Kubernetes For Everyone

Kubernetes introduction and features

How Kubernetes works?

In Kubernetes, there is a master node and multiple worker nodes, each worker node can handle multiple pods.
Pods are just a bunch of containers clustered together as a working unit. You can start designing your applications using pods.
Once your pods are ready, you can specify pod definitions to the master node, and how many you want to deploy. From this point, Kubernetes is in control.
It takes the pods and deploys them to the worker nods. If a worker node goes down, Kubernetes starts new pods on a functioning worker node.
This makes the process of managing the containers easy and simple.
It makes it easy to build and add more features and improving the application to attain higher customer satisfaction.

Finally, no matter what technology you're invested in, Kubernetes can help you.

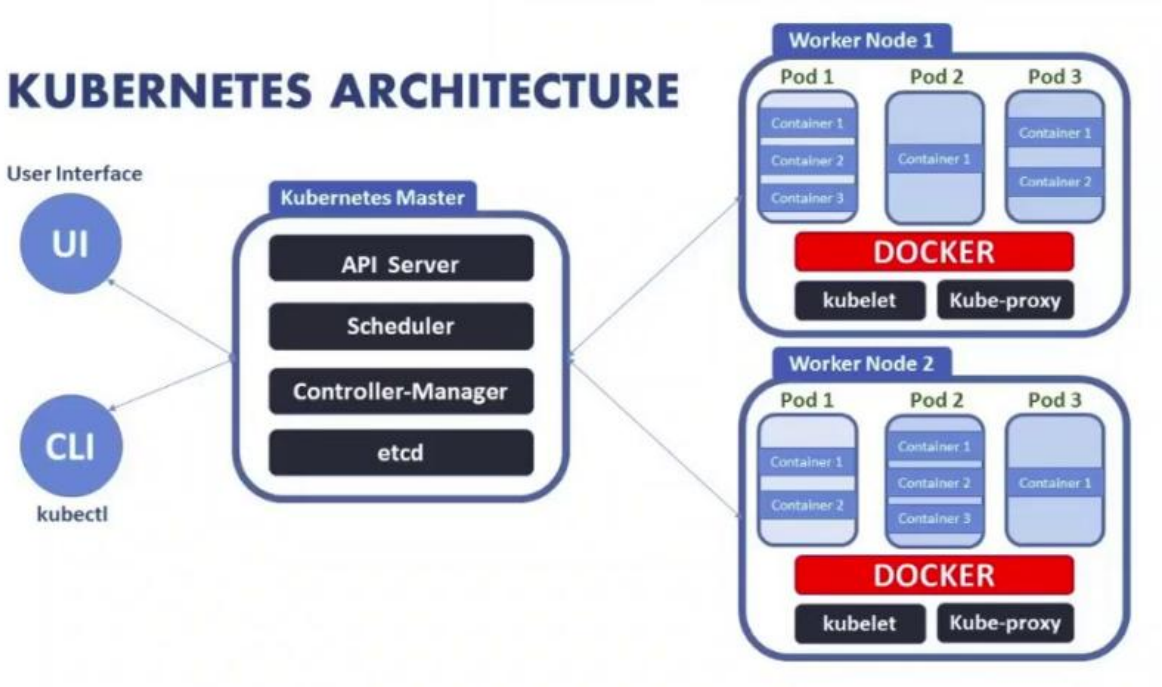


Image credits: Source: Knoldus Inc

What is the Master node and Worker node in #Kubernetes?

Explained below,

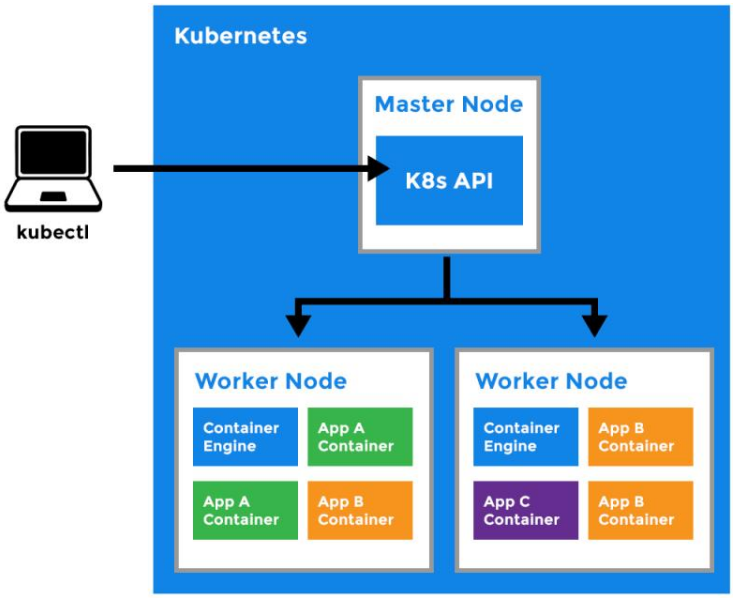
#Containerization is the trend that is taking over the world, allowing firms to run any kind of different applications in a variety of different environments. To keep track of all these containers, to schedule, to manage, and to orchestrate them, we all require an orchestration tool. Kubernetes does it exponentially well.

Kubernetes is a master-slave type of architecture. It operated with Master node and worker node principles.
What exactly they do?

Master Node:
>The main machine that controls the nodes
> Main entry point for all administrative tasks
> It handles the orchestration of the worker nodes

Worker Node:
> It is a worker machine in Kubernetes (used to be known as a minion)
> This machine performs the requested tasks. The Master Node controls each Node
> Runs containers inside pods
> This is where the Docker engine runs and takes care of downloading images and starting containers

Know in-depth concepts here in the original article:
https://blog.risingstack.com/what-is-kubernetes-how-to-get-started/



---

*[The most awaited DevOps conference 'swampUP' is happening virtually - **Save Your Seat**]*

#Containers are the de-facto deployment format of today.
But where does #Kubernetes comes in the play?

While tools such as #Docker provide the actual containers, we also need tools to take care of things such as replication, failovers, orchestration, and that is where Kubernetes comes into play.

The Kubernetes API is a great tool for automating a deployment pipeline. Deployments are not only more reliable, but also much faster, because we're no longer dealing with VMs.

When working with Kubernetes, you have to become accustomed with concepts and namings like pods, services, and replication controllers. If you're not already familiar yet, no worries, there are some excellent resources available to learn Kubernetes and get up to speed.

Some key features of Kubernetes that make it unique,

> Service Discovery
> Health Check Capability
> Simplified Monitoring
> Self-healing
> Secret and configuration management
> Horizontal scaling
> Storage Management
> Networking
> Services
> ConfigMap and Secret
> Logging
> Rolling update or rollback
> Load balancing

What feature do you like the most in Kubernetes?

BTW, take a look at these tips, tricks, and lessons for taking containerized apps to k8s:
https://lnkd.in/eZtxx-Z

---

*[The most awaited DevOps conference 'swampUP' is happening virtually - **Save Your Seat**]*

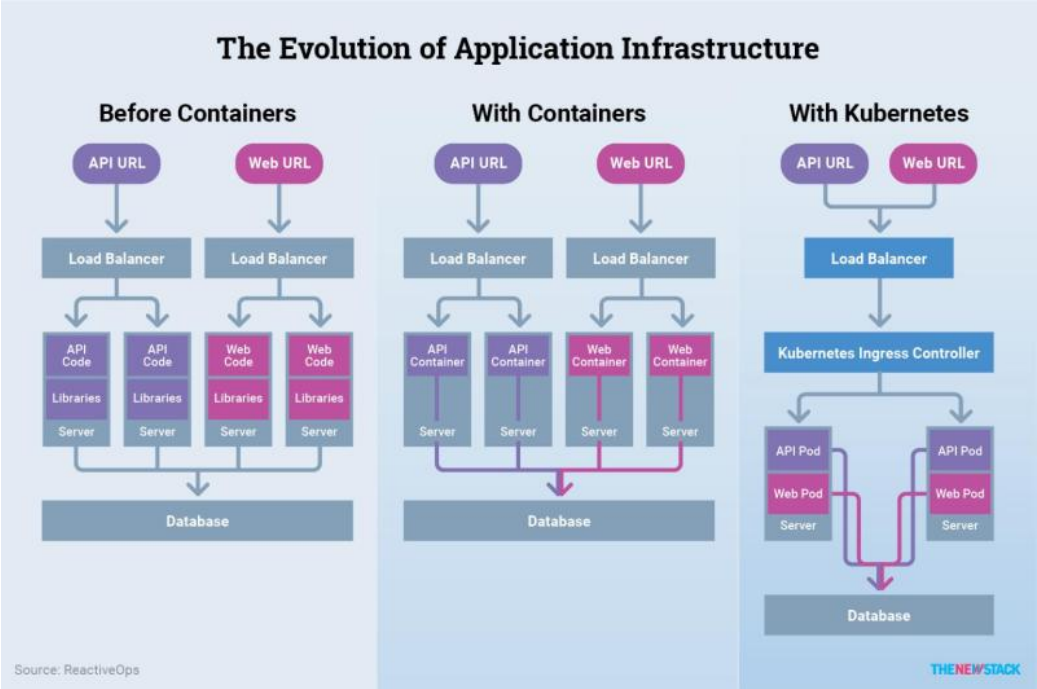


Image credits: TheNewStack

#Kubernetes features that we all like:)

> Automatic binpacking: This is where Kubernetes helps in automatically placing containers based on their resource requirements, limits, and other constraints, without compromising on availability.

> Service discovery and load balancing: In simple words, service discovery is the process of figuring out how to connect to a service.

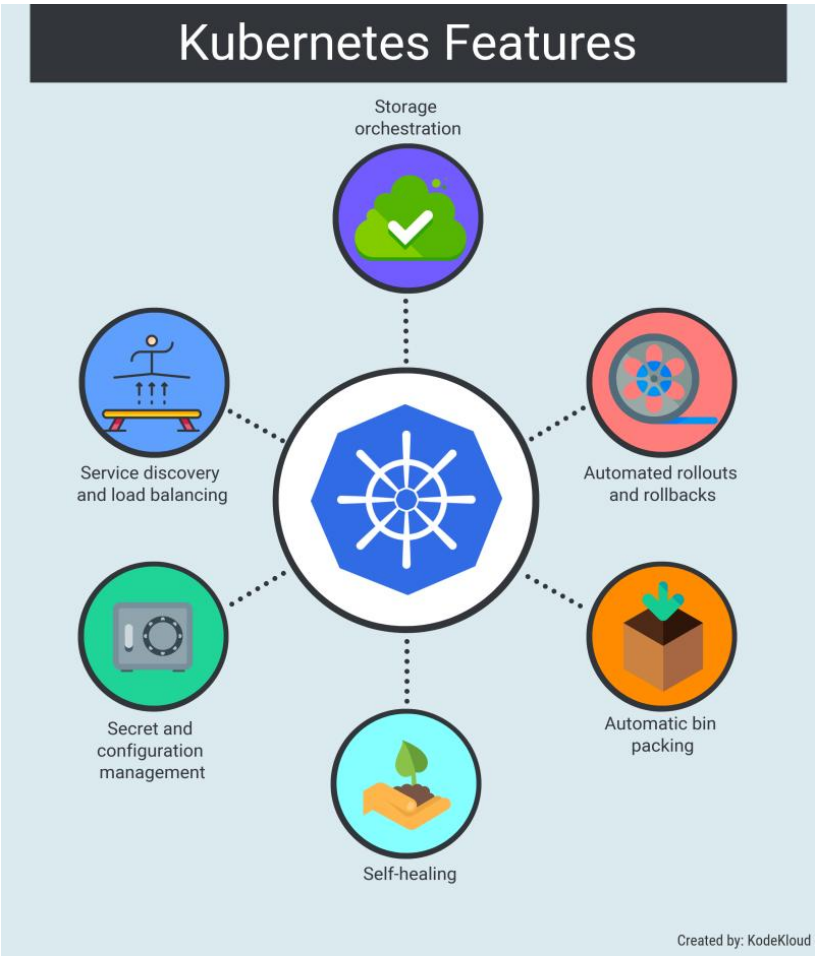
>Self-healing: Restarts the containers that fail, replaces, and reschedules containers when nodes die.

> Automated rollouts and rollbacks: With this feature, Kubernetes does progressively roll out changes, and it ensures it doesn't kill all your instances at the same time.

> Secrets and configuration management: Kubernetes has a built-in mechanism of storing configuration values that you would prefer to keep private. Sensitive information such as user name, passwords with encryption, and other credentials can be kept confidentially.

> Storage orchestration: Automatically mount the storage system of your choice, whether from local storage, a public cloud provider such as GCP or AWS, or a network storage system such as NFS, iSCSI, Gluster, Ceph, Cinder, or Flocker.

See more in the original article: https://lnkd.in/e8MzdeV

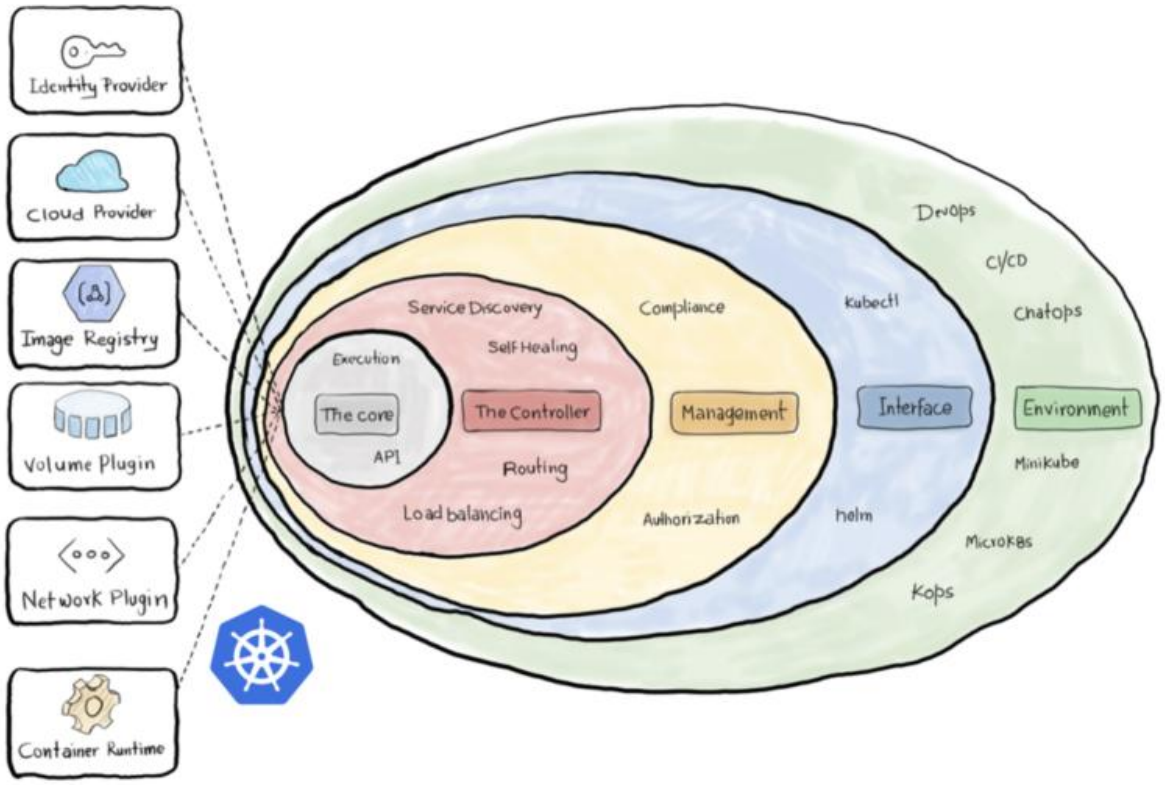


#Kubernetes helps in easy automation of container lifecycle management.
However, the concept of Kubernetes is quite complex.

Here are the top practices of Kubernetes:)

> Dis-allow root users
> Dis-allow privileged containers
> Use pod resource limits and requests
> Dis-allow new capabilities addition
> Dis-allow any changes to the parameters of the kernel
> Dis-allow using bin mounts
> Use read-only root filesystem
> Dis-allow dock socket bind mount access
> Dis-allow usage of host ports and networks
> Keep smaller base image
What else? Know more: https://lnkd.in/ejetevG

#Kubernetes Core Features.



1. Container runtime: Kubernetes uses Container Runtime Interface (CRI) to transparently manage your containers without necessarily having to know (or deal with) the runtime used.

2. The Network Plugin: As we discussed earlier, a container orchestration system is responsible (among other things) for managing the network through which containers and services communicate.

3. The Volume Plugin: A volume broadly refers to the storage that will be availed for the pod.

4. Image Registry: Kubernetes must contact an image registry (whether public or private) to be able to pull images and spin out container.

5. Cloud Provider: Kubernetes can be deployed on almost any platform you may think of.

6. Identity Provider: You can use your own identity provider system to authenticate your users to the cluster as long as it uses OpenID connect. Read this amazing article: https://lnkd.in/eySj5aG

Kubernetes setup:

How much time can you devote to setting up the #Kubernetes? 🔩🔧⚙️

That's the question to ask yourself.

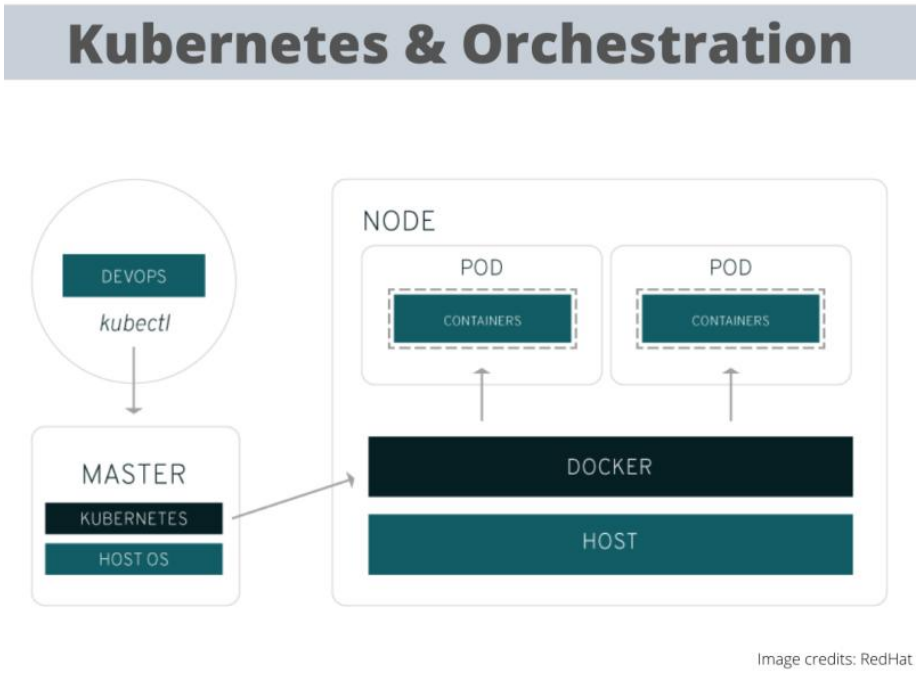


Image credits: RedHat

Because installing and setting up Kubernetes can be daunting. Yes!

Kubernetes itself (meaning the plain, open-source version) does not have a built-in installer, nor does it offer much in the way of one-size-fits-all default configurations. You'll likely need to tweak (or write from scratch) a lot of configuration files before you get your cluster up and running smoothly.

Thus, the process of installing and configuring Kubernetes can be a very daunting one that consumes many days of work.
Some Kubernetes distributions offer interactive installer scripts that help automate much of the setup process. If you use one of these, setup and installation is easier to accomplish in a day or two. But it's still by no means a turnkey process.

A third option is to run Kubernetes as a managed service in the cloud using a solution like Google Kubernetes Engine, but the downside is, you have less choice and control in determining how to configure your Kubernetes environment.

Know more on the facts to consider before selecting Kubernetes: bit.ly/KubernetesSetup

---

#Kubernetes autoscaling. How does it work?

Scaling is an essential operational practice that used to be done manually for a long time concerning applications, with the introduction of tools like Kubernetes, the things have changed dramatically in the software industry.

In the context of the Kubernetes cluster, there are typically two things you would like to scale as a user, Pods, and Nodes.

There are three types of scaling:
> HorizontalPodAutoscaler
> VerticalPodAutoscaler, and
> Cluster Autoscaler.

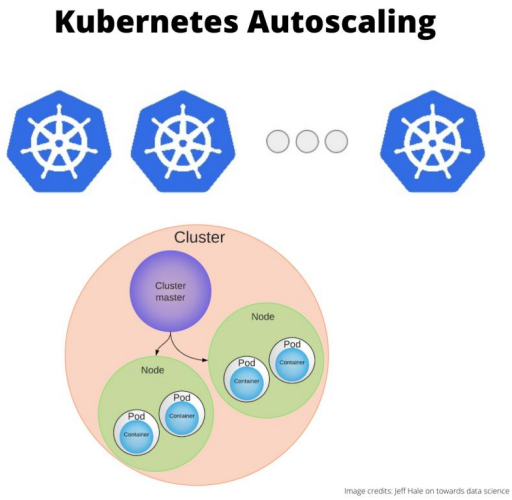
With these techniques, Kubernetes can take intelligent scaling decisions automatically.

HorizontalPodAutoscaler refers to increasing the number of Pods serving the application, in response to the present computational needs.

VerticalPodAutoscaler involves expanding the resources of the Pods.

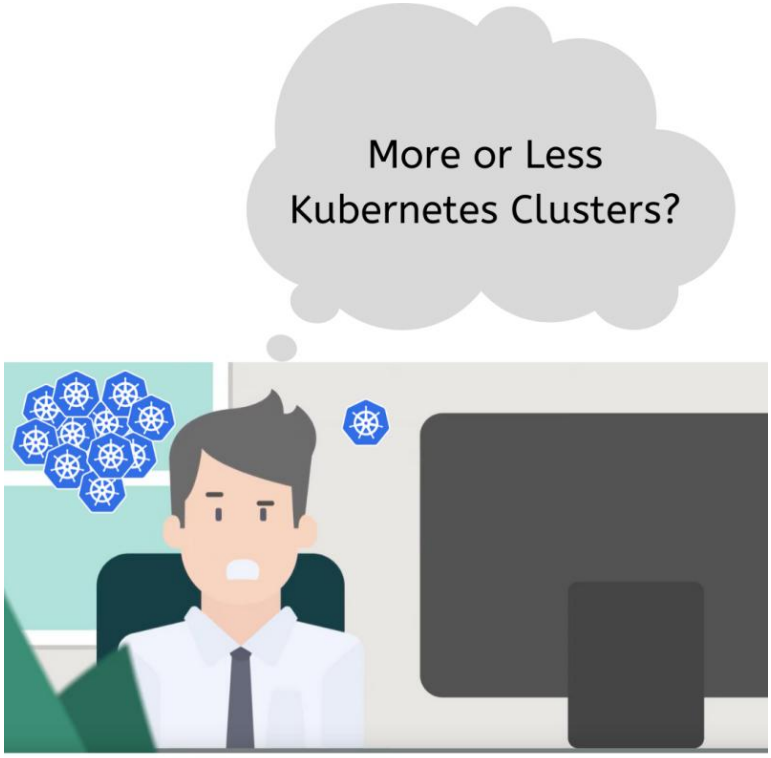
Cluster Autoscaler (CA) scales your node clusters based on the number of pending pods. It checks to see whether there are any pending pods and increases the size of the cluster so that these pods can be created.

Mastering autoscaling needs some patience and persistent efforts to see which technique suits your app's needs by doing trial and error.
Continuous learning and experimentation is the key:)



More or less #Kubernetes clusters.

How to decide?



Kubernetes is devised as a highly available cluster of computers that are connected to work as a single unit for more power and efficiency.
The cluster forms the heart of Kubernetes: It can schedule and run containers across a group of machines, be they physical or virtual, on-premises or in the cloud.

A Kubernetes cluster is formed out of 2 types of resources:
> Master is coordinating the cluster
> Nodes are where we run applications

Kubernetes is still a bit of sophisticated technology and has a steep learning curve, even after a couple of years working with it, you'll still wonder if you got it all under control.

But when your company asks you to decide on using and implementing Kubernetes, one question you will have is, deciding on the Kubernetes clusters.

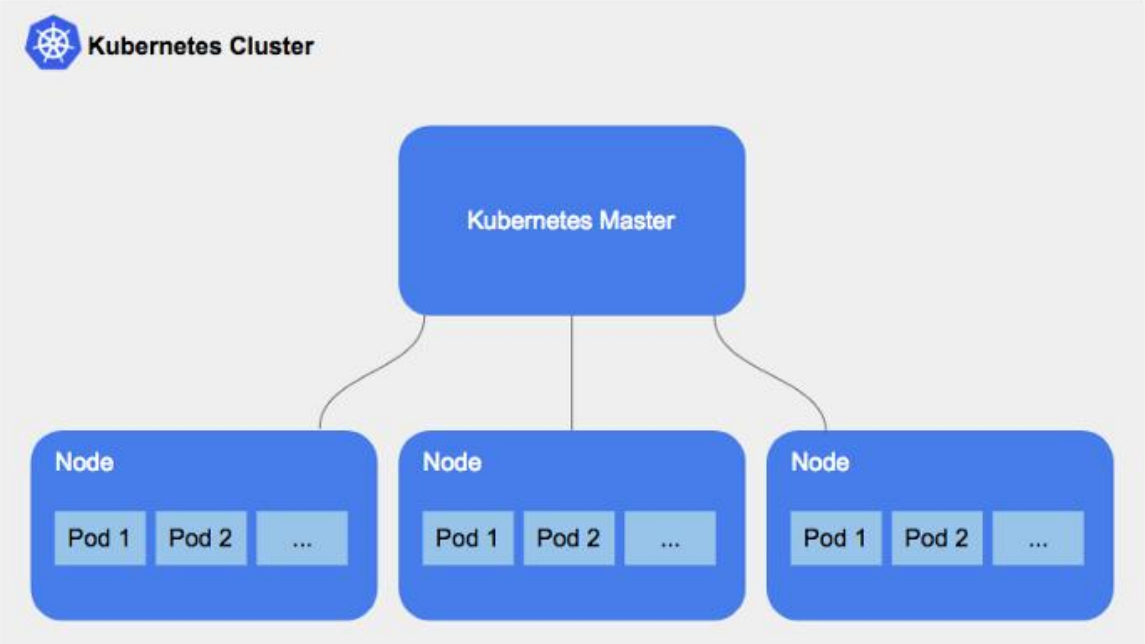
My friend Sander has written an amazing article on this, take a look - https://lnkd.in/eSC5vpa

---

Kubernetes security:

Keep your clusters updated with the latest #Kubernetes security patches.

See how and why...



Just like any application, Kubernetes is continuously updating new features and security updates. Hence, it is imperative that the underlying nodes and Kubernetes clusters need to be in parallel and up to date as well.

The standard “zonal” Kubernetes Engine clusters will have only one master node backing them, but you can create “regional” clusters that provide multi-zone feature, highly available masters. One crucial thing to remember here is, while creating a cluster, be sure to select the “regional” option.

By using Kubernetes Engine, you can keep your Kubernetes cluster up to date with just a few clicks. It is highly recommended to use Kubernetes Engine regional clusters for the high-availability masters and automatic node upgrades to have a hassle-free upgrade experience.
(Source: cloud.google.com)

See my in-depth article on Kubernetes security best practices: https://lnkd.in/eZq9mGs

---

In 2018, a severe vulnerability in #Kubernetes (CVE-2018–1002105) was disclosed...

This vulnerability allowed an unauthorized and unauthenticated user to gain full admin privileges on a cluster and perform privilege escalation.

In one more incident, a security firm RedLock said that hackers accessed one of Tesla's Amazon cloud accounts, and they used it to run cryptocurrency-mining malware. The initial point of entry for the Tesla cloud breach was an unsecured administrative console for Kubernetes.
So many scary stories!

Do you know how important is security in Kubernetes?
Know here in my in-depth article - https://lnkd.in/ecviJ6c

'Role-Based Access Control' is one of the security best practices in #Kubernetes.

Know more about it below...

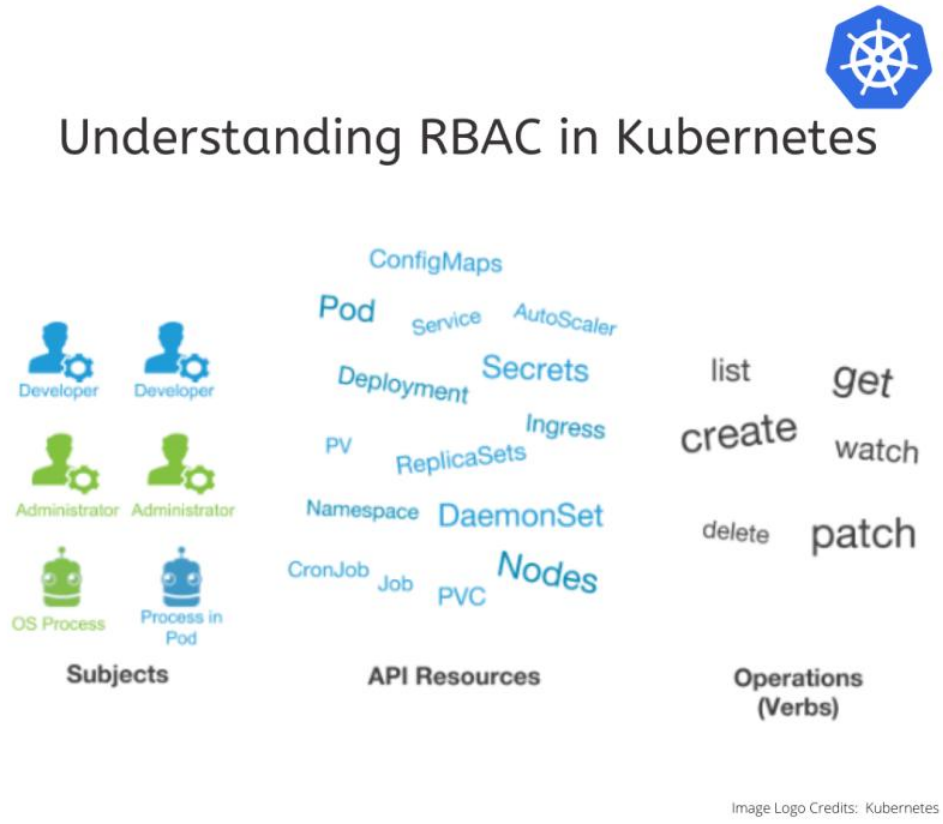


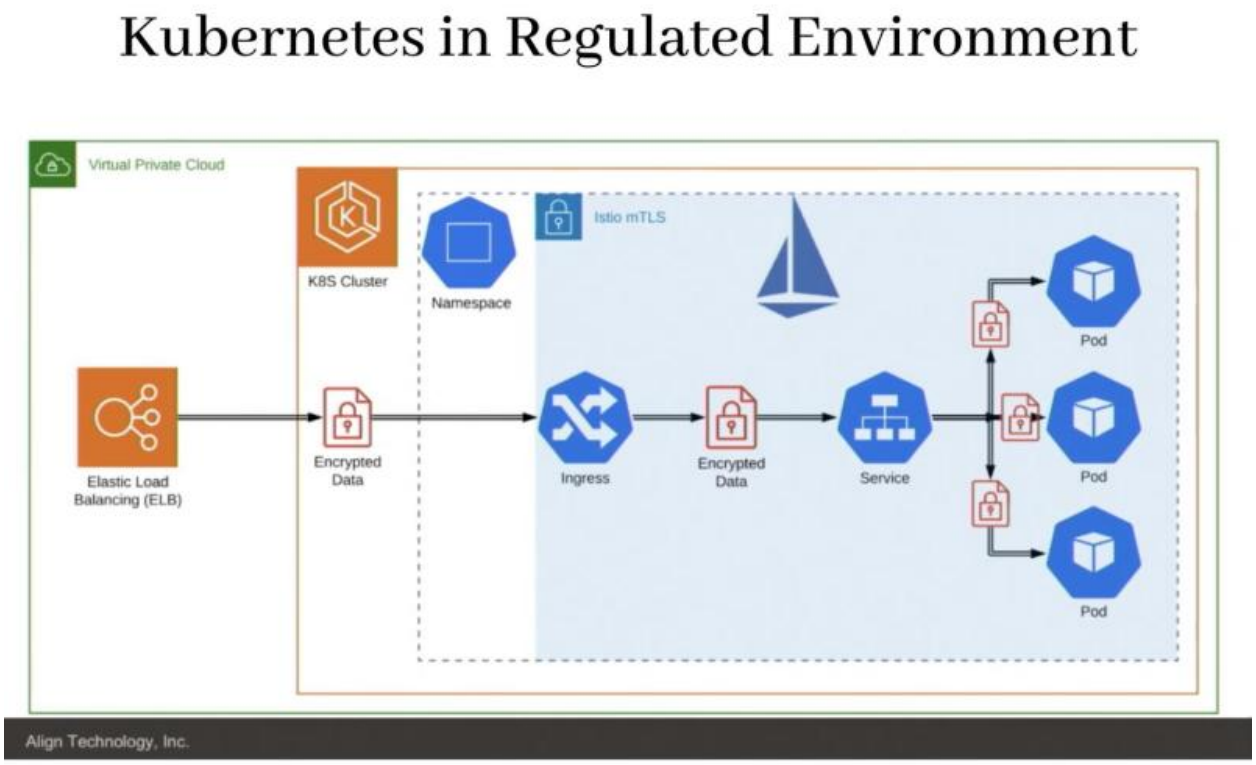
Image Logo Credits: Kubernetes

RBAC allows Kubernetes architects to specify which types of actions are permitted for a user and what kind of tasks they are going to perform depending on their role in the organization This is how you create roles based on the different kinds of access your users and applications need to various resources, and later assign only the required and minimum permissions for appropriate access to the roles. Minimum or restricting access to only specified and well-identified users who must perform defined actions on a resource is critical in securing your cluster and is one of the security best practices. To tighten the security and ease handling a large number of accounts, RBAC makes use of an intermediate item called binding. Via role binding mechanism, you can create “roles,” which will have a set of capabilities, then assign each user one or more roles. For example, some users might just have permission to list pods, and some other users may have permission to get, list, watch, create, update, patch, delete pods. Writing an article on this and will be out soon.

No doubt, It is highly challenging to embrace #CloudNative and #DevOps in regulated industries.

As #microservices and container-based infrastructure are enriching how the software is built these days, new challenges with security and compliance appear for regulated firms.

Regulated industries imply several challenges
> Strong restrictions on secured networks
> Fine-grained audit trails
> Strong ACLs models
> Full lifecycle governance
> Integration with 3rd parties



Here is an example where Artem Semenov from Align Technology is showing us the basic requirements for making #K8S compliant with sensitive data handling regulations and possible technical solutions - https://lnkd.in/g2G-rFX

Kubernetes case studies:

Delivering Pizza with #Kubernetes?

Domino's #CloudNative story is mind-blowing. Read:)

The company makes splashy headlines with ambitions to deliver pizzas with driverless cars and drones, along with AI and other automation processes. Domino's currently offers 15 digital ways to order pizza.

How does Domino's deliver so many new solutions, features, and updates, while it's hot? By cultivating an experimental culture of cloud-native innovation within the company.

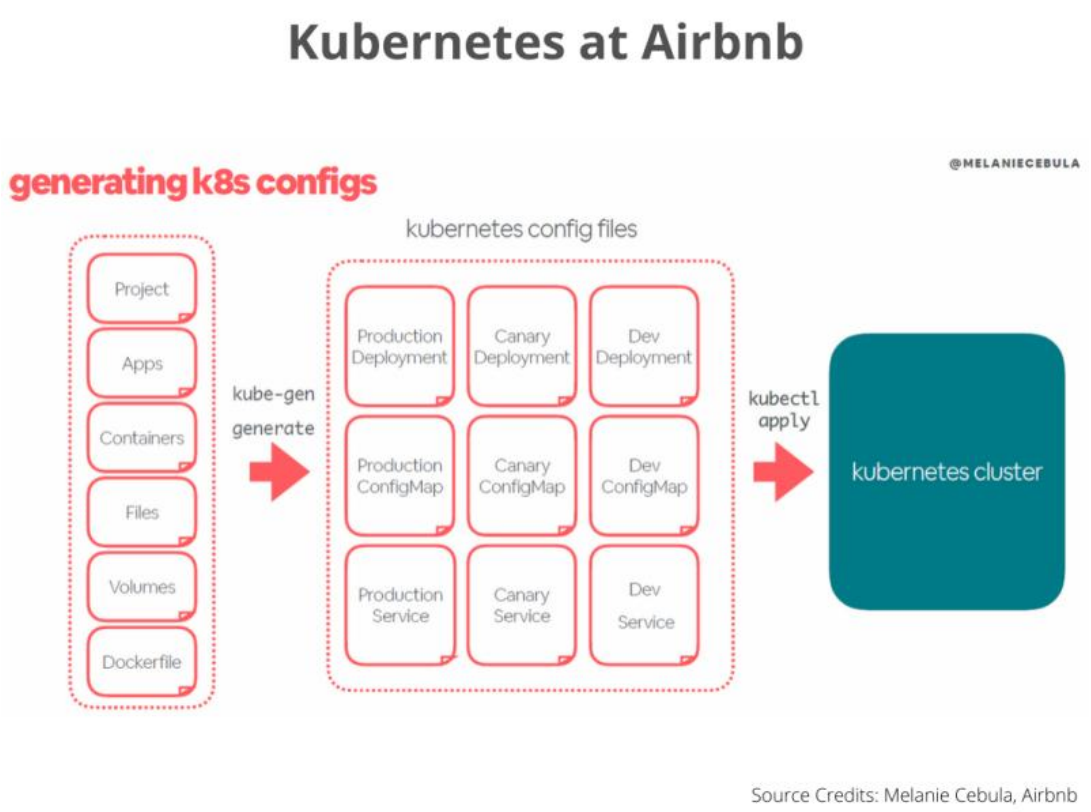
Applications play a significant role in Domino's business strategy.
Domino's intends to create new business value and speed their time to market by rewriting core applications to run as microservices.
Dominos teams are modernizing these core applications in-house with microservices, but each team uses a different platform.

The in-house teams chose a comprehensive, production-grade Kubernetes distribution platform with enterprise security features and full lifecycle management support and with Kubernetes, Domino's is

evaluating the feasibility and level of effort to convert current systems and processes to a container-based architecture.

Read the full story: https://lnkd.in/eZzagGH
Kubernetes best practices for taking your containers all the way to production: https://lnkd.in/eZtxx-Z

How did Airbnb enable 1,000+ engineers with #Kubernetes?



Source Credits: Melanie Cebula, Airbnb

This is the talk summary of Melanie Cebula at Qcon London. About the way her team wraps Kubernetes into easy-to-consume internal services for its development teams.

Instead of creating a set of dreaded YAML files per environment, development teams need only provide their project-specific, service-focused inputs and then run the internal service kube-gen (alias k gen).
This simple command takes care of generating all the required YAML files, ensuring their correctness, and finally applying them in the corresponding Kubernetes cluster(s).

The infrastructure team at Airbnb is saving hundreds, if not thousands, of hours for 1,000+ engineers who can now use a much simpler abstraction that has been adapted to their needs, with a user experience that's familiar to them.

Know more about this story at: https://lnkd.in/egDqpHE

The figure above shows the kube-gen wrapper generates the needed configuration files per environment at Airbnb. Source: Melanie Cebula, Airbnb.

A 50-year-old audio company using #Kubernetes? That's insane. Read:)

It supports rapid development for millions of #IoT products with Kubernetes. How?

Bose has been a big player in helping IoT devices and audio enabling systems for several years. Bose engineering leadership team always wanted to move to a microservices architecture.

When the demand started growing, they had to look for a solution that can help their engineering platform team to deploy services to production quickly without any hassle. For this, they evaluated and found many alternative platforms but finally chose Kubernetes due to its scaled IoT platform-as-a-service running on AWS and vibrant community aspect.

They launched a revised platform along with Prometheus, An open-source monitoring system to serve more than 3 million connected IoT devices. Today, Bose has over 1,800 namespaces and 340 worker nodes in one of its live production clusters. Bose has more than 100 engineers working on this platform already, this platform is helping them make 30,000 nonproduction deployments every year.
Read the original story: https://lnkd.in/eJgBRHN

Also, take a look at this video on CI/CD enablement for connected device products with OTA capabilities: http://bit.ly/IoTCICD

Amadeus's lift and shift to #Kubernetes.

An inspiring #CloudNative story for you. Read.....

Amadeus had two choices: Either pour more concrete and extend the data center or move the workload to the cloud & this made them go with Google Cloud.

So within 18 months, Amadeus had lifted and shifted one of their most critical application 'Master Pricer' to the Google Cloud Platform.
Now you know, the next step for them was to move to Kubernetes since it made more sense with Google Cloud Platform.

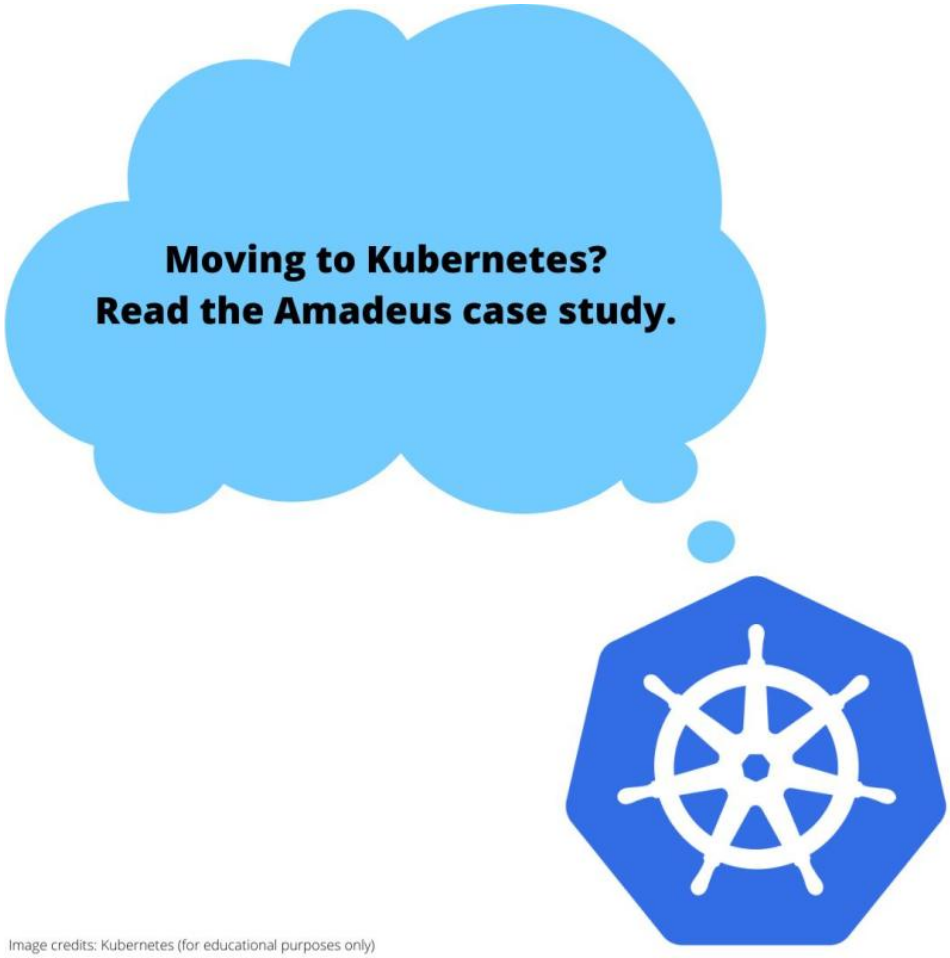


Image credits: Kubernetes (for educational purposes only)

The aim is, they wanted to go faster with Kubernetes, and the challenge was to add a disciplinary policy of learning Kubernetes across the team.
So, the team at Amadeus started learning how to operate Kubernetes and how to monitor it, do alerting.

During the migration, Amadeus had engineers from both Google and Red Hat on site to help them get to grips with OpenShift and the container orchestration technology Kubernetes.
The overall goal for Amadeus is to move all production workloads to run on a single operating model with Kubernetes.

The company now feels it made the right bet.
Credits:
https://lnkd.in/gS3n_vy
http://bit.ly/AmadeusGCP

#Kubernetes has helped Adidas to deploy faster, safer, with more quality and scale quickly.
Read further...

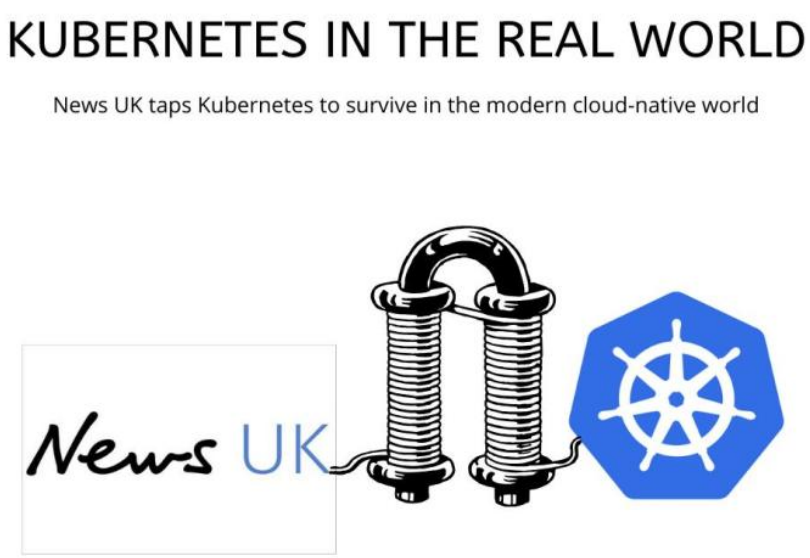
Adidas understood the importance of Kubernetes over VMs, and now their tech stack is wholly powered by Kubernetes. Before creating a VM would take days or even weeks that would impact the developers' productivity and the business overall.
Kubernetes helped Adidas to get rid of the overhead that comes with a VM-based infrastructure.

Deployments that used to take four to five days can now be deployed four to five times a day with the help of Kubernetes. Currently, Adidas has over 4,000 pods running on Kubernetes, achieving the velocity it needs to develop applications faster than ever.

Their lead infrastructure engineers say that, it is easy to set up and configure the new tool, but the problem comes in scaling.
They also stressed on the point that training is essential for engineers working on the platform.

Source credits: TechGenix

#Kubernetes is sexy because it attracts modern engineers that care about #CloudNative technologies. 😝😝😝



Read how News UK utilized the power of Kubernetes to save itself in the cloud-native world.
The critical goal for News UK was to be better able to scale up its environment around breaking news events & unpredictable reader volumes.

They thought if VMs can help them, but soon they realized that VMs take long to spin up and when there is a spike of traffic, it is not fast enough to bring new capacity into the AutoScalingGroup (that's what Marcin Cuber, a former cloud DevOps engineer at News UK has to say)

They adopted Docker and Kubernetes. Docker containers running in Kubernetes are smaller and lightweight, and they can easily help to scale up or scale down as required.

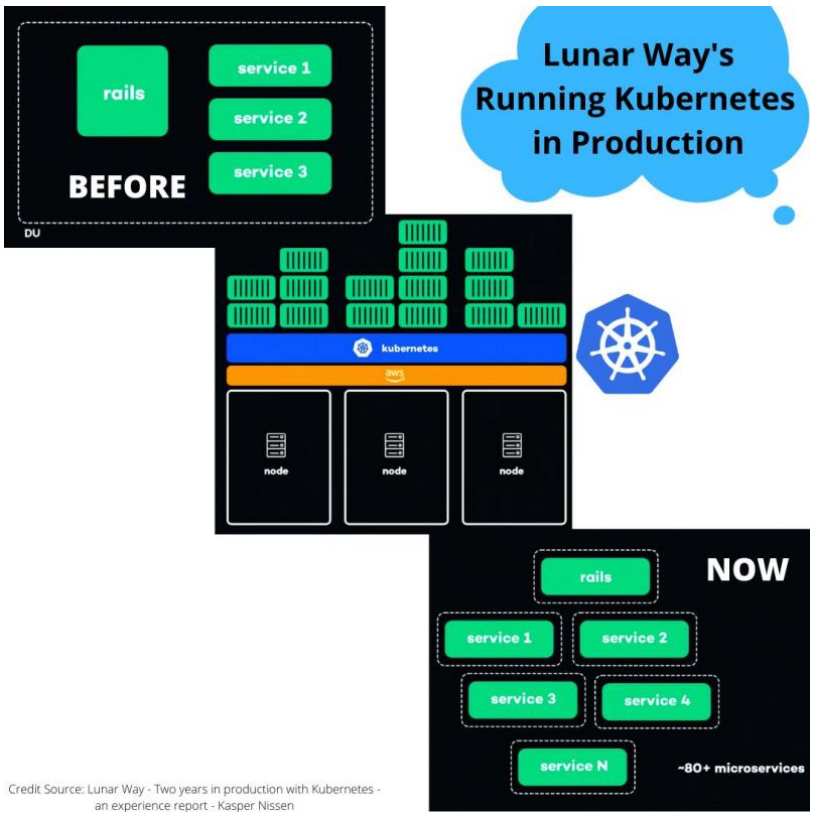
Cuber also had some advice for any organization looking to adopt Docker and Kubernetes.
> make your Docker images as small as possible and to focus on running stateless applications with Kubernetes
> run health checks for your applications and to use YAML to deploy anything

News UK also wanted to cut cloud costs, so they paired EKS clusters with AWS spot instances, and they also used AWS Lambda to make this work efficiently.

The full case study: https://lnkd.in/e4iAq2r

A banking app's must-read story of running #Kubernetes in production.

A journey that affirms you don't have to be too big to use Kubernetes.



They started their #CloudNative journey by splitting the massive monolith application into smaller microservices.
To spin up these microservices, they used Ansible, Terraform, and Jenkins and to deploy these microservices as a whole unit (as shown in the image).
Then they suddenly started to experience some of the scaling issues with Microservices. So, they didn't get any of the microservices benefits. Hence they started looking for ways to get out of this complexity by shifting their focus from machine-oriented to application-oriented architecture.

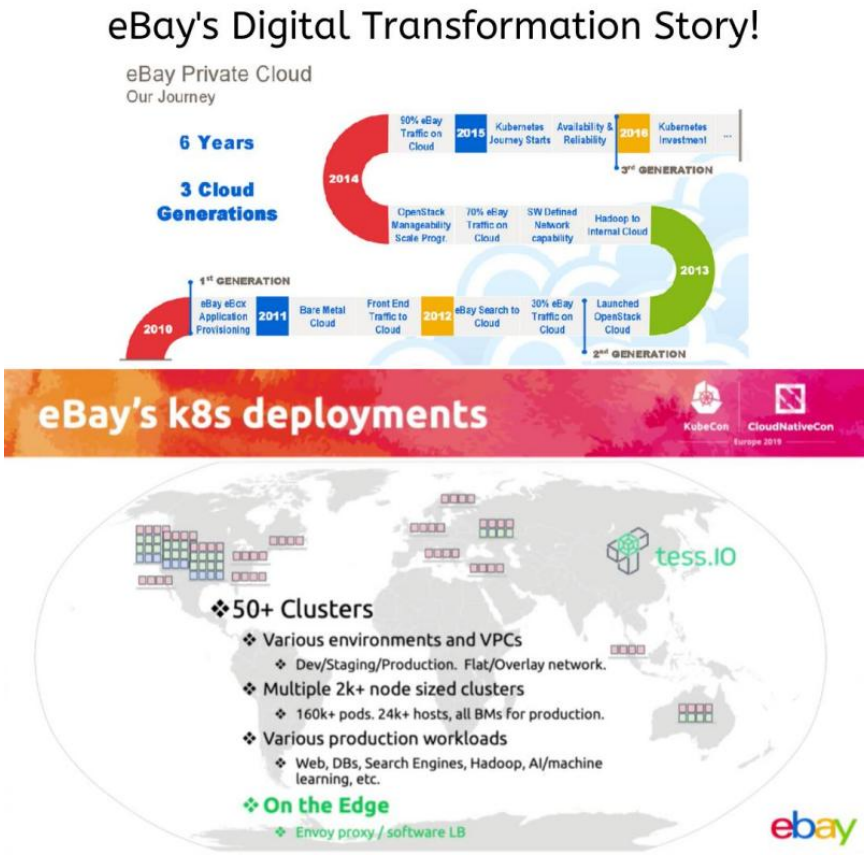
They chose Kubernetes as the abstraction layer along with AWS, not worrying about where the containers are running, and this is how they were able to manage microservices and unlocked the velocity of microservices. They also chose Kubernetes from a security perspective and to specify how the applications should run.

Now they run around 80+ microservices now in production with the help of Kubernetes:)

Watch and learn how they did it in this video 'Running Kubernetes in production at Lunar Way by Kasper Nissen.' - https://lnkd.in/eU9s3JX

Why did eBay choose #Kubernetes?

Daily, eBay handles 300 billion data queries & a massive amount of data that's above 500 petabytes.



eBay has to move massive amounts of data & manage the traffic, keeping in mind a smooth user experience while still ensuring a secure, stable environment that's flexible enough to encourage innovation.

In the fall of 2018, the company announced they were in the midst of a three-year plan they called "re-platforming."

eBay's 90% of cloud technology was dependent on OpenStack, and they are in the move to ditch OpenStack altogether.

eBay is "re-platforming, itself with Kubernetes, Docker, & Apache Kafka, a stream processing platform that increases data handling and decreases latency.

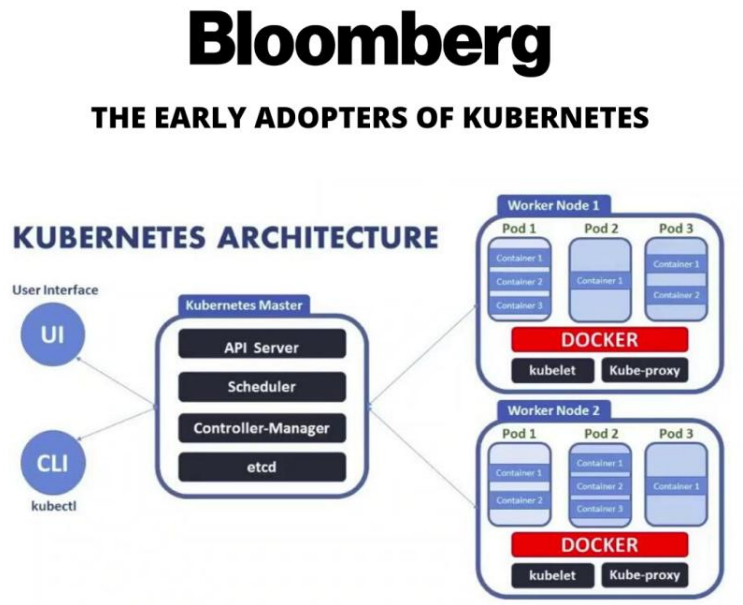
The goal is to improve the user experience and to promote productivity with their engineers and programmers & completely revamp its data center infrastructure.

The other activities in this re-platforming include designing their own custom servers and rolling out a new, decentralized strategy for their data centers. Like Facebook & Microsoft, eBay is relying on open-sourcing to design their custom servers.

Such an inspiring case study.

Bloomberg is one of the first companies to adopt #Kubernetes.

They used Kubernetes into production in 2017.



The aim was to bring up new applications and services to users as fast as possible and free up developers from operational tasks. After evaluating many offerings from different firms, they selected Kubernetes as they thought it aligned exactly with what they were trying to solve.

One of the key aims at Bloomberg was to make better use of existing hardware investments using different features of Kubernetes. As a result, they were able to very efficiently use the hardware to the point where they could get close to 90 to 95 percent utilization rates (as per Andrey Rybka, head of the compute infrastructure team at Bloomberg)

Nothing great comes easy; Kubernetes makes many things simpler only if you know how to use it. As the developers initially found it challenging to use, the teams had many training programs around Kubernetes at Bloomberg.

Shopify's #Kubernetes journey is just minded blowing:)

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.

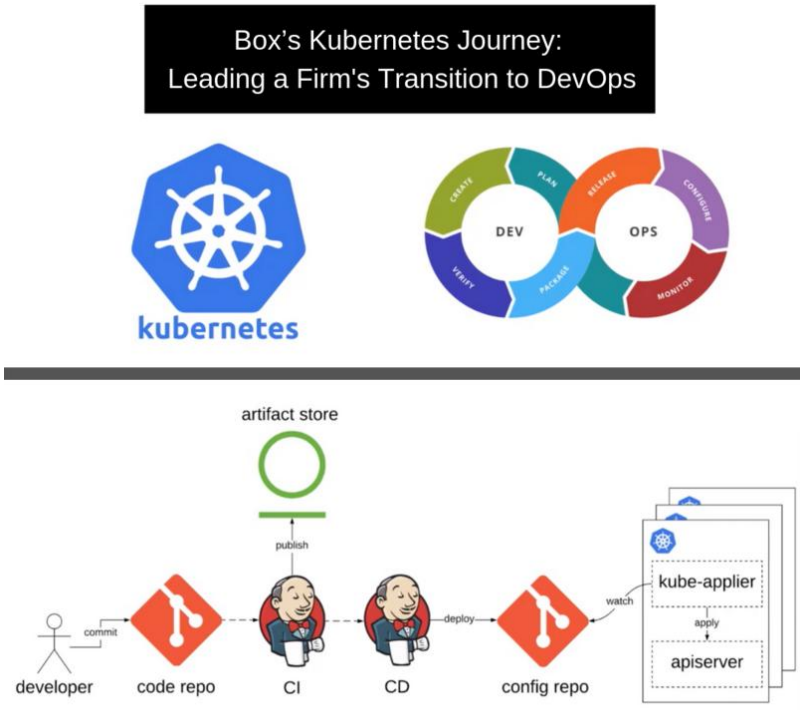
The Shopify engineering team believed in three principles: providing a 'paved road, 'hide complexity' and 'self-serve.'

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

All credits to Niko Kurtti, QCon & InfoQ.

Box's #Kubernetes journey is one of the finest #CloudNative inspirations. Read…

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

I hope you all are #GoT fans here...
Let me tell you the #Kubernetes story at HBO!

The engineers started panicking as they knew the unpredictable traffic for the most anticipated Game of Thrones season seven premiere is going to be HUGE.

One of the challenges they found out was the under-utilization of the deployed resources.
Node.js code tends only to use a single CPU core.
AWS EC2 instances that had excellent networking capabilities tended to be based on dual-core CPUs.
As such, HBO was only using 50 percent of the deployed CPU capacity across its deployment. The ability to spin up new instances on EC2 wasn't quite as fast as what HBO needed.

HBO also found that in times of peak demand for Game of Thrones, it was also running out of available IP addresses to help deliver the content to viewers.

"We went from not running a single service inside of a container to hosting all of Games of Thrones season 7 with Kubernetes," Illya Chekrygin, Senior Staff Engineer at HBO told the KubeCon audience.

At last, the HBO chose Kubernetes among other alternatives, basically because of its vibrant and active community.
Credits: KubeCon 2017, eWEEK

Italy's biggest traditional bank is embracing #Kubernetes?

A conventional bank running its real business on such a young technology?

No way, 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.

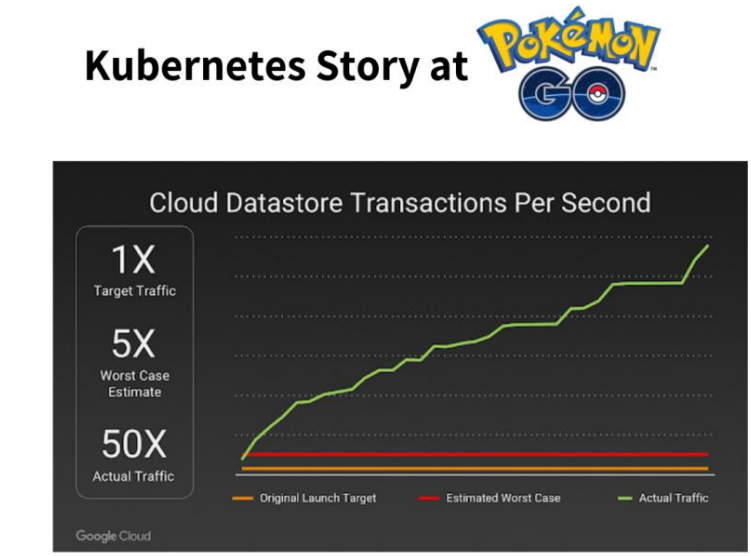
The bank's engineering team came up with an initiative strategy in 2018 to throw away the old way of thinking and started embracing the technologies like microservices, container architecture, and migrate from monolithic to multi-tier applications. It was transforming itself into a software company,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

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

The answer is #Kubernetes. Read the story...



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:
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.

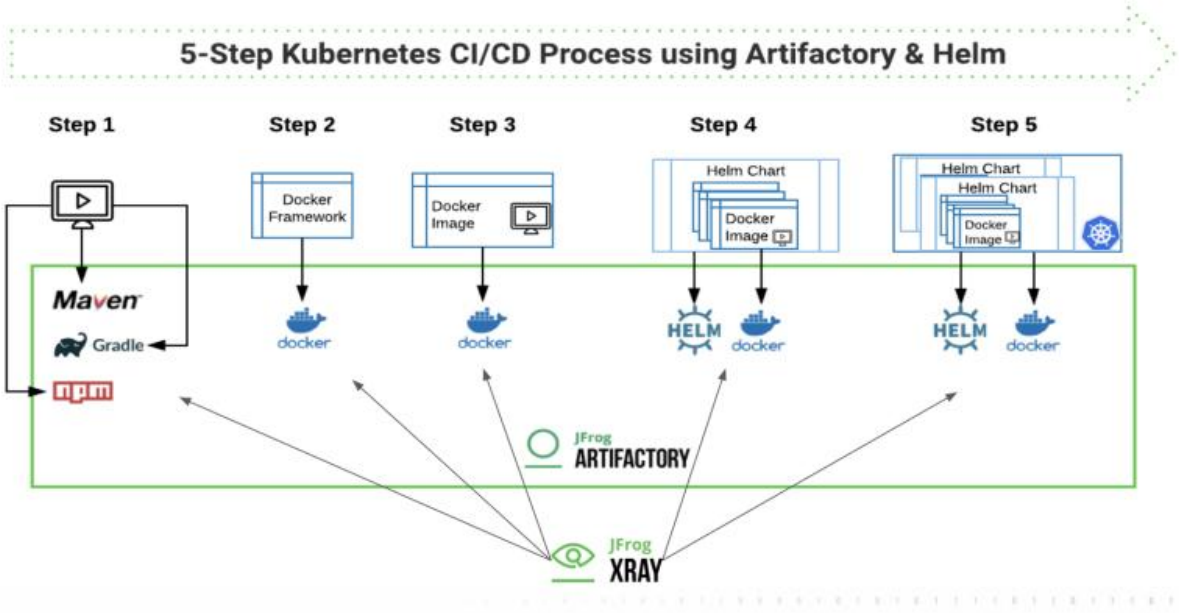
The Solution:
The magic of containers. The application logic for the game ran on Google Container Engine (GKE) powered by the open-source Kubernetes project.

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.

CI/CD with Kubernetes:

How can you quickly achieve CI/CD automation with #Kubernetes and roll it out across your organization?



Step 1: Develop your microservice using dependencies from registries that are proxied in Artifactory. The resulting App package can be a .war or .jar file.

Step 2: Create a Docker Framework using Tomcat and Java-8 on Ubuntu as a base image. Push this image to a Docker registry in Artifactory, where it is also scanned by Xray to assure security and license compliance.

Step 3: Create the Docker image for the microservice by adding the .war/.jar file to the Docker Framework, and push the image to a Docker registry in Artifactory, where it is scanned by Xray.

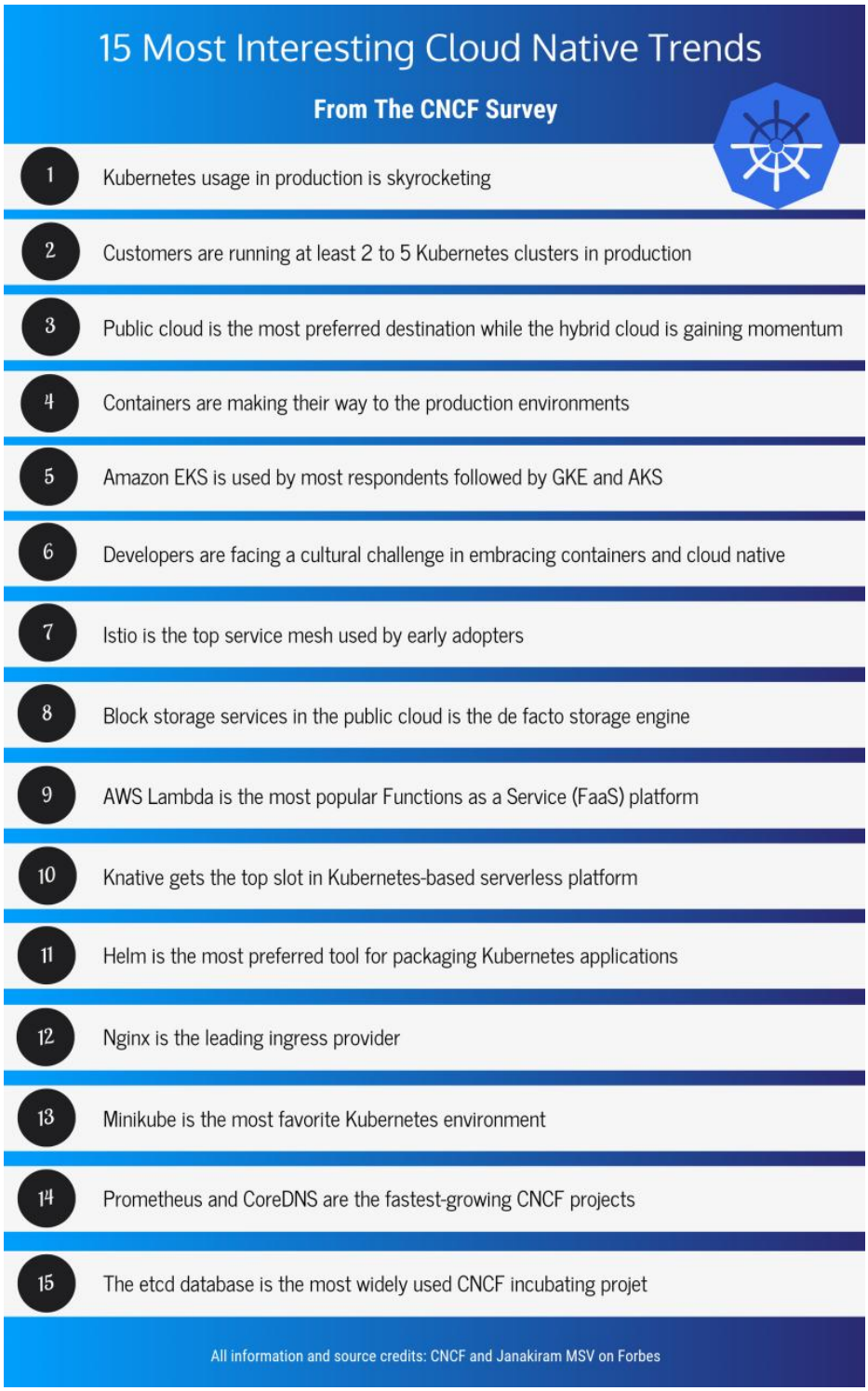
Step 4: Create a Helm chart for the microservice, and push it to a Helm repository in Artifactory.

Step 5: Deploy the microservice from the secure Docker registry to the Kubernetes cluster using the Helm Chart.
See the in-depth article: https://lnkd.in/e4Vkc3m

Survey and findings:

#Kubernetes usage in production is skyrocketing 🚀
What else?

Here are 15 interesting takeaways from the #CNCF annual survey.

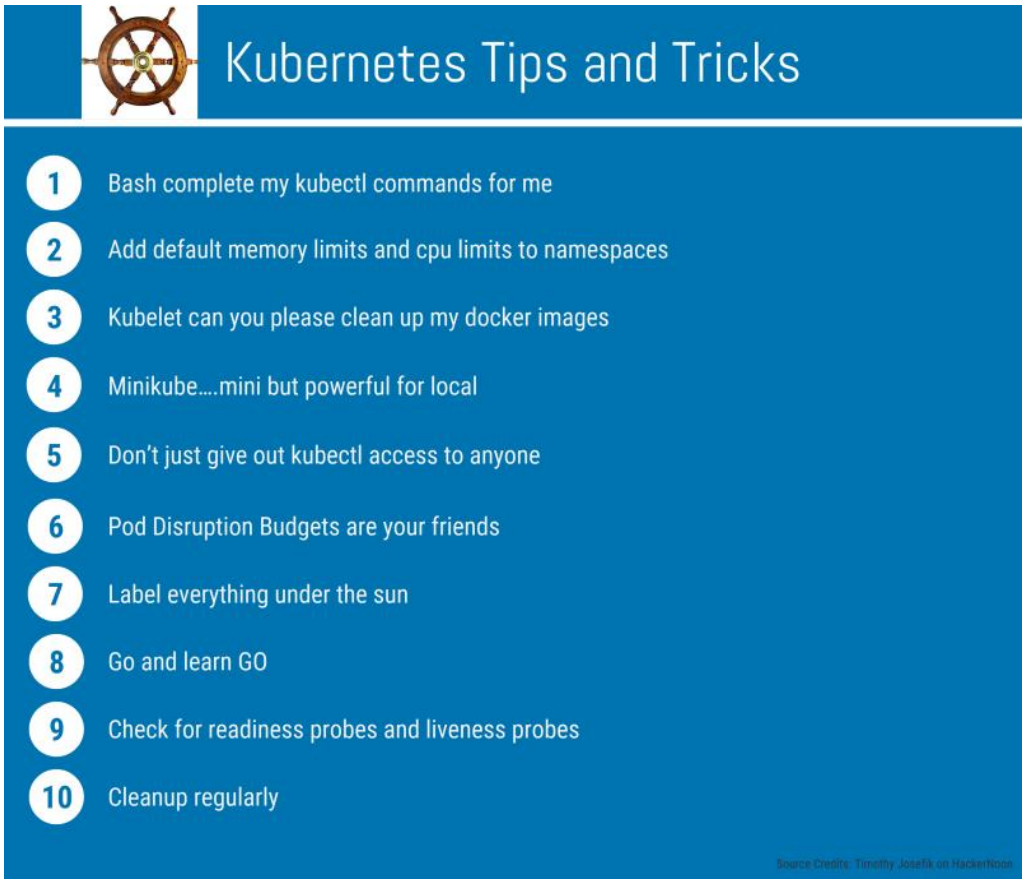


All information and source credit goes to CNCF - https://lnkd.in/eD9fN2R

and Janakiram MSV's article here: https://lnkd.in/eb6GNZS

Tips and tricks:

#Kubernetes is the ultimate avatar of cloud-native development.



Here are some tips and tricks shared by Timothy Josefik on HackerNoon.

The whole article is here: https://lnkd.in/eGdmrkR

#Kubernetes has become a synonym for #CloudNative tech.

More and more companies are trying to use Kubernetes in production, and that's a good move.

Take a look at these 10 Kubernetes production checklist.

---

10 Kubernetes Production Checklist

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

Free resources:

Learn a new skill while working from home!

Sharing some free #Kubernetes resources for everyone.

Image Logo Credits: Kubernetes

> Kubernetes the hard way: https://lnkd.in/eu_rkry

> Introduction to Kubernetes: https://lnkd.in/ebHEyaY

> Learn Kubernetes by Playing the "Game of Pods": https://lnkd.in/epGmpd9

> Kubernetes by example: https://lnkd.in/eETkYKW

> Getting started with Kubernetes: https://lnkd.in/eqD8qWA

> Kubernetes hands-on labs: https://lnkd.in/eEgfyDG

> Learning path - Kubernetes: https://lnkd.in/ea5H-WH

> Fundamentals of Containers, Kubernetes, and Red Hat OpenShift: https://lnkd.in/ea_tfrt

> Zero to hero with Kubernetes: https://lnkd.in/eJJRjck

That is it:)

---

*The most awaited DevOps conference, 'swampUP' is happening virtually - **Save Your Seat**.*

All credits to Kubernetes for helping companies scale and win big time