SANGFOR

To better understand K8S related knowledge and opportunities for beginner in 30 mins

What is a container

- Container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliable from one computing environment to another, in other word, container packages software into standardized unites for development, shipment and deployment.

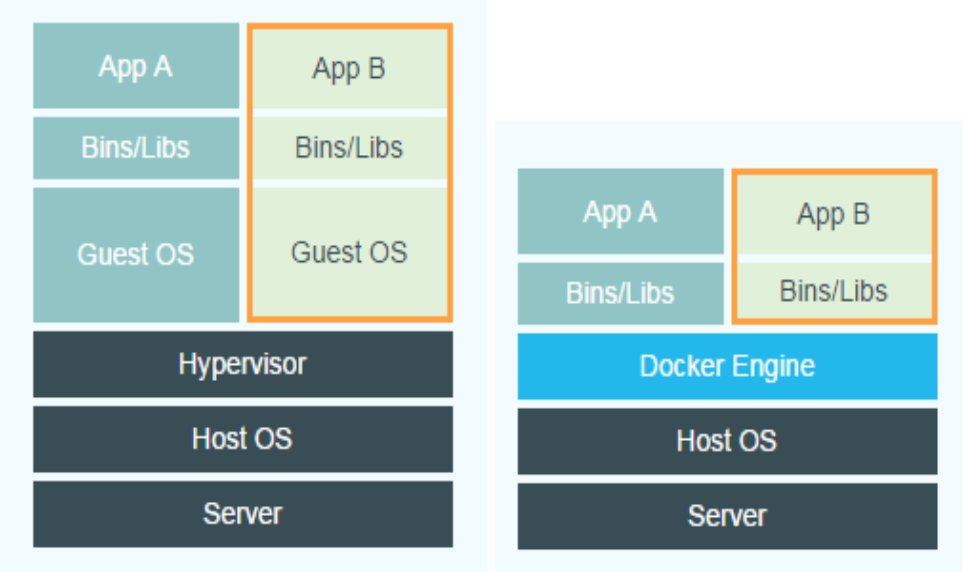
- Docker is a kind of containerized technology that is able to package all code and its dependencies into a separate standard unit, developers are able to directly use it

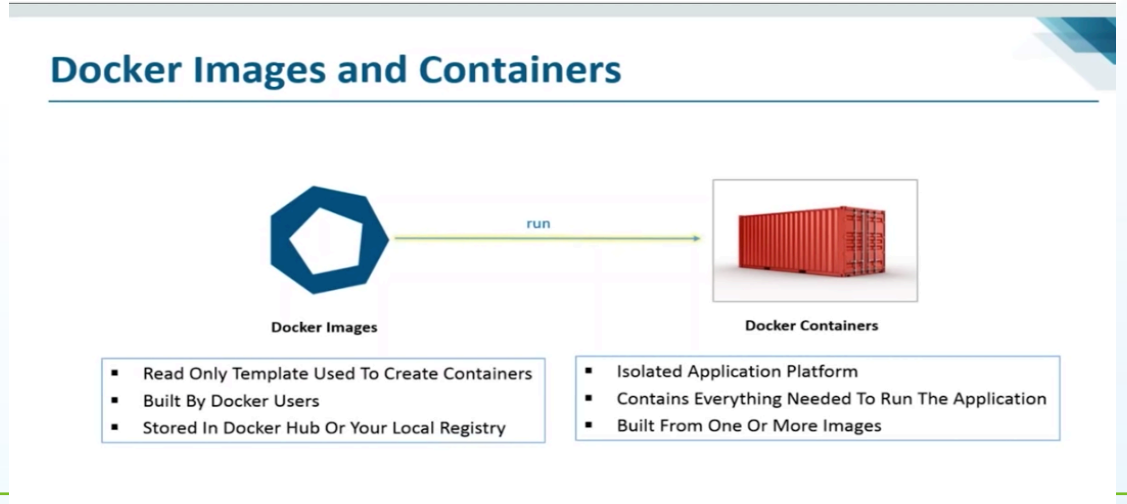
No infrastructure and developing language limitation

What is the difference compared to VM

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient.

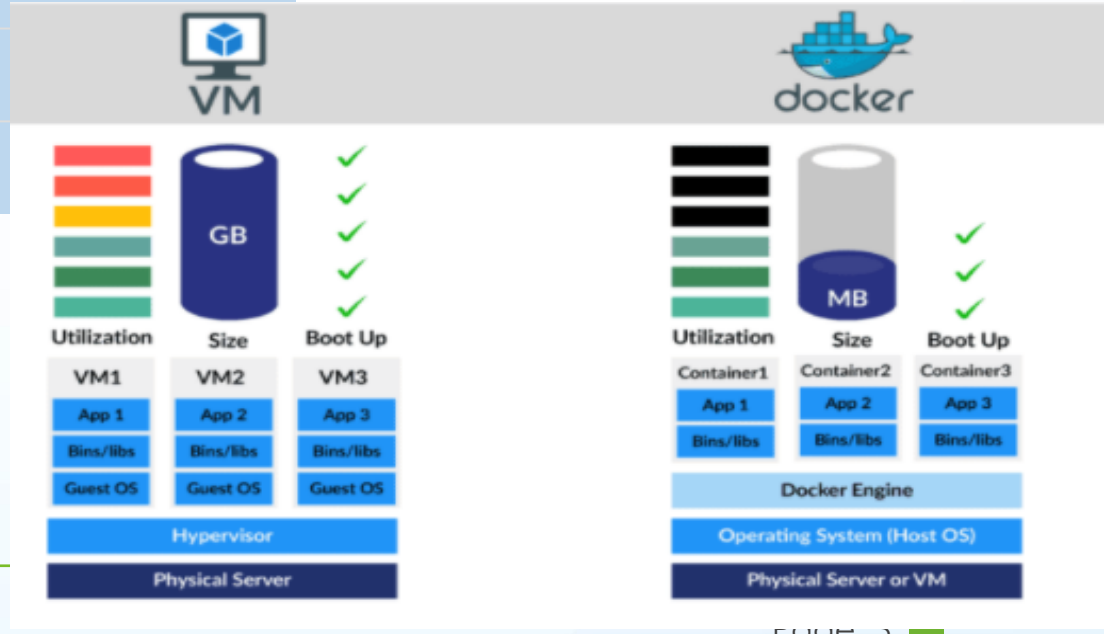
So, you will see the docker engine replaces the hypervisor and then create the containers on top it





Advantages and benefits of docker

	Containers	Virtual Machines(VMs)
Boot-Time	Boots in a few seconds.	It takes a few minutes for VMs to boot.
Runs on	Dockers make use of the execution engine.	VMs make use of the hypervisor.
Memory Efficiency	No space is needed to virtualize, hence less memory (MB)	Requires entire OS to be loaded before starting the surface, so less efficient (GB)
Isolation	Prone to adversities as no provisions for isolation systems.	Interference possibility is minimum because of the efficient isolation mechanism.
Deployment	Deploying is easy as only a single image, containerized can be used across all platforms.	Deployment is comparatively lengthy as separate instances are responsible for execution.
Performance	Thousands of dockers	Limited VMs
Application migration	Across all platforms	Can't support



Real world comparisons

SANGFOR

Physical DC

Dedicated house, garden and foundation

Virtual DC

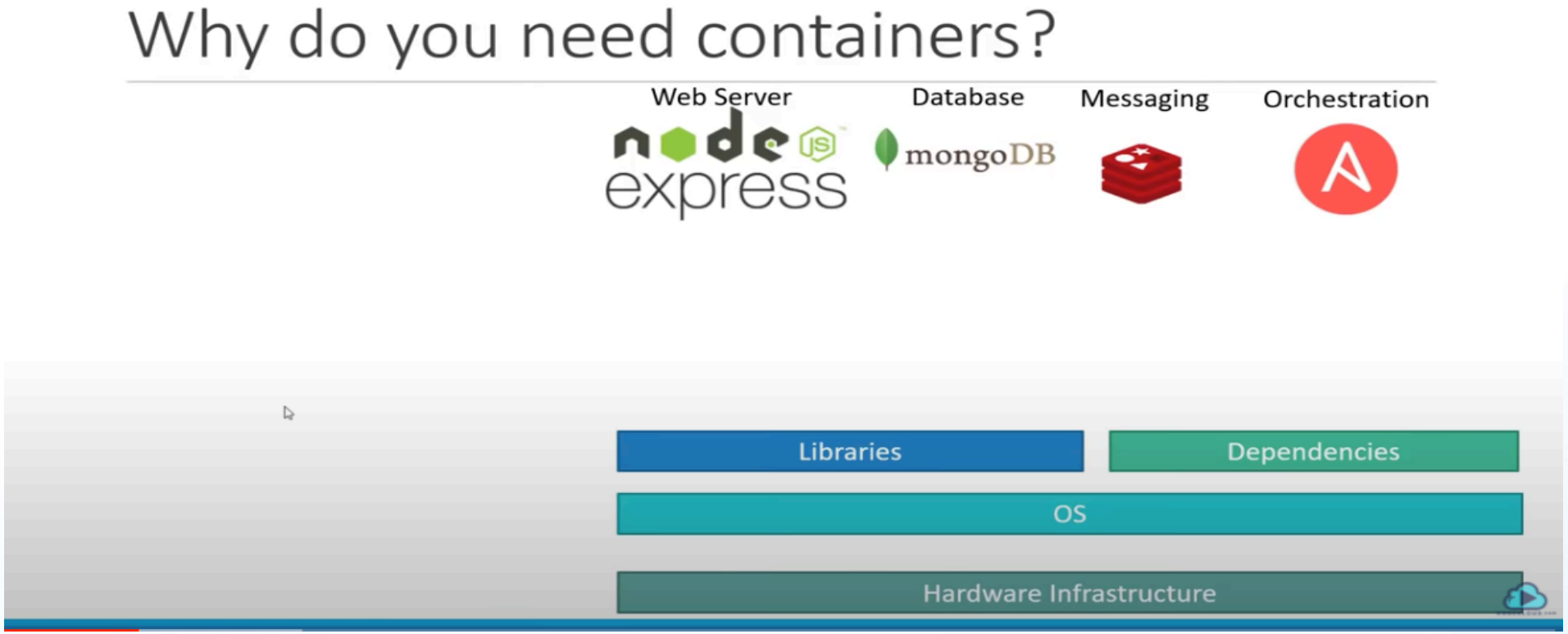
Independent suite, and have may suites in block Dedicated toilet and kitchen, and sharing the block and garden

Container DC

Hotel, capsule house, Sharing most of public foundations, have the same environment

A simple video to help you better understand container

- https://www.youtube.com/watch?v=rmf04yII2K0



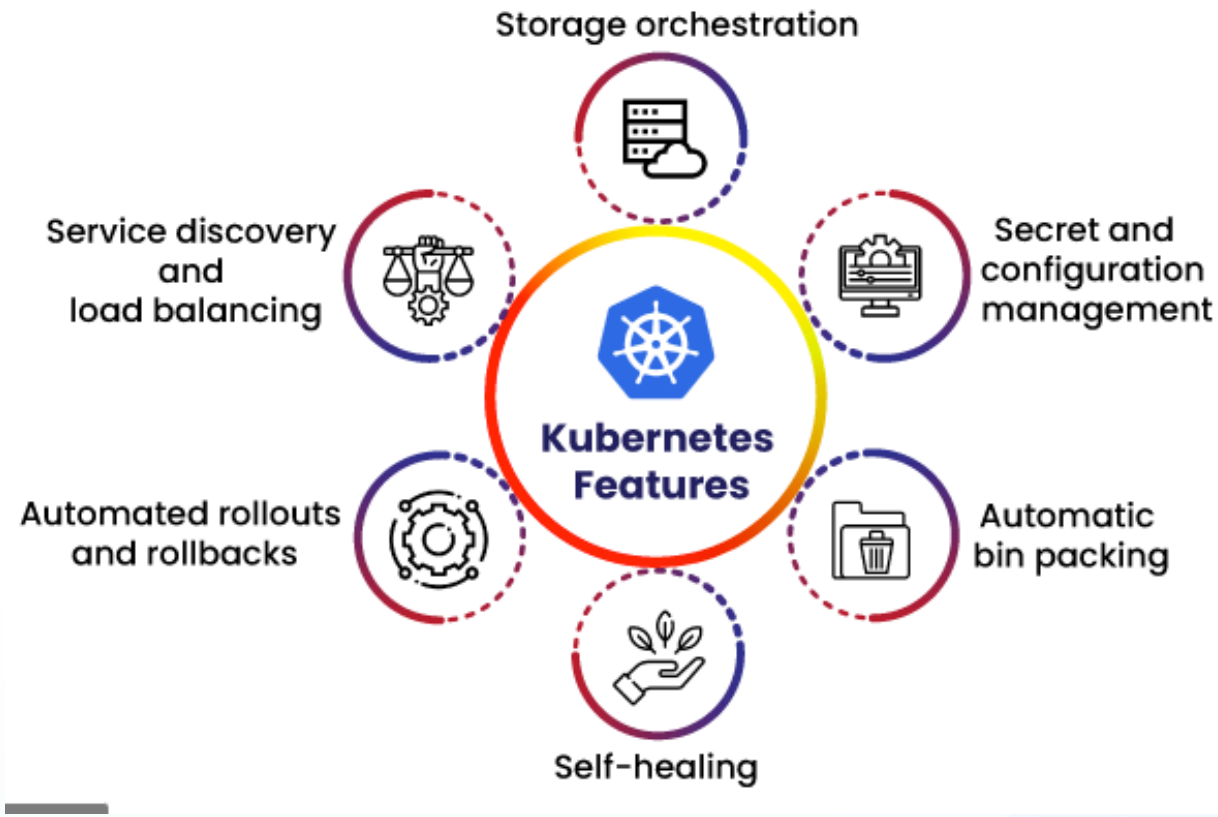
Why need Kubernetes once customer has docker

Let‘s imagine if we have hundreds or thousands of containers or microservices, we want to combine them to run a big application or scale them, even handle the issues when it encounters some unknow problems, as well as container recovery? How can I handle it in this case? Kubernetes(k8s) is the best tool, Kubernetes looks like SCP on HCI to manage and schedule the VMs.

Kubernetes definition

- Kubernetes is an open-source container orchestration platform that enables customer to eliminate many of manual process involved in developing and scaling containerized applications, and also help you easily and efficiently deploy and manage the cluster.

- So it is always used in larger or complicated use case such as microservice management, DevOps, and multi-cloud deployment.



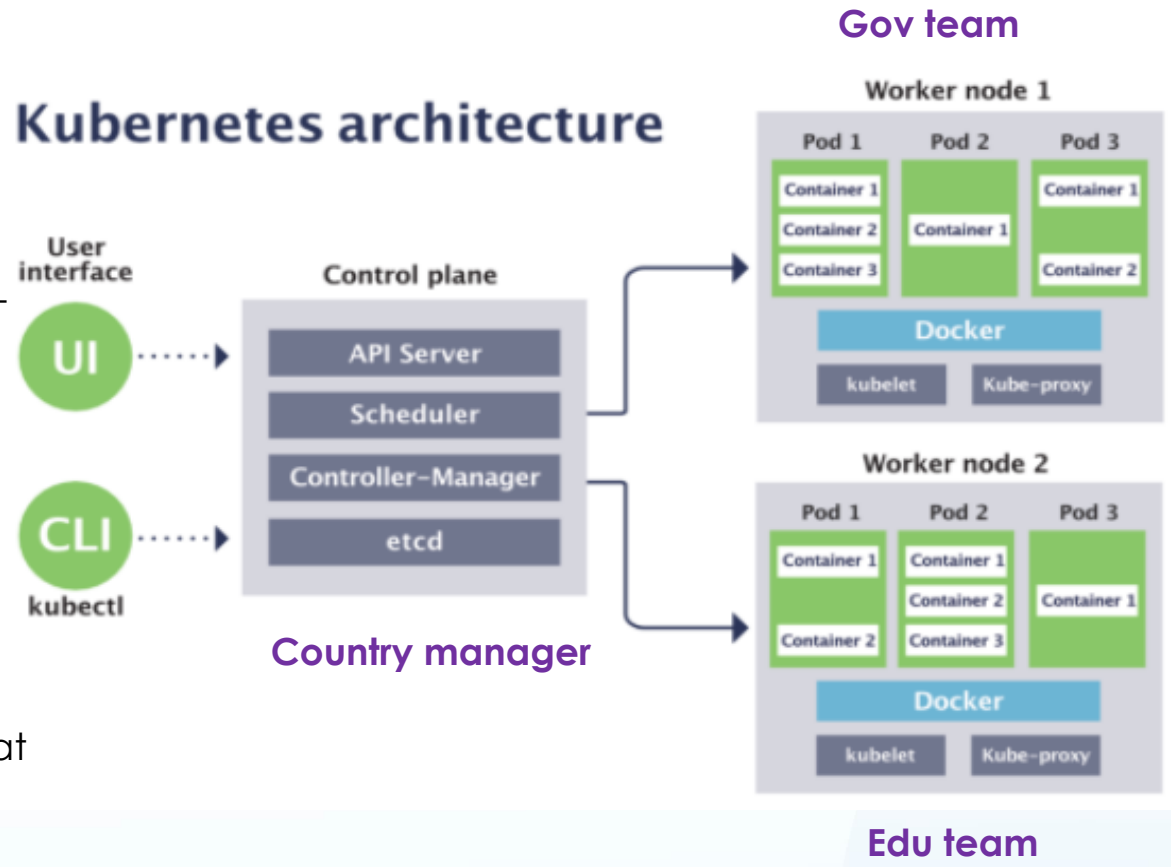
Kubernetes architecture brief introduction

Kubernetes cluster have two parts:

- Control plane(master cluster)—looks like the country manager
- Worker node(Host the Pods that running containers)—looks like a working team

Control Plane role: Detecting and responding to cluster events
Kube-API server: allow different parts and components communicate – looks Like internal communication language
Scheduler: monitoring the cluster healthy and schedule the right resource and node to run pods—Make sales do right thing on right position
Kuber-controller: make sure the cluster is running and available—Looks like you can replace someone once he quits or gets sick
Etcd: store the configuration data and information about the state of clusters ---Looks like GSM to record everyone's daily works

Worker Nodes: A computer or server. **Pods:** smallest and simplest unit that is made of series of containers, Pods always run on a VM or Computer
Kubelet: execute the actions from Master—looks like your team leader
Kube Proxy: Solve the network communication inside or outside --looks like your assistant



The major benefits of Kubernetes

Improved resource utilization

Shortened software development cycles

Market leader

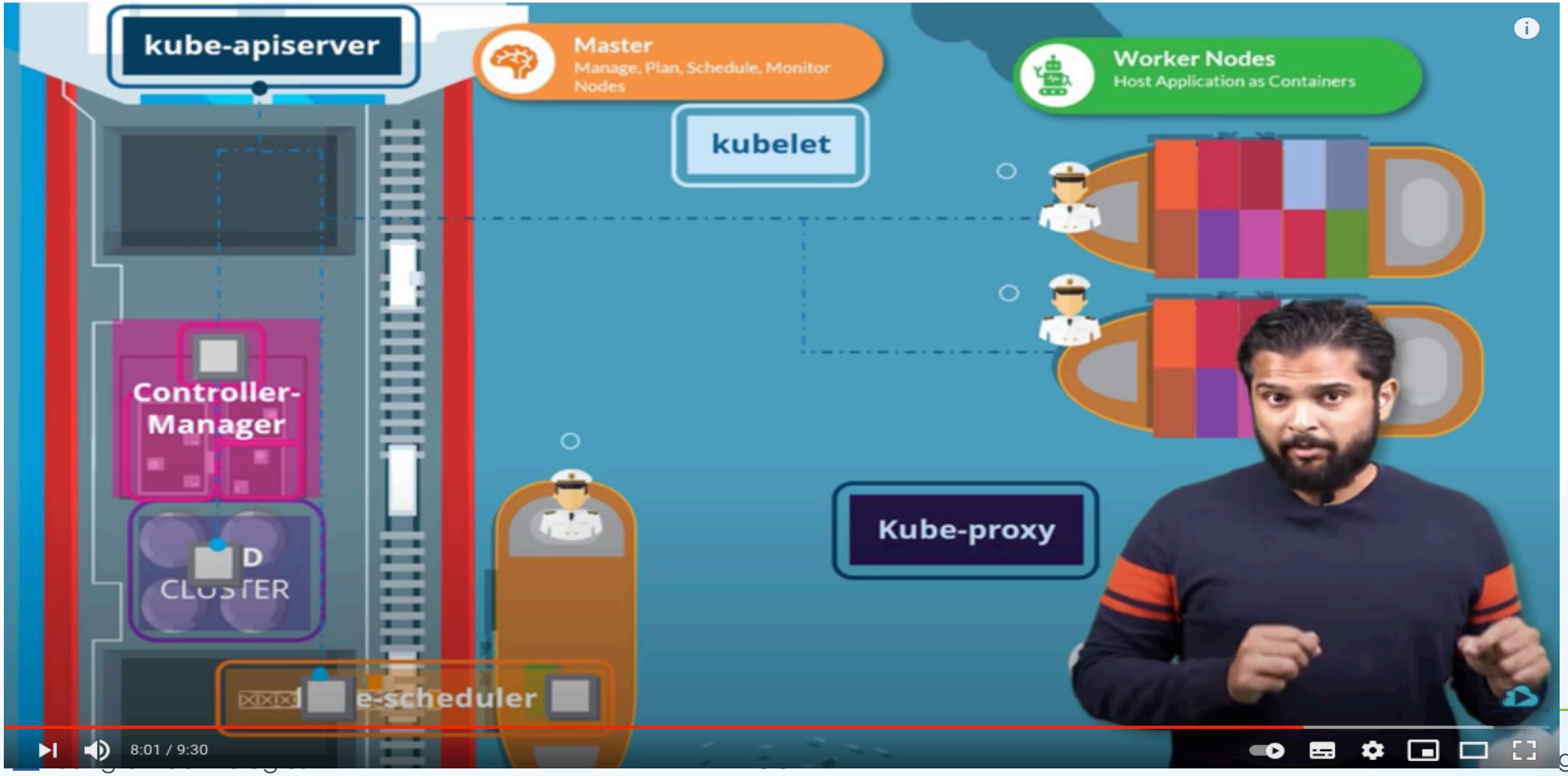
Save money both for cloud infrastructure and manpower

Enable your move to cloud easily and efficiently

Kubernetes architecture simplified

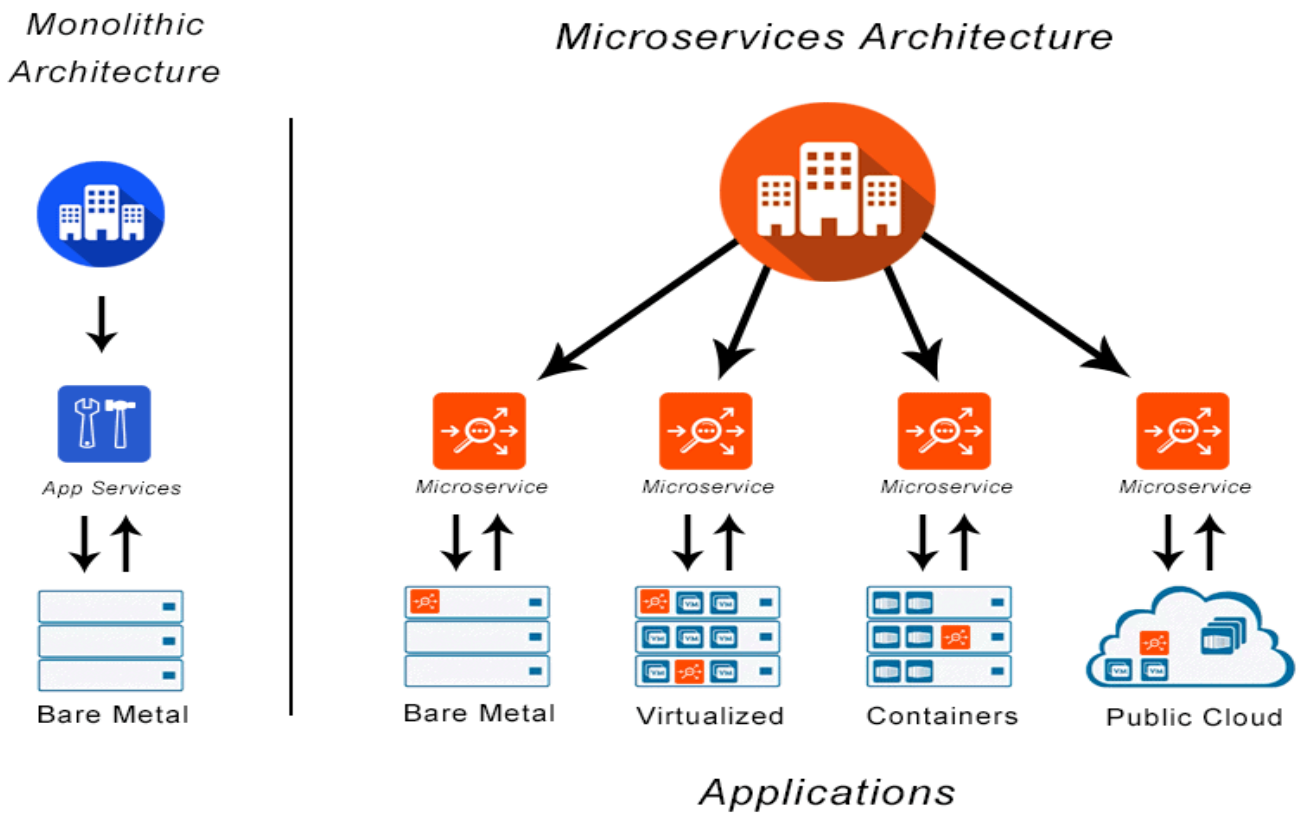
https://www.youtube.com/watch?v=8C_SCDbUJTg

a very interesting video for non-technical guys



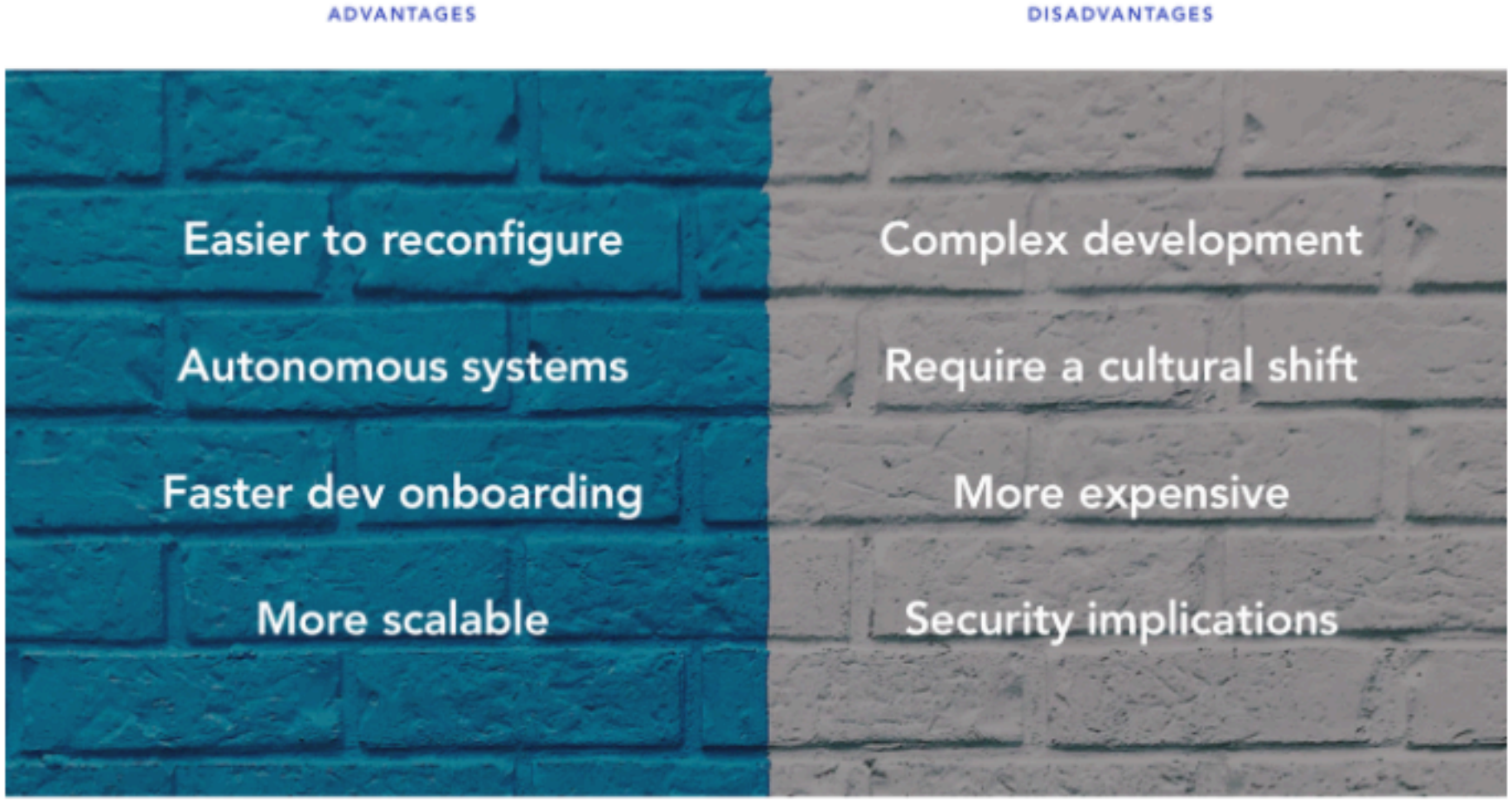
What is microservice

- Firstly, it's an application developing architecture not is an infrastructure architecture
- Monolithic application is like a big container where all the software components of an application are assembled together and tightly packaged
- Microservice application means a big application consists of small autonomous services, and each small service can be independently deployed and scaled, and also can be run in different place
- It looks like you buy Lego Bricks for you children instead of fully assembled toys.



The advantages and disadvantages of Microservice

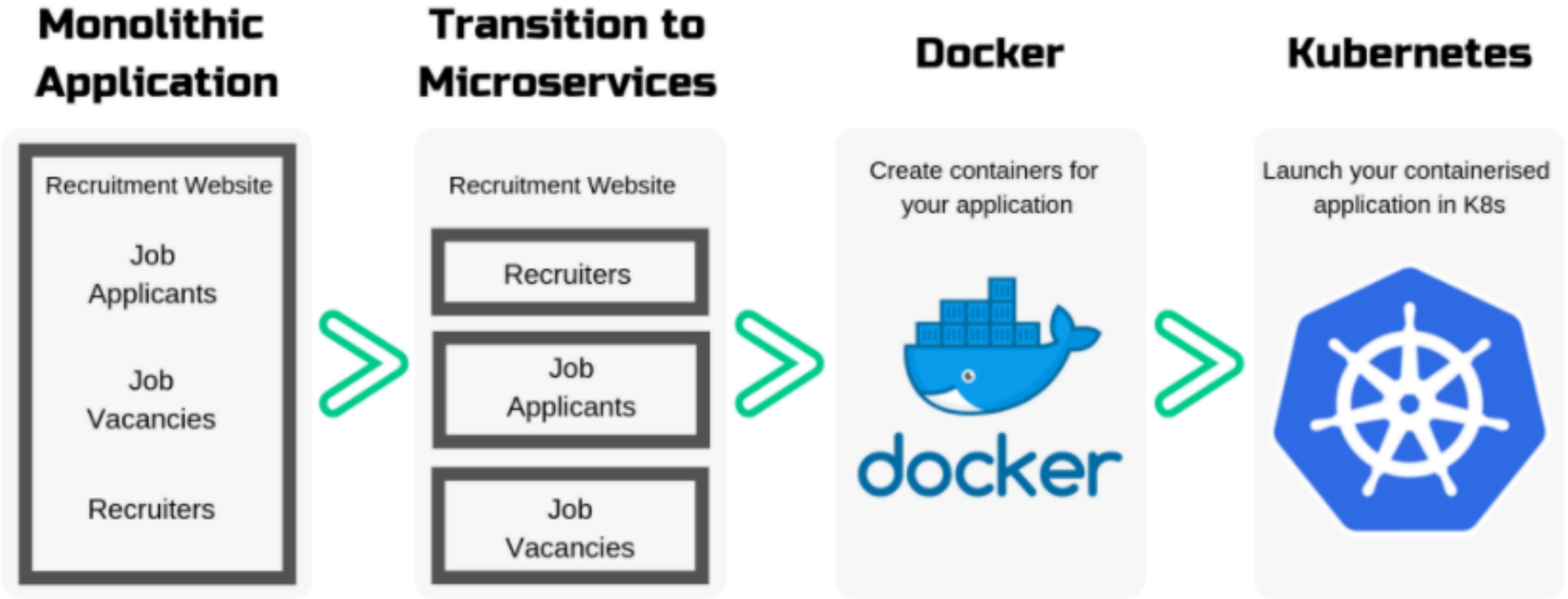
The major benefits of Microservice are allowing enterprises to iterate their application or business faster and more reliable(for developers and testers). One small service going down will not affect the entire application.



However, how to handle the workload of microservice

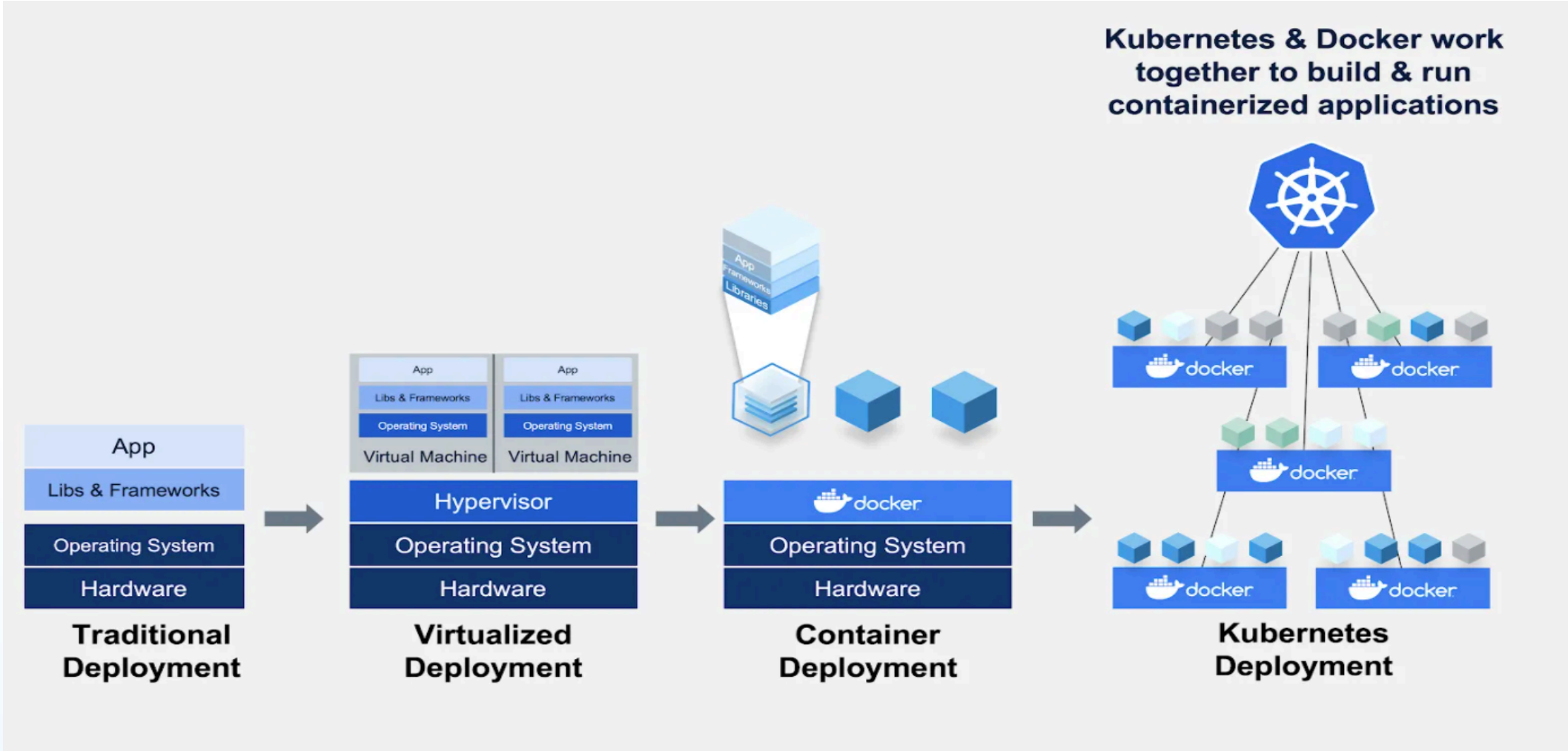
SANGFOR

So, the best way is we can use docker to containerize your microservice components, and then deploy, manage and launch your containerized application with K8s. That will significantly reduce the workload and save your time.



So the final logic is…

Because the architecture of application is shifting from monolithic to microservice, and microservice application means a big application will divide into a lot of small autonomous services or components, the small service can be packaged as container, and then many of containerized services can be deployed and managed in different cloud infrastructure by Kubernetes. Finally, to achieve the application development, testing and deployment as well as scaling quickly and efficiently….



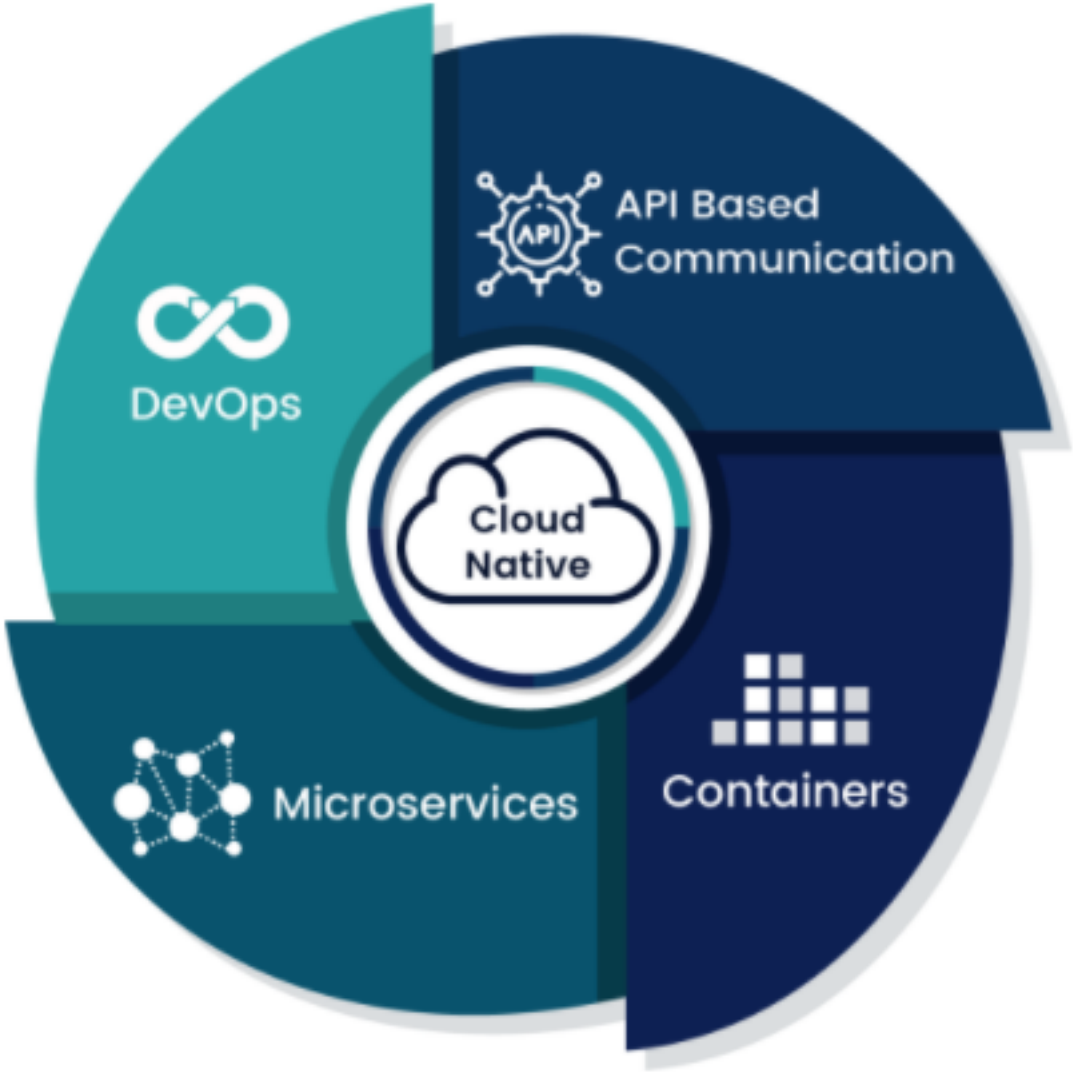
Relationship between Microservice and K8s

- https://www.youtube.com/watch?v=j3XufmvEMiM

What exactly is Cloud-Native

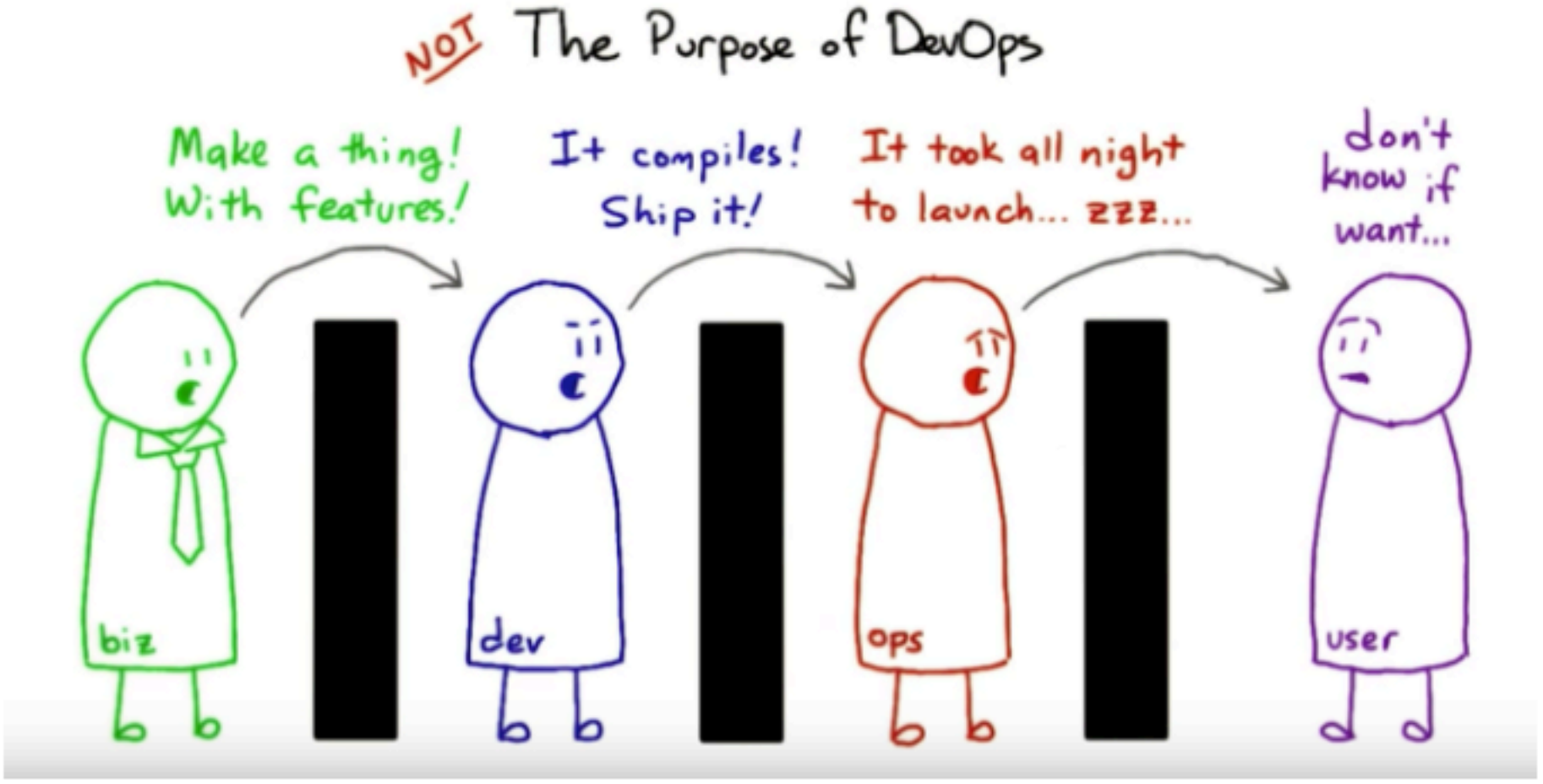
If an application has four following characteristics, we can call it cloud native app

https://www.youtube.com/watch?v=9Ik96SBalvs



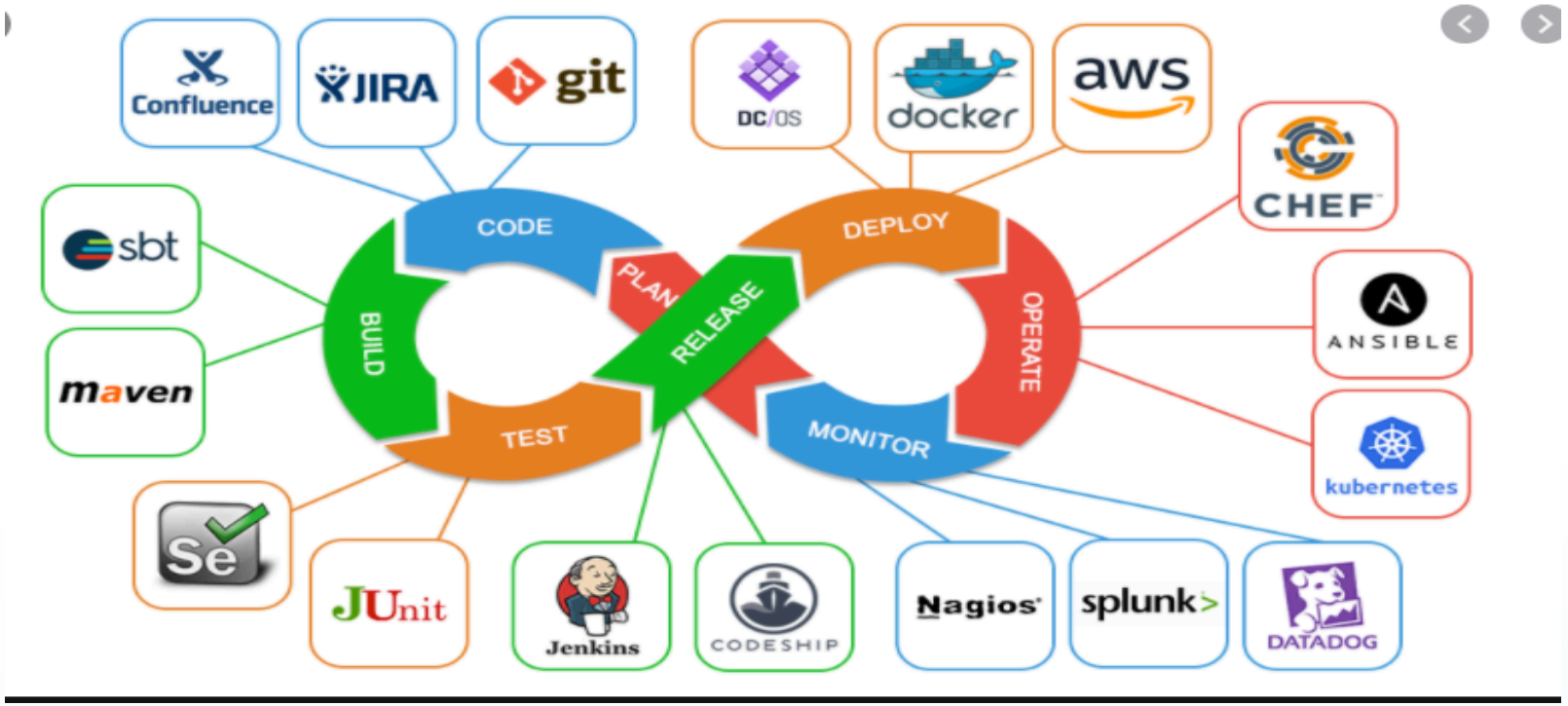
DevOps?

Firstly, DevOps is not a specific technology, it's the combination of cultural philosophies, and tools that increase an organization ability to deliver service and applications faster and better, and also reduce the effort spent on the internal collaboration.



DevOps related process and tools

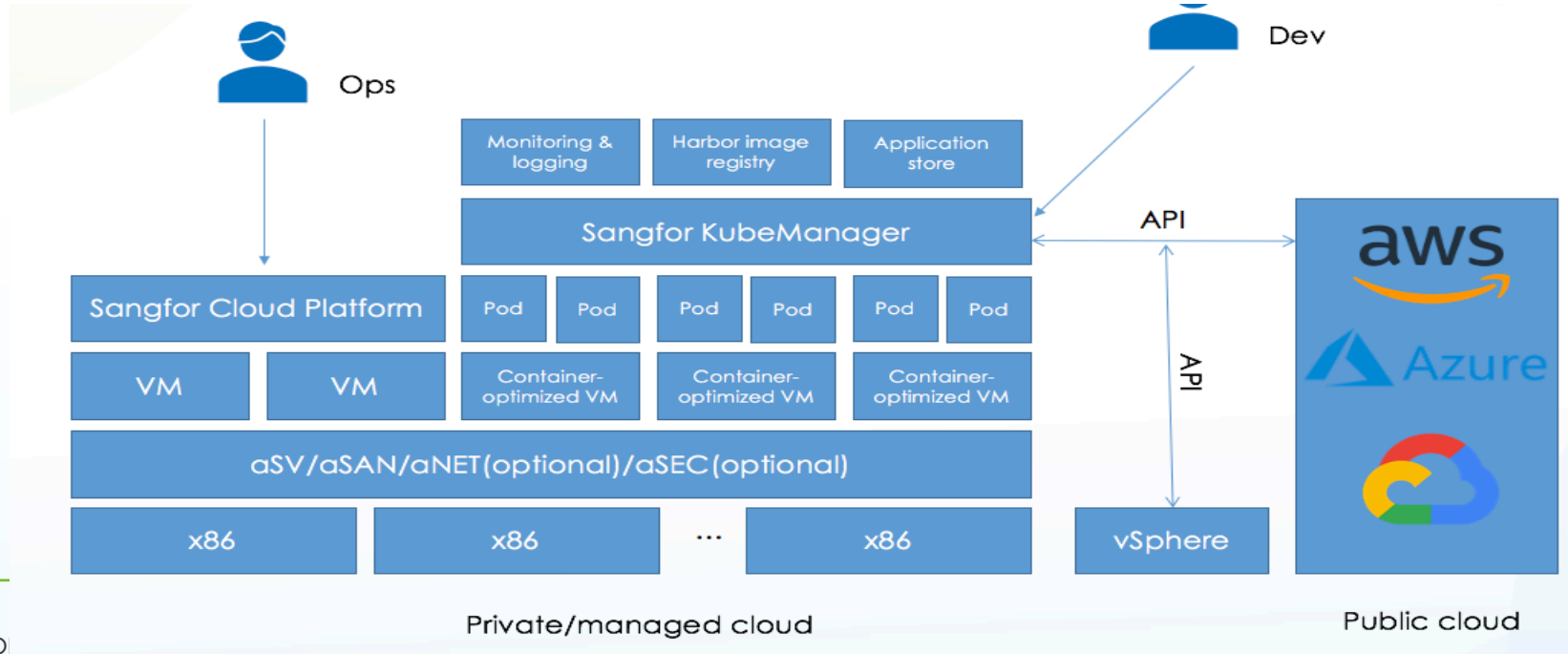
Companies need to make a strategic decision to execute it. Actually Sangfor R&D team is shiftting to DevOps practice in 2021



https://www.youtube.com/watch?v=_l94-tJlovg

Is our k8s solution capability ready so far?

- Yes,HCI-based Kubernetes solution is ready, and we already got PO and project POC successfully
- **Recommended customer persona:**
  - Have dedicated developing and testing team in FSI, Healthcare or high education section
  - local ISV and touch their application or developing team leader
  - Customer desired a unified infrastructure to run VM based application and Container based apps at the same time
  - Desire to get a cost-effective and simple solution
  - Key role is Developing team leader or Application team leader
- However, the following use cases are not recommended: DevOps, Microservice governance.



PaaS capability Matrix

	Sangfor PaaS (based on K8s)	**DevOps**	**Service Mesh**
Container scheduling	★		
Auto-scaling of Pods	★		
Container storage interface	★		
Monitoring & alerting	★		
Image registry	★		
App store	★		
Multi-cluster management	★		
Logging	★		
CI/CD		★	
Jenkins integration		★	
Code repository		★	

PaaS capability Matrix

	Sangfor PaaS (based on K8s)	DevOps	Service Mesh
Version control		★	
Automated test		★	
Pipeline		★	
Artifact management		★	
Circuit-breaker			★
Application performance monitoring			★
Service discovery			★
Load balancing			★
Encryption			★
Authentication & authorization			★

With K8S does not mean that customers have the full digital capabilities

Container and k8s are only part of the capabilities of the digital infrastructure

Customer need a unified digital platform in the future

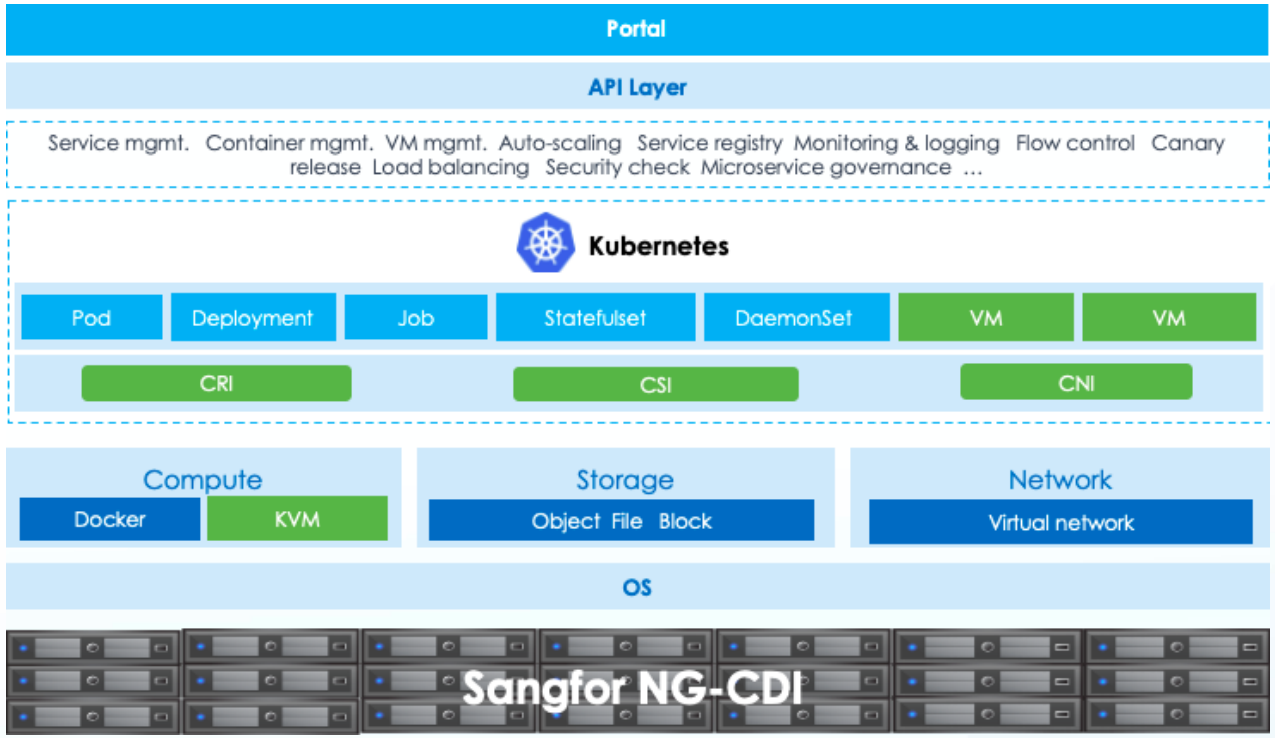
IT team have to consider how to building a **unified platform** across multi cloud that integrated and use different underlying technologies to service the future digital applications

Business Scenarios	 Self-Service terminal	 Internet Medical	 Visiting Wards	 Tailored Medical Service	 Remote Consultation
Cloud Engine Features	Built-In Backup Functionality	Full Lifecycle Security Ability	Data Protection	Big Data & AI	VM & Container
Infrastructure Platform Capability	Cloud Native	Distributed Architecture	Large Volume Storage	Hybrid Cloud	GPU/AI

Sangfor cloud new concept: NG-CDI

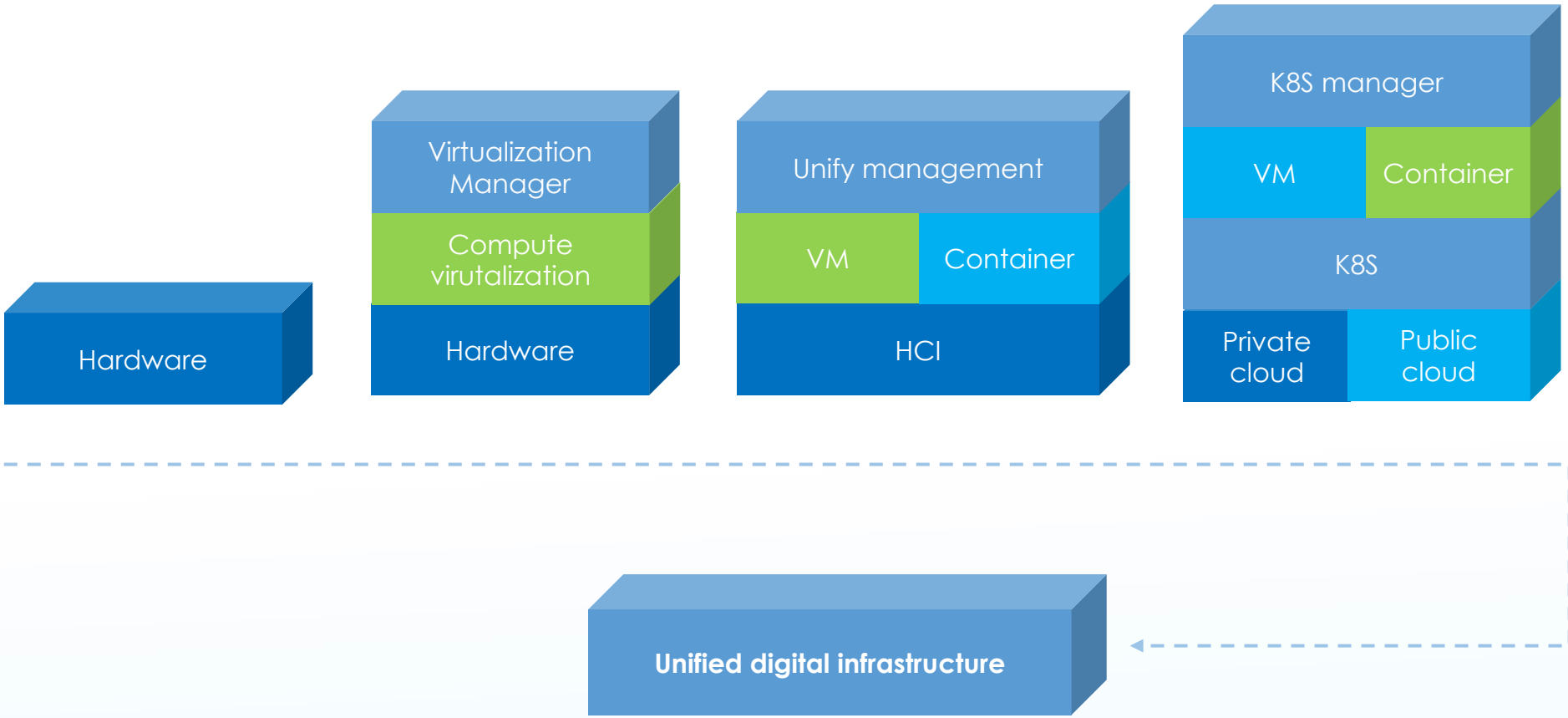
NG-CDI: next generation converged digital infrastructure

- Running and protecting both for traditional apps and cloud-native apps
- The capabilities of NG-CDI are designed not only for cloud native apps, but also include running AI, BIG data, GPU,VDI, SAP HANA, ect…
- Unified monitoring and management for the entire infrastructure(VM and container)
- Compared to the current HCI-based architecture, it's pure cloud-native, that's why it works with a variety of 3rd-party storage services and public cloud services through API.
- Will be released by the end of this year



We thought the future of data center is NG-CDI

Sangfor offers a smooth evolution path enable customer go there



What is benefit of NG-CDI to you

- How to better support and service the digital applications is a one of the most important factor to most CIOs who identify and evaluate a good solution – ensure we deliver the matched story
- A very good topic or story to open the door that allows you to better communicate with CIOs and give them different value points and recommendations to their IT capabilities –High-value information input to your customers
- Allow you to design and propose a smooth cloud path to your customer infrastructure, even now their current infrastructure is pretty traditional (physical or 3-tier architecture)— offering different options to them
- Talk less about our product and talk more the planning. At the same time let you know better to understand customer's true idea and requirements from C level -–Learn more from your customer.
- Better to influence the key roles(CIO) in larger sized customer who already has full virtualization platform

Build a stable and secure digital foundation for a digital world