name to a specific IP address, and they are stored in Domain Name Services. So in this command, we are taking the IP address and using our base DNS root name to create a new subdomain specifically for this new droplet. If we were to do this through the DO interface we would need to go to the Networking Tab and select things from Dropdown menus.

This is just one example of how we use R as a tool for Systems Administration. Another tool we have created is called `monitR` and it's a package to monitor the full stack of services that might be offered to a specific client. This tool has the back-end functions to manage the data, building upon existing system administration tools and frameworks. It also has a Shiny dashboard that allows us to visualize all the data for our clients. In conclusion, R has many uses aside from the classical statistical analyses and shouldn't be limited as a language solely for Data Scientists.