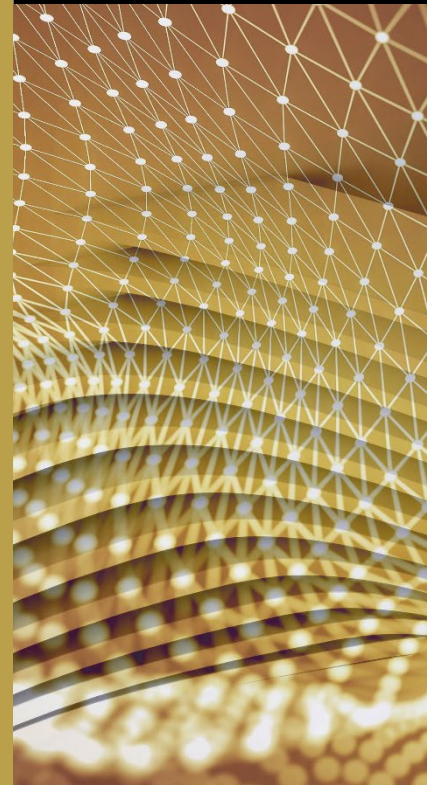
WASHINGTON D.C. | 2024

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

SEPTEMBER 18, 2024

David Shepard, Senior Software Engineer
Morgan Farrah, Assistant Technical Engagement Lead
Maxfield Kassel, DevSecOps Intern

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Document Markings

The following markings MUST be included in work product when attached to this form and when it is published.

For purposes of double anonymous peer review, markings may be temporarily omitted to ensure anonymity of the author(s).

Carnegie Mellon University 2024

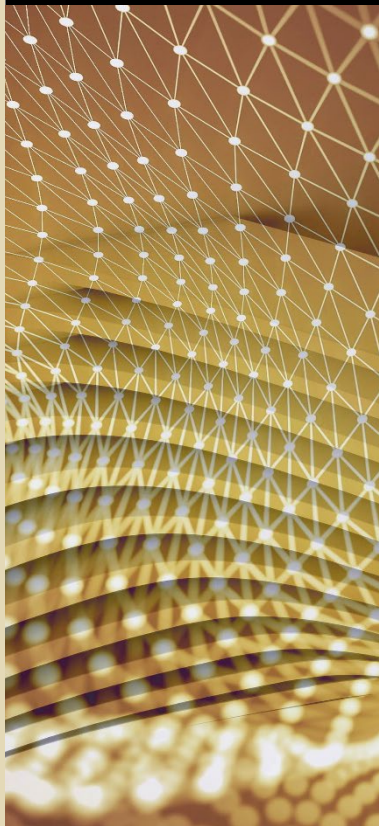
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM24-1185

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

About Polar…

Polar – OSS Observability Framework

A DevSecOps-first Approach to Software Development

Sharing our Methodology as well as our Source Code

The Details – Nix-based Containers

Conclusion

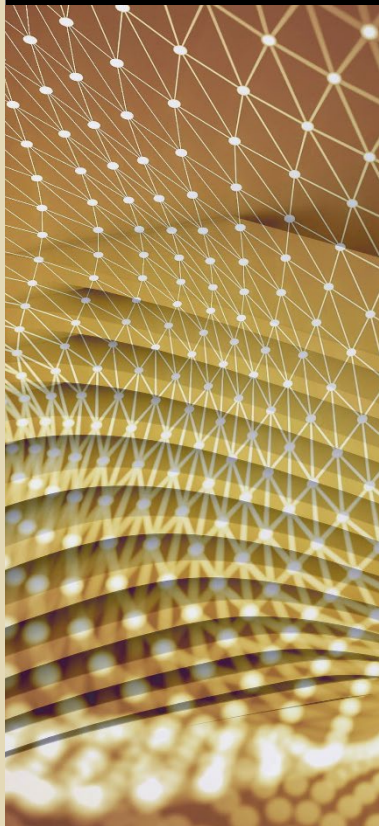
[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Polar Improves DevSecOps Awareness and Enables Informed Decision Making

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

About Polar…

Polar – OSS Observability Framework

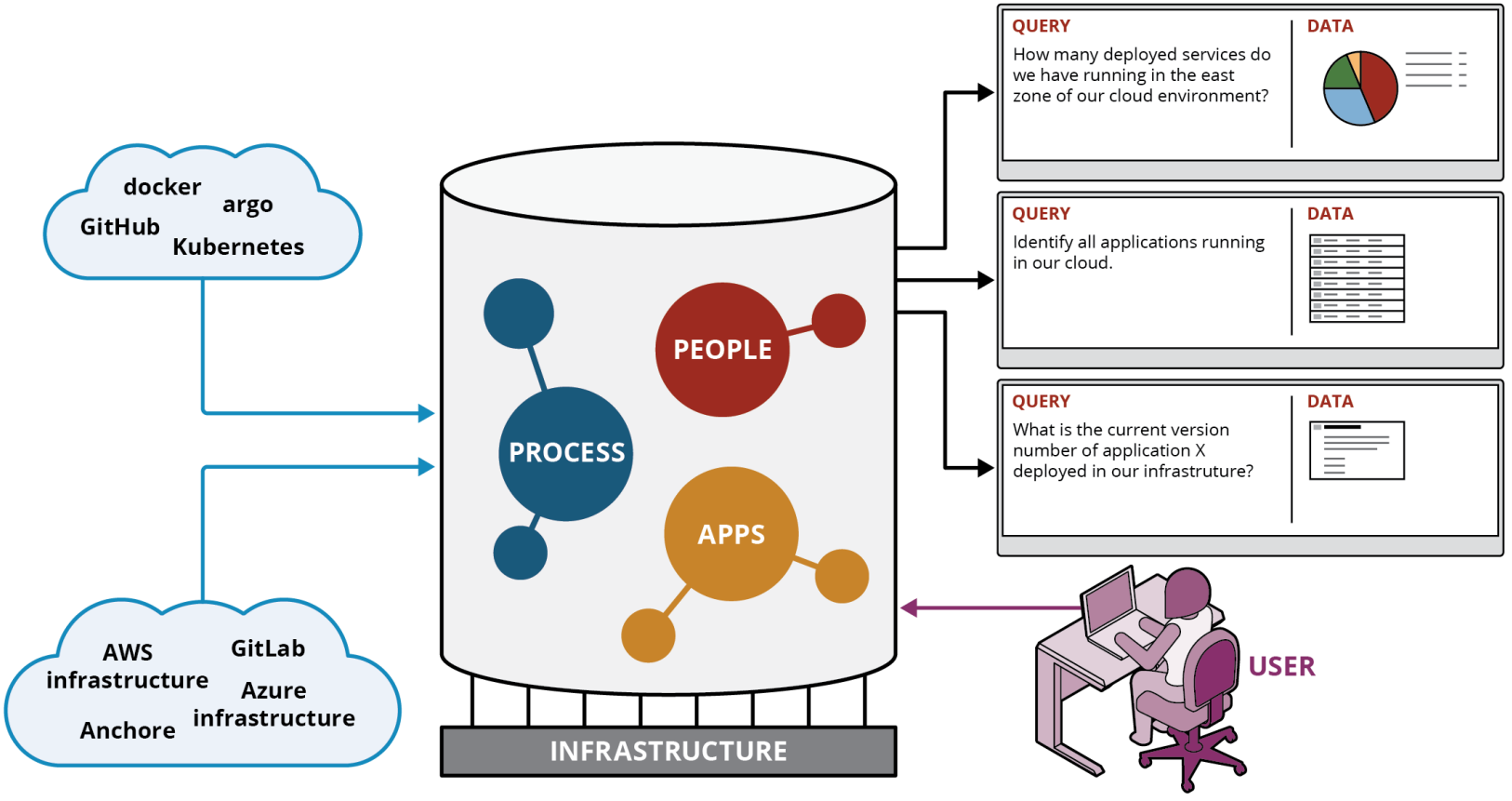
A DevSecOps-first Approach to Software Development

Sharing our Methodology as well as our Source Code

The Details – Nix-based Containers

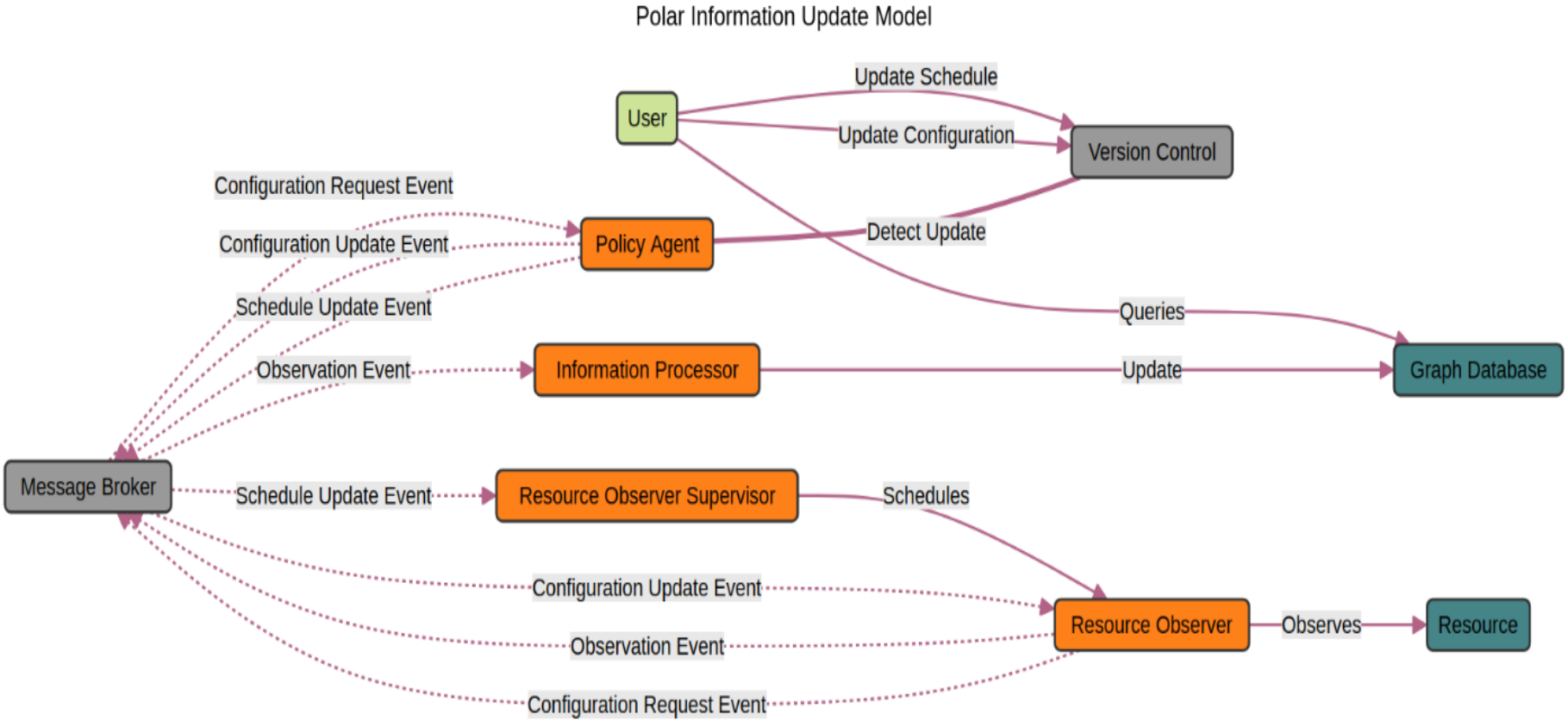
Conclusion

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.



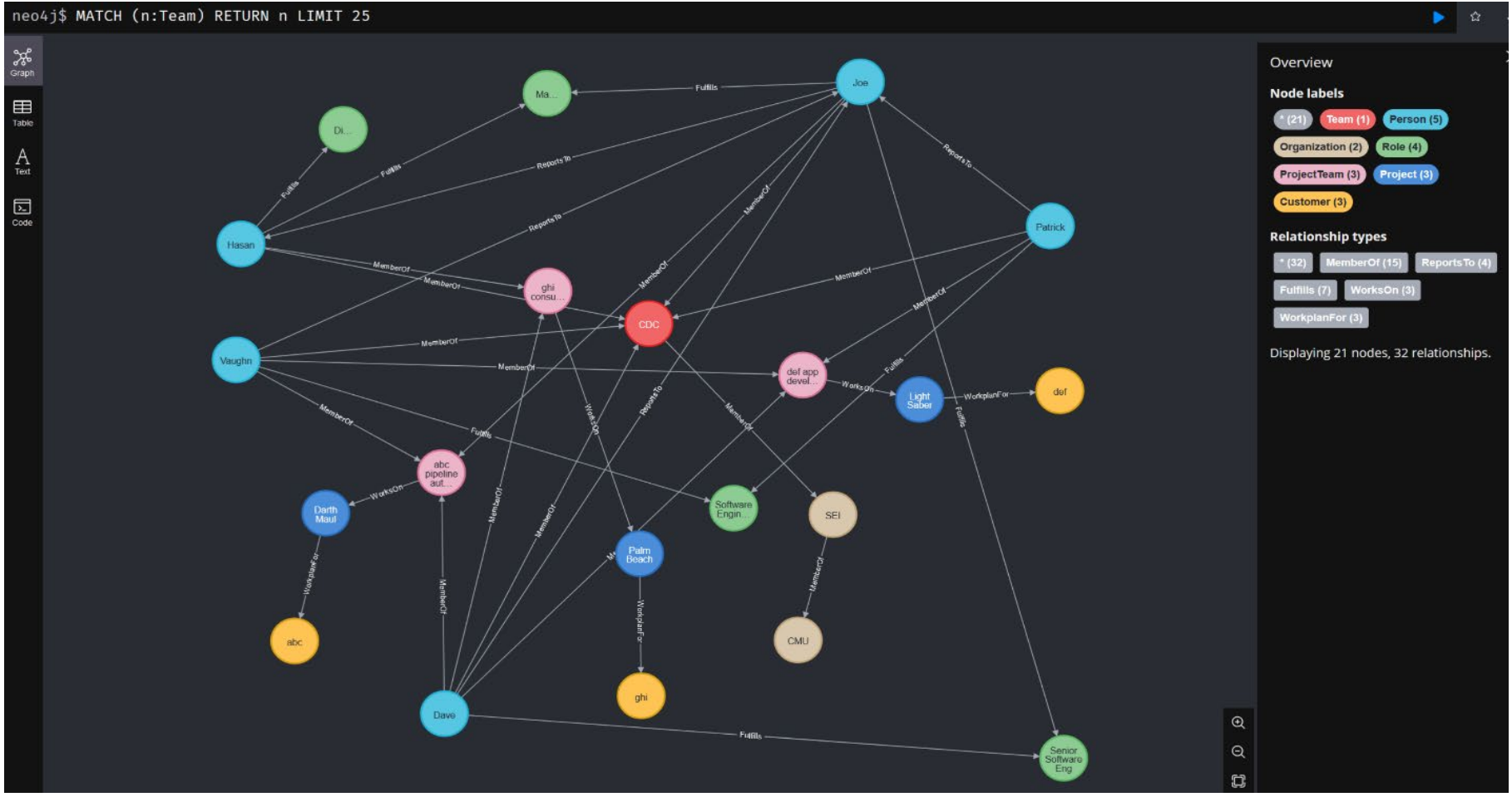
© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.



© 2024 Carnegie Mellon University

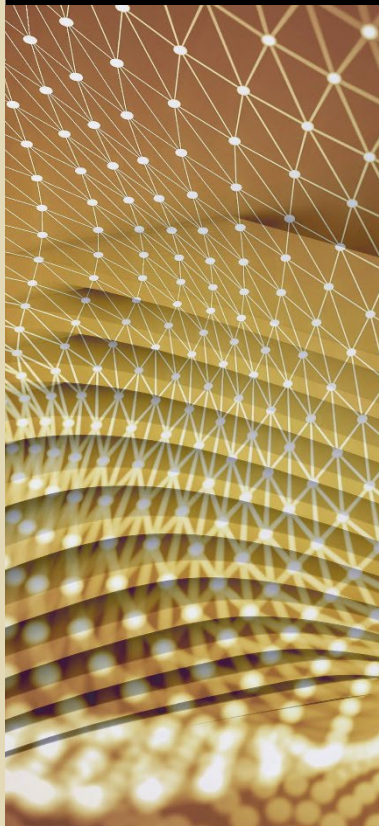
[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.



© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

About Polar…

Polar – OSS Observability Framework

A DevSecOps-first Approach to Software Development

Sharing our Methodology as well as our Source Code

The Details – Nix-based Containers

Conclusion

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

It’s Not Just a Process We Follow, **It’s Part of Who We Are**

- We're a rather small team... We wear a lot of "hats"
- We help others with their DevSecOps challenges. It's important to us that we're doing things:
  - **"The right way"**
  - **"Eating our own dog food"**
  - **"Whatever other metaphor you prefer"**
- Writing code first and then adding in DevSecOps concepts doesn't give you the benefits of DevSecOps
- Writing code without a process guarantees your software will have all of the well-known shortcomings of modern software

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Security is Part of Everyone's Process Because…
It Has To Be

Security affects our choice of:

- Work environment
- Languages used
- Tools for development
- Software dependencies included
- Testing performed
- Packaging method
- Operating environment

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

But I Only Have X Time to Deliver Y Value...

"We don't have the time to DevSecOps"

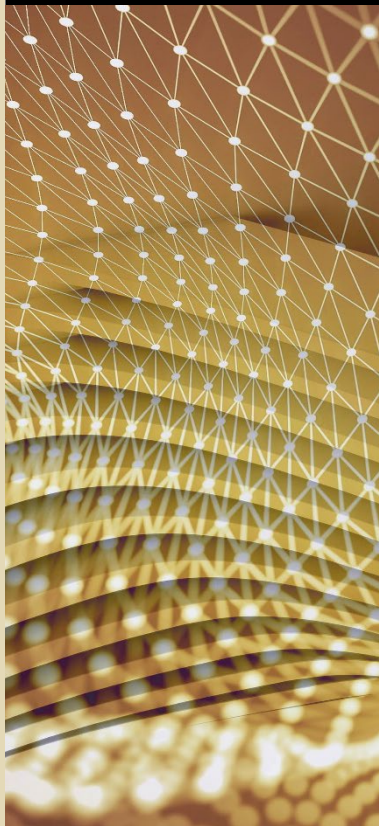
"We don't have the budget to DevSecOps"

DevSecOps "machinery" is always going to be "best effort", based on some pre-determined set of time and budget that works for your projects. Our goal is to make following some amount of process feasible/appealing to everyone, scalable for anyone, and to demonstrate how we're doing that on our own projects.

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

About Polar…

Polar – OSS Observability Framework

A DevSecOps-first Approach to Software Development

Sharing our Methodology as well as our Source Code

The Details – Nix-based Containers

Conclusion

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

A Typical Process for Setting up a Software Project

1. Ensure you are using a "supported" operating system
2. Install language runtimes, compilers, interpreters, SDKs, and 42 other things. Repeat twice because you missed something
3. Spend 2 hours figuring out which version(s) of Node, Python, or Java is required
4. Find out that 3 libraries you installed aren't compatible with each other
5. Reach out to the developer who wrote it and be told that you missed step 47
6. Reset your computer and try again tomorrow
7. Eventually, document your "process" for others to follow

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

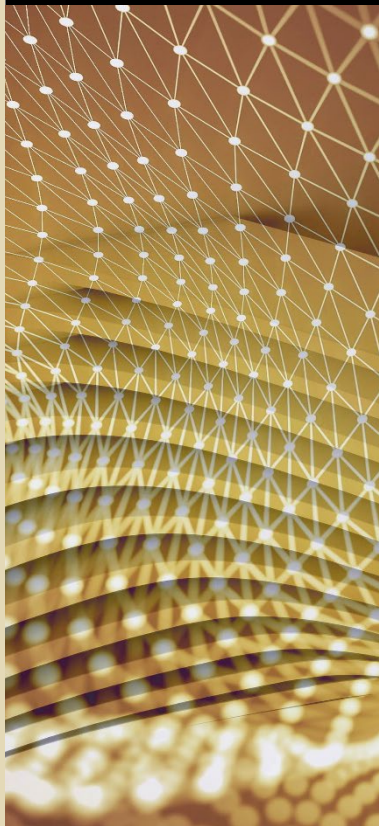
How We Want to Work

- When wearing our Developer (hard) hat...
  - Work in a safe environment
  - Ensure developers can run tests before checking code in
  - Ensure environment parity
- When wearing our Security hat...
  - Know the baseline for dependencies and tools has been validated
  - Know that basic security validation tools are available to all
- When wearing our Operations hat...
  - Know the software is deployed exactly as it is built and tested
  - Know there is full traceability of deployed artifacts and dependencies

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

About Polar…

Polar – OSS Observability Framework

A DevSecOps-first Approach to Software Development

Sharing our Methodology as well as our Source Code

The Details – Nix-based Containers

Conclusion

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

What is “Right Sized” DevSecOps?

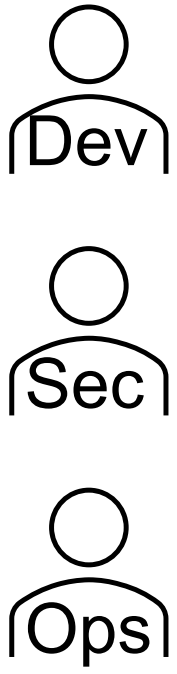
- **Accessible for everyone**
- Works with pipelines
- Repeatable even on different operating systems
- Easy and straightforward to "do the right thing"
- Isolated to ensure we won’t interfere with other preinstalled software

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

What is “Right Sized” DevSecOps?

- **Accessible for everyone**
- Works with pipelines
- Repeatable even on different operating systems
- Easy and straightforward to "do the right thing"
- Isolated to ensure we won’t interfere with other preinstalled software



© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

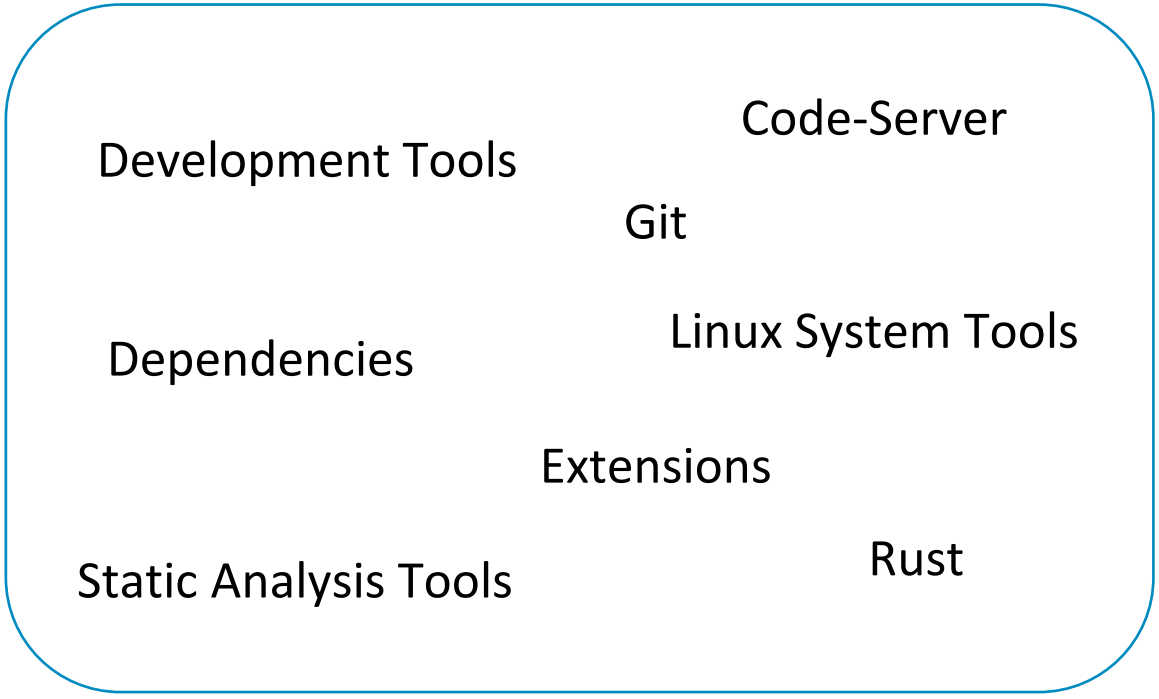
How a Development Container Helps With DevSecOps

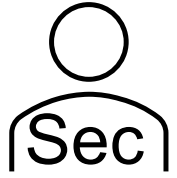
- Ability for anyone to follow proper DevSecOps principles
- Shared environment with all tools, dependencies and software needed for development and testing
- Creates an isolated, secure environment for development and deployment
- Simplifies deployment, ensuring code runs reliably in the same environment across different stages

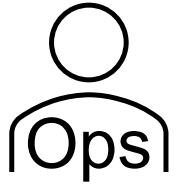
© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Inside Our Container





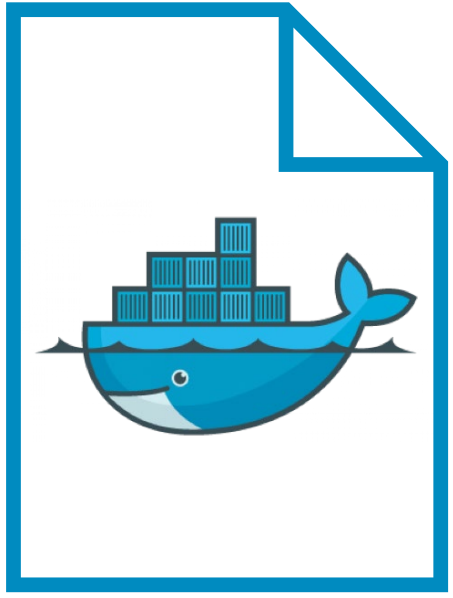


© 2024 Carnegie Mellon University

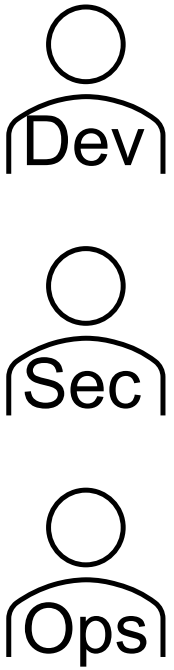
[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Using DockerFiles to Create Development Containers

- Most popular way of creating docker images
- Text-based file
- Simple process to create
- DockerFiles are versioned
- Usually are not reproducible...



DockerFile



© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

What is Nix?

- Package manager, operating system, shell environment, and much more
- Declarative
- Reproducible
- Flexible
- Cross-platform

Nix Logo

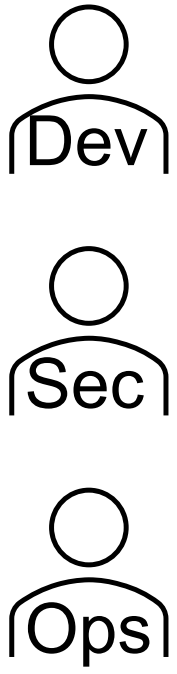
© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Building Development Containers Using Nix

Nix helps us to:

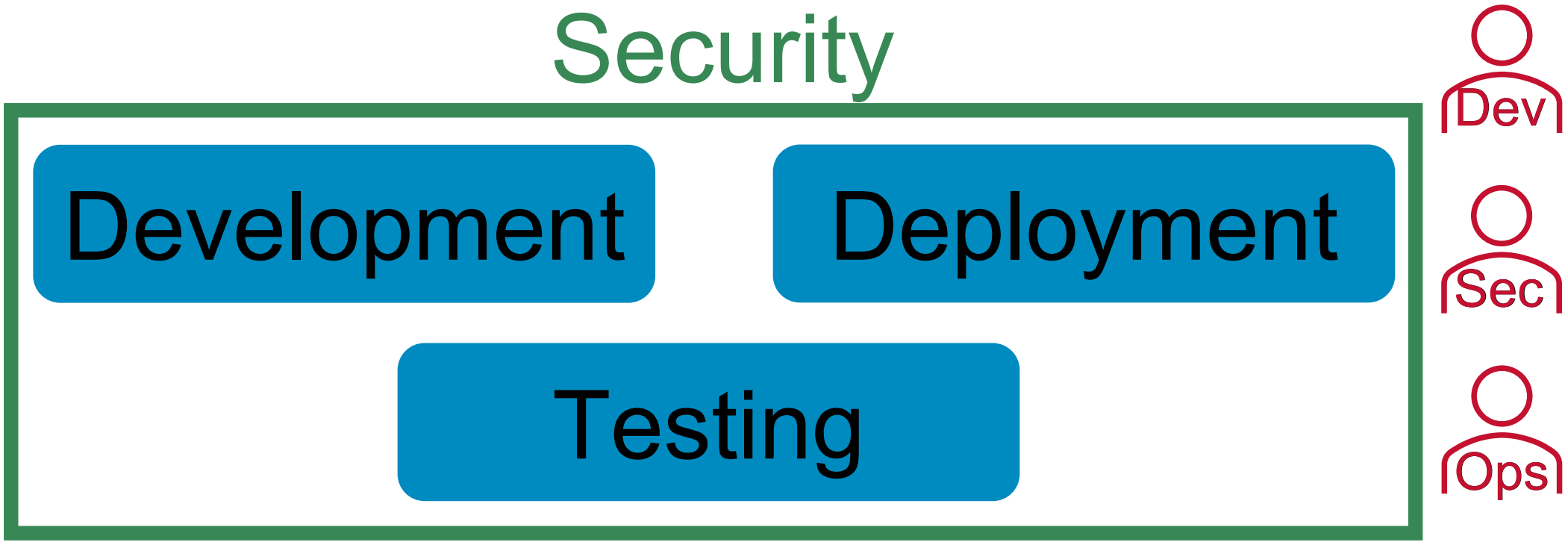
- Specify all software and files to be included
- Version-lock all software and libraries, by default
- Choose when a build recipe is permitted to change when/if needed
- Create images compatible with most orchestration software, such as Docker, Podman, Kubernetes, etc.



© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

How are we Using the Container?



© 2024 Carnegie Mellon University

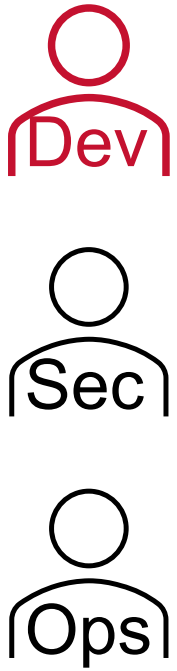
[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Development

Local install: Docker, Podman, etc. installed on your machine

Remote install: A remote machine with Docker, Podman, Kubernetes, etc and web browser that can connect to that machine

Can develop remotely, with no need to install any new tools, everything is already inside the container

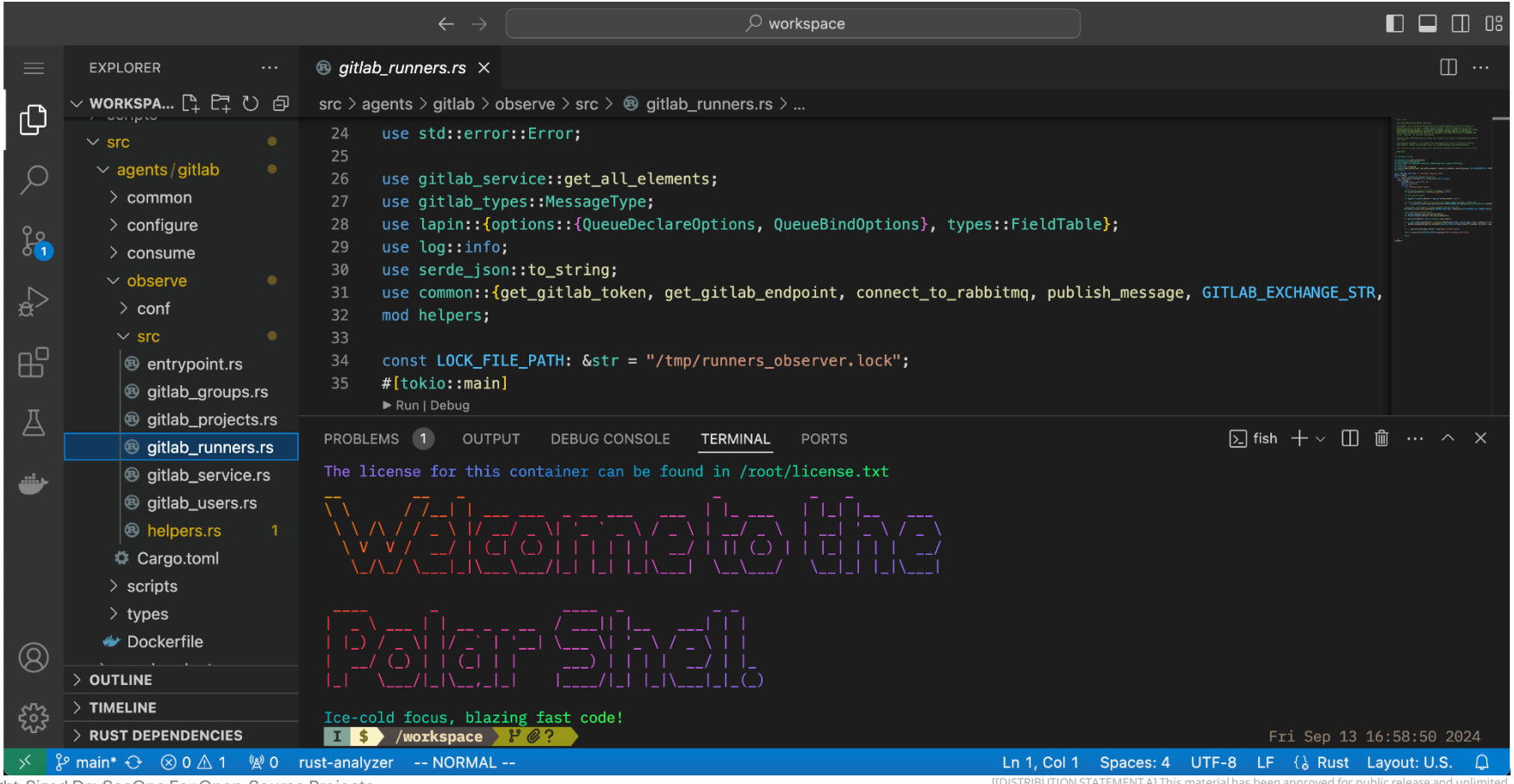


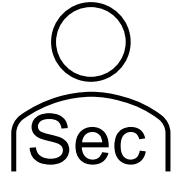
© 2024 Carnegie Mellon University

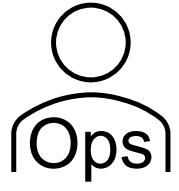
[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Development Inside the Container

Carnegie Mellon University
Software Engineering Institute

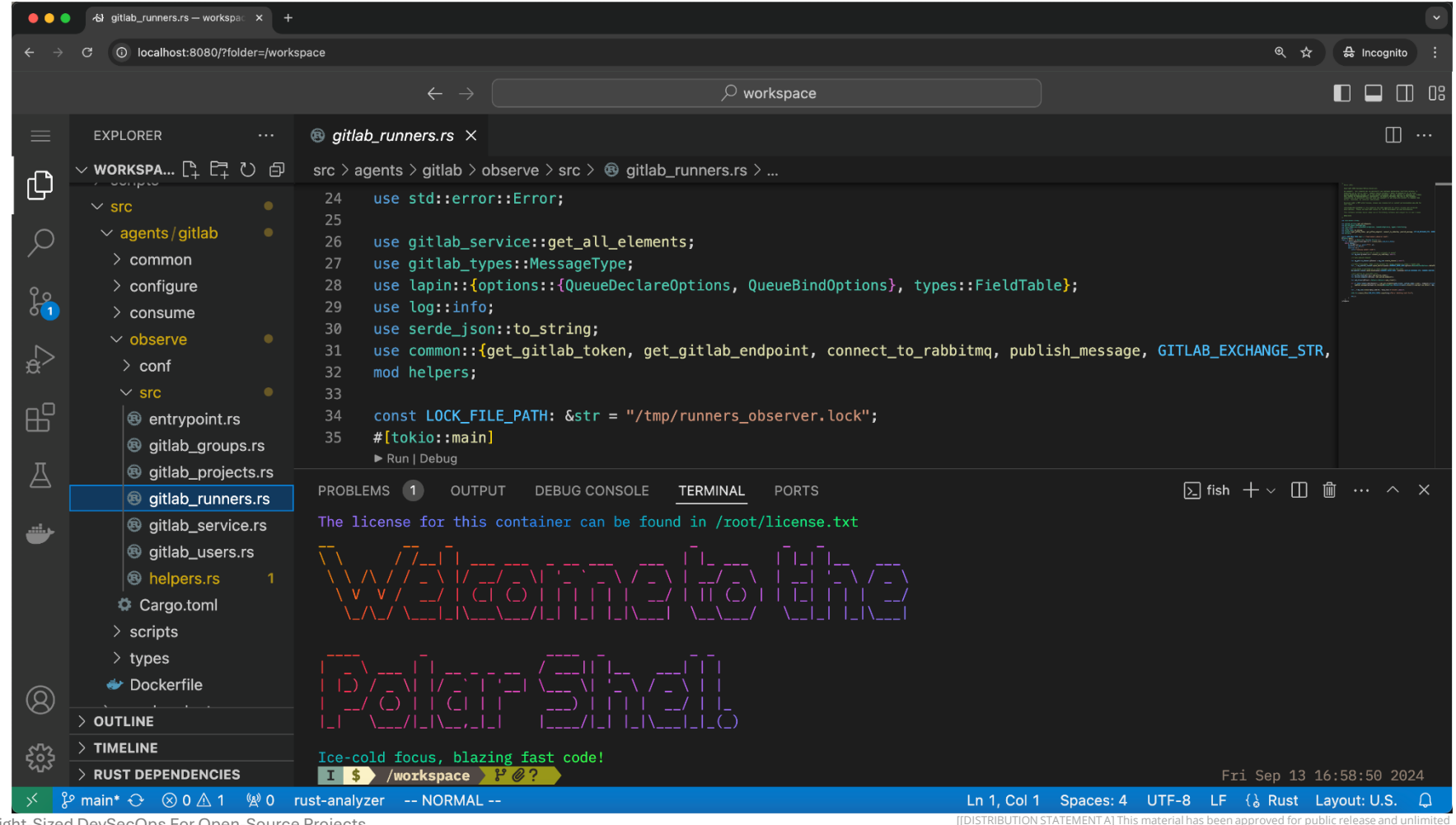






© 2024 Carnegie Mellon University
[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Development Inside the Container



Carnegie Mellon University Software Engineering Institute

Dev

Sec

Ops

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Testing

For Rust-based development, the dev container has Rust-related testing tools:

- Cargo Spellcheck: Finds spelling errors.
- Cargo Deny: Checks dependencies for licensing and other issues
- Cargo Clippy: Lints code for improvements.
- Cargo Udeps: Identifies unused dependencies.
- Cargo Semver Checks: Verifies semantic versioning.
- Cargo Bloat: Analyzes binary size.
- Unused Features: Analyzes unused features.
- Nosey Parker: Scans for sensitive information.

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Testing

We provide basic automation of these tools to:

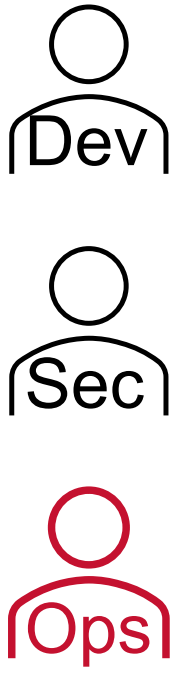
- Run them quickly
- Retrieve log output
- Include scan results in artifact/product container

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Deploying Polar

- We have automation to convert the dev container into a minimized release container with just the executables and other artifacts:
  - Binaries
  - Scan results
  - SBOMs
- Easy to hook into a CI/CD pipeline (Jenkins, GitLab, etc.)
- Incredibly light weight - 50x smaller than the development container
- No worry about OS or program compatibility



© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

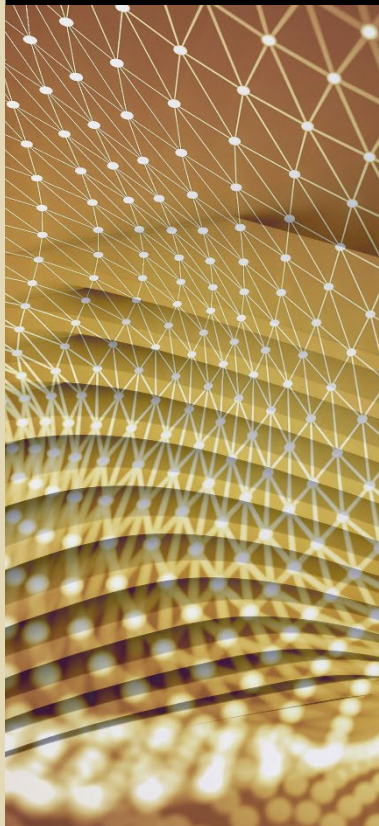
Security Properties afforded by the Container

- Isolation – Programs running in the container are separated from other processes on the computer, as well as the file system
- Auditable – From a Nix Flake you are able to determine what software is running inside the container and its version
- **No external tools** – All the software, tools, libraries are built-in, so nothing needs to be downloaded when you are in the development container
- No guessing about what tool versions a developer was using when they made a change
- No ad-hoc package management, controlled by individual software suites
- Security-hardened as part of the build process (FIPS compliance, extraneous features removed, etc.)

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Carnegie Mellon University
Software Engineering Institute



Right-Sized DevSecOps For Open-Source Projects

About Polar…

Polar – OSS Observability Framework

A DevSecOps-first Approach to Software Development

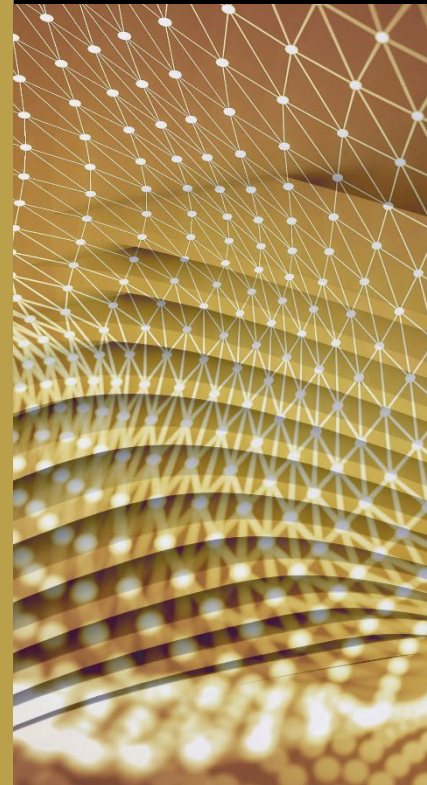
Sharing our Methodology as well as our Source Code

The Details – Nix-based Containers

Conclusion

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Carnegie Mellon University
Software Engineering Institute



Questions?

SEPTEMBER 12, 2024

David Shepard, Senior Software Engineer
Morgan Farrah, Assistant Technical Engagement Lead
Maxfield Kassel, DevSecOps Intern

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Contact Us!

Carnegie Mellon University
Software Engineering Institute

David Shepard
Senior Software Engineer

Telephone: +1 724.833.2021
Email: djshepard@sei.cmu.edu

Maxfield Kassel
DevSecOps Intern

Email: mkassel@sei.cmu.edu

Morgan Farrah
Assistant Technical Engagement Lead

Email: mefarrah@sei.cmu.edu

© 2024 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.