

AnvilOps: Increasing Accessibility of Kubernetes with Automated Builds and Deployments

Brendan Swanson, Emma Zheng, LJ Lumas, Haniye Kashgarani



Rosen Center for
Advanced Computing



Anvil is supported by the National Science Foundation
under Grant No. 2005632.



Kubernetes and Composable Infrastructure

Composable Infrastructure: hardware resources are pooled and dynamically allocated to workloads

Kubernetes: A platform for orchestrating containerized services.

- Dynamic allocation of compute, storage, network resources
- Autoscaling and self-healing for robust applications
- Resource quotas and role-based access control

Useful for:

- Interactive data analysis (e.g. Jupyter notebooks)
- Big Data pipelines
- Persistent services

Running Composable Workloads

At the Rosen Center:

- Anvil Composable Subsystem (NSF award #2005632)
- Geddes Composable Platform (NSF award #2018926)

Tech stack:

- Kubernetes (container orchestration)
- Rancher (cluster management)
- Harbor (image registry)

Deploying applications to Kubernetes is:

- Time-consuming
- Error-prone
- Inaccessible to beginners

Guiding Principles for AnvilOps

- Simplify the build process
- (Re)deploy resources automatically
- Least required configuration possible

Demo!

The screenshot shows a web browser window with the URL `anvilops.rcac.purdue.edu`. The page header features the **AnvilOps** logo and the text *Anvil Composable*, along with links for **Docs** and **Sign In**.

Deploy your app in seconds.

AnvilOps lets you deploy your app to Anvil Composable without writing Kubernetes manifests.

Deploy Now

Seamless Git Integration

AnvilOps continuously keeps your deployment in sync with your Git repository. Here's how it works:

-  **1** You push a commit to your Git repository
-  **2** AnvilOps builds a container image from your code
-  **3** Your app is automatically deployed in the Anvil Composable cluster

Kubernetes is hard. Let us handle it for you.

Tech Stack

Frontend

TailwindCSS
shadcn/ui

React
Vite, React Query, React Router

 OpenAPI

Backend

Node.js
Express, TypeScript, Prisma

 Postgres

Building

 GitHub Apps

 Railpack

 BuildKit

 Harbor

 Kubernetes

 Rancher

Features

- Deploy applications with minimal configuration
- Rebuild and redeploy after a commit is pushed or tests run
- Dockerfile-free image builds
- Scale deployments up and down
- Monitor application status and logs in real-time
- Deployed applications count towards users' resource quotas
- Distributed as a Helm chart

Looking Ahead

Improvements we'd like to see in AnvilOps

- Singularity support
- Built-in monitoring and HTTP request logging
- Log rotation and retention policy
- Web-based SSH into running containers
- Creating ephemeral deployments from pull requests
- Support for non-HTTP services
- Health check configuration
- High availability database templates
- Support more Git providers
- Custom domain support
- Command-line interface
- Cron jobs
- Running BuildKit builds with fake root or user namespaces
- Evaluate and improve user experience / accessibility

Thank You

- Mentors: LJ Lumas and Haniye Kashgarani
- Purdue University's Rosen Center for Advanced Computing
- National Science Foundation

🔗 docs.anvilcloud.rcac.purdue.edu

🔗 github.com/PurdueRCAC/AnvilOps

✉ swans150@purdue.edu, zheng861@purdue.edu