KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

Adapting Kubernetes to Constrained IP Address Environments

Satyadeep Musuvathy - SWE, Google
Mahesh Narayanan - Prod. Manager, Google

Agenda

- Context
- Problem Definition
- Overview of Solutions
- Invitation for Community Discussion

Kubernetes

A Networking View- Fundamentals

Kubernetes Networking Model

Every Pod gets its own IP

- All containers within the pod share this IP address
- Pod IPs form a flat space within the cluster
    - every pod can ***directly*** talk to every other pod based on it’s IP address (no proxy needed)

Service IPs are tracked in terms of pod IPs (EndPoints)

- By default, assumes that DNAT to a pod IP is sufficient to send traffic to a pod

Pod IP allocation happens per-node, with blocks of IPs being pre-assigned to each node

- Allows for efficient, distributed allocation, while not having to do a global coordination

Implications of K8s Networking Model

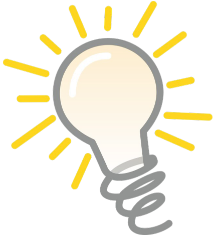
Kubernetes is hungry for IP addresses

- ➢ By default a 100 node cluster needs a /17 CIDRs.
- ➢ Pods are the atom of allocation and workload scale.
    - ○ Among node, service and pods, IPs for pods drives the demand for IP addresses.

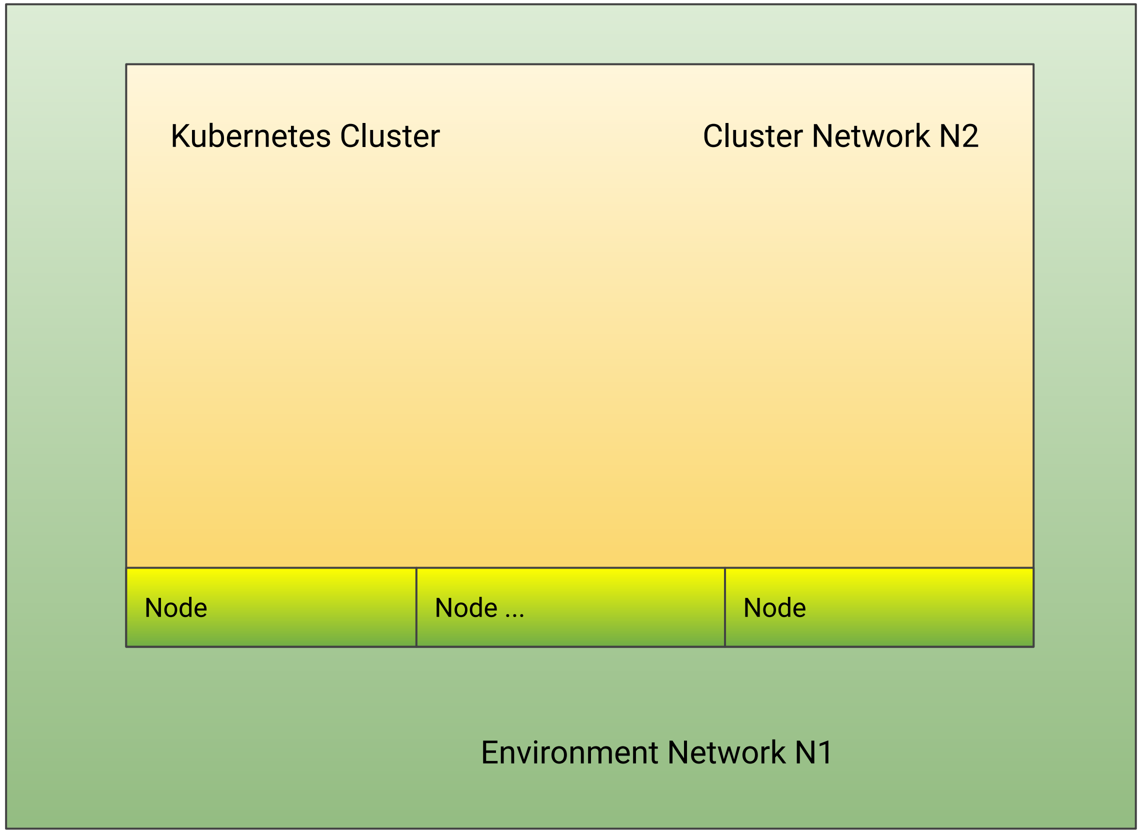
IP’s cannot be re-used too quickly

- ➢ Pod IP allocations happen in a distributed, un-coordinated manner, changes to pod IPs can take some time for it to be reflected across the cluster.
- ➢ For safety, it’s desirous to have a buffer of free IPs at each node minimize IP reuse during allocation.
- ➢ This further adds to the demand for IP addresses within Kubernetes.

Node & Cluster: The Networking Abstractions



Constraints arise in the interactions between N1 and N2



https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/

Two Deployment Modes

Flat Mode: Cluster network shares addresses space with the environment

Benefit

- ➢ Pods become first class citizens in the environment, simplifying connectivity and cross cluster use-cases

Trade off

- ➢ Lack of segmentation and management overhead of routing to pod IPs in the underlying environment.

Island Mode: Cluster network does not share address space with the environment

Benefit

- ➢ Re-use same pod CIDR block across many clusters providing IP efficicency

Trade off

- ➢ All access from outside the cluster is via Service, requiring translation or overlay for inter-cluster connectivity

Constrained IP Address Environments

Supply and Demand Constraints → We just don’t have enough IPs to create clusters

Customer and Deployment perspective

- Growing kubernetes adoption in existing fragmented environments and lack of a large contiguous block.

- Hybrid and multi-cloud adoption and having to share the address space across the various environments

- Organizational challenges between application and infrastructure (on-prem and cloud) teams in being able to coordinate and find large free blocks that works across the organization

- Adoption of newer technologies, like flat service meshes, that can need direct IP address connectivity across endpoints to be able to load balance services, even across clusters and network boundaries

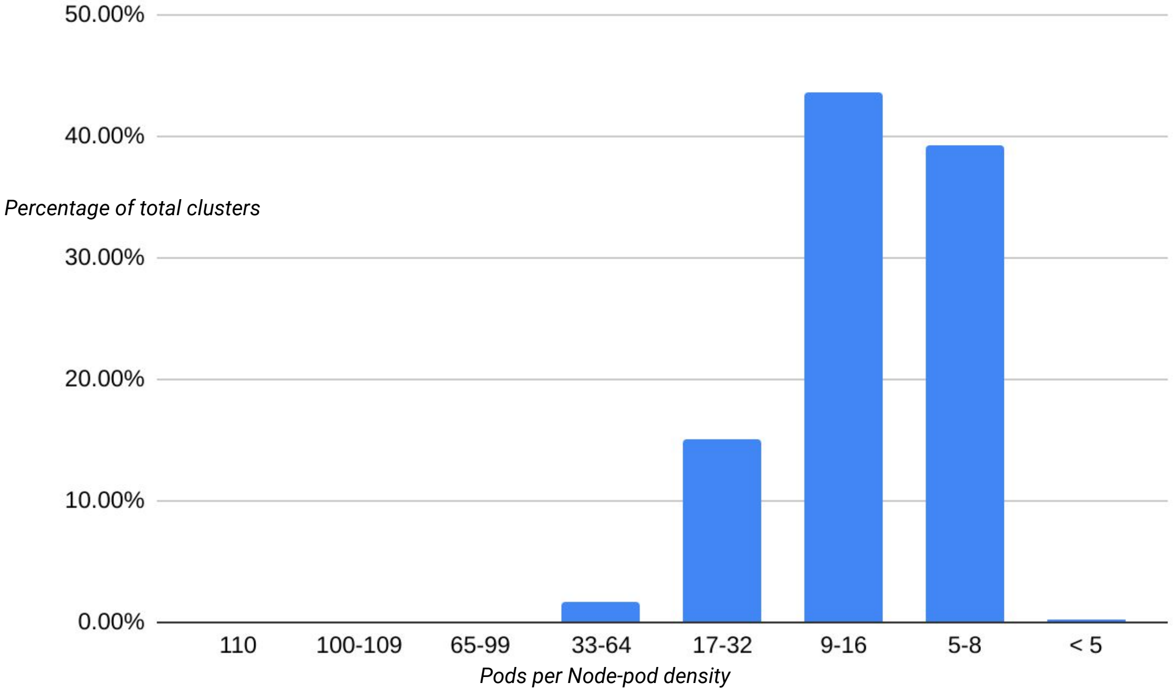
- Applications that want direct pod endpoint connectivity for stickiness without going through a service IP translation

Solution: Optimize IP Utilization

Crux of the Problem: We need to make certain assumptions about Pod Density on a Node beforehand

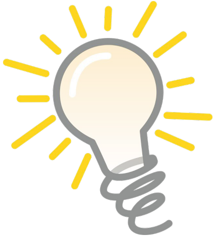
Drivers for low Pod Density

- Resource utilization in Nodes: CPU and Memory consideration (and in some cases bandwidth)
- Deployments in new markets such as Edge compute, where the size of a cluster is small
- From a high availability perspective, users may prefer many small clusters to a few large ones

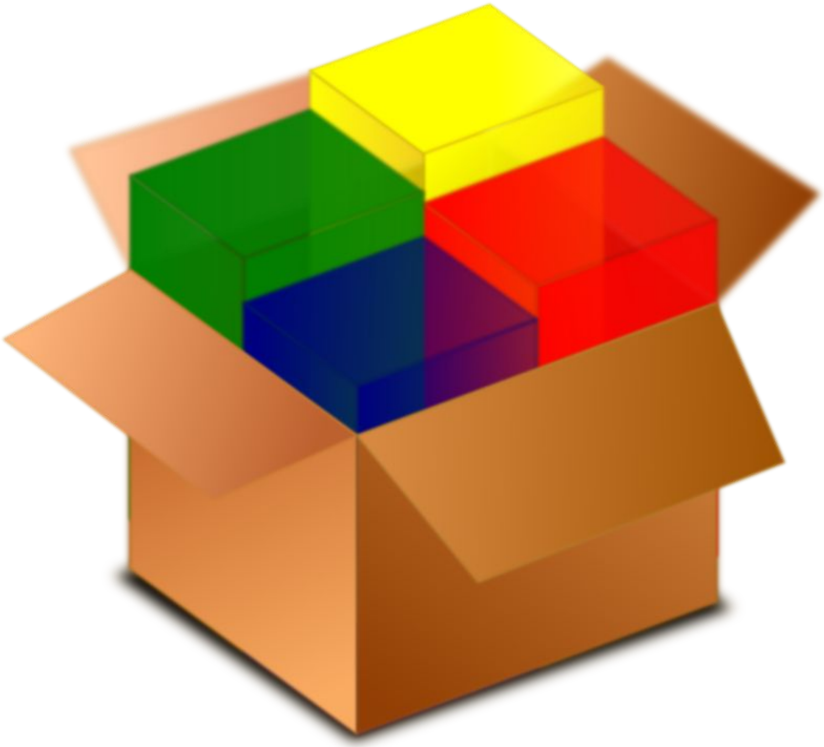


Pod Density	Pod CIDR per Node	Pod CIDR Range Needed	Savings per Node	% saved per Node
65-110	/24	/25	128	50.00%
33-64	/24	/26	192	75.00%
17-32	/24	/27	224	87.50%
9-16	/24	/28	240	93.75%
8	/24	/29	248	96.88%

Solution: Optimize IP Utilization



Max Pods + Buffer <= Node podCIDR



https://pixabay.com/vectors/package-cardboard-box-box-parcel-153360/

https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/

Solution: Dynamic & Discontiguous Pod CIDR

Migration Across Environments

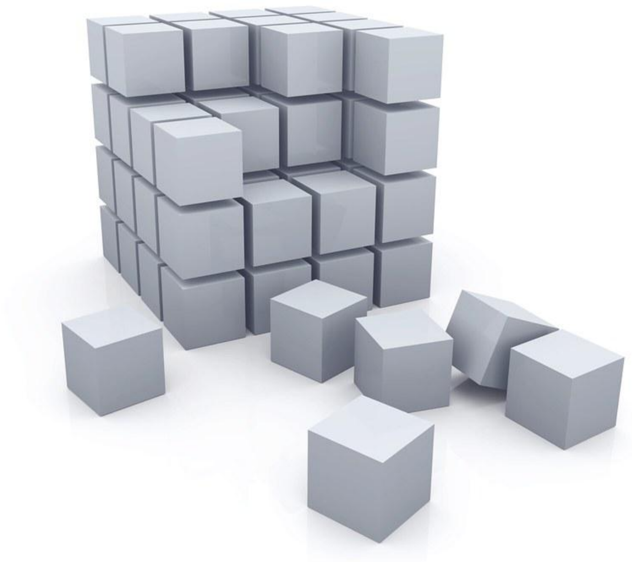
- Customers migrating few workloads at a time to Cloud. As the Cloud side starts getting more gravity, more IPs need to be added dynamically for the gradually increasing Cluster

Dynamic Scale Increase

- Customers see an uptake of their service or an upcoming event (Black Friday) and want to proactively expand
- Given the stability of their current clusters, in-depth considerations in managing a multi-cluster they don't want to solve the scale problem by creating another cluster

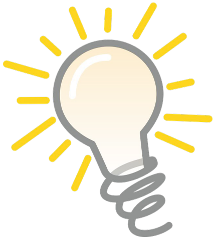
Fragmented Ranges

- Getting a large contiguous block is really difficult, it's a problem that becomes worse as time passes
- Organizational challenges makes it difficult to fulfill a large CIDR block request



https://www.flickr.com/photos/61423903@N06/7632796322

Solution: Dynamic & Discontiguous Pod CIDR



Don’t use cluster pod CIDR to identify cluster originated traffic

Allows for Discontiguous Pod Cluster CIDR to be a piure IPAM problem.

https://www.flickr.com/photos/61423903@N06/7632796322

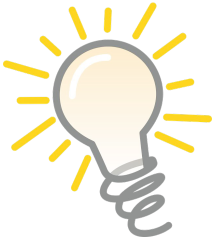
https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/

Solution: Clusters As Islands

- Ability to reuse IPs across Cluster Islands, hence providing IP savings
- Customers want to emulate their existing LAN networks where there is VLAN or routing level segmentation
- Network segmentation especially on cloud where fate-sharing is not needed between all Clusters and Network environment
- Clusters are self serving and do not need to be accessed from outside

https://picryl.com/media/view-in-cambridge-1831-2c38ad

Solution: Clusters As Islands



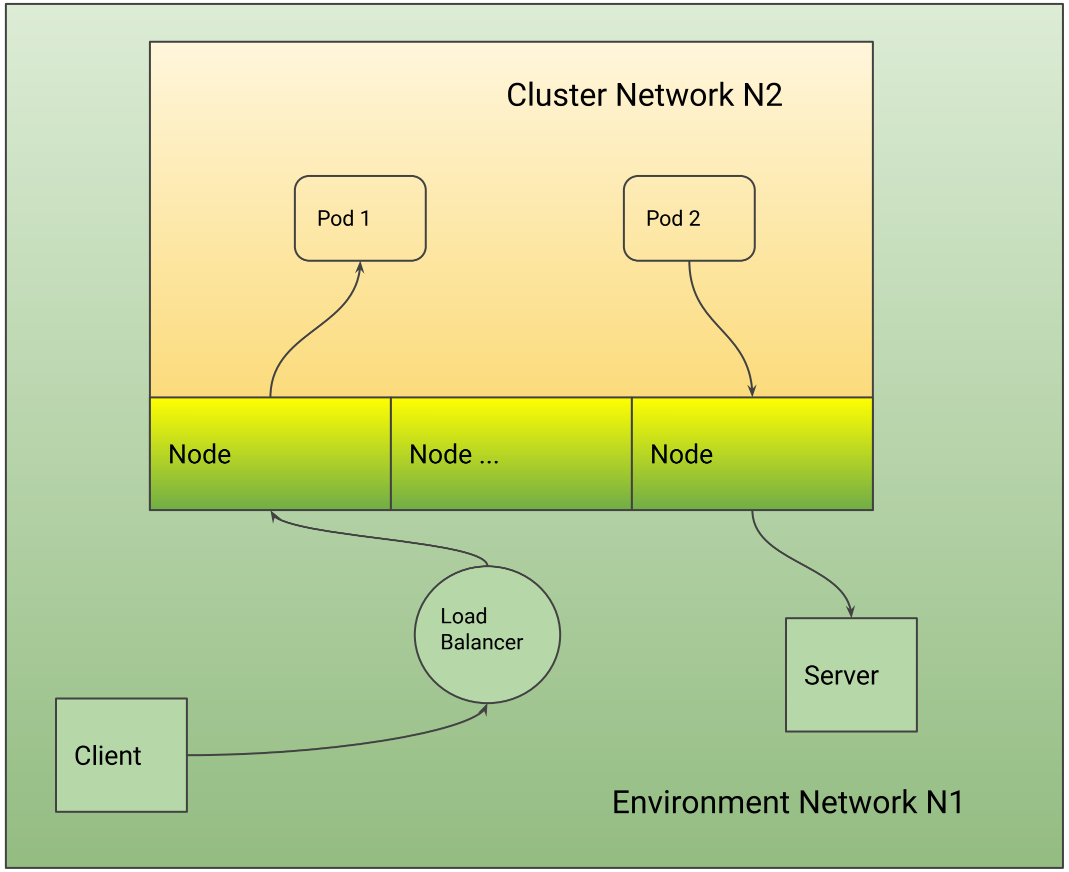
Only Service based connectivity for external traffic.

ServiceType:LoadBalancer or Ingress

https://picryl.com/media/view-in-cambridge-1831-2c38ad

https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/

Solution: Clusters As Islands



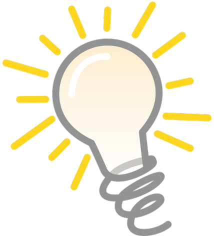
https://picryl.com/media/view-in-cambridge-1831-2c38ad

Solution: Clusters As Hybrid Islands

- Applications staggered between on-prem and cloud
  - unidirectional from on-prem to cloud or from cloud to on-prem
  - bidirectional as well
- Each environment acts as an Island, optimizing IPs
  - On-Prem and Cloud have overlapping IPs
- Communication between the environments happens through a firewall proxy
  - Deployed on-prem
  - Deployed in a standalone VPC
- New Ranges available in cloud but users wary of using it on-prem: CGN, ClassE, Publicly used Private IP

https://en.wikipedia.org/wiki/File:Pound_layer_cake.jpg

Solution: Clusters As Hybrid Islands

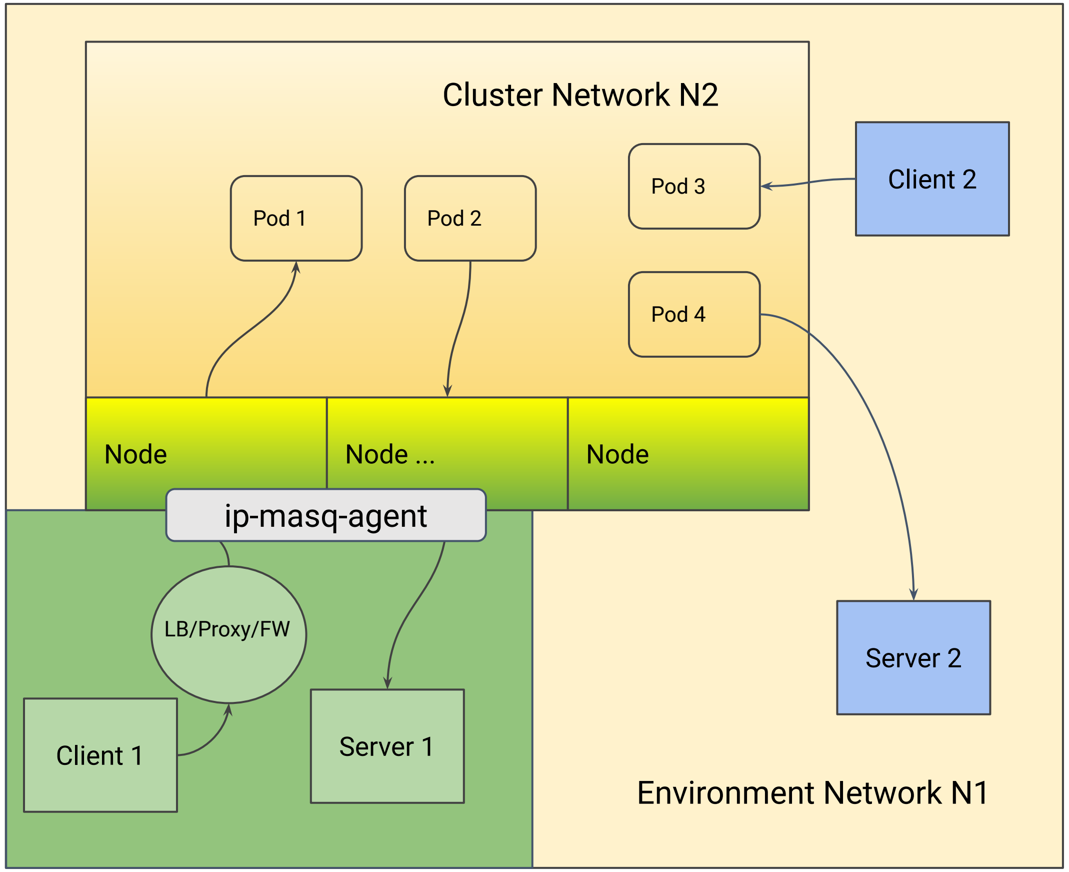


Use ip-masq-agent to masquerade for some ranges.

https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/

https://en.wikipedia.org/wiki/File:Pound_layer_cake.jpg

Solution: Clusters As Hybrid Islands



non-rfc-1918 works well for cluster CIDR

https://en.wikipedia.org/wiki/File:Pound_layer_cake.jpg

Evolving Kubernetes

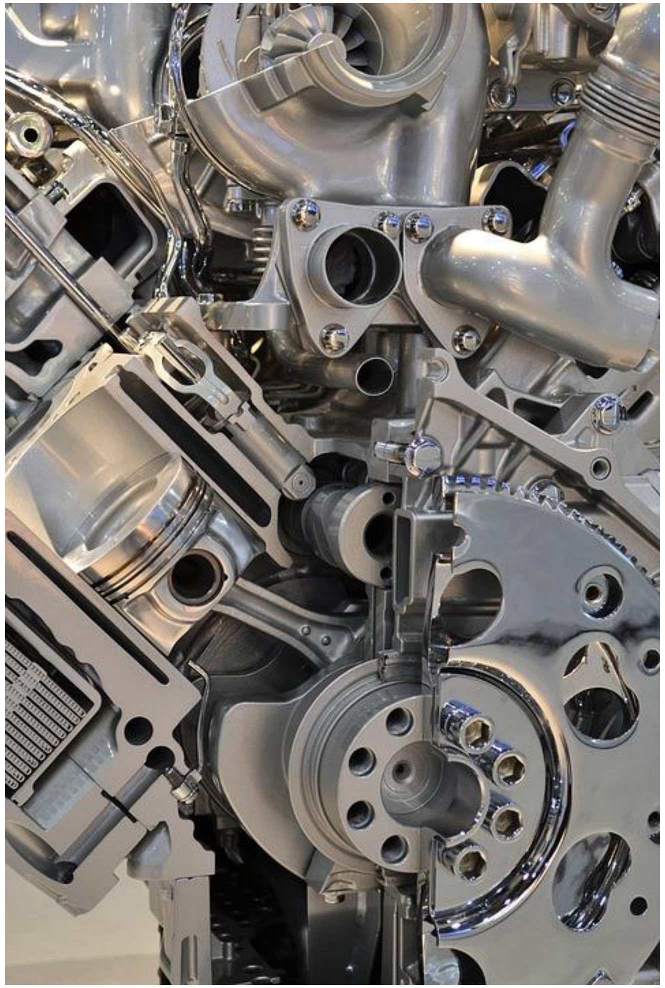
Invitation for community discussion

Kubernetes Improvements

KEP to use per-node information as an alternative to cluster CIDR to detect cluster originated traffic.

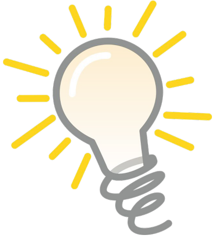
Are we missing Egress as a complement to Ingress ?

https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/



https://pixabay.com/photos/modern-engine-motor-chopper-2773254/

IPv6 - Food for thought



IPv6 only helps with IPAM if ‘only-v6’.

Two Approaches: NAT Gateway vs IPv4 Islands with dual stack

https://pixabay.com/illustrations/idea-enlightenment-light-bulb-light-4438932/

https://www.pexels.com/photo/african-elephant-animal-elephant-portrait-1785286/

Thank You!