Guide to Create Virtual Servers on Digital Ocean using Cloud-Init for Arch Linux Users

Introduction

In this guide, we'll walk through the process of creating a Arch Linux droplet on DigitalOcean using cloud-Init. Ensuring a smooth and automated setup for your virtual servers! During this process, you will learn how to use doctl, SSH keys, API tokens, and cloud-init.

DigitalOcean is a cloud provider that offers the ability to deploy virtual servers, known as *droplets*, and other cloud-based resources like databases, storage, and networking tools (DigitalOcean, 2024). When creating a droplet, we can automate the setup process to save time and create servers with consistent repeatable configurations using Cloud-Init.

Cloud-init is an industry standard tool that allows you to automate the initialization of your Linux instances. This means that you can use a cloud-init file to configure droplets at deployment that automatically sets up things like new users, firewall rules, app installations, and SSH keys (DigitalOcean, 2024). This tutorial uses doctl to add a cloud-init file with information on users, programs, and your SSH key.

Prerequisites

- A computer running Arch Linux
- Neovim installed on your system
- The Arch Linux disk image
- All code provided should be run through the Terminal

Table of Contents

How to Create an SSH Key pair

For the initial set up, you will begin by creating a Secure Shell Protocol (SSH) key pair using our local machine. SSH key pairs are a more secure alternative to traditional password-based authentication by using Public Key

Cryptography (DigitalOcean, 2024).

In this section, we will be using the terminal to create two plain text files in the `.ssh` directory that will be our keys.

- We will create a "hw-key" as our private key
- And we will create "hw-key.pub" as our public key

1. Copy then run the following code and after changing the appropriate information:

NOTE: You will need to change "*youremail@email.com*" to your actual information.

```
ssh-keygen -t ed25519 -f ~/.ssh/hw-key -C "youremail@email.com"
```

What does this code mean?

- `ssh-keygen`: Generates the public and private key pair (DigitalOcean, 2024)
- `-t`: Type of encryption for the key
- `-f`: Specifying filename and location
- `-C`: To add a comment

Successful key creation will look something like this:

Image of the SSH key making confirmation

How to Install Doctl

`doctl` is the official DigitalOcean command line interface (CLI) and it allows you to interact with the DigitalOcean API via the command line (DigitalOcean, 2020).

This section will teach you how to:

- Install doctl
- Create an API token
- Grant Account Access to Doctl with an API Token
- Validate That Doctl is Working

Install Doctl

1. Copy and run the following code to download doctl

```
sudo pacman -S doctl
```

What does this code mean?

- `sudo`: Allows a user to execute a command as the root user
- `pacman`: The package manager for Arch Linux (Arch Linux, n.d.)
- `-S`: It stands for synchronize and is used install or update packages from official repositories

Create an API Token

This step will be done on DigitalOcean to create a new API token for your account. An API token is a unique identifier used to authenticate and authorize requests to an Application Programming Interface (API). It acts as a digital key that allows applications, scripts, or users to interact with an API securely without needing to provide credentials like usernames and passwords directly (DigitalOcean, 2024).

It's important that the API token has both read and write access. It needs write access to be able to create your droplet, otherwise it would only be able to gather information.

NOTE: The API token string is only displayed once, so be sure to save it in a safe place for late use

1. Click **API** on the left side menu
2. Click the **Generate New Token**

A New Personal Access Token page will appear and you will need to fill out the following fields:

Image of the API settings

3. Type a name for the token
4. Choose when the token expires
5. Select **Full Access** (this will give the API token both read and write access)
6. Click **Generate Token**
7. Save your API token somewhere safe for later use

Granting Account Access to Doctl with an API Token

1. Copy and run the following code to connect your API token

NOTE: Be sure to give this authentication a name by changing "NAME"

```
doctl auth init --context NAME
```

What does this code mean?

- `auth`: Used to manage authentication with DigitalOcean
- `init`: Initializes the authentication process by prompting you to enter a DigitalOcean API token (DigitalOcean, 2024)
- `--context`: Allows you to save the authentication settings under a specific context name

2. Enter in the API token string (the one you made earlier) when prompted by `doctl auth init`
3. Copy and run the following code to switch to the correct authenticated account

NOTE: Change "NAME" to the name of the account you want to switch to that appears after `doctl auth list`

```
doctl auth list
doctl auth switch --context NAME
```

What does this code mean?

- `list`: Displays a list of all the saved authentications
- `switch`: Switch between different authentications

Validate that doctl is working

1. Copy and run the following code to confirm you have successfully authorized doctl

```
doctl account get
```

What does this code mean?

- `account`: Allows you to manage or retrieve information about the account
- `get`: Retrieves information about the currently authenticated account

Successful output looks like this:

Image of doctl validation confirmation

How to add your public key to your DigitalOcean account

You will now add your SSH keys to your DigitalOcean account using doctl. You will be using terminal commands to add your new public key text file.

1. Copy and run the following code to upload your public key to your DigitalOcean account

Note: You will need to change "git-user" to your desired key name and "~/.ssh/hw-key.pub" to your public key file location

```
doctl compute ssh-key import git-user --public-key-file ~/.ssh/hw-key.pub
```

What does this code mean?

- `compute`: Subcommand for managing DigitalOcean Droplets and other compute-related resources
- `import`: To import our SSH key
- `--public-key-file`: Specifies the location of the public key file (DigitalOcean, 2024)

NOTE: A successful import will look like this:

Image of the public key import to DigitalOcean

2. Copy and run the following code to verify a successful import

```
doctl compute ssh-key list
```

- `list`: Display a list

If you need to get the contents of your public key, you can run the following code:

```
cat ~/.ssh/hw-key.pub
```

What does this code mean?

- `cat`: Short for concatenate. It is used to read and output the contents of a file to the terminal

---

How to Create a Droplet on DigitalOcean

This step will teach about creating your droplet on the DigitalOcean using doctl. Droplets are Linux-based virtual machines (VMs) that run on top of virtualized hardware.

This section will teach you how to:

- Upload an Arch Linux Image to DigitalOcean
- Set up Cloud-Init
- Create a new Arch Linux Droplet

Upload an Arch Linux Image to DigitalOcean

You will be uploading an Arch Linux image on the DigitalOcean website for your droplet. It's essentially a snapshot that contains information on everything from the files and folders to the operating system and boot information (DigitalOcean, 2024). This image will be the basis for the droplet to be built with.

You can download the Arch Linux image here.

1. Click the **Manage** dropdown menu in the left side menu
2. Select **Backups & Snapshots**
3. Select **Custom Images**
4. Click the blue **Upload Image** button.
5. Upload the Arch Linux image

NOTE: After clicking upload, a new settings box will open and you will need to select the following settings

6. Select **Arch Linux** in the Distribution dropdown menu
7. Select **San Francisco 3** in the Choose a Datacenter Region Section
    - We choose San Francisco 3 as our data center in the example because it is the closest to our location
8. Click **Upload Image** to finish

Setting up Cloud-Init

Cloud-init will allow you to set up a server with some initial configurations. For this guide, the example includes packages in the cloud-init configuration that are some examples of commonly used packages (DigitalOcean, 2022).

1. Copy and run the following code to create your cloud-config.yaml file

```
nvim cloud-config.yaml
```

2. Copy and paste the following code

NOTE: Use "ctrl + shift + v" to paste with the correct formatting

```
#cloud-config
users:
    name: user-name #change me
    primary_group: user-group # change me
    groups: wheel
    shell: /bin/bash
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    ssh-authorized-keys:
      - ssh-ed25519 your-ssh-public-key # change this
packages:
  - ripgrep
  - rsync
  - neovim
  - fd
  - less
  - man-db
  - bash-completion
  - tmux
disable_root: true
```

What does this code mean?

- `shell: /bin/bash`: Sets the default shell to bash/bin
- `sudo: ['ALL=(ALL) NOPASSWD:ALL']`: Grants the user full sudo privileges without needing to enter a password
- `ssh-authorized-keys`: Allows the user to log in via SSH using public key authentication (DigitalOcean, 2014)
- `disable_root: true:`: Disables root login for the server (DigitalOcean, 2014)

3. Press the "i" key after pasting the text to enter insert mode to edit the file contents
4. Change the information (at least the ssh-authorized-keys)
5. Press "esc" key to exit inset mode
6. Type `:` then `wq` and then press enter key to save and finish

Create a New Arch Linux Droplet

Back in your terminal, you will be running the following `doctl` command to create the Droplets.

1. Copy and paste the following code into your terminal

```
doctl compute droplet create --image 165084633 --size s-1vcpu-1gb-amd --region
sfo3 --ssh-keys 43491384 --user-data-file ~/cloud-config.yaml --wait
exampleDroplet
```

What does this code mean?

- `doctl compute droplet create`: The command doctl requires to create Droplets (DigitalOcean, 2024)
- `--image`: The OS image used to create the Droplet. For this example, the Droplet uses Arch Linux
- `--size s-1vcpu-1gb`: The number of processors and the amount of RAM each Droplet has. In this case, each Droplet has one processor and 1 GB of RAM. The example chooses a low amount but you will want to choose an appropriate amount based on what you'll be doing with the droplet (do-api.dev, n.d.)
- `--region sfo3`: The region to create the Droplets in. In this example, doctl deploys the Droplets into the San Francisco 3 datacenter region because it's the closest location. Choose your data center for whichever is the closest location
- `--ssh-keys`: The SSH keys to import into the Droplet from your DigitalOcean account. You can retrieve a list of available keys by running `doctl compute ssh-key list`
- `--user-data-file <path-to-your-cloud-init-file>`: Specifies the path to your cloud-config.yaml file. For example, `~/cloud-config.yaml`. (DigitalOcean, 2024)

Successful droplet creation looks like:

Image of the completed droplet creation

How to Connect to Your Droplet Using SSH

1. Open your Terminal
2. Run the following code to create a config file:

```
nvim config
```

3. Copy the following code into your new config file

```
Host arch
  HostName 24.144.89.149 #change IP address to the IP address of your droplet
  User arch #change "arch" to match your name in your cloud-init.yaml file
  PreferredAuthentications publickey
  IdentityFile ~/.ssh/hw-key #change "hw-key" to match the name of your private SSH key
  StrictHostKeyChecking no
  UserKnownHostsFile /dev/null
```

What does this code mean?

- `Host`: The host alias you use when you run the SSH command
- `HostName`: The IP address of the droplet you're connecting to
- `User`: The username to be used when connecting to the remote host
- `PreferredAuthentications publickey`: Specifies that you should use public key authentication to login
- `IdentityFile`: The path to the private SSH key on your local machine
- `StrictHostKeyChecking no`: Disables host key checking
- `UserKnownHostsFile`: Tells SSH not to store the server's host key in the known hosts file

4. Copy the IP address of the droplet from DigitalOcean

Image of completed droplet

5. Change the **HostName** to the IP address of your droplet
6. Save the Config file to finish

You're now ready to connect to your droplet using SSH! You can now connect to your droplet by using `ssh arch`. This will depend on what you named your host in the previous steps.

References

DigitalOcean. (2024, September). Droplets documentation. DigitalOcean. https://docs.digitalocean.com/products/droplets/

DigitalOcean. (2022, September). Automate Droplet setup with cloud-init. DigitalOcean. https://docs.digitalocean.com/products/droplets/how-to/automate-setup-with-cloud-init/

DigitalOcean. (2020, April). Install doctl: The DigitalOcean command-line client. DigitalOcean. https://docs.digitalocean.com/reference/doctl/how-to/install/

DigitalOcean. (2024, August). Create a personal access token. DigitalOcean. https://docs.digitalocean.com/reference/api/create-personal-access-token/

Cloud-init. (n.d.). Introduction to cloud-init. Cloud-init. https://docs.cloud-init.io/en/latest/explanation/introduction.html

Cloudflare. (n.d.). What is SSH?. Cloudflare. https://www.cloudflare.com/learning/access-management/what-is-ssh/

DigitalOcean. (n.d.). Droplet list using doctl reference. DigitalOcean. https://docs.digitalocean.com/reference/doctl/reference/compute/droplet/list/

DigitalOcean. (2014, October). How to use cloud-config for your initial server setup. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-use-cloud-config-for-your-initial-server-setup

DigitalOcean. (2024, April). SSH essentials: Working with SSH servers, clients, and keys. DigitalOcean. https://www.digitalocean.com/community/tutorials/ssh-essentials-working-with-ssh-servers-clients-and-keys

Arch Linux. (n.d.). Cloud-init. Arch Linux. https://wiki.archlinux.org/title/Cloud-init

Arch Linux. (n.d.). Pacman. Arch Linux. https://wiki.archlinux.org/title/Pacman

do-api.dev. (n.d.). Slugs API documentation. https://slugs.do-api.dev/

DigitalOcean. (2024, May). Custom images. DigitalOcean. https://docs.digitalocean.com/glossary/custom-images/

Arch Linux. (n.d.). Arch manual pages. Arch Linux. https://man.archlinux.org/