threat stack
Part of F5

Threat Stack Container Security Monitoring for AWS Fargate

AWS Fargate abstracts away the infrastructure that containers run on, but it also creates challenges for traditional security techniques. As infrastructure becomes increasingly atomic and short-lived, it's imperative to know as soon as possible about suspicious access to containers, or if there are behaviors present that could indicate an active threat. Use of Fargate in and of itself does not eliminate all security concerns.

Deep Visibility

While AWS provides a robust set of native access controls, if you need to deeply audit all activity within running Fargate tasks, you'll need more visibility into workloads. Threat Stack augments existing Fargate security controls by adding runtime observability at the application, process, and network levels.

As an AWS Advanced Technology Partner, we've used our experience to apply our cloud security expertise to AWS Fargate—which allows customers to take advantage of the benefits of serverless compute engines to run Amazon Elastic Container Services (ECS) and Amazon Elastic Kubernetes Services (EKS).

This paper aims to outline our approach to monitoring Fargate environments, describes general use-cases, and reviews available deployment options.

Full Stack, Even For Fargate

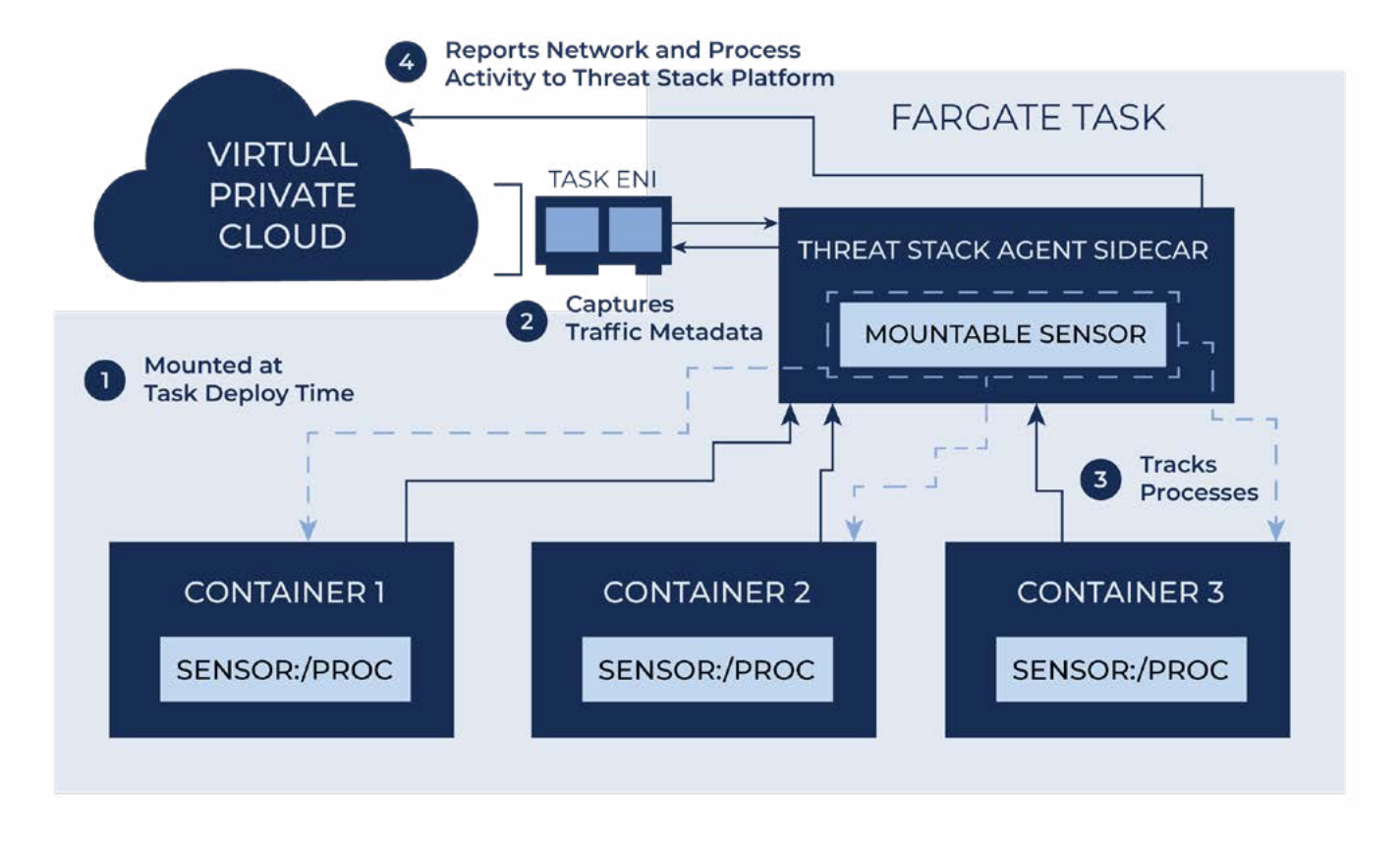
Fargate shifts the AWS shared responsibility model, but there's still interplay between layers that you need to account for from a security standpoint. Threat Stack secures multiple layers at runtime by monitoring four key aspects:

1. **Process activity inside Fargate containers and Kubernetes**
2. **Network flow data within, and external to, Fargate tasks**
3. **Application code running on top of Fargate**
4. **AWS CloudTrail logs alongside Fargate**

Threat Stack gives you the visibility you need as risky behaviors traverse these layers. Threat Stack Application Security Monitoring protects Node.js, Python, and Ruby code that Fargate supports. Threat Stack's AWS integration allows you to easily customize alerts for specific behaviors observed within Fargate-related CloudTrail events (e.g., Amazon ECS and EKS, Amazon VPC, AWS IAM). And with a monitoring agent specifically built for Fargate, Threat Stack provides network flow monitoring and process tracking within Fargate tasks running on ECS or EKS.

Fargate Agent Instrumentation

Our Fargate Agent uses the sidecar design pattern that's familiar to containerized microservices architectures. The Agent receives its own vCPU and memory and does not compete for the same resources as monitored workloads. This ensures predictable performance as it monitors network and process activity within tasks.



Netflow Monitoring

For the network monitoring component, the Threat Stack Fargate Agent captures traffic metadata off of the task's shared ENI. The Agent observes all packets and will track unique TCP netflows until completion or timeout. Since it runs as a sidecar within the task itself, Threat Stack can also surface intra-task traffic between individual containers, capturing metadata that does not appear in VPC flow logs. The Agent packages this data and ships it off to the Threat Stack platform for alerting, search indexing, and downstream analytics.

Process Monitoring

For the process monitoring piece, the Agent hosts a binary that is mounted into the tasks's containers at deploy time. As containers run, it scans the /proc folder and communicates this metadata back to the Agent to track process activity. And since the binary mounts dynamically as part of the task provisioning process, you won't need to rebuild existing containers or Kubernetes in your image repository.

Security Monitoring Use-Cases

Threat Stack's approach to full-stack security observability into AWS Fargate allows customers to address the following monitoring use-cases.

Are Containers Making Unexpected Network Connections?

For the network monitoring component, the Threat Stack Fargate Agent captures traffic metadata off of the task's shared ENI. The Agent observes all packets and will track unique TCP netflows until completion or timeout. Since it runs as a sidecar within the task itself, Threat Stack can also surface intra-task traffic between individual containers or Kubernetes, capturing metadata that does not appear in VPC flow logs. The Agent packages this data and ships it off to the Threat Stack platform for alerting, search indexing, and downstream analytics.

Threat Stack provides default detections for Fargate, including:

- Interactive Sessions
- SSHD Binaries
- Data Exfiltration Attempts
- Unexpected Network Connections

Are There Unexpected Processes Running In Fargate Containers?

Know when there are unexpected processes executing within Fargate containers, so you can investigate for signs of risky activity. Since tasks should be immutable and processes are predefined, it's an instant red flag to observe unique new processes in Fargate. Threat Stack alerts are easily customizable, enabling customers to extend process-tracking rule logic with their own executable names and computed hashes.

Are There Unexpected Logins To Fargate Containers?

Know when untrusted entities access running containers or Kubernetes. There's rarely a good reason for remote logins to a container, making it a serious issue that could signify a bad configuration, a foolish engineer, or the initial stages of an attack. Threat Stack provides out-of-the-box detections for SSH activity within Fargate, so you can investigate ASAP.

Deployment

Threat Stack's workflows for alert generation, triage, deposition, and integration all remain the same when monitoring within Fargate environments. Threat Stack supports monitoring of the Fargate launch type on both ECS and EKS.

Deploy the Threat Stack Fargate Agent by adding it to your Fargate task definitions. The Agent runs as a sidecar container as part of each task instantiation. (We recommend storing Agent deployment keys via AWS Systems Manager Parameter Store.)

You Can't Secure What You Can't See

With Fargate, AWS assumes more responsibility for infrastructure security, but there's still attack surface to account for. Threat Stack's instrumentation is easy to deploy so there are no security blindspots.

Please contact your Threat Stack customer success manager or sales representative to schedule a demo: https://www.threatstack.com/#threat-stack-demo.