WHITE PAPER

7 security best practices for DevOps managing containerized EKS workloads in EKS

In collaboration with

TIGERA

Table of contents

Introduction

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that runs and scales Kubernetes applications in AWS. EKS ensures a secure, scalable, and managed Kubernetes environment with security patches automatically applied to your cluster's control plane. However, containerized workloads running in EKS are still at risk of potential security threats and vulnerabilities in your AWS environment.

Securing workloads from build time to runtime in EKS

Security is a constantly evolving challenge, and it doesn't take long for threat actors to expose and exploit container vulnerabilities—or for compliance and access issues to arise. Hence, improving container security architecture and automation from the build phase to the eventual runtime in production has become a priority for security teams.

Denial-of-service (DDoS) attacks and kernel and orchestration exploits have plagued containers, threatening enterprise cloud assets, productivity, and innovation. EKS addresses potential Kubernetes OS issues through patches and leverages Calico Cloud for container security, runtime threat defense, and observability and troubleshooting features to handle new vulnerabilities and risks at the workload level.

Following are seven security best practices used by the DevOps team to manage containerized workloads in EKS using Calico Cloud.

1. Least privilege access with microsegmentation

Isolation can initially cause frustration among DevOps engineers, but it's one of the most important security best practices for all software development teams should establish within their applications. Restricting software access reduces the potential blast radius associated with software errors or malware injected into the container. Containers, with their portable, read-only image, have a smaller footprint but lack true sandboxing. If an attacker exploits any host OS vulnerability, all containers sharing the OS could be compromised.

We can achieve user-level isolation and access restrictions through identity and access management (IAM) permissions within AWS, restrict ports and protocols on AWS resources by creating security groups in AWS, and even enforce Kubernetes RBAC to restrict the service accounts that can modify network policies, which control the traffic within your EKS clusters.

Calico Cloud with EKS allows organizations to enforce workload isolation to limit the blast radius associated with potential privilege escalation incidents. Regardless of the permissions assigned to service accounts within the cluster, Calico's security policy framework can limit the traffic permitted between workloads based on a well-defined Kubernetes label schema—and then limit that traffic further based on approved ports and protocols between those identity-aware workloads.

2. Reduce attack surface with zero-trust architecture

Docker and Kubernetes offer faster container deployment, but a vulnerable orchestration layer expands the attack surface. Misconfigured Kubernetes, like over-provisioned privileged access, could open direct control over the container fleet to an attacker. Application breakdown or exposed internals can significantly broaden the attack surface.

Calico Cloud with EKS enables you to go one step further by enforcing the concept of least privilege via security policies. By limiting workload communications to specific ports and protocols between approved workloads, you can prevent much broader attack surfaces. You can also limit the user's CRUD (create, read, update, delete) permissions on objects within Kubernetes with Kubernetes RBAC. That way, you could assign a team manager read-only access to a policy without worrying about any possible misconfiguration from that manager.

3. Misconfigured containers

Yet another attack vector is compromised container credentials (e.g., API key or username/password), which invites an attacker to spoof into the database and cloud services.

Calico Cloud with EKS protects containers during development and production, reducing the attack surface with vulnerability and misconfiguration detection. With image assurance and configuration assessment based on standardized benchmarks such as CIS, you can detect misconfigured Kubernetes environments and address the gaps in real-time.

4. Strengthening security with rule-based controls

Rapidly changing container infrastructure rules and signature-based controls don't align well with static regulatory compliance requirements. It's why securing a dynamic container infrastructure using traditional network and endpoint controls won't work. Security Groups in AWS work diligently to secure the north-south traffic going in and out of the EKS cluster. However, it's difficult to target specific pods in network namespaces, and for that security group rule to consistently work, since workloads are ephemeral (subject to die regularly).

Calico Cloud, unlike traditional firewall technologies, has identity-awareness of application workloads running within a Kubernetes cluster. Traditional security tools worked better with legacy VMs with fixed IP addresses. However, to secure your EKS clusters, you need Kubernetes-native context about objects, such as label selectors and services. Conventionally, endpoint security teams would target those endpoints (mainly VM's/servers) by IP address. Calico's security policies allow organizations to target workloads based on a fixed label schema, which works consistently in highly distributed environments.

5. Automatic detection and blocking of vulnerable container

Containers use images stored in publicly available repositories like GitHub, with dependencies on other images and libraries. Source code vulnerability can spread to thousands of other containers. A clever attacker can also run their malicious images, compromising host data.

Amazon EKS integrates with Amazon ECR (Elastic Container Registry)—a managed AWS service that provides users the ability to push their software packed in containers into an AWS-managed registry. Unlike self-managed registries, users can avail of this service model to speed-up delivery and avoid the worry associated with installing or scaling infrastructure to accommodate this. Users within EKS can then pull down images over HTTPS protocol with the conveniences of automatic encryption and access control. The obvious advantage here is if an organization uses a public registry to source its images, there's a very real possibility that those images are carrying some form of a vulnerability. With ECR, the access control of the registry is controlled by the IAM service. This means that you can configure specific users or EC2 instances to access ECR, and restrict unwanted users or instances from pulling/modifying images in the artifactory—as part of a zero-trust initiative.

Calico Cloud protects containerized workloads during build and runtime. It assesses and mitigates risk associated with container image vulnerabilities for cloud-native workloads by continuously scanning for vulnerabilities and misconfigurations before they are deployed to EKS clusters, and automatically blocking deployments that fail to meet security requirements. It quickly assesses the risk of deployed applications when new vulnerabilities are discovered by providing alerts in the Dynamic Service and Threat Graph. For runtime threats that manage to evade image scanning workflow, Calico Cloud also provides an extensive networking and security policy framework to identify and mitigate threats from workflows attempting to establish unusual connections within or outside of the EKS cluster—completing the end-to-end zero-trust initiative.

6. Keeping up with new attack vectors

Kubernetes pods running containers use unique IP addresses for connectivity. Attackers can exploit these IP addresses as gateways to launch attacks either internally or from external networks.

Regardless of whether the threat is known or unknown (zero-day incident), Calico Cloud allows organizations to monitor and troubleshoot security issues in real-time. Users can use GlobalThreatFeeds to visualize and enforce restrictions against known bad IPsets associated with malware, ransomware, or botnet feeds. In case of a breach or vulnerability, users get instant granular information on compromised services and can evaluate the blast radius via:

- Dynamic Service and Threat Graph
- Performance Hotspot Visualization
- Dynamic Packet Captures for workloads and namespaces
- Intrusion detection and prevention (IDS/IPS) to detect and mitigate Advanced Persistent Threats (APTs)

7. Adhering to regulatory compliance frameworks

Most enterprises are subject to corporate and/or regulatory compliance requirements. From an operational perspective, this may involve isolating workloads containing sensitive data, or restricting who is allowed to access specific resources. There may also be requirements to implement access control frameworks such as security zones (e.g., trusted, untrusted, and DMZ). Even more advanced controls are sometimes needed, like building a moat around PCI-DSS workloads or logging all HIPAA data transactions.

Furthermore, auditors need proof that you are enforcing these controls, but capturing the information required to show proof can be challenging, especially in a dynamic, distributed Kubernetes environment where workloads are ephemeral. For example, auditors will want to know what security controls are currently implemented, whether control changes be detected, and if compliance is verified for any given day and time.

Calico Cloud provides controls and capabilities to remain compliant with the above scenario and continuously monitors your cloud-native environment for compliance. It retains a daily history of your compliance status. Calico also includes predefined compliance report formats, as well as a resource for creating customized reports.

Conclusion

Containerized workload security in AWS and EKS is of the highest priority. Security and compliance are considered shared responsibilities when using a managed service like EKS. The cloud provider provides the security of the cloud platform, and the users on the platform build security within the cloud for their workloads. Tigera and EKS together provide complete security for cloud-native applications using containerized workloads and AWS. As an AWS user, you benefit from the ease of cluster deployment with a managed Kubernetes service while also having the option of benefitting from managed IAM and registry services that scale to meet the security requirements of the organization. Tigera provides additional security features to reduce the attack surface and detect known and unknown threats with automated mitigation of security threats and vulnerabilities with your build, deploy, and runtime. The shared responsibility model describes this as security of the cloud and security in the cloud. Best practices mentioned above allow organizations to manage better and secure their containerized workloads running in EKS, using native AWS services as well as Tigera's active CNAPP platform.

You can start your 1-month free trial of Calico Cloud today by subscribing via AWS Marketplace.