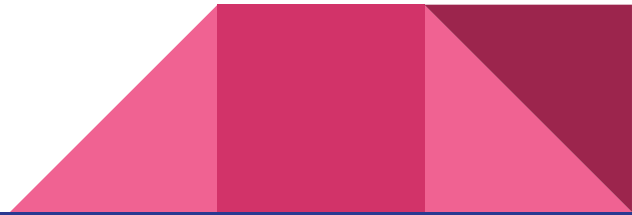
First Look: Container Security

How containers enable better AppSec

Problem

Application Security is hard!

- Securing delivery pipelines from code to production is complex
  - Harder when multiple dev languages or artifact formats are involved
- Enforcing runtime constraints and policy requires host specific configuration
  - Ex: SELinux/AppArmor profiles difficult to securely distribute and kept up to date



Solution

Containers simplify!

Let’s see how using containers can simplify AppSec

Securing the supply-chain

Simplify artifact management

- CI/CD pipelines tooled to deliver a *single* kind of app artifact: container images
- Images stored in centralized *registries* with simple access and retention policies

Enforce artifact signatures

- Container images are signed during the build process and metadata is securely stored
- Image signatures are enforced by container runtime -- no custom code or functionality required

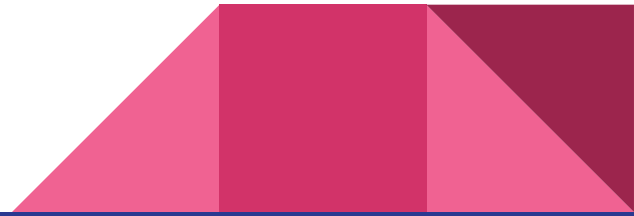
Manage application dependencies

- Base container images used by developers can be verified, scanned and versioned
- Updates to app dependencies can be made to a single image and propagate downstream automatically

Runtime protections

Lightweight process isolation

- Kernel *namespaces* provide a secure but inexpensive means to further isolate individual app process
- We can deploy apps densely without relying on a hypervisor
- Memory, network, and filesystem boundaries strongly enforced

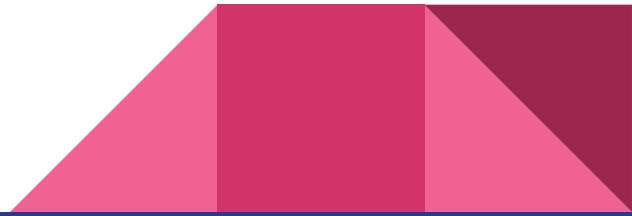


Simplified private networking

- Container runtime enables push-button configuration of *bridged* networking between apps on the same machine
  - Allows for safe *sidecar* patterns and *proxies*
- Networking plugins extend private networking via an encrypted *service-mesh* across machines

Filesystem security

- Apps are restricted to their own private storage *volumes* by default
- Access to the host filesystem (or other app volumes) must be given explicitly
  - This access can be read-only and enforced by kernel
- No host filesystem user/group configuration required



Resources