

Digital ocean API: <https://developers.digitalocean.com/documentation/v2/>

DROPLETS:

A [Droplet](#) is a Digital Ocean virtual machine. By sending requests to the Droplet endpoint, you can list, create, or delete Droplets.

Some of the attributes will have an object value. The region and image objects will all contain the standard attributes of their associated types. Find more information about each of these objects in their respective sections.

-cn	Creating a node.
--conf	Config files path.
-dn	Deleting a node.
-ln	Listing node or nodes.
-pon	Power on for a node or nodes.
-pfn	Power off for a node or nodes.
-sn	Shutdown a node.
-cs	Creating a snapshot.
-ds	Deleting a snapshot.
-ls	Listing a snapshot or snaphots.
-hc	Health check.
-lf	Listing a floating ips.
-cf	Creating a floating ips.
-df	Deleting a floating ip.
-af	Assigning a floating ip.
-uf	Unassigning a floating ip.
-fip	Floating ip.
-sc	Status check.
-aid	Action id.

CREATE A DROPLET:

SINGLE-DROPLET:

For creation of a single Droplet.

Templet: `python automate.py -cn single --conf <PATH_OF_CONF_FILE> -n <DROPLET_NAME>`

Example: `python automate.py -cn single --conf ./config_file/conf_files/create_single.conf -n drop1`

MULTIPLE-DROPLET:

To create multiple Droplets, creating multiple Droplets is very similar to creating a single Droplet, but instead of sending name as a string, send names as an array of strings. A Droplet will be created for each name you send using the associated information. Up to ten Droplets may be created at a time. The possible fields are:

Templet: `python automate.py -cn count --conf <PATH_OF_CONF_FILE> -n <DROPLET_NAME>`

Example: `python automate.py -cn multi --conf ./config_file/conf_files/create_single.conf -n systembox workbox`

FROM A SNAPSHOT:

Templet: `python automate.py -cn single --conf <PATH_OF_CONF_FILE> -n <DROPLET_NAME> -sid <SNAPSHOT_ID>`

Example: `python automate.py -cn single --conf ./config_file/conf_files/create_single.conf -n drop1-sid 5486321`

DELETEING A DROPLET:

To delete a Droplet, send a DELETE request to `/v2/droplets/$DROPLET_ID`.

No response body will be sent back, but the response code will indicate success. Specifically, the response code will be a 204, which means that the action was successful with no returned body data.

DELETE A SINGLE DROPLET:

Templet: `python automate.py -dn single -n <DROPLET_NAME>`

Example: `python automate.py -dn single -n drop1`

DELETE A MULTIPLE DROPLET:

Templet: `python automate.py -dn all`

Templet: `python automate.py -dn multi -n <DROPLET_NAMES>`

Example: `python automate.py -dn multi -n systembox workbox`

LIST OF DROPLETS:

To list all Droplets in your account

Templet: `python automate.py -ln all`

Templet: `python automate.py -ln single -n <DROPLET_NAME>`

Example: `python automate.py -ln single -n drop1`

POWER ON DROPLETS:

To power on a Droplet, send a POST request to `/v2/droplets/$DROPLET_ID/actions`. Set the "type" attribute to power on.

Templet: `python automate.py -pon all`

Templet: `python automate.py -pon single -n <DROPLET_NAME>`

Example: `python automate.py -pon single -n drop1`

POWER OFF DROPLETS:

To power off a Droplet, send a POST request to `/v2/droplets/$DROPLET_ID/actions`. Set the "type" attribute to power off.

A power off event is a hard shutdown and should only be used if the shutdown action is not successful. It is similar to cutting the power on a server and could lead to complications.

Templet: `python automate.py -pfn all`

Templet: `python automate.py -pfn single -n <DROPLET_NAME>`

Example: `python automate.py -pfn single -n drop1`

SHUTDOWN DROPLETS:

To shutdown a Droplet, send a POST request to `/v2/droplets/$DROPLET_ID/actions`. Set the "type" attribute to shutdown.

A shutdown action is an attempt to shutdown the Droplet in a graceful way, similar to using the shutdown command from the console. Since a shutdown command can fail, this action guarantees that the command is issued, not that it succeeds. The preferred way to turn off a Droplet is to attempt a shutdown, with a reasonable timeout, followed by a power off action to ensure the Droplet is off.

Templet: `python automate.py -sn all`

Templet: `python automate.py -sn single -n <DROPLET_NAME>`

Example: `python automate.py -sn single -n drop1`

CREATE SNAPSHOT:

To snapshot a Droplet, send a POST request to `/v2/droplets/$DROPLET_ID/actions`. Set the "type" attribute to snapshot and the "name" attribute to the name you would like to give the created image.

Templet: `python automate.py -cs single -n <DROPLET_NAME>`

Example: `python automate.py -cs single -n drop1`

DELETE SNAPSHOTS:

To delete a snapshot, send a DELETE request to `/v2/snapshots/$SNAPSHOT_ID`.

A status of 204 will be given. This indicates that the request was processed successfully, but that no response body is needed.

Templet: `python automate.py -ds all`

delete a snapshot belonging to an active droplet:

Templet: `python automate.py -ds single -n <DROPLET_NAME>`

Example: `python automate.py -ds single -n drop1`

delete a snapshot of a disabled/inactive/deleted droplet with it's imageid/snapshotid(run -ls all to know the snapshotid/imageid):

Templet: `python automate.py -ds single -im <SNAPSHOT_ID>`

Example: `python automate.py -ds single -im 5486321`

LIST OF SNAPSHOTS:

To list all of the snapshots available on your account, send a GET request to `/v2/snapshots`.

The response will be a JSON object with a key called snapshots. This will be set to an array of snapshot objects, each of which will contain the standard snapshot attributes:

Templet: `python automate.py -ls all`

Templet: `python automate.py -ls single <DROPLET_NAME>`

Example: `python automate.py -ls single drop1`

CREATE AND COPY HOSTS & IP LIST FILES :

Templet: `python automate.py -etc all -usr woclo -cmd "python /opt/woclo/bin/ip.py" -pwd`

HEALTH_CHECK OR RUN A CMD REMOTELY IN NODES:

Templet: `python automate.py -hc all -cmd <CMD> -pwd <PASSWORD>`

Templet: `python automate.py -hc single -cmd <CMD> -pwd <PASSWORD> -n <DROPLET_NAME>`

Example: `python automate.py -hc single -cmd <CMD> -pwd abcd123 -n drop1`

LIST OF FLOATING_IPS:

Floating IP objects represent a publicly-accessible static IP addresses that can be mapped to one of your Droplets. They can be used to create highly available setups or other configurations requiring movable addresses.

Floating IPs are bound to a specific region.

Templet: `python automate.py -lf all`

Templet: `python automate.py -lf single -n node-master`

CREATE A FLOATING_IP FOR A DROPLET:

To assign a Floating IP to a Droplet, send a POST request to `/v2/floating_ips/$FLOATING_IP_ADDR/actions`. Set the "type" attribute to assign and the "droplet_id" attribute to the Droplet's ID.

Templet: `python automate.py -cf single -n <DROPLET_NAME>`

Example: `python automate.py -cf single -n drop1`

DELETE AN ASSIGNED OR UNASSIGNED FLOATING_IP:

To delete a Floating IP and remove it from your account, send a DELETE request to `/v2/floating_ips/$FLOATING_IP_ADDR`.

No response body will be sent back, but the response code will indicate success. Specifically, the response code will be a 204, which means that the action was successful with no returned body data.

IF ASSIGNED TO A DROPLET:

Templet: `python automate.py -df single -n <DROPLET_NAME>`

Example: `python automate.py -df single -n drop1`

IF NOT ASSIGNED TO A DROPLET:

Templet: `python automate.py -df single -fip <FLOATING_IP>(use -lf all for list of floating_ip's)`

Example: `python automate.py -df single -fip 45.55.54.63(use -lf all for list of floating_ip's)`

ASSIGNING AN EXISTING FLOATING_IP TO A DROPLET:(yesterday's floating_ip):

To assign a Existing Floating IP to a Droplet.

Templet: `python automate.py -af single -n <DROPLET_NAME> -fip <FLOATING_IP>`

Example: `python automate.py -af single -n drop1 -fip 45.55.54.63`

UNASSIGNING A DROPLETS FLOATING_IP:

To unassign a Floating IP, send a POST request to `/v2/floating_ips/$FLOATING_IP_ADDR/actions`. Set the "type" attribute to unassign. The Floating IP will be reserved in the region but not assigned to a Droplet.

Templet: `python automate.py -uf single -fip <FLOATING_IP>`

Example: `python automate.py -uf single -fip 45.55.54.63`

CHECKING STATUS OF AN ACTION PERFORMED ON FLOATING_IP:(these actions include assigning,unassigning etc):

you can get the action id from the response when an action is performed.

Templet: `python automate.py -sc single -fip <FLOATING_IP> -aid <ACTION_ID>`

Example: `python automate.py -sc single -fip 45.55.54.63 -aid 12345678`