CONTINO

DevSecOps Container Pipeline

James Strong - Cloud Native Director
- AWS APN Ambassador
- Weightlifter
- Adjectives!

James Strong

Cloud Native Director

@strongjz

in strongjz

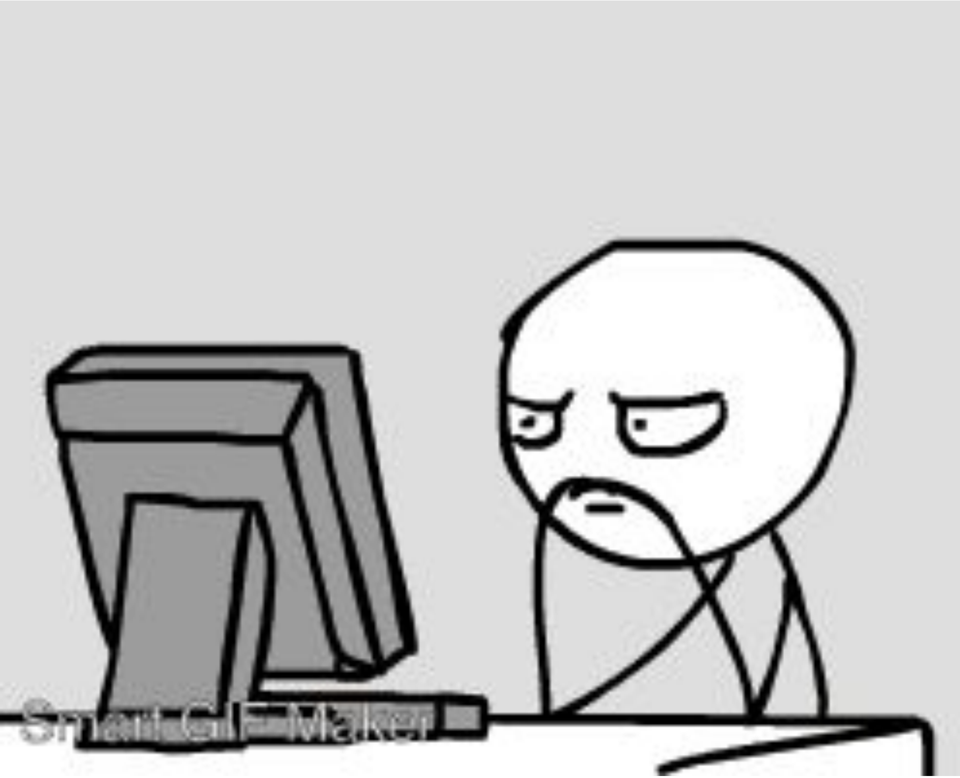
Agenda

- Development
- Pipeline
- Container Security
- Runtime

CONTAINERS AND DEVSECOPS
SO HOT RIGHT NOW

Development

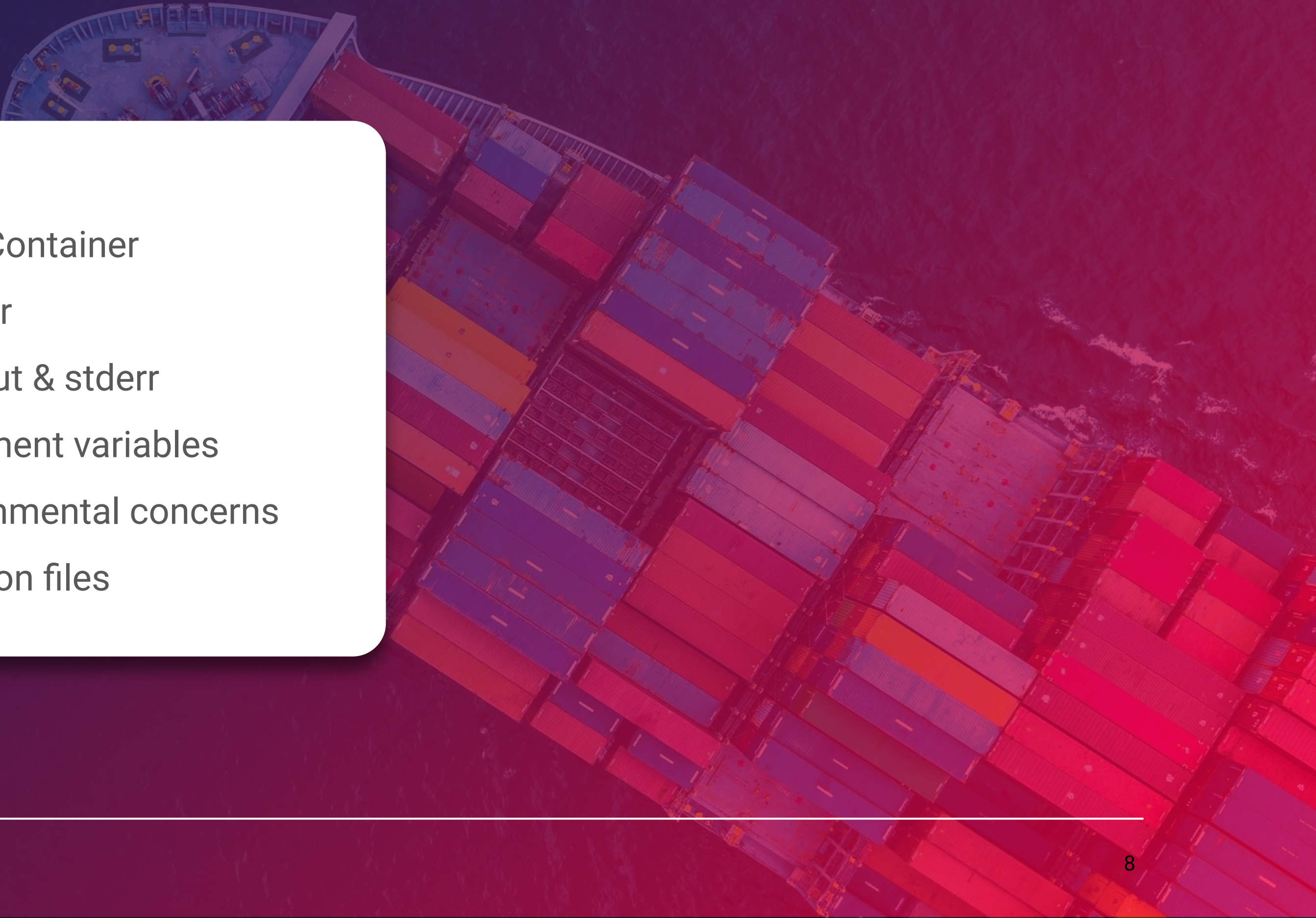
Development



- Commit Signing
- Style Guide Linting
- OWASP Dependency Checks
- 3 Musketeers
- Pre Commit Hooks

Containers

Containers



- Minimal OS
- One Process per Container
- Run with local user
- Write logs to stdout & stderr
- Leverage environment variables
- Separating environmental concerns
- Mount configuration files

Minimal Container OS

Alpine

Debian Slim

Custom

Secure Containers

- DISABLE ROOT
- Least Privilege
- Run time Protections
- Pipeline builds
- Network policies
- DISABLE ROOT

Versioning

- Invest in Strategy
- Containers follow build Versions of Software
- Metadata

Latest Tag

- Avoid using :latest tags
    - Unable to control
    - Unknown updates
- Versions the way to go
- Container digest

Base Container

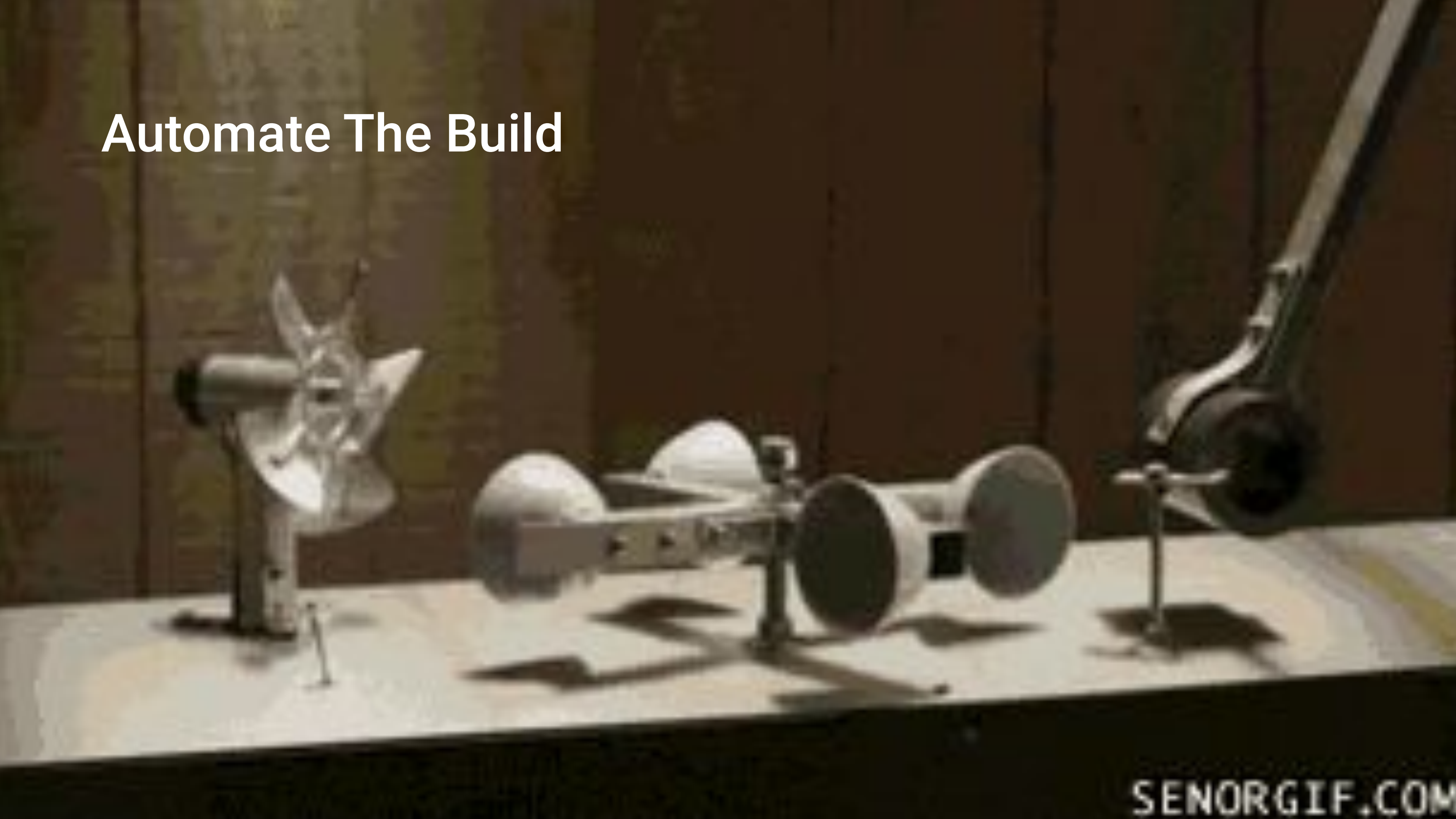
- Reduce Build Times
- Scratch Container

Testing

- Early & Often
- Track Differences
- Points of Measure

Pipeline

Automate The Build



Build Pipeline

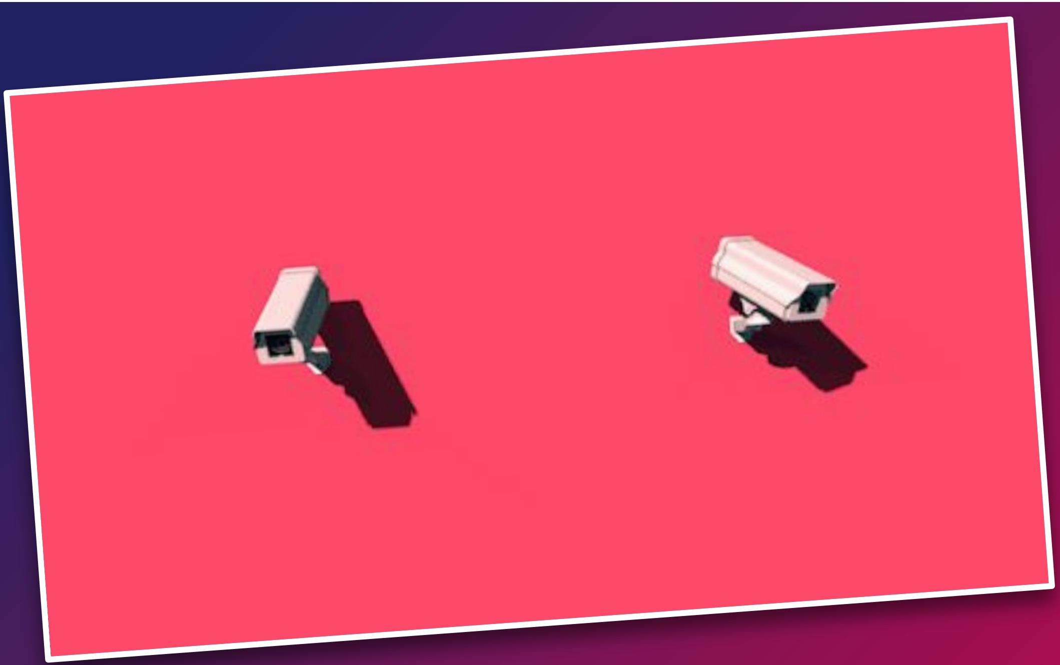
- Automating your build pipelines
  - Visibility
  - Troubleshooting
  - Defect Remediation
- Build small autonomous pieces

CONTINO

CI/CD

1. Maintain a code repository
2. Automate the build
3. Keep the build fast
4. Make the build self-testing
5. Commit early, commit often
6. Every commit gets built
7. Everyone can see the results of the build
8. Automate the deployment

Secure Pipelines



- Signed Images
- Verify Trusted Images
- Kickoff Security Assessment

Runtime

Minimal Host OS

Redhat
CoreOS

RancherOS

Ubuntu Core OS

AWS BottleRocket

Container Runtime Security

- Pipeline
- Run time
- Registry

Network Isolation

- Per Namespace
- Default Deny

Operating Environment

- Dev, Test, Prod, Others
- Shared Kernel
- Kube-bench

CONTINO

Network Policies

- Per Namespace
- Per application
- Examples

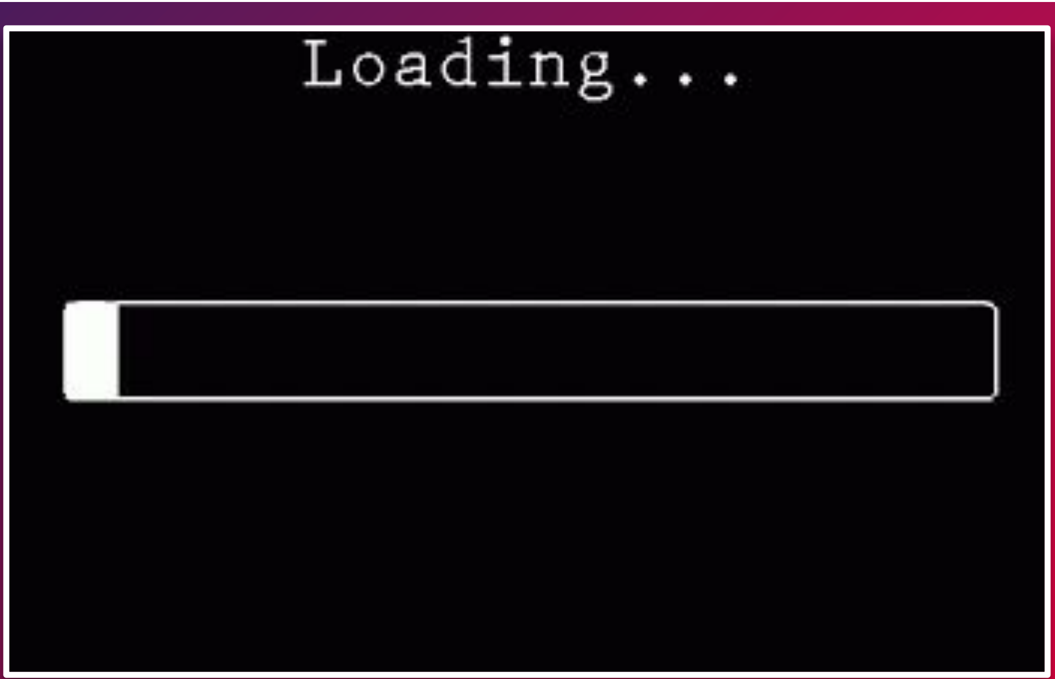
CI/CD + Security

1. Maintain a code repository
2. Automate the build
3. Keep the build fast
4. Make the build self-testing
5. Commit early, commit often
6. Every commit to the mainline gets built
7. Everyone can see the results of the build
8. Automate the deployment

1. Immutable artifacts
2. Static Code Analysis
3. CVE Scanning
4. Least Privileged
5. Network Isolation
6. Run Time protection
7. Signed Commits
8. Signed Images

The Demo

- Github Repo
- AWS ECR - Image Scanning - CVE
- AWS CodePipeline/Build - CI/CD
- AWS ECR - Immutable Tags
- Run Time Security - Falco
- Logging - FireLens
- Alerting - Cloudwatch



Enough Slides lets Demo this