White paper
Cisco public

Cisco Application Centric Infrastructure Kubernetes Design

© 2026 Cisco and/or its affiliates. All rights reserved.

Contents

Introduction 3

Executive summary 3

Isovalent Networking for Kubernetes (Cilium) 6

Alternative CNI: Calico 6

Calico BGP design 6

Configurations 9

iBGP for pod subnet redistribution 9

eBGP for pod subnet redistribution 10

Conclusion 11

Version history 12

© 2026 Cisco and/or its affiliates. All rights reserved.

Introduction

This design document covers Kubernetes deployments with Cisco® Application Centric Infrastructure (Cisco ACI®) and provides guidance on networking best-practices for deploying Kubernetes clusters. While it aims to cover a wide range of scenarios and design considerations, please note that not all options are covered. Alternative designs exist, and users are encouraged to conduct their own research and adjust designs to meet specific needs and requirements.

This design document serves as a guide to help navigate various concepts and options to find a balance between features and complexity when deploying Kubernetes clusters in the data center. The goal is to provide users with the knowledge and tools needed to implement effective and efficient solutions.

In this design document, we will be discussing the current best practices for integrating Cisco ACI with the following Container Network Interface (CNI) plugins:

- Isovalent® Networking for Kubernetes: Cilium
- Calico

Executive summary

Application modernization is a growing trend, with many organizations refactoring or re-architecting existing applications to leverage cloud-native technologies for improved agility and scalability. However, a significant portion of applications remain in their legacy state due to various factors (including complexity, cost, and risk aversion), resulting in a hybrid environment where both modernized and traditional applications coexist. To effectively manage traffic between modernized, containerized workloads and legacy applications, a solution that bridges the gap between these environments is essential. A Cisco Data-Center (DC) fabric (Cisco ACI or Cisco NX-OS) can provide network policies and connectivity for legacy workloads while the Kubernetes CNI manages network policies within Kubernetes clusters hosting modernized applications. By leveraging the designs provided for the Cisco DC fabrics and Kubernetes, organizations can achieve consistent security and observability across both their legacy and cloud-native environments, ensuring seamless communication and control.

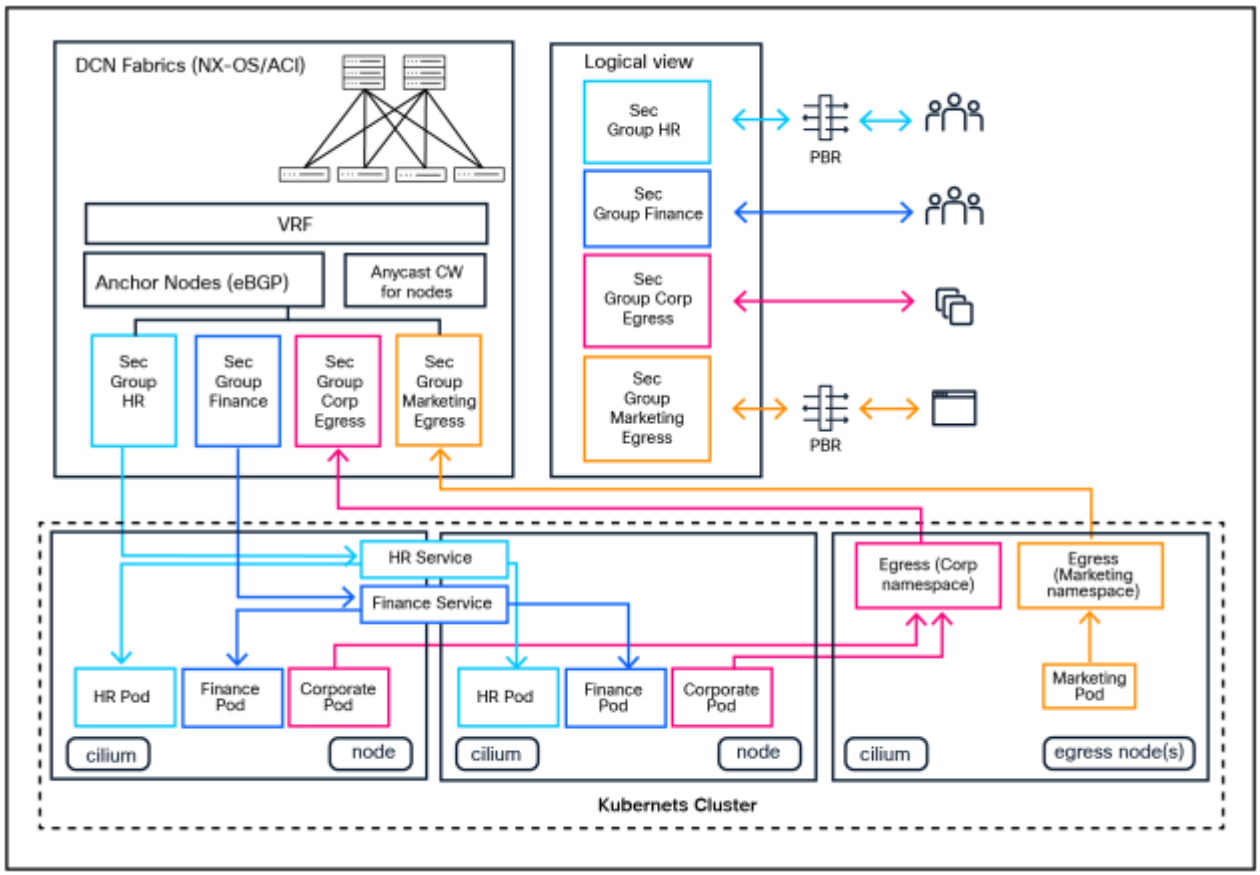
Within Kubernetes, the CNI of choice will enforce east/west network policies, controlling the communication between microservices and pods within the cluster. This east/west policy enforcement is crucial for securing Kubernetes environments, limiting the blast radius of potential security breaches and ensuring compliance with regulatory requirements. For north/south traffic, enforcement can be handled by the underlying fabric, by the CNI itself, or through a combination of both, providing flexibility in how external access and ingress/egress traffic are secured.

© 2026 Cisco and/or its affiliates. All rights reserved.

At a high level, the following traffic patterns are considered:

- Node-to-node communication: this forms the foundation for much of the networking functionality within a Kubernetes cluster, enabling pods, services, and control-plane components to interact effectively.
- External service access: in Kubernetes, a service is an abstraction that defines a logical set of pods and a policy by which to access them. It provides a stable IP address (IP) for accessing applications running in the pods, regardless of how the pods are scaled or rescheduled. The IPs allocated to these services are deterministic. Each service is advertised to the Cisco DC fabrics through Border Gateway Protocol (BGP) as a /32-host route, which can easily be classified using external EPGs (ExtEPGs). This allows the user to apply security policies that control communication to the service, and where needed, apply service redirection to (for example) a firewall. BGP is a well-established and highly scalable routing protocol. Advertising Kubernetes services through BGP allows network administrators to easily scale services and adapt to changing network conditions.
- Egress communication (also known as pod-initiated traffic): pod-initiated traffic encompasses any network connection originating from a Kubernetes pod to external services, APIs, databases, or other resources. Pods in Kubernetes are ephemeral, meaning they can be created, destroyed, and rescheduled frequently. As a result, pods receive dynamically assigned IP addresses. Relying on these IPs for security policies or auditing is unreliable because they change. Kubernetes typically uses Source Network Address Translation (SNAT) on the worker nodes to allow pods to access external networks. SNAT replaces the pod's IP address with the node's IP address. This means that external services only see the node's IP and lose the original identity and context of the pod that initiated the traffic. Egress gateways in Kubernetes provide controlled exit points for traffic leaving the cluster. High Availability (HA) for these egress gateways further enhances the reliability and resilience of this egress. Similar to service IP allocation, allocating IP addresses to egress gateways is deterministic. A user can leverage labels in Kubernetes to specify how pods, or even entire namespaces, egress the cluster. By combining the deterministic characteristics of egress gateway and external EPGs or ESGs with IP selectors, the user can apply granular control to which services pods have access.

© 2026 Cisco and/or its affiliates. All rights reserved.



The above illustration offers a simplified view of how policies can be applied to control north/south communication to and from a Kubernetes cluster. In the above example the HR users can access the HR application – but are first redirected to a firewall for inspection. The users of the Finance application do not have to pass through the firewall and can only access the Finance services. Pods in the Corporate Namespace can access other applications within the data center, whereas pods in the Marketing namespace only have access to the internet through a firewall.

To summarize, the combination of Cisco data-center fabrics and Kubernetes offers a unified solution for managing hybrid environments, bridging legacy and cloud-native applications:

- Security and observability: the Cisco DC fabrics provide consistent security policies for legacy applications, while the CNI enforces granular network policies within Kubernetes clusters.
- Traffic control: the Kubernetes CNI handles east-west traffic, optimizing internal communications, while Cisco ACI excels in north/south traffic management, ensuring precise control and the ability to insert services.
- Service and egress management: deterministic external service access through BGP and controlled egress communication using egress gateways and ESGs enhance scalability and security.

© 2026 Cisco and/or its affiliates. All rights reserved.

Isovalent Networking for Kubernetes (Cilium)

Isovalent Networking for Kubernetes (Cilium) is the **preferred** Container Network Interface (CNI) solution to pair with Cisco ACI or NX-OS VXLAN-based fabrics.

For detailed design documentation, please refer to the following link: https://isovalent.com/briefs/designing-isovalent-enterprise-for-cisco-aci-and-nexus.

Alternative CNI: Calico

Another popular CNI option is Calico. Since Calico supports running BGP, a very similar design can be used for environments using Calico as the CNI. This allows for seamless connectivity with Cisco ACI or NX-OS VXLAN-based fabrics, following the same architectural principles outlined for Cilium.

Calico supports two main network modes: direct container routing (no overlay transport protocol) or network overlay using VXLAN or IPinIP (default) encapsulations to exchange traffic between workloads. The direct routing approach means the underlying network is aware of the IP addresses used by workloads. Conversely, the overlay network approach means the underlying physical network is not aware of the workloads' IP addresses. In that mode, the physical network only needs to provide IP connectivity between K8s nodes while container to container communications is handled by the Calico network plugin directly. This, however, comes at the cost of additional performance overhead as well as complexity in interconnecting your container-based workloads with external non-containerized workloads.

When the underlying network is aware of the workloads' IP addresses, an overlay is not necessary. The network can directly route traffic between workloads inside and outside of the cluster as well as allowing direct access to the services running on the cluster. This is the preferred Calico mode of deployment when running on premises. This guide details the recommended Cisco ACI configuration when deploying Calico in direct routing mode.

You can read more about Calico at https://docs.projectcalico.org/.

Calico BGP design

Calico uses Border Gateway Protocol (either internal [iBGP] or external [eBGP]) to distribute pod routing information between Kubernetes nodes and the fabric, enabling efficient and scalable networking within the cluster. Specifically, Calico can operate in two modes for this purpose:

- iBGP within the cluster: Calico uses iBGP within the cluster to distribute pod routing information between Kubernetes nodes, and this can be configured either as a full mesh or with route reflectors. This internal iBGP setup allows nodes to learn about pod IPs hosted on other nodes directly, enabling pod-to-pod communication without requiring overlay networks.
- External BGP routers: alternatively, Calico can rely on external BGP (eBGP) routers outside the Kubernetes cluster to distribute pod-routing information. In this scenario, Calico nodes peer with these external routers, which then propagate the pod routes across the broader network infrastructure. This approach integrates Kubernetes pod networking with existing network routing policies and infrastructure.

© 2026 Cisco and/or its affiliates. All rights reserved.

By leveraging either iBGP within the cluster or external BGP routers, Calico ensures that pod IP routes are efficiently distributed between Kubernetes nodes, enabling scalable and performant networking for containerized workloads.

Note: Regardless of how the pod subnet is distributed, eBGP will be used for service advertisements.

Options for pod-route distribution

1. iBGP full mesh (default, up to 100 nodes)

- By default, Calico establishes an iBGP full mesh, where each node peers with every other node in the cluster.
- **Calico POD routing:**
  - Calico natOutgoing should be enabled if the pod subnet is not advertised to the fabric.
  - Calico encapsulation is disabled.
- **Advantages:**
  - Simple to deploy and manage for small to medium clusters
  - Scales efficiently up to approximately 100 nodes
  - The pod subnet can optionally be advertised to the fabric.
  - eBGP can be used to advertise service IPs for ingress or egress traffic.

2. iBGP with route reflectors (scalable option for large clusters)

- For clusters larger than 100 nodes, a full mesh becomes inefficient due to the exponential number of BGP sessions.
- **Use route reflectors:**
  - Calico nodes form iBGP sessions only with two (for redundancy) or more designated route reflectors inside the cluster.
  - The route reflectors redistribute pod routes between all nodes, drastically reducing the number of BGP sessions each node needs to maintain.
  - This architecture allows for much greater scale, supporting very large Kubernetes clusters efficiently.
- **Calico POD routing:**
  - Calico natOutgoing should be enabled if the pod subnet is not advertised to the fabric.
  - Calico encapsulation is disabled.
- **Advantages:**
  - Highly scalable compared to iBGP full mesh
  - The pod subnet can optionally be advertised to the fabric.
  - eBGP can be used to advertise service IPs for ingress or egress traffic.

© 2026 Cisco and/or its affiliates. All rights reserved.

3. eBGP high scale, pod subnet routable in fabric

- For users who need maximum scale and want the pod subnet to be routable within the Cisco ACI fabric, iBGP is disabled on Calico nodes.
- **eBGP only:**
  - Calico nodes do **not** peer with each other using iBGP.
  - Each Calico node peers only with ACI via eBGP.
  - ACI takes the responsibility for redistributing pod subnet routes across the fabric and back to the Calico nodes.
  - Calico natOutgoing and encapsulation are disabled.
- **Advantages:**
  - Enables the pod subnet to be visible and routable throughout the ACI fabric.
  - Supports very large-scale Kubernetes deployments.
- **ACI configuration:** The default ACI config needs a minor change:
  - The ACI BGP peer must be configured with **AS Override** and **Disable Peer AS Check** options. These settings allow pod routes to be sent back into the same AS (the Kubernetes cluster AS), which is necessary for pod subnet redistribution within the Kubernetes cluster.

For all three options, pod-to-pod communication occurs without any network encapsulation. Each Calico node is assigned a unique /24 (by default) subnet carved from the primary pod subnet, and this /24 is used for the pods scheduled on that node. Through either iBGP or eBGP, Calico redistributes these /24 subnets so that every node learns the routing information for all other nodes' pod subnets. This allows each node to route traffic directly to the node IP hosting the destination pod, enabling efficient, direct pod-to-pod communication across the entire cluster.

Summary table

Option	Scale	Pod subnet routable in Cisco ACI?	Notes
iBGP full mesh	< 100	Optional	
iBGP with route reflectors	> 100	Optional	
Cisco ACI handling eBGP for pod subnet	> 100	Mandatory	AS override, disable peer AS check required in ACI BGP peer config

Although, by default, pod subnet advertisement to the fabric is enabled when using iBGP (options 1 and 2), we recommend using the eBGP option (option 3) if pod subnet routing within the ACI fabric is required.

Relying on eBGP for pod subnet advertisement simplifies the network design by avoiding the need to run both eBGP and iBGP in parallel, reducing operational complexity. In our opinion, the iBGP-based designs should be reserved for cases where the pod subnet should not be routable in the fabric, thereby keeping the pod network private to the Kubernetes cluster. To disable Calico's advertising the pod subnet to the fabric, the user can configure a BGPFilter. Optionally, ACI can also be configured with an import-route control filter to protect against Calico misconfigurations.

© 2026 Cisco and/or its affiliates. All rights reserved.

Additional notes:

- Calico does not support MagLev and Cisco ACI does not support ECMP resilient hashing, because such a change in the ECMP next hops for an exposed service will result in connection reset for pre-existing flows. If this is not acceptable, please consider using Isovalent Networking for Kubernetes.
- Bidirectional Forwarding Detection (BFD) requires Calico Enterprise.

Configurations

The Cisco ACI configuration for Calico is mostly identical to the one for Isovalent Networking for Kubernetes (Cilium). Please reference the ACI configuration section in the Cilium design guide at this link: https://isovalent.com/briefs/designing-isovalent-enterprise-for-cisco-aci-and-nexus.

iBGP for pod subnet redistribution

If iBGP is used for the pod subnet advertisement between the Calico nodes, and the pod subnet advertisement to ACI is filtered out by Calico, no changes to the ACI configuration are required.

Calico should be configured with the following:

- A BGPFilter to block pod subnet advertisement
- No node-to-node mesh
- natOutgoing enabled

```
---
kind: BGPFilter
apiVersion: projectcalico.org/v3
metadata:
  name: no-pod-subnet-to-aci
spec:
  exportV4:
    - action: Reject
      matchOperator: In
      cidr: <POD_CIDR>
```

© 2026 Cisco and/or its affiliates. All rights reserved.

```
apiVersion: projectcalico.org/v3
kind: BGPConfiguration
metadata:
  name: default
spec:
  asNumber: <AS>
  nodeToNodeMeshEnabled: true
  serviceLoadBalancerIPs:
  - cidr: <LB_CIDR>
```

```
apiVersion: projectcalico.org/v3
kind: IPPool
metadata:
   name: default-ipv4-ippool
spec:
  blockSize: 24
  cidr: <POD_CIDR>
  ipipMode: Never
  natOutgoing: true
  nodeSelector: all()
  vxlanMode: Never
```

eBGP for pod subnet redistribution

If eBGP is used for pod subnet redistribution, the ACI BGP peer configuration needs to be changed by adding the following two options:

- AS override
- Disable peer AS

These settings allow pod routes to be sent back into the same AS (the Kubernetes cluster AS), which is necessary for pod subnet redistribution within the Kubernetes cluster.

Calico should be configured to **disable**.

- Node-to-node mesh
- NatOutgoing

© 2026 Cisco and/or its affiliates. All rights reserved.

```
apiVersion: projectcalico.org/v3
kind: BGPConfiguration
metadata:
  name: default
spec:
  asNumber: <AS>
  nodeToNodeMeshEnabled: false
  serviceLoadBalancerIPs:
  - cidr: <LB_CIDR>
```

```
apiVersion: projectcalico.org/v3
kind: IPPool
metadata:
   name: default-ipv4-ippool
spec:
blockSize: 24
  cidr: <POD_CIDR>
  ipipMode: Never
  natOutgoing: false
  nodeSelector: all()
  vxlanMode: Never
```

Conclusion

By combining Cisco ACI and either Calico or Cilium (Isovalent Networking for Kubernetes) CNI plugins customers can design Kubernetes clusters that can deliver both high performance (no overlays' overhead) as well as providing exceptional resilience while keeping the design simple to manage and troubleshoot.

© 2026 Cisco and/or its affiliates. All rights reserved.

Version history

Version	Date	Change
1.0	July 10, 2019	Initial Release
2.0	January 14, 2022	Floating SVI
2.1	December 16, 2022	Kube-Router Support
2.2	July 26, 2023	Cilium Support, Tested Kubernetes Distros
2.3	July, 2025	Removed Kube-Router, added link to new Cilium Design, only Floating SVI design is supported

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Printed in USA C11-743182-05 01/26

© 2026 Cisco and/or its affiliates. All rights reserved.