Using DNS in your Substrate Cluster

Document Level Classification

200

Introduction

Substrate accounts use Amazon Route 53 provide the following Domain Name System (DNS) capabilities:

1. Substrate account-specific subdomain and public hosted zone.
2. Support for custom subdomains and public hosted zones
3. A private hosted zone for resolving addresses in `ol.epicgames.net`.

Substrate account-specific subdomain

Your Substrate account comes provisioned with a subdomain and public hosted zone that uses the naming convention `<account-identifier>.<account-tier>.<epic-region>.on.epicgames.com`. For example, if your Substrate account is `abcd-dev`, the public hosted zone would be `abcd.dev.use1a.on.epicgames.com`. This provides the following benefits:

1. You have a dedicated subdomain for your service hostnames without conflicting with other clusters. For example, `myservice.abcd.dev.use1a.on.epicgames.com`.
2. You have the capability to automatically validate and renew SSL certificates provisioned using AWS Certificate Manager (ACM).

Your Substrate cluster uses external-dns to automatically manage DNS records in Route 53 based on `host`-s defined in your *Ingress* configurations. This provides the following benefits:

1. You define the hostname for your service using *Ingress* configurations.
2. You do not need to manually manage or update DNS records.

Using the account-specific public hosted zone allows you to get deploying quickly. However, service hostnames (for example,

myservice.abcd.dev.use1a.on.epicgames.com) cannot be migrated to other AWS/Substrate accounts or regions. You can use a custom subdomain instead. **It is strongly recommended that you setup custom subdomain and public hosted zone for your public/user-facing services**.

Custom subdomains

Custom subdomains and public hosted zones allow you to provision and manage your own `*.on.epicgames.com` subdomain - for example `my-brand-services.on.epicgames.com`. Setting this up is a one-time manual process (or using Terraform), requires access to Route 53 in OldProd, and involves the following steps:

1. Inventory existing subdomains in the *on.epicgames.com* hosted zone (in OldProd) to check availability and ensure there are no conflicts with your intended custom subdomain.

2. Create a new public hosted zone in your substrate account with your subdomain name.

3. Create a NS record in the *on.epicgames.com* hosted zone (in OldProd) to delegate responsibility to your new custom public hosted zone.

If you have Route 53 access to OldProd, review these instructions to perform the above steps using Terraform. If you do not have access to OldProd, or would like assistance, reach out via slack in #cloud-ops-support-ext.

DNS resolution for `ol.epicgames.net`

Resolution of ol.epicgames.net records is performed via a centralized Route53 Resolver Endpoint. To get to the central resolver, the VPC needs to be attached to the Service Network, and also opt into resolving ol.epicgames.net records over the Service Network. This opt-in will mean a route 53 forwarding rule for the ol.epicgames.net domain is associated with the VPC allowing the resolution to occur.

Route 53 External DNS

The route53-external-dns service is a small operator that lives within a Kubernetes cluster. route53-external-dns watches kube-api for events on services. When a service adds, removes, or changes an annotation prefixed with `external-dns.alpha.kubernetes.io`, then route53-external-dns will automatically connect to AWS Route 53 and update the DNS records for that service accordingly. Essentially, this service integrates the Kubernetes environment to Route 53.

Applying external-dns Annotations to Your Ingress Configuration

To define the DNS record for the ingress. Use the external-dns annotation.
https://github.com/kubernetes-sigs/external-dns/blob/master/docs/faq.md#how-do-i-specify-a-dns-name-for-my-kubernetes-objects

Annotations: https://github.com/kubernetes-sigs/external-dns/blob/master/docs/annotations/annotations.md

Commonly used Annotations Within Epic

```
external-dns.alpha.kubernetes.io/hostname: dns-name.<account>.<env>.use
```

```
external-dns.alpha.kubernetes.io/hostname: dns-name.<account>.<env>.use
external-dns.alpha.kubernetes.io/hostname: dns-name.<account>.<env>.use
external-dns.alpha.kubernetes.io/set-identifier: "1"
external-dns.alpha.kubernetes.io/ingress-hostname-source: annotation-on
```