

# Internal Developer Portal

Self-Service Developer Portal for Faster Onboarding & Streamlined Deployments

## CHALLENGES

Onboarding developers into complex CI/CD environments is often slow and manual, involving multiple tools, scripts, and approvals. Teams face high cognitive load navigating scattered systems for deployments, documentation, and environment provisioning. This leads to reduced productivity, inconsistent workflows, and constant dependency on DevOps teams for routine tasks. Without standardization and self-service, innovation slows and operational overhead grows.

## INTERNAL DEVELOPER PORTAL

**OpsMx Internal Developer Portal (IDP)** simplifies onboarding and streamlines developer workflows through a self-service portal powered by open source **Backstage**. It centralizes access to CI/CD pipelines, microservices, templates, and documentation—allowing developers to provision environments, register apps, and manage deployments independently. With built-in RBAC, policy enforcement, and CNOE-based integration, the portal standardizes DevOps best practices while boosting developer velocity. OpsMx helps teams shift left and scale efficiently—without sacrificing governance or control.



## FEATURES



## Developer Portal Based on CNOE

- Built on Cloud-Native Open Enterprise (CNOE) standards to ensure seamless DevOps integration.
- Standardize best practices for onboarding, deployment, and service management across teams.

The screenshot shows the 'CNOE Catalog' interface. On the left is a sidebar with navigation links: Home, APIs, Docs, Create..., Tech Radar, and Settings. The main content area is titled 'CNOE Catalog' and includes a search bar, a 'CREATE' button, and a 'SUPPORT' link. Below these are filters for 'Kind' (set to 'Template') and 'Type' (set to 'all'). A 'PERSONAL' section shows 'Owned' (0) and 'Starred' (0) items. A 'CNOE' section shows 'All' (7) items. The main list, titled 'All templates (7)', contains the following data:

NAME	TYPE	DESCRIPTION	TAGS	ACTIONS
create a new github repo and onboard it to argocd	service	creates a github...		[icon] [icon] [icon]
Onboard a basic microservice as an argocd Application	service	Using the name...		[icon] [icon] [icon]
Onboard a helm-chart with local override values	service	Helm chart		[icon] [icon] [icon]
Onboard a ISD-Helm-chart with local override values	service	ISD-Helm chart		[icon] [icon] [icon]
Onboard a vCluster and adds it to argocd	service	Onboard a vClus...		[icon] [icon] [icon]
Onboard multiple microservices as ONE ArgoCD Application with Multiple Sources	service	Create Multi...		[icon] [icon] [icon]
Onboarding argocd applicationset that creates one argocd application per environment in specified stages with or without manual approval	other	takes input the...		[icon] [icon] [icon]



## Application Onboarding

- Automate service and application registration in a centralized service catalog.
- Provide a self-service workflow for onboarding new applications with minimal manual intervention.

The screenshot shows the 'Create a new component' interface. The sidebar is the same as in the previous screenshot. The main content area is titled 'Create a new component' and includes a subtitle 'Create new software components using standard templates in your organization'. Below this is a section titled 'Onboard a basic microservice as an argocd Application' with a description: 'Using the name field the contents of skeleton folder are copied to workspace after substituting for values, the workspace is pushed into a newly created github repo, an argocd application with the same name is created'. A progress bar shows two steps: '1 Configuration Options' and '2 Review'. The '1 Configuration Options' step is active. Below the progress bar are two input fields: 'name \*' (with a hint 'name of this application') and 'image' (with a hint 'docker image with tag, example nginx:1.25.0'). At the bottom right are 'BACK' and 'REVIEW' buttons.



## Repo Onboarding

- Add to Argo CD the repos in which the code for Kubernetes manifest exists
- Add webhooks to prevent rate-limiting issues
- Validating the YAML files to prevent code-errors when deployed to Argo CD

**Create a new component**  
Create new software components using standard templates in your organization

**create a new github repo and onboard it to argocd**  
creates a github repo given the name, creates a webhook to argocd endpoint, then adds it to argocd given github credentials from aws secret

Progress: 1 (New repository information) | 2 (Review)

**name \***  
Repository name

**description**  
Repository description

**awsSecret \***  
AWS secret name from where to create secret to add the repo to argocd

**argocd project**  
argocd project to which this repo will be associated

BACK REVIEW



## Automated Infrastructure Provisioning

- Enable developers to provision Kubernetes, cloud, and on-prem environments on demand.
- Reduce DevOps workload by automating infrastructure setup and environment allocation.

**Create a new component**  
Create new software components using standard templates in your organization

**Onboard a vCluster and adds it to argocd**  
Onboard a vCluster and adds it to argocd

Progress: 1 (Configuration Options) | 2 (Review)

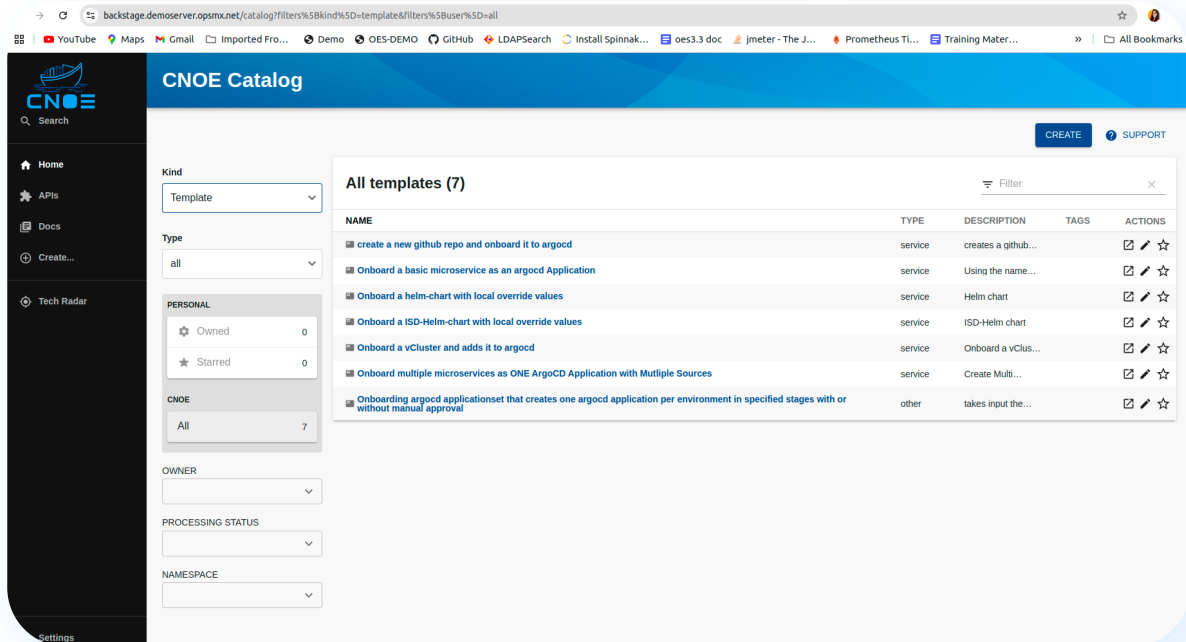
**name \***  
name of this application

BACK REVIEW



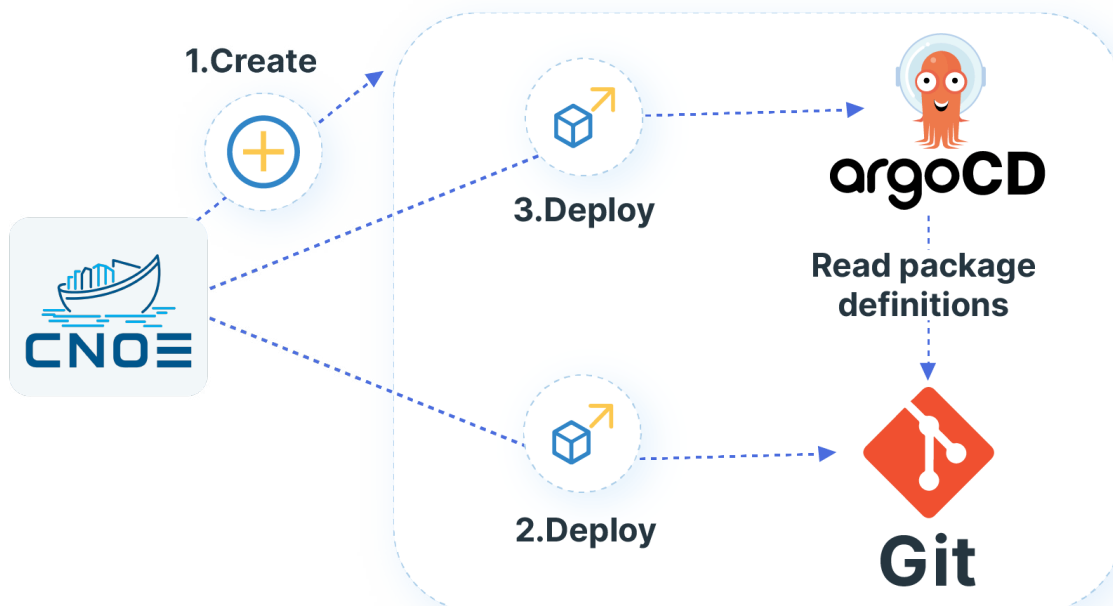
## Deployment Templates

- Access pre-configured deployment templates for Kubernetes, multicloud, and hybrid environments.
- Ensure consistent, secure, and repeatable application deployments with reusable workflows



## Backstage Support & Lifecycle Management (LCM)

- Manage Backstage instances, plugins, and lifecycle processes from a single control plane.
- Enable service discovery, developer documentation, and operational workflows through Backstage.



## KEY BENEFITS



### 5X faster

Faster developer onboarding & productivity. Automate service registration and infrastructure provisioning to reduce setup time. Enable developers to start coding and deploying without DevOps bottlenecks.



### Standardize CI/CD

CI/CD Workflows use pre-approved deployment templates for consistent application rollouts. Maintain governance with CNOE-backed security and compliance policies.



### Reduce Overheads by 80%

Reduction in Ops overhead with self-service automation. Allow developers to manage deployments and environments independently. Free up Ops teams from manual infrastructure setup and configuration requests.

## ABOUT US

OpsMx simplifies and accelerates comprehensive application security from developer to deployment building on AI-driven automation, open source security tools, and a deep understanding of the software delivery process. OpsMx enables tens of thousands of developers at both fast moving innovators and global enterprises to deliver more secure software faster.

## FOR MORE INFORMATION, CONTACT US:

OPSMX, INC | 350 OAKMEAD PKWY, SUNNYVALE, CA 94085 | [INFO@OPSMX.COM](mailto:INFO@OPSMX.COM)

[WWW.OPSMX.COM/SECURE SOFTWARE DELIVERY](https://www.opsmx.com/secure-software-delivery)