Managing traffic between applications in your Substrate cluster

Document Level Classification

200

Introduction

In a scenario where you have multiple applications (`serviceA` and `serviceB`) deployed to the same Substrate cluster, you can use the following options to allow network traffic between the applications (`serviceA` needs to communicate with `serviceB`).

Option 1 - Use the *Ingress* pattern

High-level steps:

1. `serviceB` configures an *Ingress* as described in Managing inbound traffic to your application.
2. The security group for `serviceB` authorizes network traffic from the cluster itself using annotations as described in Managing inbound traffic to your application.
3. `serviceA` can communicate with `serviceB` using the hostname configured on the Application Load Balancer (e.g., `serviceB.abcd.dev.use1a.on.epicgames.com`)

With this option, the inbound traffic to `serviceB` is always passing through the Application Load Balancer, allowing for HTTPS, request logging, metrics, scaling, etc. However, the drawback is that the traffic originating from `serviceA` needs to egress from the cluster and connect using the Application Load Balancer hostname. This traffic path can add latency to the request.

Option 2 - Use Kubernetes Service DNS

High-level steps:

1. `serviceB` configures a Service as described in Create a Kubernetes Service.
2. `serviceA` can communicate with `serviceB` using the Service DNS name (e.g., `serviceB.mynamespace.svc.cluster.local`)

With this option, traffic between `serviceA` and `serviceB` is local to the cluster and will have lower latency compared to Option 1. However, from `serviceB`'s perspective, additional metrics and capabilities provided by the Application Load Balancer is not available for these requests.

Page Information:
Page ID: 81068462