Deploying Dockerized Rails App to AWS EC2/ECS #1

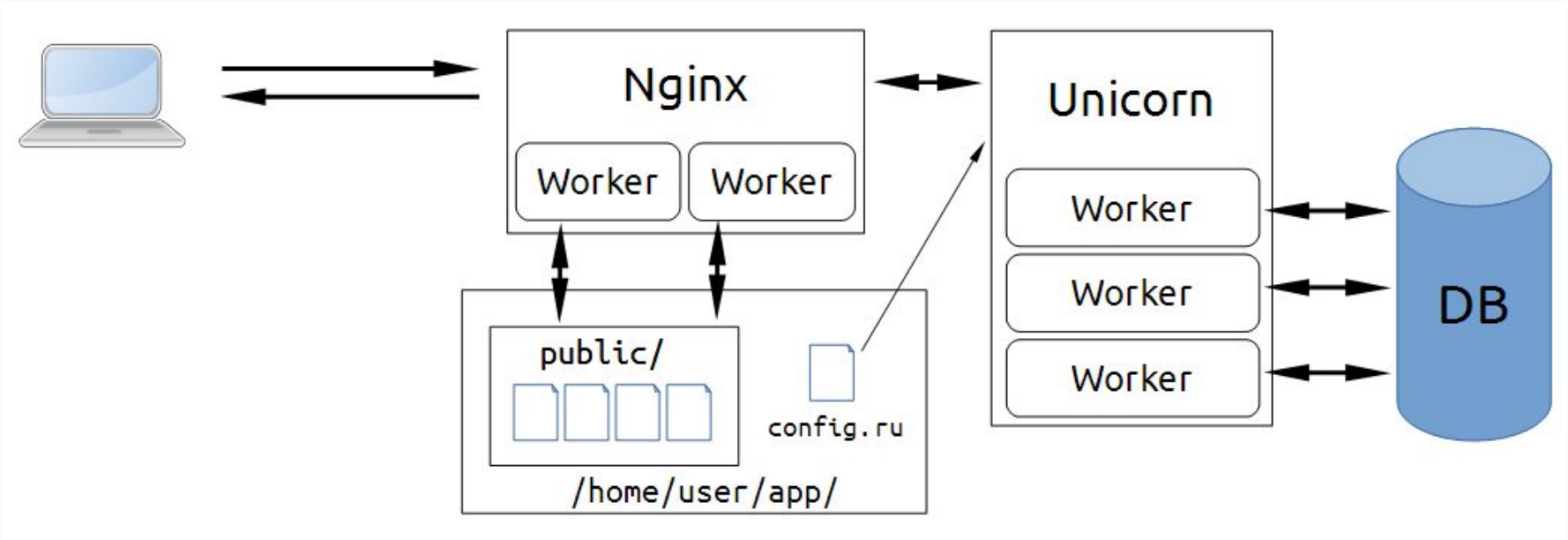
Ankit, Trung

Growth Session #18 - September 13-14 2018

Steps We Will Follow

- Infrastructure Provisioning (Launching the instance, RDS etc.) using Terraform.
- Secret Manager engine: Vault.
- Setting up underlying components:
  - Web Server (Nginx, Apache).
  - Application Server (Puma, Passenger).
  - Docker configuration (Dockerfile).
  - Installing docker on EC2 instance.
  - Deployment automation tool(Mina, Capistrano) or Use CI/CD Tool

Request Process



Nginx Vs Apache

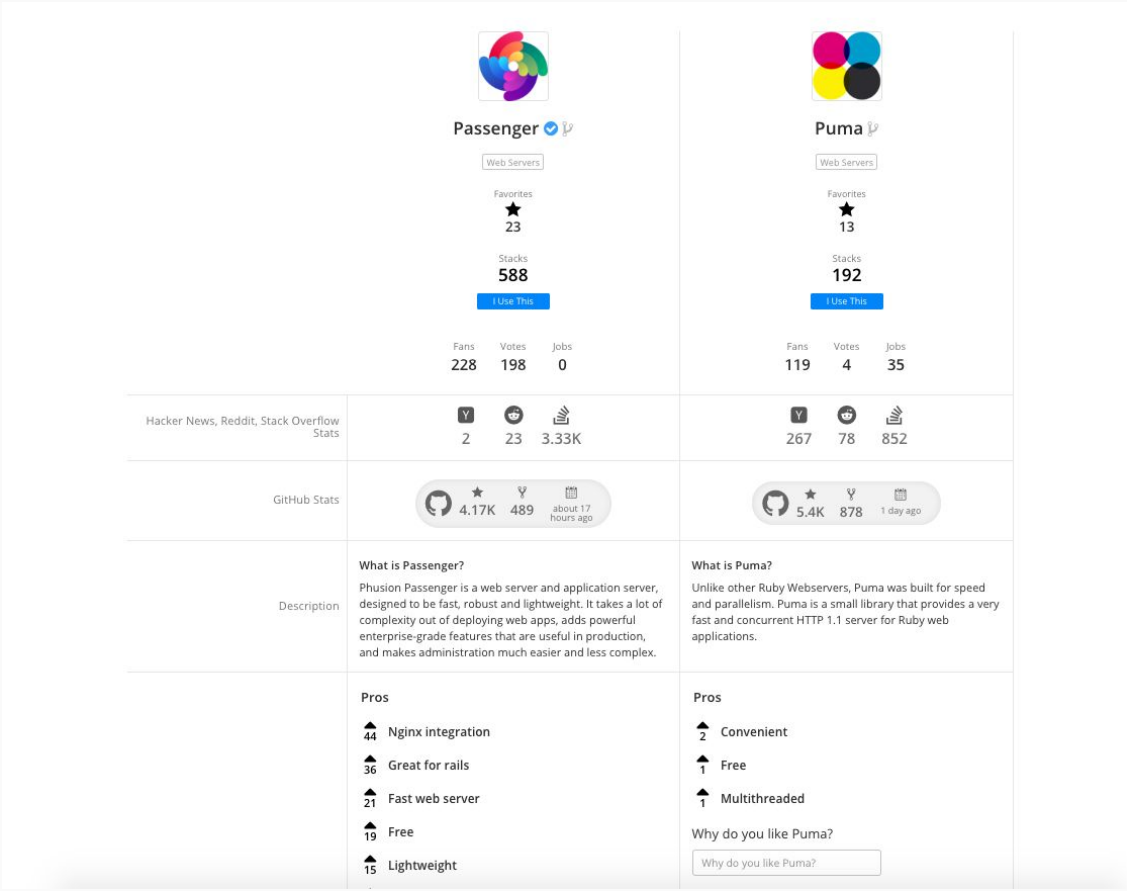
Nginx should be preferred because:

1. It’s lightweight.
2. It is Designed for High Concurrency.
3. As opposed to Apache’s threaded- or process-oriented architecture (process-per-connection or thread-per-connection model), Nginx uses a scalable, event-driven (asynchronous) architecture. It employs a reliable process model that is tailored to the available hardware resources.
4. Each nginx worker process can handle thousands of connections.

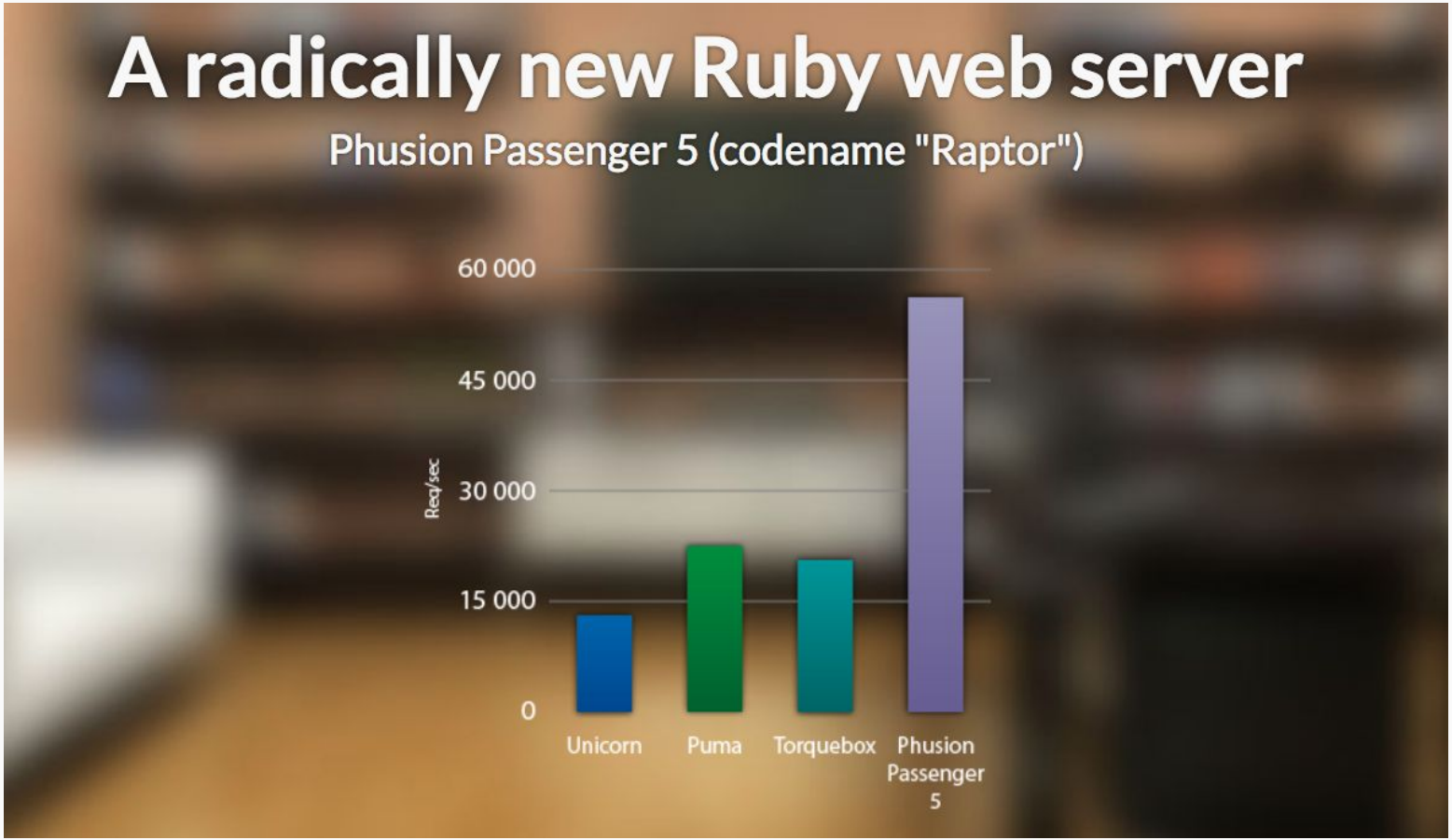
Nginx Processes

```
ubuntu@ip-172-31-2-126:/var/log/nginx$ ps aux | grep nginx
root     17196  0.0  0.2 179076  5696 ?        Ss   09:40   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data 17197  0.0  0.4 179076  8384 ?        S    09:40   0:00 nginx: worker process
ubuntu   17258  0.0  0.0  12916   968 pts/2    S+   09:50   0:00 grep nginx
```

Passenger vs Puma



Passenger vs Puma



Passenger vs Puma: Multithreading

- Normally one process can process one request.
- Multithreading can allow us to have multiple requests in a single process.
- Passenger does not have multithreading in free version while puma does.

But overall Phusion Passenger 5 claims to be upto 4 times faster than other app servers.

Techniques We Can Use

Dockerize Rails app and Nginx – separate containers

Dockerize Rails app and Nginx – single container

Dockerize Rails app into container and run Nginx on host 

Push, pull and run Docker image on another host

Achievement and Progress

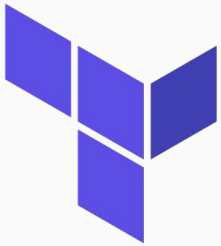
1. Set up infrastructure using Terraform.
2. Installed Docker on ec2.
3. Figured out a way to run Vcliq using docker locally.
4. Pushed the Vcliq to EC2 and tried to build the docker image, all processes completed

Nginx Config

/etc/nginx/sites-enabled/nginx.conf:

```
upstream puma_sample_rails_docker_app {
  server 127.0.0.1:3000;
}


server {

  listen 80;

  client_max_body_size 4G;
  keepalive_timeout 10;

  error_page 500 502 504 /500.html;
  error_page 503 @503;

  server_name localhost puma_sample_rails_docker_app;
  root /var/www/sample_rails_docker_app/public;
  try_files $uri/index.html $uri @puma_sample_rails_docker_app;

  location @puma_sample_rails_docker_app {
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_redirect off;

    proxy_pass http://puma_sample_rails_docker_app;
    # limit_req zone=one;
    access_log /var/log/nginx.access.log;
    error_log /var/log/nginx.error.log;
  }

  location ^~ /assets/ {
    gzip_static on;
    expires max;
    add_header Cache-Control public;
```

About Terraform



HashiCorp
Terraform

INFRASTRUCTURE AS CODE

Define infrastructure as code to increase operator productivity and transparency.

COLLABORATE & SHARE

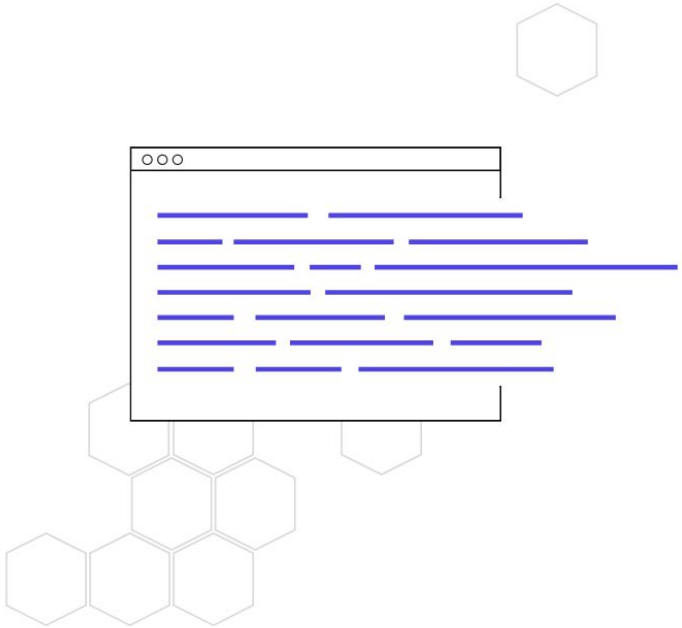
Terraform configuration can be stored in version control, shared, and collaborated on by teams of operators.

EVOLVE YOUR INFRASTRUCTURE

Track the complete history of infrastructure versions.

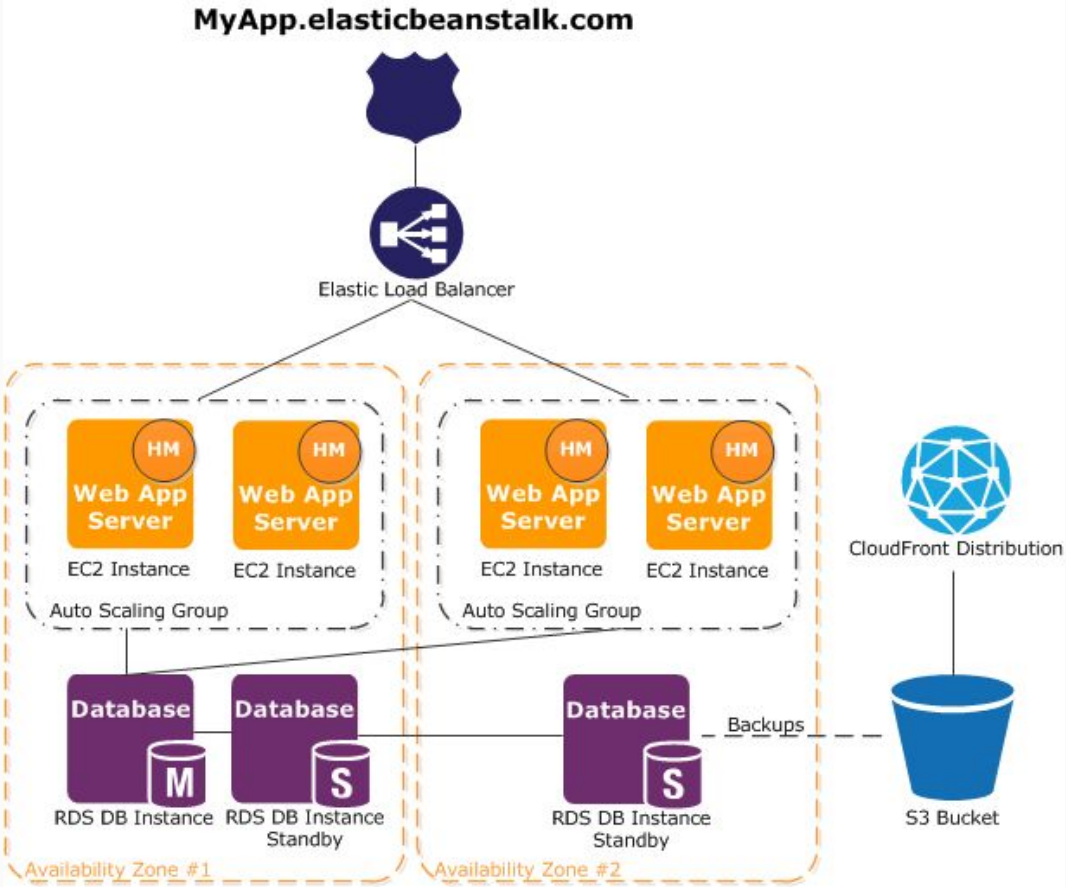
AUTOMATION FRIENDLY

If it can be codified, it can be automated.



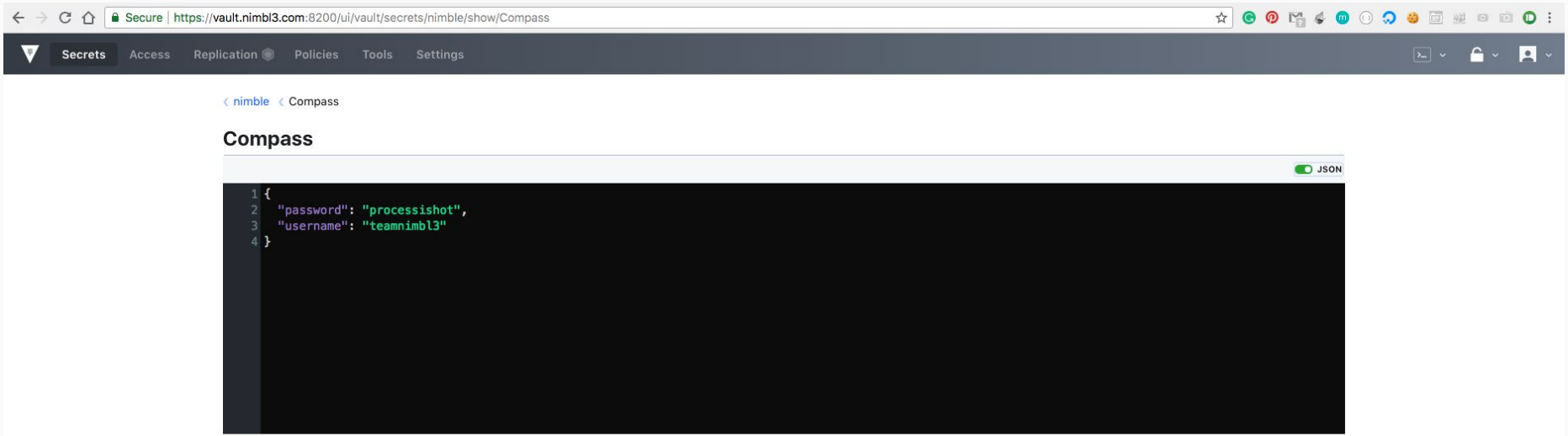
What can it do?

HashiCorp
Terraform

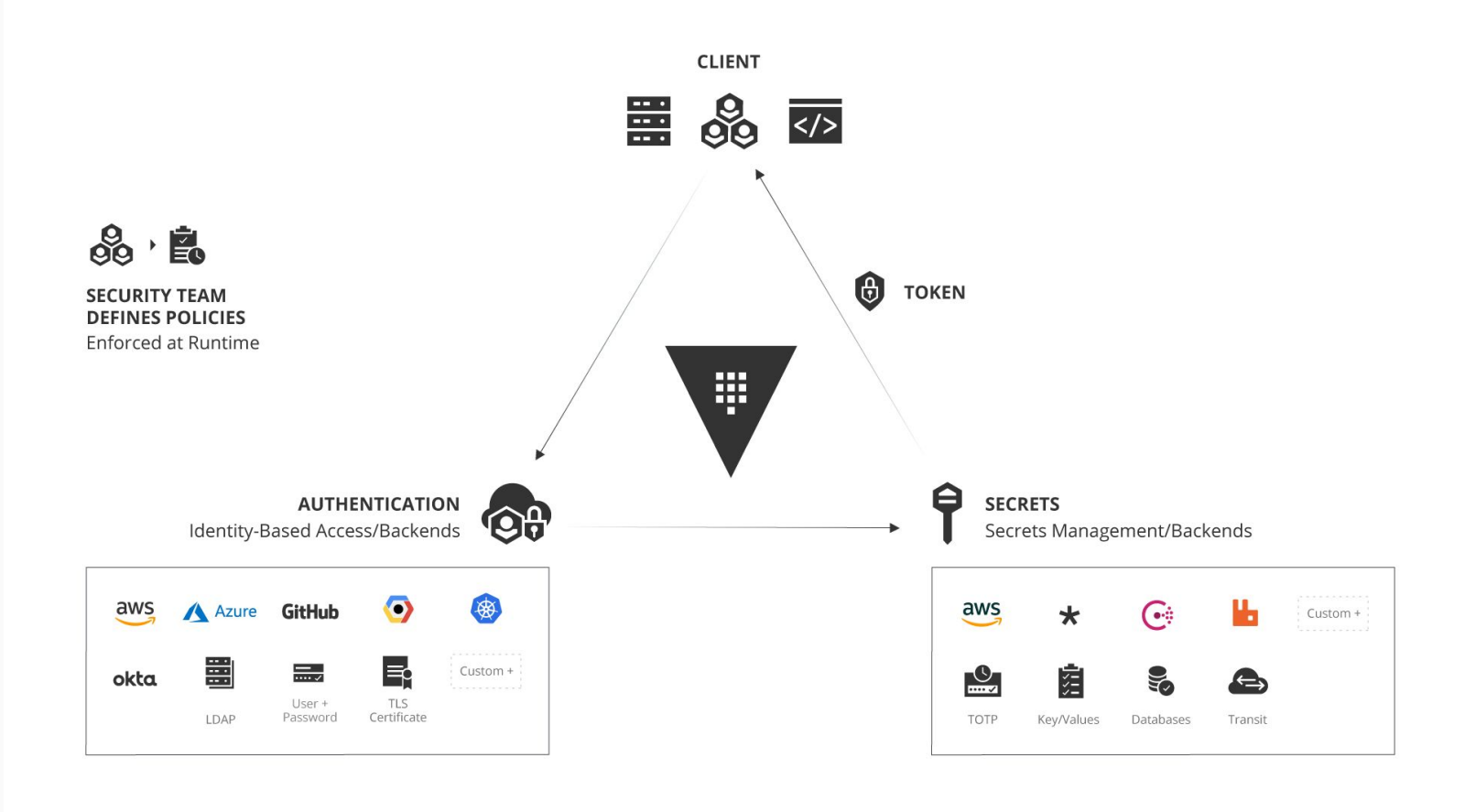


What is Vault?

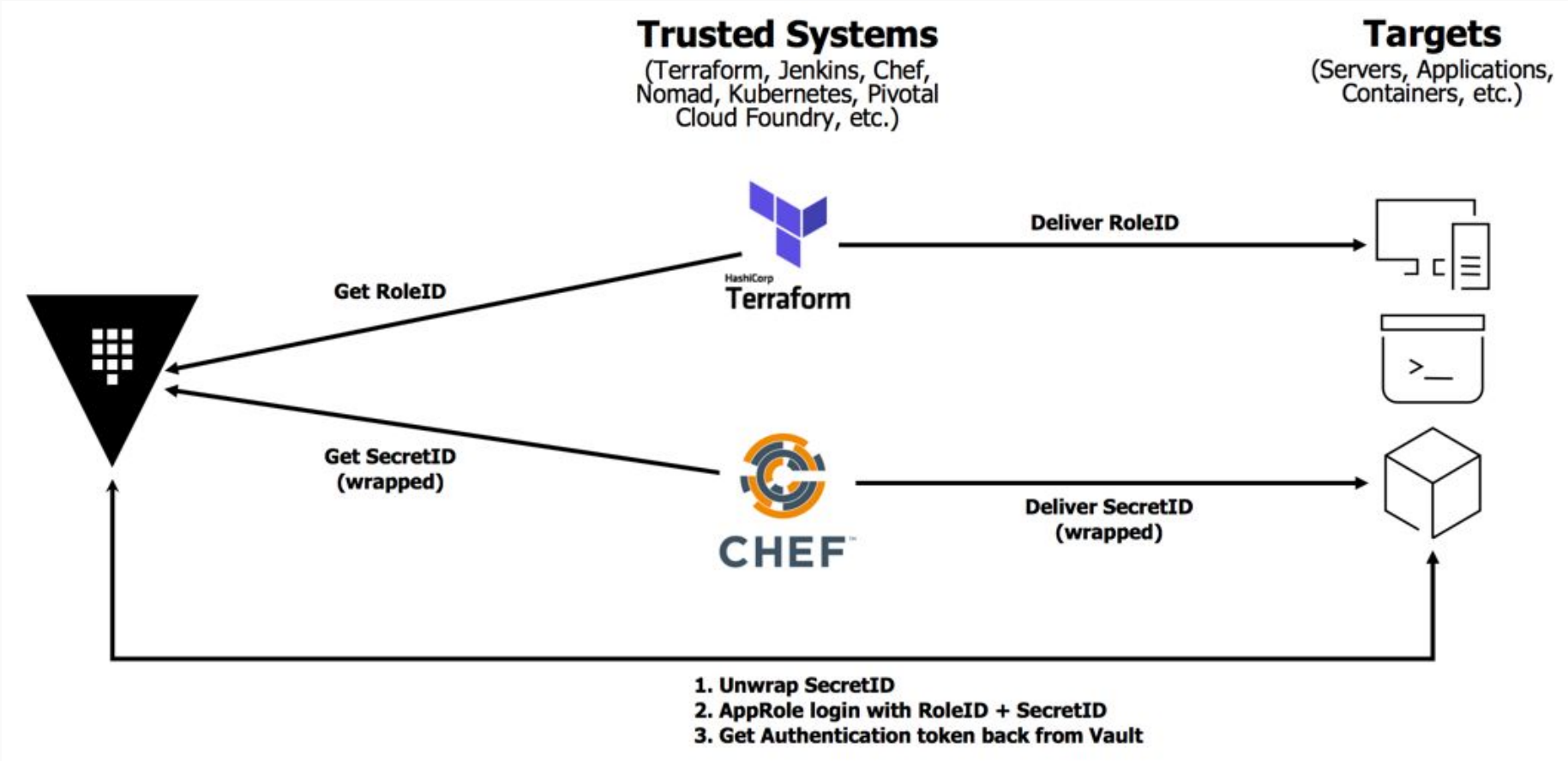
HashiCorp Vault



Vault + AWS?


CLIENT
TOKEN
SECURITY TEAM
DEFINES POLICIES
Enforced at Runtime
AUTHENTICATION
Identity-Based Access/Backends
SECRETS
Secrets Management/Backends
aws
Azure
GitHub
okta
LDAP
User + Password
TLS Certificate
Custom +
aws
Custom +
TOTP
Key/Values
Databases
Transit

Vault + AWS + Terraform



DEMO TIME

Next Steps

- Driving more infrastructure setup like: AWS Route 53, ECS, ECR...
- Templating the setup so that we can move more projects over AWS easier and faster.

Complexity made simple

Thanks!

Contact Nimbl3

hello@nimbl3.com

399 Sukhumvit Road, Interchange 21
Klongtoey nua, Wattana
Bangkok 10110

28C Stanley St,
Singapore 068737

20th Floor, Central Tower
28 Queen's Road
Central, Hong Kong

nimbl3.com