Traffic Management with k8s and the ACME protocol In Azure

Alan Blythe

Who am I ?

Architect & Dev

Chief Architect at UniKey Technologies


3364

ABOUT UniKey

A growth stage company

Technology focused early growth stage company

Build, integrate, and deploy our eKey platform through partners or directly to the market

Operating in the consumer retail and commercial spaces through Kevo and SR2

> 1mm users registered, 1200 new users per day, > 1mm units sold with our tech

UniKey's PARTNERS

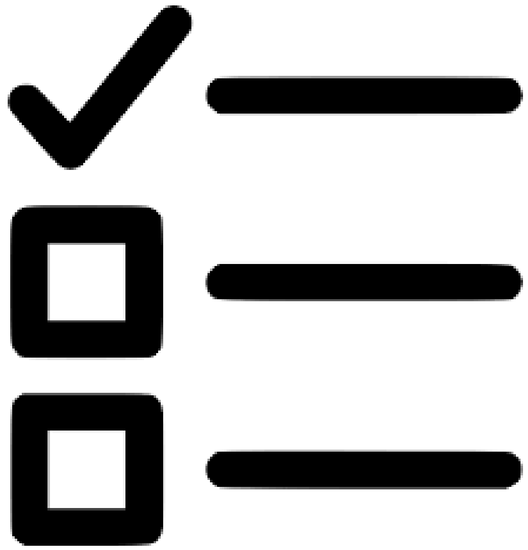
Partners in Residential (IoT), Commercial, and Automotive

What motivated this talk?

- A good low cost/low risk use of k8s
- 3rd party traffic management solutions can be costly
- Managing TLS/x509 certificates can be a PITA
- Managing medium to large scale traffic is difficult

A Quick Review

- Microservices & Orchestration
- Containers, Docker
- IaaS, PaaS, SaaS
- OSI Model: Layer 4, Layer 7
- Traffic Management



Kubernetes and Helm

- A way to deploy, host, manage, and scale various kinds of containerized applications
- Highly extensible through custom resources
- “Helm helps you manage Kubernetes applications — Helm Charts help you define, install, and upgrade even the most complex Kubernetes application.”
- Packages are called “charts”

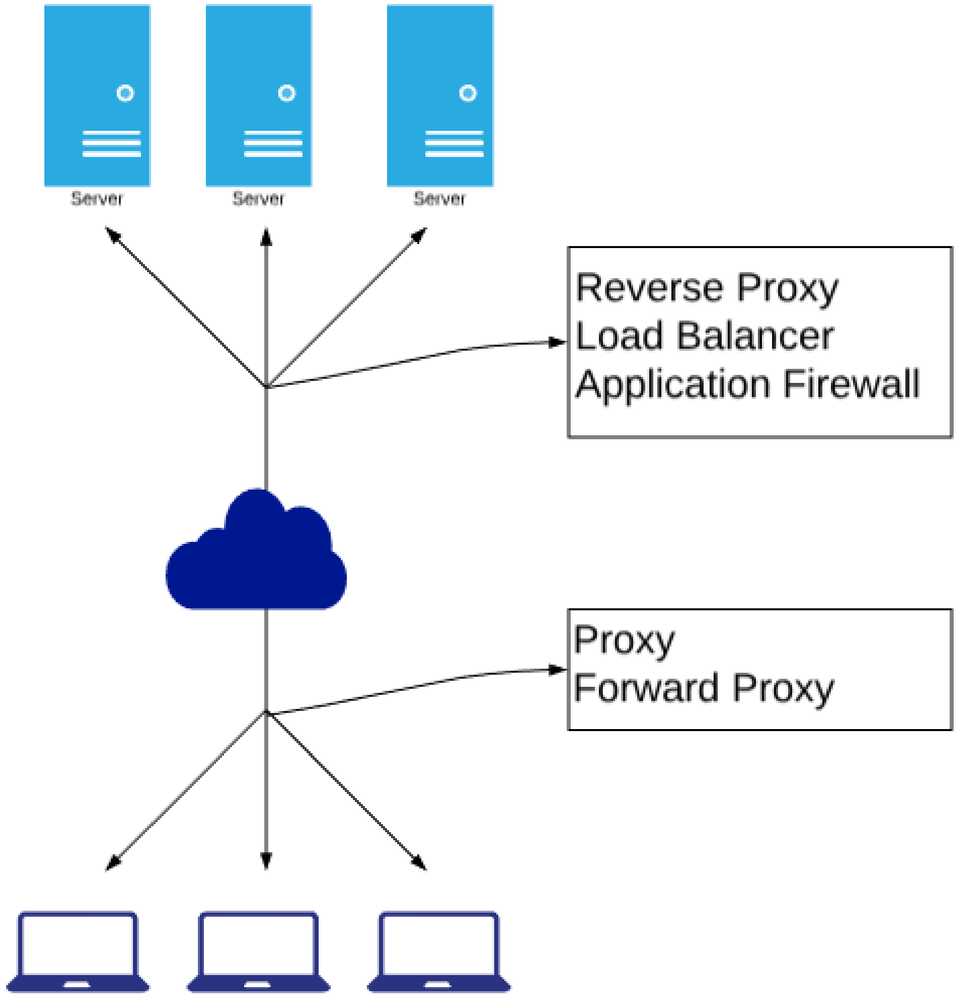
Script 1

- Deploy a new cluster on Azure

Web Traffic Management

- What is it? Examining, Routing, Termination, Inspection, and Protection
- Why? Service Level Commitments, Scalability, Deployment
- Tools and Techniques
  - DNS
  - Proxies
  - Load Balancers
  - Gateways
  - Firewalls
  - SSL Offloading
  - URL Rewriting

Web Traffic Management



Script 2

- Viewing the AKS dashboard

Azure Traffic Management Options

- Azure Traffic Manager: DNS based load balancer
- Azure Load Balancer: Layer 4 load balancer
- Azure Application Gateways: Layer 7 load balancer
  - Don't support URL rewriting
  - SSL Certificates can be offloaded
  - Domain must also be validated on the AppSvc
- Virtual Machines / Containers running NGINX or HaProxy
- Azure API Management: A service for publishing, managing and securing your APIs. Support for OpenAPI v3 (formerly Swagger)
- Third Party Tools from companies such as F5 and CloudFlare

Recommendations

- Use DNS cnames when possible, can’t be used for root domains
- Use a proxy+load balancer that supports URL rewriting
- Use Azure API Management or others such as Apigee for APIs
  - Administrative and Developer Portal
  - Understand usage and Rate Limit
- Use Client Certificates to authenticate proxy backends
- Automate Certificate Management
  - Implement Let’s Encrypt and the ACME protocol
  - No support for Extended Validation certificates

Script 3

- Deploy a website to our k8s cluster

HaProxy

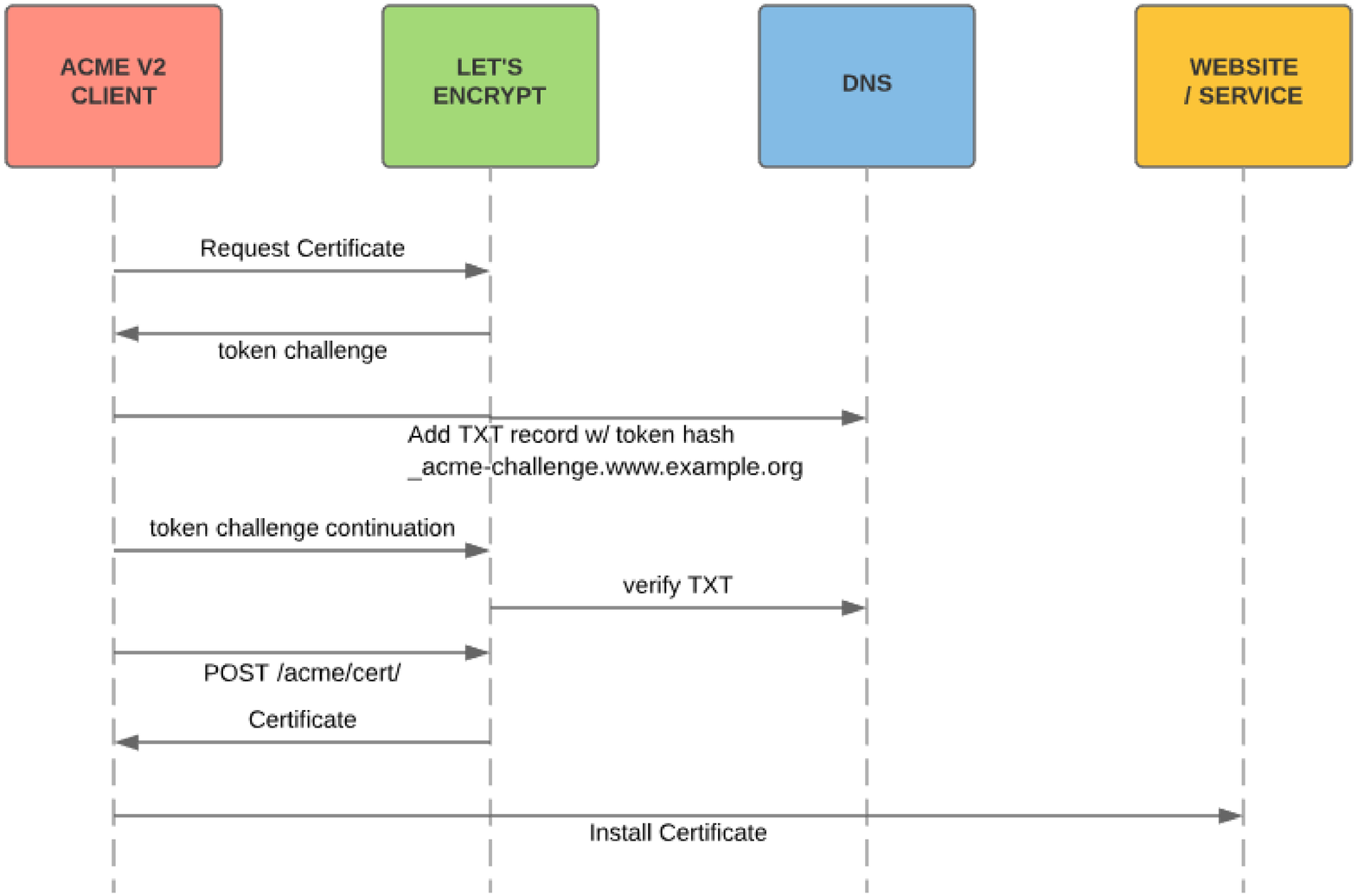
- Layer 4 and Layer 7 load balancer
- First developed in 2000, original author is still involved
- Features: SSL/TLS termination, Rate limiting, URL rewriting, Gzip compression, Client Certificates
- Used by many large companies: GoDaddy, GitHub, Stack Overflow, Reddit, SpeedTest.net, Twitter

ACME Protocol

- Automatic Certificate Management Environment
- Designed and run by the Internet Security Research Group
- The service is marketed as Let's Encrypt
- https://datatracker.ietf.org/doc/rfc8555

- Never have to convert PEM, DER, or PFX again!
- Never forget to renew a SSL cert again!
- Spend more time writing code

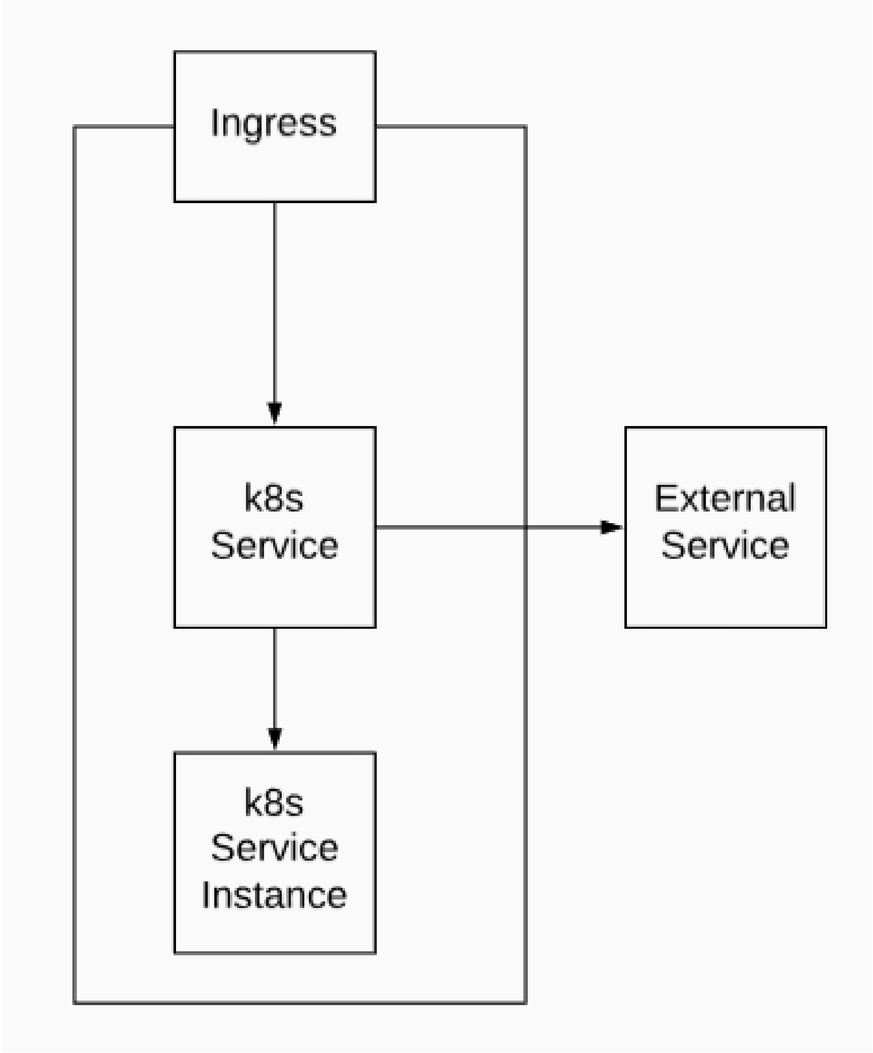
ACME Protocol



AppsCode Helm Chart

- AppsCode Helm Chart for a HaProxy Ingress Controller
- Extends k8s, custom resource types
- Provides automatic certificate management using ACME and Let's Encrypt
- Simplified HaProxy Configuration
- Allows URL rewriting, no need to configure domains on deployed applications
- If services are moved within k8s, Ingress can be updated to point internally

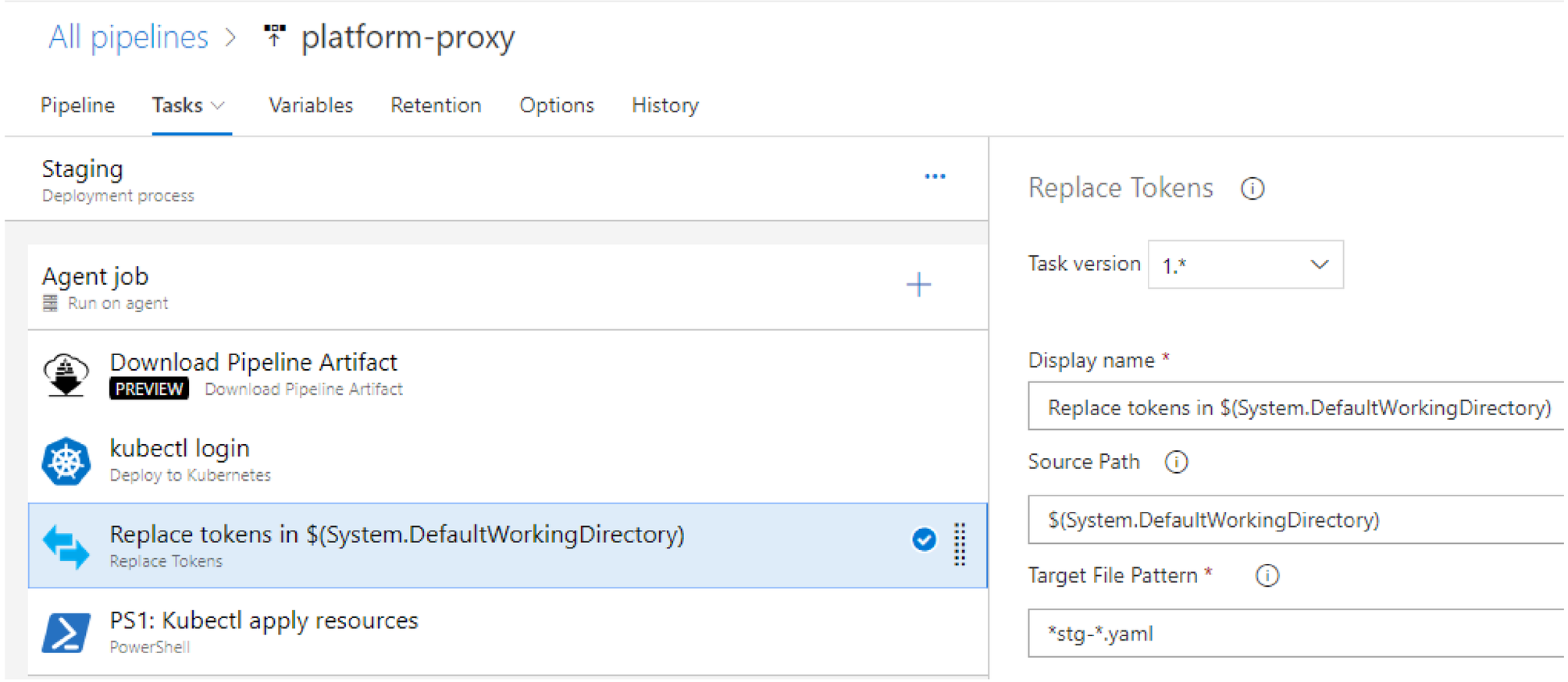
How it works


Ingress
k8s
Service
External
Service
k8s
Service
Instance

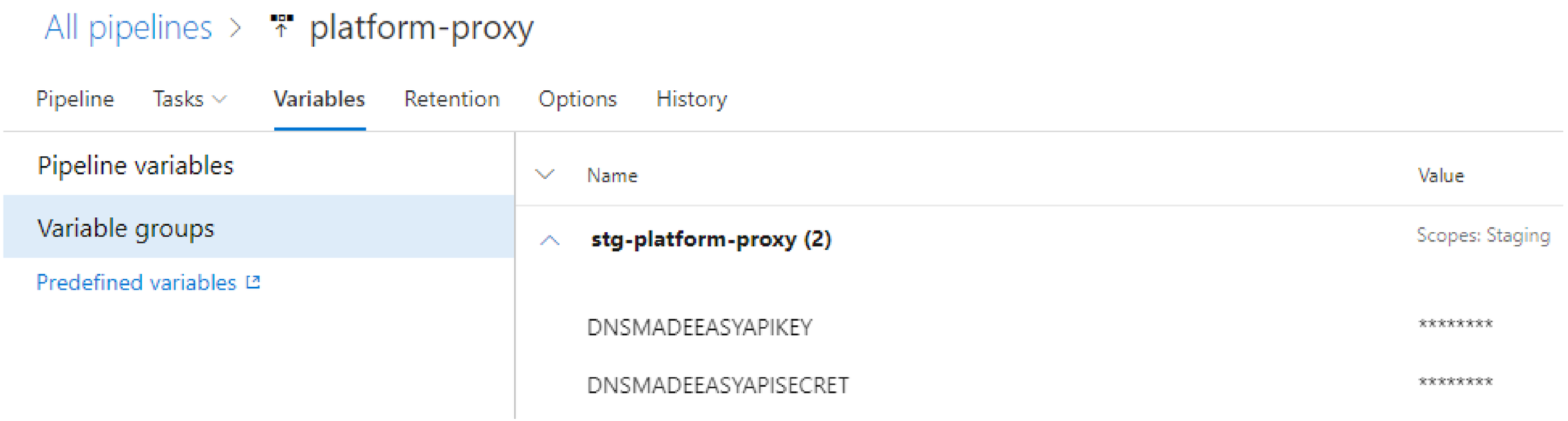
Script 4

- Rewrites

Azure DevOps Release Pipeline



Azure DevOps Release Pipeline



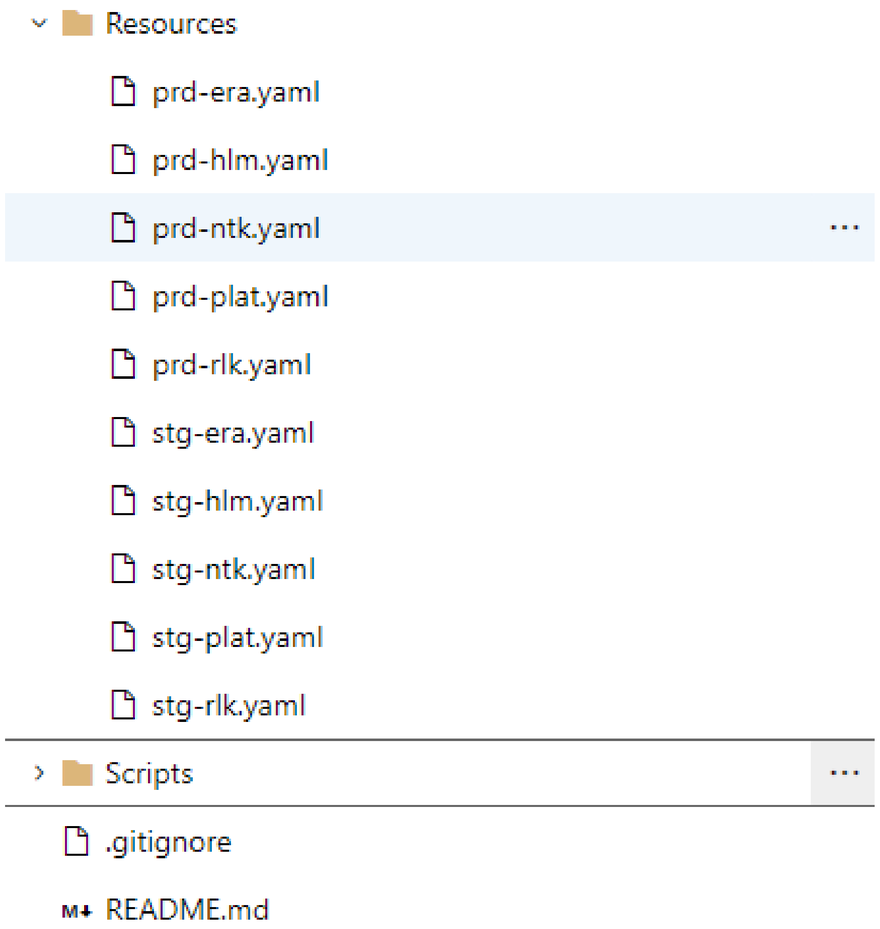
Azure DevOps Release Pipeline

```
param(
[String]$path,
[String]$environmentCode
)

Write-Output 'found yaml files:'

Foreach ($file in Get-ChildItem -Path $path -Filter "$($environmentCode)*.yaml")
{
    Write-Host "Deploying file: $($file.Name)"
    kubectl -v=8 apply -f $file.FullName
}
```

Azure DevOps Release Pipeline



Azure DevOps Release Pipeline

```
4  ---
5  apiVersion: v1
6  kind: Secret
7  metadata:
8    name: voyager-dnsmadeeasy
9    namespace: stg-hlm-proxy
0  type: Opaque
1  stringData:
2    DNSMADEEASY_API_KEY: "__DNSMADEEASYAPIKEY__"
3    DNSMADEEASY_API_SECRET: "__DNSMADEEASYAPISECRET__"
4    DNSMADEEASY_SANDBOX: "0"
5  ---
```

Misc Tools

- SSL Server Tests
- Uptime monitoring tools
- SSL Trackers

Resources

- Let's Encrypt: https://letsencrypt.org
- ACME Clients: https://letsencrypt.org/docs/client-options/
- k8s: https://kubernetes.io/
- AppsCode Voyager: https://github.com/appscode/voyager
- TrackSSL: https://trackssl.com
- SiteUptime: https://www.siteuptime.com
- SSL Server Test: https://www.ssllabs.com/ssltest/

- https://github.com/alanblythe/aksproxy

Thank you

LinkedIn

https://www.linkedin.com/in/alan-blythe/

EMAIL

alan@unikey.com

alanblythe@hotmail.com

TWITTER

@blythealan

GITHUB

alanblythe