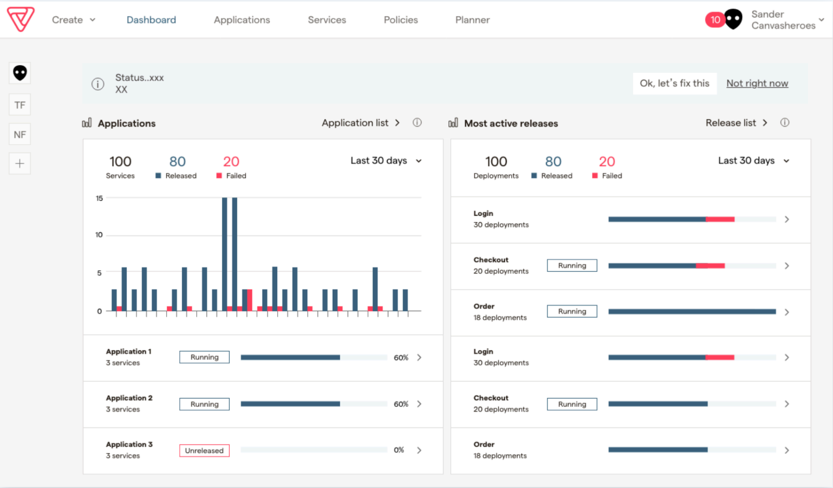
TECHNICAL DATA SHEET - HOW VAMP WORKS

Cloud-Native Release Orchestration

What is Vamp?

Vamp is an AIOps platform for Cloud-Native Release Orchestration that helps companies to release software faster and more frequently with near-100% reliability for a great customer experience. Vamp works seamlessly with your CI/CD pipeline and connects real-time customer experience to your software delivery pipeline.



Status..xxx
XX

100 Services	80 ■ Released	20 ■ Failed	
Application 1 3 services	Running		60%
Application 2 3 services	Running		60%
Application 3 3 services	Unreleased		0%

100 Deployments	80 ■ Released · 20 ■ Failed
Login 30 deployments	
Checkout 20 deployments	Running
Order 18 deployments	Running
Login 30 deployments	
Checkout 20 deployments	Running
Order 18 deployments	

How does it work?

RELEASE AUTOMATION

Vamp's release strategies are policy-based and work by comparing a service's KPI against the expectations you defined in your release policy. Vamp provides a library of policies that you can use out of the box or customised to your needs. These policies are templates which means you can standardise your release process and ensure it is able to scale across services, applications and teams.

Vamp's release strategies are ideal for services that use Kubernetes Deployments and expose an API that is consumed by customer facing web and mobile applications. The strategy works by progressively exposing larger and larger groups of user to the new version and evaluating the results before exposing the next group. This is especially useful if your KPIs include real-time measures of customer behaviour as well as technical performance KPIs. For example, with access to the right KPIs Vamp can quickly mitigate broken checkout flows.

In technical terms, these application level progressive releases use condition-based, traffic shaping to control which groups of users are presented with the new version of a service. Depending on the type of Ingress you are using and whether you are using a service mesh, this user segmentation can use a combination of HTTP headers, cookies and/or IP-address ranges.

Progressive strategies can be used across clusters as well as within clusters. In this case, Vamp orchestrates an ordered rollout of new versions of a service across 2 or more clusters. These progressive, cross-cluster rollouts can be mixed with Blue/Green strategies, for example to rollout a new version of a new message-based service across multiple cloud regions.

Vamp also supports Kubernetes StafefulSets or asynchronous, message-based services that do not expose an API.

CONTINUOUS VALIDATION

Vamp calculates a range of top-level and detailed application and traffic metrics. The top level KPIs match well to the SRE Golden Signals and provide an overall indication of the health of a service. These include **Kubernetes health** (Pod availability and restart deltas), **resource usage** (CPU and memory) and **traffic health**.

Kubernetes health and resource usage KPIs are always available. The traffic health KPIs depends on your infrastructure choices: the type of Ingress you're using and whether you are using a service mesh. Whether your services are message-based or expose an API. The type of APIs you are providing: GraphQL, gRPC, REST, etc. And whether you do TLS termination at the edge or not.

In addition to the KPIs Vamp calculates, Vamp can also capture histories of metrics, logs, traces and alerts taken from sources such as DataDog, Instana, Prometheus, Splunk and Sentry. The addition of these telemetry gives you even richer insights and observability into the performance of your services which is essential for accurately diagnosing issues and preventing future occurrences.

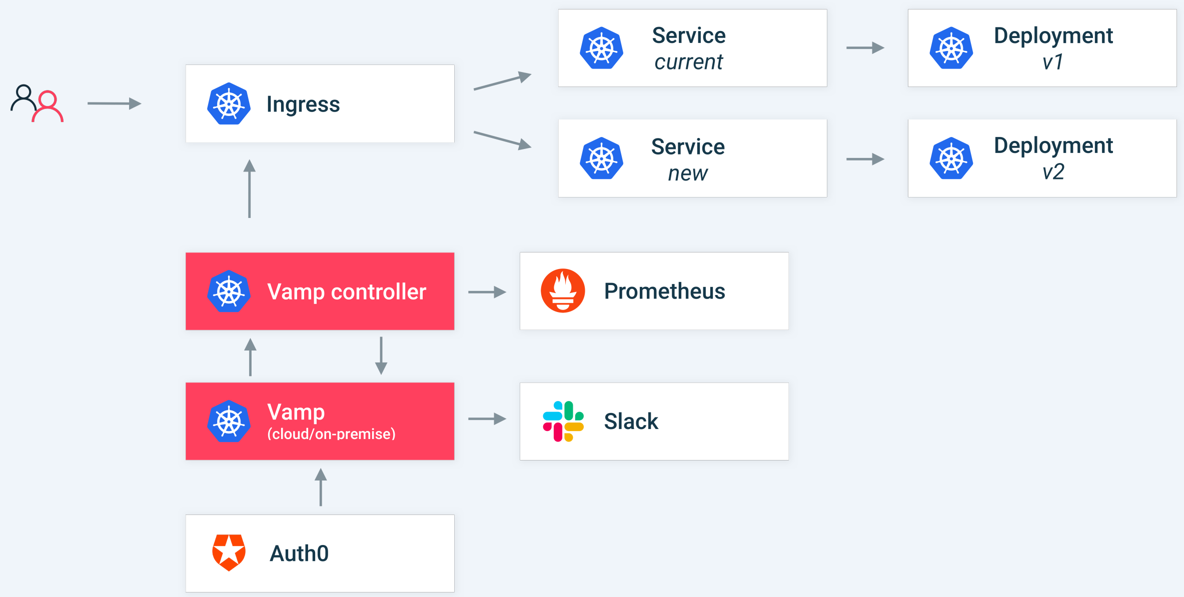
All these different KPIs are calculated every 30s and aggregated in Vamp to provide a repayable history of how a particular version performed. Vamp automatically validates that the new version is behaving correctly by comparing that service’s KPI against your expectations. If the new version doesn’t meet your expectations then it is rolled back by calling your CD pipeline, for example ArgoCD, via a webhook.

RELEASE OBSERVABILITY

With Vamp you can create dashboards around your release data, using flexible filtering and criteria. By creating and fine-tuning visualizations that slice and dice your data, you can track easily track KPIs like deploy and release frequency, succes rate, failure rate, number of rollbacks and more. This information helps teams to identify bottlenecks and continuously improve their software performance.

All events and user activities that Vamp detects, receives and generates are logged for audit trails. In addition to this, Vamp provides Role-Based Access Control (RBAC) so you can segregate duties within your team and grant only the amount of access to users that they need to perform their jobs.

Where does Vamp fit in my environment?



What do I need to use Vamp?

Vamp is role-based and is designed to be used by technical and less technical people. However, you need to have at least one person who has an operational knowledge of Kubernetes.

The minimum requirement to use Vamp is a Kubernetes v13+ cluster with a node with 0,3 vCPU and 0,5Gb memory free.

GET STARTED

Schedule a free **30-minute guided tour** with Olaf and he will walk you through Vamp's key features and benefits.

Olaf Molenveld
CTO & Co-Founder of Vamp

Visit **www.vamp.io** to learn more.