



# Using Ansible to create droplets on Digital Ocean:

**First test your account on Digital Ocean for creating Linux instance manually and then automated the process using Ansible.**

Create an API token on Digital Ocean, using below method.

Next, in the top menu, click on API.

In the Personal Access Tokens section, click the Generate new token button:

This screenshot shows the 'Personal Access Tokens' section of the DigitalOcean API dashboard. It includes a title 'Personal Access Tokens', a subtitle 'Tokens you have generated to access the DigitalOcean API', and a 'Generate new token' button in the top right corner.

You will be taken to the New Personal Access Token screen:

This screenshot shows the 'New Personal Access Token' form. It has a title 'New Personal Access Token' and a close button (X) in the top right. The form contains a 'Token Name' section with a text input field containing 'token' and a green checkmark. Below this is a 'Select Scopes' section with two checkboxes: 'Read (Default)' and 'Write (Optional)', both of which are checked. At the bottom of the form is a large blue button labeled 'Generate Token'.

Here, provide the following information:

Your desired token name (for your own reference)



Select the scope for this token (read or read/write)

Then click the Generate Token button.

Your token will be generated and presented to you on your Personal Access Tokens page. The actual token is the long string of numbers and letters, under the name:



Be sure to record your personal access token now. It will not be shown again, for security purposes.

Using the API:

```
export TOKEN=<token-id>
```

**Try creating droplets on the Digital Ocean cloud manually using below steps.**

### **Example: Create a New Droplet**

To create a new Droplet, send a POST request to `/v2/droplets`. For a full list of attributes that must be set to successfully create a droplet, see the full documentation. The following example creates an Ubuntu 14.04 droplet called "My-Droplet" in the NYC 2 data center, with 512MB RAM:

```
curl -X POST "https://api.digitalocean.com/v2/droplets" \
-d '{"name": "My-Droplet", "region": "nyc2", "size": "512mb", "image": "ubuntu-14-04-x64"}' \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json"
```

**Note:** This request, like any other request that makes a change to your account, requires that your token has "write" scope assigned to it.

### **Example: Create Multiple Droplets**

You can also create multiple Droplets with the same attributes using a single API request by sending a POST to `/v2/droplets`. Instead of providing a single name in the request, provide an array of names. The following example creates two Ubuntu 14.04 Droplets, one called "sub-01.example.com" and one called "sub-02.example.com". They both are in the NYC 2 data center, with 512MB RAM:

```
$ curl -X POST "https://api.digitalocean.com/v2/droplets" \
-d '{"names": ["sub-01.example.com", "sub-02.example.com"], "region": "nyc2", "size": "512mb", "image": "ubuntu-14-04-x64"}' \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json"
```



## Using ansible to provision Droplet:

Configure Ansible,

1. `sudo apt-get install python-pip`
2. `sudo pip install 'dopy>=0.3.5,<=0.3.5'` .... Install '*dopy*'
3. create a new directory and move the '*ansible.cfg*' file.
4. Update the *.cfg* file with below entry.,

```
[defaults]
hostfile = hosts
```

update the hosts file with below entries, here the group name is

```
[digitalocean]
localhost ansible_connection=local
```

There are three ways we can tell Ansible about the API token:

1. Provide it directly on each DigitalOcean task, using the `api_token` parameter.
2. Define it as a variable in the playbook or hosts file, and use that variable for the `api_token` parameter.
3. Export it as an environment variable, as either `DO_API_TOKEN` or `DO_API_KEY`.

Now define the Digital\_Ocean resources in the playbook as written below.

---

```
digitalocean.yml
```

```
---
- hosts: digitalocean

  vars:
    do_token:
186a6fd279015640995dce8938ed632020b2c9e6513af57c9cd7473217e280c1
    droplets:
      - droplet-one
      - droplet-two

  tasks:
    - name: ensure ssh key exists
      user: >
```



```
name={{ ansible_user_id }}
generate_ssh_key=yes
ssh_key_file=.ssh/id_rsa

- name: ensure key exists at DigitalOcean
  digital_ocean: >
    state=present
    command=ssh
    name=my_ssh_key
    ssh_pub_key={{ lookup('file', '~/.ssh/id_rsa.pub') }}
    api_token={{ do_token }}
    register: my_ssh_key

- name: ensure droplets exist
  digital_ocean: >
    state=absent
    command=droplet
    name={{ item }}
    unique_name=yes
    size_id=512mb
    region_id=blr1
    image_id=ubuntu-14-04-x64
    ssh_key_ids={{ my_ssh_key.ssh_key.id }}
    api_token={{ do_token }}
    with_items: "{{ droplets }}"
    register: droplet_details
```

---

Run below command to create Droplet on Digital Ocean using Ansible.

```
$ ansible-playbook digitalocean.yml
```