aws

Architecture Diagrams

Expose Microservices Using Amazon EKS

Expose Microservices Using Amazon EKS: Architecture Diagrams

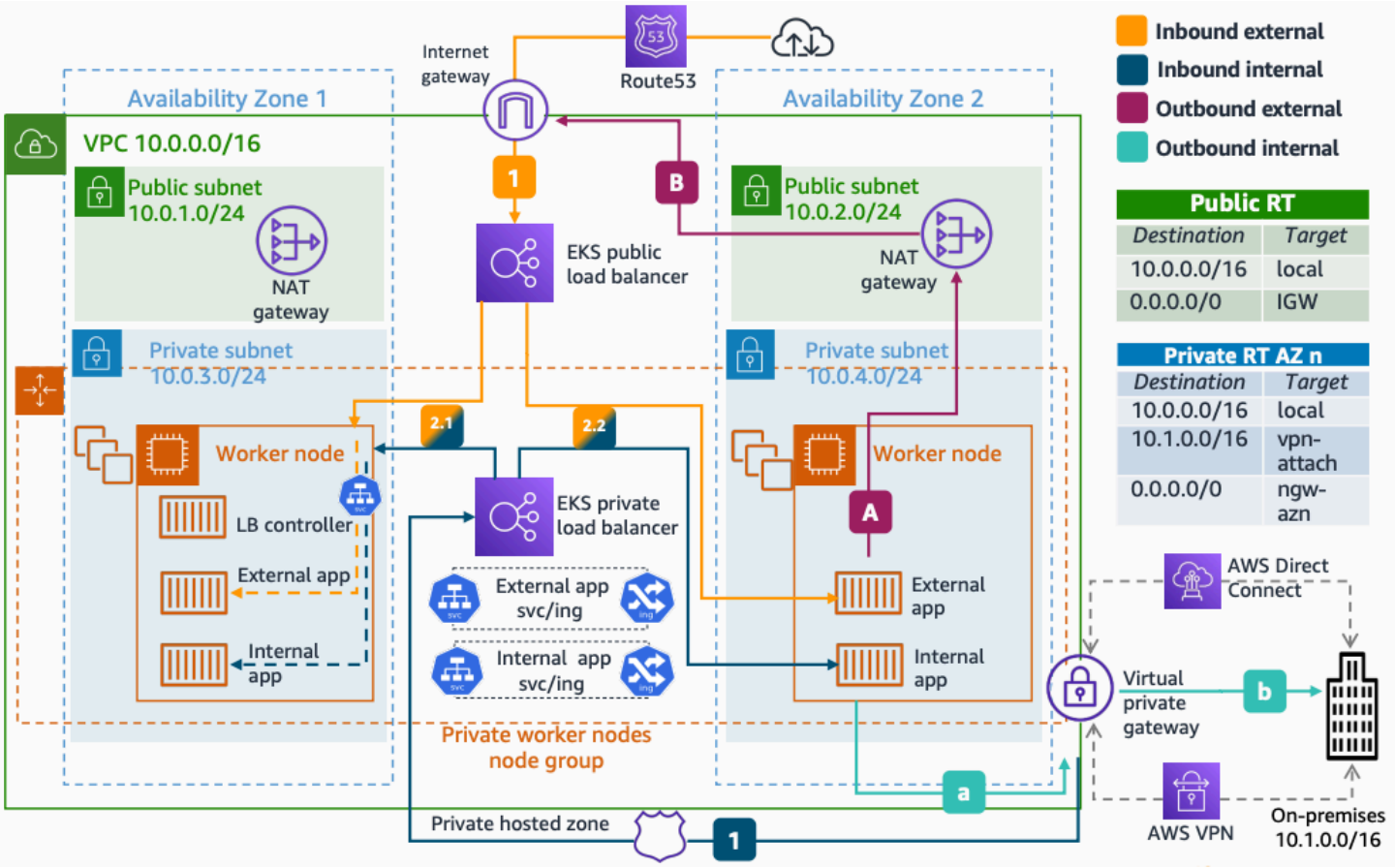
Table of Contents

Expose Microservices in a Hybrid Scenario Using Amazon EKS

Publication date: **February 22, 2022 (Diagram history)**

This architecture shows how to expose Amazon Elastic Kubernetes Service microservices hosted in private subnets to the internet and on-premises networks. The AWS Load Balancer Controller manages Elastic Load Balancers (ELBs) for Kubernetes services and ingresses.

Expose Microservices in a Hybrid Scenario Using Amazon EKS



The following steps describe the inbound external flow:

1. Amazon Route 53 resolves incoming requests to the public ELB deployed by the AWS Load Balancer Controller.

2. The ELBs forward traffic to applications. You can choose between two modes: instance mode (traffic sent to a worker node, then the service redirects traffic to the pod) or IP mode (traffic directed to the IP of the pod directly). See cluster networking details for more information.

The following steps describe the inbound internal flow:

1. Amazon Route 53 resolves incoming requests to the private ELB deployed by the AWS Load Balancer Controller using a private hosted zone.
2. The ELBs forward traffic to applications in instance mode or IP mode.

The following steps describe the outbound external flow:

A. When a pod in a private subnet initiates an outbound request to the internet, the private route table forwards traffic to the NAT gateway (NGW).
B. The public route table forwards traffic from the NGW to the internet gateway (IGW).

The following steps describe the outbound internal flow:

a. A pod in a private subnet initiates an outbound request to the on-premises network. The private route table forwards traffic to the virtual private gateway (VGW).
b. Traffic reaches the on-premises network over the VPN or AWS Direct Connect connection.

> **Note**
>
> You can use ingress controllers such as the NGINX ingress controller as an alternative to the AWS Load Balancer Controller. If you use AWS Fargate for Amazon EKS, you only have pod ENIs in the private subnets and must use ELBs with IP mode.

Further reading

For additional information, refer to the following resources:

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	February 22, 2022
Initial publication	Reference architecture diagram first published.	February 22, 2022
Initial publication	Reference architecture diagram first published.	February 22, 2022

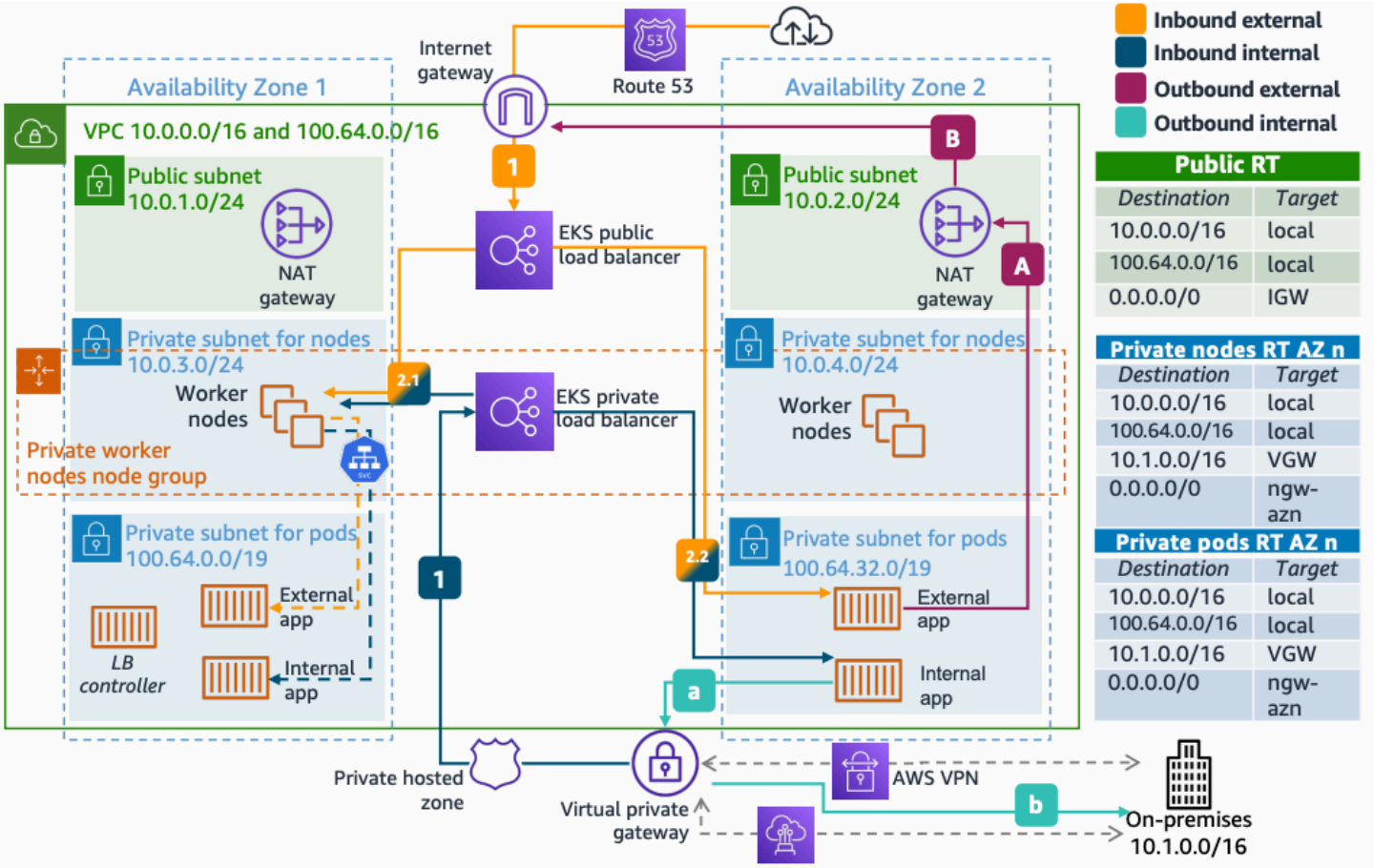
Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.

Deal with Pod IP Exhaustion

This architecture shows how to deal with pod IP exhaustion by adding secondary CIDR blocks from the RFC 6598 address space to your Amazon VPC. Using the CNI Custom Networking feature, pods no longer consume RFC 1918 IP addresses in the VPC.

Deal with Pod IP Exhaustion



The following steps describe the inbound external flow:

1. Amazon Route 53 resolves incoming requests to the public ELB deployed by the AWS Load Balancer Controller.
2. The ELBs forward traffic to applications. You choose between instance mode (traffic sent to a worker node, then the service redirects to the pod) or IP mode (traffic directed to the pod IP directly).

The following steps describe the inbound internal flow:

1. Amazon Route 53 resolves incoming requests to the private ELB deployed by the AWS Load Balancer Controller using a private hosted zone.
2. The ELBs forward traffic to applications in instance mode or IP mode.

The following steps describe the outbound external flow:

A. A pod in a private subnet initiates an outbound request to the internet. The private route table forwards traffic to the NAT gateway (NGW).

B. The public route table forwards traffic from the NGW to the internet gateway (IGW).

The following steps describe the outbound internal flow:

a. A pod in a private subnet initiates an outbound request to the on-premises network. The private route table forwards traffic to the virtual private gateway (VGW).

b. Traffic reaches the on-premises network over the VPN or AWS Direct Connect connection.

> **Note**
>
> The default behavior of Amazon EKS is to source NAT pod traffic to the primary IP address of the hosting worker node. AWS Fargate for Amazon EKS supports additional CIDRs. The ENIConfig custom resource defines the subnet in which pods are scheduled. See this blog post for multi-account settings.

Further reading

For additional information, refer to the following resources:

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	February 22, 2022
Initial publication	Reference architecture diagram first published.	February 22, 2022
Initial publication	Reference architecture diagram first published.	February 22, 2022

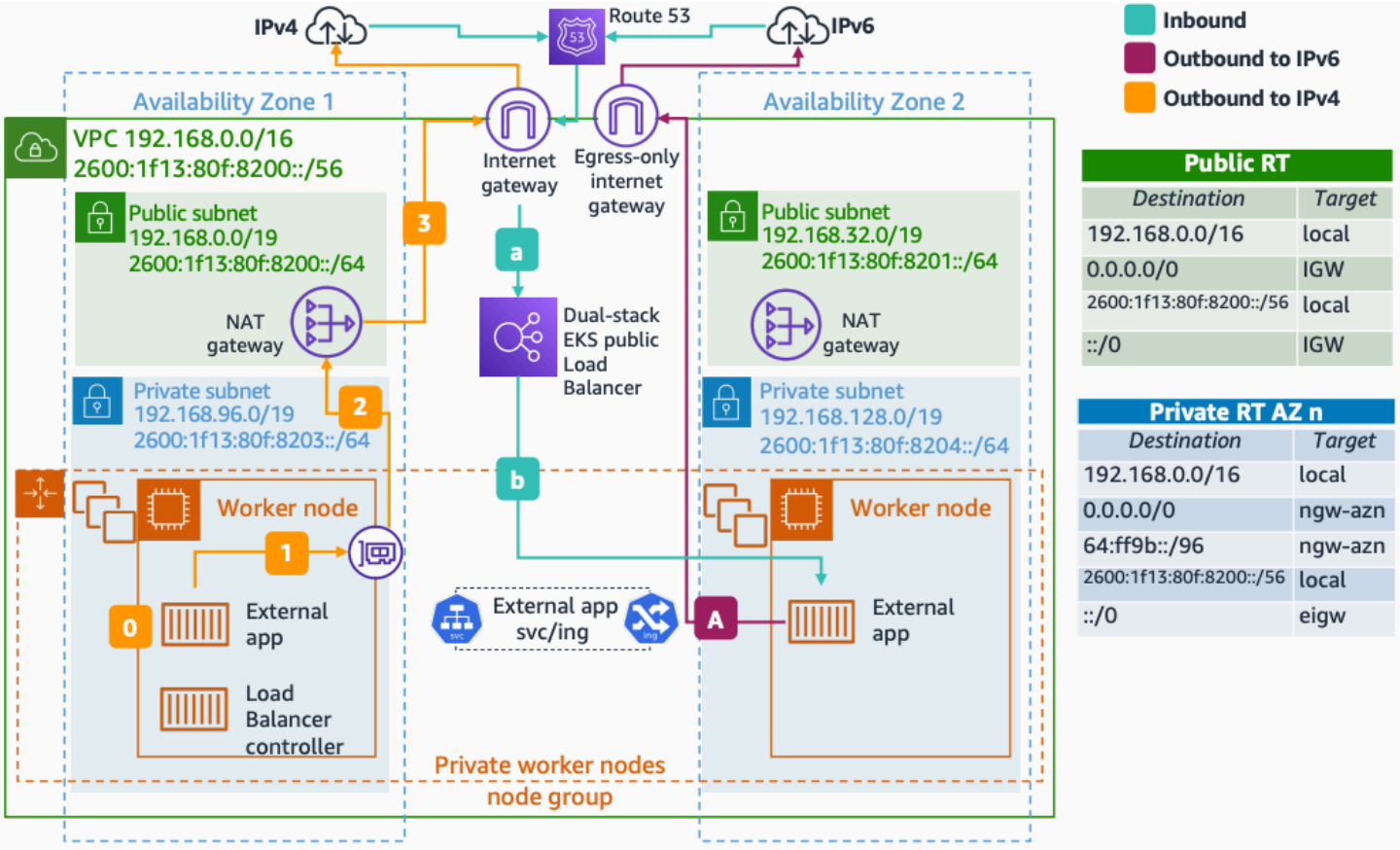
Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.

Expose Amazon EKS Microservices in IPv6 Clusters

This architecture shows how to expose Amazon EKS microservices with IPv6 and connect to both IPv6 and IPv4 endpoints on the internet. Moving to IPv6 also solves pod IP exhaustion because you do not need to work around IPv4 limits.

Expose Amazon EKS Microservices in IPv6 Clusters



The following steps describe the inbound flow:

a. Amazon Route 53 resolves incoming requests to the public ELBs in dual-stack mode deployed by the AWS Load Balancer Controller.

b. The ELB forwards traffic to the IPv6 pods using IP mode.

The following steps describe the outbound IPv6 flow:

A. Any pod communication from within private subnets to IPv6 endpoints outside the cluster routes through an egress-only internet gateway (EIGW).

The following steps describe the outbound IPv4 flow:

1. A pod in a private subnet initiates an outbound request to an IPv4 address on the internet and performs a DNS lookup. Upon receiving an IPv4 "A" response, the pod establishes a connection using the IPv4 address from the host-local 169.254.172.0/22 IP range.
2. The pod's node-only unique IPv4 address is translated through NAT to the IPv4 (VPC) address of the primary network interface attached to the node.
3. The private route table forwards traffic to the NAT gateway, and the private IPv4 address of a node is translated to the public IPv4 address of the gateway.

> **Note**
>
> At the time of this writing, ALB and NLB support dual-stack for only internet-facing endpoints. Amazon EKS implements a host-local CNI plugin chained along with VPC CNI to allocate and configure an IPv4 address for a pod from the 169.254.172.0/22 range.

Further reading

For additional information, refer to the following resources:

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date

Initial publication	Reference architecture diagram first published.	February 22, 2022
Initial publication	Reference architecture diagram first published.	February 22, 2022
Initial publication	Reference architecture diagram first published.	February 22, 2022

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.