Advanced Networking for Substrate

Document Level Classification

300

- Why would I use IxM? What value does it bring me? What is its purpose?
- I have an advanced understanding of AWS networking. I want to see the detailed connectivity within a Substrate Environment as an architecture diagram.
- Where can I get help on Substrate networking related questions?

Introduction

The purpose of this page is to answer the most Frequently Asked Questions on the subject of Advanced Network Topics in Substrate. It is mostly an FAW style page, except for the **'Update'** sections below

Updates being made to Substrate networking (Net3.0)

Once Net2.5 (Service Network) has been implemented (see prompt below), the next steps will be to migrate to Net3.0. This will likely be done around mid 2024. It will consist of Private NAT Gateways, EKS in IP-v6 only mode, Transit Gateway and VPC Lattice. Inter Net3.0 services would communicate via IPv6. Non Kubernetes workloads should run dual stack so as to communicate with Substrate or Corporate. BYOIP will let Epic use IPv6 CIDRs that are owned by them, assign them to their VPCs.

Using VPC Lattice for controlling access between services can be done using Terraform Infrastructure-as-Code (IaC) and would help remove the need for custom tooling like IxM for developers.

Historically, multi-region was seen as unneeded for services like the Substrate Proxy, as all the downstream systems were in a single region, but some teams are hitting EC2 capacity constraints that are driving multi-region expansion, so there is a first-mover problem. Epic needs to

have patterns to support multi-region networking and Net3.0 hosted services are strong candidates to be those first movers.

Management access to Net3.0 resources will be done via Teleport.

Updates being made to Substrate networking (Phase Net 2.5 - Service Network)

Due to issues in scalability with Substrate networking and overly complicated network solutions like Substrate Proxy, networking changes will soon be made to how it works. The Service Network simplifies connectivity without wholesale change of the Substrate network. It will allow private communication between Service Network attached VPCs and services instead of public communication in the current Substrate architecture, eliminate the need for Substrate Proxy to communicate with services in OldProd, and also eliminate the need to use IxM to expose services publicly; instead opting for a simpler Security Group based management.

- For more information please see the page https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93488715.
- For information on the currently existing architecture, see the Networking main page.

I want to be able to access an RDS database within my Substrate environment securely. How do I do this?

See the following link to access RDS using Teleport: Accessing RDS Databases and EC2 Instances with Teleport

I want to be able to securely access an RDS database in another Substrate or AWS account. How do I do this?

The method would be similar to accessing an RDS database in the same account, as Teleport provides a Central place to access EC2 instances and RDS databases across the organization. You may view Accessing RDS Databases and EC2 Instances with Teleport

I want to securely access another team's services in a different Substrate account. How do I do this?

On the other team's services; you want to carry out the following: Managing inbound traffic to your application.

Essentially, the other team would need to create a Kubernetes Ingress object (along with an Ingress Controller), which is what allows its Service to be exposed outside the cluster.

I want to secure my services running in Substrate with an HTTPS Certificate. How do I do this using ACM?

Follow the instructions here: Creating an ACM certificate in your Substrate account.

WARNING: The remainder of the info in this guide is current through Q3 of 2023; but is under active development to move to Net2.5 (Service Network). Please contact Rob Iballand team for the most up to date information on the current state of Substrate networking.

I want to be able to visualize and conceptually understand what Substrate networking gives me without needing to know about AWS networking.

WARNING: This is the current method of Substrate functionality; Throughout the remainder of 2023, Net2.5 and eventually Net3.0 will be implemented. The following paragraph is kept for historical purposes.

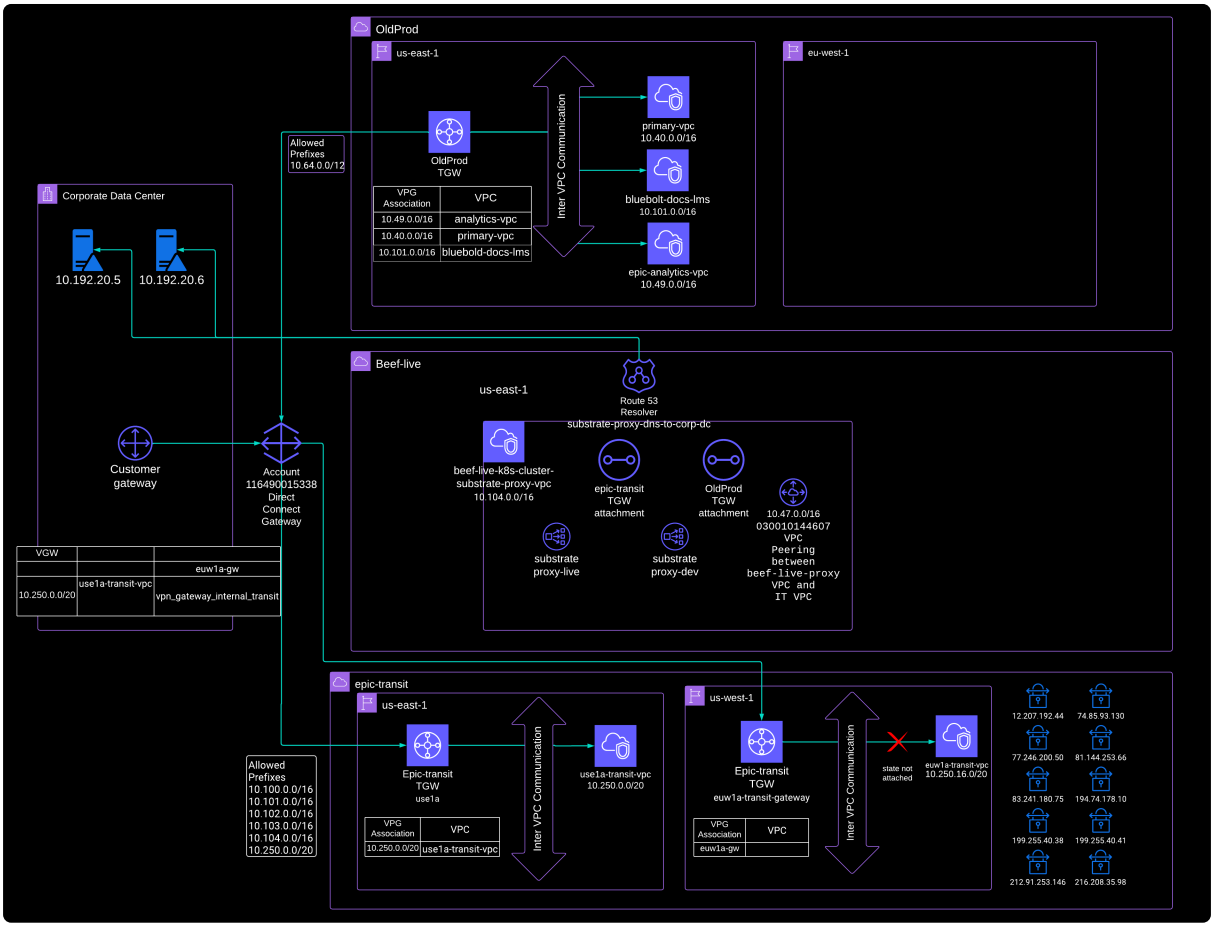
Substrate is a standardized environment for developers to deploy their containerized workloads to Kubernetes clusters. Substrate accounts usually re-use the same CIDR block for each network. Substrate requires services to access each other via public networks and exposed load balancers or API Gateways.

By default, Substrate accounts cannot communicate with the OldProd and Epic-Transit Accounts because they repeat the use of the same IP CIDR block. There is a Substrate proxy VPC as shown below which allows this communication.

The beef-live account is where the Substrate Proxy VPC beef-live-k8s-cluster-substrate-proxy-vpc 10.104.0.0/16. is located. This VPC has TGW attachments from both OldProd TGW and Epic-Transit TGW. A route destined to 10.104.0.0/16, the beef-live-k8s-cluster substrate-proxy-vpc (vpc-0f329e34b5039912d), is present in both OldProd and Epic-Transit TGW route tables.

The Substrate Proxy Environments cannot communicate with the other networks in OldProd via the TGW, because the network can't route back to specific Substrate networks, when they re-use the same IP CIDR block. The Substrate Proxy can communicate with the Primary-VPC 10.40.0.0/16 and the epic-analytics-vpc 10.49.0.0/16, because they are distinct IP CIDR blocks and have explicit routing.

Essentially, Substrate accounts can communicate with OldProd, Epic Transit and even On-prem via the Substrate proxy, but not vice versa. See the diagram on the bottom of this page for more clarity.



The Substrate Proxy is a reverse proxy that connects services running in different workload accounts. These service's cannot route back to the Substrate account because of the IP re-use, so service DNS records are created pointing to an Endpoint IP in the Substrate network. That endpoint carries traffic directly to the Substrate Proxy outside the normal routed network.

The Substrate Proxy is a set of two EKS clusters running Nginx containers that are exposed to Substrate accounts via PrivateLink VPC Endpoints that listen on TCP 22, 443, 445, and 3269. The Substrate accounts create a local PrivateLink Endpoint that is linked to the Substrate Proxy PrivateLink Service.

Substrate to Substrate connections need to connect to public endpoints because the IP address conflicts between Substrate VPCs prevent them from being peered or routed through networks.

Why would I use IxM? What value does it bring me? What is its purpose?

WARNING: This is no longer the current method of Substrate functionality. The following paragraph is kept for historical purposes.

The primary use-case for IxM is was to grant access between services in different AWS accounts over the public internet. So for example, to access the publicly facing ALB of one service in Account A, the public IP's of the NAT Gateway in Account B need to be whitelisted in the security group of the ALB in Account A. Rather than having to manually remember and whitelist these specific IP ranges / CIDRs, a 'prefix list' can represent a range of CIDR's with an easy to remember name. All IxM did was manage these prefix lists for you, so you don't have to manage even those.

However, since the rollout of the Service Network, most teams should be looking at migrating from internet facing load balancers to internal load balancers on the Service Network. If your load balancer is on the Service Network you should no longer be using IxM and you should instead be managing ingress traffic using Ingress annotations. For instructions on using ingress annotations for internal load balancers reference Manage inbound network traffic to your Internal Application Load Balancer (with Service Network Annotations)

In cases where your load balancer is internet facing, teams should also be looking at managing ingress traffic with annotations instead of using IxM as this allows you to use IaC to manage access instead of relying on the

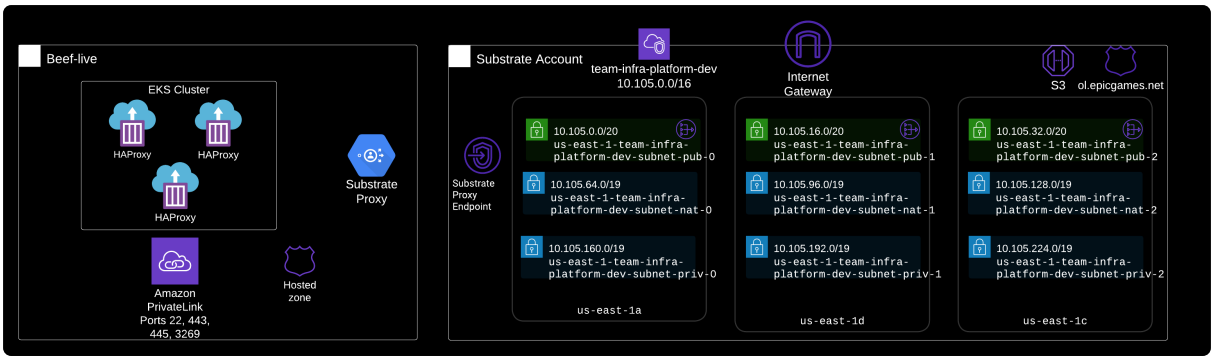
IxM tool. For instructions on using ingress annotations for internet facing load balancers reference Manage inbound network traffic to your Public Application Load Balancer (with Security Groups, Prefix Lists, or Inbound CIDR annotations)

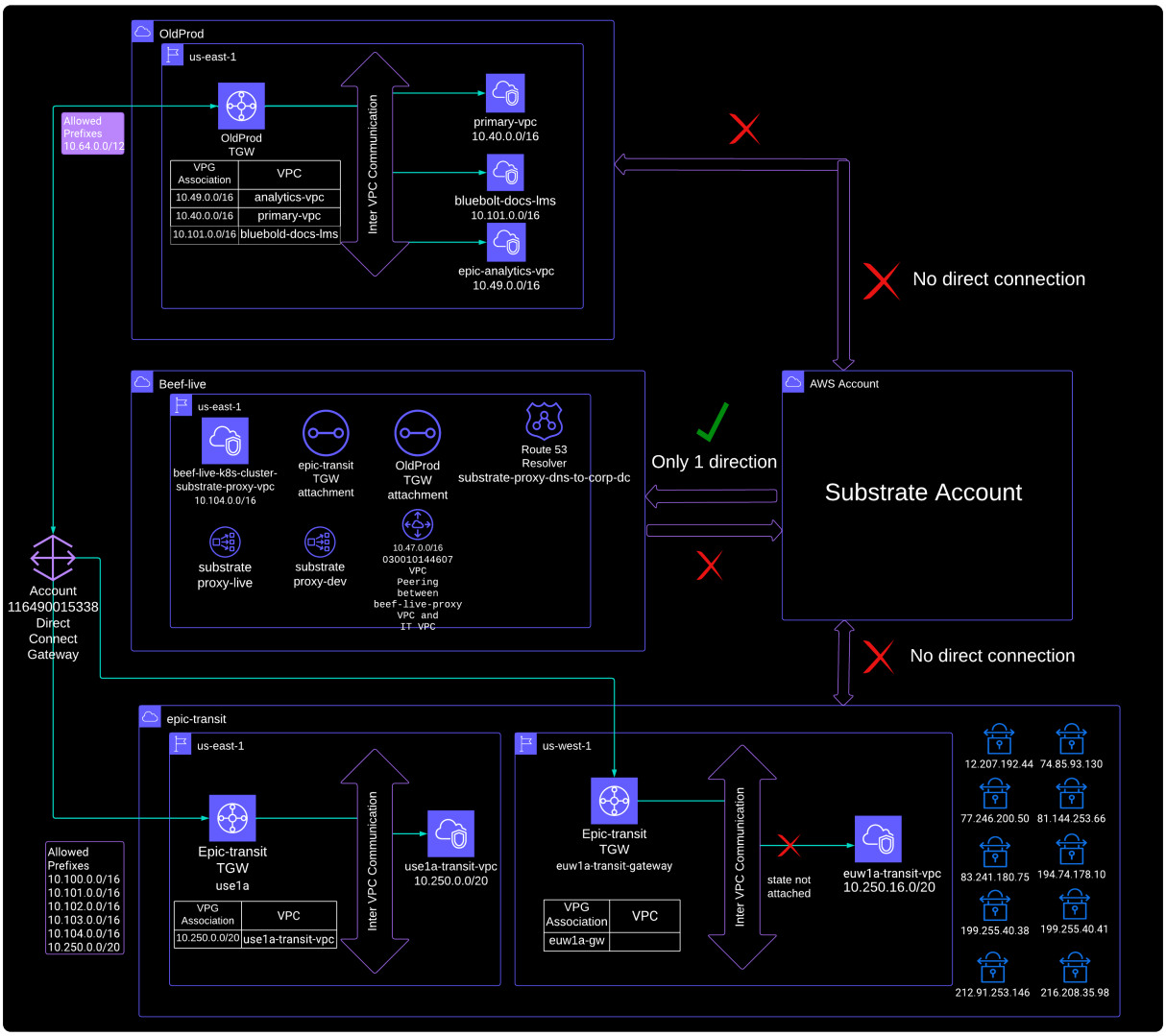
I have an advanced understanding of AWS networking. I want to see the detailed connectivity within a Substrate Environment as an architecture diagram.

WARNING: This is the current method of Substrate functionality; Throughout the remainder of 2023, Net2.5 and eventually Net3.0 will be implemented. The following paragraph is kept for historical purposes.

In the first diagram below, the VPC towards the right is an example of a single Substrate VPC. Each Substrate VPC has a single Internet Gateway and usually a Transit Gateway. The Substrate VPC can communicate with OldProd and Epic Transit via the Substrate proxy *only*. The Substrate proyx is shown in the beef-live account towards the left.

The lower diagram shows an abstracted view of a Substrate account but more details with how the connections between these accounts are uni-directional. As mentioned above, if the reverse is required for any reason (OldProd to Substrate) then it needs to use the prefix lists (IxM) rather than go through the Substrate proxy.


Beef-live
EKS Cluster
HAProxy
HAProxy
HAProxy
Substrate Proxy
Amazon PrivateLink Ports 22, 443, 445, 3269
Hosted zone
Substrate Account
team-infra-platform-dev 10.105.0.0/16
Internet Gateway
S3
ol.epicgames.net
Substrate Proxy Endpoint
10.105.0.0/20 us-east-1-team-infra-platform-dev-subnet-pub-0
10.105.64.0/19 us-east-1-team-infra-platform-dev-subnet-nat-0
10.105.160.0/19 us-east-1-team-infra-platform-dev-subnet-priv-0
us-east-1a
10.105.16.0/20 us-east-1-team-infra-platform-dev-subnet-pub-1
10.105.96.0/19 us-east-1-team-infra-platform-dev-subnet-nat-1
10.105.192.0/19 us-east-1-team-infra-platform-dev-subnet-priv-1
us-east-1d
10.105.32.0/20 us-east-1-team-infra-platform-dev-subnet-pub-2
10.105.128.0/19 us-east-1-team-infra-platform-dev-subnet-nat-2
10.105.224.0/19 us-east-1-team-infra-platform-dev-subnet-priv-2
us-east-1c


OldProd
us-east-1
Allowed Prefixes 10.64.0.0/12
OldProd TGW
VPG Association | VPC
10.49.0.0/16 | analytics-vpc
10.40.0.0/16 | primary-vpc
10.101.0.0/16 | bluebold-docs-lms
Inter VPC Communication
primary-vpc 10.40.0.0/16
bluebolt-docs-lms 10.101.0.0/16
epic-analytics-vpc 10.49.0.0/16
No direct connection
Beef-live
us-east-1
beef-live-k8s-cluster-substrate-proxy-vpc 10.104.0.0/16
epic-transit TGW attachment
OldProd TGW attachment
Route 53 Resolver substrate-proxy-dns-to-corp-dc
substrate proxy-live
substrate proxy-dev
10.47.0.0/16 030010144607 VPC Peering between beef-live-proxy VPC and IT VPC
Only 1 direction
AWS Account
Substrate Account
Account 116490015338 Direct Connect Gateway
No direct connection
epic-transit
us-east-1
Epic-transit TGW use1a
VPG Association | VPC
10.250.0.0/20 | use1a-transit-vpc
Inter VPC Communication
use1a-transit-vpc 10.250.0.0/20
Allowed Prefixes
10.100.0.0/16
10.101.0.0/16
10.102.0.0/16
10.103.0.0/16
10.104.0.0/16
10.250.0.0/20
us-west-1
Epic-transit TGW euw1a-transit-gateway
VPG Association | VPC
euw1a-gw
Inter VPC Communication
state not attached
euw1a-transit-vpc 10.250.16.0/20
12.207.192.44
74.85.93.130
77.246.200.50
81.144.253.66
83.241.180.75
194.74.178.10
199.255.40.38
199.255.40.41
212.91.253.146
216.208.35.98

Where can I get help on Substrate networking related questions?

Reach out on Slack on *#cloud-ops-support-ext* or *#cloud-engineering-ext*. (Ask your manager to add you to those channels if you aren't already in them). For a more direct contact, reach out to Rob Iballand his team