**1. Secure the container host.**

* Containers should be hosted in a container-focused OS. The overall attack surface will be reduced if services that aren’t required to host your container workloads are removed.
* Monitoring tools should be added to provide visibility on your hosts’ health.
* A strong set of security controls, like the ones available in the [Trend Micro™ Deep Security™](https://www.trendmicro.com/en_us/business/products/hybrid-cloud/deep-security.html) solution, should be used in order to secure container host systems. The integrity of these systems, which run all container workloads, will be guaranteed if they are properly secured.

**2. Secure the networking environment.**

* Take advantage of controls like an intrusion prevention system (IPS) and web filtering for traffic moving from north to south, and to and from the internet, in order to stop attacks and filter malicious content.
* To monitor inter-container traffic, an IPS should also be deployed. Internal traffic monitoring is critical to your defense since attackers that have already gained a foothold in a network can quickly move laterally to expand their reach.

**3. Secure your management stack.**

* The container registry should be properly secured and monitored.
* The Kubernetes installation should be locked down. In addition, features like Pod and network policies should be taken advantage of to enforce your organization’s security and development standards.
* Security tools such as the [Trend Micro Deep Security Smart Check](https://www.trendmicro.com/en_ph/business/products/hybrid-cloud/smart-check-image-scanning.html) should be used to scan and validate the configuration of each container as it is added to the container registry. This ensures that only containers complying with the team’s development and security standards are being deployed.

**4. Build on a secure foundation.**

* Review and watch project teams’ communications regarding dependencies used in applications. When a piece of software is patched, those changes need to be integrated into the application to reduce risks.
* Make sure that containers are free from malware or known vulnerabilities and do not have exposed secrets by using a container image scanner such as Deep Security Smart Check. This will also sweep for custom indicators of compromise (IoCs). By using a container image scanner, you can mitigate risks before further development or deployment to production.

**5. Secure your build pipeline.**

* Developer workstations should be installed with strong endpoint controls, ones that can be found in the [Trend Micro Apex One™](https://www.trendmicro.com/en_ph/business/products/user-protection/sps/endpoint.html) solution. Doing this helps prevent malware, visits to malicious websites, and other security challenges and container threats.
* An access control scheme, one that is thorough and consistent, must be implemented. A critical step to safeguarding the integrity of your pipeline is ensuring that only authorized users can access code repositories, integrate branches, and trigger builds that get pushed to production.
* The servers running the said tools should also be secured. Using security tools such as Trend Micro Deep Security can provide a set of strong controls with minimal overhead, helping your team meet security goals and standards.

**6. Secure your application.**

* The code should follow best practices in order to improve its quality. After all, simple mistakes or poor design choices can result in many security vulnerabilities. Investing time and effort in ensuring the quality of your code always pays security dividends.
* Runtime self-protection controls should be used to help connect the dots between security vulnerabilities and issues in specific lines of code. Doing this closes the gap during root-cause analysis and allows better security outcomes to be achieved.

**A last word on securing containers**

The use of container technology increases the speed and efficiency of the development process while maintaining consistency across the board. However, [threats](https://blog.trendmicro.com/trendlabs-security-intelligence/aesddos-botnet-malware-infiltrates-containers-via-exposed-docker-apis/) to containers have started to emerge recently, and it is imperative that containers be properly secured to thwart potential risks to organizations. Fortunately, despite the many aspects of the development process and supporting infrastructure that developers need to take into account, the steps mentioned above can help secure containers with efficacy.