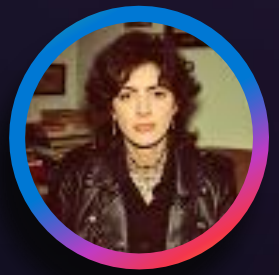
Three Horizons

by Red Hat®, Microsoft® e GitHub®

Transform Development. Transform Business. Transform Everything.

Get to know our amazing team



Paula Silva
Software Global Black Belt Team
Microsoft

Ana Basso
Ecosystem Solution Architect
Red Hat

João Peixoto
Senior Cloud Services Black Belt
Red Hat

Marcos Falcirolli
Partner Development Manager
Microsoft

DevOps practices are delivering results

Speed & quality

Security and testing are "shifting left", enabling faster delivery

Choice & flexibility

Cloud native trends have created choice and flexibility for dev teams

Expert utilization

Central teams provide recommendations on security, governance, technology & risk assessments

But has limits

Complexity creates challenges

Limits scaling

Teams have limited bandwidth

Approved projects take **days to weeks** to get started

Developers just want to code

Poor developer experience

Develop a new approach for every project

Must win approval from key stakeholders

Unclear processes lead to uncertainty and rework

Organizational waste

Wasteful reworking of previously solved problems

Lack of process documentation

“Organizational amnesia”

Developers determine speed to innovation

The developer experience matters

Centering developers

Helps identify system problems that are typically overlooked

Improving systems

Enhance speed and impact, reliability and availability, security and compliance

Four key areas drive developer Experience

Speed & Innovation

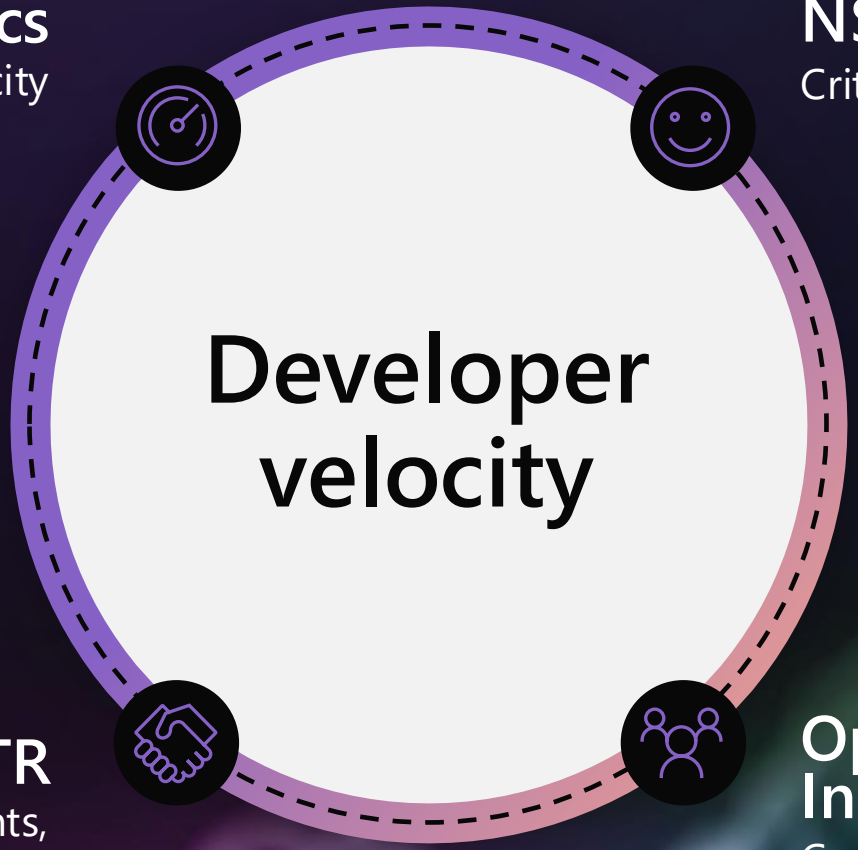
Talent & Happiness

Trust & Quality

Collaboration & Scale

DORA metrics
Code velocity

NSAT[1]
Critical dev workflow times



MTTR
For incidents, security and compliance bugs

Open Source & Inner Source
Contributions and manual steps for workflows

[1] NSAT = Net satisfaction score [2] MTTR = Mean-time-to-remediate

Common challenges...

DevOps practices are maturing…

Challenges

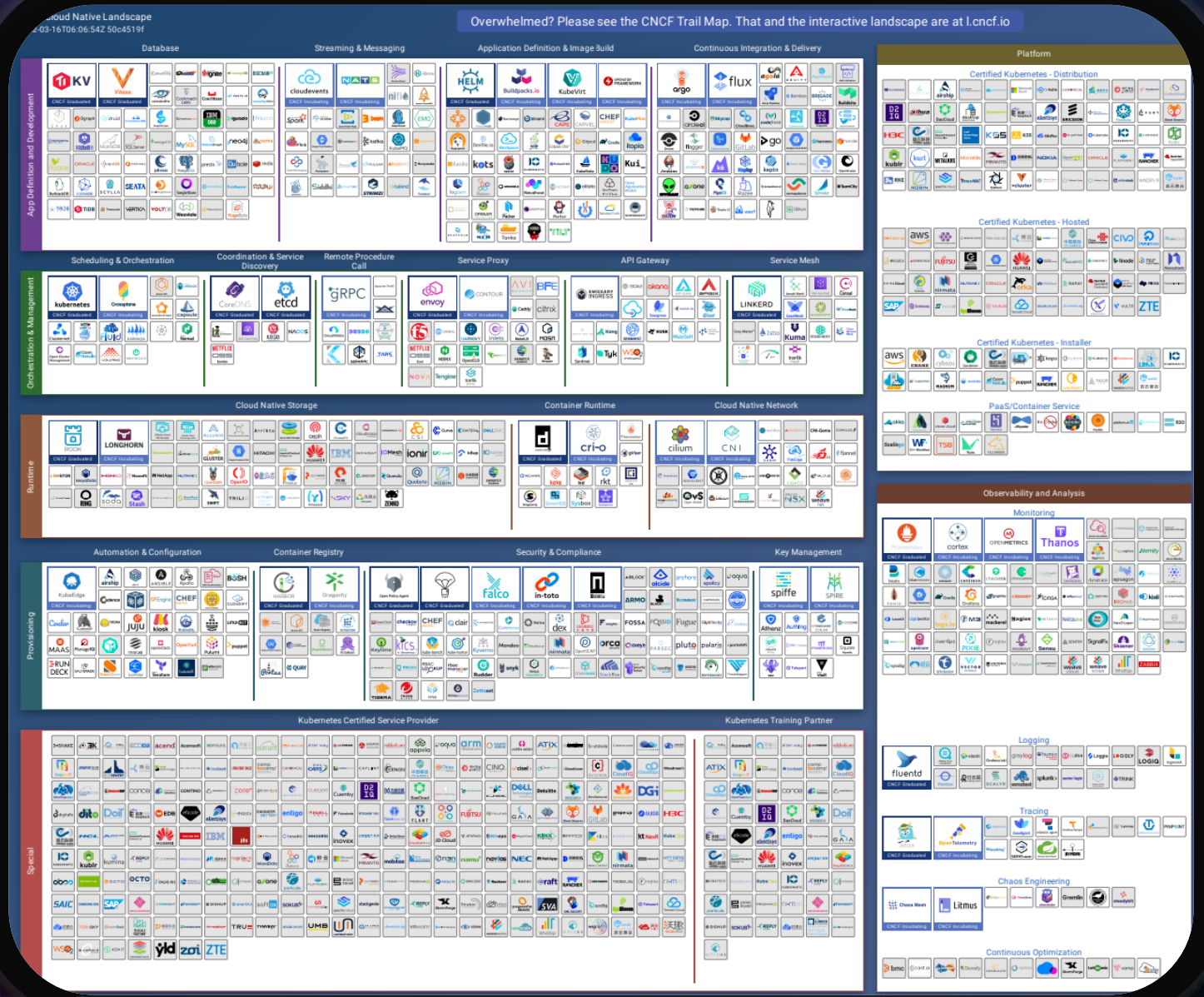
- Central teams need to support…
  - Variable app stacks, infra, software
  - Complex integrations across diverse stacks
  - The same “agile” pace of multiple dev teams
- Dev teams need to manually understand, maintain compliance, and other recommendations
- Tribal knowledge and organizational amnesia

Result

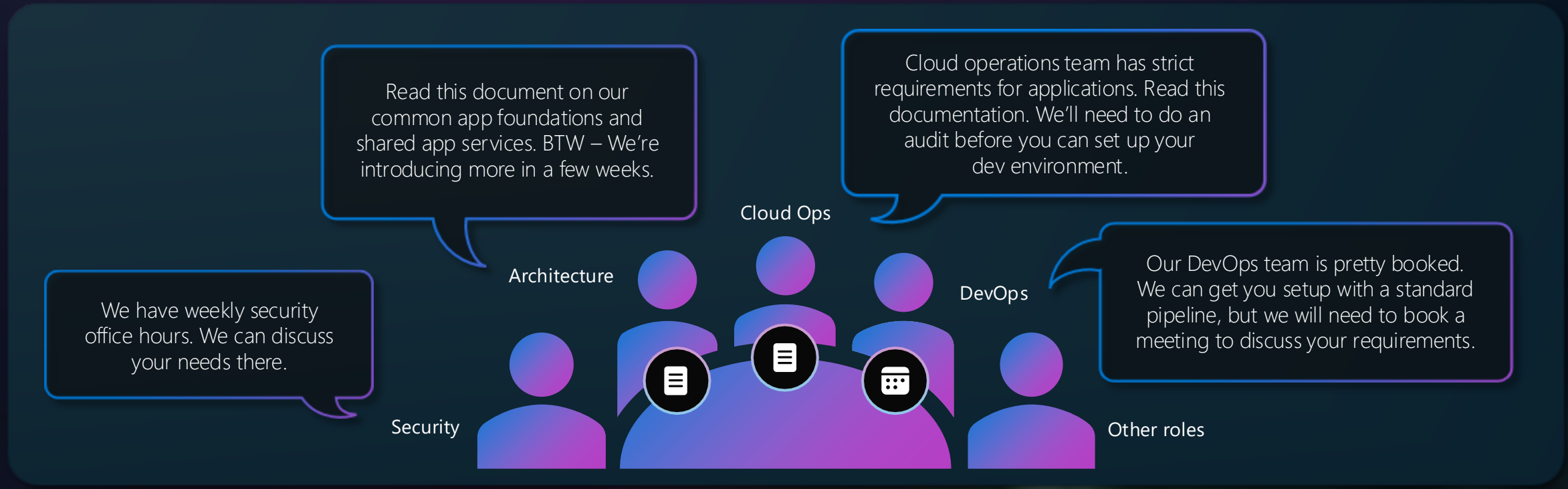
- Inconsistent standardization and compliance
- Potential for sub-optimal designs, lack of reuse
- Increased cost, risk
- Manual process reduces developer velocity and time to business value

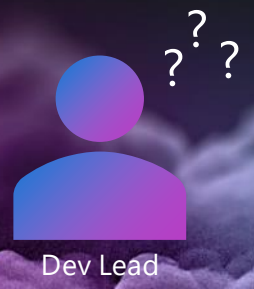
Some of the jobs to be done…

- Standardization: frameworks, opinionated stacks, …
- Discovery, e.g. catalog for apps, infra, docs
- Common approaches & solution templates
- Isolation (projects, workloads)
- User management/RBAC
- UX – CLI / UX
- Tool flexibility/addons
- App deployment
- Policies (infra, build, deployment, …)
- Observability
- Cost / Quotas
- Maintenance


oud Native Landscape
-03-16T06:06:54Z 50c4519f
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Continuous Optimization

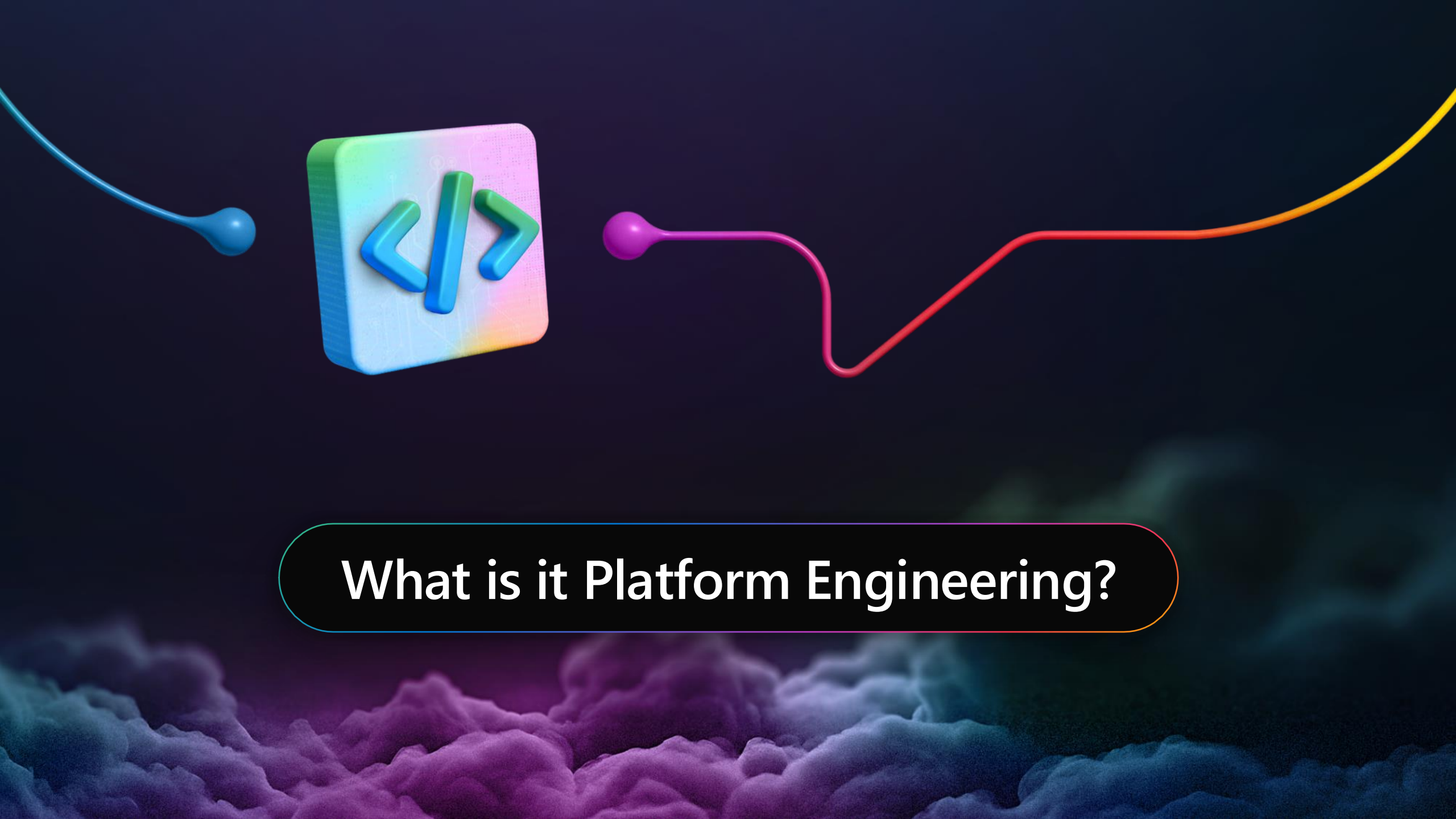
Just starting projects is challenging





Uh oh. Our management asked us to start by the end of this week. What now?

How can we keep all this up to date as things change? I just want to code.


What is it Platform Engineering?

Platform Engineering is the discipline of **designing and building self-service capabilities to minimize cognitive load for developers** and to enable fast flow software delivery.

Platform Engineering is about **solution**.

Balancing the needs

Developers

- Agility
- Collaboration
- Flexibility
- Productivity

Platform engineers

- Security
- Governance
- Standardization
- Cost management

Common goals for Platform Engineering practices

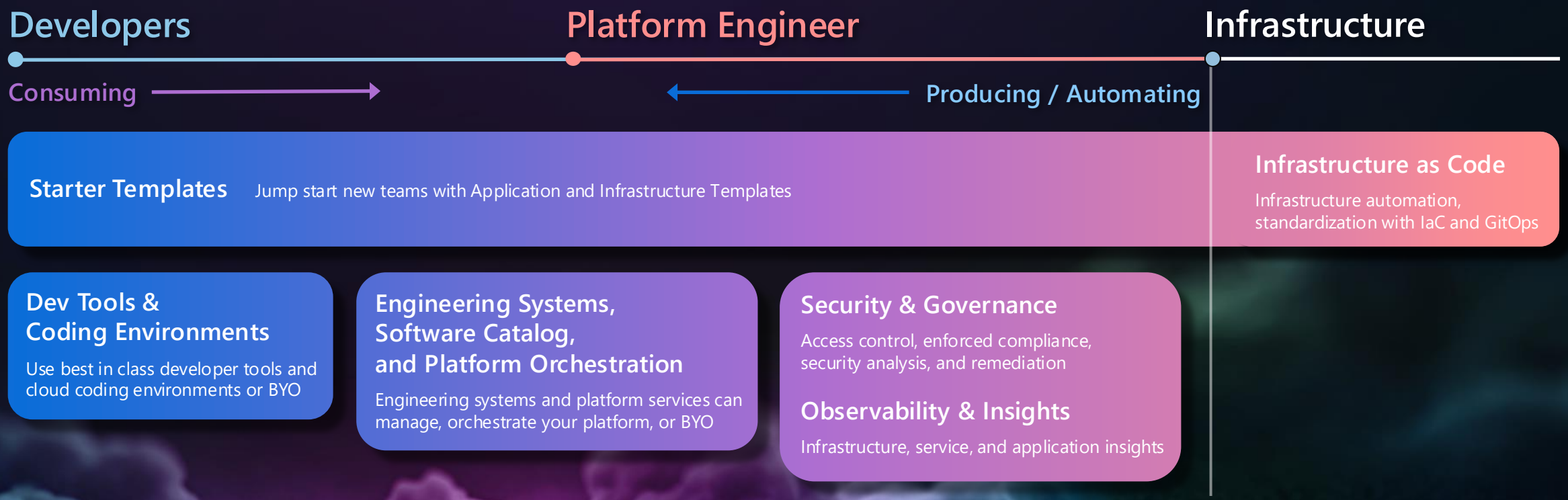
Self-service with guardrails

- Keep developers focused on delivering business value
- Secure & governed development, with flexibility
- Rapid & secure onboarding/offboarding
- Improve collaboration
- Costs optimization

Culture shift to product mentality

- Engineering Platform managed as a product
- Devs are the primary customer
- Automate away as much as possible
- Security, architecture, operations are key stakeholders

Key aspects of an Engineering Platform



Platform engineering first steps

At each step, establish metrics that are key to your success at that stage.
Use these to distill insights that determine if, and when, to move to the next step.

Establish platform engineering "product team"

Create a dedicated product team, which could be virtual, responsible for working across the organization to create a self-service developer platform.

Inventory & centralization

Perform an inventory of existing tools, engineering systems, and application platforms. Evaluate usage to determine where to invest and candidates for deprecation. Begin centralizing and reducing redundancy.

Automate "high toil" areas

Identify areas of high developer toil and automate them. These may be areas of "human in the loop" processes, etc.

Blaze paved paths

Use templates to create "paved paths" that help developers "fall into the pit of success" by easily following organizational best practices --- orchestration.

Deploy environments as a service

Develop guardrails and ensure separation of concerns security, e.g. leverage techniques for "on-behalf-of" infrastructure provisioning to prevent over privileged access.

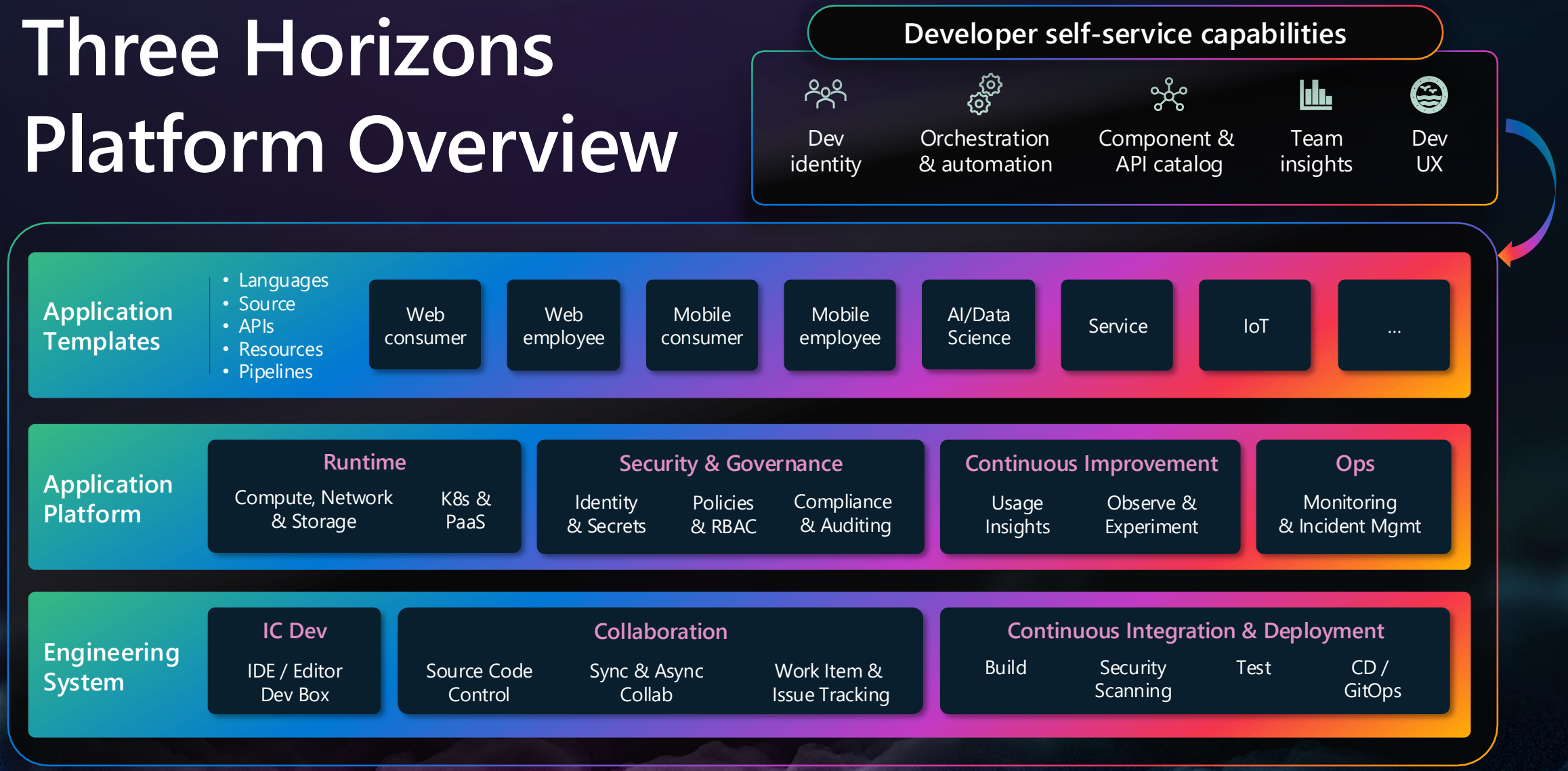
Optimize self-service developer experiences

Create self-service with guardrails that are easy to consume, secure and are compliant by design. Platform is offered as a "product".

Our solution...

Three Horizons is an enterprise **engineering platform** that addresses the growing complexity of **modern software development**. By integrating **Red Hat Developer Hub, Microsoft Azure, and GitHub Platform**, it provides organizations with **a unified solution** for application development, deployment, and operations.

Three Horizons Platform Overview



Three Horizons is not just a platform. It is a new way to structure your development with more practicality and results.

Key Differentiators

- ✓ **Unified Portal:** Red Hat Developer Hub installable on ARO or AKS based on customer preference
- ✓ **Complete Automation:** Infrastructure provisioning and CI/CD via GitHub Actions and Ansible templates
- ✓ **Flexibility:** Applications can be developed and deployed on AKS, ARO, or both
- ✓ **Developer Experience:** Accelerated complete cycle, with AI assistance via GitHub Platform
- ✓ **Transformation Potential:** Integrated engineering platforms enable significant reduction in environment setup time and accelerate development cycles, transforming software delivery from a barrier to a strategic business enabler.

For Development Teams

- 🚀 Accelerated Onboarding - Developers become productive in minutes, not days or weeks
- 🔄 Consistent Environments - Standardized DevBox templates ensure everyone works in identical environments
- 🧠 Reduced Cognitive Load - Golden Paths guide developers through common workflows without requiring deep platform knowledge
- 🤖 AI-Assisted Development - GitHub Copilot integration enhances developer productivity and code quality
- 📈 Streamlined Lifecycle - Seamless transitions from Inner Loop Development to Outer Loop Deployment deployment

For Platform Teams

- 🛠️ Simplified Management - Single pane of glass for administering the entire developer platform
- 📊 Enhanced Visibility - Comprehensive monitoring and analytics across all platform components
- 🔒 Integrated Security - Unified security controls and Security Policies
- ⚙️ Standardized Operations - Consistent Operations and automation
- 💼 Governance Compliance - Built-in controls to meet organizational and regulatory requirements

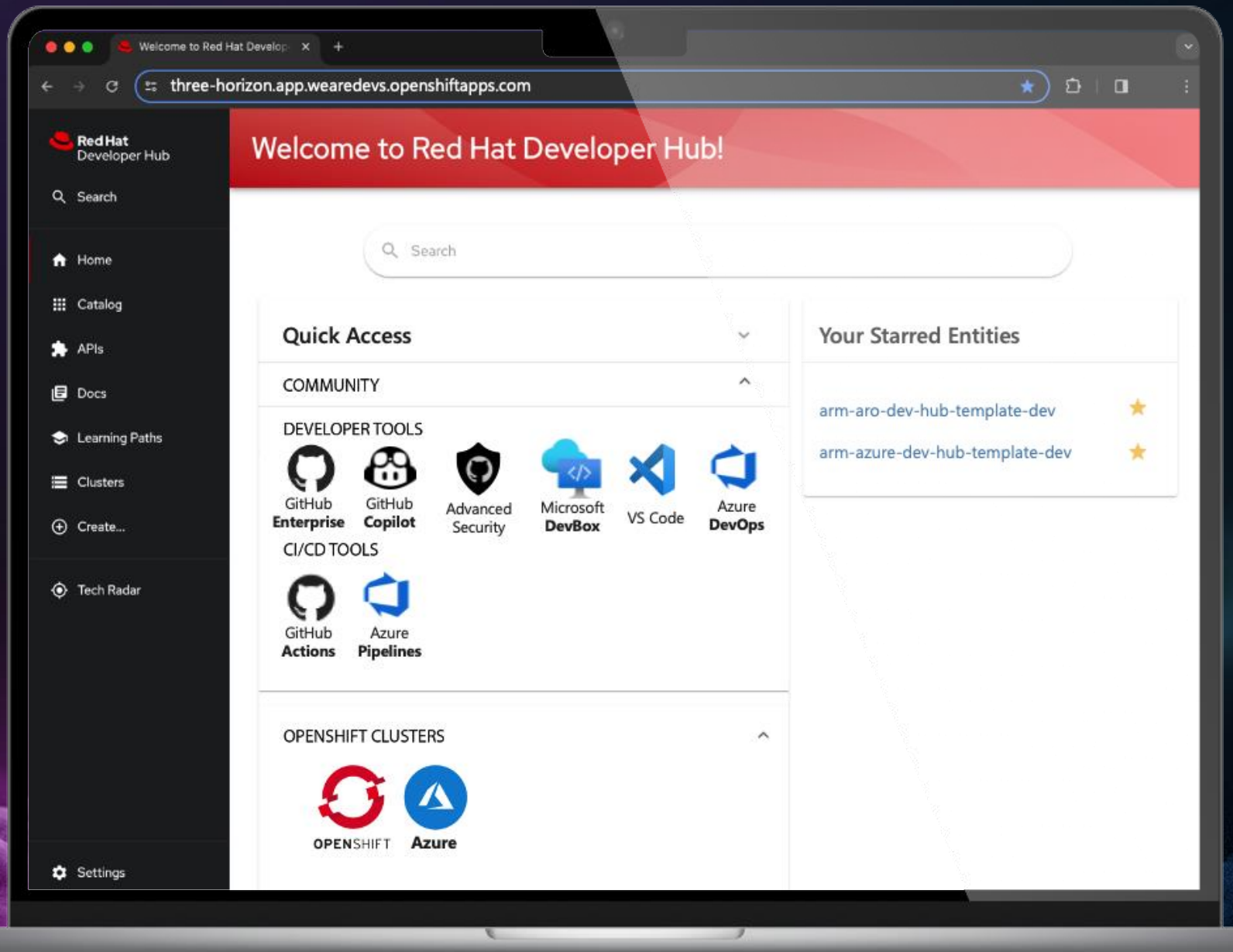
For Business Leaders

- ⏱️ Faster Time-to-Market - Reduce development cycles and accelerate innovation
- 💰 Lower Operational Costs - Optimize resource utilization and reduce platform maintenance overhead
- 🔍 Improved Quality - Consistent processes and automated testing increase overall software quality
- 🌐 Scalable Solutions - Easily scale from small teams to enterprise-wide adoption
- 💡 Competitive Advantage - Deploy new features and capabilities ahead of competitors

Enterprise challenges addressed

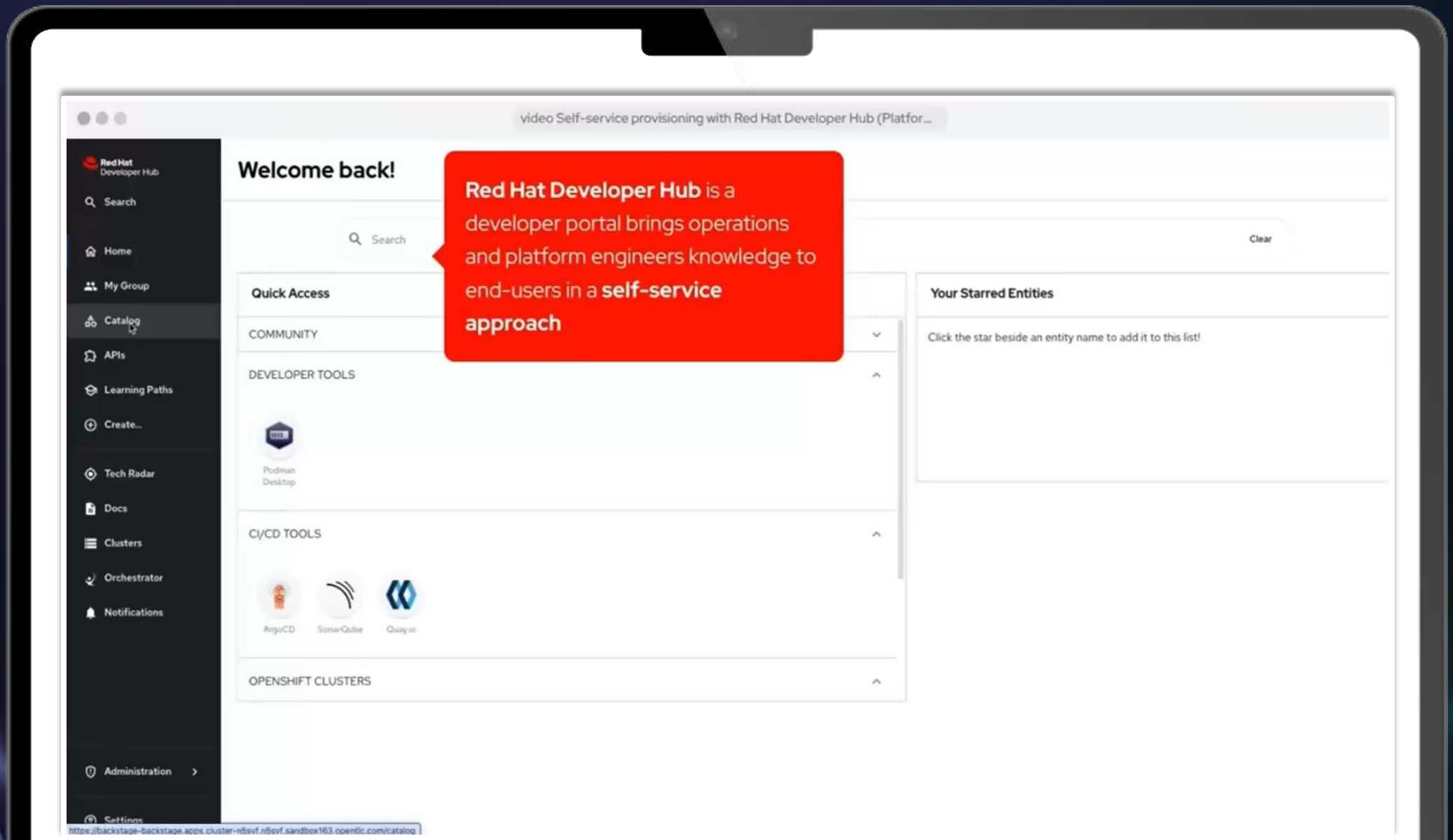
This integrated approach solves critical platform engineering challenges:

- **Developer cognitive overload:** Reducing the burden on developers through curated toolchains and contextual guidance Microsoft
- **Technical debt accumulation:** Addressing long-term sustainability through infrastructure as code and automated testing Red Hat
- **Complexity management:** Abstracting complexity through standardized interfaces and workflows GitHub
- **Security and compliance struggles:** Embedding controls directly into development pathways
- **Multi-environment inconsistency:** Providing uniform experiences across diverse infrastructure Microsoft



Demo

Software Catalog


video Self-service provisioning with Red Hat Developer Hub (Platfor...
Red Hat Developer Hub
Search
Home
My Group
Catalog
APIs
Learning Paths
Create...
Tech Radar
Docs
Clusters
Orchestrator
Notifications
Administration
Settings
Welcome back!
Red Hat Developer Hub is a developer portal brings operations and platform engineers knowledge to end-users in a self-service approach
Search
Clear
Quick Access
COMMUNITY
DEVELOPER TOOLS
Podman Desktop
CI/CD TOOLS
ArgoCD
SonarQube
Quay.io
OPENSHIFT CLUSTERS
Your Starred Entities
Click the star beside an entity name to add it to this list!

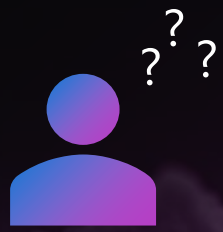
Simplifying a governed project lifecycle

Before: Manual

Read this document on our common app foundations and shared app services. BTW - We're introducing more in a few weeks.

IT operations team has strict requirements for applications. Read this documentation. We'll need to do a security audit before you can set up your dev environment.

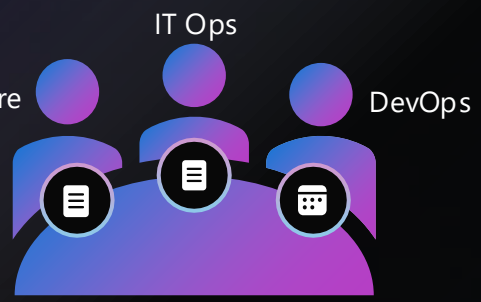
Our DevOps team is pretty booked. We can get you setup with a standard pipeline, but we will need to book a meeting to discuss your requirements.



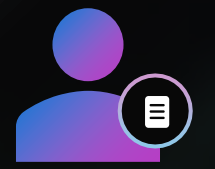
Uh oh. Our management asked us to start coding the end of this week. What now?

How can we keep all this up to date as things change?

After: Self-service

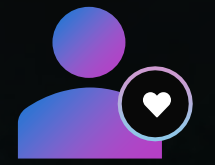


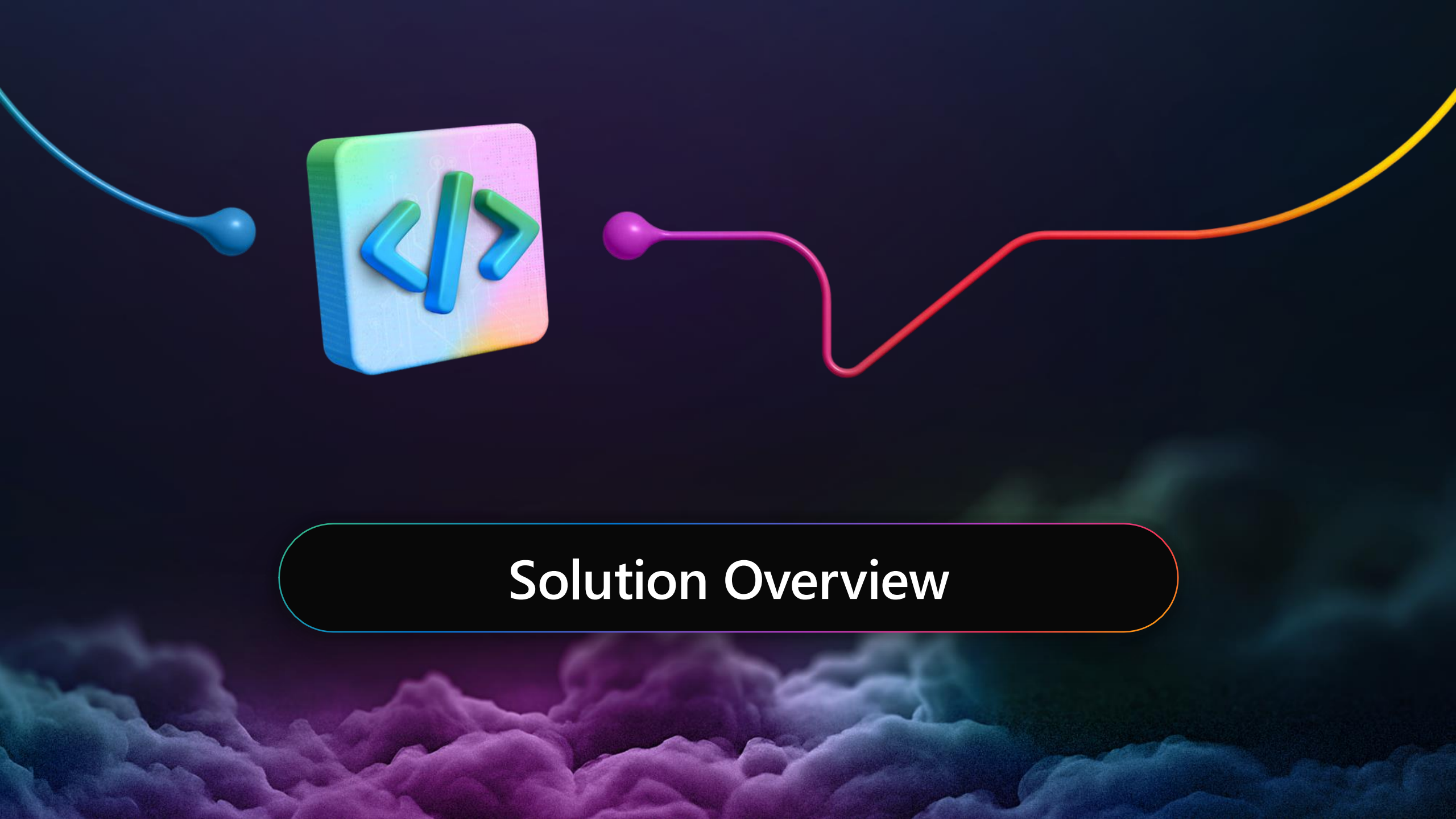
We've worked everyone to codify best practices into a set of Application Templates. Pick one to get started – setup is automated.



Self-service Template

- Sample source code
- Dev tool config for included languages
- Automation and deployment pipelines
- IaC for approved infra, observability
- Config for dev or test services
- Starting API references, SDKs





Solution Overview

Evolving DevOps

DevOps

"Union of people, process, and technology to enable continuous delivery of value to end users."

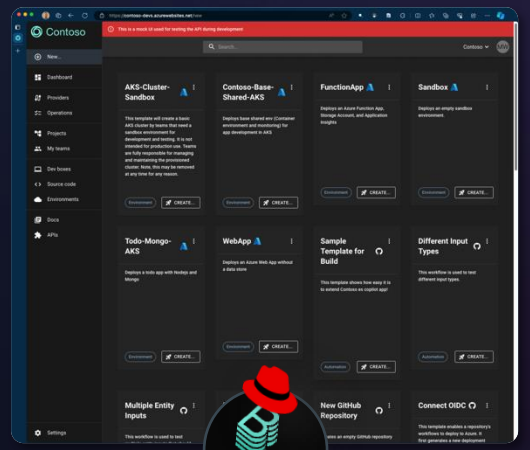
DevSecOps

"Union of people, process, and technology with security as a shared responsibility to enable continuous delivery of value to end users."

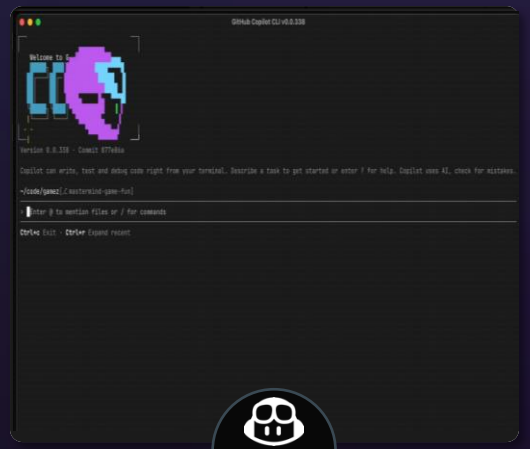
Agentic DevOps

AI-powered agents operating as a member of your dev team to automate, optimize, and accelerate every stage of the software development lifecycle.

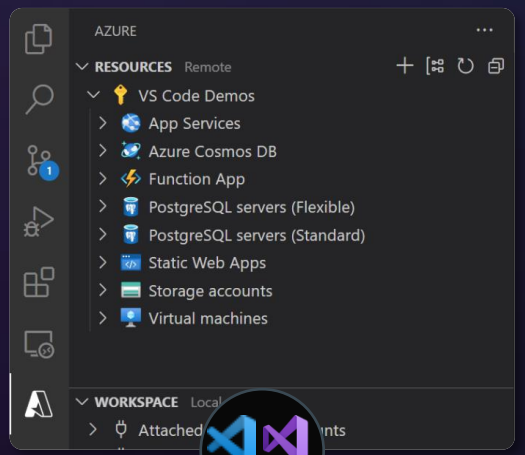
Three Horizons & Agentic DevOps Platform



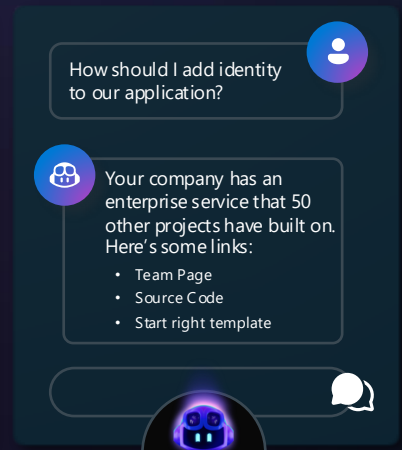
Developer Hub



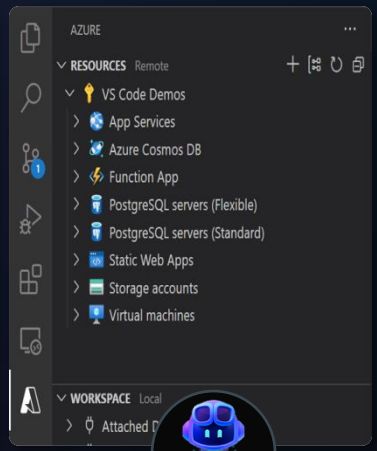
Copilot </> CLI



VS Family



Agent Mode



Coding Agent

Developer self-service foundation

Developer Platform API

Templates & Entity Graph → Platform Orchestrator

Entity Providers: Dev Box/ Codespaces | API Center | Power Platform | Service Catalog | Azure Deployment Environments | GitHub Actions/ Azure DevOps Pipelines | Argo CD/Flux (GitOps)

IaC and Configuration: Terraform | RED HAT ANSIBLE

Composable base
Extensible and interoperable with security and RBAC built in by default

Template driven
Configure & provision common or app specific infrastructure and tools

Engineering Systems: AI Foundry | GitHub Platform | Azure DevOps

Tools & Agents: MCP | Playwright | Semantic Kernel | Microsoft Agent Framework

Application Platform: Azure RedHat OpenShift | AKS | ACA | Azure Arc | aws | Hybrid / Multi-cloud

Monitoring & Observability

Azure Monitor | Application Insights | Log Analytics | Azure SRE Agent

Prometheus | Grafana | Datadog

Azure Security

GitHub Advanced Security

Container Scanning

IaC Scanning

Azure Key Vault

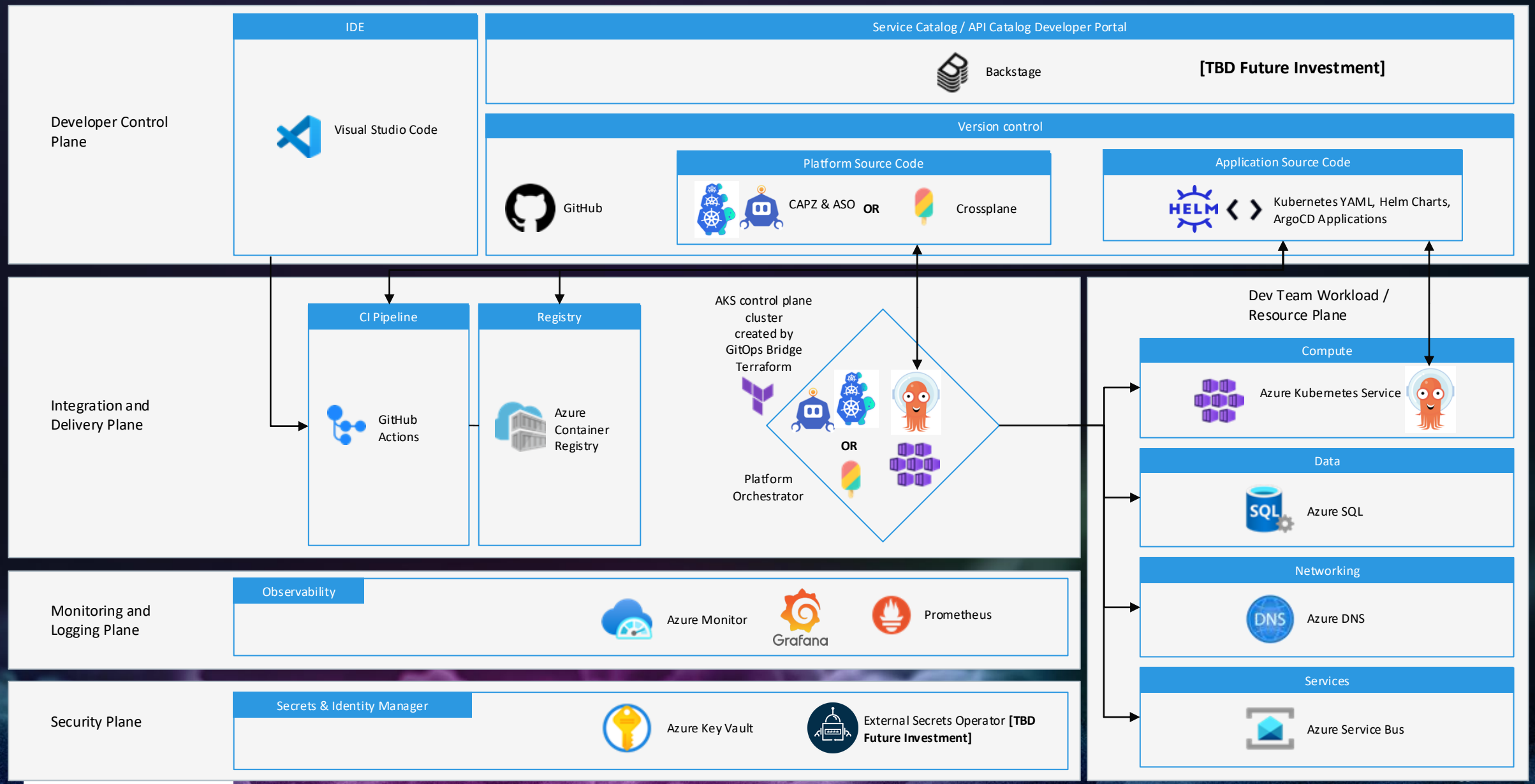
Defender for Cloud

Microsoft Purview

Azure Sentinel

AKS Internal Developer Platform

https://github.com/Azure-Samples/aks-platform-engineering



Across the SDLC

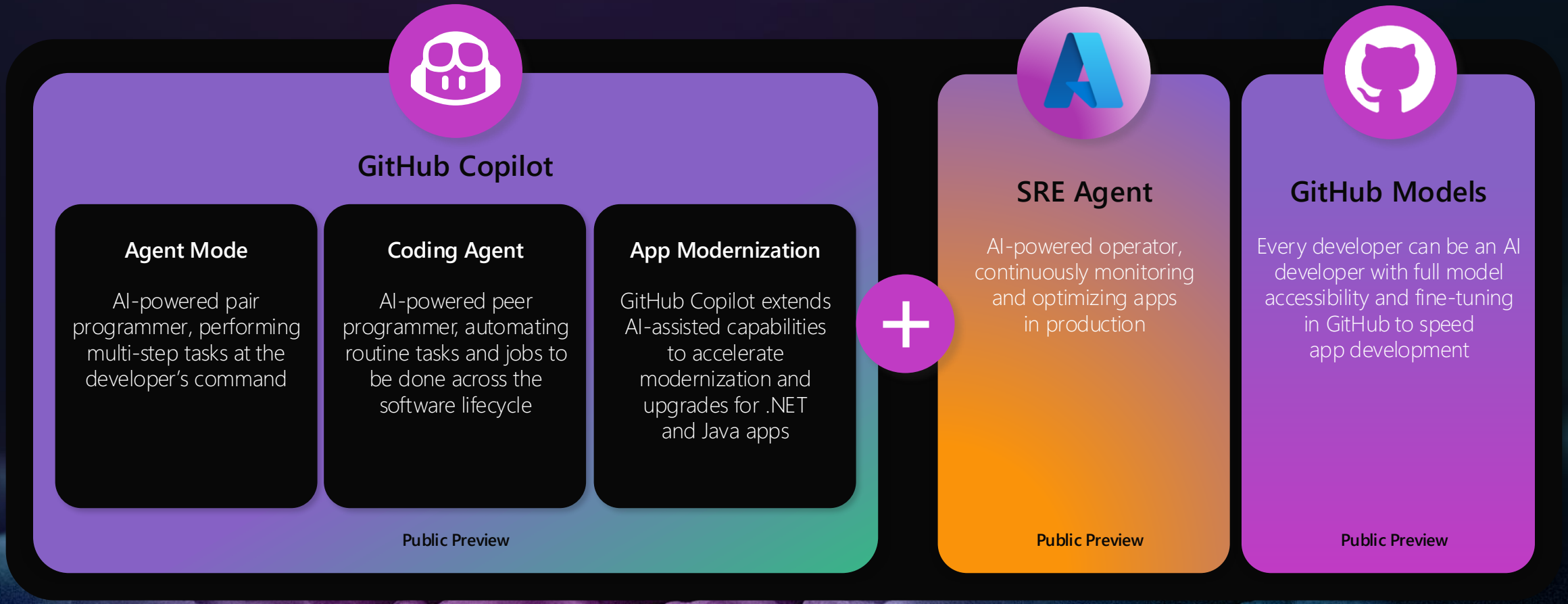
Policy & Governance

Plan	Code	Verify	Deploy	Operate
Planning Agent	Agent Mode	Autofix	Actions	Metrics
Issues	Coding Agent	Code Security	Spark Runtime	Models
Spaces	Spark Workbench	Secret Protection	AI Workflows	SRE Agent
Boards	Codespaces	Code Review	Pipelines	
		Pull Requests		

Integrations & MCP

Agentic DevOps

AI-powered agents operating as a member of your dev team—automating, optimizing, and accelerating every stage of the software lifecycle



Azure DevOps + GitHub

No longer separate choices

Azure DevOps with GitHub Repos
Your path to Agentic AI

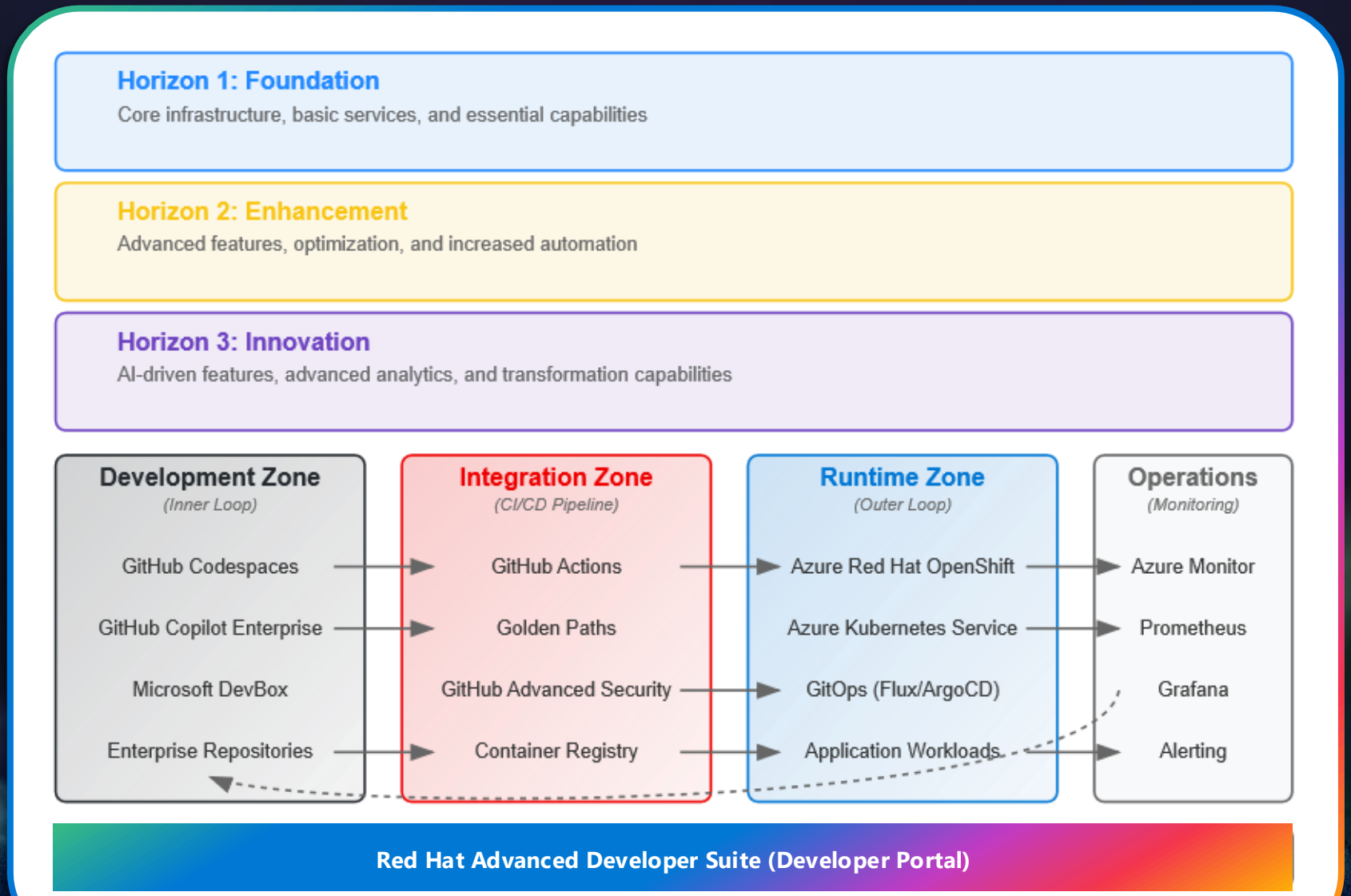
GitHub Copilot SWE Agent "Padawan"

AI-powered peer programmer, automating routine tasks and jobs to be done across the software lifecycle.

Azure SRE Agent

AI-powered operator, continuously monitoring and optimizing apps in production.

Platform architecture

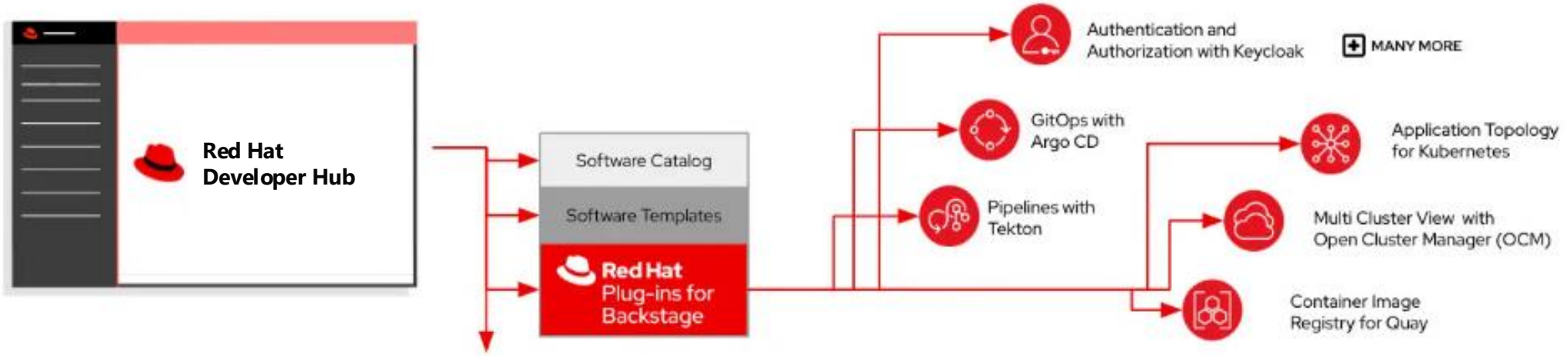


The Three Horizons Playbook for platform engineering provides a strategic framework for building, implementing, and evolving enterprise platform engineering capabilities across three-time horizons:

- **Horizon 1 (Present)**: Optimizing current platform capabilities, focusing on immediate value
- **Horizon 2 (Near Future)**: Developing emerging capabilities like internal developer platforms and golden paths
- **Horizon 3 (Future)**: Exploring transformative innovations including AI-powered developer experiences

Developer Hub

Implementing platform engineering



Integrates with industry standards and technologies through a broad ecosystem.

ai git GitHub **Copilot** Advanced Security Azure **DevOps**

Based on Backstage, an open source platform for building developer portals.

Backstage

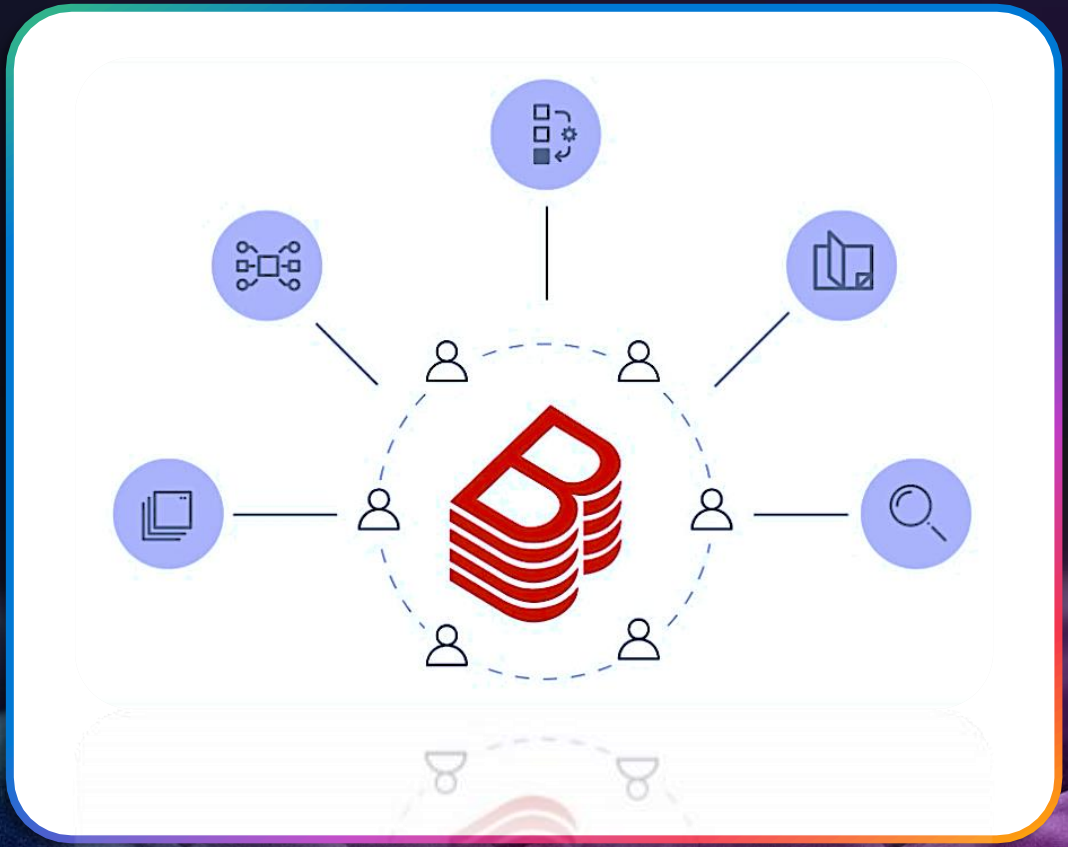
Consistent developer experience across environments.

Physical Virtual Hybrid

Microsoft Azure

OPENSHIFT by Red Hat

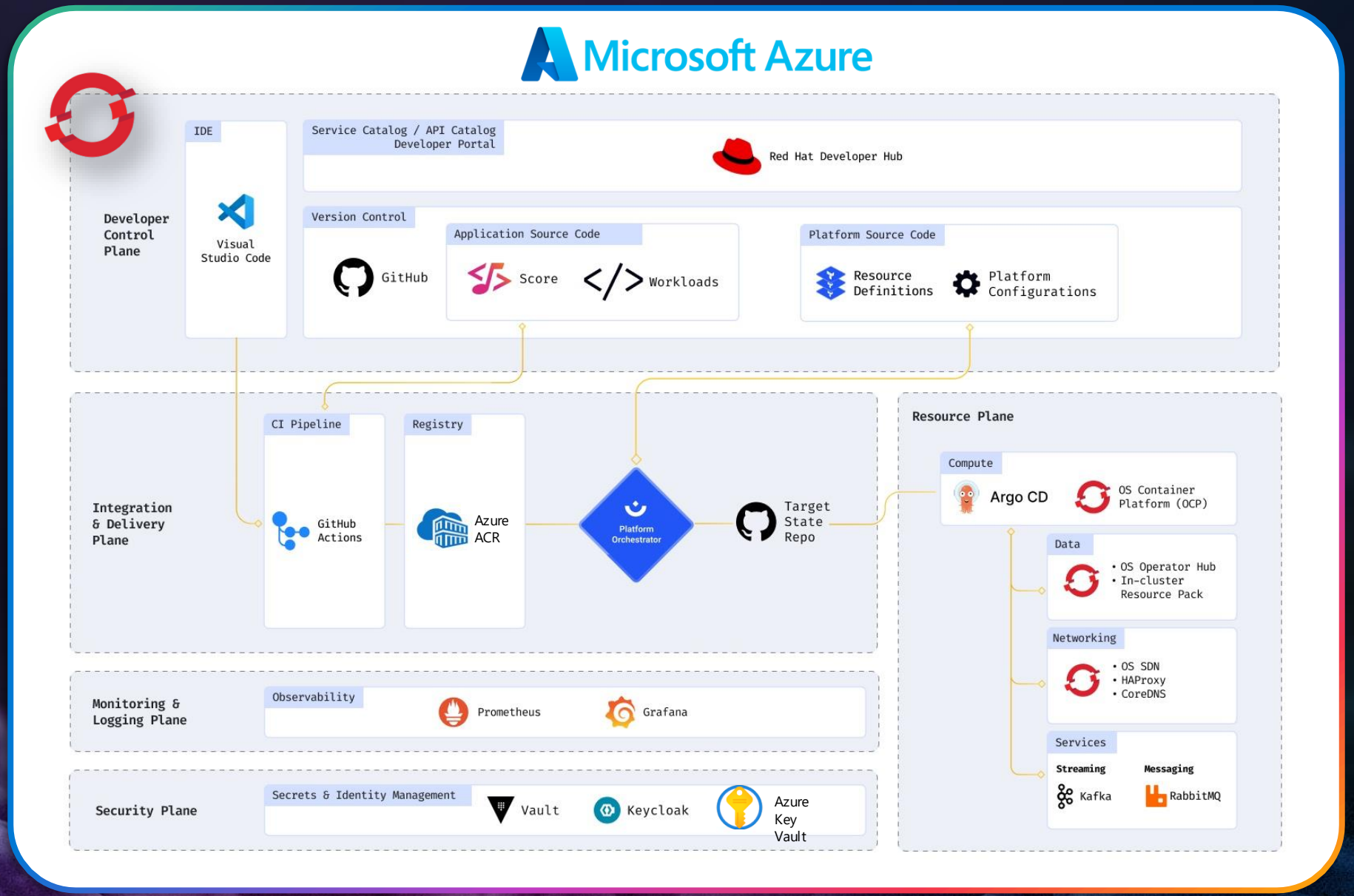
Extending Backstage functionality...



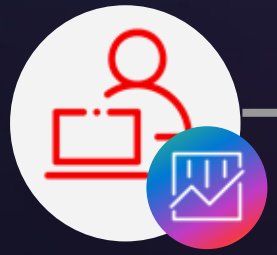
Red Hat Developer Hub and its associated plugins extend the upstream Backstage product by providing additional features such as integration with OpenShift, enterprise role-based access control (RBAC), and dynamic plugins.

- The ability to aggregate data from different sources, CI/CD, cloud providers, access management, source control, etc.
- A Software Catalog that functions as the central source to locate applications, APIs, and resources.
- An ecosystem of additional open-source plugins that extend Backstages' functionality.
- Software Templates to create templates to start a new application/microservice. Dozens of templates already exist, but enterprises can create and utilize their own custom to their environment.
- Tech Docs are simpler to update with Markdown and Git. Easily index content from plugins and other sources, searching across systems.

Internal Developer Platform on OpenShift



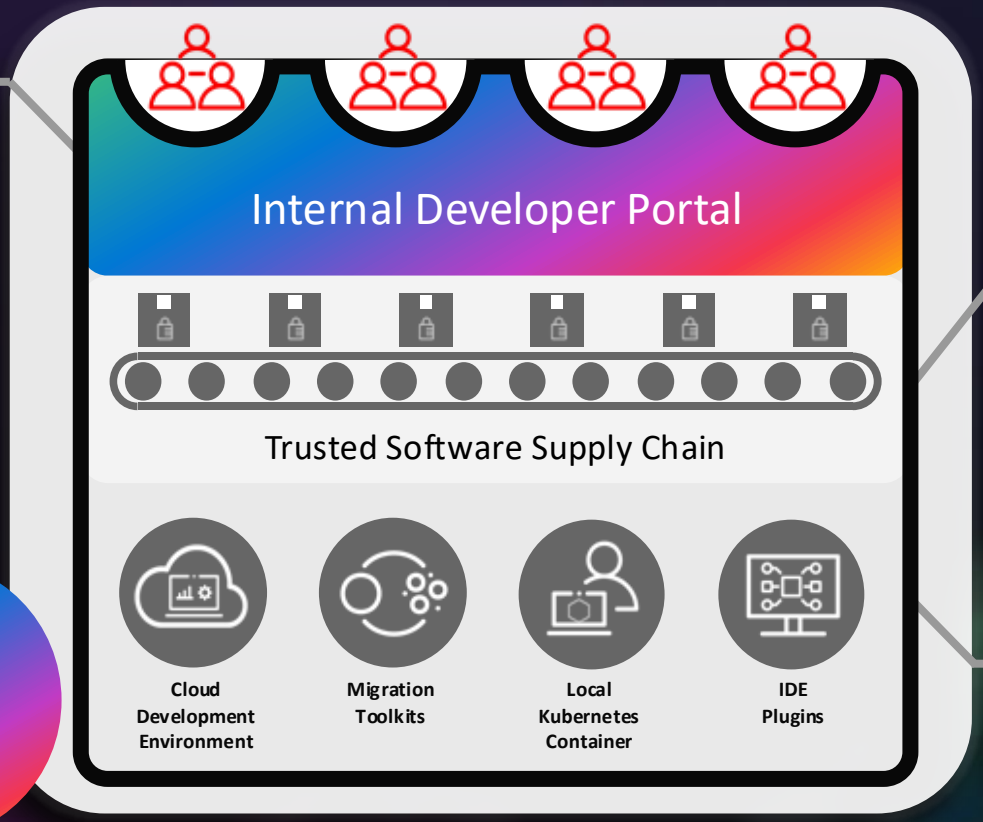
Red Hat Developer Hub



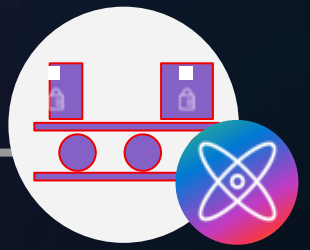
Increase Developer Productivity & Experience

Simplifies development with self-service access to a centralized source of truth, consistent workflows, automated templates.

Development Teams

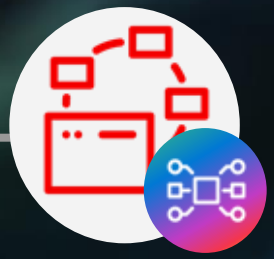


Platform Engineers



Enhanced Software Supply Chain Security

Delivers turnkey supply chain security with automated signing, provenance tracking, trusted content guidance, and real-time risk monitoring

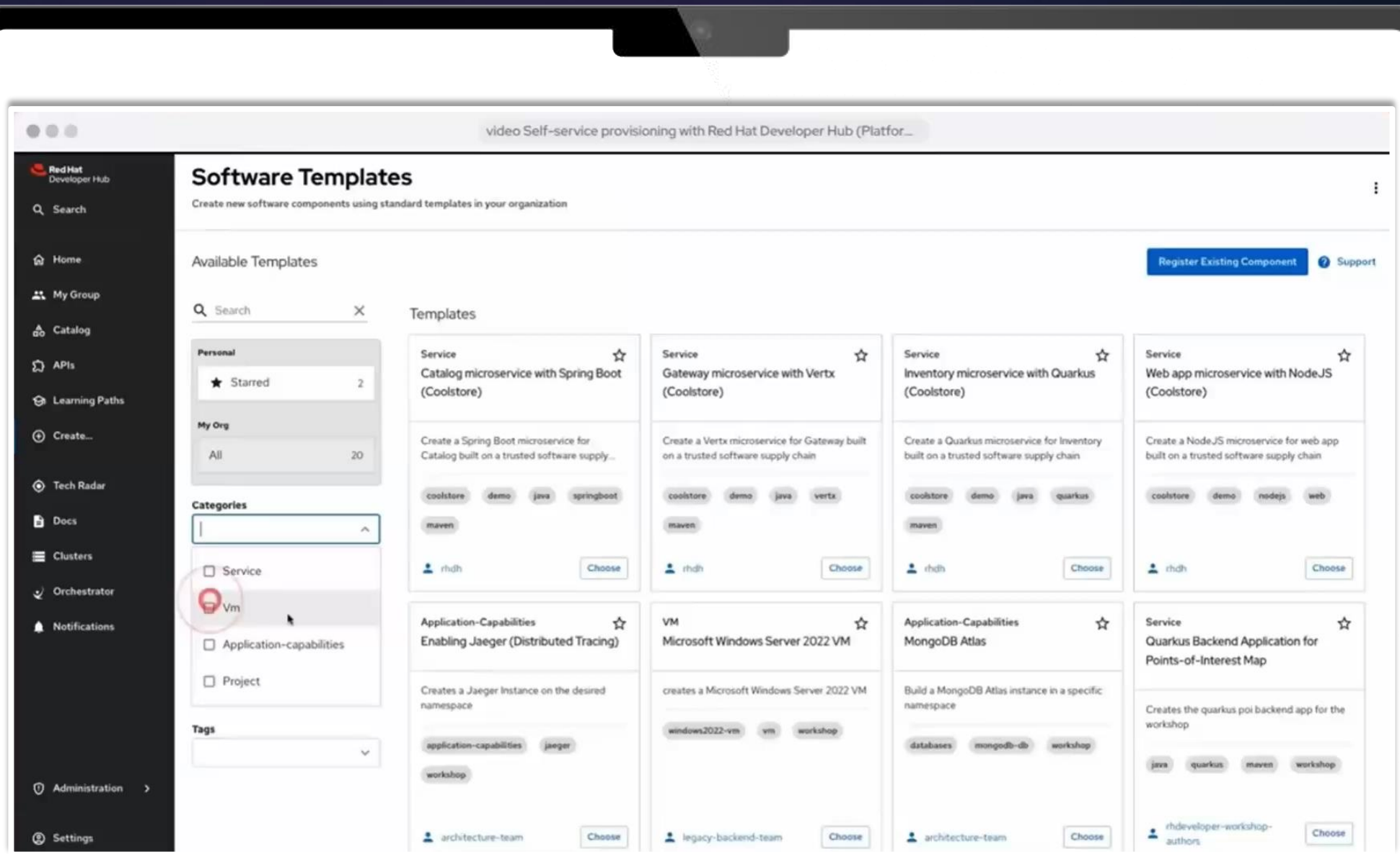


Modernization & Infrastructure Adaptability

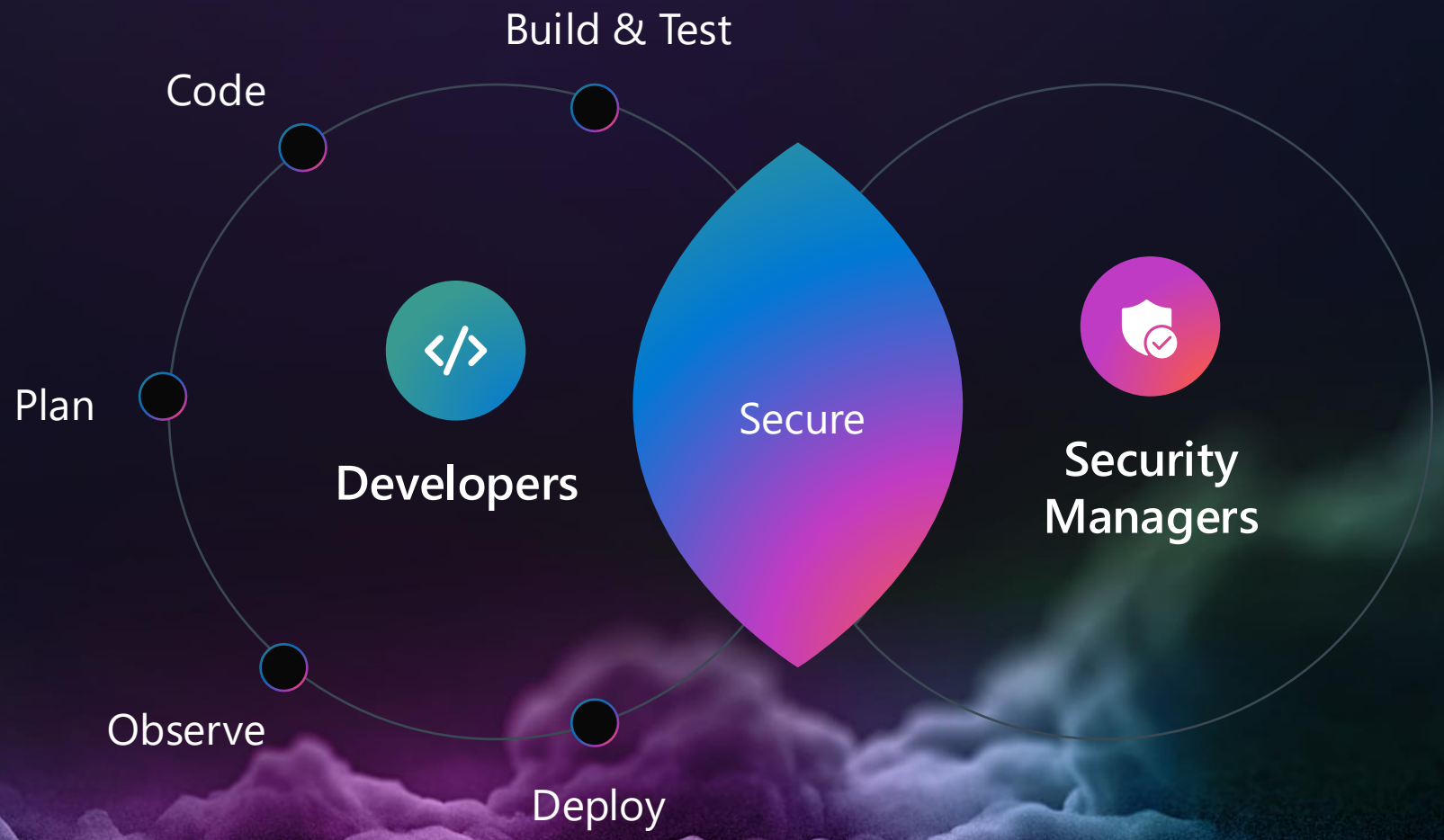
Streamlines modernization with CI/CD integration, automated workload migration, easy application upgrades, and secure, rootless container management.

Demo

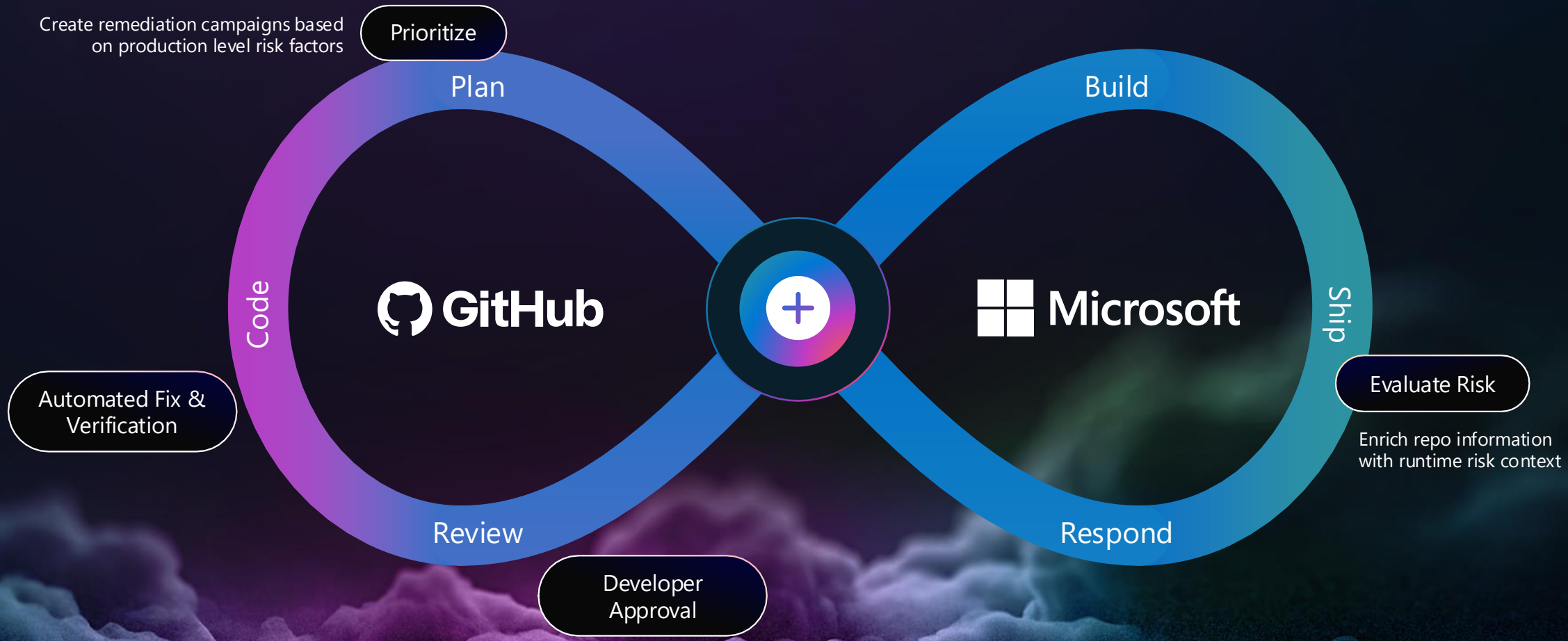
VM Deployment


video Self-service provisioning with Red Hat Developer Hub (Platfor...
Red Hat Developer Hub
Search
Home
My Group
Catalog
APIs
Learning Paths
Create...
Tech Radar
Docs
Clusters
Orchestrator
Notifications
Administration
Settings
Software Templates
Create new software components using standard templates in your organization
Available Templates
Register Existing Component
Support
Search
Personal
Starred 2
My Org
All 20
Categories
Service
Vm
Application-capabilities
Project
Tags
Templates
Service
Catalog microservice with Spring Boot (Coolstore)
Create a Spring Boot microservice for Catalog built on a trusted software supply...
coolstore demo java springboot maven
rhdh
Choose
Service
Gateway microservice with Vertx (Coolstore)
Create a Vertx microservice for Gateway built on a trusted software supply chain
coolstore demo java vertx maven
rhdh
Choose
Service
Inventory microservice with Quarkus (Coolstore)
Create a Quarkus microservice for Inventory built on a trusted software supply chain
coolstore demo java quarkus maven
rhdh
Choose
Service
Web app microservice with NodeJS (Coolstore)
Create a NodeJS microservice for web app built on a trusted software supply chain
coolstore demo nodejs web
rhdh
Choose
Application-Capabilities
Enabling Jaeger (Distributed Tracing)
Creates a Jaeger Instance on the desired namespace
application-capabilities jaeger workshop
architecture-team
Choose
VM
Microsoft Windows Server 2022 VM
creates a Microsoft Windows Server 2022 VM
windows2022-vm vm workshop
legacy-backend-team
Choose
Application-Capabilities
MongoDB Atlas
Build a MongoDB Atlas instance in a specific namespace
databases mongodb-db workshop
architecture-team
Choose
Service
Quarkus Backend Application for Points-of-Interest Map
Creates the quarkus poi backend app for the workshop
java quarkus maven workshop
rhdeveloper-workshop-authors
Choose

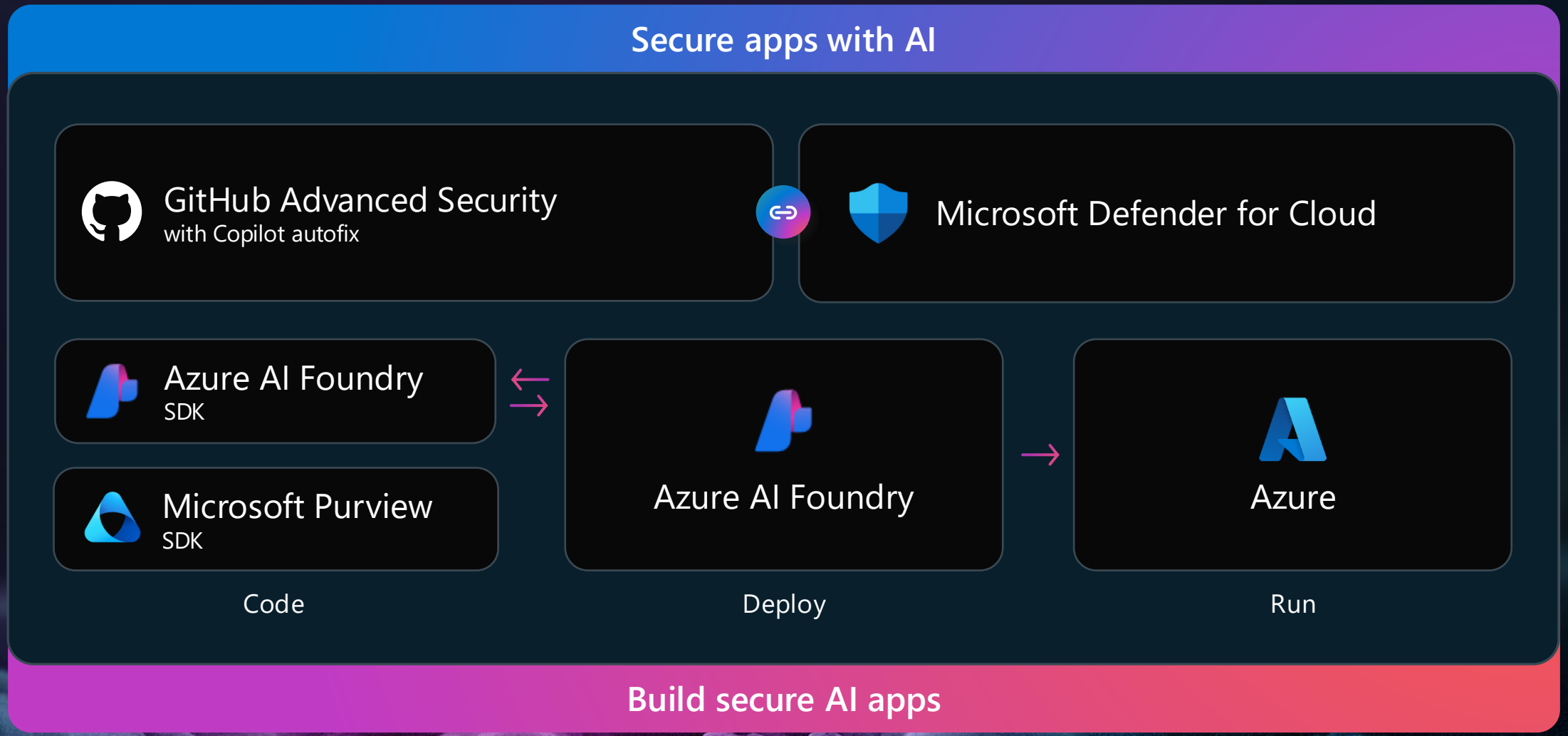
Securing Developer Workflow



Secure with AI from Code to Cloud

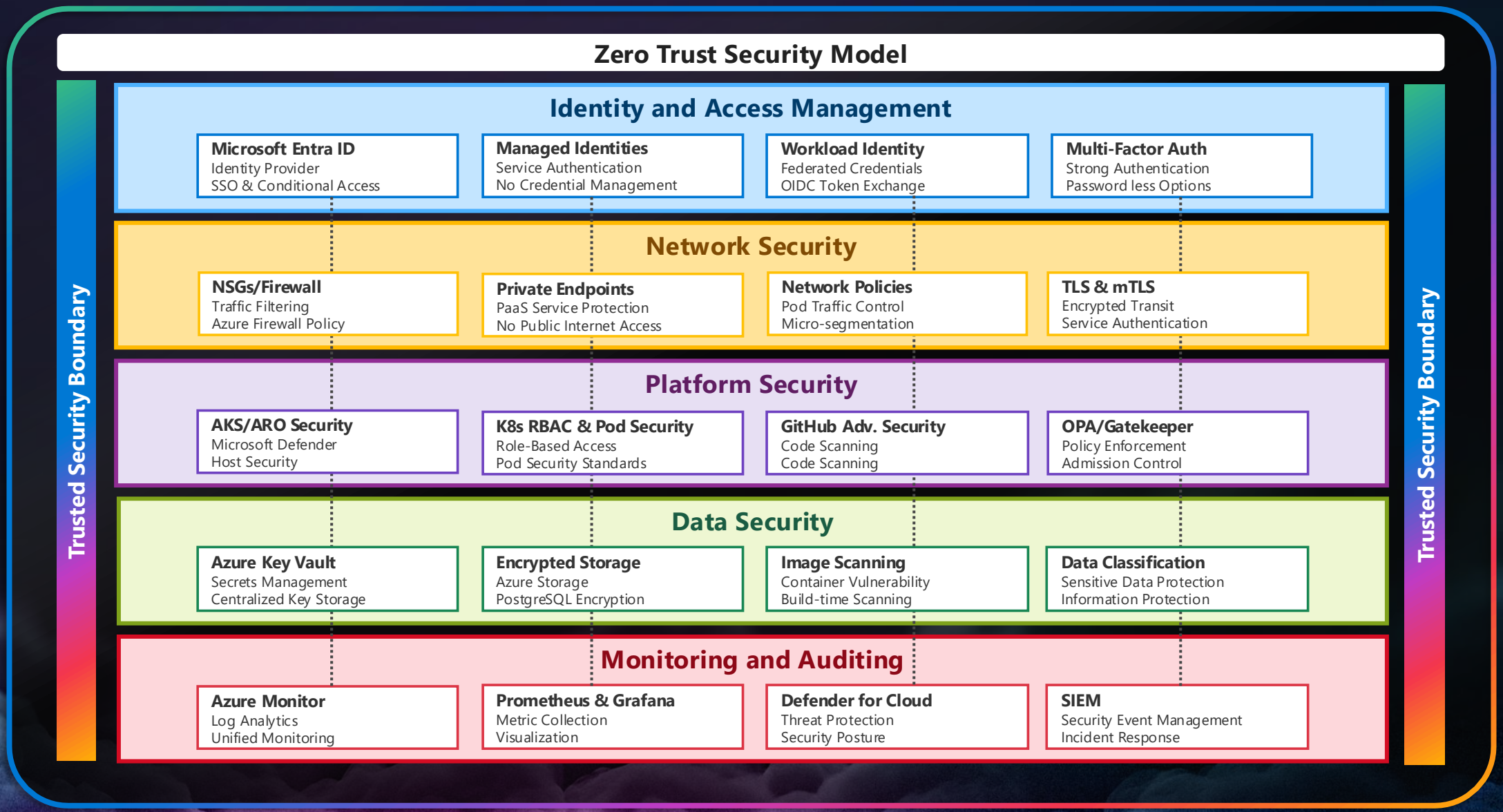


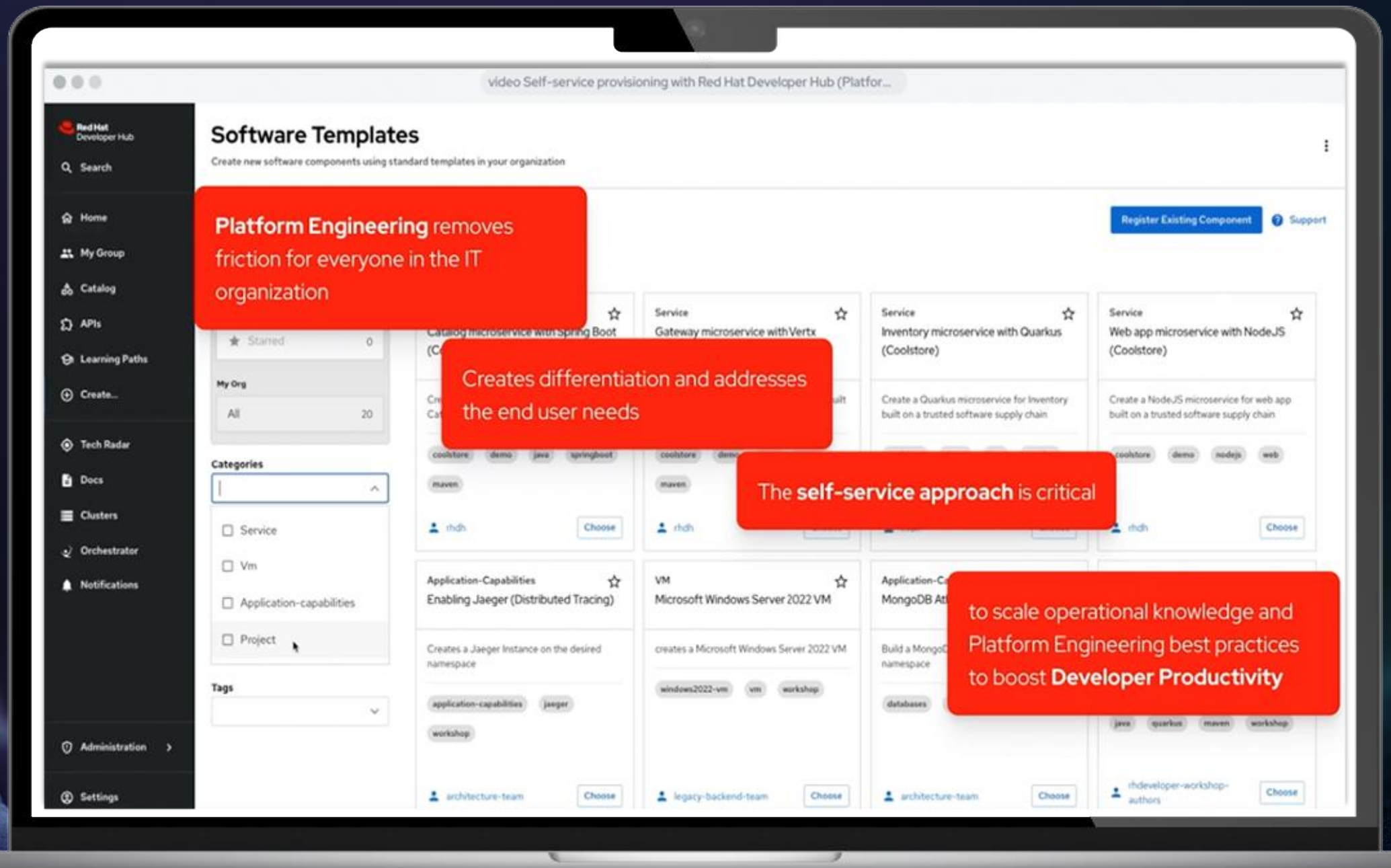
Secure AI apps from code to cloud



Develop apps securely with a unified solution

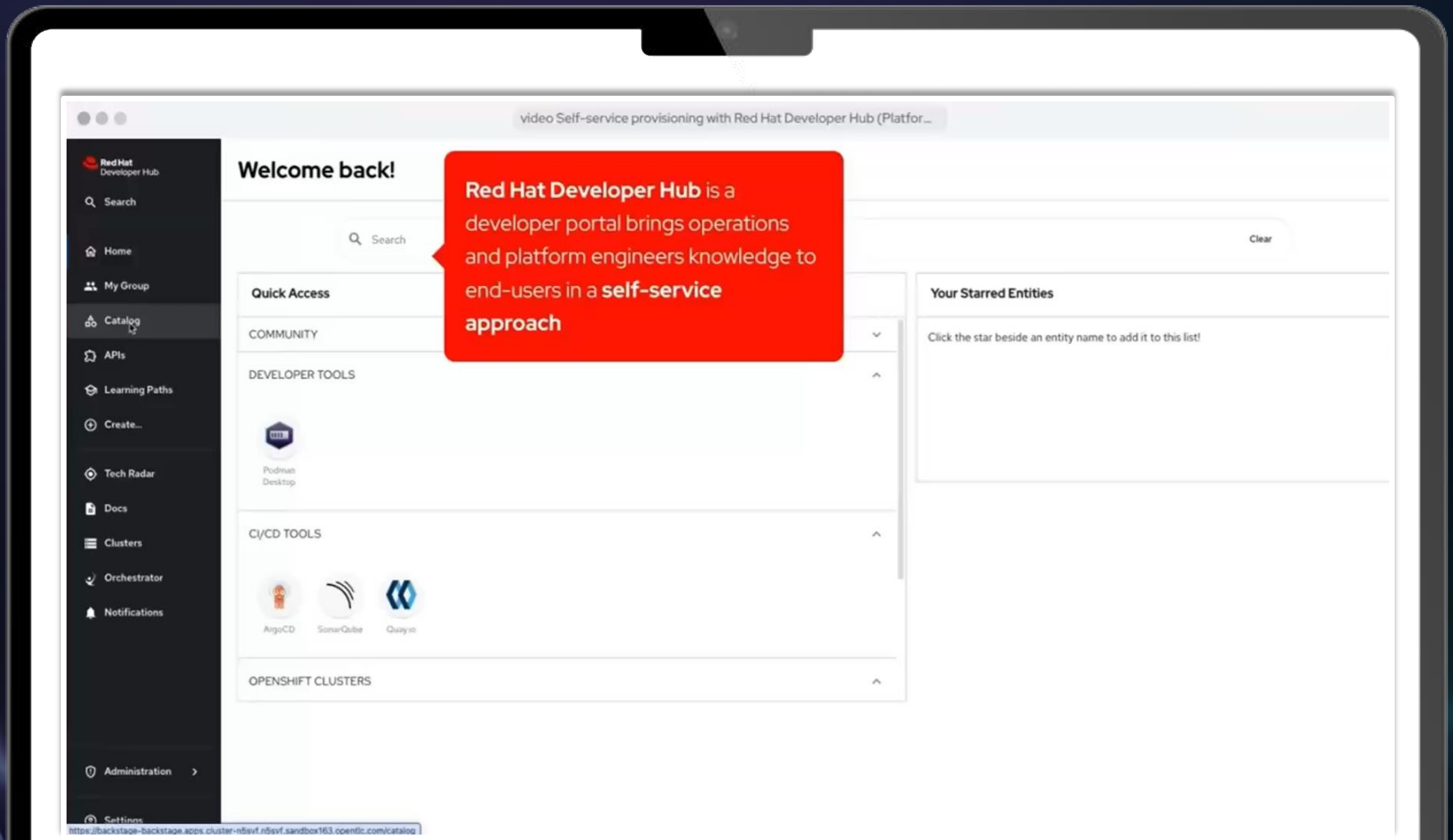
Security architecture




video Self-service provisioning with Red Hat Developer Hub (Platfor...
Red Hat Developer Hub
Search
Home
My Group
Catalog
APIs
Learning Paths
Create...
Tech Radar
Docs
Clusters
Orchestrator
Notifications
Administration
Settings
Software Templates
Create new software components using standard templates in your organization
Register Existing Component
Support
Platform Engineering removes friction for everyone in the IT organization
Starred 0
My Org
All 20
Categories
Service
Vm
Application-capabilities
Project
Tags
Service
Gateway microservice with Vertx
Service
Inventory microservice with Quarkus (Coolstore)
Create a Quarkus microservice for Inventory built on a trusted software supply chain
Service
Web app microservice with NodeJS (Coolstore)
Create a NodeJS microservice for web app built on a trusted software supply chain
Creates differentiation and addresses the end user needs
The self-service approach is critical
Application-Capabilities
Enabling Jaeger (Distributed Tracing)
Creates a Jaeger Instance on the desired namespace
VM
Microsoft Windows Server 2022 VM
creates a Microsoft Windows Server 2022 VM
to scale operational knowledge and Platform Engineering best practices to boost Developer Productivity
Choose

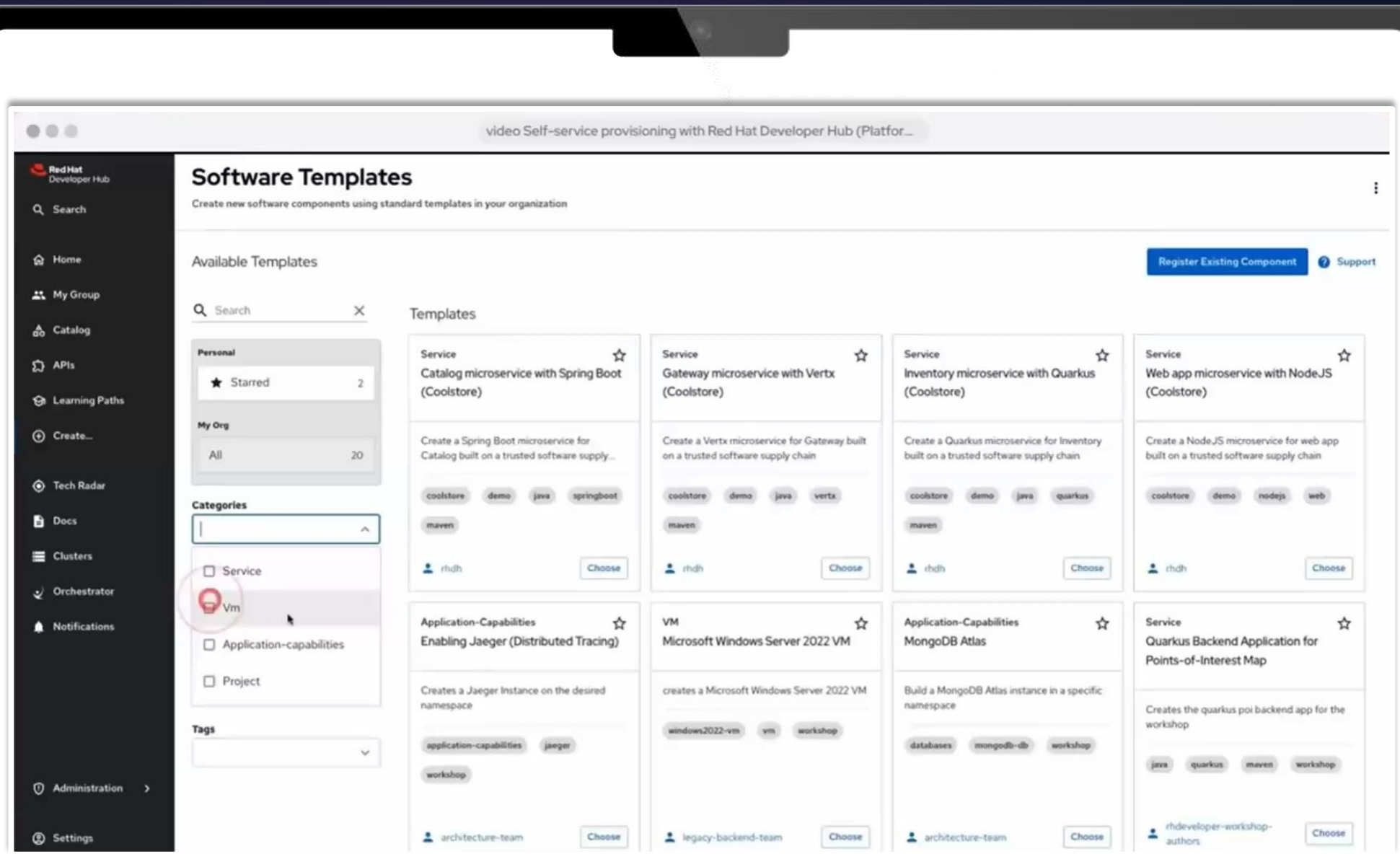
Demo

Software Catalog


video Self-service provisioning with Red Hat Developer Hub (Platfor...
Red Hat Developer Hub
Search
Home
My Group
Catalog
APIs
Learning Paths
Create...
Tech Radar
Docs
Clusters
Orchestrator
Notifications
Administration
Settings
Welcome back!
Red Hat Developer Hub is a developer portal brings operations and platform engineers knowledge to end-users in a self-service approach
Search
Clear
Quick Access
COMMUNITY
DEVELOPER TOOLS
Podman Desktop
CI/CD TOOLS
ArgoCD
SonarQube
Quay.io
OPENSHIFT CLUSTERS
Your Starred Entities
Click the star beside an entity name to add it to this list!

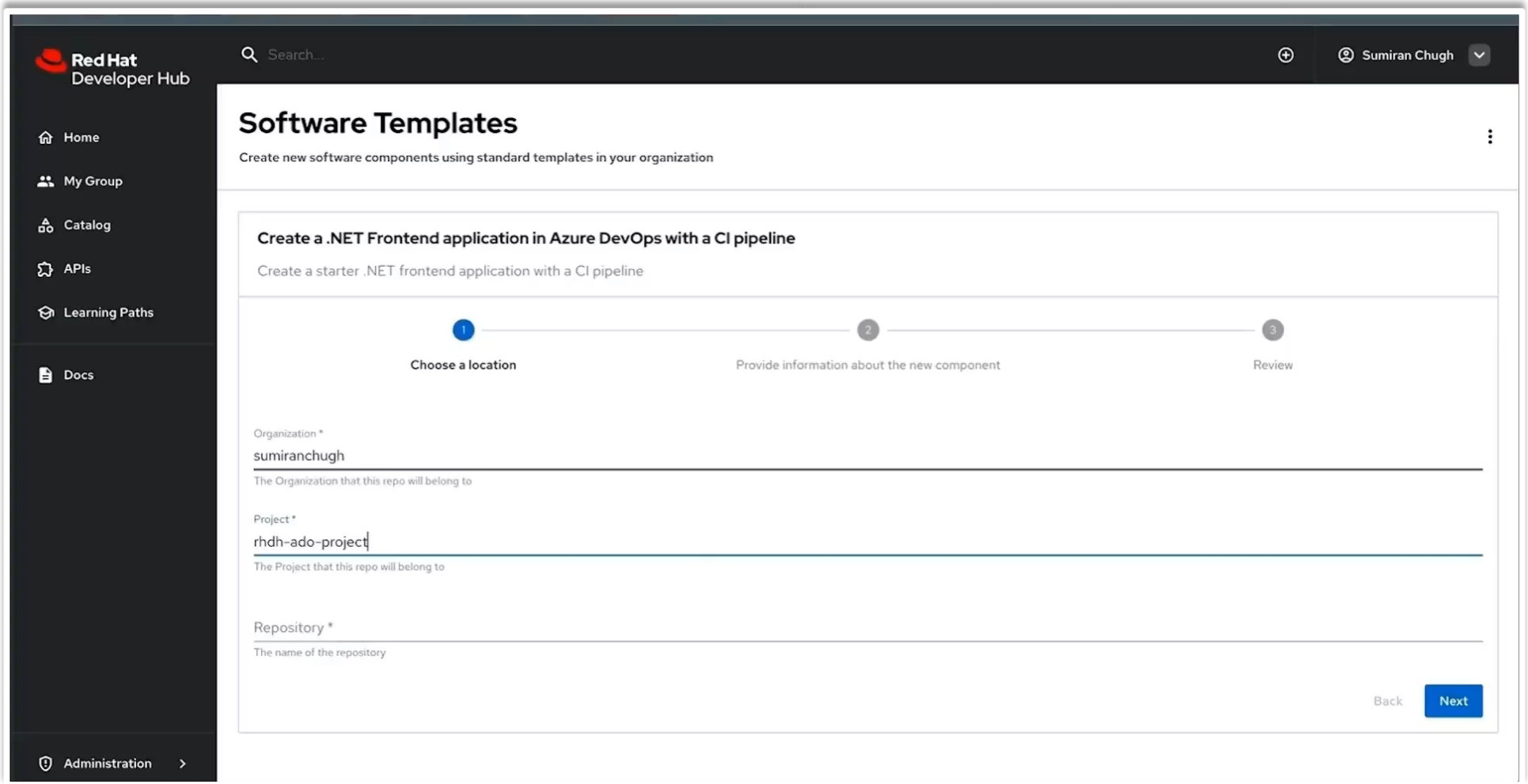
Demo

VM Deployment


video Self-service provisioning with Red Hat Developer Hub (Platfor...
Red Hat Developer Hub
Search
Home
My Group
Catalog
APIs
Learning Paths
Create...
Tech Radar
Docs
Clusters
Orchestrator
Notifications
Administration
Settings
Software Templates
Create new software components using standard templates in your organization
Available Templates
Register Existing Component
Support
Search
Personal
Starred 2
My Org
All 20
Categories
Service
Vm
Application-capabilities
Project
Tags
Templates
Service
Catalog microservice with Spring Boot (Coolstore)
Create a Spring Boot microservice for Catalog built on a trusted software supply...
coolstore demo java springboot maven
rhdh
Choose
Service
Gateway microservice with Vertx (Coolstore)
Create a Vertx microservice for Gateway built on a trusted software supply chain
coolstore demo java vertx maven
rhdh
Choose
Service
Inventory microservice with Quarkus (Coolstore)
Create a Quarkus microservice for Inventory built on a trusted software supply chain
coolstore demo java quarkus maven
rhdh
Choose
Service
Web app microservice with NodeJS (Coolstore)
Create a NodeJS microservice for web app built on a trusted software supply chain
coolstore demo nodejs web
rhdh
Choose
Application-Capabilities
Enabling Jaeger (Distributed Tracing)
Creates a Jaeger Instance on the desired namespace
application-capabilities jaeger workshop
architecture-team
Choose
VM
Microsoft Windows Server 2022 VM
creates a Microsoft Windows Server 2022 VM
windows2022-vm vm workshop
legacy-backend-team
Choose
Application-Capabilities
MongoDB Atlas
Build a MongoDB Atlas instance in a specific namespace
databases mongodb-db workshop
architecture-team
Choose
Service
Quarkus Backend Application for Points-of-Interest Map
Creates the quarkus poi backend app for the workshop
java quarkus maven workshop
rhdeveloper-workshop-authors
Choose

Demo

.Net with Azure DevOps


Red Hat
Developer Hub
Search...
Sumiran Chugh
Home
My Group
Catalog
APIs
Learning Paths
Docs
Administration
Software Templates
Create new software components using standard templates in your organization
Create a .NET Frontend application in Azure DevOps with a CI pipeline
Create a starter .NET frontend application with a CI pipeline
1
Choose a location
2
Provide information about the new component
3
Review
Organization *
sumiranchugh
The Organization that this repo will belong to
Project *
rhdh-ado-project
The Project that this repo will belong to
Repository *
The name of the repository
Back
Next

Onboarding


Developer work environment →
Procure corporate laptop
Install OS
Find team onboarding wiki
Resolving broken links
Working around out-of-date instructions
Request access to source code
Request access to cloud resources
Install & configure developer tools & SDK
Configure local network settings & VPN
Open & rebuild application to populate caches
Find out how to get access to a new test environment
Raise a ticket to request a new environment
Provide details about the application
← Application runtime environment
Clarify requirements with infrastructure teams, e.g., network, security, compliance
Go back and forth to align on requirements
Wait for admins to create and configure infrastructure
Validate whether the app code works on the test environment
Wait for admins to update or fix the infra configuration
Finally get access to the working infrastructure
Repeat the process for a new test environment
Integration Environment...
QA Environment...
Production Environment...
Prod Signoffs...
...

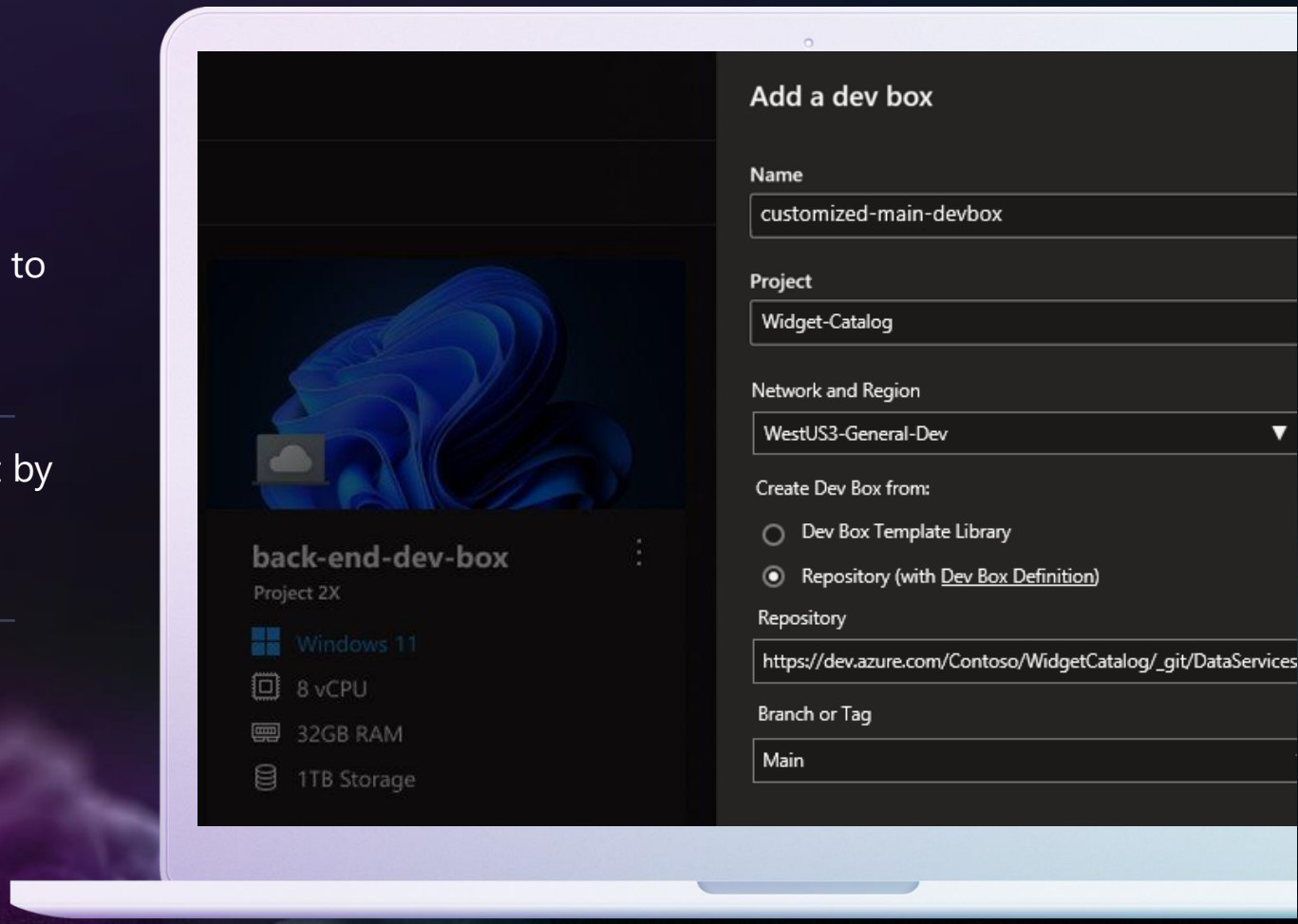
Microsoft Dev Box

Secure cloud workstations built for developer productivity

Ready-to-code: Self-service, on-demand access to task-specific workstations with scalable compute, available instantly.

Project-based: Preconfigured workstations built by dev teams with the right tools and resources for their projects.

Managed and secure: Centralized governance enables zero trust while maximizing security, compliance, and cost controls.



Microsoft Dev Box

Self-service at the Dev Portal

Dev Box 1: high-compute workspace

Dev Box 2: data engineer workspace

Developers

Deploy the Dev Boxes they need to work in their current projects

Configure dev boxes by project

Dev Box SKU

Cost controls

Toolset customization

Dev experience settings

Dev Leads

Create pools of Dev Boxes tailored to developers' projects and tasks

Centrally managed compliance

Network configurations

Security settings

Organizational policies

IT / Dev Infra Teams

Manage Dev Boxes via Intune and Microsoft Endpoint Manager

Demo

DevBox Integration

Chrome Arquivo Editar Visualização Histórico Favoritos Perfis Guia Janela Ajuda

Ter. 19 de ago. 16:41

Welcome back! | Red Hat De...

localhost:7007

Novo Chrome disponível

Search...

System Administrator

Home

Catalog

APIs

Learning Paths

Docs

Azure Dev Box

Administration

Welcome back!

Good afternoon System Administrator!

Get started

Learn about Red Hat Developer Hub.

Read documentation →

Explore

Explore components, APIs and templates.

Go to Catalog →

Learn

Explore and develop new skills.

Go to Learning Paths →

Explore Your Software Catalog

Browse the Systems, Components, Resources, and APIs that are available in your

RHDH Local

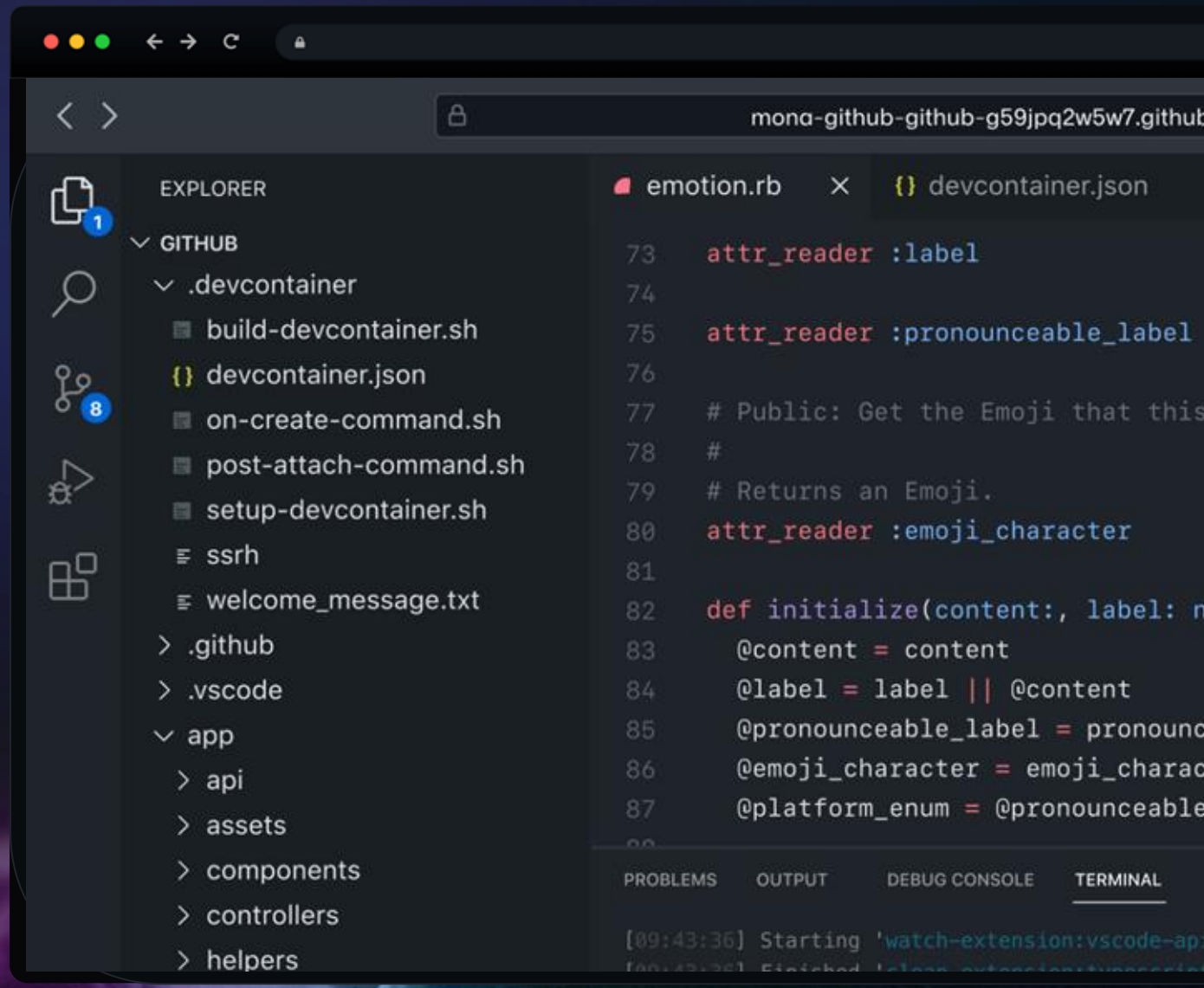
RHDH Local is used by platform engineers to test and experiment with RHDH configuration with a faster inner loop cycle time. RHDH Local uses Docker or PodMan to run the RHDH

localhost:7007/azure-devbox

GitHub Codespaces

Blazing fast cloud developer environments

Use Visual Studio Code, Jupyter, or JetBrains with the editor, terminal, debugger, version control, settings sync, and all of your extensions. Work from any device in a browser, or hand off to your desktop.



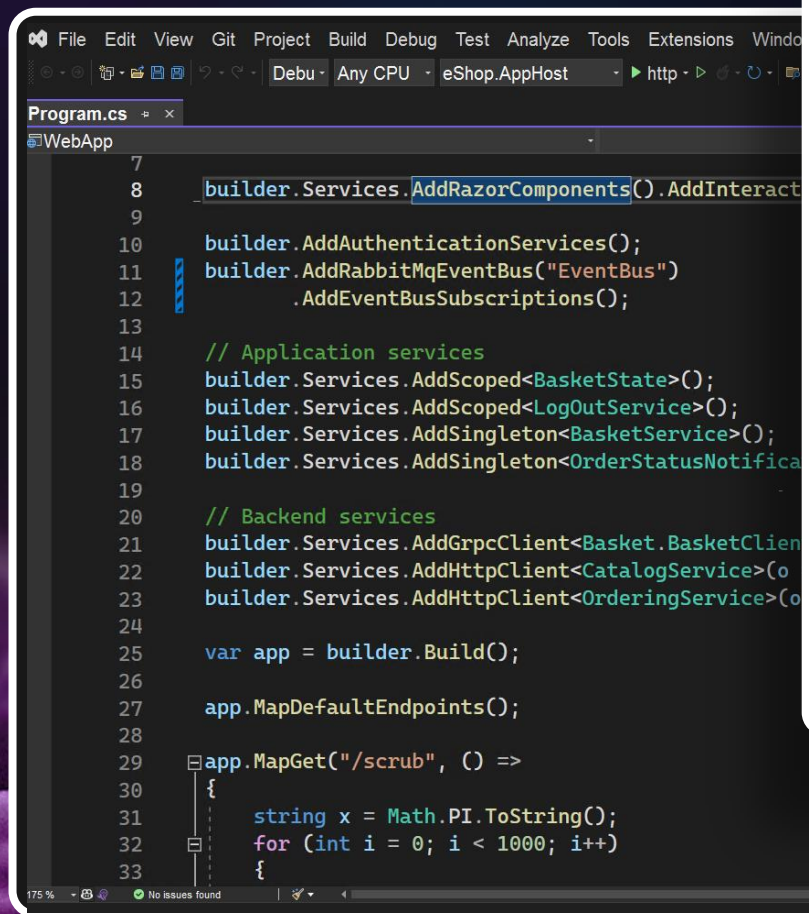
Inner Loop Developer Experience

Visual Studio 2022

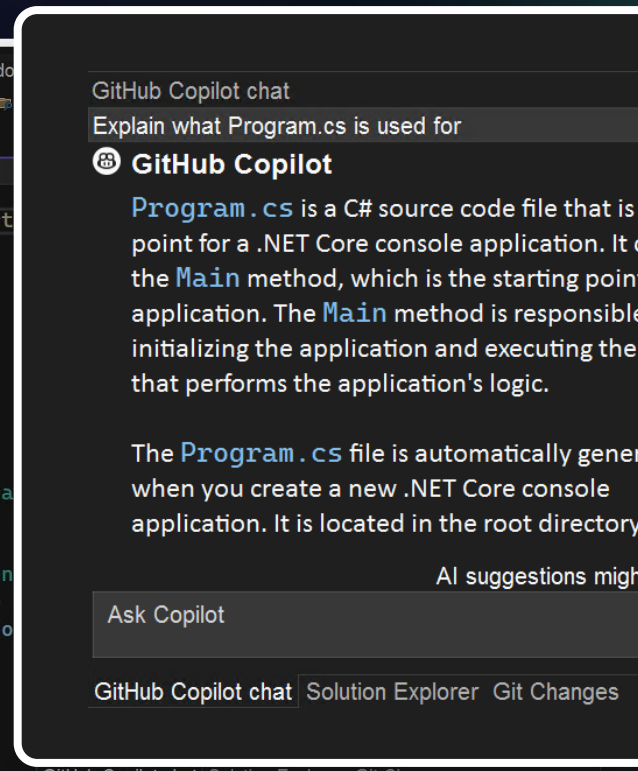
Optimized for Dev Box

- IDE and GCM single-sign-on
- Config as code for settings and extensions
- Prepopulated caches
- Detect and default to dev drives
- VS 2022 and 2019 pre-built images
- WinGet packages for VS
- Intune policies for VS

Microsoft Dev Box



GitHub Copilot



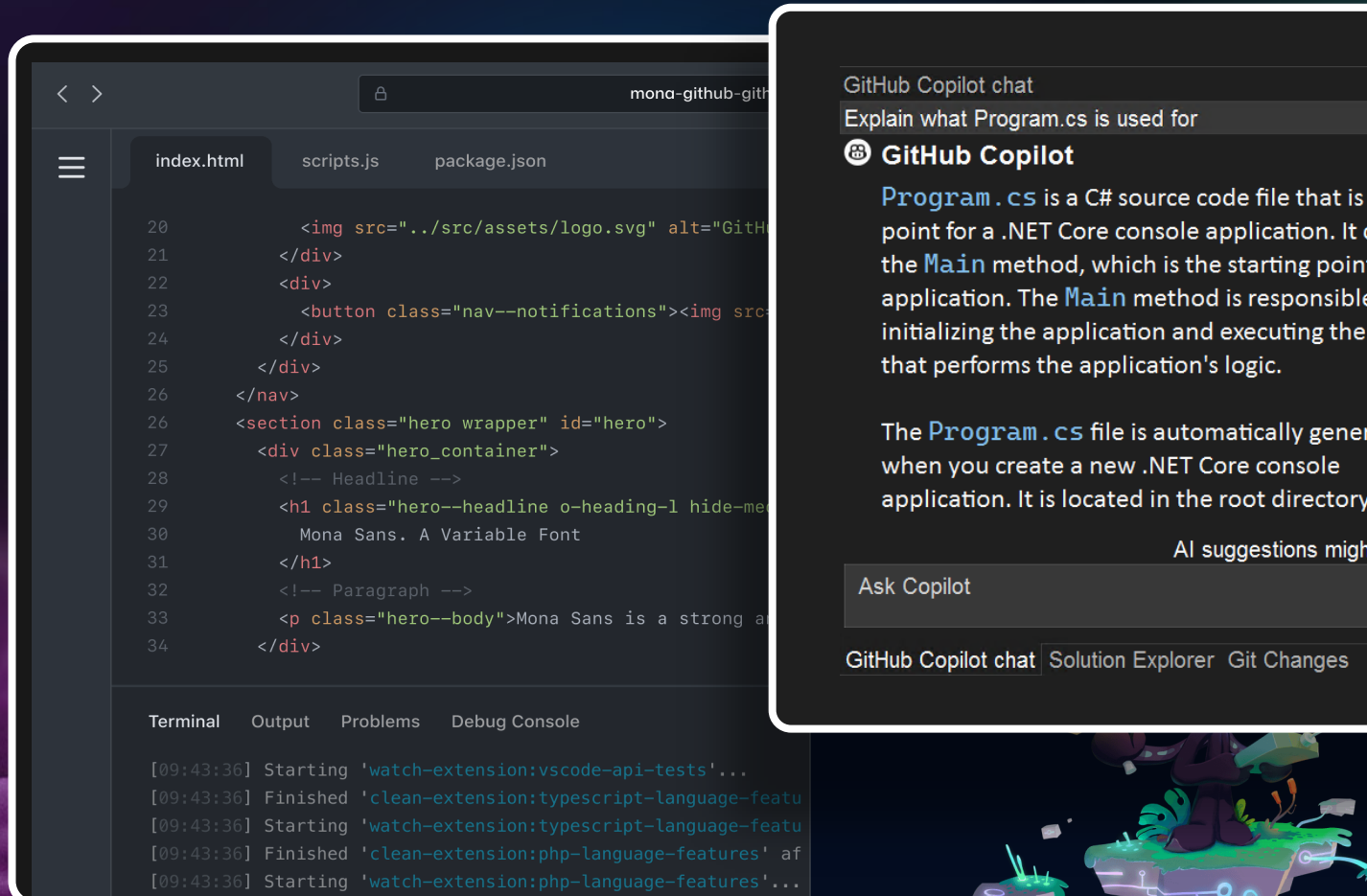
Inner Loop Developer Experience

Visual Studio Code

- Built-in Git
- Extensible, Customizable
- Any platform, any language
- IntelliSense
- Run & Debugging experience

GitHub Codespaces

GitHub Copilot



Benchmarking

Ability	Three Horizons	Fragmented Platforms	Custom Build Solution
Infrastructure	Native Azure integration with full automation	Multiple disconnected tools	Significant integration effort
App Innovation	Automated CI/CD with built-in modern practices	Technology-isolated pipelines	Requires constant maintenance and evolution
Data & AI	Simplified integration of AI data and models into applications	Silos between data and development teams	Complexity in integrating data and models
Governance	Unified policies and controls across stack	Gaps between platforms	Manual and inconsistent implementation
Implementation Time	Weeks with pre-integrated components	Months per component	Years to full maturity
Maintenance	Continuous evolution via established platforms	Overhead of multiple systems	High reliance on in-house competencies
Scalability	From MVP to enterprise scale without redesign	Limitations by fragmentation	Frequent redevelopment to scale

The three main pillars

RedHat Advanced Developer Suite

- Backstage-based Internal Developer Portal *(Self-Service)*

GitHub Platform

- Source control, GitHub Advanced Security. Actions CI/CD, and GitHub Copilot Enterprise, Coding Agent, GitHub Models
- CodeSpaces

Azure Infrastructure

- Developer Hub deployment on ARO *(Azure RedHat OpenShift)*
- Postgres Managed
- Defender for Cloud
- Azure Monitor, Grafana, Prometheus, Application Insights
- Azure AI Foundry
- Agentic Framework
- Azure SRE Agent
- Microsoft Purview
- Microsoft Sentinel
- Azure Arc for Conteiners *(Multi-cloud and hybrid deployment scenarios)*
- Microsoft DevBox (Optional)

Azure + GitHub as Development & Productivity Hub

- **Tool Consolidation**
- **Environment Standardization**
- **Pipeline Templates**
- **Developer Portal**
  Red Hat Developer Hub
- **GitHub Copilot Integration**
- **Unified Development Experience**
- **Code Quality Automation**
- **Developer Self-Service**

Azure + GitHub as Security Control Plane

- **Shift-Left Security**
  GHAS
- **AI-Powered Remediation**
  Copilot Autofix
- **Unified Security Platform**
- **Compliance Automation**
- **Vulnerability Management**
- **Security Policy Enforcement**
- **Threat Detection & Response**
- **Security Metrics & Reporting**

Observability & Intelligent Operations

- **Unified Monitoring**
  Azure Monitor
- **AI-Powered Operations**
  SRE Agent
- **Predictive Analytics**
- **Self-Healing Workflows**
- **Performance Optimization**
- **Incident Management**
- **Capacity Planning**
- **Operational Intelligence**

Call to Action

Learn how Three Horizons can streamline your development process and deliver measurable improvements in productivity and efficiency. Schedule a consultation to discuss your specific requirements and see a demonstration of the platform capabilities.

Transform Development. Transform Business. Transform Everything.