


Accelerate

Your Modern Apps To Production

Robert Jensen
Lead Systems Engineer @Vmware

 @rhjensen / jensenr@vmware.com

11/5/2023



Why You Want To Accelerate Your Apps To Production ?

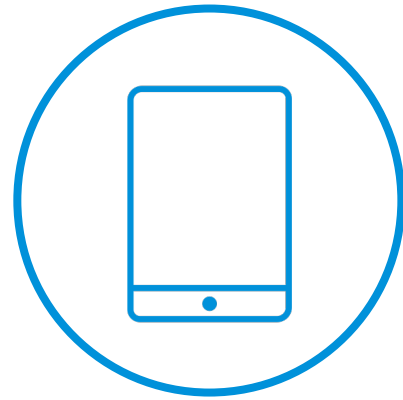


Apps are Everywhere

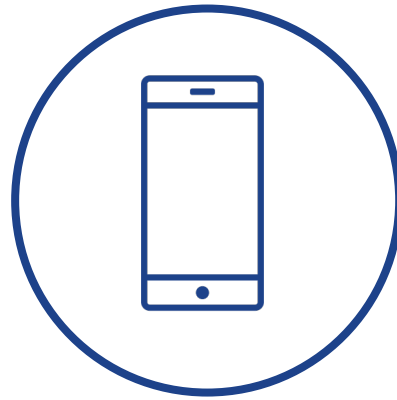
The “Why”



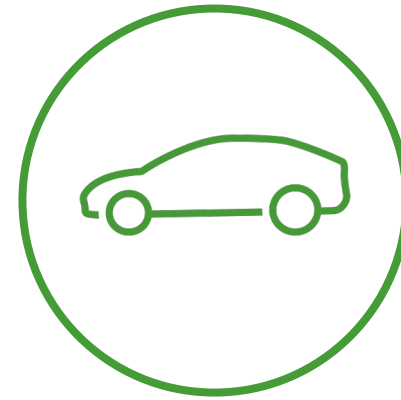
Website



Tablet



Phone



Car

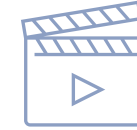


House

The Why

Every Company ~~is~~ a Software Company

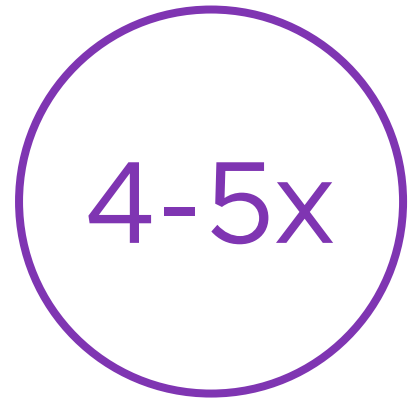
needs to be a custom



Good Software Drives Business Outcomes

Top performers compared to peers

The Why



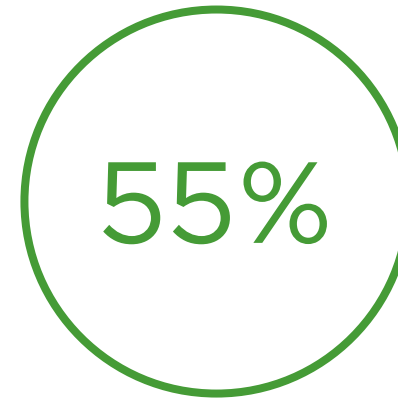
Increased Revenue Growth



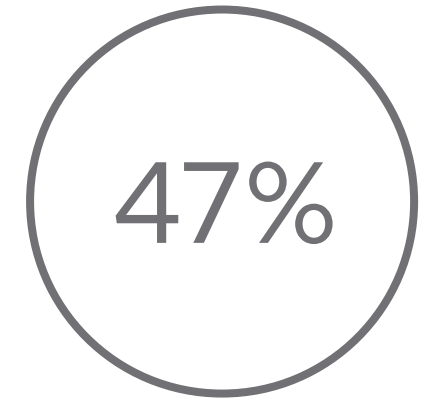
Higher Shareholder Returns



Higher Operating Margins



Increased Innovation



Higher Retention Rates

Not just for the Netflix and Spotifys of the world

Dinero.dk

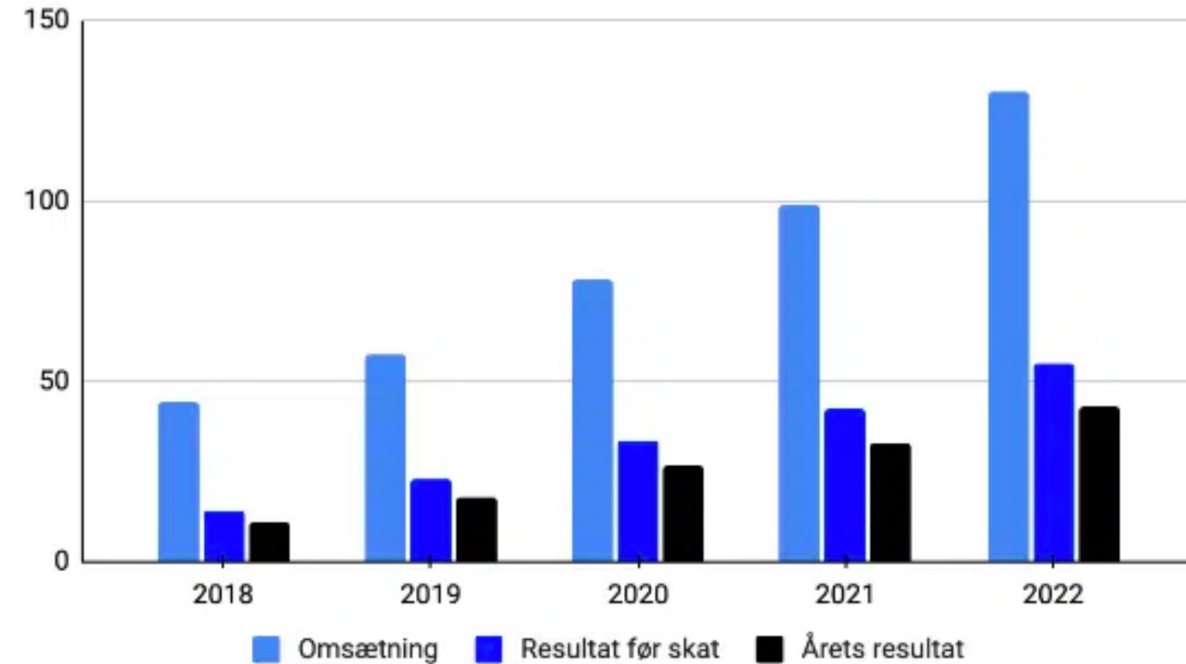


Martin Buch Thorborg
@thorborg

Hvor ofte tror du vi lægger en ny version af Dinero online?

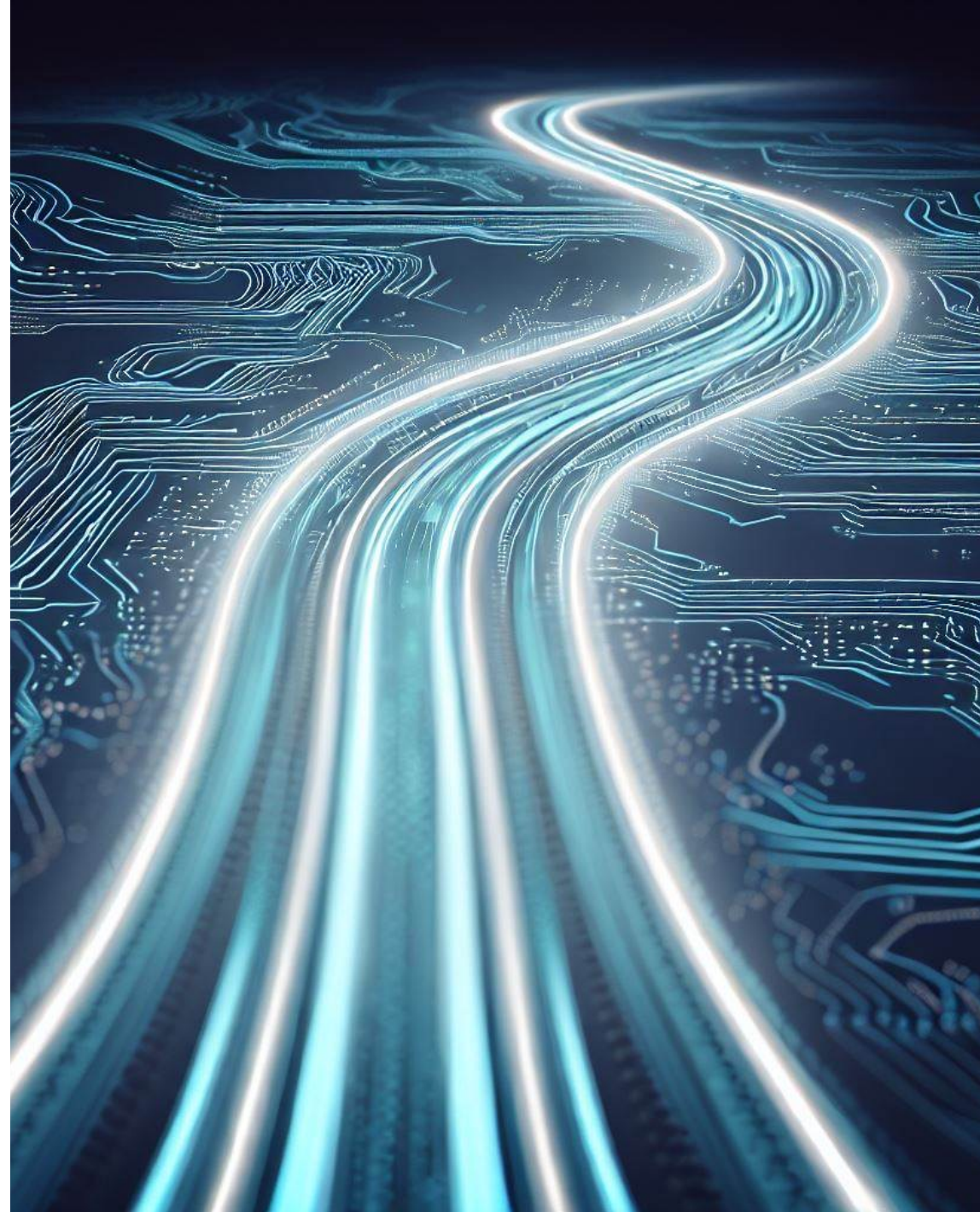
19.00 · 23/01/2023

Nøgletal - i mio. kroner



Reference

How do you get there ?



Supply-chain Levels for Software Artifacts, or SLSA ("salsa").

It's a security framework, a checklist of standards and controls to prevent tampering, improve integrity, and secure packages and infrastructure. It's how you get from "safe enough" to being as resilient as possible, at any link in the chain.

SLSA Levels

Level	Description	Example
1	Documentation of the build process	Unsigned provenance
2	Tamper resistance of the build service	Hosted source/build, signed provenance
3	Extra resistance to specific threats	Security controls on host, non-falsifiable provenance
4	Highest levels of confidence and trust	Two-party review + hermetic builds

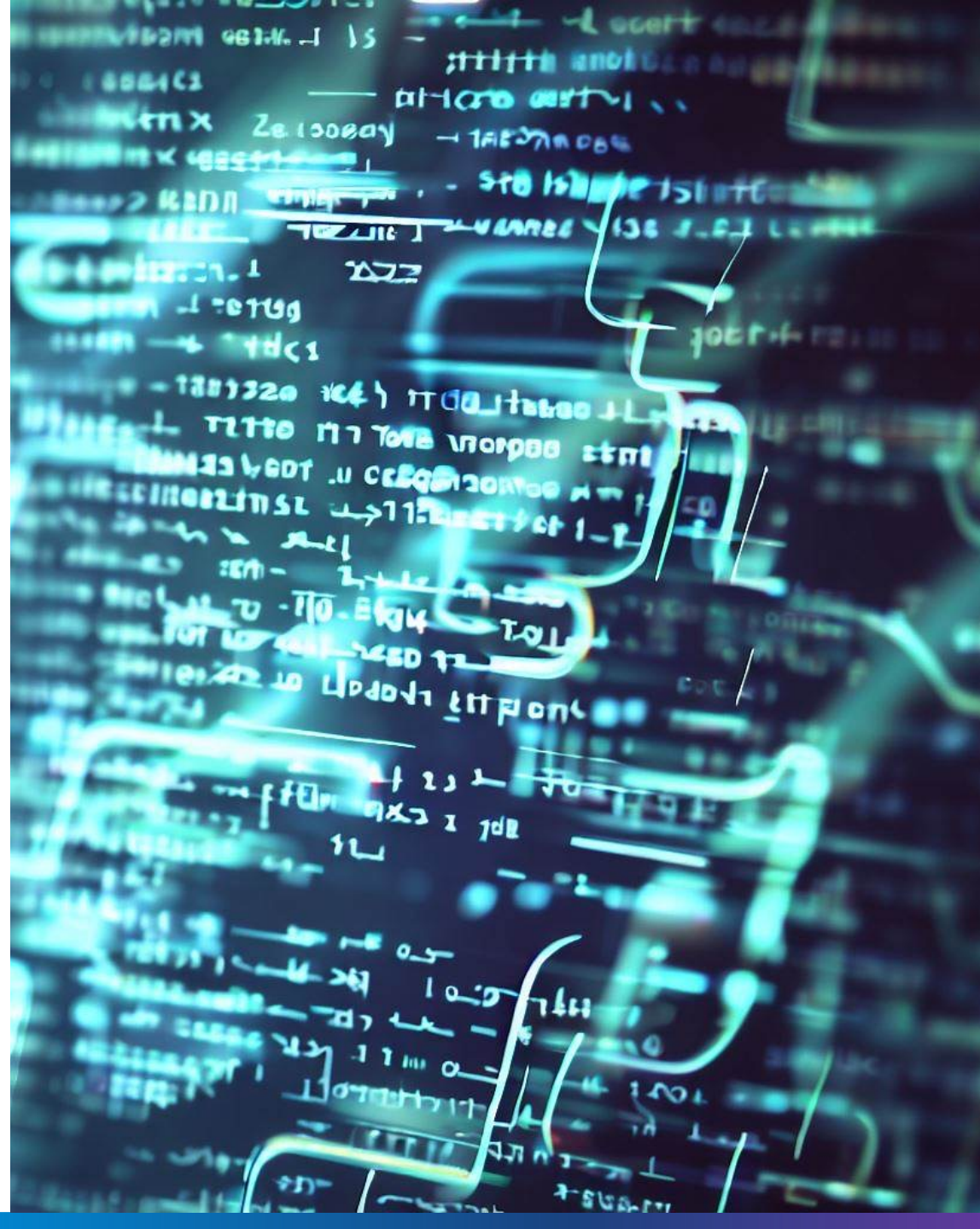
<https://slsa.dev/spec/v0.1/levels>

Level Requirements

Requirement	SLSA 1	SLSA 2	SLSA 3	SLSA 4
Source - Version controlled		✓	✓	✓
Source - Verified history			✓	✓
Source - Retained indefinitely			18 mo.	✓
Source - Two-person reviewed				✓
Build - Scripted build	✓	✓	✓	✓
Build - Build service		✓	✓	✓
Build - Build as code			✓	✓
Build - Ephemeral environment			✓	✓
Build - Isolated			✓	✓
Build - Parameterless				✓
Build - Hermetic				✓
Build - Reproducible				○

<https://slsa.dev/spec/v0.1/requirements>

Scripted Builds



What Tools To Use ?

Remember “As Code” & “Idempotence”



- Terraform
- Ansible
- Bash
- Puppet
- SaltStack (Salt)
- Powershell
- Python
- Golang
- Others ?

Idempotence

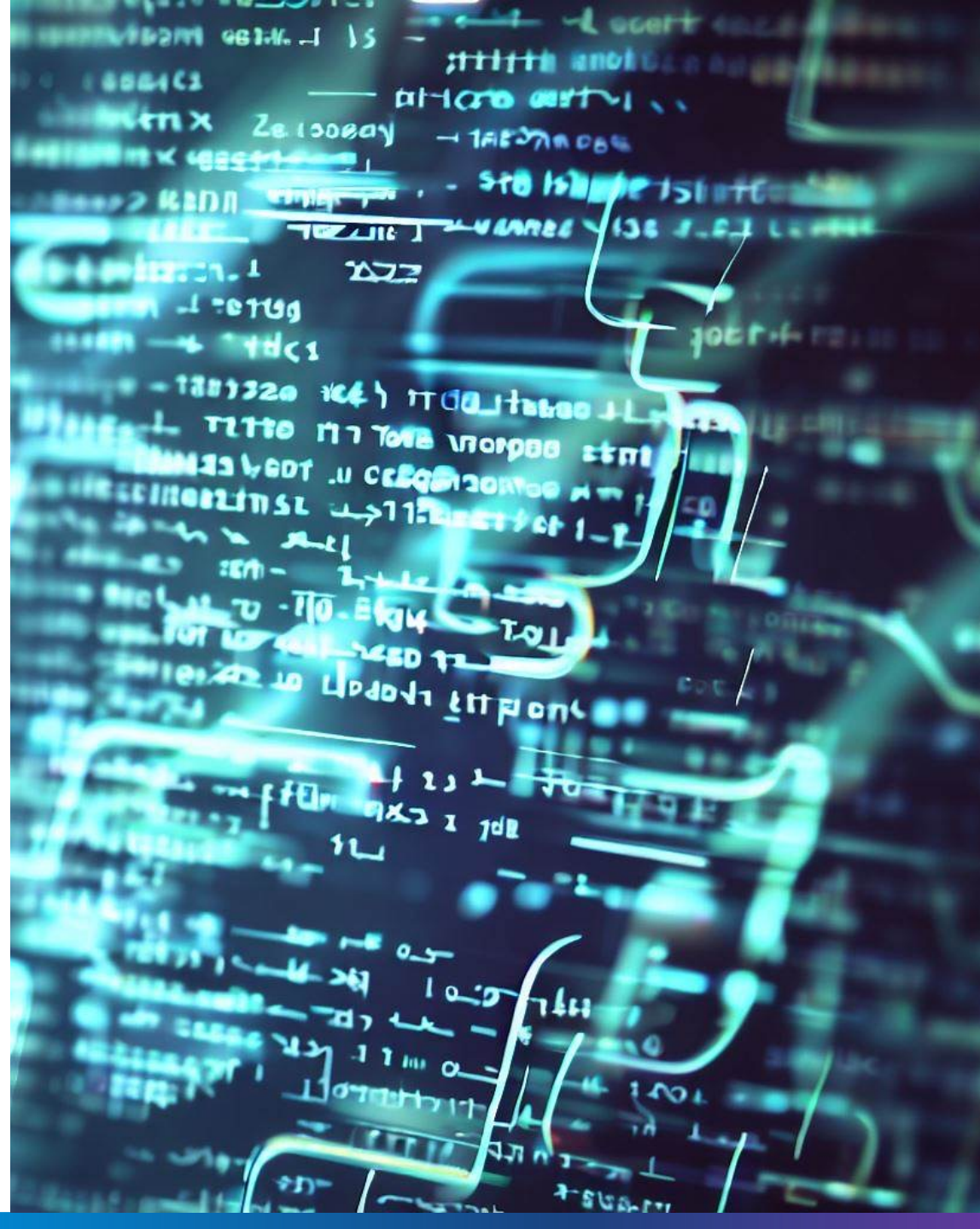
Get the same result every time

```
Dockerfile 1, U x
Dockerfile > ...
1 FROM ubuntu
2 RUN apt update && apt upgrade -y
3 RUN apt install apache2 -y
4 COPY . .
5 CMD [main.go]
```

```
Dockerfile U x
Dockerfile > ...
1 FROM ubuntu:jammy-20230425
2 RUN apt update && apt install apache2=2.4.52-1ubuntu4.5 -y
3 COPY . .
4 CMD [main.go]
```



Version Controlled



The power of Git

The screenshot displays a Git commit history interface. At the top, a commit on the `master` branch is shown with the following details:

Branch	Description	Date	Author	Commit
<code>master</code> 2		28 Apr 2023 10:54	Robert Jensen	9afab749

Below this, a commit on the `main` branch is shown with the following details:

Branch	Description	Date	Author	Commit
<code>main</code>	initial commit	28 Apr 2023 10:52	Robert Jensen	538da353

The `main` branch section also includes a list of recent commits:

- minor update** (rhjensen79 committed 13 hours ago) - Verified (529401a)
- working with new login feature** (rhjensen79 committed 13 hours ago) - Verified (e1ac057)
- fix** (rhjensen79 committed yesterday) - Verified (62c8333)

Red arrows point to the `Verified` status of each commit in the list.

Build Service



Build Service

Which one to use ?



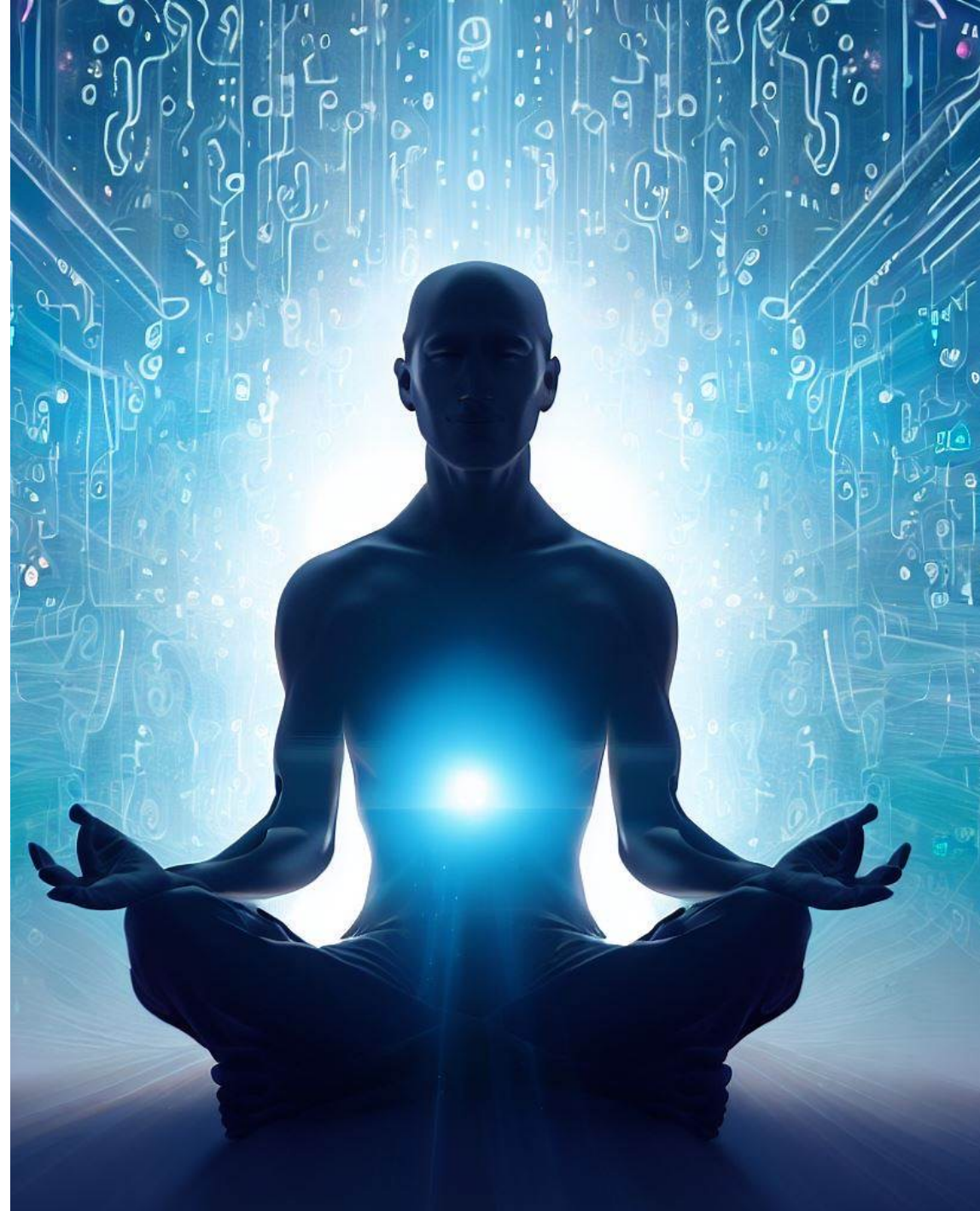
- Github Action
- Jenkins
- Travis CI
- Aria Pipelines
- Others



Simple Github Actions
example of building
code, at every push.

```
1  name: Go
2  on:
3    push:
4      branches: [ "main" ]
5  jobs:
6    build:
7      runs-on: ubuntu-22.04
8      steps:
9        - uses: actions/checkout@v3
10       - name: Set up Go
11         uses: actions/setup-go@v3
12         with:
13           go-version: 1.19
14       - name: Build
15         run: go build -v ./...
16       - name: Test
17         run: go test -v ./...
```

What is the/one solution ?



Backstage





Backstage

«Happy developers make happy code»

The Spotify Journey

.. back in 2016



hyper-growth
mode



speed-to-
market



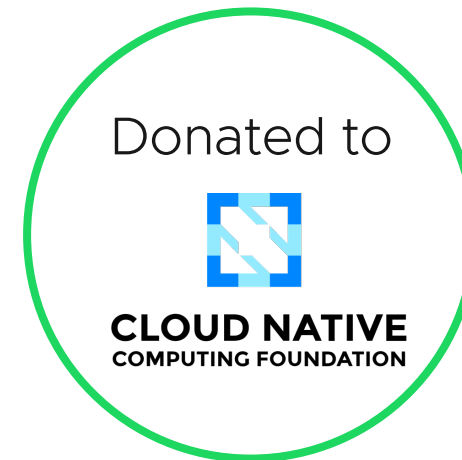
developer
effectiveness



Backstage is an open platform for building developer portals.



2016



2020

Attributes Of Developer Portal

Lowers **cognitive load** for developers and boosts Developer Productivity by creating



Provides a developer centric view



Abstracts away underlying technology

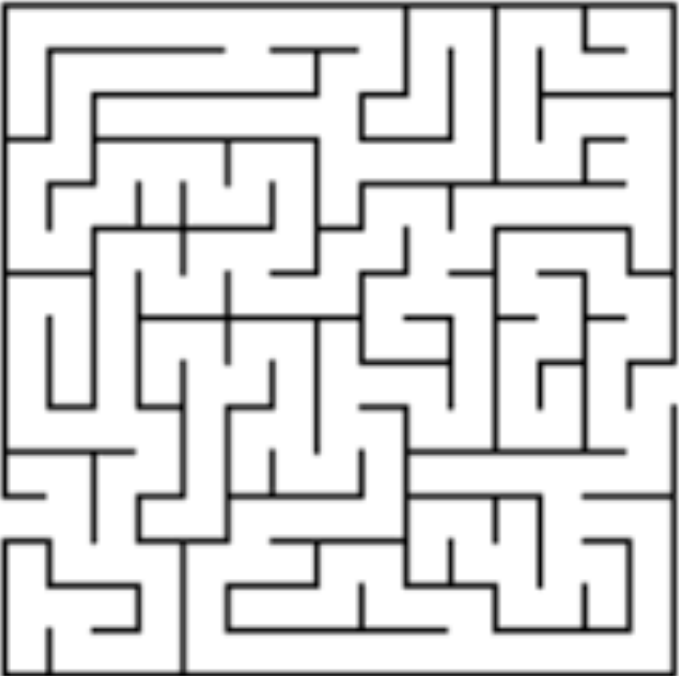


Provides a pluggable framework

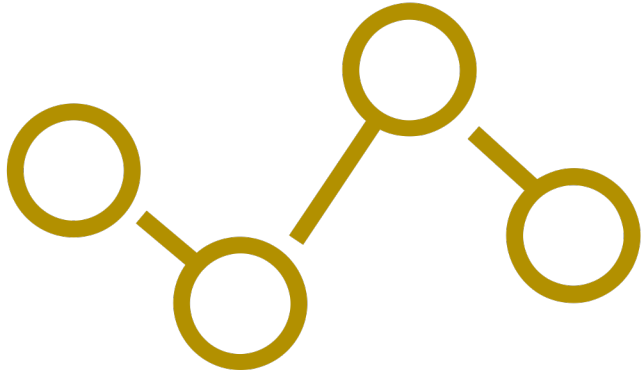


Maximizes Time to Value

Finding the golden path | path to production



This is the way



Golden Path

Backstage plugin ecosystem

Customizable and extensible plugin architecture

- Built with modern technologies and common frameworks
- Makes it easy to develop for and contribute to your dev portal
- Cloud-agnostic and vendor-neutral

Azure Sites by FRISS Infrastructure Azure Sites (Apps & Functions) support for a given entity. View the current status of the site, quickly jump to site's Overview page, or Log Stream page. Explore	backstage-plugin-api-linter by Zalando Linting API Linter is a quality assurance tool that checks the compliance of API's specifications to Zalando's API rules. Explore	Badges by Andreas Stenius Discovery The badges plugin offers a set of badges that can be used outside of Backstage, showing information related to data from the catalog. Explore
Bazaar by Axis Communications AB Discovery A marketplace where engineers can propose projects suitable for inner sourcing Explore	Bitrise by SDA SE CI/CD View Bitrise builds and download the build artifacts within Backstage. Explore	Bugsnag by roadie.io Monitoring View and monitor Bugsnag errors. Explore
Buildkite by roadie.io CI/CD View Buildkite CI builds for your service in Backstage. Explore	Catalog Graph by SDA SE Discovery Extend the Backstage Software Catalog with a graph that shows all entities and their relationships providing an easier way to discover the ecosystem. Explore	CircleCI by Spotify CI/CD Automate your development process with CI hosted in the cloud or on a private server. Explore
Cloud Carbon Footprint by Thoughtworks Metrics View your cloud carbon footprint by estimating energy use (kilowatt-hours) and carbon emissions (metric tons CO2e) from public cloud usage. Explore	Cloudify by Cloudify Orchestration Cloudify provides a remote execution and environment management backend for Kubernetes, Terraform, Ansible, etc. Explore	CodeScene by CodeScene Quality CodeScene is a multi-purpose tool bridging code, business and people. See hidden risks and social patterns in your code. Prioritize and reduce technical debt. Explore



Backstage in open source



**CLOUD NATIVE
COMPUTING FOUNDATION**



**1,500+
contributors**



**3,200+
project forks**



**400+
adopters**



**7,300+
members**



**13,000+
contributions**

Backstage is building a proven track record across industries



NETFLIX

ROKU



TOYOTA

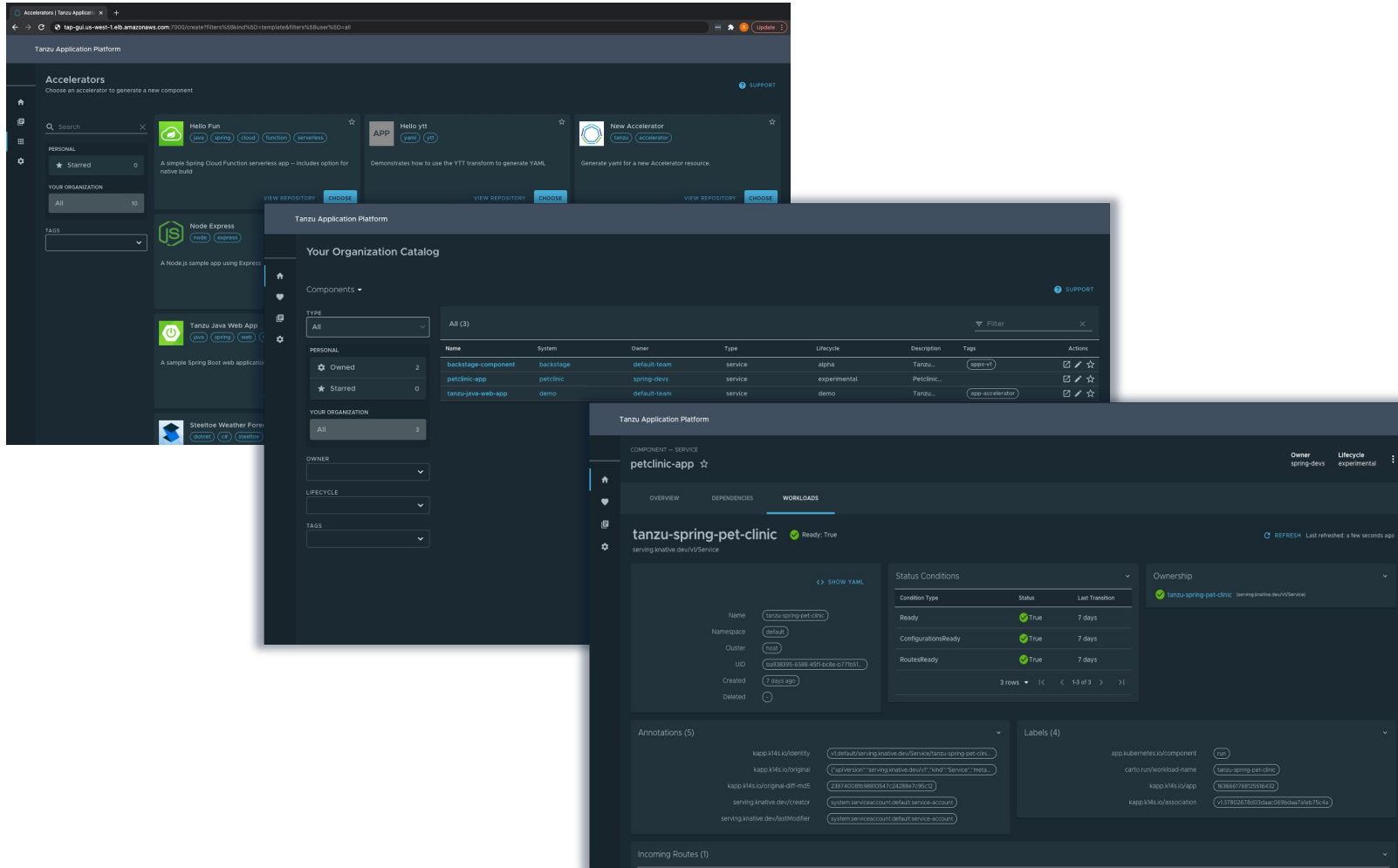
Currently ~~75-224~~ 241 public adopters

<https://github.com/backstage/backstage/blob/master/ADOPTERS.md>

Tanzu Application Platform (TAP)



Based on Backstage, but with it's own opinions (and focus)



Pre installed with

- Docs
- Accelerators
- API's
- Supply Chain
- Security Analytics

Only runs on K8S

Overview

The screenshot displays the VMware Tanzu Application Platform interface. The main heading is "Your Organization Catalog". On the left, there is a sidebar with navigation options: "Systems", "PERSONAL" (with "Owned" and "Starred" counts of 0), "YOUR ORGANIZATION" (with "All" count of 4), "OWNER", and "TAGS". The main content area shows a table of systems under the "All (4)" filter. The table has columns for Name, Owner, Description, Tags, and Actions. The systems listed are cncf, demo, tanzuquiz, and tanzutrends, all owned by denmark-team. The tanzuquiz system has tags for "quiz" and "postgres".

Name	Owner	Description	Tags	Actions
cncf	denmark-team	CNCF Demo		[Edit] [Delete] [Star]
demo	denmark-team	Demo Appl...		[Edit] [Delete] [Star]
tanzuquiz	denmark-team	Tanzu Quiz...	quiz, postgres	[Edit] [Delete] [Star]
tanzutrends	denmark-team	Tanzu...		[Edit] [Delete] [Star]

- Owners
- Systems
- Groups
- Dependencies
- Resources

Documentation

The screenshot displays the VMware Tanzu Application Platform interface for the 'tanzutrends' system. The interface is dark-themed and includes a navigation sidebar on the left. The main content area is divided into several sections:

- About:** Contains 'VIEW SOURCE' and 'VIEW TECHDOCS' buttons, a description 'Tanzu Trends Demo Application', and metadata including 'OWNER: denmark-team', 'DOMAIN: tap-gui-domain', and 'TAGS: No Tags'.
- Relations:** A graph showing the relationship between 'group:denmark-team' (owner) and 'system:tanzutrends' (ownedBy). The system has three parts: 'component:frontend', 'component:scrape', and 'resource:tanzutrends-db'.
- Has components:** A table listing components with their names, owners, types, lifecycles, and descriptions.
- Has resources:** A section for resources, currently empty.
- APIs:** A section indicating that the system does not contain any APIs.

Name	Owner	Type	Lifecycle	Description
frontend	denmark-team	website	production	The frontend for the Tanzu...
scrape	denmark-team	service	production	The scraper for the Tanzu...

- Based on Mkdocs
- Written in markdown
- Documentation & Code in same repo.
- Build when you push

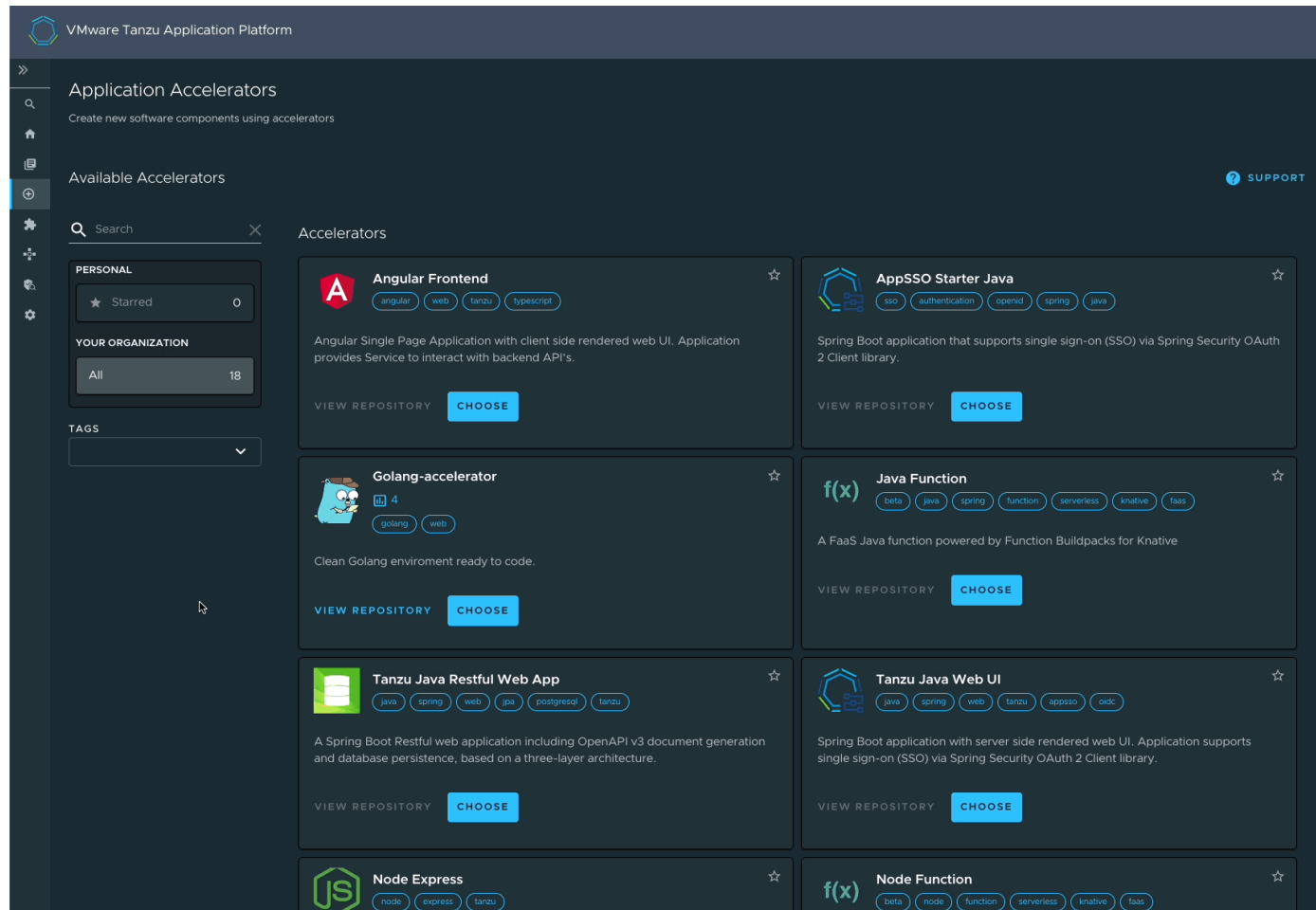
API

The screenshot shows the VMware Tanzu Application Platform API Explorer interface. The header displays the VMware logo and the text "VMware Tanzu Application Platform". Below the header, the page title is "APIs" and the subtitle is "Your Organization API Explorer". A "SUPPORT" link is visible in the top right corner. The interface is divided into several sections:

- Left Sidebar:** Contains navigation icons and filter sections for "Type", "PERSONAL", "YOUR ORGANIZATION", "OWNER", and "LIFECYCLE".
- Filter Section:** A "Type" dropdown menu is set to "All".
- Table:** A table with columns: Name, System, Owner, Type, Lifecycle, Description, Tags, and Actions. It contains one entry: "fastapi-test-vmlab.tanzu.dk" with system "demo", owner "denmark-team", type "openapi", lifecycle "vmlab.tanzu.dk", and description "A set of API...".

- Based on OpenAPI
- Auto updated with app
- Dependencies
- Internal / External Api's
- Swagger interface

Application Accelerators



- Templates for Apps, Baselines etc.
- Can be "anything"
- Takes input
- Customizable using Sed, YTT, etc.
- "Golden Path" build in.

Supply Chain

The screenshot displays the VMware Tanzu Application Platform Supply Chain UI. The main view shows a pipeline flow for the 'frontend' application. The pipeline consists of the following stages:

- Source Provider (GitRepository, 14 hours ago)
- Source Tester (PipelineRun, 14 hours ago)
- Source Scanner (Grype, 14 hours ago)
- Image Provider (Image, 12 hours ago)
- Image Scanner (Grype, 11 hours ago)
- Config Provider (PodIntent, 15 hours ago)

The pipeline is currently in a 'Ready' state with 0 errors and 0 warnings. The 'Source Provider' stage is selected, and its details are shown in the 'Stage Detail' panel below.

Stage Detail: Source Provider
14 hours ago

Overview

Name	frontend
Kind	GitRepository
Source URL	https://github.com/TanzuDK/TanzuTrends.git
Version	source.toolkit.fluxcd.io/v1beta1
Namespace	default

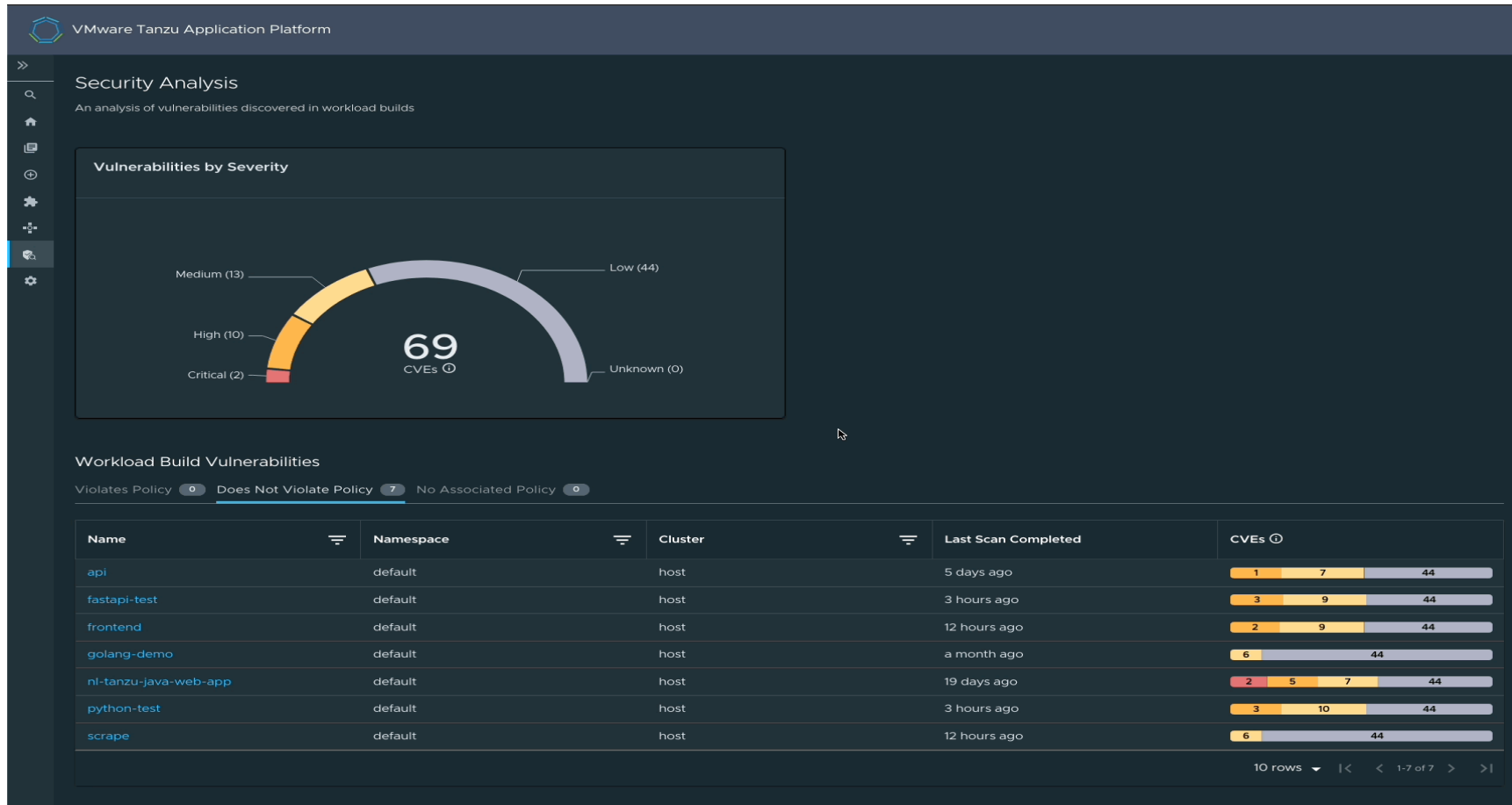
Conditions

Condition Type	Status	Last Transition
Ready	True	2023-05-07T20:06:00Z
ArtifactInStorage	True	2023-05-07T20:06:00Z

5 rows | 1-2 of 2

- Supply Chain UI
- CVE info
- Test info
- Overview

Security Analysis



- Security overview

Run

The screenshot displays the VMware Tanzu Application Platform (TAP) interface. The main section is titled "Your Organization Catalog" and shows a list of application components. The components are organized into a table with columns for Name, System, Owner, Type, Lifecycle, Description, Tags, and Actions. The components listed are:

Name	System	Owner	Type	Lifecycle	Description	Tags	Actions
api	tanzuquiz	denmark-team	api	production	The api for...	golang	📄 ✎ ☆
fastapi-test	demo	denmark-team	service	experimental	FastAPI fr...	python, fastapi, api	📄 ✎ ☆
frontend	tanzutrends	denmark-team	website	production	The...	python, web	📄 ✎ ☆
golang-demo	cncf	denmark-team	service	experimental	A golang...	golang, web	📄 ✎ ☆
nl-tanzu-java-web-app	java-web-app	dutch-team	service	experimental	Tanzu Java...	app-accelerator, java, spring, web, tanzu	📄 ✎ ☆
python-test	demo	denmark-team	service	experimental	A python...	python, streamlit	📄 ✎ ☆
scrape	tanzutrends	denmark-team	service	production	The scraper...	golang	📄 ✎ ☆

The interface also includes a sidebar with navigation options (Home, Components), filters (Type, PERSONAL, YOUR ORGANIZATION, OWNER, LIFECYCLE, TAGS), and buttons for "REGISTER ENTITY" and "SUPPORT".

- Run the app, on the TAP cluster (or a dedicated run cluster)

Built with an open source-first mindset

Innovative, interoperable, scalable and secure solutions

Tanzu Application Platform is backed by some of the most mature and popular open-source projects available today

In addition to Backstage Backstage

Garnering 200+ plugins Backstage has gained tremendous traction by helping organizations build self-service developer portals

Carvel CARVEL

Developers build, deploy, and manage their own apps and package them so they are more easily distributable

Cartographer CARTOGRAPHER

Operator teams create secure, reusable supply chains that define all of application CI and CD in one place

And many more....



grype



TEKTON



Buildpacks.io

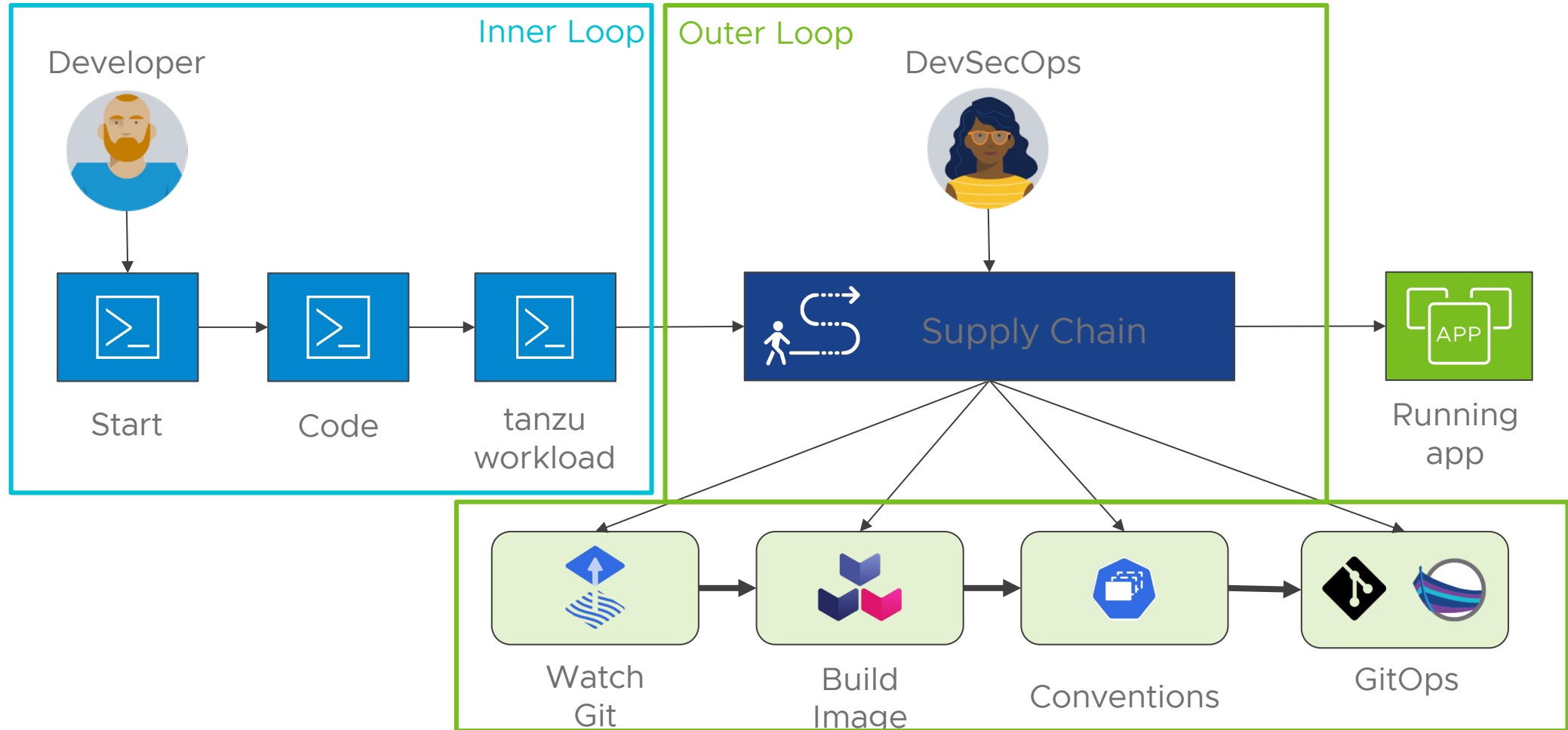


cosign

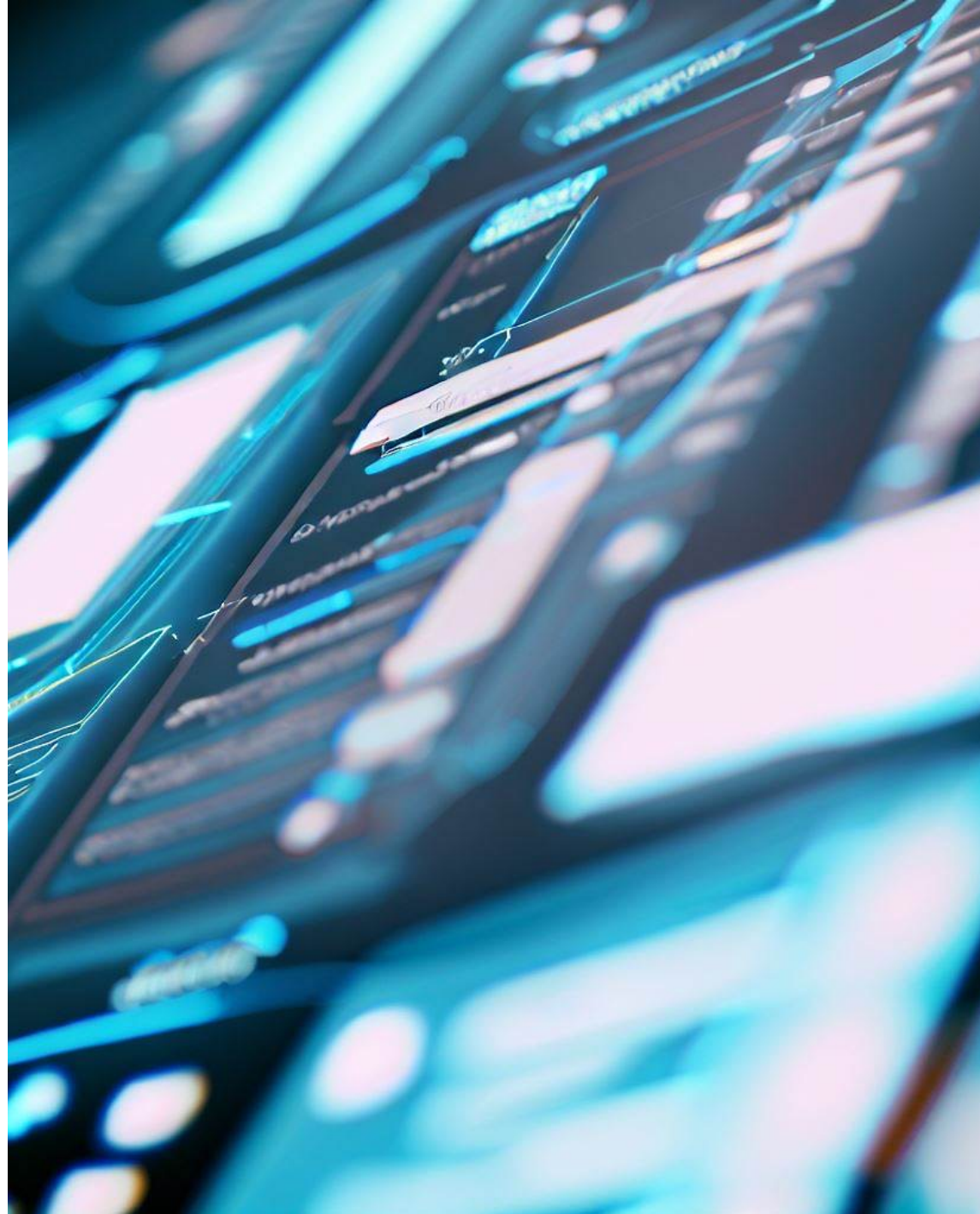
TILT

Building open-source software and contributing to communities is at the core of VMware's engineering spirit

Deploying an App with Tanzu Application Platform



Come by the booth, for a
“real” demo and a talk.



Thank You

