

Kubernetes Stories

BEDTIME KUBERNETES STORIES FOR DEVOPS ARCHITECTS



PREPARED BY

PAVAN BELAGATTI

Kubernetes is like teenage sex; many people just talk about it a lot more than they actually do it.

But....

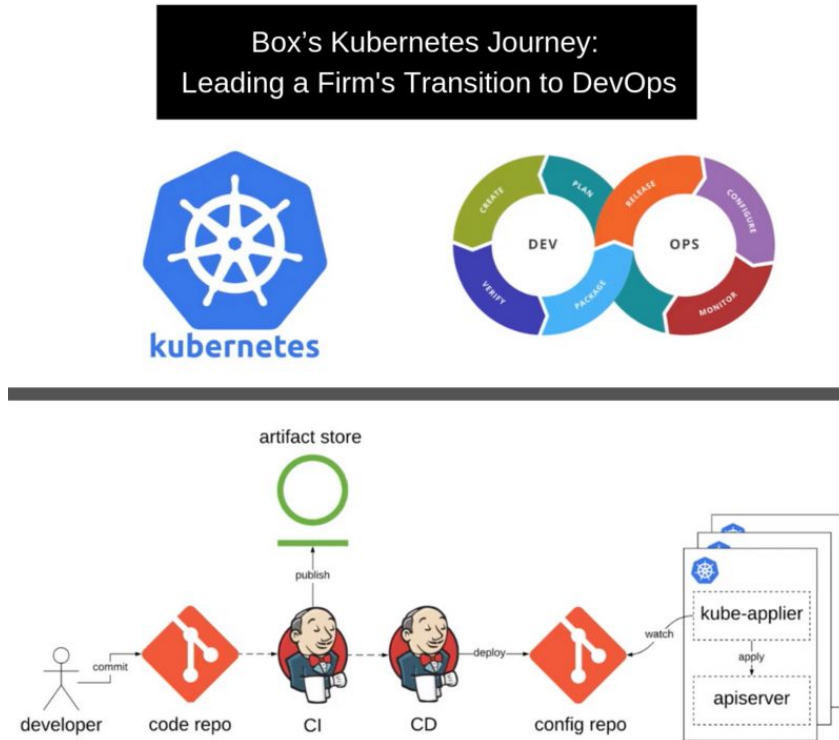
How about learning from someone who has already done a Ph.D. in the subject?

Presenting 10 mind-blowing stories that should help you gain real-world Kubernetes usage.

- **Box's Kubernetes Journey: An interesting case study:)**

A few years ago at Box, it was taking up to six months to build a new microservice.

Fast forward to today, it takes only a couple of days.



How did they manage to speed up?

Two key factors made it possible,

1. Kubernetes technology
2. DevOps practices

Founded in 2005, Box was a monolithic PHP application and had grown over time to millions of lines of code. The monolithic nature of their application led to them basically building very tightly coupled designs, and this tight coupling was coming in their way. It was resulting in them not being able to innovate as quickly as they wanted to.

Bugs in one part of their application would require them to roll back the entire application. So many engineers working on the same code base with millions of lines of code, bugs were not that uncommon. It was increasingly hard to ship features or even bug fixes on time. So they

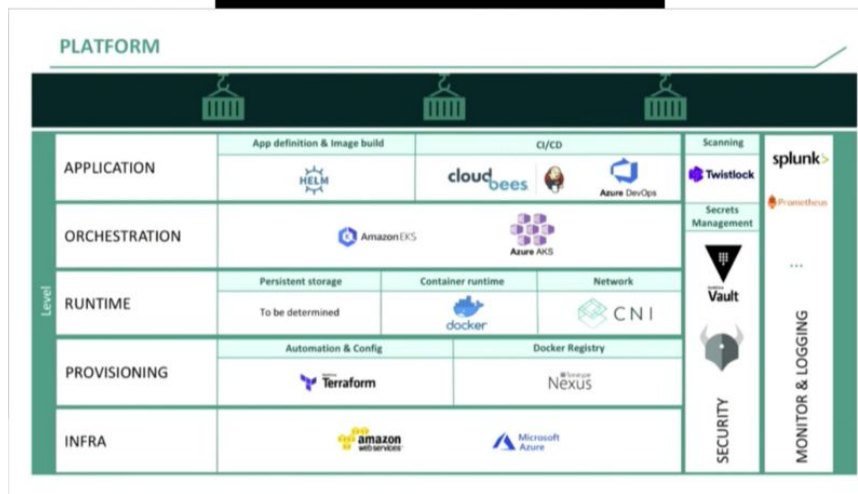
looked out for a solution and decided to go with the microservices approach. But then they started to face another set of problems....

That's where Kubernetes came in:) See the full video talk by Kunal Parmar, Senior Engineering Manager at Box: <https://lnkd.in/etnJTbE>

- **A 250 Year Old Bank's Cloud-Native Kubernetes Journey:**

A 250 Year Old Bank's Cloud-Native Journey

Financial sector Enterprising bank	20,000 Total number of employees
Amsterdam Headquarter	400+ Development Teams
Agile organization DevOps / Hybrid cloud	3,000+ Applications



For them, it all started with using containers in the beginning and they began to face some problems in the initial stages since it was a bank (financial sector), they usually face more challenges with compliance, governance and the priority was more on security. On the cloud-native landscape, as there are many tools, it was confusing for them to choose which tool for what as they didn't want each developer team selecting different tools and facing a catastrophic separation from others and licensing issues.

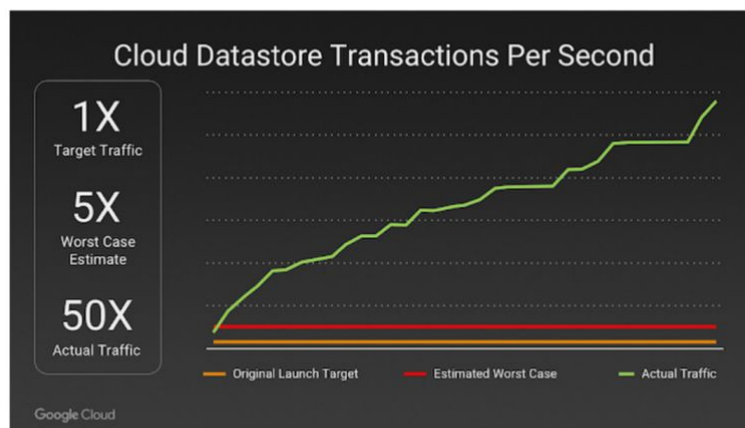
So the need for them was to come up with some clear guidelines for developers, the best cloud features they can consume easily before moving to a cloud-native approach to creating a uniform way of working. They also came up with a plan of having a regulated team that previously worked on tools and processes to share knowledge and best practices. They created a team called 'Stratus.' The mission of this team 'Stratus' is to enable development teams to quickly deliver secure and high-quality software by providing them with easy to use platforms, security, portability across clouds on enterprise-level, and reusable software components.

The keynote talk video is here: <http://bit.ly/2FfUGUE>

- **How did 'Pokemon Go' able to scale so efficiently?**

The answer is Kubernetes.

Kubernetes Story at



Source credits: Google Cloud (cloud.google.com)
Pokémon GO (pokemongo.com)

500+ million downloads and 20+ million daily active users. That's HUGE.

Pokemon Go engineers never thought their user base would increase exponentially surpassing the expectations within a short time. Even the servers couldn't handle this much traffic.

The Challenge:

Created by [Pavan Belagatti](#)

The horizontal scaling on one side but Pokemon Go also faced a severe challenge when it came to vertical scaling because of the real-time activity by millions of users worldwide. Niantic was not prepared for this.

The Solution:

The magic of containers. The application logic for the game ran on Google Container Engine (GKE) powered by the open-source Kubernetes project.

Niantic chose GKE for its ability to orchestrate their container cluster at planetary-scale, freeing its team to focus on deploying live changes for their players. In this way, Niantic used Google Cloud to turn Pokémon GO into a service for millions of players, continuously adapting and improving. This got them more time to concentrate on building the game's application logic and new features rather than worrying about the scaling part.

"Going Viral" is not always easy to predict but you can always have Kubernetes in your tech stack.

- **Shopify engineering team's journey to building their own PaaS with Kubernetes**



Shopify was one of the pioneers in large-scale users of Docker in production.

They ran 100% of their production traffic in hundreds of containers. Shopify engineering team saw the real value of containerization and also aspired to introduce a real orchestration layer.

They started looking at orchestration solutions, and the technology behind Kubernetes fascinated them.

It all started in 2016 where all the engineers were happy running services everywhere with a simple stack that included Chef, Docker, AWS, and Heroku. But just like any other company that is in the growth phase, the Shopify encountered some challenges when this Canadian e-commerce company saw 80k+ requests per second during peak demand. Wohooo:)

Many processes were not scalable, and they needed a quick solution. The Shopify team recognized that they needed to increase their focus on tested infrastructure, and automation that works as expected, every time.

The Shopify engineering team believed in three principles: providing a 'paved road', 'hide complexity' and 'self-serve.'

Read this fascinating story here: <https://lnkd.in/eN34vAm>

All credits to Niko Kurtti, QCon & InfoQ.

- **Italy's biggest traditional bank is embracing Kubernetes.**

A conventional bank running its real business on such a young technology?
Are you kidding me?

Nope, I am not kidding. Italy's banking group, Intesa Sanpaolo, has made this transition.

These are banks who still run their ATM networks on 30-year-old mainframe technology and embracing the hottest trend & tech is nearly unbelievable. Even though ING, the banking and financial corporation changed the way the banks were seen by upgrading itself with Kubernetes and DevOps practices very early in the game, there was still a stigma with adopting Kubernetes in the highly regulated and controlled environments like Healthcare, Banks, etc.

The bank's engineering team came up with an initiative strategy in 2018 to throw away the old way of thinking and started embracing the technologies like microservices, container architecture, and migrate from monolithic to multi-tier applications. It was transforming itself into a software company, unbelievable.

Today the bank runs more than 3,000 applications. Of those, more than 120 are now running in production using the new microservices architecture, including two of the 10 most business-critical for the bank. Read the full case here: https://lnkd.in/e_c5fbg

- **Kubernetes success story at Pinterest**

With over 250 million monthly active users and serving over 10 billion recommendations every single day, that is huge. (The numbers might have changed now)



As they knew these numbers are going to grow day by day, they began to realize the pain of scalability and performance issues.

Their initial strategy was to move their workload from EC2 instances to Docker containers; hence they first moved their services to Docker to free up engineering time spent on Puppet and to have an immutable infrastructure.

And then the next strategy was to move to Kubernetes:)

Now they can take ideas from ideation to production in a matter of minutes whereas earlier they used to take hours or even days. They have cut down so much of overhead cost by utilizing Kubernetes and have removed a lot of manual work without making engineers worry about the underlying infrastructure.

Read their impressive story - <https://lnkd.in/eTxwFXX>

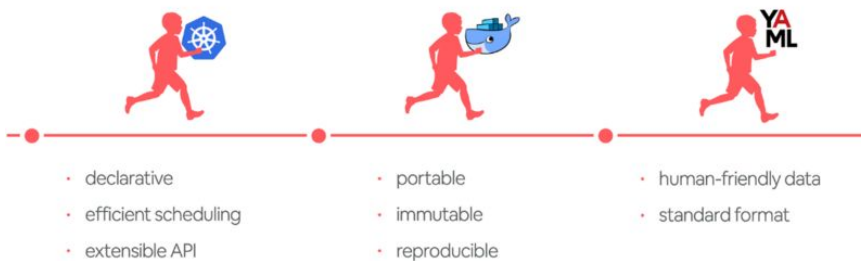
• Airbnb's Kubernetes story

Airbnb's transition from a monolithic to a microservices architecture is pretty amazing. They needed to scale continuous delivery horizontally, and the goal was to make continuous delivery available to the company's 1000 or so engineers so they could add new services. Airbnb adopted to support over 1000 engineers concurrently configuring and deploying over 250 critical services to Kubernetes (at a frequency of about 500 deploys per day on average).



Why kubernetes?

@MELANIECEBULA



Info credits: Slides by Melanie Cebula at QCon London 2019

I want you to see this excellent presentation from Melanie Cebula, the infrastructure engineer at Airbnb: <https://lnkd.in/eZWpme3> Also, take a look at these Kubernetes best practices from frog's mouth - <https://lnkd.in/eZtxx-Z>

● The New York Times Kubernetes story

Today the majority of their customer-facing applications are running on Kubernetes.

What an amazing story:)

The biggest impact has been to speed up deployment and productivity. Legacy deployments that took up to 45 minutes are now pushed in just a few.



It's also given developers more freedom and less bottlenecks.

The New York Times has gone from a ticket-based system for requesting resources and weekly deploy schedules to allowing developers to push updates independently.

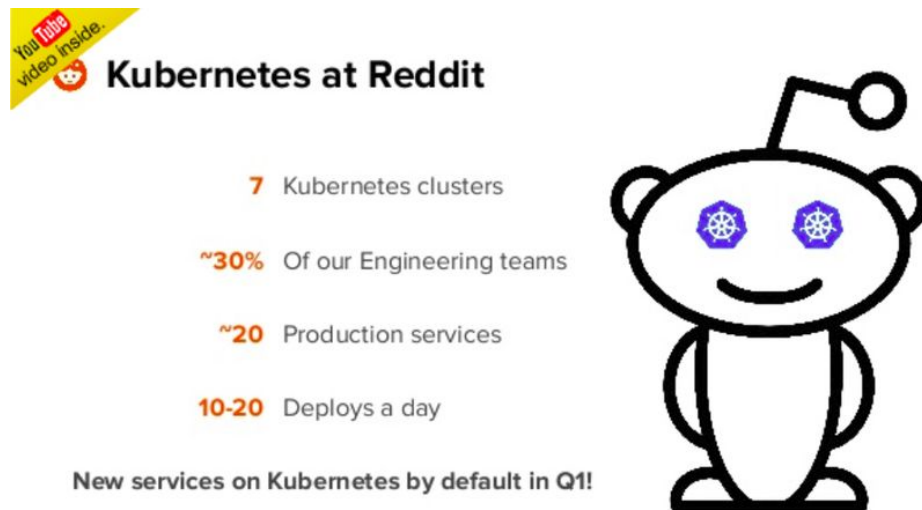
Check out the evolution & the fascinating story of The New York Times tech stack -

<http://bit.ly/nytechstack>

- **Kubernetes at Reddit** 🚀🚀🚀

Reddit is one of the top busiest sites in the world.

Kubernetes forms the core of Reddit's internal Infrastructure.



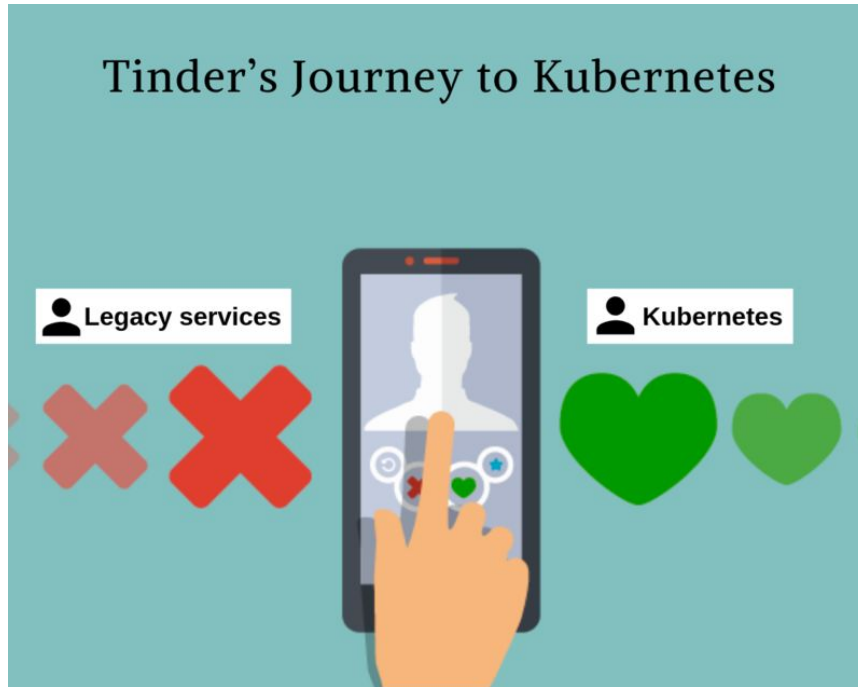
For many years, the Reddit infrastructure team followed traditional ways of provisioning and configuring. However, this didn't go long until they saw some huge drawbacks and failures happening while doing the things the old way.

They moved to 'Kubernetes.'

See this amazing video where their infrastructure release engineering manager describes the Kubernetes story at Reddit. <https://lnkd.in/eSH2H8K>

- **Tinder's Kubernetes story:**

Due to high traffic volume, Tinder's engineering team faced challenges of scale and stability. What did they do? Kubernetes.' Yes, the answer is Kubernetes.



Tinder's engineering team solved interesting challenges to migrate 200 services and run a Kubernetes cluster at scale totaling 1,000 nodes, 15,000 pods, and 48,000 running containers.

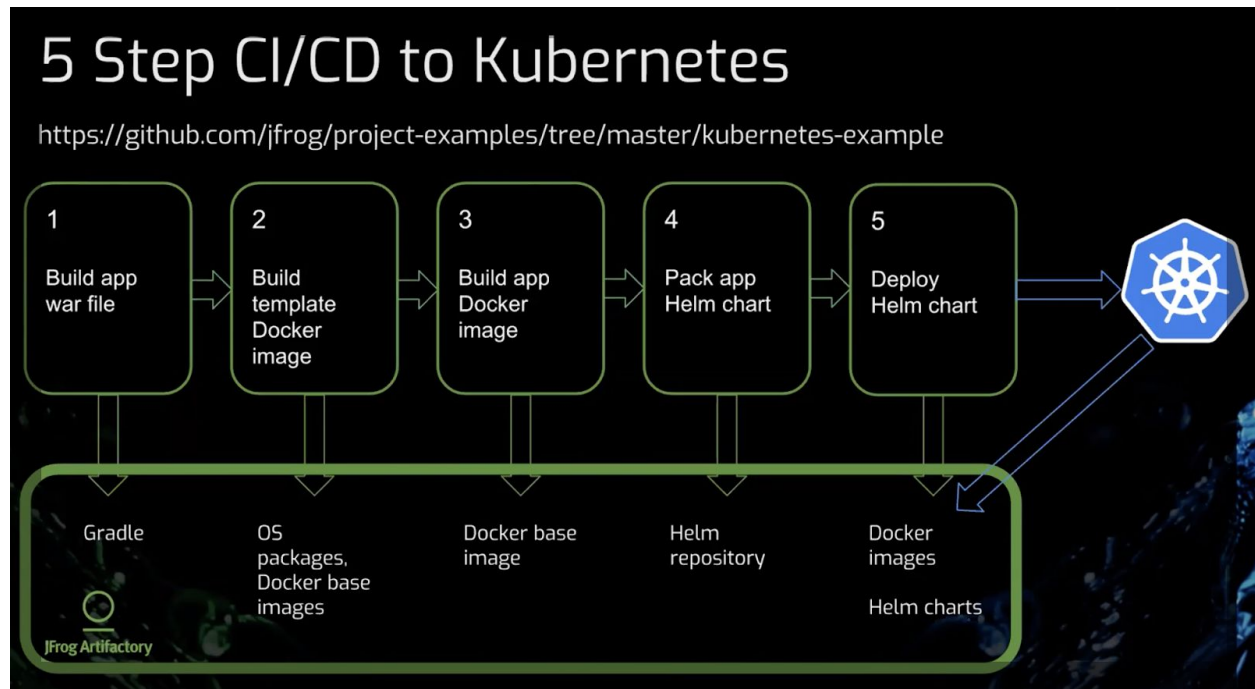
Was that easy? No ways. However, they had to do it for the smooth business operations going further.

One of their Engineering leaders said, "As we onboarded more and more services to Kubernetes, we found ourselves running a DNS service that was answering 250,000 requests per second."

Fantastic culture, Tinder's entire engineering organization now has knowledge and experience on how to containerize and deploy their applications on Kubernetes.

Read this fascinating case study - <http://bit.ly/KubernetesatTinder>

- Try This Simple 5-Step Kubernetes CI/CD Process



Step 1. Develop your microservice. This can be a .war or .jar file.

Step 2. Create a Docker Framework using Tomcat and Java-8 on Ubuntu as a base image.

Step 3. Create the Docker image for the microservice by adding the .war/.jar file to the Docker Framework.

Step 4. Create a Helm chart for the microservice.

Step 5. Deploy the microservice to the Kubernetes cluster using the Helm Chart.

Try: <http://bit.ly/CICDPROCESS>







- Take a look at this Kubernetes production checklist.

10 Kubernetes Production Checklist



- | | |
|----------------------------------|------------------------------|
| 1 Provision and deploy | 6 Backup/Restore |
| 2 Installation and Configuration | 7 Networking |
| 3 Security | 8 HA, DR, and Scalability |
| 4 Monitoring and Performance | 9 Cost optimization |
| 5 Logging | 10 Documentation and testing |

- Here are some more tips for taking your containers all the way to production

TASKS	QUESTIONS	
Logging	<ul style="list-style-type: none"> ■ How is your application logging set up? ■ Where will the logs be saved? ■ Do you need logs files or perhaps using STDOUT/STDERR is sufficient? ■ How will you handle multiple log files? 	 <p>Consider turning your logs to symlinks by setting /dev/stdout or /dev/stderr thereby ensuring all your logs are part of the container log.</p> <p>K8S GURU TIP</p>
Data Persistency	<ul style="list-style-type: none"> ■ Is your application stateful? ■ Does it require data persistence? ■ What part of your data needs persistency? 	 <p>Don't store all your data on a persistent storage. Store only persistent data.</p> <p>K8S GURU TIP</p>
Termination Signals	<ul style="list-style-type: none"> ■ How do you handle termination signals? 	 <p>Use trap in your container bash entrypoint to catch termination signals and handle them properly.</p> <p>K8S GURU TIP</p>
Application Restart	<ul style="list-style-type: none"> ■ How will you survive a restart? ■ What happens if you kill the pod? ■ What happens if you crash the process in the pod? ■ What happens if the k8s node crashes? ■ How does the application behave? 	 <p>A great way to test your application recovery is to kill the pods or, kill the nodes, and see what happens?</p> <p>K8S GURU TIP</p>
High Availability	<ul style="list-style-type: none"> ■ How should I set up my nodes and load balancer to achieve zero service unavailability of my application/service? 	 <p>Plan for zero service unavailability allowing for pod scheduling when performing cluster scaling (down) and planned node maintenance.</p> <p>K8S GURU TIP</p>
Probes	<ul style="list-style-type: none"> ■ Do your applications have endpoints that can be used to check health and readiness using the Liveness and Readiness Probes? 	 <p>Proper use of probes can help you implement a great "auto-healing" process for your applications and will save your engineers many sleepless nights.</p> <p>K8S GURU TIP</p>

Know more: <http://bit.ly/2J9ieeU>

Just a small favor, can you please follow me on Twitter?

Here is my profile - https://twitter.com/Pavan_Belagatti

Thanks:)