ambassador

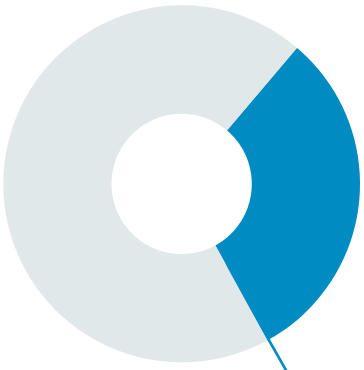
SOLUTION BRIEF

Bridge the gap between local and remote development environments

Telepresence accelerates collaborative development, debugging, and testing on Kubernetes by connecting developers' local machines to shared remote Kubernetes clusters.

Streamlining the Kubernetes development lifecycle: New tools and best practices are required for fast feedback

Cloud native applications have gone mainstream in enterprise IT environments where scalability, security, and efficiency are critical.



5.6 million
Developers using Kubernees worldwide representing

31%
of all backend developers.

(source: SlashData)

This rapid adoption of cloud technologies, such as Docker and Kubernetes, combined with the increasing numbers of enterprise developers, means that effective collaboration across development teams and cloud environment cost control are now table stakes.

Adopting cloud-native development best practices is step one in shipping features and getting applications to market faster. Kubernetes as the de facto container orchestration platform for building cloud-native applications helps deliver speed but also poses new challenges, such as navigating microservice architectures and the complexities and dependencies they create.

DEVELOPER BENEFITS

Share and collaborate easily
Share work in progress with teammates, colleagues and stakeholders for ease of collaboration and feedback.

Improve developer experience
Use the existing local development, build, and debugging workflows and tools the dev team knows and loves to focus on the core business: coding and shipping software.

Faster feedback loops
Access external dependencies in the remote Kubernetes cluster directly and test against the service being deployed for an instant feedback loop.

Shift testing left
Create a remote-to-local debugging experience. Catch bugs before they reach production and get faster feedback without the configuration headache of remote debugging.

Faster app development and delivery
Code and ship faster, free up 50% of dev time from manual tasks and launch features and apps into the world faster.

Telepresence makes the remote local for faster development

The typical local development environment has resource constraints that can't manage the workloads of all services a developer is working with. But developers need access to local tools to maintain productivity and ensure their developer experience allows them to code and test with speed. Eliminating the resource constraint problem and accelerating cloud-native application development and delivery requires the ability to connect the local development environment to a remote Kubernetes cluster.

Telepresence: Tackling the challenges of developing locally with Kubernetes

Telepresence is designed for developers who need to ship their application into remote Kubernetes clusters but still need to code, test, and debug services locally.

How Telepresence works

With Telepresence, Kubernetes developers code normally, with existing tools and workflows, as though their laptop is in their Kubernetes cluster. Telepresence runs code locally and forwards requests to and from the remote Kubernetes cluster, bypassing the much slower process of waiting for a container to build, be pushed to a registry, and deployed before seeing the impact of code changes.

BUSINESS BENEFITS

Reduce cloud costs

Achieve a lower total cost of ownership (TCO) by saving money on cloud infrastructure costs.

Accelerate time to market

Speed time to market and boost innovation velocity with new features and applications that drive revenue and customer value.

Enable collaboration in dev teams

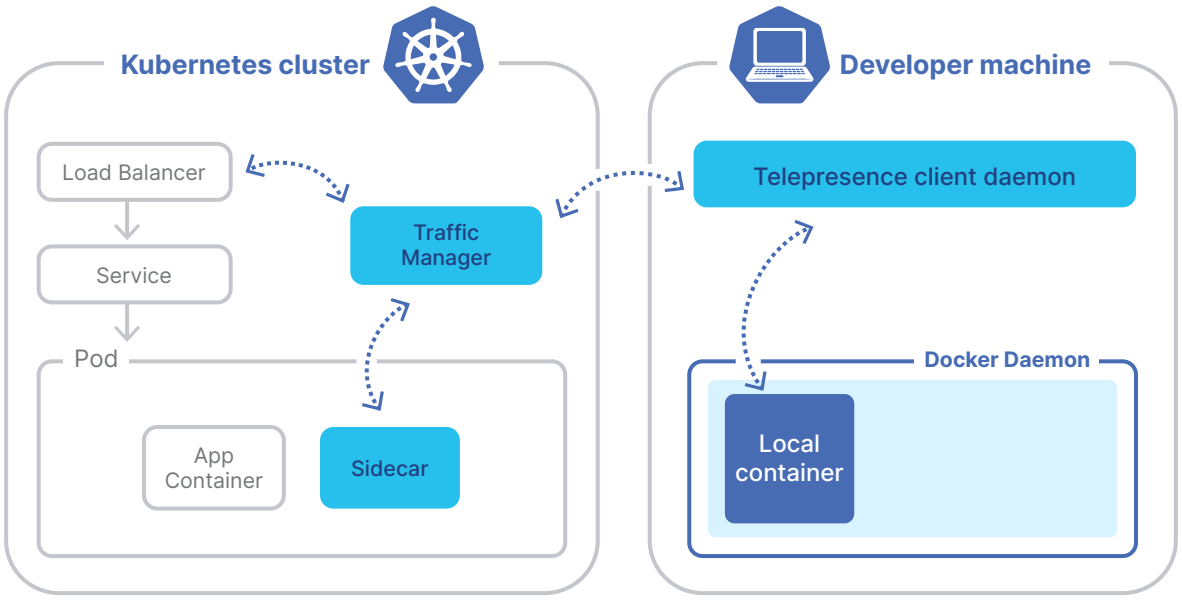
Enable collaboration by investing in tools that help speed up feedback and keep bugs out of production.

Boost developer productivity

Help developers to onboard faster, contribute early, collaborate easily, and retain the workflows and tools that fuel their productivity.

Reduce spend

Invest in cloud-native developer tooling to reduce spend on bigger and more powerful hardware or unpredictable pay-as-you-go cloud-based environments.



What makes Telepresence unique:

Personal intercepts

Personal intercepts enable developers to define conditions for when a remote request should be routed to their local container. The conditions could be anything from only routing requests that include a specific HTTP header, to requests targeting a specific route or path of an API. Developers can easily intercept or selectively reroute remote traffic to the service on their local machine, and test (and share with stakeholders) how their current changes look and interact with remote dependencies–all without impacting other developers and production.

Ambassador Cloud

Ambassador Cloud provides a single interface to develop, view, and manage Kubernetes services across the entire software lifecycle. Set up a Kubernetes development workflow in minutes and configure and use Telepresence without having to drop down to the CLI or Terminal.

Ambassador Cloud enables teams working with Kubernetes to code faster while shipping safely.

Core features

FEATURES	USE CASES
Remote-to-local development bridge	Developers can seamlessly and reliably bridge their local development machine to a remote Kubernetes cluster, enabling rapid exploration of remote resources and bi-directional proxying (routing) of remote and local requests. This enables developers to code and debug a service locally and test with remote traffic as if the service was running locally. By removing the slow code, build, push loop of container-based deployment, developers can reduce their feedback loop time by 50%.
Team collaboration with shared development environments	Enable easy sharing of cloud-based development and staging environments, including multi-user local-to-remote collaboration with **personal intercepts**. Telepresence can selectively and securely reroute traffic from a remote cluster to local services. Using request-based isolation (via HTTP headers or path metadata), developers can work locally and interact with services running a shared remote cluster, without overwriting others' changes or causing conflicts. Fewer remote development environments are required, saving cloud costs and maintenance expenses.
Intercept specification (CRDs, YAML)	Codify local development environment and service configuration using the Kubernetes-native Telepresence Intercept Specification. Platform engineers, DevOps, or tech leads can enable “one-command” automated development environment configuration for repeatable execution of service build steps, test setup scripts and CLI commands, and Telepresence intercepts. By consolidating and sharing this service setup configuration, teams can avoid the dreaded “works on my machine” phenomena, which in turn enables rapid onboarding and a consistent development experience across development teams.

FEATURES	USE CASES
Save intercepts	Enable Telepresence configuration to be easily shared and used by non Kubernetes-experts. With “Saved Intercepts”, successful intercept commands and configurations can be shared with and used by teammates. This means that teammates don’t have to be Kubernetes or Telepresence experts to start using and enjoying the benefits of the remote-to-local development experience.
Collaborative development	Get secure, shareable preview URLs showing local work in progress connected to a remote cluster. Developers can then test, explore and interact in the same way as end users of the website or application will. Access to preview URLs are protected using authentication through Ambassador Cloud, and by default, access to the URL is only available to users within an organization. URLs can be made publicly accessible for sharing with outside collaborators.
Ambassador Cloud dashboard	Get additional collaboration, auditing and security features. The **Ambassador Cloud** dashboard makes it easy for developers to track the services they have intercepted, the preview URLs they have shared, and which team members have accessed their preview URLs.
Technical Support and Assistance	Telepresence subscriptions also come with technical product support and assistance from a team of Support Engineers who are experts on Ambassador Labs products. Our Support Engineers work with users to diagnose and resolve issues and answer questions about product use and behavior. As you bridge the gap between your local and remote dev environments, our team can offer guidance regarding product configuration, providing the expert help you need to make the “remote, local” and do so with confidence.

Interested in learning more?

LEARN MORE

SCHEDULE A DEMO

ambassador

 | www.getambassador.io