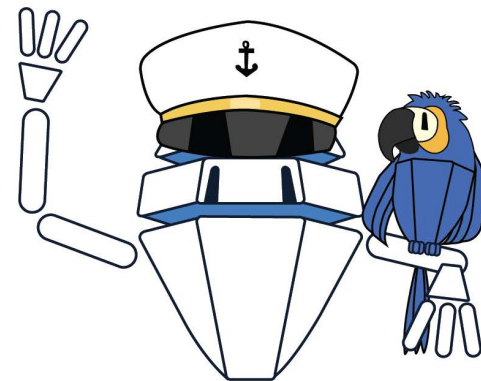




# The Illustrated Guide To Troubleshooting Kubernetes Applications



*with Captain Stacky  
and his friends*





# nce upon a time...

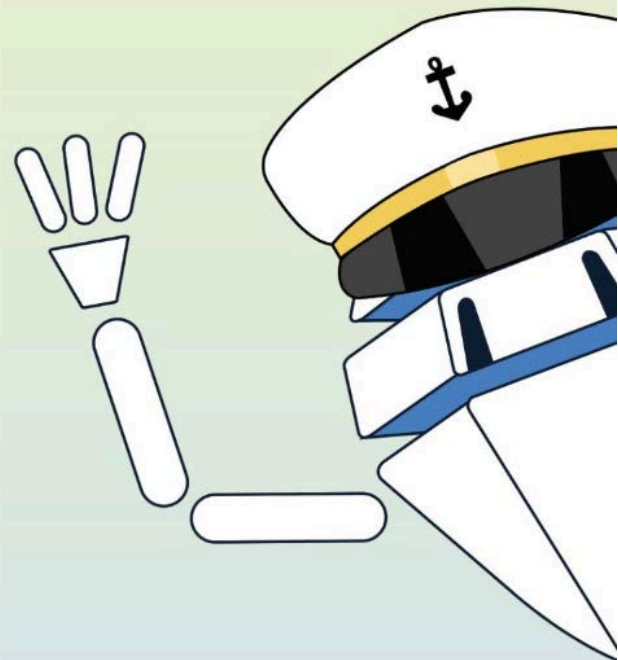
## ...there was a magical world called Kubernetes®.

It was a place where developers could easily deploy and manage their applications, but there was a problem! Even though Kubernetes was the de facto standard for container orchestration, troubleshooting Kubernetes applications was so complicated!

Stacky knew that with the right tools, developers and engineers could troubleshoot their problems faster. And, eventually with his help, they did!

Now, Stacky is ready to share his knowledge with you! He'll show you the different tools that engineers typically use for Kubernetes troubleshooting and a better way to handle things when they go wrong.

So come along on this exciting adventure and let Stacky help you make your Kubernetes journey a smooth one!





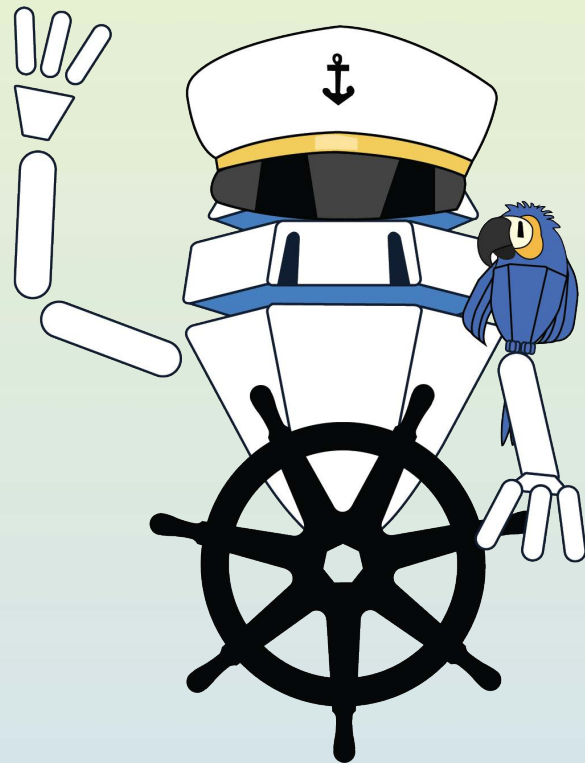
# Setting Sail to Troubleshoot Kubernetes

**Stacky heard that his friends were having a really tough time because the Kubernetes apps they worked with had issues that needed fixing. And if they didn't get fixed, ordinary people like you and me couldn't use their apps on computers and phones without experiencing any problems.**

When Stacky heard what his friends were up against, he set sail on a grand adventure to explore the challenges they were facing.

In his Captain's Log, he jotted down some ideas about Kubernetes troubleshooting — things that could help make sure that people's apps would work smoothly and without any trouble.

But first, he wanted to see what tools his friends used for troubleshooting to make sure he could help them remediate issues faster.





# First Stop: Tiger Tracking Changes!

**Meet Tiger. Even though he has stripes, Stacky calls him a trend-spotter since Tiger uses metrics to track changes in performance patterns.**

- The tool that Tiger uses to capture metrics is the popular open-source solution, Prometheus™.
- With Prometheus, Tiger can gather and store metrics and use Grafana® to visualize them, making for a powerful approach that tracks changes in performance patterns over time and detects issues when they occur.
- Metrics play an important role in understanding why applications work in a certain way, and are probably the most common way engineers track changes.
- PromQL® is frequently used to query specific metrics, and OpenMetrics has emerged as industry standards for transmitting cloud-native metrics and metric collection at scale. Some commercial options for metrics collection include Chronosphere® and ContainIQ™.

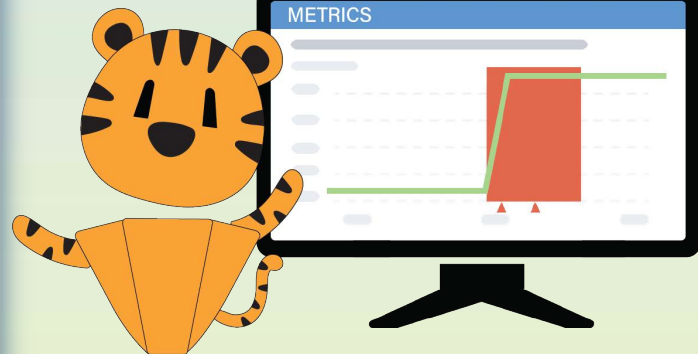
**Stacky thought it was great that Tiger wanted to use data to make informed decisions and gain insights into various areas. Then he sailed on to see what tools his other friends were using.**



How do you troubleshoot?



I use metrics to track changes in performance patterns.





# Next Up: Monkey Hunting Treasure!

**Meet Monkey. Stacky calls her a treasure hunter because she looks at various logs from multiple sources to navigate issues.**

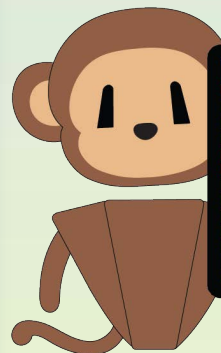
- The approach that Monkey uses to troubleshoot issues in a cluster is to manually collect, aggregate, analyze and visualize logs. These logs serve as a valuable source of information, containing warnings, errors and other behavioral data about the resources running in the cluster.
- The first step is to gather logs from individual resources such as pods, nodes or applications. Each log provides a piece of the puzzle. The next step is to put the pieces together to determine what happened.
- However, utilizing logs for troubleshooting requires a deep understanding of how resources and services are related and how to navigate through the logs effectively.
- Several solutions, such as Grafana, Loki and Logz.io, can help simplify the treasure-hunting process.

**Stacky saw that Monkey had a lot on her plate, causing her to go bananas! He wanted to help her, too, but there were more friends to see before he could find the best troubleshooting solution!**

How do you troubleshoot?



I search for details in the logs to find out what went wrong.



LOGS				
>	api-controller...	default		Warning
>	api-server-1887...	default	10   11   12   13	Warning
>	api-server-1888...	default	10   11   12   13	Warning
>	api-server-1889...	default	10   11   12   13	Warning
>	api-server-1890...	default	10   11   12   13	Warning
>	api-server-1891...	default	10   11   12   13	Warning
>	api-server-1892...	default	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning
>	ingress-nginx...	kube-system	10   11   12   13	Warning



## Next Up: Elephant Sniffs for Traces!


**Meet Elephant. He uses his long trunk to seek out trace information to get an end-to-end picture of what's going on.**

- Elephant uses traces to track the flow of data through a system, providing insights into how different components interact with each other.
- Many engineers rely on an open-source framework to capture and process trace information. A popular one is Jaeger, a monitoring and troubleshooting solution focused on complex, distributed systems.
- If a customer's transaction requires multiple services, seeing and troubleshooting the bigger picture can be tricky and traces are really helpful in these situations.
- OpenTelemetry™ is rapidly becoming the standard for capturing trace information. Another widely used solution is Epsagon®.

**Stacky thought that Elephant sniffing for traces to figure out what's happening was a great idea. But was that enough to fix all of his problems, optimize performance and improve reliability?**



How do you troubleshoot?



I use trace data to get a big picture view of what's happening.



TRACES



## Final Stop: Zebra on the Lookout!

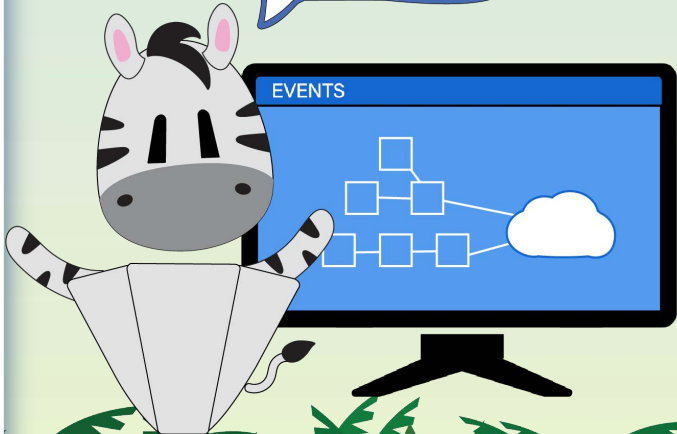
**Meet Zebra. She's an event watcher. To identify potential threats and opportunities, she uses an approach similar to the one she uses in the Savanna grasslands: she watches for events and changes.**

- By closely monitoring all the changes and events happening in a Kubernetes landscape – things like images pulled, containers started or pods killed because they ran out of memory – Zebra can better understand how the system is functioning and identify potential issues before they become critical so she can take action as needed.
- Bringing these events from all resources together into a single tool can speed things up, too, rather than using a command line to query them one at a time.
- Because they track events and changes in configuration, tools like Elastic can be particularly helpful as they provide a centralized way to monitor events across multiple resources, making it easier to maintain system reliability.

**Stacky agreed Zebra's tool was pretty cool. But seeing how many ways there were to troubleshoot, he wondered if there was an easier way to remediate issues faster!**



How do you troubleshoot?



I monitor events to understand what's happening in my Kubernetes cluster.



# Stacky's 6 Troubleshooting Challenges



Stacky listened carefully to the various ways his friends were troubleshooting their Kubernetes apps. He knew that if he could find a solution to these six challenges, his friends could be a lot more effective at ensuring system performance and reliability.

## 1. Environment complexity

When troubleshooting issues, the number of resources and the environment complexity that on-call engineers need to deal with is simply mind blowing.

## 2. Constant change

In highly dynamic environments, it can be a struggle to understand the system's current state and even more difficult to know what change caused an issue.

## 3. Lack of knowledge

Knowing how to gather and analyze observability data for monitoring and troubleshooting is a challenge. Without deep knowledge of Kubernetes, you barely know where to look.

## 4. Too much context switching

The information needed to solve an issue is spread across many tools. Continuously switching between them takes time and can be difficult to navigate.

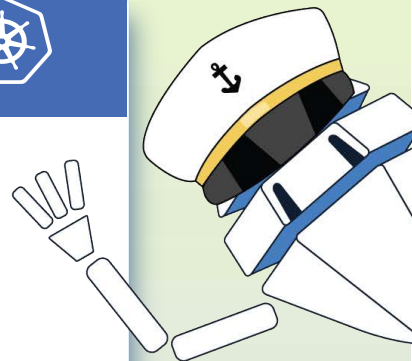
## 5. Cost of engineering time and toil

Open source observability may be free, but it is not cheap — it takes a lot of time to configure, maintain and run open source observability solutions.

## 6. Lack of historical data

Kubernetes does not retain historical data for very long, such as events, metrics or logs. If this data is not available, you will not get a complete picture of what happened and it will be harder to solve the problem.

With these challenges in mind, he set off to find a solution.







# Gather 'Round. I Found a Solution!

After sailing around the world, Stacky came back to the islands and invited his friends to join him on board his new container ship. He asked them:

**"What if we combined all of your approaches in a single tool? Then you could understand issues better and fix them faster! That's what StackState does."**

His friends were a bit dubious but they let him explain:

"StackState collects and correlates all the essential data types — metrics, logs, events and traces — and then shows the impact of issues on the business and accurately identifies the cause of those issues. StackState even incorporates expert practices to help teams detect and fix issues automatically!"

"That sounds cool," said Zebra. "So, does that mean that any engineer or developer can troubleshoot the operation of Kubernetes-based applications and services, even if they lack specific knowledge of the application, service architecture or Kubernetes itself?"

"Exactly!" said Stacky. "Let me tell you how."





# StackState Has Perfected Troubleshooting!

## StackState has become the most efficient tool for troubleshooting Kubernetes applications by focusing on 4 key areas:

- 1. Expert practices:** StackState applies expert Kubernetes practices right out of the box, providing pre-configured monitors that look for common problems to help identify and prevent issues before they escalate.
- 2. Comprehensive data collection:** StackState collects all essential metric, log, event and trace data including golden signals such as latency, saturation, throughput and error rate. Then it correlates all relevant data, providing full context for analysis and decision-making.
- 3. Dependency visualization:** StackState automatically discovers and visualizes all Kubernetes service and resource dependencies, providing real-time insight into the dynamic environment. The dependency maps helps teams keep track of changes and maintain a complete understanding of their system.
- 4. Intelligent remediation:** StackState provides troubleshooting hints, visual assistance and step-by-step remediation guidance tailored to your specific environment. This step-by-step guidance allows everyone on the team to quickly and accurately identify and fix issues, reducing downtime and improving system reliability.





## See for Yourself!

StackState is a comprehensive, standalone SaaS solution that provides deep insight into your Kubernetes clusters. It aggregates and correlates metrics, logs, events and traces, showing relationships and dependencies across services.

**IT WORKS!**

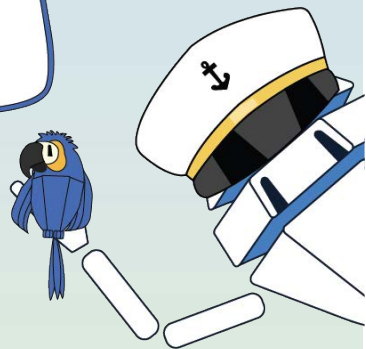


StackState helps SREs, engineers and developers supporting Kubernetes-based applications to effectively troubleshoot performance and reliability issues. It makes issue remediation a breeze!

### **Take it from Stacky's friends...**

By using StackState, they got much faster at problem solving. They can now focus on their real jobs, like adding innovative, new features and improving overall processes.

**Seeing is believing!**



**Experience StackState on your terms.**

**Schedule a 20-minute StackState demo with one of our experts.**

**Explore our product with a self-guided tour. See firsthand how our tool helps you resolve the most common troubleshooting challenges.**

**Sign up for a free trial. Use your data and put us to the test. It only takes five minutes to set up!**



# StackState

Designed to help engineers of all skill levels who build and support Kubernetes-based applications, StackState provides the most effective solution available for Kubernetes troubleshooting. Our unique approach to SaaS observability helps teams quickly detect and remediate issues so they can ensure optimal system performance and reliability for customers. With StackState's comprehensive observability data, the most complete dependency map available, out-of-the-box applied knowledge and step-by-step troubleshooting guidance, any engineer can remediate issues accurately and with less toil.

Leading enterprises like KPN, Vodafone, Accenture and Danske Bank rely on StackState to ensure the performance and reliability of their business-critical services. We are proud to be recognized as a High Performer in the G2 spring 2023 report based on the strength of [our customer reviews](#).

For more information, visit us on [www.stackstate.com](http://www.stackstate.com) and follow us on:



© 2023 StackState. All rights reserved.

All trademarks, service marks and trade names of StackState (StackState, 4T, StackVista, time-traveling topology and the StackState logo) (collectively "Marks") are pending trademarks™ or registered trademarks® in the name of StackVista Group B.V. or its affiliates, partners, vendors or licensors. You may not use, copy, reproduce, republish, upload, post, transmit, distribute, or modify the Marks in any way, including in advertising or publicity pertaining to distribution of materials on the Services, without prior written consent of StackState. 0523v00

