the factory cloud runners

Running trusted payloads with Nomad and Waypoint

Bram Vogelaar
@attachmentgenie

$ whoami

- Used to be a Molecular Biologist
- Then became a Dev
- Now an Ops
- Currently Cloud Engineer @ The Factory

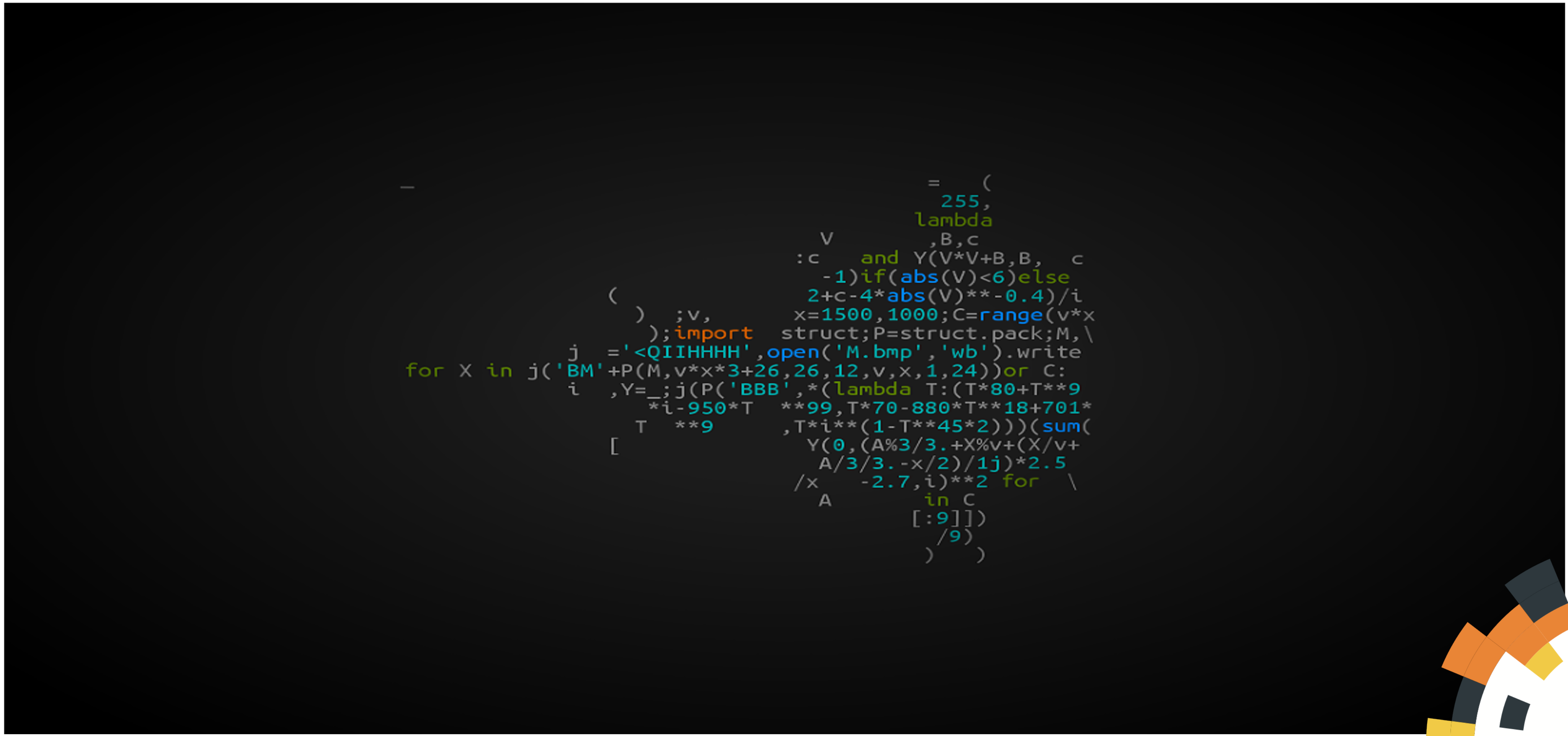
Monday Morning: Standup time!

"The intern "solved" the problem with $product by "fixing" the database connection"

Wait! What?

#nocontext

$problems++?



```
_                                       =   (
                                         255,
                                       lambda
                                V       ,B,c
                              :c   and Y(V*V+B,B,   c
                                -1)if(abs(V)<6)else
               (               2+c-4*abs(V)**-0.4)/i
                 )  ;v,       x=1500,1000;C=range(v*x
                  );import  struct;P=struct.pack;M,\
            j  ='<QIIHHHH',open('M.bmp','wb').write
for X in j('BM'+P(M,v*x*3+26,26,12,v,x,1,24))or C:
            i  ,Y=_;j(P('BBB',*(lambda T:(T*80+T**9
                  *i-950*T  **99,T*70-880*T**18+701*
                 T  **9     ,T*i**(1-T**45*2)))(sum(
               [               Y(0,(A%3/3.+X%v+(X/v+
                                A/3/3.-x/2)/1j)*2.5
                              /x   -2.7,i)**2 for  \
                                A       in C
                                       [:9]])
                                         /9)
                                        )   )
```

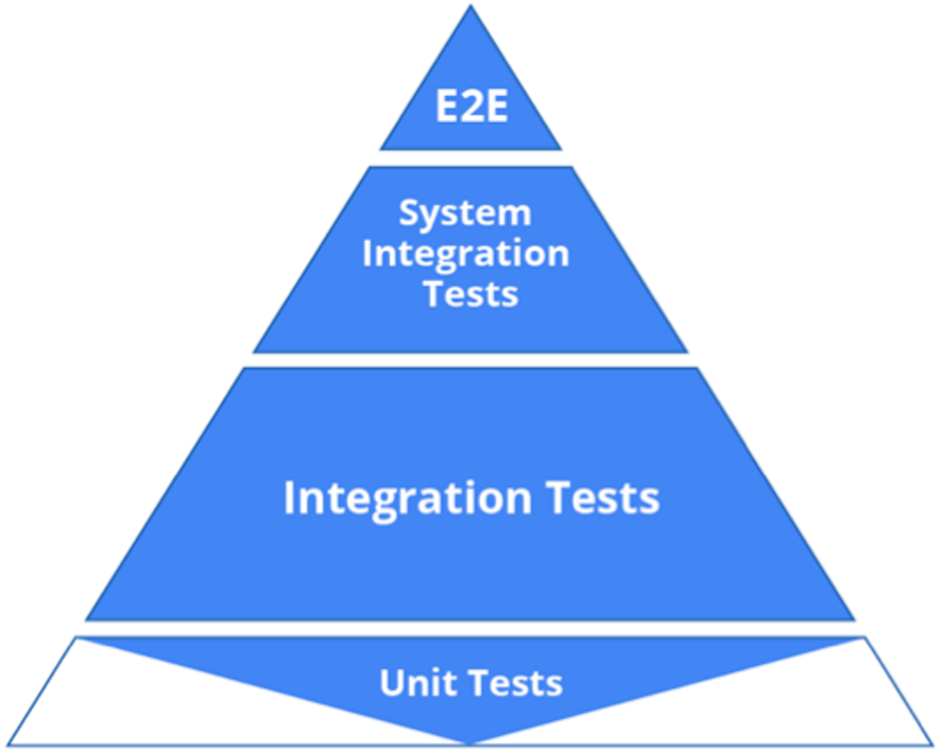
Access Security

CI/CD is in place to provide trust

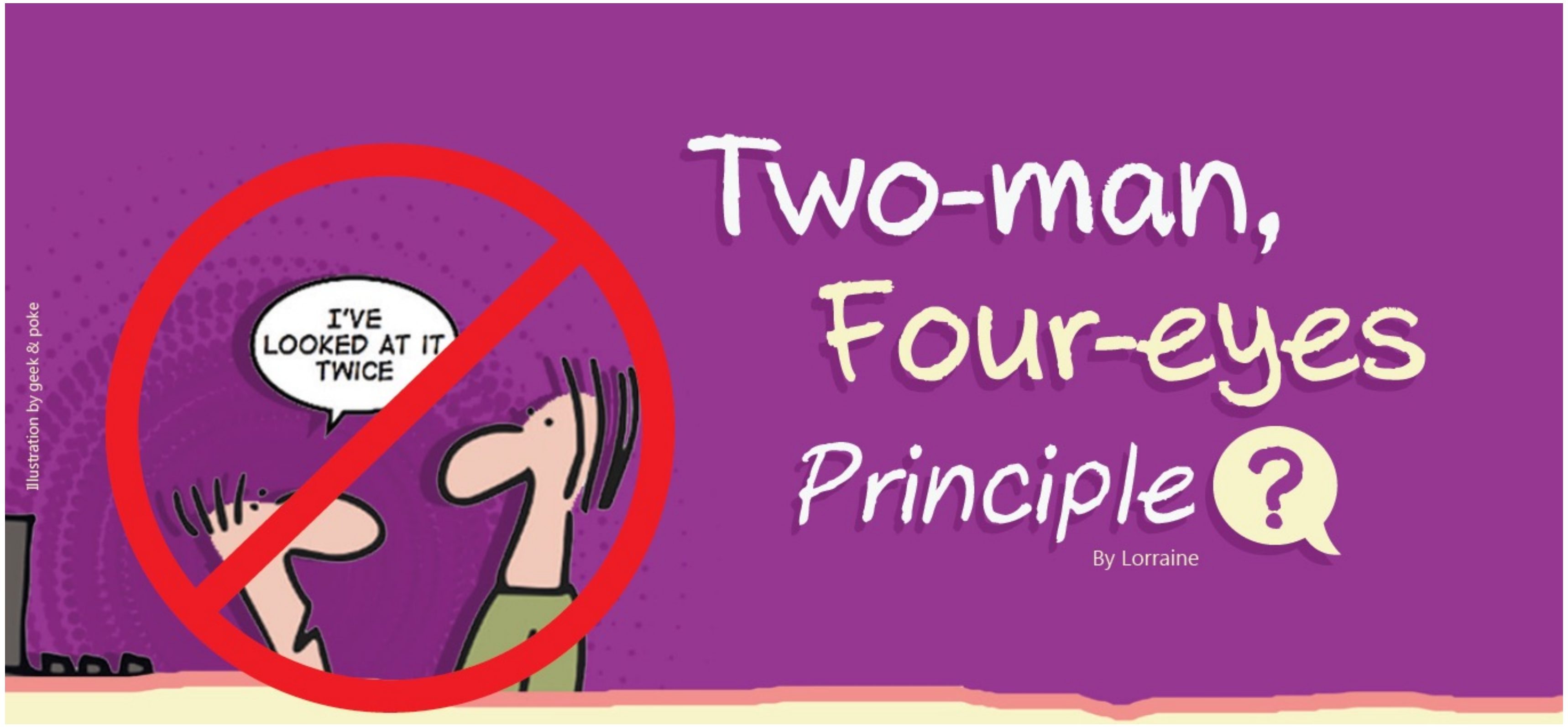
The discipline of experimenting on a distributed system in order to build *confidence* in the system's capability to withstand turbulent conditions *in production*

Chaos Engineering Definition

Testing Pyramid



Shared Responsibility



Waypoint

- Modern workflow to release across platforms.
- Pipelines written in (H)ashiCorp (C)onfiguration (L)anguage
- Extendable by with (self-written) plugins

https://www.waypointproject.io/

Common nomenclature for pipeline steps

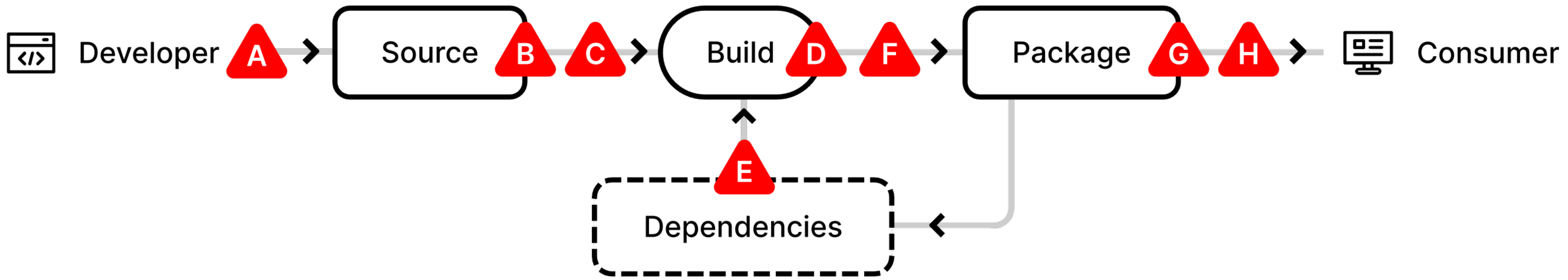
```
project = ”lorem-ipsum”
app ”lorem-ipsum” {
  build {}
  deploy {}
  release {}
}
```

Vault

- Open Source tool to do secrets management
- Secure, store and tightly control access to tokens, passwords, certificates, encryption keys for protecting secrets and other sensitive data using a UI, CLI, or HTTP API.
- Certificate management
- Supports Password as a Service for tools like SSH and MySQL

https://www.vaultproject.io/

Supply Chain Attack Vectors



A Bypassed code review
B Compromised source control system
C Modified code after source control
D Compromised build platform
E Using a bad dependency
F Bypassed CI/CD
G Compromised package repo
H Using a bad package

https://slsa.dev/

Don’t trust the internet

Standing on the shoulders of giants, has pro’s and con’s

- Npm
- Maven
- Composer
- Rubygems

Be Self Sufficient!

- Build from local ensure continuity
  - OS repositories
  - Image repositories
  - Code packages
- Create a Software Bill of Materials (SBOM)
  - What is included in the first place?
  - Unwanted Licenses
  - Vulnerable Packages / Images

Vulnerability Scanning

clair

snyk

Garbage in, garbage out?

```
app ”lorem-ipsum” {
 build {
       hook {
            when = "before"
            command = ["./snyk_scan.sh", "build starting"]
        }
        hook {
            when = "after"
            command = ["./artifact_fingerprint.sh", "build finished"]
        }
    }
}
```

Don’t promote code, but promote artifacts

docker

Nomad

- Open Source tool for dynamic workload scheduling
- Batch, containerized, and non-containerized applications.
- Has native Consul and Vault integrations.
- Has token based access setup.
- Jobs written in (H)ashiCorp (C)onfiguration (L)anguage

https://www.nomadproject.io/

Nomad Job Structure

```
job "lorem-ipsum" {
  group ”frontend" {
    network {
      port "http" { static = ”8080” }
    }
    task "server" {
      driver = "docker"
      config {
        image = ”cicero/lorem-ipsum"
        ports = ["http"]
      }
  }
}
```

Trusted Content

Docker Content Trust

```
$ docker trust key generate bram
$ docker trust key load key.pem --name bram
$ docker trust signer add --key cert.pem jeff registry.example.com/cicero/lorem-ipsum
$ docker trust sign registry.example.com/cicero/lorem-ipsum:1
$ export DOCKER_CONTENT_TRUST=1
$ docker push registry.example.com/cicero/lorem-ipsum:1
```

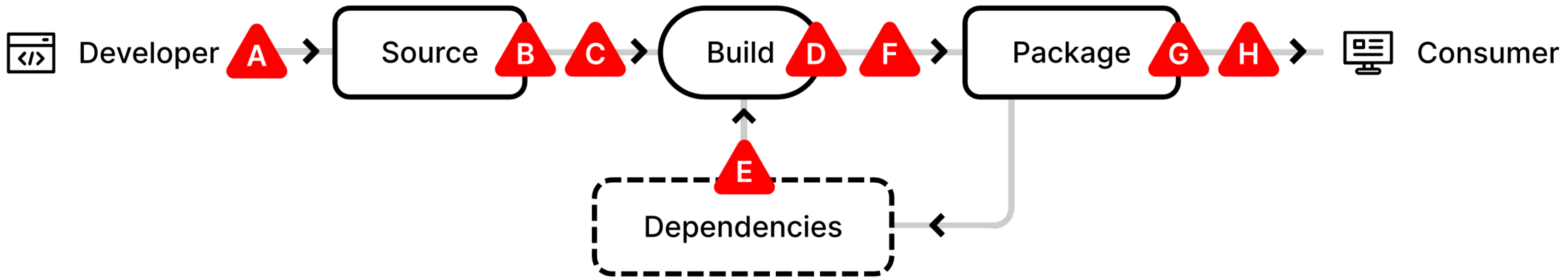
Docker Content Trust

```
$ docker trust inspect --pretty registry.example.com/admin/demo:1

$ docker trust revoke registry.example.com/admin/demo:1

$ export DOCKER_CONTENT_TRUST=1
$ docker pull registry.example.com/user/image:1
```

Q.E.D?



A Bypassed code review
B Compromised source control system
C Modified code after source control
D Compromised build platform
E Using a bad dependency
F Bypassed CI/CD
G Compromised package repo
H Using a bad package

https://slsa.dev/

Contact

bram@attachmentgenie.com

@attachmentgenie

https://www.slideshare.net/attachmentgenie