cisco Live!

Let's go

The bridge to possible

Deploying IPv6 in the Cloud

Shannon McFarland, CCIE #5245
Distinguished Engineer – Emerging Technologies & Incubation
@eyepv6

BRKIPV-3927

IPv6

Deploying and Securing IPv6

Learn from specialists about IPv6 in their respective area. From the fundamentals of the Neighbor Discovery Protocol, deployment guidelines, security in the network, and troubleshooting.

START

Monday, June 5 | 8:30 a.m.
BRKENT-2109
Let's Deploy IPv6 NOW

Monday, June 5 | 9:30 a.m.
BRKSEC-2044
Secure Operations for an IPv6 Network

Monday, June 5 | 11:00 a.m.
BRKMER-1752
Experience the Journey to IPv6-Only With Cisco Meraki

Monday, June 5 | 1:00 p.m.
BRKIPV-2191
IPv6:: It's Happening!

Tuesday, June 6 | 10:30 a.m.
BRKIPV-2000
Verifying your Systems Transition to IPv6

Tuesday, June 6 | 1:00 p.m.
BRKIPV-2751
IPv6 with Cisco IOS Routing and Meraki Access - A Practical Guide

Tuesday, June 6 | 2:30 p.m.
BRKENT-2008
Goodbye Legacy, the move to an IPv6-Only Enterprise

Tuesday, June 6 | 3:00 p.m.
BRKENT-3002
IPv6 Security in the Local Area with First Hop Security

Tuesday, June 6 | 3:00 p.m.
BRKIPV-1616
IPv6 - What Do you Mean there isn't a Broadcast?

FINISH

Wednesday, June 7 | 10:30 a.m.
BRKIPV-3927
Deploying IPv6 in the Cloud

cisco Live!
Las Vegas, NV | June 4-8, 2023

If you are unable to attend a live session, you can watch it in the On-Demand Library after the event.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cisco Webex App

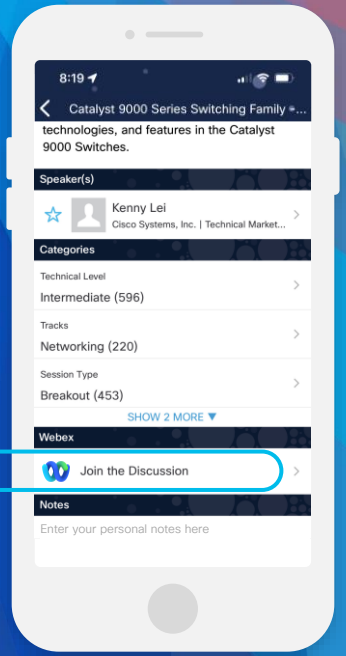
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 9, 2023.



https://ciscolive.ciscoevents.com/ciscolivebot/#BRKIPV-3927

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Agenda

- Introduction
- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- Docker
- Kubernetes (k8s)
- Service Mesh

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Dislaimer

- You won’t learn the basics of IPv6
- You won’t learn the basics of Cloud
- You won’t see a single reference to a Cisco product or solution <insert record scratch here> ☺
- This is about technical awareness of IPv6 deployment in Cloud environments, related services and open source projects

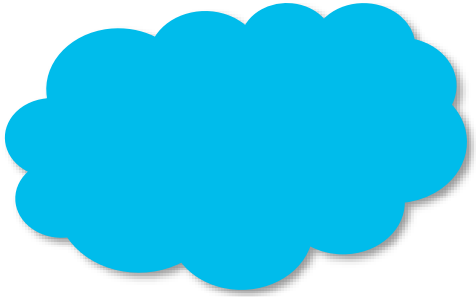
© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Intro – IPv6 & Cloud

This is an IPv6 address

2001:db8:bad:beef::/64

This is a cloud



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Goal #1:
Can we expose an IPv6-enabled application that resides in a Cloud environment to internal and external users?

Goal #2:
Can we do it IPv6-only?
(no dual-stack)

The Hard Stuff – IPv6 + Cloud

- Inside of a cloud stack you have a lot of moving parts and they all ride on IP:
  - API endpoints
  - Provisioning, Orchestration and Management services
  - Boatload of protocols and databases and high-availability components
  - Virtual networking services <> Physical networking
- It has been a bumpy road to getting a solid IPv6 implementation in any cloud stack
- If you are doing IPv6 with a hyperscaler (AWS, Azure, GCP, etc.), lots of the IPv6 address design decision making is made for you (you will see what I mean)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

AWS

Reference

AWS IPv6 Support

- Status: https://docs.aws.amazon.com/general/latest/gr/aws-ipv6-support.html
- Best Practices (nice doc): https://docs.aws.amazon.com/whitepapers/latest/ipv6-on-aws/IPv6-on-AWS.html
- Reference Architecture (excellent): https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/IPv6-reference-architectures-for-AWS-and-hybrid-networks-ra.pdf

Service name	Dual stack support	IPv6 only support	Public endpoints support IPv6	Private endpoints support IPv6
AWS App Mesh	Yes	Yes	Yes	No
Amazon Athena	Yes	No	Yes	No
Amazon Aurora	Yes	No	Yes	No
Amazon CloudFront	Yes	No	No	No
AWS Cloud Map	Yes	Yes	Yes	No
AWS Database Migration Service	Yes	No	No	No
AWS Direct Connect	Yes	Yes	No	No
Amazon EC2	Yes	Yes	Yes	No
Amazon ECS	Yes	No	No	No
Amazon EKS	Yes	Yes	No	No
Elastic Load Balancing	Yes	Yes	No	No
Amazon ElastiCache	Yes	Yes	No	No
AWS Fargate	Yes	No	No	No
AWS Global Accelerator	Yes	No	No	No
AWS IoT	Yes	Yes	No	No
AWS Lambda	No	No	Yes	No
Amazon Lightsail	Yes	No	No	No
AWS Network Firewall	Yes	Yes	No	No
AWS PrivateLink	Yes	Yes	Yes	No
Amazon RDS	Yes	No	Yes	No
Amazon Route 53	Yes	Yes	No	No
Amazon S3	Yes	No	Yes	No
AWS Secrets Manager	Yes	No	Yes	No
AWS Shield	Yes	Yes	No	No
AWS Site-to-Site VPN	Yes	No	Yes	No
AWS Transit Gateway	Yes	No	Yes	No
Amazon VPC	Yes	Yes	Yes	No
AWS WAF	Yes	Yes	No	No
Amazon WorkSpaces	Yes	No	No	No

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

AWS VPC IPv6 Support

- Lots of support with lots of options with lots of gotchas
- VPC connectivity options (VPC peering, TGW, etc.): https://docs.aws.amazon.com/whitepapers/latest/ipv6-on-aws/amazon-vpc-connectivity-options-for-ipv6.html
- IPv6-only
  - Must still have an IPv4 CIDR assigned, but you can have IPv6-only subnets
  - Private/Public Subnets + IGW/EIGW (both use GUA prefixes)
  - Private subnets use EIGWs for outbound IPv6 connectivity
- Dual stack – Private/Public Subnets + IGW/EIGW (both use GUA prefixes)
- IPv6-to-IPv4 options with DNS64/NAT64/egress gateways
- BYOIPv6:
  - https://aws.amazon.com/blogs/networking-and-content-delivery/bring-your-ipv6-address-space-to-amazon-vpc-ip-address-manager-ipam/
  - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-byoip.html#prepare-for-byoip

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

AWS VPC with IPv6

```
# aws ec2 create-vpc --cidr-block 10.1.0.0/16 --amazon-provided-ipv6-cidr-block

# aws ec2 describe-vpcs --vpc-id vpc-0a09572a6ec42042d --output=json | grep "Ipv6CidrBlock"
            "Ipv6CidrBlockAssociationSet": [
                    "Ipv6CidrBlock": "2600:1f13:3ca:f800::/56",
                    "Ipv6CidrBlockState": {
```

```
# aws ec2 create-subnet --vpc-id vpc-0a09572a6ec42042d --ipv6-cidr-block 2600:1f13:3ca:f801::/64 --ipv6-native

# aws ec2 describe-subnets --subnet-id subnet-00eec22f685de45df --output=text | grep
"IPV6CIDRBLOCKASSOCIATIONSET"    12s
IPV6CIDRBLOCKASSOCIATIONSET  subnet-cidr-assoc-0c63a6e7564645d42     2600:1f13:3ca:f801::/64
```

- --ipv6-native - indicates that this is an IPv6-only subnet
- Blah, blah, blah – Internet Gateways, route tables, security groups, etc..

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

AWS EC2 IPv6 Support

- IPv6-only
  - Requires all the usual stuff (IGW, Routes, etc)
  - EC2 Instance Connect still requires an IPv4 address
  - When an EC2 instance is launched on an IPv6-only subnet, RBN (Resource-based Naming) is used: “Resource name: i-097019ae9f2908087.ec2.internal”

The instance does not have a public IPv4 address

To connect using the EC2 Instance Connect browser-based client, the instance must have a public IPv4 address.

```
# ping i-097019ae9f2908087.ec2.internal
64 bytes from i-097019ae9f2908087.ec2.internal (169.254.118.19): icmp_seq=1 ttl=255 time=0.020 ms

# ping -6 i-097019ae9f2908087.ec2.internal
64 bytes from i-097019ae9f2908087.ec2.internal (2600:1f18:d:f70a:8204:7788:7feb:7892): icmp_seq=1 ttl=64
```

```
# ssh -i “<key_name>” ec2-user@2600:1f18:d:f70a:8204:7788:7feb:7892
```

```
# ping -6 www.google.com
PING www.google.com(bl-in-f106.1e100.net (2607:f8b0:4004:c17::6a)) 56 data bytes
64 bytes from bl-in-f106.1e100.net (2607:f8b0:4004:c17::6a): icmp_seq=1 ttl=99 time=1.21 ms
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

AWS EC2 IPv6-only + AWS Local Services

IMDS(Instance Metadata Service) access over IPv6:

```
# aws ec2 --region us-east-1 modify-instance-metadata-options --instance-id i-0f5f622b322259b4d --http-protocol-ipv6 enabled
```

```
# curl -w "\n" --connect-timeout 10 'http://[fd00:ec2::254]/latest/meta-data/hostname'
i-0f5f622b322259b4d.ec2.internal
```

Time sync update: /etc/chrony.conf:

```
# use the Amazon Time Sync Service (if available)
server fd00:ec2::123 prefer iburst minpoll 4 maxpoll 4
#server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

```
# chronyc sources
^* fd00:ec2::123                 3   4   177    15
-62us[  -82us] +/-  538us
^- 2604:4300:f35:16::3           2   6    17    37
+4085us[+4068us] +/-   31ms
```

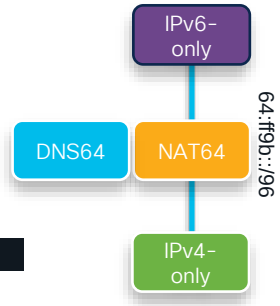
DNS:

```
# dig @fd00:ec2::253 -6 AAAA cisco.com
. . .<output summarized>
;; ANSWER SECTION:
cisco.com.        300    IN    AAAA  2001:420:1101:1::185

;; Query time: 33 msec
;; SERVER: fd00:ec2::253#53(fd00:ec2::253)
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

IPv6-only TO IPv4-only (NAT64/DNS64)



Enable DNS64 on IPv6-only Subnet:

```
# aws ec2 --region us-east-1 modify-subnet-attribute --subnet-id <subnet_id> --enable-dns64
```

Add a route to the DNS64 prefix (RFC 6052):

64:ff9b::/96	nat-0a8f285d64dbc09b9

Create security group rules for appropriate protocols/src/dst:

IP version	Type	Protocol	Port range	Source
IPv4	HTTP	TCP	80	0.0.0.0/0

Check DNS64 operation:

```
# nslookup ec2-44-210-107-93.compute-1.amazonaws.com
Server:     fd00:ec2::253
Address:    fd00:ec2::253#53

Non-authoritative answer:
Name: ec2-44-210-107-93.compute-1.amazonaws.com
Address: 44.210.107.93
Name: ec2-44-210-107-93.compute-1.amazonaws.com
Address: 64:ff9b::2cd2:6b5d
```

Rock-n-roll:

```
# curl -6 -o /dev/null -s -w "%{http_code}\n"
http://ec2-44-210-107-93.compute-1.amazonaws.com
200
# curl -6 -o /dev/null -s -w "%{http_code}\n"
http://[64:ff9b::2cd2:6b5d]
200
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

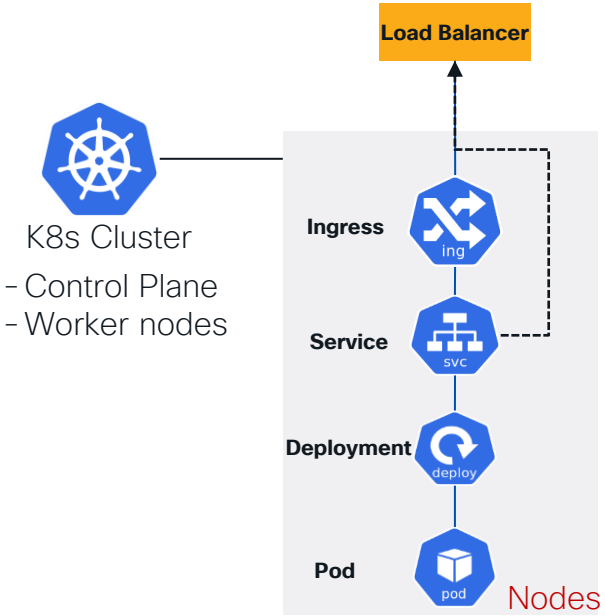
AWS NLB/ALB IPv6 Support

- Network Load Balancer:
  - IPv6 support (LB, listener, target groups)
  - IP Address Type: “ipv4” or “dualstack” (no IPv6-only)
- Application Load Balancer:
  - IPv6 support (LB, listener, target groups)
  - IP Address Type: “ipv4” or “dualstack” (no IPv6-only)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Amazon EKS

Kubernetes Primer



- Kubernetes Cluster
  - All the stuff on the left except for the “Load Balancer” ☺
  - Control plane components
  - Worker nodes that are registered with the control plane
- Pod
  - One or more containers
- Deployment
  - Create and manage Pods
- Service
  - Expose application(s) that run on Pods
- Ingress & Ingress controller (optional)
  - Exposes HTTP/HTTPS routes for services
  - The controller helps create/manage load balancers
- Load Balancer
  - An ingress/service-managed resource that creates/manages environment-specific load balancing resources (AWS LB, MetalLB, etc.)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Amazon EKS IPv6 Support

- Launched in January of 2022
- The most important thing you will read about EKS and IPv6: https://docs.aws.amazon.com/eks/latest/userguide/cni-ipv6.html
- Leverages existing VPC IPv6 features
  - /56 Prefix – Global Unicast Address
    - /64 per subnet
    - If you want a "private" IPv6 deployment, change route table (::/0) to use Egress-only Internet Gateway (EIGW): https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html
- Node: IPv6 prefix assignment occurs at node startup
- Currently only supported on AWS Nitro-based EC2 instances
- eksctl is your friend (auto-adds proper tagging on subnets, etc..) – https://docs.aws.amazon.com/eks/latest/userguide/cni-ipv6.html
- AWS resources use NAT64 (AWS NAT Gateway) and DNS64 (AWS Route 53) to communicate with IPv4-only destinations – EKS uses egress-only IPv4
- Kubernetes services receive IPv6 address out of the ULA space – Must (still) expose services via a load balancer – lots of stuff involved
  - https://docs.aws.amazon.com/eks/latest/userguide/aws-load-balancer-controller.html
  - https://docs.aws.amazon.com/eks/latest/userguide/network-load-balancing.html#network-load-balancing-service-sample-manifest

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

eksctl – Example (1)

Create a cluster:

```
# eksctl create cluster \
-f v6-eksctl.yaml
```

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: v6-eksctl
  version: "1.22"
  region: us-east-2

kubernetesNetworkConfig:
  ipFamily: IPv6

addons:
  - name: vpc-cni
    version: latest
  - name: coredns
    version: latest
  - name: kube-proxy
    version: latest

vpc:
  clusterEndpoints:
    publicAccess: true
    privateAccess: true

iam:
  withOIDC: true

managedNodeGroups:
  - name: v6-nodegroup
    instanceType: c5.large
```

Must use Nitro-based

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

eksctl – Example (2) – The Basics

```
# kubectl get nodes -o wide
<OUTPUT SUMMARIZED>
NAME                                          STATUS   INTERNAL-IP
ip-192-168-6-94.us-east-2.compute.internal    Ready    2600:1f16:37b:9600:f6bc:f20d:e98:74c9
ip-192-168-85-214.us-east-2.compute.internal  Ready    2600:1f16:37b:9602:c665:fe04:88c3:60a

# kubectl get po -A -o wide
<OUTPUT SUMMARIZED>
NAMESPACE     NAME                       READY   IP
kube-system   aws-node-fwkv4             1/1     2600:1f16:37b:9602:c665:fe04:88c3:60a
kube-system   aws-node-rx2z2             1/1     2600:1f16:37b:9600:f6bc:f20d:e98:74c9
kube-system   coredns-5db97b446d-fgps2   1/1     2600:1f16:37b:9600:19ad::
kube-system   coredns-5db97b446d-x4fbt   1/1     2600:1f16:37b:9600:19ad::1
kube-system   kube-proxy-pkr6j           1/1     2600:1f16:37b:9602:c665:fe04:88c3:60a
kube-system   kube-proxy-tt7b8           1/1     2600:1f16:37b:9600:f6bc:f20d:e98:74c9
```

More on this later, but with Kubernetes, IPv6-only will show a literal IPv6 address, with dual-stack you likely only see IPv4 in the “get” output unless you dig. You can work around this using the “--output” flag.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

eksctl – Example (3)

```
# curl -o iam_policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-
controller/v2.4.3/docs/install/iam_policy.json
# aws iam create-policy \
    --policy-name AWSLoadBalancerControllerIAMPolicy \
    --policy-document file://iam_policy.json
# eksctl create iamserviceaccount \
  --cluster=v6-eksctl \
  --region=us-east-2 \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name "AmazonEKSLoadBalancerControllerRole" \
  --attach-policy-arn=arn:aws:iam::<account_number>:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

```
# kubectl apply \
    --validate=false \
    -f https://github.com/jetstack/cert-manager/releases/download/v1.5.4/cert-manager.yaml
# curl -Lo v2_4_3_full.yaml https://github.com/kubernetes-sigs/aws-load-balancer-
controller/releases/download/v2.4.3/v2_4_3_full.yaml
# sed -i.bak -e '480,488d' ./v2_4_3_full.yaml
# sed -i.bak -e 's|your-cluster-name|v6-eksctl|' ./v2_4_3_full.yaml
# kubectl apply -f v2_4_3_full.yaml
# curl -Lo v2_4_3_ingclass.yaml https://github.com/kubernetes-sigs/aws-load-balancer-
controller/releases/download/v2.4.3/v2_4_3_ingclass.yaml
# kubectl apply -f v2_4_3_ingclass.yaml
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

eksctl – Example (4) – Deploy a Sample Service

https://docs.aws.amazon.com/eks/latest/userguide/network-load-balancing.html

Understanding these annotations is critical!
https://kubernetes-sigs.github.io/aws-load-balancer-controller/v2.2/guide/service/annotations/

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
      service.beta.kubernetes.io/aws-load-balancer-type: "external"
      service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "ip"
      service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
      service.beta.kubernetes.io/aws-load-balancer-ip-address-type: dualstack
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

eksctl – Example (5) – Test it out

```
# kubectl get svc/nginx
NAME   TYPE         CLUSTER-IP           EXTERNAL-IP                                                                  PORT(S)
nginx  LoadBalancer fdc7:550e:3df2::5140 k8s-default-nginx-33153e8b51-548d2285fee81414.elb.us-east-2.amazonaws.com   80:30343/TCP
```

```
# curl -6 k8s-default-nginx-33153e8b51-548d2285fee81414.elb.us-east-2.amazonaws.com
. . .
<h1>Welcome to nginx!</h1>
```

Scheme	internet-facing
IP address type	dualstack **Edit IP address type**
VPC	vpc-0f91b5012dafd6494
Availability Zones	subnet-0f7fb482d9146c32a - us-east-2a IPv4 address: Assigned by AWS IPv6 address: Assigned from CIDR 2600:1f16:37b:9602::/64

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Reference

EKS Cluster Creation – UI-created

Choose cluster IP address family Info

Specify the IP address type for pods and services in your cluster.

- ○ IPv4
- ◉ IPv6

Select the IP address type that pods and services in your cluster will receive.

- Amazon EKS does not support dual stack clusters. However, if your worker nodes contain an IPv4 address, EKS will configure IPv6 pod routing so that pods can communicate with cluster external IPv4 endpoints.

Configure Kubernetes service IP address range Info

Specify the range from which cluster services will receive IP addresses.

Networking

VPC Info

vpc-0e4de82701b8f8f2f

Cluster IP address family Info

IPv6

Service IPv6 range Info

fd7a:c1cb:c06a::/108

Auto-assign IP settings Info

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

- [x] Enable auto-assign IPv6 address Info
- [ ] Enable auto-assign public IPv4 address Info
- [ ] Enable auto-assign customer-owned IPv4 address Info
  Option disabled because no customer owned pools found.

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Other AWS Stuff

Other AWS Services

- S3: https://docs.aws.amazon.com/AmazonS3/latest/userguide/ipv6-access.html

```
# curl -v  http://s3.dualstack.us-west-2.amazonaws.com/
*   Trying 2600:1fa0:40c0:9408:34da:fac0:::80...
* Connected to s3.dualstack.us-west-2.amazonaws.com (2600:1fa0:40c0:9408:34da:fac0::) port 80 (#0)
```

- ECR

```
# docker run -it public.ecr.aws/<repo>/test_image:latest
Unable to find image 'public.ecr.aws/<repo>/test_image:latest' locally
latest: Pulling from <repo>/test_image

### TCPDUMP
IP6 2600:9000:24f4:e600:1d:242c:6e40:21.443 > 2600:1f18:d:f70a:8dd0:f1c7:72c:6571.56626: tcp 32812
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Microsoft Azure

Reference

Microsoft Azure

- Azure IPv6 Regions: https://azure.microsoft.com/en-us/updates/ipv6-for-azure-vms/
- Known support for IPv6:

READ THIS!

  - Azure VNET: https://learn.microsoft.com/en-us/azure/virtual-network/ip-services/ipv6-overview
  - Azure Load Balancer:
    - https://learn.microsoft.com/en-us/azure/load-balancer/virtual-network-ipv4-ipv6-dual-stack-standard-load-balancer-cli

READ THIS!

    - https://github.com/MicrosoftDocs/azure-docs/blob/main/articles/load-balancer/load-balancer-ipv6-overview.md
  - Azure Front Door: https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview
  - Azure Express Route: https://learn.microsoft.com/en-us/azure/expressroute/expressroute-howto-add-ipv6-portal
  - Azure DNS: https://learn.microsoft.com/en-us/azure/dns/dns-faq
  - Azure Kubernetes Service: https://learn.microsoft.com/en-us/azure/aks/configure-kubenet-dual-stack?tabs=azure-cli%2Ckubectl

New!

  - Azure AD – IPv6 support is here: https://learn.microsoft.com/en-us/troubleshoot/azure/active-directory/azure-ad-ipv6-support and https://learn.microsoft.com/en-us/troubleshoot/azure/active-directory/azure-ad-ipv6-support

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Azure VNET

- Dual stack – IPv6-only is not supported for VMs
- Self-defined prefixes (non-routable)
  - /64 subnet only
  - IPv6 prefix guide: https://learn.microsoft.com/en-us/azure/virtual-network/ip-services/public-ip-address-prefix
  - Public IPv6 is used only for the Azure Load Balancer
- Network Security Group (NSG) support
- VNET peering support
- VPN Gateways – IPv4 only

VNET Subnet

Name

default

Subnet address range *

10.0.0.0/24

10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

☑ Add IPv6 address space

IPv6 address space

fd00:bad:beef:cafe::/64

VM ipconfig (IPv4 and IPv6)

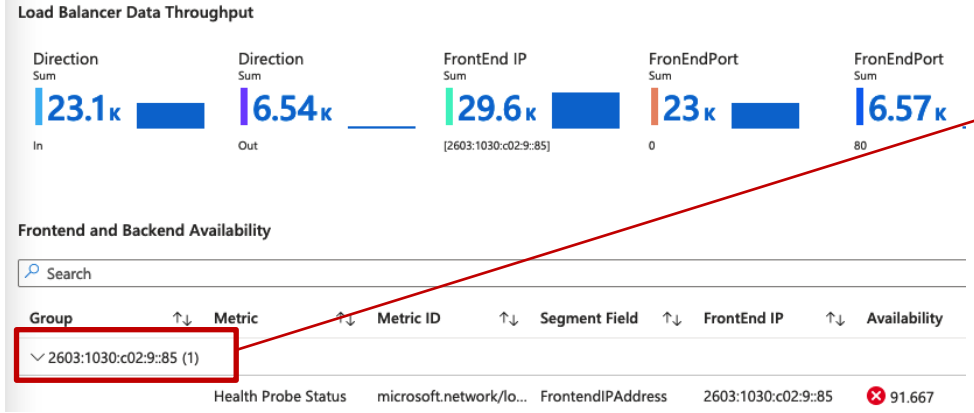
Name	IP Version	Type	Private IP address	Public IP address
ipconfig1	IPv4	Primary	10.0.0.5 (Dynamic)	20.236.32.71 (ds-vm-2-ip)
ipconfig2	IPv6	Secondary	fd00:bad:beef:cafe::5 (Dynamic)	-

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Azure Load Balancer

- https://github.com/MicrosoftDocs/azure-docs/blob/main/articles/load-balancer/load-balancer-ipv6-overview.md
- Public and internal load balancer types
- IPv6 Frontend and Backend support
- No NAT64
- **Used** for **inbound** and **outbound** IPv6 (external to Azure)

```
azureuser@ds-vm-2:~$ curl -6 -o /dev/null -s -w "%{http_code}\n" http://\[2603:1030:c02:9::85\]/
200
```



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Azure LB – Wizard(1)

1

Create public IP address

IP Version * ⓘ
○ IPv4 ◉ IPv6 ○ Both

SKU * ⓘ
◉ Standard ○ Basic

Tier
◉ Regional ○ Global

IPv6 IP Address Configuration

Name *
v6-ext

IP address assignment
○ Dynamic ◉ Static

Idle timeout (minutes) ⓘ

DNS name label ⓘ

Subscription *
Visual Studio Enterprise

Resource group *
v6
Create new

Location *
West US 2

Availability zone * ⓘ
Zone-redundant

2

Subscription *
Visual Studio Enterprise

Resource group *
v6
Create new

Instance details

Name *
ds-lb-1

Region *
West US 2

SKU * ⓘ
◉ Standard
○ Gateway
○ Basic

Microsoft recommends Standard SKU load balancer for production workloads.
Learn more about pricing differences between Standard and Basic SKU

Type * ⓘ
◉ Public
○ Internal

Tier *
◉ Regional
○ Global

3

Add frontend IP configuration

Name *
ds-fe-1

IP version
○ IPv4 ◉ IPv6

IP type
◉ IP address ○ IP prefix

Public IP address *
v6-ext (2603:1030:c02:9::85)
Create new

Gateway Load balancer ⓘ
None

4

Add backend pool

Name *
ds-be-1

Virtual network ⓘ
ds-vnet (v6)

Backend Pool Configuration
○ NIC
◉ IP address

IP addresses

You can only add resources IP address in the Virtual Network. The configuration is associated with the IP address and will apply to any resource which has this IP address assigned.

IP address	Resource Name
fd00:bad:beef:cafe::4	ds-vm-1 (v6)
fd00:bad:beef:cafe::5	ds-vm-2 (v6)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Azure LB – Wizard(2)

5

Name *
ds-lb-rule-http

IP Version *
- IPv4
- IPv6

Frontend IP address *
ds-fe-1 (2603:1030:c02:9::85)

Backend pool *
ds-be-1

Protocol *
- TCP
- UDP

Port *
80

Backend port *
80

Health probe *
(new) ds-probe
Create new

6

Name *
ds-out-rule

IP Version *
- IPv4
- IPv6

Frontend IP address *
1 selected

Protocol
- All
- TCP
- UDP

Idle timeout (minutes)
4
Max: 100

TCP Reset
- Enabled
- Disabled

Backend pool *
ds-be-1 (2 instances)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Azure Kubernetes Service (AKS)

- NEW! https://learn.microsoft.com/en-us/azure/aks/configure-kubenet-dual-stack?tabs=azure-cli%2Ckubectl
  - Great primer on Networking with AKS: https://learn.microsoft.com/en-us/azure/aks/concepts-network
- IPv4 or Dual stack only (no IPv6 only)
- Nodes, pods and services get ULA addresses

```
# az aks create -l westus -g dual-rg -n aks-dual --ip-families ipv4,ipv6
```

```
"podCidrs": [
   "10.244.0.0/16",
   "fd27:bf16:4508:3013::/64"
],
"serviceCidr": "10.0.0.0/16",
"serviceCidrs": [
   "10.0.0.0/16",
   "fd82:eeaf:949e:b8a3::/108"
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

AKS – Service/Deployment – Example

Service

```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  externalTrafficPolicy: Local
  ipFamilyPolicy: PreferDualStack
  ipFamilies:
  - IPv6
  - IPv4
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

Deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: nginx
```

Rat hole alert! ipFamiles order determines which family is set for the ClusterIP.

https://kubernetes.io/docs/concepts/services-networking/dual-stack/

```
# kubectl get pods -o custom-columns="NAME:.metadata.name,IPs:.status.podIPs[*].ip,NODE:.spec.nodeName,READY:.status.conditions[?(@.type=='Ready')].status"
NAME                    IPs                                    NODE                                READY
nginx-7597c656c9-fpj27  10.244.1.3,fd27:bf16:4508:3013:1::3    aks-nodepool1-34985876-vmss000000   True
```

```
# kubectl get services
NAME         TYPE           CLUSTER-IP                  EXTERNAL-IP                     PORT(S)        AGE
nginx.       LoadBalancer   fd82:eeaf:949e:b8a3::7977   2a01:111:f100:3000::a83e:1a6b   80:30508/TCP   5m36s
```

```
# # curl -6 -o /dev/null -s -w "%{http_code}\n" http://\[2a01:111:f100:3000::a83e:1a6b\]/
200
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Google Cloud Platform

Google Cloud Platform (GCP)

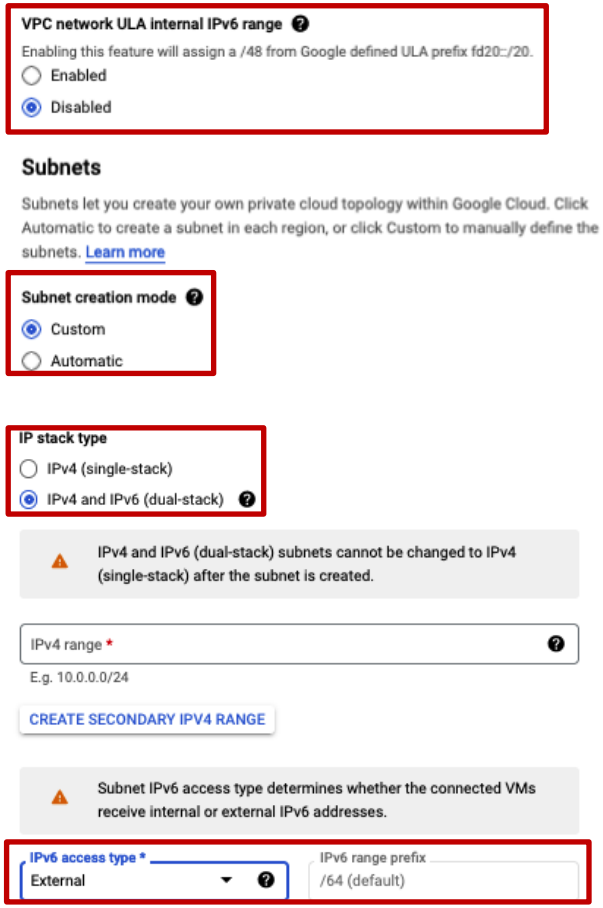
- Google services have had IPv6 for a long time (Gmail, etc.)
- GCP is late to the game and has very spotty IPv6 support: https://cloud.google.com/blog/products/networking/getting-started-with-ipv6-on-google-cloud
- VPCs:
  - https://cloud.google.com/vpc/docs/subnets#ipv6-ranges
  - IPv6 subnets supported in custom mode – IPv4 OR dual stack, **no IPv6-only**
  - **/48 per VPC network, /64 prefix per subnet (odd /65 routes, but it doesn't seem to break anything [not using SLAAC]), /96 prefix per VM, /128 per interface**
  - Internal (ULA range) , external (GCP provided GUA)
  - Full IPv6 firewall support
  - Cannot change IPv6 access type (internal/external) after it is chosen
  - Cannot change VPC network /48
  - **No Inter-VPC IPv6 routing**
- Cloud Load Balancer: https://cloud.google.com/load-balancing/docs/ipv6

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Reference

VPC

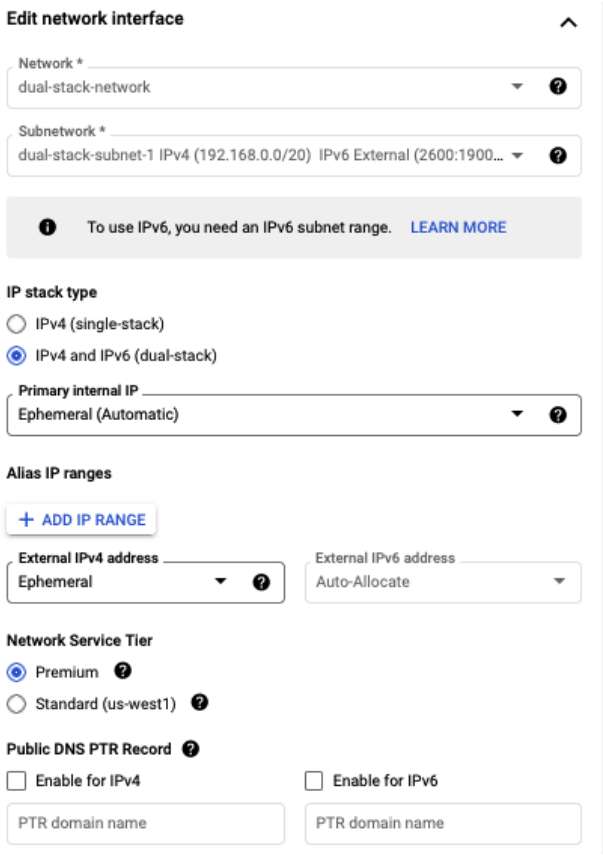
- ULA (internal) or GUA (External)
- Must use “Custom” mode
- Dual stack (no IPv6-only)
- Access type: Internal/External



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Compute Engine

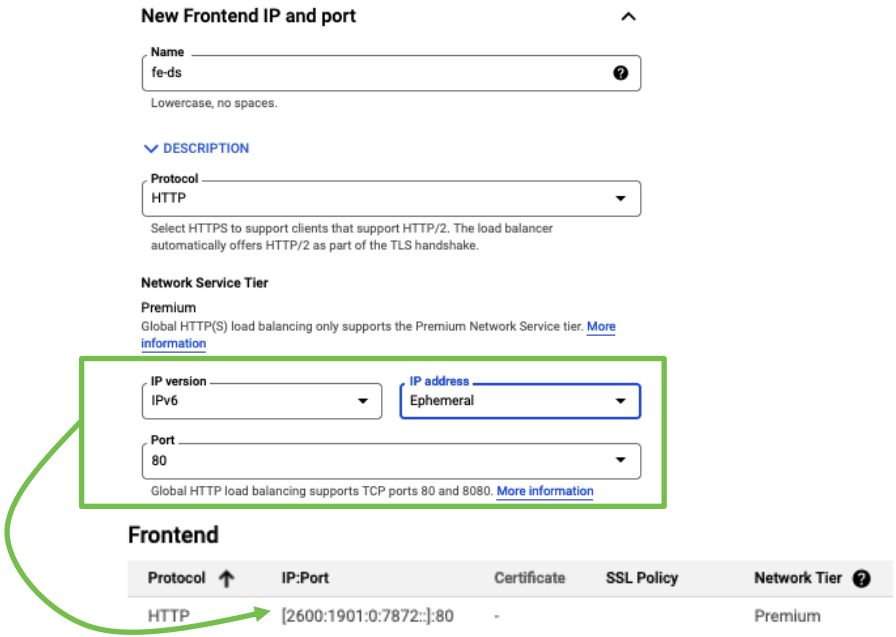
- Network + Subnetwork
- IP stack type: IPv4 or Dual stack (no IPv6-only)
- Auto-Allocate only (DHCPv6)
- Stuff works as you would expect it:
  - Add firewall rules and go
  - SSH over IPv6

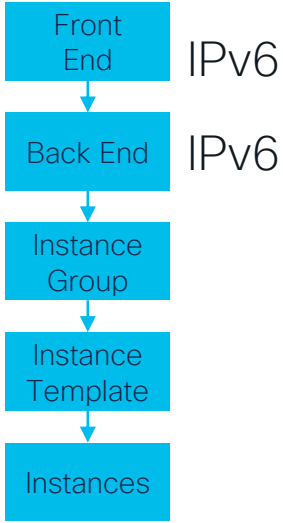


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

NEW: Cloud Load Balancing

- https://cloud.google.com/load-balancing/docs/network/networklb-backend-service
- https://cloud.google.com/blog/products/networking/getting-started-with-ipv6-on-google-cloud





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

GKE (Google Kubernetes Engine)

- https://cloud.google.com/kubernetes-engine/docs/concepts/alias-ips#dual_stack_network
- No IPv6-only support
- IPv6 is **not** supported for LoadBalancer Services – Can expose a service via ClusterIP or NodePort (not awesome): https://cloud.google.com/kubernetes-engine/docs/concepts/alias-ips#dual_stack_network and https://cloud.google.com/kubernetes-engine/docs/concepts/alias-ips
  - ”Dual-stack Services are supported for ClusterIP and NodePort Services, but not for LoadBalancer Services.”
- Public clusters – 2600:1900::/28
- Private clusters – fd20::/20
- /64 for each subnet, /96 for the primary node range, /112 for the Pod range, /112 for the service range

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

GKE – Cluster Creation – CLI

- https://cloud.google.com/kubernetes-engine/docs/how-to/alias-ips#dual-stack
- There are key flags you must set in your gcloud command:

Note: this is just a summary of IPv6-centric flags:

```
# gcloud container clusters create <cluster_name> \
--enable-ip-alias \
--network "<your-dual-stack-network>" \
--subnetwork "<your-dual-stack-subnetwork>" \
--enable-dataplane-v2 \
--stack-type=ipv4-ipv6
## If creating the network/subnet at the same time as the cluster, add
##--ipv6-access-type=external"
```

Status	Name	Creation Time	Template	Per instance config	Internal IP	External IP
✓	gke-cluster-1-default-pool-36570d7d-5t34	Apr 21, 2023, 1:21:03 PM UTC-06:00	gke-cluster-1-default-pool-36570d7d		10.0.0.12 (nic0)	34.102.65.40 2600:1900:4120:cd23:0:9:0:0
✓	gke-cluster-1-default-pool-36570d7d-dv3k	Apr 21, 2023, 1:21:03 PM UTC-06:00	gke-cluster-1-default-pool-36570d7d		10.0.0.11 (nic0)	34.94.110.234 2600:1900:4120:cd23:0:8:0:0
✓	gke-cluster-1-default-pool-36570d7d-j4mn	Apr 21, 2023, 1:21:03 PM UTC-06:00	gke-cluster-1-default-pool-36570d7d		10.0.0.10 (nic0)	34.94.141.156 2600:1900:4120:cd23:0:7:0:0

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

GKE – Example – Deploy a Sample Service

Reference

- NodePort and ClusterIP examples

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  ipFamilyPolicy: PreferDualStack
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx
  type: NodePort
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  ipFamilyPolicy: PreferDualStack
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx
  type: ClusterIP
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Reference

GKE – Services (NodePort or ClusterIP)

```
# kubectl describe svc/nginx
Name:                     nginx
Namespace:                default
Labels:                   <none>
Annotations:              cloud.google.com/neg: {"ingress":true}
Selector:                 app=nginx
Type:                     NodePort
IP Family Policy:         PreferDualStack
IP Families:              IPv4,IPv6
IP:                       10.64.4.118
IPs:                      10.64.4.118,2600:2d00:0:4:3855:52a6:a0f8:f7dc
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30929/TCP
Endpoints:                10.60.0.4:80,10.60.1.4:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

```
# kubectl describe svc/nginx
Name:                     nginx
Namespace:                default
Labels:                   <none>
Annotations:              cloud.google.com/neg: {"ingress":true}
Selector:                 app=nginx
Type:                     ClusterIP
IP Family Policy:         PreferDualStack
IP Families:              IPv4,IPv6
IP:                       10.64.11.184
IPs:                      10.64.11.184,2600:2d00:0:4:3855:52a6:a0f8:3b2a
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
Endpoints:                10.60.0.5:80,10.60.1.5:80
Session Affinity:         None
Events:                   <none>
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Reference

GKE – Services (LoadBalancer) – Using ”Type: LoadBalancer” **doesn’t** work

```
# kubectl describe svc/nginx
Name:                     nginx
Namespace:                default
Labels:                   <none>
Annotations:              cloud.google.com/neg: {"ingress":true}
Selector:                 app=nginx
Type:                     LoadBalancer
IP Family Policy:         PreferDualStack
IP Families:              IPv4,IPv6
IP:                       10.64.13.120
IPs:                      10.64.13.120,2600:2d00:0:4:3855:52a6:a0f8:6092
LoadBalancer Ingress:     34.168.253.165
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31873/TCP
Endpoints:                10.60.1.4:80,10.60.2.4:80
Session Affinity:         None
External Traffic Policy:  Cluster
```

No external IPv6 on the ingress

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

GKE – Pods and Nodes

```
# kubectl describe po nginx-deployment-544dc8b7c4-49lck
Name:         nginx-deployment-544dc8b7c4-49lck
Namespace:    default
Priority:     0
Node:         gke-gke-ds-2-default-pool-43a47a45-qtdh/192.168.0.12
Start Time:   Thu, 15 Sep 2022 12:32:11 -0600
Labels:       app=nginx
              pod-template-hash=544dc8b7c4
Annotations:  <none>
Status:       Running
IP:           10.60.1.4
IPs:
  IP:           10.60.1.4
  IP:           2600:1900:4040:2cb2:0:5:0:4
```

```
# kubectl describe node gke-gke-ds-2-default-pool-43a47a45-qtdh
Name:               gke-gke-ds-2-default-pool-43a47a45-qtdh
Addresses:
  InternalIP:  192.168.0.12
  InternalIP:  2600:1900:4040:2cb2:0:5:0:0
  ExternalIP:  34.168.253.165
  Hostname:    gke-gke-ds-2-default-pool-43a47a45-qtdh
PodCIDR:                      10.60.1.0/24
PodCIDRs:                     10.60.1.0/24,2600:1900:4040:2cb2:0:5:0:0/112
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Other GCP Services

- In theory, Anthos is supposed to support dual stack (still no native IPv6 LB support), but I didn't bother testing it: https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/how-to/dual-stack-networking
- The support for IPv6 at GCP is so lacking that I gave up testing support for things
- I 'think' GCR works – it resolves:

```
# ping6 gcr.io
PING6(56=40+8+8 bytes) 2601:282:4103:41d0:a0b3:cf87:1c45:1520 --> 2607:f8b0:4023:1009::52
16 bytes from 2607:f8b0:4023:1009::52, icmp_seq=0 hlim=101 time=29.643 ms
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Docker

Docker IPv6 Support

- Docker has per-OS dual stack support
- IPv6-only support is only available on Linux hosts
- https://docs.docker.com/config/daemon/ipv6/

/etc/docker/daemon.json or Preferences > Docker Engine (Docker for Desktop)

```
{
 "ipv6": true,
 "fixed-cidr-v6": "2001:db8:cafe:1::/64"
}
```

If fixed-cidr is local only to the Docker host, you must add a route on the host (and rest of network) to that prefix:

```
ip -6 route add 2001:db8:cafe:1::/64 via dev docker0
```

- Be aware of old (not properly fixed) and new bugs – when you restart Docker after enabling IPv6, any RA learned default gateways will be removed
  - https://github.com/mailcow/mailcow-dockerized-docs/issues/354
  - https://github.com/docker/for-linux/issues/1373
  - Enable again:

    `sysctl net.ipv6.conf.eth0.accept_ra=2`

Broken

```
Destination Next Hop Flag Met Ref Use If
[::]/0      [::]     !n   -1  1     0 lo
```

```
Destination Next Hop  Flag  Met  Ref   Use If
[::]/0      gateway   UGDAe 1024 1     0   eth0
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Docker Registry

- You can start Docker Registry from a manual ‘docker run’:

```
# docker run -d -p 5000:5000 --restart=always -v /reg:/var/lib/registry --name registry registry:2
```

- If using an insecure registry, on the ‘client’ (host that is connecting to the registry), manually identify the registry:

```
{
  "ipv6": true,
  "fixed-cidr-v6": "2600:1f18:d:f700:dd3d::/80",
  "insecure-registries": ["docker-registry.example.com:5000"]
}
```

Must be a name, not literal address

- Do the usual tag/push/pulls:

```
# docker tag alpine docker-registry.example.com:5000/alpine
# docker push docker-registry.example.com:5000/alpine:latest
```

```
$ sudo tcpdump -i eth0 port 5000 -q -nn
20:37:34.338851 IP6 2600:1f18:d:f700:6928:7603:2e2f:37ba.57870 > 2600:1f18:d:f700:6b5c:242:ac11:2.5000: tcp 0
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Docker Hub

- As of January 24th, 2022 Docker Hub support is out of beta
- Use **registry.ipv6.docker.com**:

```
# docker pull registry.ipv6.docker.com/library/ubuntu:latest
latest: Pulling from library/ubuntu
d19f32bd9e41: Pull complete
Digest: sha256:34fea4f31bf187bc915536831fd0afc9d214755bf700b5cdb1336c82516d154e
Status: Downloaded newer image for registry.ipv6.docker.com/library/ubuntu:latest
registry.ipv6.docker.com/library/ubuntu:latest
```

```
# dig +short registry.ipv6.docker.com aaaa
2600:1f18:2148:bc01:6d1f:799d:692e:d1cb
2600:1f18:2148:bc02:cca:f7fc:9da:da79
2600:1f18:2148:bc00:553a:c94e:dfb4:4101
```

```
IP6 2600:1f18:d:f700::a7a5.43886 > 2600:1f18:2148:bc00:553a:c94e:dfb4:4101.443: tcp 0
```

- Make it permanent:

```
{
  "registry-mirrors": [ https://registry.ipv6.docker.com ]
}
```

```
# docker pull ubuntu:latest
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Kubernetes

Kubernetes – State of the Union

- In a land far, far away and at a time long, long ago, IPv6-only support development began:
  - IPv6-only (2017) – https://github.com/kubernetes/enhancements/issues/508
  - Dual stack (2018) – https://github.com/kubernetes/enhancements/issues/563
    - Reaches GA (2021) in Kubernetes: 1.23: https://kubernetes.io/blog/2021/12/08/dual-stack-networking-ga/
  - Dual stack is enabled by default starting in 1.21 (before GA)
- KinD: https://kind.sigs.k8s.io/docs/user/configuration/
- Minikube: No IPv6 support: https://github.com/kubernetes/minikube/issues/8535

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

KinD

- https://kind.sigs.k8s.io/docs/user/configuration/#networking
  - IPv4-only
  - IPv6-only
  - Dual stack

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
networking:
  ipFamily: dual
```

  - kubectl output will show IPv4 addresses – use ‘describe’ or filters to see IPv6

```
# kubectl get svc -A -o wide
NAMESPACE     NAME         TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)
default       kubernetes   ClusterIP   10.96.0.1     <none>        443/TCP
kube-system   kube-dns     ClusterIP   10.96.0.10    <none>        53/UDP,53/TCP,9153/TCP
```

ipFamily: dual

```
# kubectl get svc -A -o wide
NAMESPACE     NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
default       kubernetes   ClusterIP   fd00:10:96::1   <none>        443/TCP
kube-system   kube-dns     ClusterIP   fd00:10:96::a   <none>        53/UDP,53/TCP,9153/TCP
```

ipFamily: ipv6

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

MetalLB with 0.37.0

```
# kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.13.7/config/manifests/metallb-native.yaml
```

```
# docker network inspect -f '{{.IPAM.Config}}' kind
[{172.18.0.0/16  172.18.0.1 map[]} {fc00:f853:ccd:e793::/64  fc00:f853:ccd:e793::1 map[]}]
```

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: example
  namespace: metallb-system
spec:
  addresses:
  - 172.18.255.1-172.18.255.250
  - fc00:f853:ccd:e793::5-fc00:f853:ccd:e793::a
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: empty
  namespace: metallb-system
```

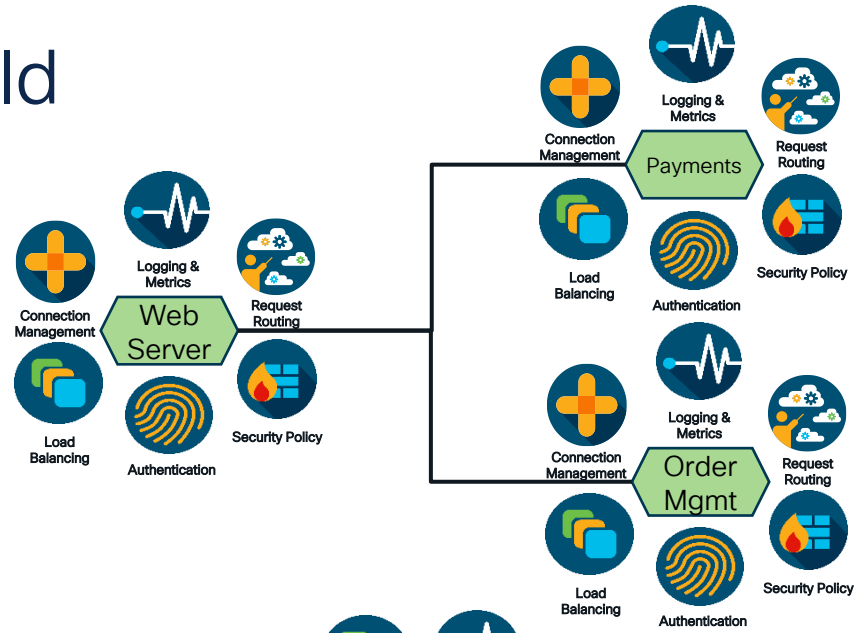
```
# kubectl get svc -A
NAMESPACE     NAME         TYPE           CLUSTER-IP         EXTERNAL-IP                          PORT(S)
default       nginx        LoadBalancer   fd00:10:96::d631   172.18.255.1,fc00:f853:ccd:e793::5   80:31899/TCP
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

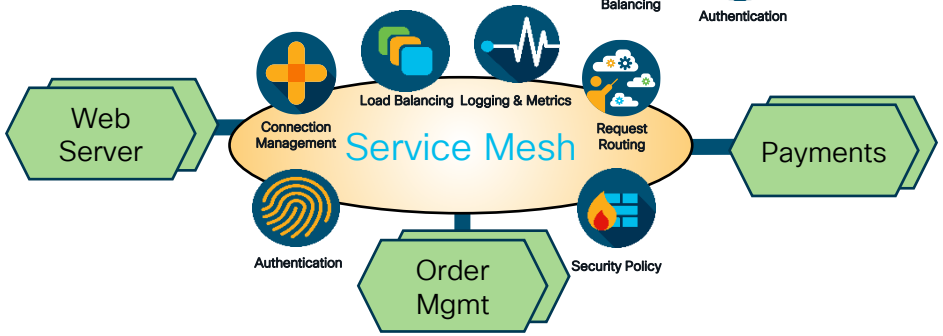
Service Mesh

Service Features in a Microservice World

- Deploy all the service features as independent components

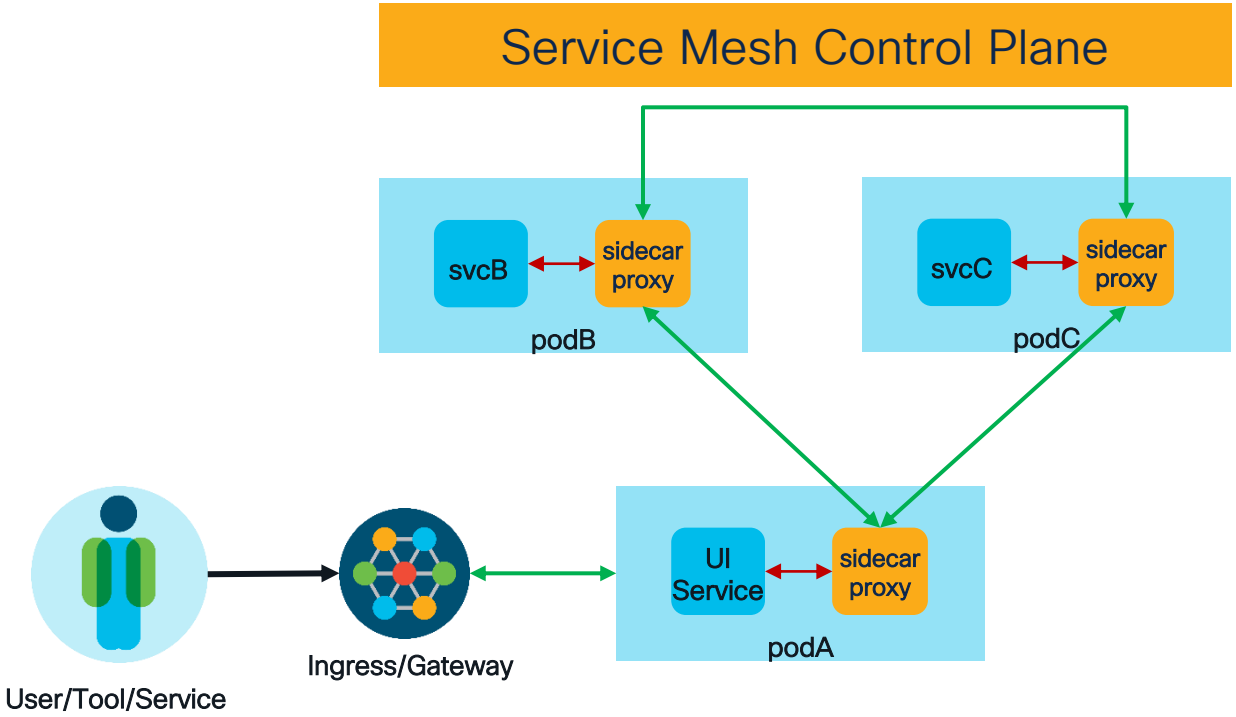


- Offload features for service-to-service communication to a policy-driven secure service mesh



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

What is a Service Mesh?



- Infrastructure layer for service-to-service communication
- Can use a mesh of sidecar proxies:
  - Can inspect API transactions at Layer 7 and 4 (TCP)
  - Intelligent routing rules can be applied between endpoints

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

IPv6 Support for Application Service Meshes

- Linkerd – https://linkerd.io/
  - No support for IPv6: https://github.com/linkerd/linkerd2/issues/3849
  - Roadmap: https://github.com/linkerd/linkerd2/milestone/23
- Istio – https://istio.io/
  - https://istio.io/latest/blog/2023/experimental-dual-stack/
  - An almost infinite number of gotchas depending on the Istio design you are using (multicluster, gateways, etc)
  - Warning: There are details such as Virtual Services, Gateways and other stuff that are specific to your application design that need to be worked through, but they don't have IPv6-specific settings per-se

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Istio with Dual Stack - A Basic Setup

- KinD
- MetalLB
- Istio with Istioctl – Must use 1.17+

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

KinD

- Create a KinD config file (example: dual.yaml):

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
networking:
  ipFamily: dual
nodes:
- role: control-plane
- role: worker
- role: worker
```

```
# kind create cluster --config dual.yaml --name dual-cls-1
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

MetalLB – 0.37.0

```
# kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.13.7/config/manifests/metallb-native.yaml
```

```
# docker network inspect -f '{{.IPAM.Config}}' kind
[{172.18.0.0/16  172.18.0.1 map[]} {fc00:f853:ccd:e793::/64   map[]}]
```

```
kubectl apply -f - <<EOF
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: example
  namespace: metallb-system
spec:
  addresses:
  - 172.18.255.1-172.18.255.250
  - fc00:f853:ccd:e793::5-fc00:f853:ccd:e793::a
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: empty
  namespace: metallb-system
EOF
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Install Istio (1)

```
# curl -L https://istio.io/downloadIstio | sh -
# export PATH="$PATH:/home/ec2-user/istio-1.17.2/bin”
# istioctl x precheck
✔ No issues found when checking the cluster. Istio is safe to install or upgrade!
```

Quick start install with Dual Stack enabled:

```
istioctl install -f - <<EOF
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
spec:
  meshConfig:
    defaultConfig:
      proxyMetadata:
        ISTIO_AGENT_DUAL_STACK: "true"
  values:
    pilot:
      env:
        ISTIO_DUAL_STACK: "true"
EOF
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Label a Namespace for Istio sidecar-injection and Deploy a Sample Service

Step 1:

```
# kubectl label --overwrite namespace default istio-injection=enabled
```

Step 2:

```
kubectl apply -f - <<EOF
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
  ipFamilyPolicy: PreferDualStack
  ipFamilies:
  - IPv6
  - IPv4
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
EOF
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Reference

Everyone loves a bunch of output ☺

Service:

```
# kubectl get svc -A
NAMESPACE        NAME                    TYPE           CLUSTER-IP         EXTERNAL-IP                PORT(S)
NAME          TYPE           CLUSTER-IP         EXTERNAL-IP                          PORT(S)        AGE    SELECTOR
kubernetes    ClusterIP      10.96.0.1          <none>                               443/TCP        104m   <none>
nginx         LoadBalancer   fd00:10:96::6a2b   172.18.255.1,fc00:f853:ccd:e793::5   80:30841/TCP   31s    app=nginx
```

Service:

```
# kubectl describe svc/nginx
Name:                     nginx
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=nginx
Type:                     LoadBalancer
IP Family Policy:         RequireDualStack
IP Families:              IPv6,IPv4
IP:                       fd00:10:96::6a2b
IPs:                      fd00:10:96::6a2b,10.96.196.252
LoadBalancer Ingress:     172.18.255.1, fc00:f853:ccd:e793::5
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30841/TCP
Endpoints:                [fd00:10:244:1::5]:80
Session Affinity:         None
External Traffic Policy:  Cluster
```

Pod:

```
# kubectl describe po nginx-deployment-6b7f675859-dfq2k
. . .output summarized. . .
Name:             nginx-deployment-6b7f675859-dfq2k
Namespace:        default
Priority:         0
Service Account:  default
Node:             dual-cls-1-worker2/172.18.0.4
Status:           Running
IP:               10.244.1.5
IPs:
  IP:           10.244.1.5
  IP:           fd00:10:244:1::5
```


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Reference

What does Envoy see?

View from the Envoy configuration for the nginx pod (fd00:10:244:1::5)

```
# istioctl proxy-config endpoints nginx-deployment-6b7f675859-dfq2k
<OUTPUT_SUMMARIZED>
ENDPOINT                                          STATUS      OUTLIER CHECK     CLUSTER
10.244.1.5:80                                     HEALTHY     OK                outbound|80||nginx.default.svc.cluster.local
fd00:10:244:1::5:80                               HEALTHY     OK                outbound|80||nginx.default.svc.cluster.local
```

Pod

```
# istioctl proxy-config listeners nginx-deployment-6b7f675859-dfq2k
<OUTPUT_SUMMARIZED>
ADDRESS          PORT  MATCH                                                DESTINATION
fd00:10:96::6a2b 80    Trans: raw_buffer; App: http/1.1,h2c                 Route: nginx.default.svc.cluster.local:80
fd00:10:96::6a2b 80    ALL                                                  Cluster: outbound|80||nginx.default.svc.cluster.local
```

Service
(ClusterIP of Load balancer)

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Summary

Summary

- IPv6 with hyperscalers is hit-or-miss. AWS is way, way ahead here, but by no means perfect
- If all you care about (today) is serving IPv6-enabled applications to the public Internet, you can do that across each hyperscaler
- You end up having more IPv6 ‘power’ by running your own cloud services, but this is likely a short-term gain:
  - Kubernetes, OpenStack
  - + all your own networking goodness
  - + designing your own address plan
- **Get started now and don’t be paralyzed by the perceived workload – Start small, start simple and do one app in one part of the cloud – Rinse>repeat**

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Fill out your session surveys!

Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!

Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.

These points help you get on the leaderboard and increase your chances of winning daily and grand prizes

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

Thank you

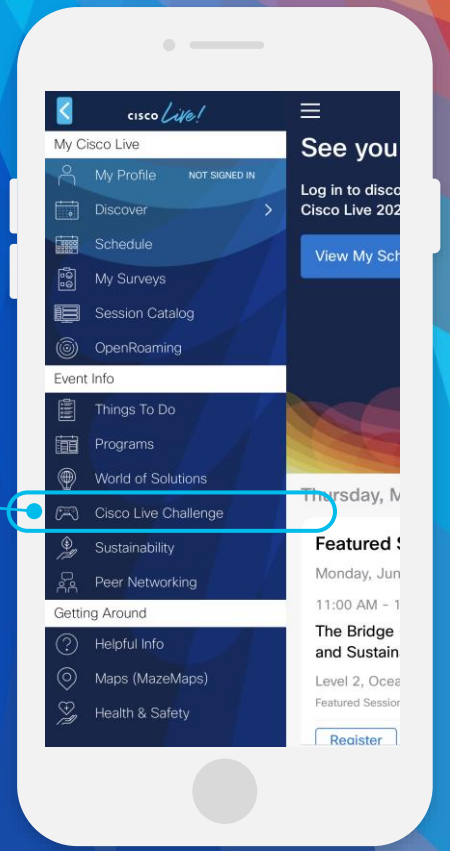
Cisco Live **Challenge**

Gamify your Cisco Live experience!

Get points for attending this session!

How:

1. Open the Cisco Events App.
2. Click on 'Cisco Live Challenge' in the side menu.
3. Click on View Your Badges at the top.
4. Click the + at the bottom of the screen and scan the QR code:



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

cisco Live!

Let's go