

Facilitating Web Service Deployment in LC's Cloud Ecosystem

Urwah Mir (LLNL), J. Consolati (LLNL), J. Long (LLNL), C. Cook (LLNL), T. Heer (LLNL), T. Mendoza (Matalino Software), N. O' Neil (LLNL), O. Venezuela (LLNL), K. Wong (UCSD)

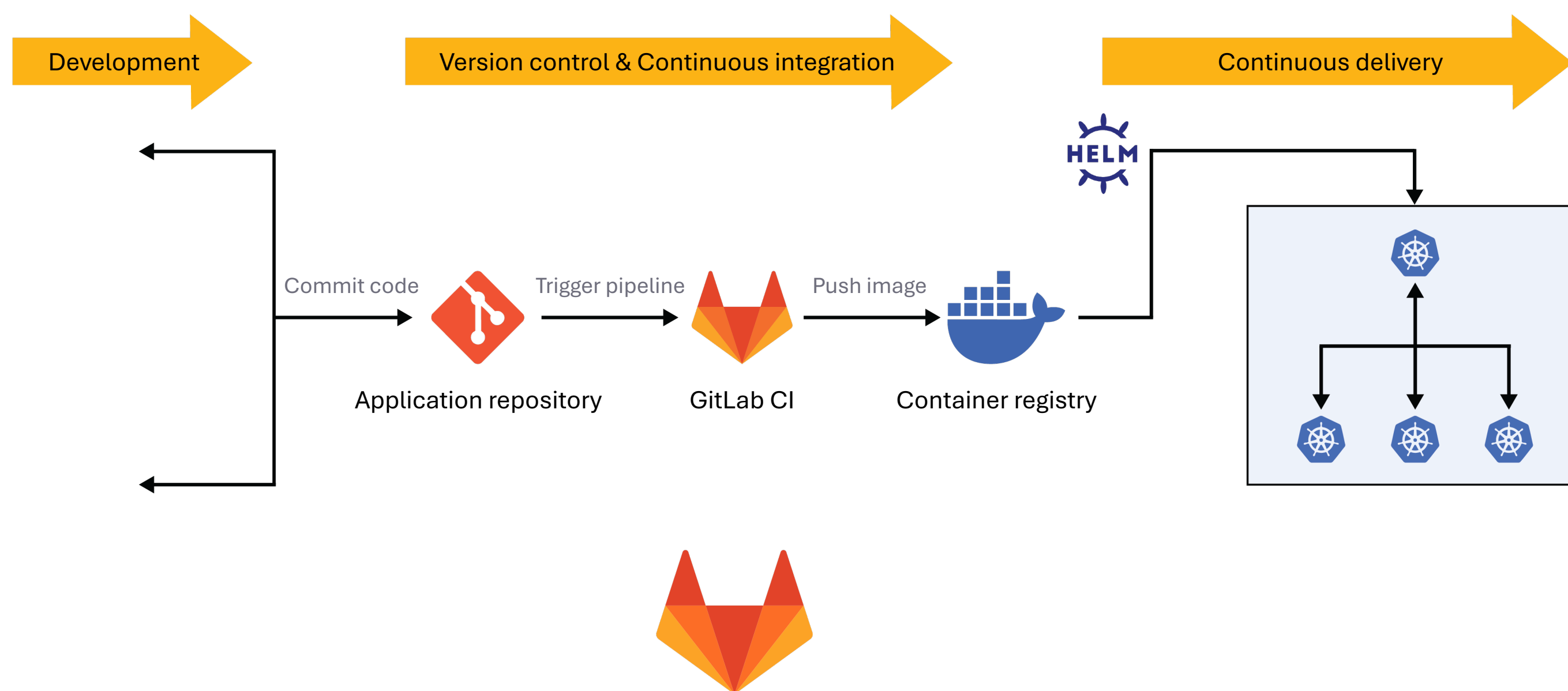
Modern web technologies evolve rapidly, and users increasingly need more services at a pace that traditional infrastructure cannot support. This growing number of services require ongoing maintenance and time-sensitive cybersecurity patches, all of which must be managed with minimal impact to users – by a team of **4 FTE**.

To address these challenges and simplify service management, our team has adopted **DevOps and Cloud/OpenShift application deployments**.

Streamlined Deployments & Upgrades

GitLab automated pipelines leverage containers to streamline application deployment and accelerate upgrades:

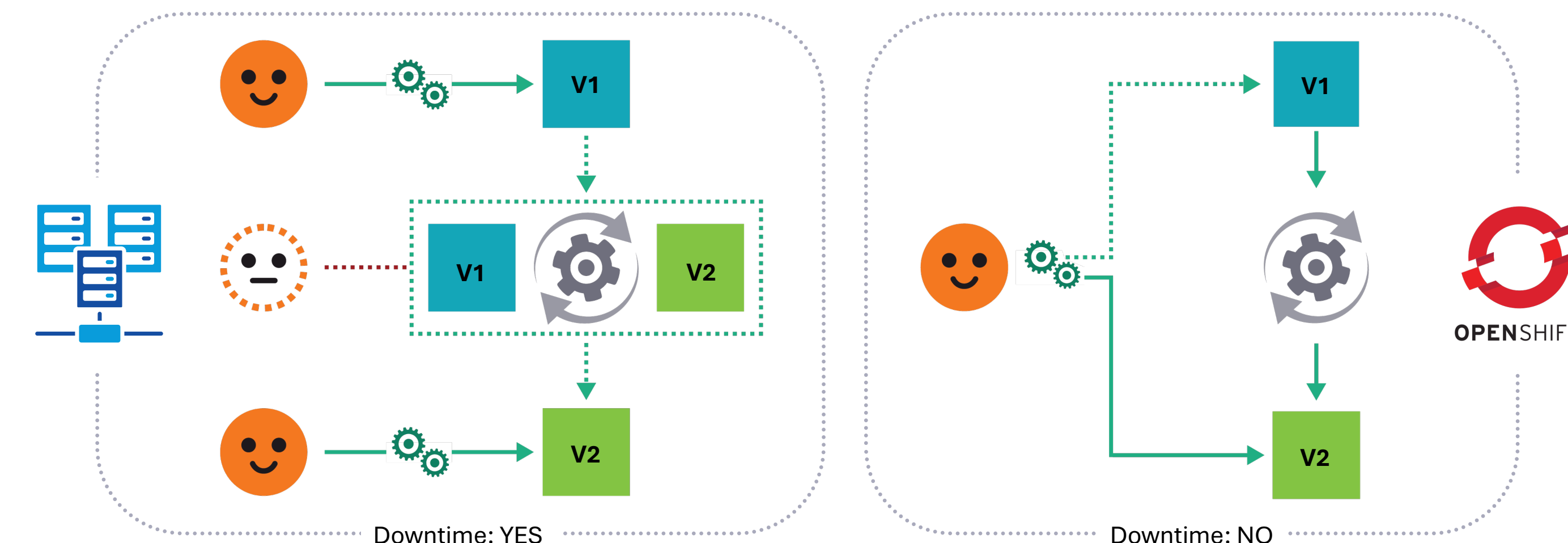
- **Containerization** ensures consistent environments, eliminating the need for manual host configurations
- Premade **pipeline templates** enable faster deployments of new services, reducing lead time
- **Single-click upgrades** through automated pipelines minimize human error and manual intervention, optimizing the upgrade process



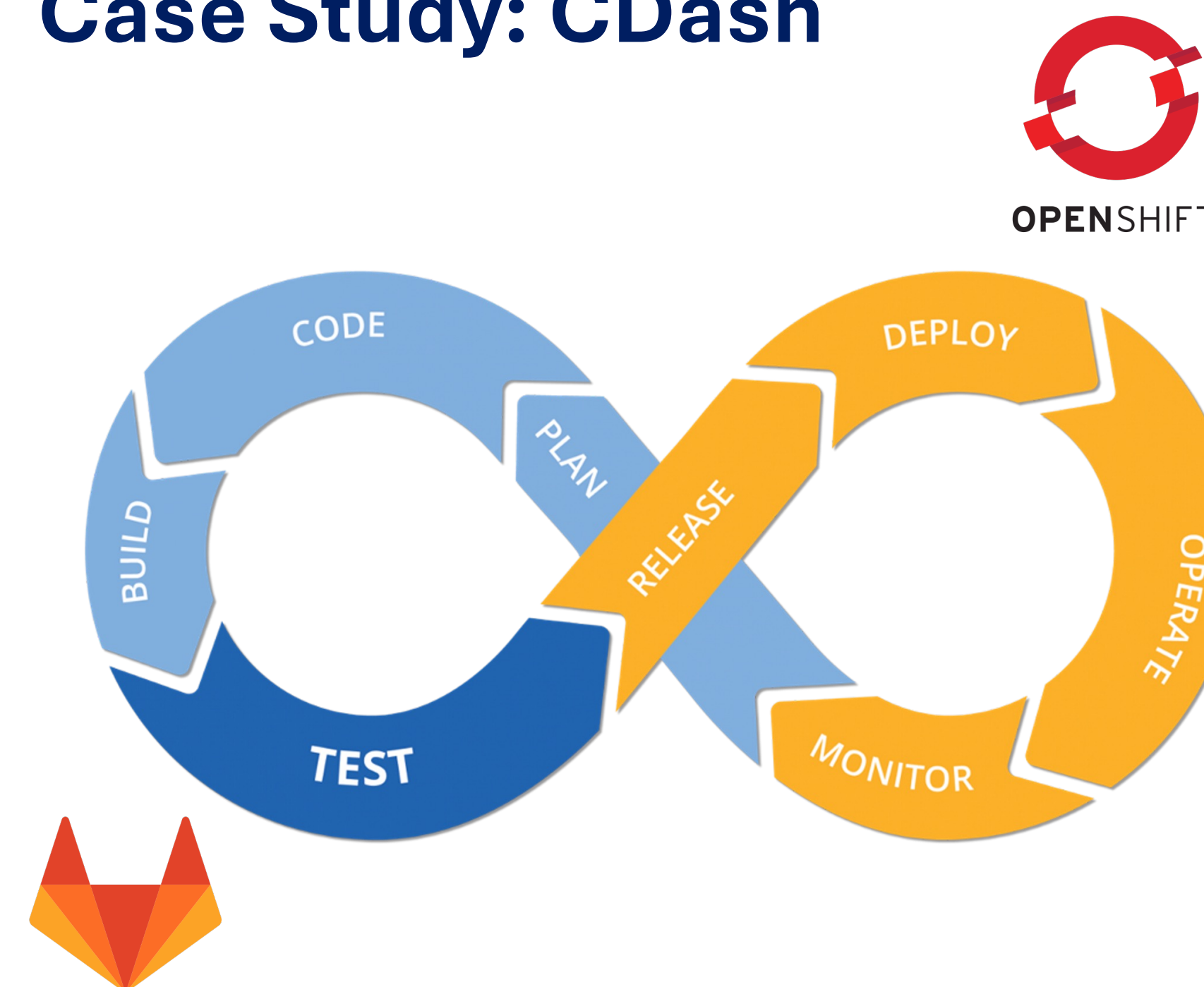
Automated Infrastructure & Dynamic Scaling

OpenShift empowers a single team to deploy web applications, while improving the end-user experience:

- OpenShift automates resource management, eliminating the need for manual server setup and inter-team dependencies
- **OpenShift Operators** automate application lifecycles, allowing rapid deployment of modern tools and features for users
- **Zero downtime upgrades** with Helm enable upgrades without service disruption
- Resources dynamically scale with user demand, ensuring services remain highly available and responsive



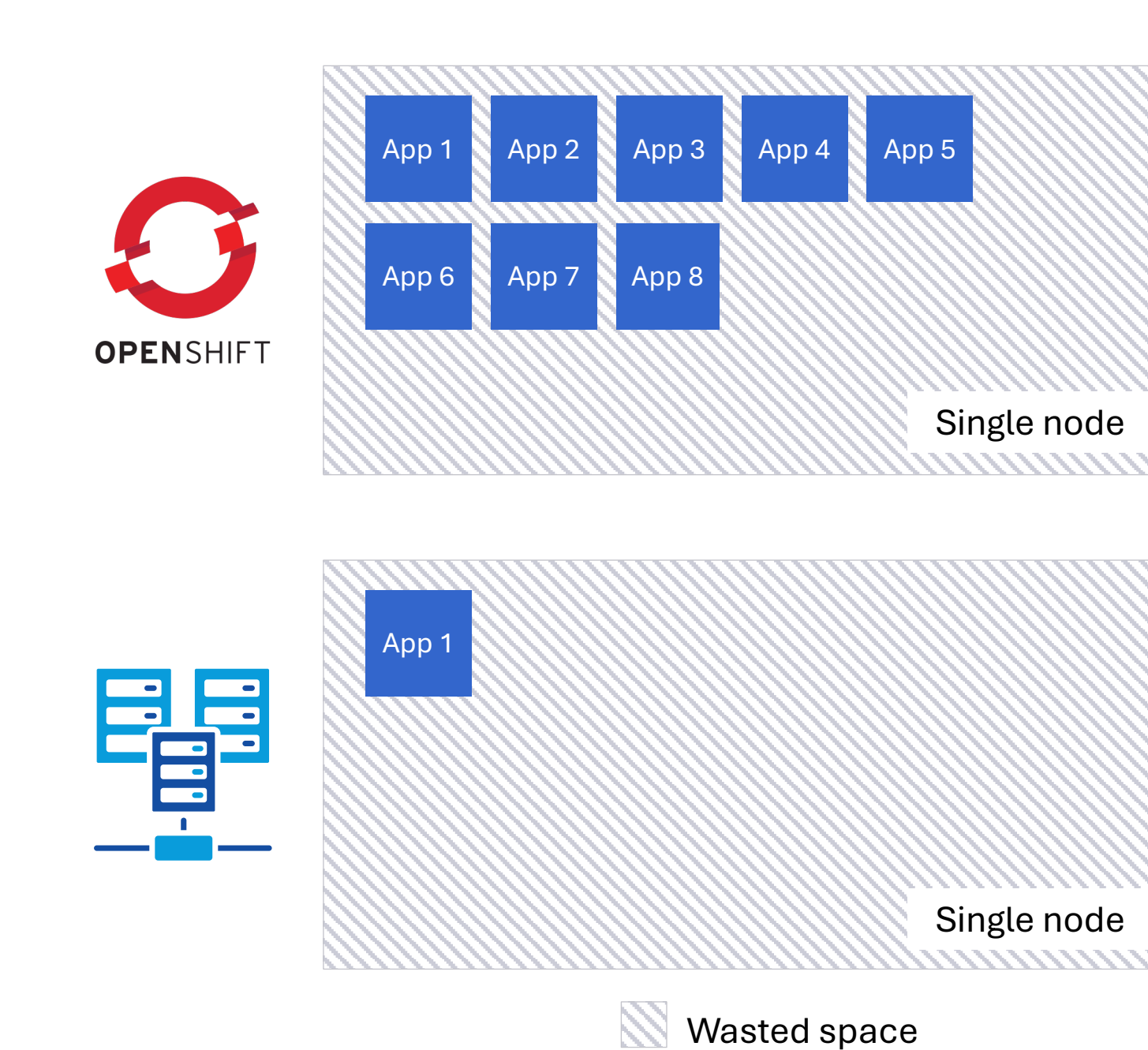
Case Study: CDash



CDash was deployed using DevOps and OpenShift

- Fills GitLab's gap for test result visualization, requested by WSC
- Built and deployed via a GitLab automated pipeline
- Seamless upgrades with Helm in OpenShift

Cost-Efficient Scaling



DevOps and OpenShift-based deployments have **reduced hardware by 40%**.

- Containerization creates isolated environments, allowing multiple applications to run on shared physical hardware without conflict
- Less hardware reduces setup and maintenance effort; freeing up our small team
- OpenShift dynamically adjusts resources, ensuring applications can run efficiently without overallocation

Identified Issues/Next Steps:

As we move more workflows into the cloud, we have identified gaps between our traditional infrastructure and cloud services:

- OpenShift doesn't recognize user details from our traditional LC systems, impacting permissions and access management in cloud
- Many user workloads require access to LC production storage, which isn't accessible in OpenShift yet
- Object Store (S3) as a potential solution

Conclusions

DevOps and Cloud/OpenShift deployments enable our team to effectively support more services:

- Containers provide consistent, isolated environments, optimizing resource use
- Deployment and maintenance are facilitated through GitLab automated pipelines
- OpenShift ensures applications are highly available, increasing user productivity

Collaborators

