

A step-by-step guide for implementing microsegmentation

Unlike traditional applications that rely on network-based segmentation, cloud-native applications require a dynamic policy driven approach to segment their workloads. This paper provides a step-by-step guide for implementing microsegmentation for modern cloud-native workloads.



Table of Contents

Introduction	3
What is microsegmentation?	4
Why is microsegmentation important?	4
Types of microsegmentation	5
Workload segmentation	5
Regulatory compliance-based segmentation	5
Container segmentation	6
Kubernetes segmentation	6
Environmental segmentation	6
Application tier segmentation	6
Process-based nano-segmentation	6
User segmentation	7
7 best practices for successful microsegmentation	8
Design best practices	8
Implementation best practices	8
Simplifying microsegmentation with Calico	9
Identity-aware microsegmentation with zero trust	9
Performance at scale	9
How does microsegmentation work with Calico?	10
Calico microsegmentation feature table	11
Summary	12
Additional resources	13
Getting started	13

Introduction

Cloud-native is an approach to building and running applications that exploits the advantages of the cloud-computing delivery model. Cloud-native architecture takes full advantage of the cloud's distributed, scalable, and flexible nature, abstracting away many layers of infrastructure—networks, servers, and operating systems, for example—and enabling them to be defined in code. Cloud-native architecture also guides the design of data center infrastructure, using virtualization, containerization, microservices, and workload mobility technologies.

Due to their distributed nature, cloud-native applications generate a significant percentage of east-west traffic (75% or more), representing a fundamental change from past traffic patterns. This shift has created greater potential for free lateral movement within these application infrastructures. It is becoming increasingly difficult to defend the perimeter, and there is a need to create multiple micro-perimeters around each valuable asset the organization needs to protect.

This is where microsegmentation comes in.

What is microsegmentation?

Microsegmentation was initially conceived to moderate server-to-server traffic within a network segment but has since expanded to include traffic between segments. Modern microsegmentation can regulate how servers or workloads in one network segment can communicate with those in another.

You can base microsegmentation policies and permissions on the identity of a resource, which makes it independent from the infrastructure. In comparison, network segmentation depends on IP addresses for the networks. Microsegmentation is thus a great way to create intelligent groupings of the workloads in your data center according to their characteristics.

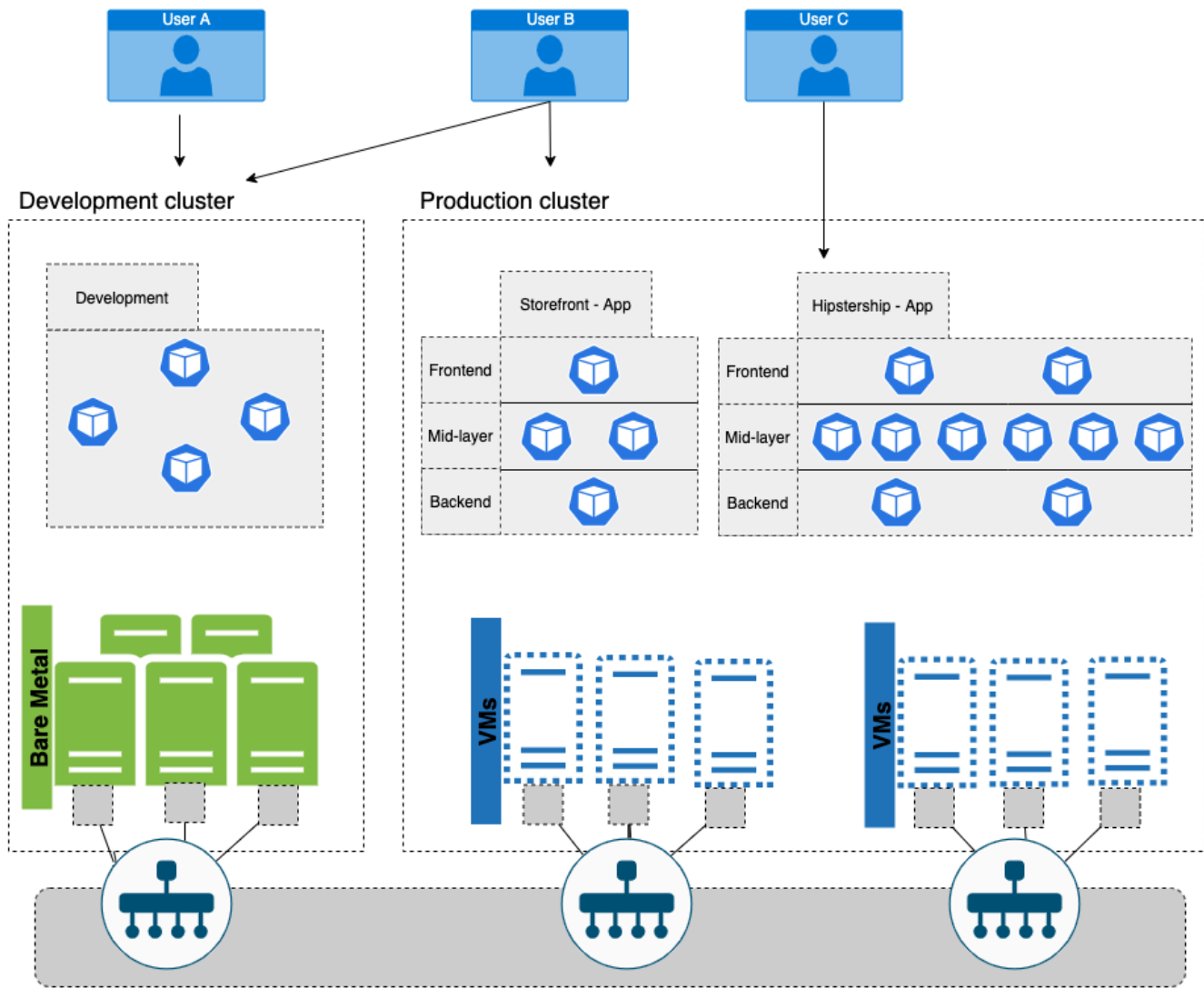
With the massive adoption of containerized and cloud-native architectures, microsegmentation is changing. Traditional segmentation solutions cannot scale and effectively enforce security in a containerized environment, such as a Kubernetes cluster. New microsegmentation technologies are emerging to discover endpoints in cloud-native environments, define policies, and apply policies dynamically to constantly changing cloud-native networks.

Why is microsegmentation important?

Traditionally, companies defended their network perimeter with various security tools deployed at the network edge (primarily the firewall). The main focus was on screening north-south traffic between the network and external traffic sources. These tools were like the moat protecting a castle, where inhabitants inside the “castle” (the network perimeter) were inherently trusted.

Microsegmentation allows network administrators to create secure “islands” within their distributed infrastructure and control access to those islands for all types of users, whether they are outsiders, customers, or employees logging in from various locations. It is central to implementing a zero trust security model. Specifically, it is a core component of zero trust network access (ZTNA) solutions. With microsegmentation, network segments can be protected with a handful of consistent identity policies rather than hundreds of unwieldy IP-based rules.

Types of microsegmentation



Workload segmentation

This type of microsegmentation ring-fences applications to protect sensitive communication. This includes controlling traffic between applications (whether they run on containerized workloads, hypervisors, or bare metal) in proprietary data centers and public or hybrid cloud environments.

Regulatory compliance-based segmentation

High-value applications that provide critical services contain sensitive or personal data or are subject to regulations (such as HIPAA, SOX, or PCI DSS), must be protected. Organizations can leverage application segmentation to improve their application security and maintain compliance.

Container segmentation

Containers might hold sensitive data or critical business information and are often deployed as microservices within a Kubernetes cluster. If the containers are not segmented, sensitive information may be exposed to anyone accessing the network.

Kubernetes segmentation

Kubernetes allows you to use network policies when implementing network segmentation. You can use network enforcement capabilities provided natively in Kubernetes, or infrastructure layers like service meshes.

The default settings in Kubernetes do not restrict communication between containers, pods, and nodes; these components can communicate between namespaces or within the same namespace. You can use network policies to restrict this communication, starting with adding a policy that denies all communication and then specifying what communication is allowed.

Kubernetes pods need to communicate with each other. When defining network policies, you should systematically list all the pods each given pod must communicate with. You should also specify a list of allowed and restricted ingress and egress public internet communication.

Environmental segmentation

In a traditional network, assets are dynamically placed throughout the development, staging, testing, production environments, and across public or hybrid clouds, making it difficult to control and protect them. Environmental microsegmentation isolates various deployment environments from each other, restricting their communication.

Application tier segmentation

It is common for an N-tiered application to have application, web, and database tiers, which might need to be protected from each other through segmentation. Microsegmentation at the application tier level prevents unauthorized lateral movement between workloads by dividing them based on roles.

A segmentation policy could, for example, permit the processing tier to communicate with the database tier but not with the web or load balancer tier. This helps reduce the attack surface.

Process-based nano-segmentation

You can achieve a higher level of granularity in your segmentation by reducing the attack surface into smaller units. This means you treat each virtual machine (VM) or bare metal server as an individual unit. You can implement this for on-premises data centers as well as cloud environments.

To implement process-based nano-segmentation, you must dynamically program precise outbound and inbound security rules for each workload. The platform can then create an adaptive perimeter designed around each computing instance, which creates a segment of one.

By combining process-based nano-segmentation with an allowed policy model, you can define permitted workload interactions for your applications without network dependencies. You can then further drill down your segments. For example, you can separate two instances of one process that run on the same machine into two isolated segments.

User segmentation

User segmentation limits application visibility to members of specified groups, using identity services like Microsoft Active Directory. Group membership and user identity form the basis of user segmentation, so there is no need to change the infrastructure. With user segmentation, each user in your VLAN might have a different policy providing different access permissions.

7 best practices for successful microsegmentation

To implement microsegmentation in your organization, you need to design an architecture that suits your security and productivity needs, and implement it gradually to avoid disrupting normal network operations.

Design best practices

1. Define your boundaries

Microsegmentation is most effective when it uses well-defined boundaries. Define the objectives of your applications according to your business needs and the categorization or identification of end users. This allows you to identify the desired boundaries for your applications, which determine the type of information that can be exchanged.

2. Use an application-centric approach

The next step is to create the boundaries for each application. Establish context-based visibility for each application and define any internal or external communication required. Identify the users that need to access the application and the specific data or application services they require.

3. Define levels of access

Applications often have tiers for services that are relevant to a particular group of users. Use a least-privilege approach, starting with the lowest privilege level and applying additional privileges for each user group and service.

Implementation best practices

4. Use a crawl-walk-run implementation model

Once you have identified the infrastructure assets that need protection, you must group these assets according to logical definitions. Structure your implementation to select a group of assets (such as applications, datasets, servers, or users) and then define their implementation process. Evaluate the methodologies and verification process to strengthen and enforce the implementation process.

5. Label assets for security

Identify assets and attribute them to enable effective policies. Applications and components should be labeled and grouped accordingly. Create policies based on labels and application visibility.

6. Author and configure policies

You can author and configure policies to enhance your visibility into the communications between assets such as servers and applications. This granular level of visibility allows you to customize your microsegmentation policies according to your business requirements.

7. Simulate and validate applications

The best way to ensure your security policies are effective is to test them using simulations. This allows you to identify and address gaps in the implementation policy.

Simplifying microsegmentation with Calico

Calico takes a universal approach that provides a unified, cloud-native workload segmentation model and single policy framework that renders multiple, siloed solutions a thing of the past. Calico's common segmentation model and unified policy framework work across all your environments, including any combination of cloud and hybrid providers, cloud instances, Kubernetes distributions, and containers.

Calico's unified solution is an operationally simpler replacement for multiple, costly, proprietary approaches and the skill-sets required to manage them. Workloads are dynamically segmented based on metadata and labels (identity) attached to those workloads, meaning you can securely deploy new or updated workloads without having to add or change your segmentation policies. This approach eliminates the complexity and operational overhead associated with policy management. At the same time, it can quickly scale to meet the expansion or contraction of your microservice environment.

Identity-aware microsegmentation with zero trust

Cloud-native applications require a security design that does not rely on network addresses to protect communication between workloads. Calico correlates security with workload identity, the focal point of Calico's microsegmentation solution to provide optimal protection. Workload identity is the critical element that enables zero trust with identity-aware microsegmentation.

Calico assigns every protected container a workload identity.

- Each identity consists of contextual attributes, including metadata from cloud-native sources.
- Identity ensures accurate network visibility. Protected workloads send and receive identity upon each communication request.
- Identity is baked into network flow visibility, eliminating reliance on contextless IP addresses and strengthening workload defenses.
- Calico verifies the identity of the communicating workloads rather than IP addresses. If the workload is neither verified nor authorized, network access requests are denied.

When deployed with existing security controls, Calico's identity-aware microsegmentation with zero trust provides a defense-in-depth security posture.

Performance at scale

Many segmentation tools utilize a central policy evaluation engine that becomes a choke point for cloud deployments. When using this legacy architecture, changes to segmentation policies can take hours to take effect. When auto-scaling your microservices or deploying modifications, this time lag results in lengthy deployment times and can cause service outages.

Calico utilizes a cloud-native distributed architecture that can accept and enforce changes across cloud-scale environments in milliseconds. This enables rapid auto-scaling of your microservices environment and the ability to rapidly thwart security incidents by rolling out segmentation policy changes in response to an attack.

Calico also deploys a declarative, pull-based policy model. Whenever a new endpoint (e.g., workload or application) is added or removed, the agent automatically pulls (downloads) the policy and maintains the configuration. Calico Enterprise doesn't need to monitor the endpoints for changes. As a result, policy changes are instantly applied to your infrastructure.

How does microsegmentation work with Calico?

1. Endpoint discovery

Calico discovers all the endpoints that need to be secured including containers and Kubernetes nodes. In addition, you can also register host endpoints and assign labels.

2. Create and enforce policies

An example of a policy is: Deny all traffic from "Zone: DMZ" to "Zone: Restricted." Calico segments endpoints according to the policy definition and based on labels that identify those endpoints. This enables you to securely deploy new or updated workloads and applications without adding or changing your segmentation policies.

3. Observe, log, and secure

Calico creates flow logs enriched with workload labels and applied policies for all the network connections. Flow logs enhance observability and simplify troubleshooting. This approach eliminates much of the complexity and operational overhead associated with managing segmentation policies.

Calico's approach to microsegmentation delivers key benefits.

- Correlates security with workload identity to enable identity-aware microsegmentation with zero trust.
- Eliminates the risks associated with lateral movement in the cluster by cyber attackers in search of sensitive data and other high-value assets.
- Provides instant response to security threats with a cloud-native distributed architecture that can dynamically enforce security policy changes across cloud-scale environments in milliseconds.
- Provides a single, unified security model and single-policy framework that works seamlessly across multiple application and workload environments.
- Works everywhere, including any combination of cloud and hybrid providers, cloud instances, Kubernetes distributions, and containers.
- Eliminates the operational inefficiencies of deploying multiple, siloed, segmentation solutions.
- Easily scales to a large number of endpoints, supported by a declarative, automated, pull-based approach to policy changes.

Calico microsegmentation feature table

VISIBILITY DEPTH & BREADTH	Calico Cloud / Enterprise
Dynamic Service graph for live view of activity	Yes
Automated endpoint discovery	Yes
Real-time & historical visualization of all communication	Yes
Granular, live view of workloads & communications	Yes
Automatic service dependency map generation	Yes
Purpose-built visualizations to aid policy development	Yes

POLICY CREATION AND ENFORCEMENT	Calico Cloud / Enterprise
Allowlist zero trust policy model	Yes
Denylist policy model	Yes
Hybrid policy model	Yes
Automatic policy generation	Yes
Build and test policies before enforcement	Yes
Hierarchical policies	Yes
New workloads automatically inherit policies	Yes
Role-based views with policies scoped to application and location	Yes

SEGMENTATION GRANULARITY	Calico Cloud / Enterprise
Workload segmentation	Yes
Environment segmentation	Yes
Application-tier segmentation	Yes
Container-level segmentation	Yes
Kubernetes segmentation	Yes
Regulatory compliance-based segmentation	Yes

BREACH DETECTION AND RESPONSE	Calico Cloud / Enterprise
Vulnerability detection for files, IP addresses, domains, and DNS	Yes
Deception/Honeypot/Honeypod capabilities	Yes
Incident response integration	Yes
Built-in threat intelligence with cloud-native firewall	Yes

Summary

Cloud adoption is at an all-time high, and Calico's cloud-native architecture takes full advantage of the public cloud's distributed, scalable, and flexible nature, abstracting away many layers of infrastructure—such as networks, servers, and operating systems—and enabling them to be defined in code. Current application deployments generate a significantly greater percentage of east-west traffic than in the past. This shift has created a larger attack surface area and greater potential for free lateral movement within these application infrastructures. Calico's common segmentation model and unified policy framework work across all environments, including any combination of cloud and hybrid providers, cloud instances, Kubernetes distributions, and containers. Using Calico's identity-aware microsegmentation, system administrators can create policies that limit network traffic between workloads based on a zero trust approach.

About Tigera

Tigera empowers organizations to secure, observe, and troubleshoot containers, Kubernetes, and cloud. Our commercial products include Calico Enterprise, a self-managed security and observability platform, and Calico Cloud, a Kubernetes-native cloud service that extends the declarative nature of Kubernetes. Our open-source offering, Calico Open Source, is the most widely adopted container networking and security solution. Tigera's solutions specify security and observability as code to ensure consistent enforcement of security policies, which enables DevOps, platform, and security teams to protect workloads, detect threats, achieve continuous compliance, and troubleshoot service issues in real time.

Powering more than 100M containers across 1.5M nodes in 166 countries, Calico software is supported across all major cloud providers and Kubernetes distributions, and is used by leading companies including AT&T, Discover, Merck, ServiceNow, HanseMerkur, RealPage, L3Harris, and Mindbody.

Additional resources

SOC 2 Security Compliance for Hosts, VMs, Containers, and Kubernetes

[Download Now](#)

PCI Compliance for Hosts, VMs, Containers, and Kubernetes

[Download Now](#)

Getting started

Learn more about microsegmentation

[Try Now](#)

For more information, please visit us at www.tigera.io

[Learn More](#)



Tigera, Inc.

58 Maiden Lane, Fl 5
San Francisco, CA 94108

+1 (415) 612-9546 / www.tigera.io

"Tigera", the Tigera logo, Calico, Calico Enterprise, and Calico Cloud are trademarks of Tigera, Inc. All rights reserved. Other trademarks are the property of their respective owners. Copyright © 2021 Tigera, Inc.