



Optimizing Kubernetes using SUSE Observability



Arshad Memon

Solutions Architect

arshad.memon@suse.com

Agenda

1. SUSE & Cloud Native

2. Why Observability?

Challenges in modern systems

The need for deep insights

A Kubernetes World

3. SUSE Observability!

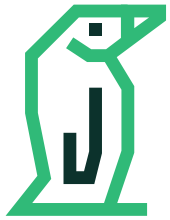
Approach

Architecture

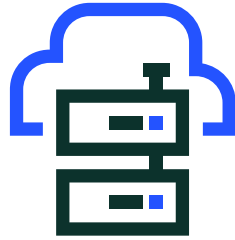
4. Demo



The Open Infrastructure Platform



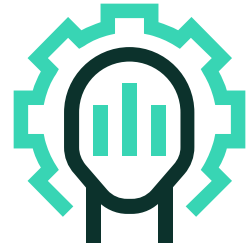
Linux



Cloud Native



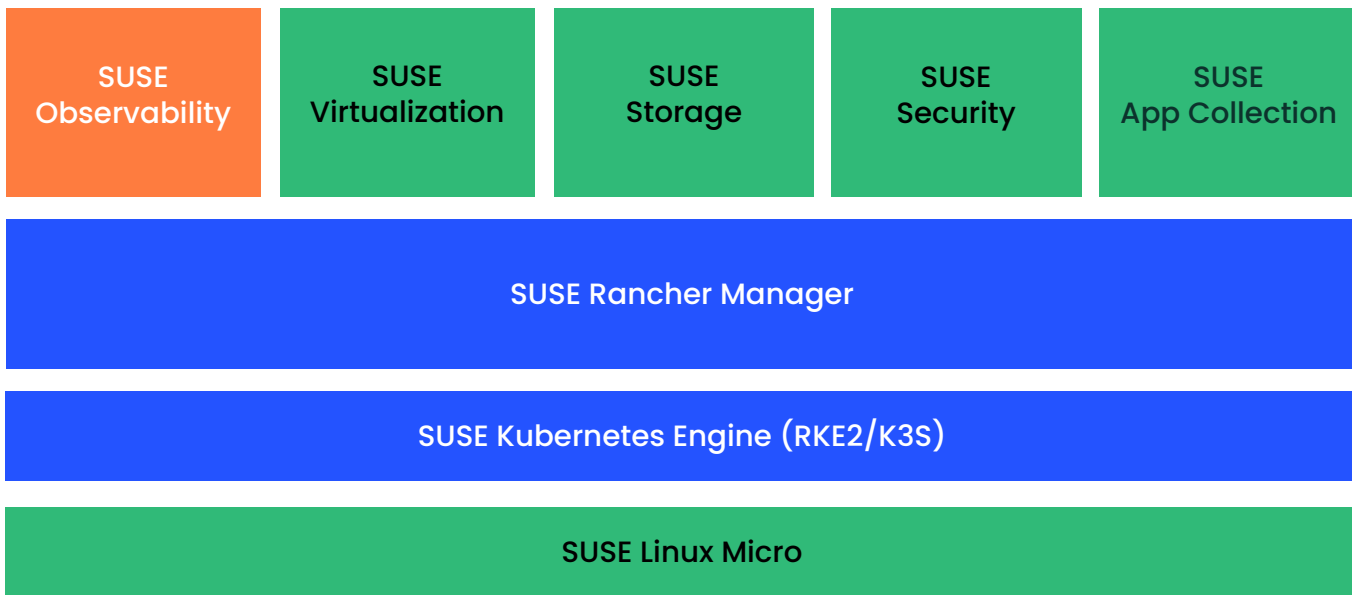
Edge



AI

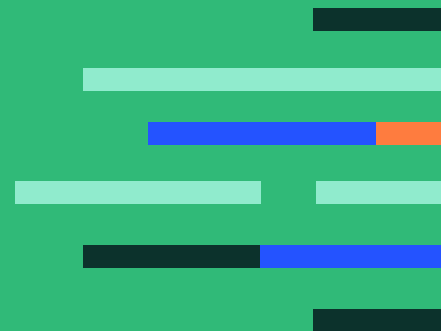


SUSE® Rancher - Complete Container Management





Why Observability?



A critical application suddenly starts misbehaving.

Traditional Monitoring Fails Us

Consequences

Users report slowdowns, errors, or complete outages. Your team is in the dark, unsure where to even begin looking for the cause.

Basic dashboards show something is wrong, but not why. Is it the network? The application code? A hidden dependency?

Downtime:

Frustrated users, lost revenue, damaged reputation.

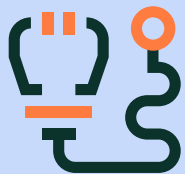
Wasted Time:

Engineers scrambling blindly, costing time and money.

Missed Opportunities:

Instead of fixing the issue, you're just trying to find it.

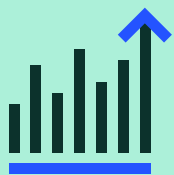
Observability is the ability to measure the internal states of a system by examining its outputs.



Health



Performance



Metrics



Logs



Traces

Observability: The Bigger Picture especially in Kubernetes

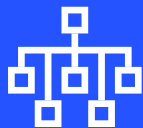
Beyond the Basics: Observing your Kubernetes clusters effectively means going beyond just metrics, logs, and traces.

The Power of Context



Time

How have deployments, resource usage, and network traffic evolved?



Relations

How do services interact across clusters?



Events

What triggered a specific issue? Was it a deployment, configuration change, or external factor?

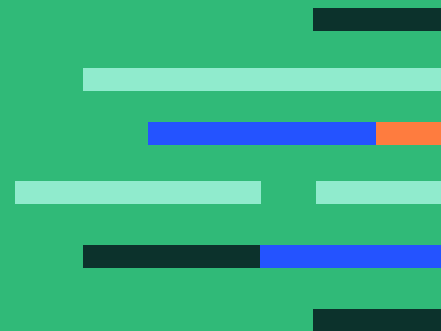


Security

Are there any security vulnerabilities or breaches across your clusters?



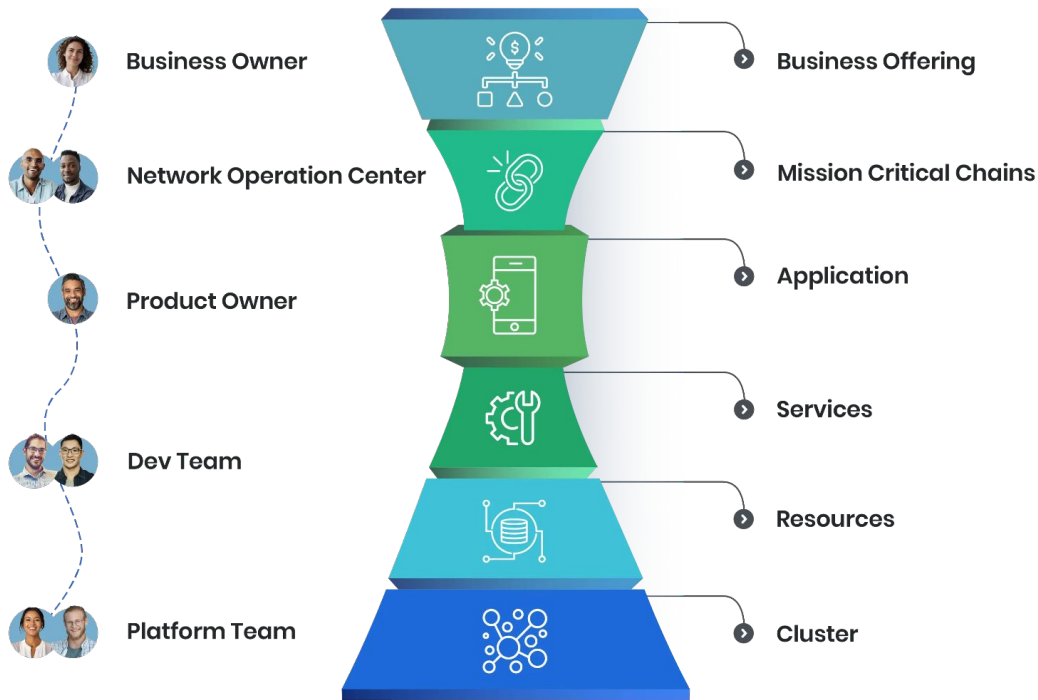
SUSE Observability!



Providing insights to diff Organization Layers

By expanding insights into your Rancher clusters, platform teams can troubleshoot issues faster.

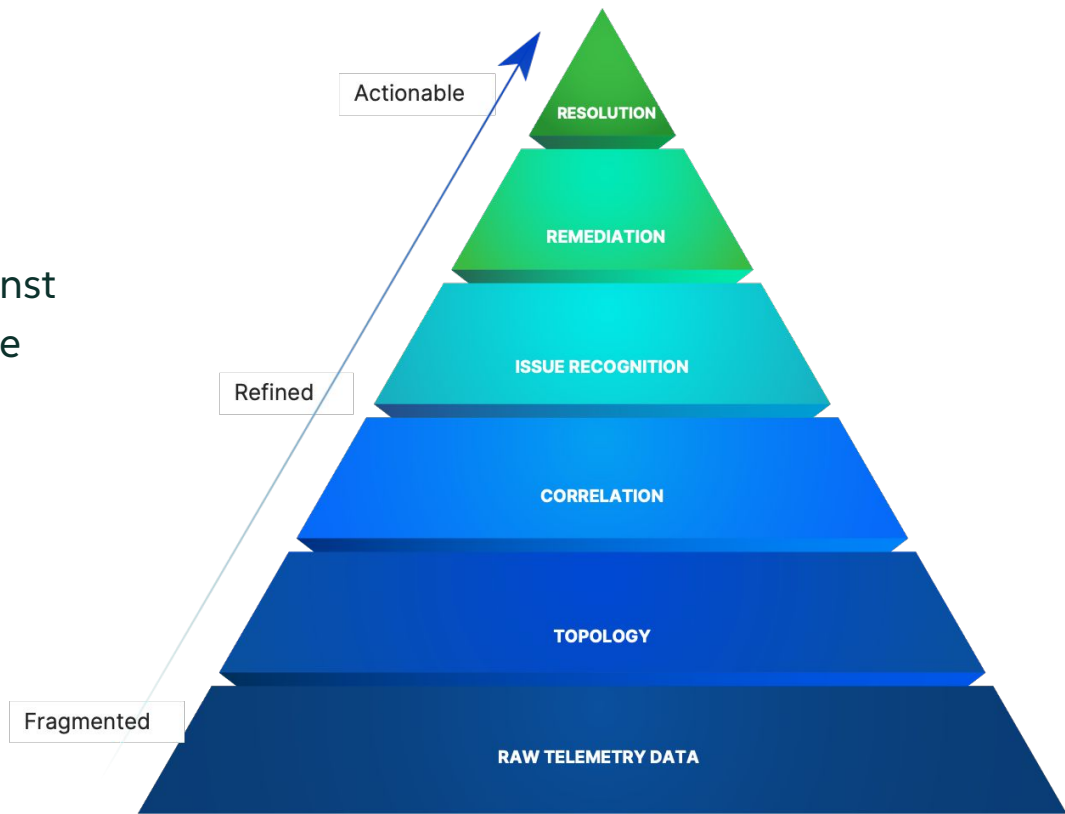
Connecting all organizational layers improves business reliability.



Look beyond distinct data sets

Key considerations

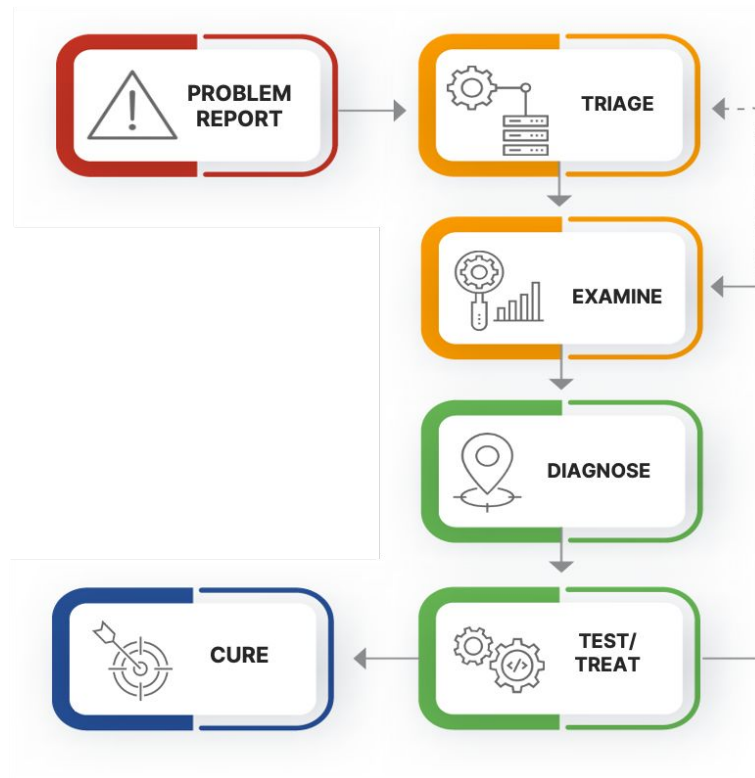
Step away from siloed data. Learn to correlate metrics, events, and traces against a comprehensive topology for a complete observability picture.



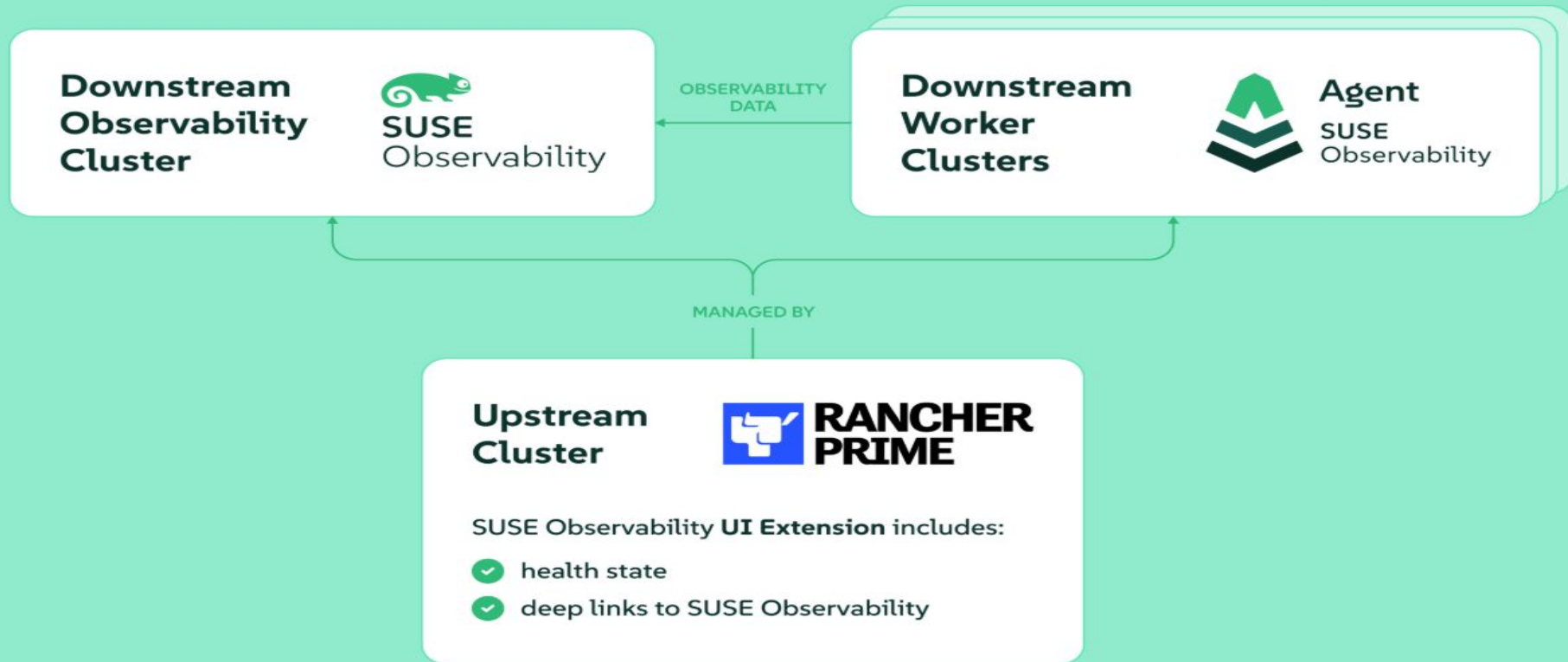
Accelerate issue resolution

Rancher Prime Observability contains a best practice for issue resolution.

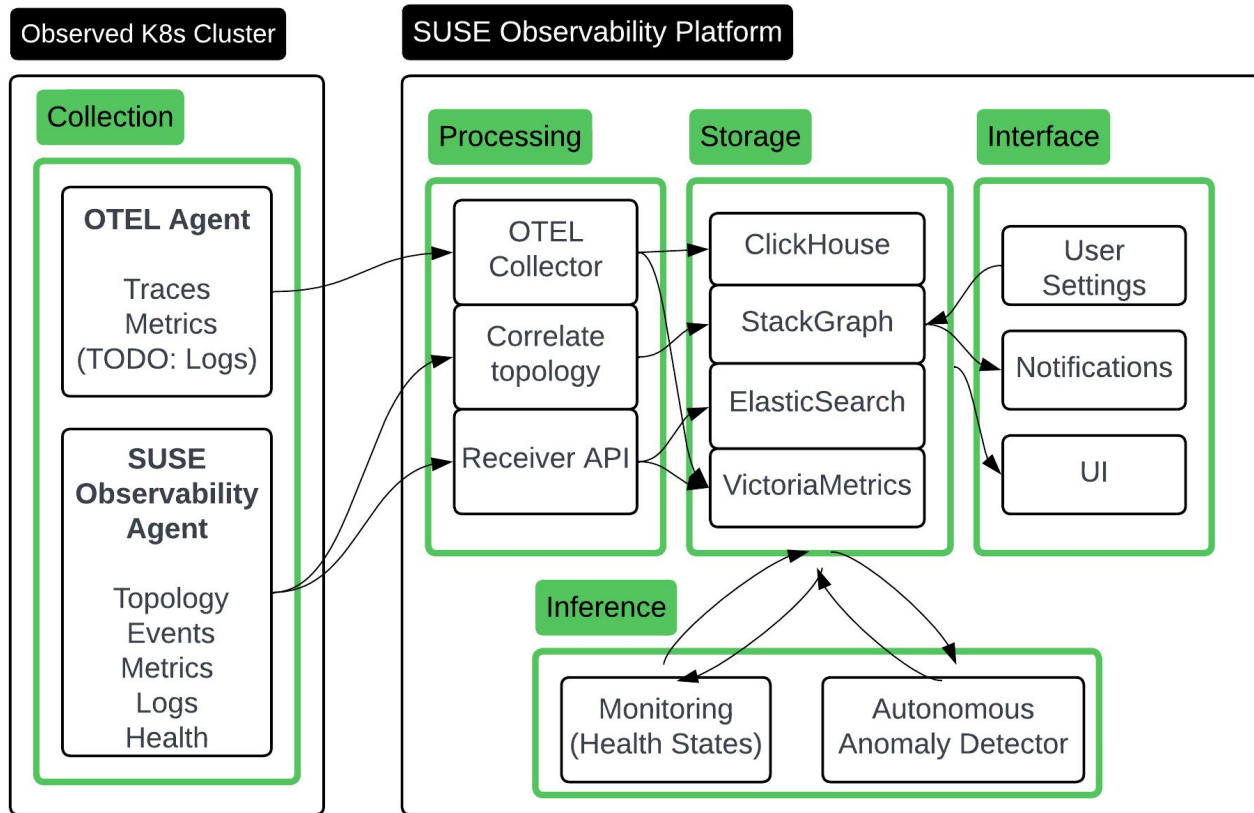
We enable engineers to navigate the quickest path from spotting issues to fixing them, inspired by practices from the SRE Handbook.



SUSE Observability



Data pipeline overview



Automatic application instrumentation

Applications can be instrumented without touching the source code. This is available for Java, Node.js, Go, Python and .NET.

An OpenTelemetry Kubernetes Operator is available to automatically instrument applications in the cluster using this zero-code instrumentation method.

Next to traces, this also provides additional metrics from the application runtime like Garbage Collection and Heap usage, automatically monitored by SUSE Observability.

<https://opentelemetry.io/docs/zero-code/>



Demo
<https://play.stackstate.com>

How to get your hands on it?

- Rancher** community users → Free to use SUSE Cloud Observability (max 10 nodes)
- Rancher Prime** customers → Licence included for self-hosted deployments
- All others → Open Source release being prepared, date T.B.A.





SUSE