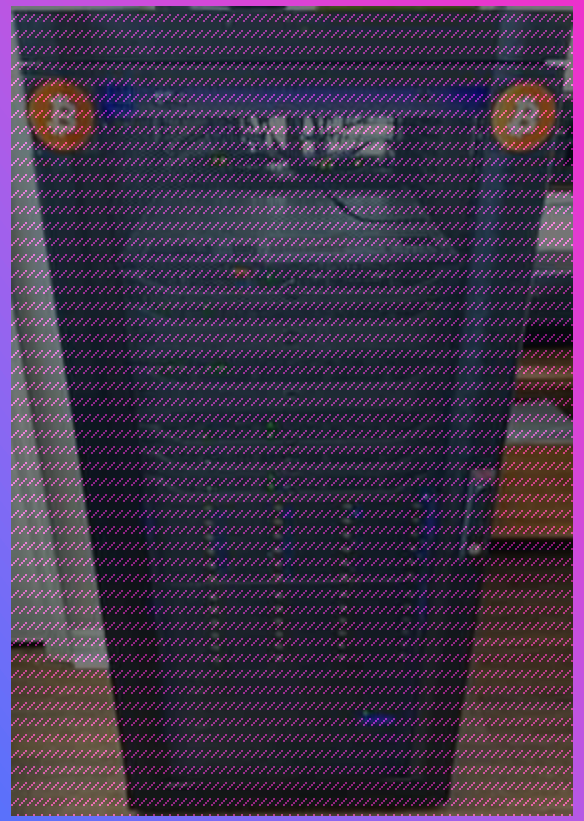
How to Build a Phishing Engagement

Coding TTP's

Ralph May

WHOAMI

- Current Penetration Tester @ BHIS
- Six years as professional Penetration Tester
- Army Veteran
- My Hobby is my Job
- From FL and part time alligator wrestler



DISCLAMER

- Some advanced topics
- Designed for Red Teams
- Automation is not great for everything
- I am going to GLOSS over some stuff
- This is MY way not THE way

OVERVIEW

- Phishing Overview
- Designing a Phish
- Coding a Phish
- Execution
- Closing

PHISHING IS HARD

- Infrastructure
- Scanners Scanning
- Domain
- Two-Factor
- Email Filtering
- User Awareness

INFRASTRUCTURE

DNS

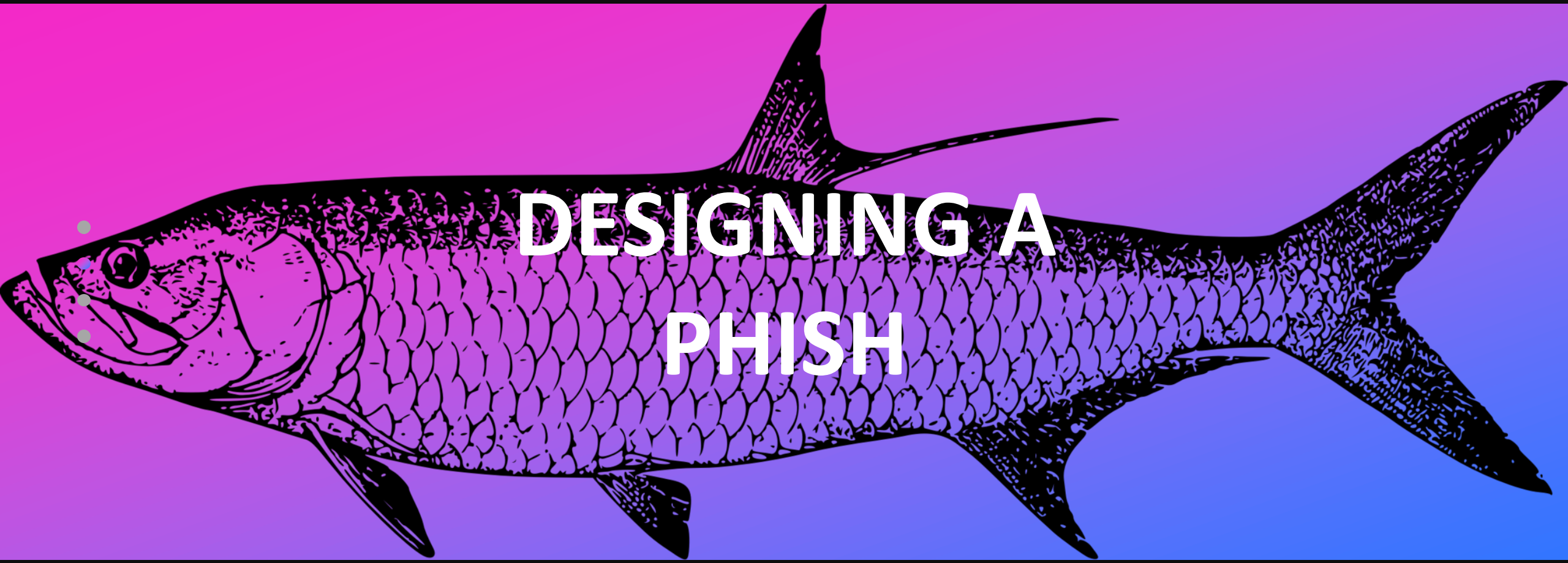
Virtual Machine

- Send Email
- Host Website
- Capture Credentials

Email Service

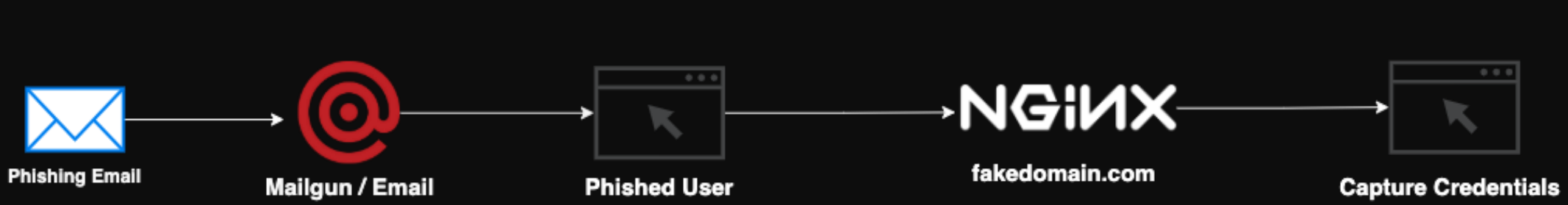
OPERATIONAL SECURITY

- Block URL scanning
- Hide IP address
- Secure the host
- Validate visitors
- Hide IOC’s

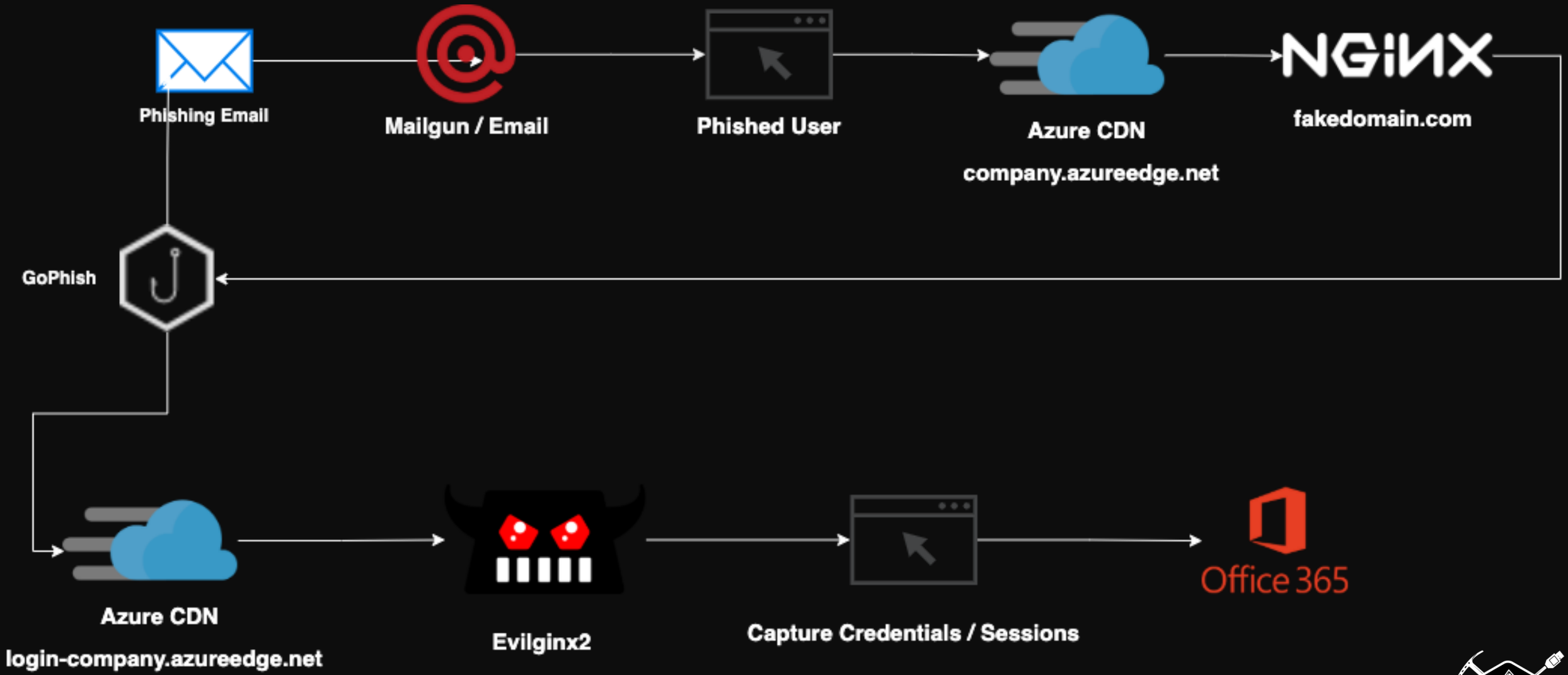


DESIGNING A PHISH

TYPICAL DESIGN

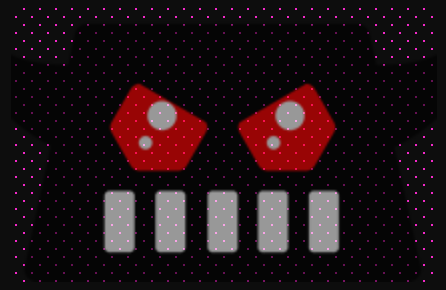


ADVANCED DESIGN



PHISHING EMAIL

- Take your time (You only have one shot)
- Test your email
  - Mailtrap.io
- Test your website
- Close the loop



EVILGINX2

- HTTP Proxy
- Capture Credentials
- Capture Session Data
- Defeat Two Factor
- Avoid cloning websites

```
root@debian-evilginx:~/tools/evilginx2# ./build/evilginx -p ./phishlets/

                                    Evilginx

                                    no nginx - pure evil

                                    by Kuba Gretzky (@mrgretzky)      version 2.0.0

[08:23:56] [inf] loaded phishlet 'google' from 'google.yaml'
[08:23:56] [inf] setting up certificates for phishlet 'google'...
[08:23:56] [^_^] successfully set up SSL/TLS certificates for domains: [accounts.it-is-almost-done.evilginx.com apis.it-is-almost-done.evilginx.com ssl.it-is-almost-done.evilginx.com content.it-is-almost-done.evilginx.com]
[08:23:59] [imp] [0] new visitor has arrived: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36 ([redacted])
[08:23:59] [inf] [0] landing URL: https://accounts.it-is-almost-done.evilginx.com/signin/v2/identifier
: sessions

+-----+----------+----------+----------+--------+-----------+---------------------+
| id  | phishlet | username | password | tokens | remote ip |        time         |
+-----+----------+----------+----------+--------+-----------+---------------------+
| 19  | google   |          |          | none   | [redacted] | 2018-05-28 08:23    |
+-----+----------+----------+----------+--------+-----------+---------------------+

[08:24:22] [^_^] [0] Username: [[redacted]@gmail.com]
[08:24:29] [^_^] [0] Password: [[redacted]]
[08:24:41] [^_^] [0] all authorization tokens intercepted!
[08:24:41] [imp] [0] redirecting to URL: https://redirect-to-this-url-after-logging-in.com
: sessions

+-----+----------+-------------------+----------+----------+-----------+---------------------+
| id  | phishlet |     username      | password |  tokens  | remote ip |        time         |
+-----+----------+-------------------+----------+----------+-----------+---------------------+
| 19  | google   | [redacted]@gmail.com | [redacted] | captured | [redacted] | 2018-05-28 08:24    |
+-----+----------+-------------------+----------+----------+-----------+---------------------+

:
```

https://github.com/kgretzky/evilginx2

EVILGINX IOC's

```
                        }
                }
        }
}

hg := []byte{0x94, 0xE1, 0x89, 0xBA, 0xA5, 0xA0, 0xAB, 0xA5, 0xA2, 0xB4}
// redirect to login page if triggered lure path
if pl != nil {
        _, err := p.cfg.GetLureByPath(pl_name, req_path)
        if err == nil {
                // redirect from lure path to login url
                rurl := pl.GetLoginUrl()
                resp := goproxy.NewResponse(req, "text/html", http.StatusFound, ""
                if resp != nil {
                        resp.Header.Add("Location", rurl)
```

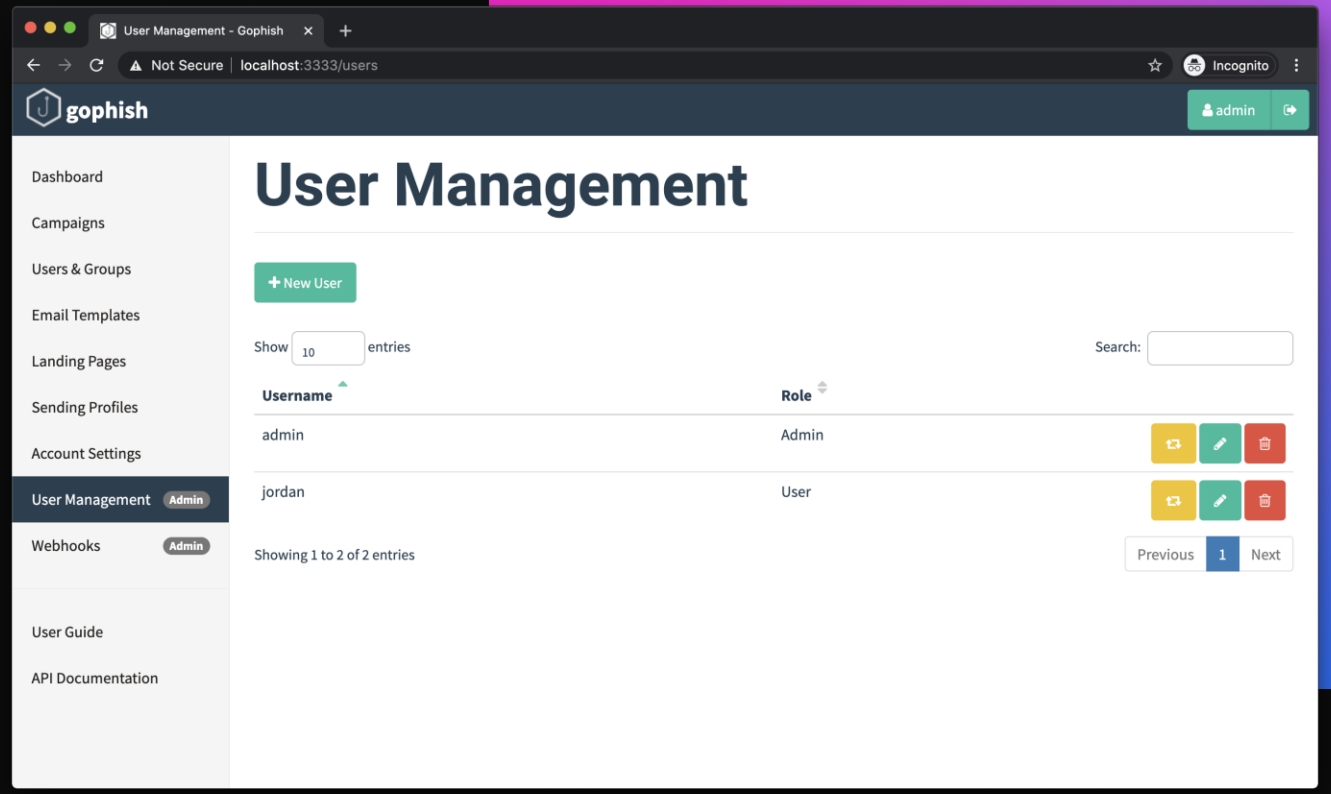
```
1  package main
2
3  import (
4          "fmt"
5  )
6
7  func main() {
8
9          hg := []byte{0x94, 0xE1, 0x89, 0xBA, 0xA5, 0xA0, 0xAB, 0xA5, 0xA2, 0xB4}
10
11         for n, b := range hg {
12                 hg[n] = b ^ 0xCC
13         }
14
15         fmt.Println(string(hg))
16 }
17
18
19
20
```

X-Evilginx

https://github.com/kgretzky/evilginx2/blob/master/core/http_proxy.go

GOPHISH

- Send our Phishing Emails
- Track interaction
- Landing-Page hosting



https://github.com/gophish/gophish

GOPHISH IOC's

```
Mime-Version: 1.0
Date: Wed, 31 Mar 2021 20:10:56 -0400
From: test@pwncompany.com
X-Mailer: gophish
Subject: Default Email from Gophish
To: "test testr" <test@test.com>
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

It works!

This is an email letting you know that your gophish
configuration was successful.
Here are the details:

Who you sent from: test@pwncompany.com
```

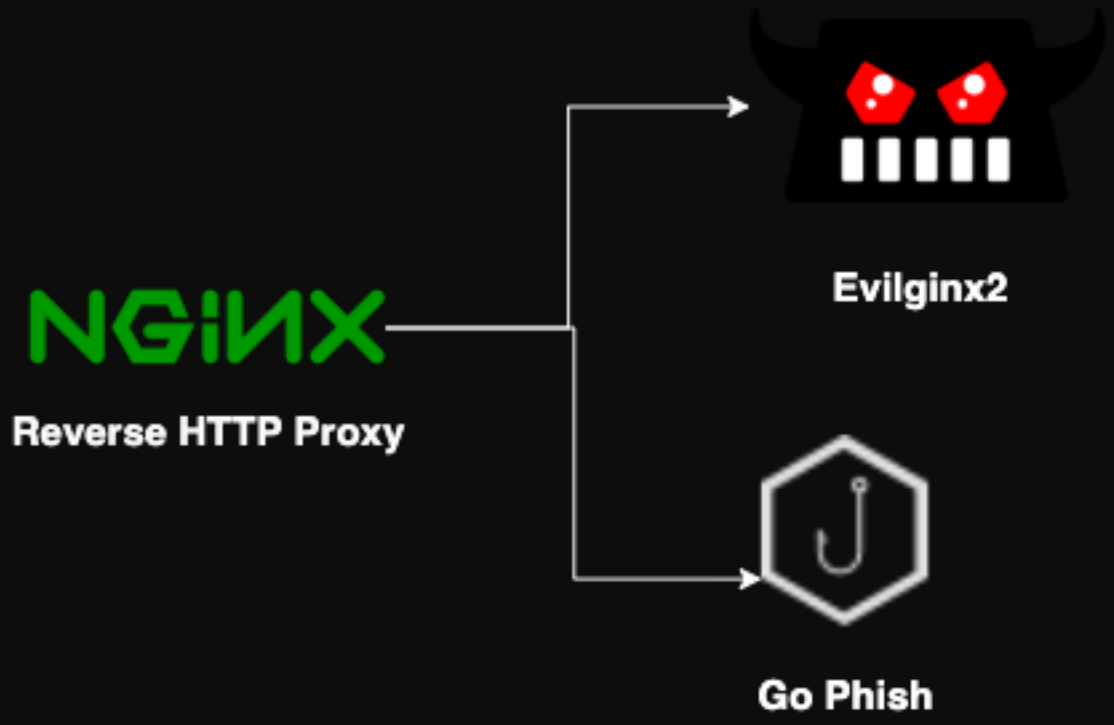
```
Mime-Version: 1.0
Date: Wed, 31 Mar 2021 22:04:52 -0400
From: test@pwncompany.com
X-Mailer:
Subject: Default Email from Gophish
To: test@test.com
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

It works!
```

```
Click Here https://mycompany-loading.azureedge.net?rid=ICAC5eN
```

```
123 var ErrSMTPNotFound = errors.New("Sending profile not found")
124
125 // ErrInvalidSendByDate indicates that the user specified a send by date that occurs before th
126 // launch date
127 var ErrInvalidSendByDate = errors.New("The launch date must be before the \"send emails by\" d
128
129 // RecipientParameter is the URL parameter that points to the result ID for a recipient.
130 const RecipientParameter = "rid"
131
```

NGINX

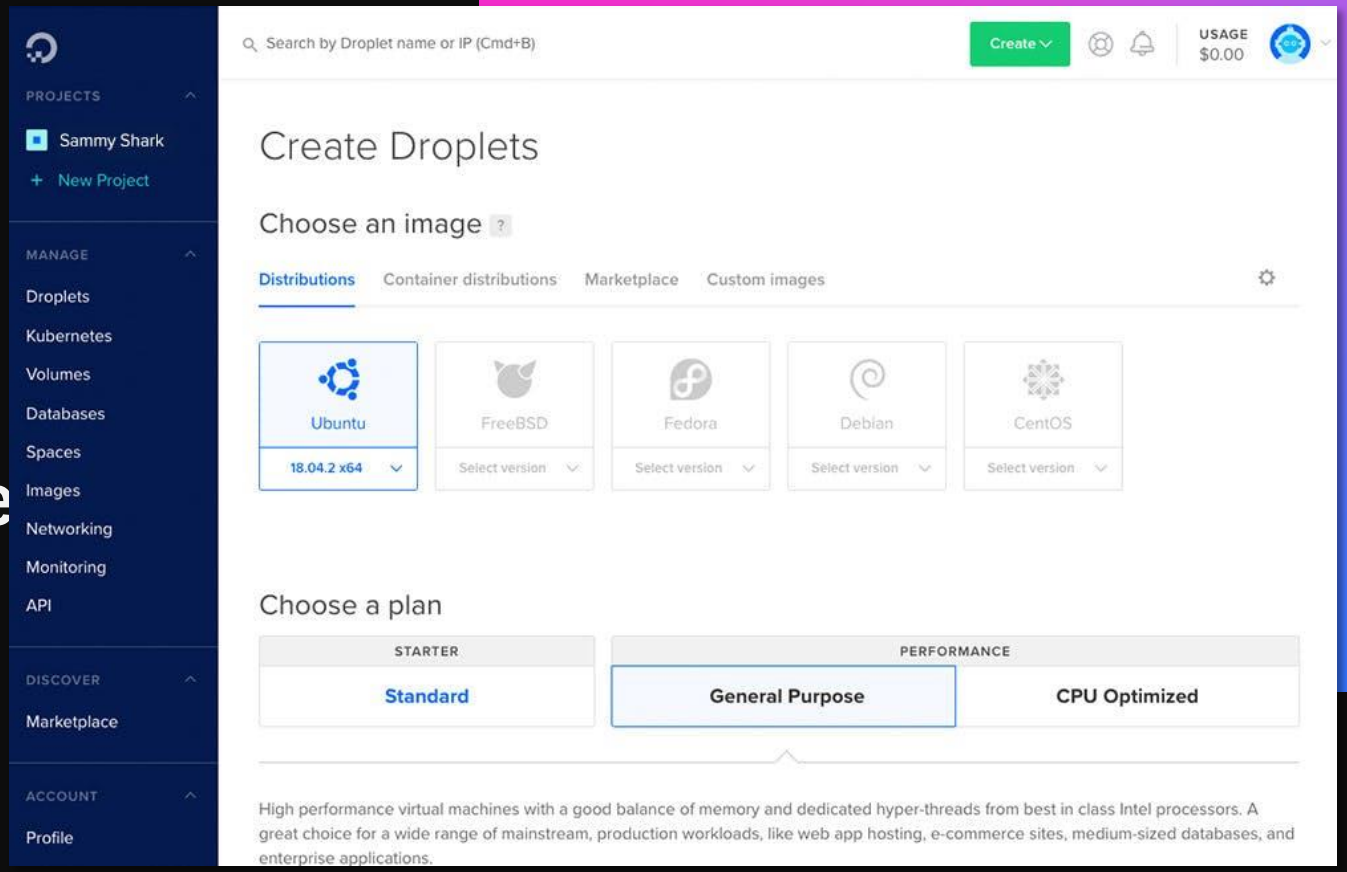
- HTTP Proxy / Router
- BLOCK ALL scanning
- Host multiple websites/containers
- SSL Certs / Let's Encrypt



https://www.nginx.com/

DIGITAL OCEAN

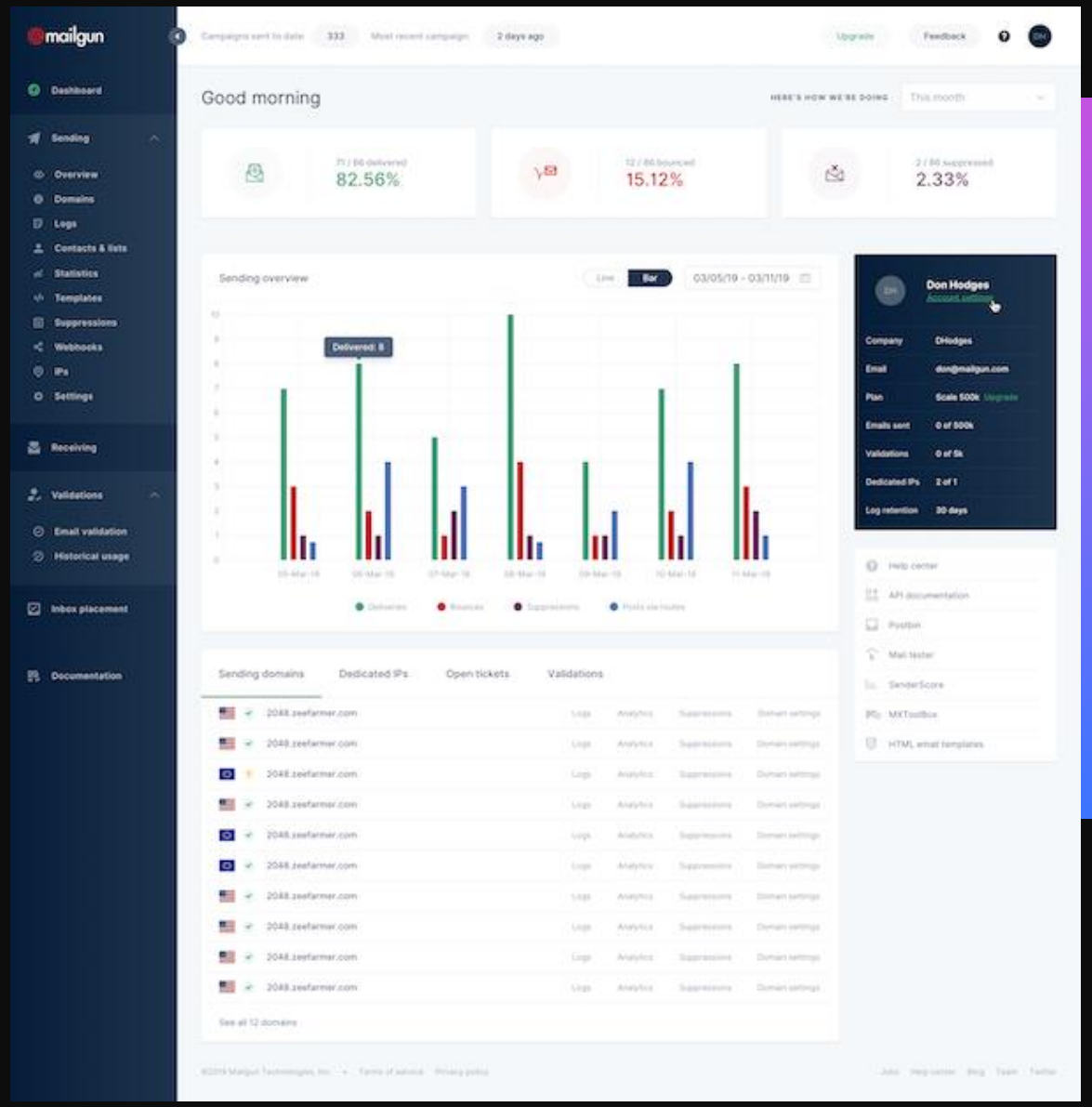
- API
- CHEAP virtual machine
- DNS



https://www.digitalocean.com/

MAILGUN

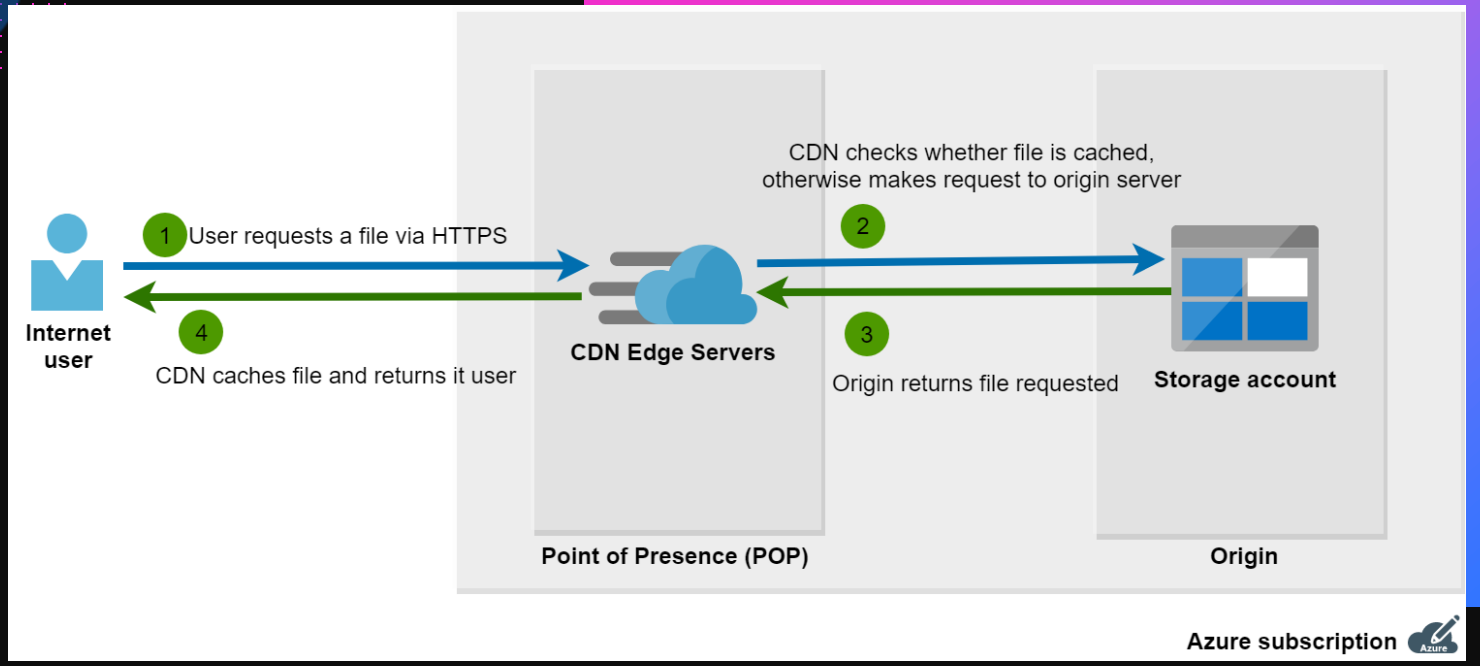
- DO NOT ROLL YOUR OWN EMAIL SERVER
- API
- Gain email reputation



https://www.mailgun.com

CDN- AZURE

- Hide our IP address
- IP Filtering
- Domain reputation
- azureedge.net

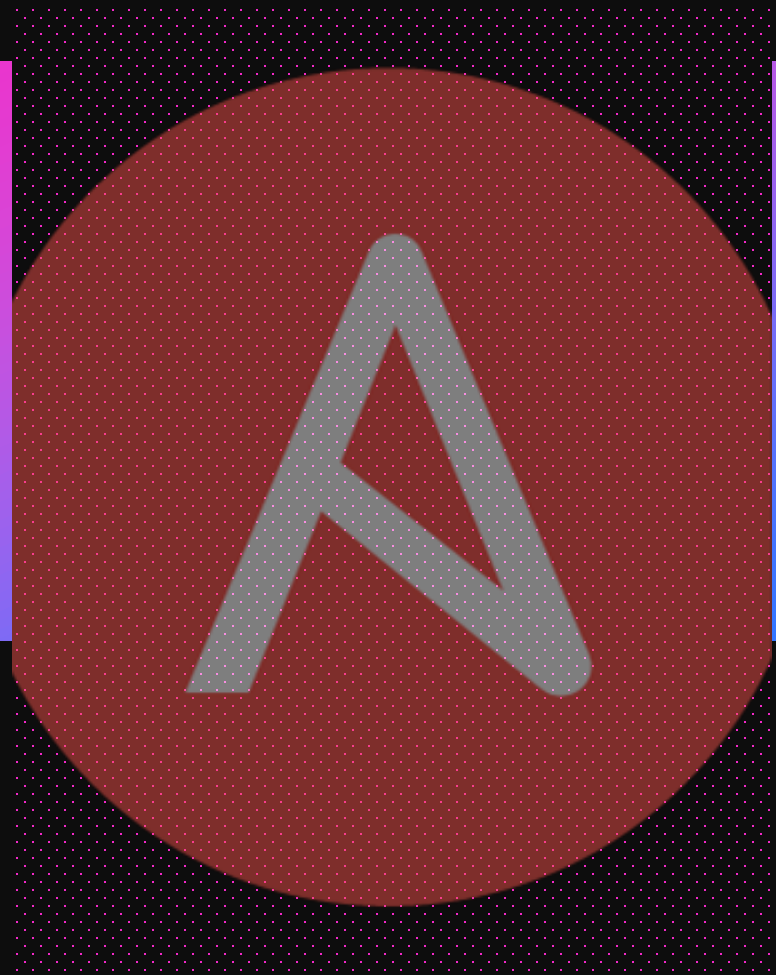


https://azure.microsoft.com/en-us/services/cdn/

CODING A PHISH

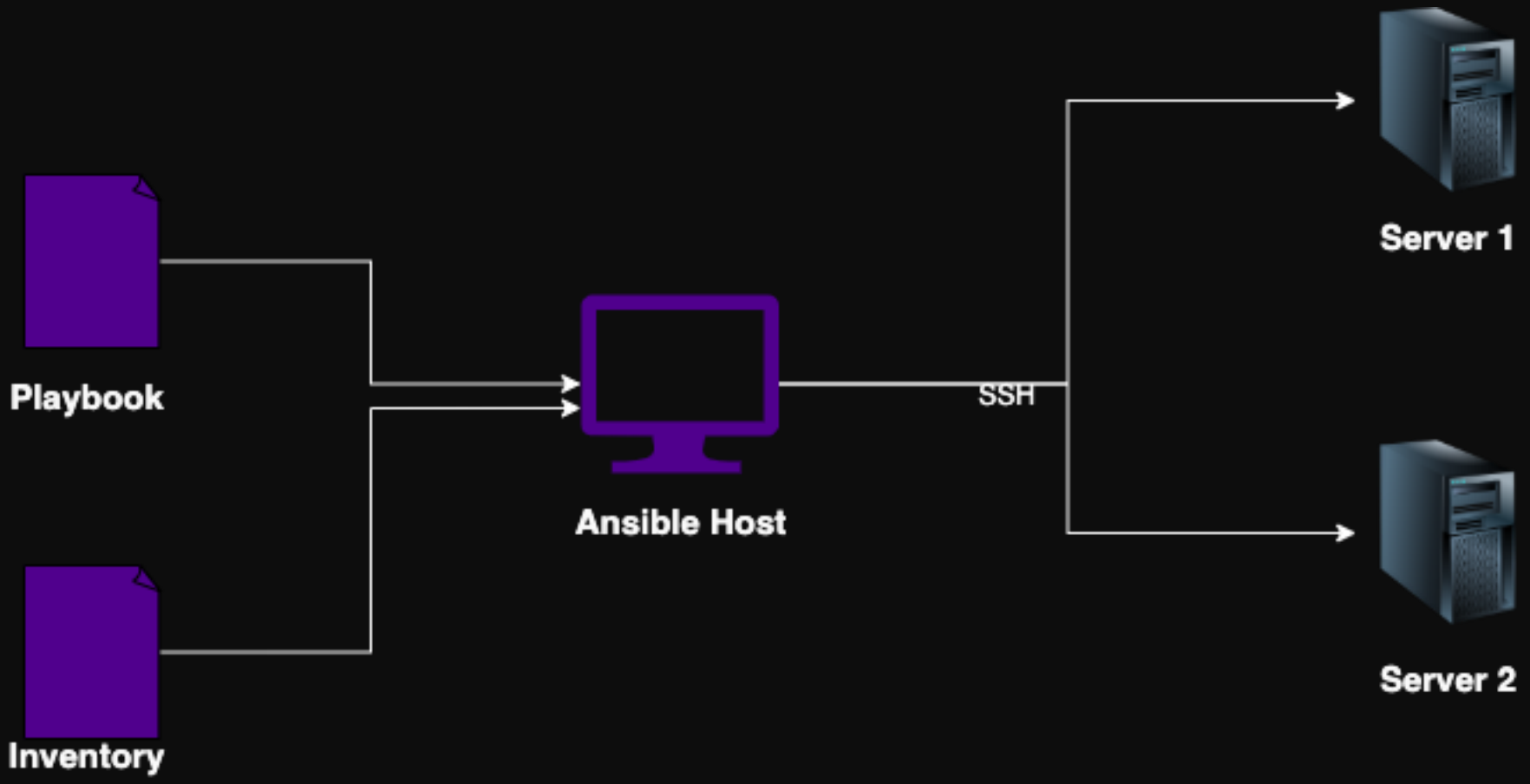
ANSIBLE

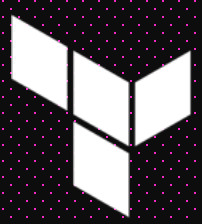
- Open-source configuration management
- Infrastructure as Code
- Python Based
- YAML files
- Red Hat Software
- No State file - procedural-style – Agent-less
- Great for OS configuration



https://www.ansible.com/

ANSIBLE





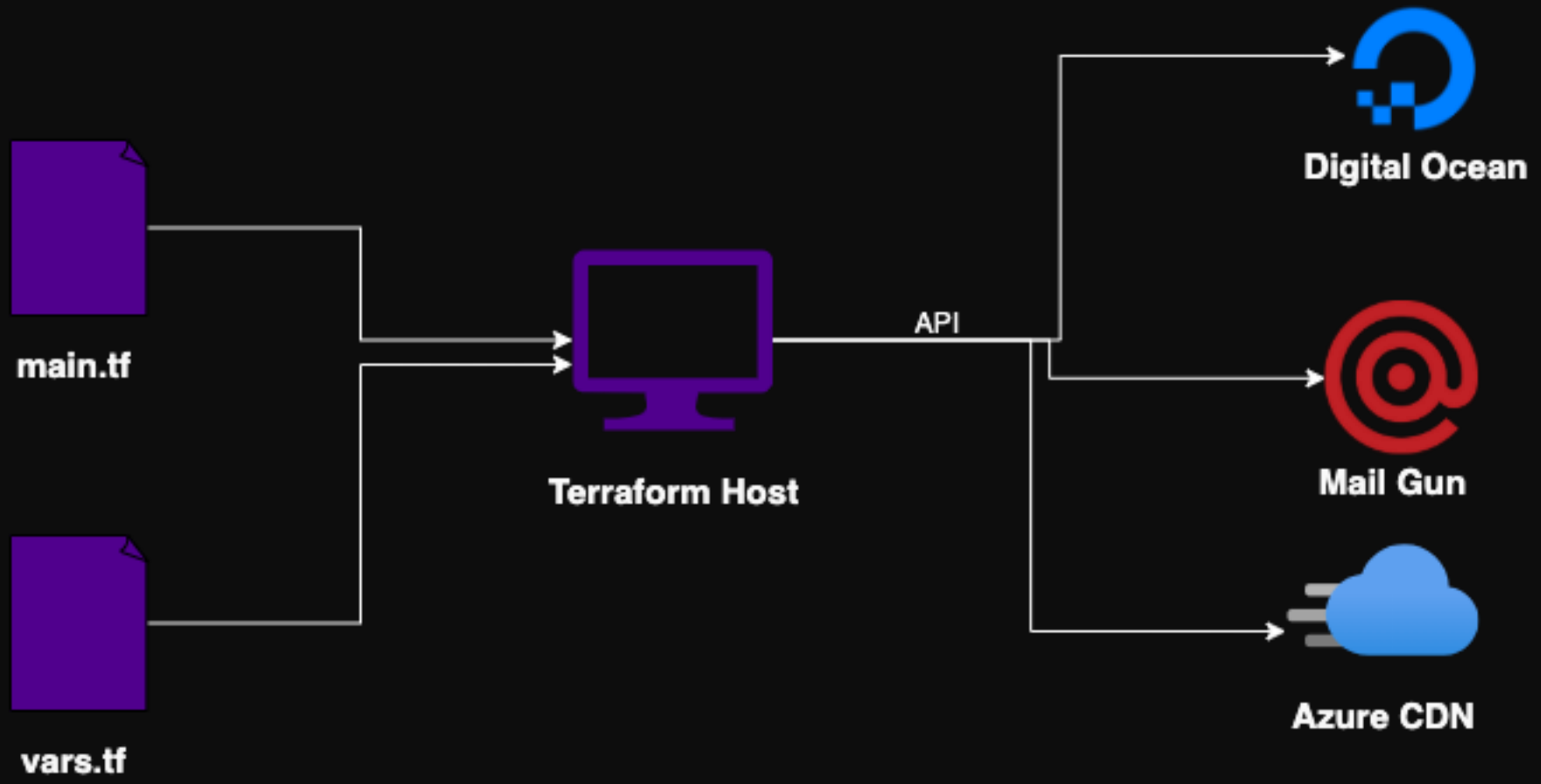
HashiCorp
Terraform

https://www.terraform.io/

TERRAFORM

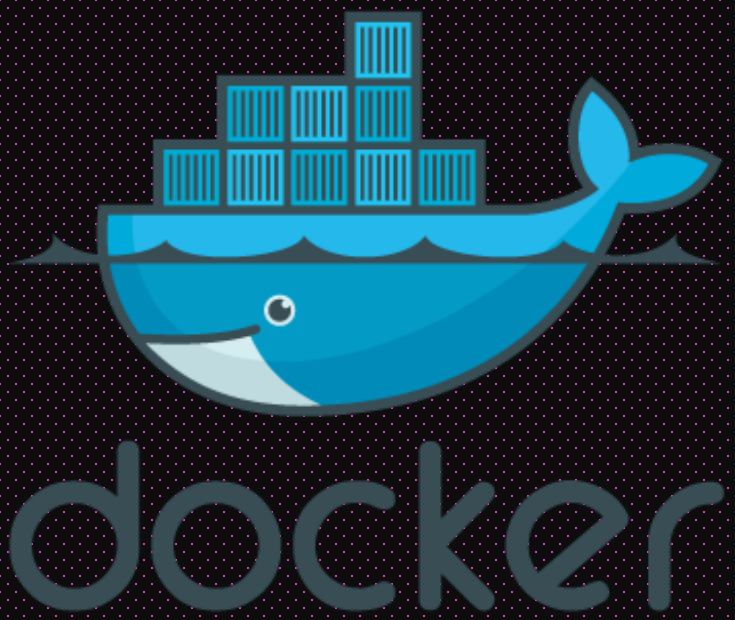
- Open-source orchestration tool
- Infrastructure as Code
- GO Lang Based
- Terraform's configuration language
- HashiCorp
- Statefile – Declarative
- Great for API's

TERRAFORM

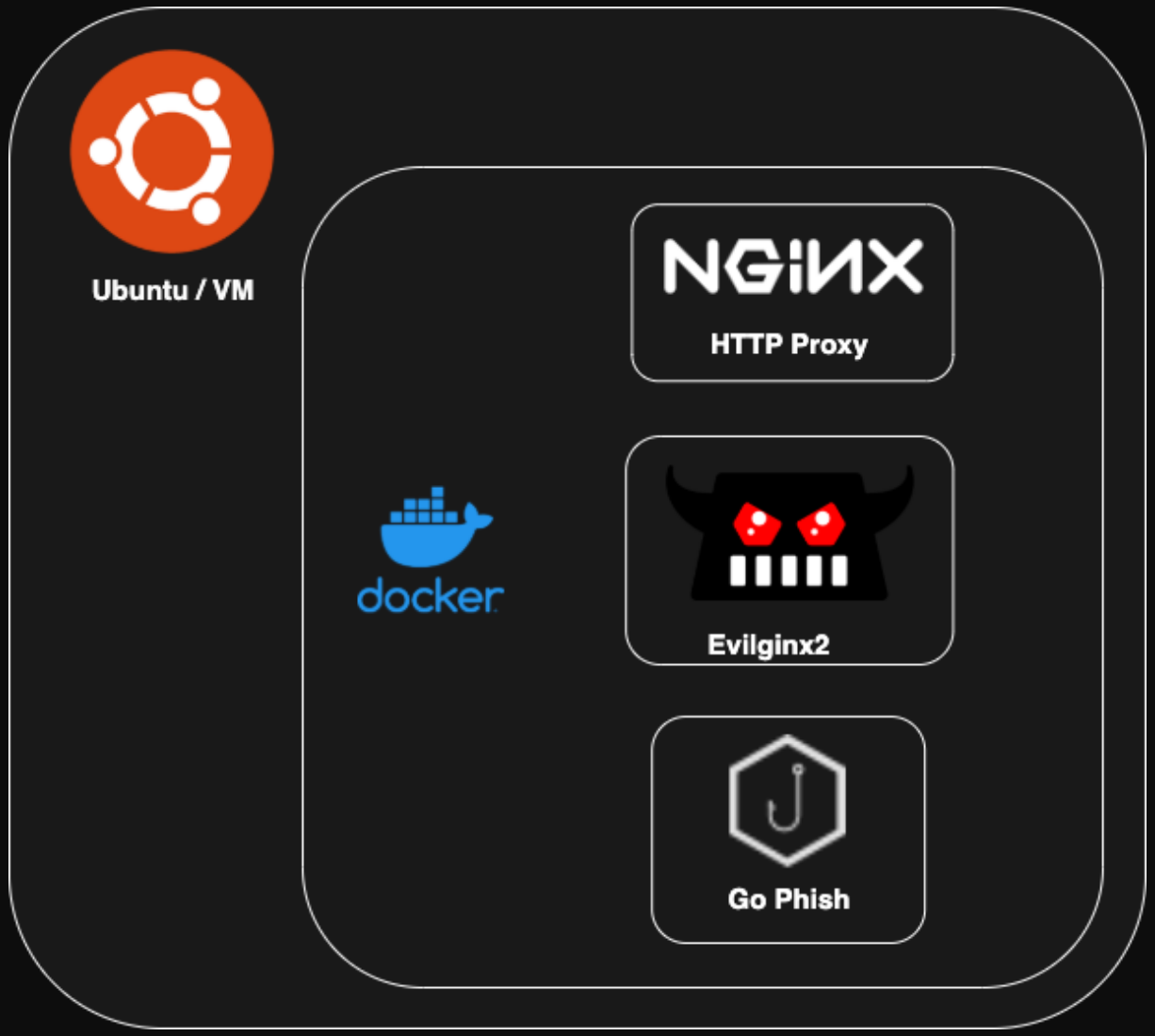


DOCKER

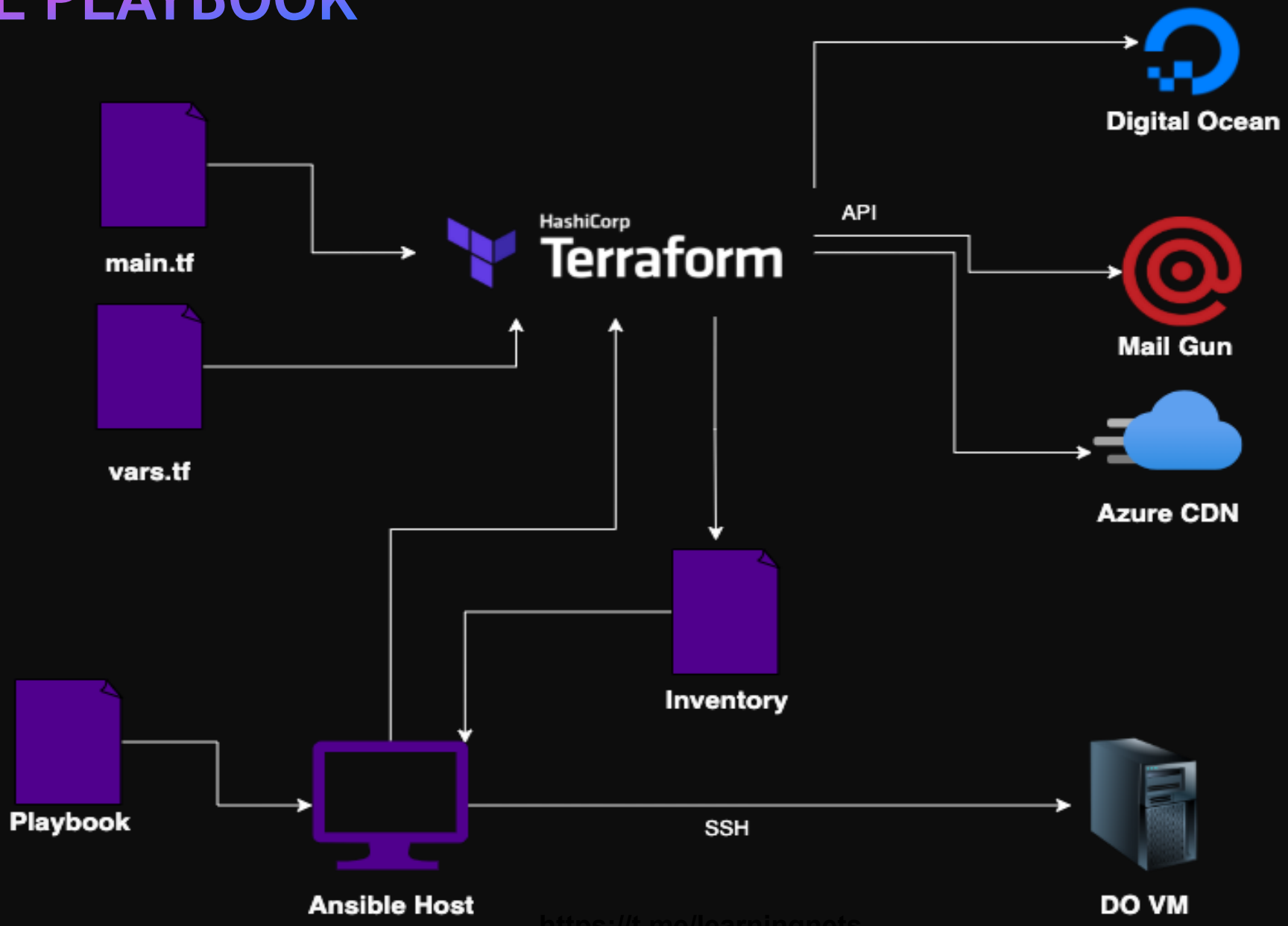
- Application containers
- Reusable
- Portable
- Simplify dependance
- Separate network stack



ANSIBLE PLAYBOOK DOCKER



ANSIBLE PLAYBOOK



ANSIBLE SECRETS

```
#Digital Ocean
digitalocean_token:  "{{lookup('community.general.lastpass', 'Business/Digitalocean', field='token' )}}"
digitalocean_token:  "564e5645645745665654 33sdgfsdfgd"
```

- Lastpass - cli
- 1password - cli
- Bitwarden - cli
- Ansible Vault - Encrypted File

EXECUTION

DEMO TIME

Let’s pray to the demo GOD’s.

- Clone repo
- Setup API keys
- Configure variables
- Run playbook

https://github.com/ralphte/build_a_phish

What's Next

- What about other cloud providers?
- Can I use this playbook for more then just phishing?
- Blog to follow presentation soon!
- Webcast - Getting started with Ansible & Terraform

Thanks

Questions?

ralphte01

Ralph May