

Harness the power of the top container orchestration platform.

Kubernetes gives developers the cloud-native
capabilities they need.



White Paper by



Understanding Kubernetes.

Virtual machines (VMs) eased the burden for developers by providing the ability to quickly spin up instances where software can be tested and developed without having to rely on an expensive bare metal machine. Containers further radicalized this process by having better portability between platforms and the cloud and being able to be run within VMs.

Kubernetes has made it even easier by creating an orchestration environment for containers. Kubernetes helps you adopt a cloud-native approach to development and microservices-oriented architecture, decreasing the burden further still in developing and deploying services and applications.

Kubernetes is quickly becoming the most popular container orchestration platform in the world. In a recent Cloud Native Computing Foundation survey, over 78% of respondents selected Kubernetes as their orchestration platform. While Kubernetes is not the only orchestration platform on the market, it has some key benefits that separate it from its competitors.¹

But before we can jump into why Kubernetes is such a powerful tool and how it supports developers, we need to first understand the components within the framework Kubernetes works with containers.

Kubernetes works by utilizing microservices and serverless architecture to form a cloud-native approach to development.



¹ <https://www.cncf.io/blog/2020/03/04/2019-cncf-survey-results-are-here-deployments-are-growing-in-size-and-speed-as-cloud-native-adoption-becomes-mainstream/>

What are containers and containerization?

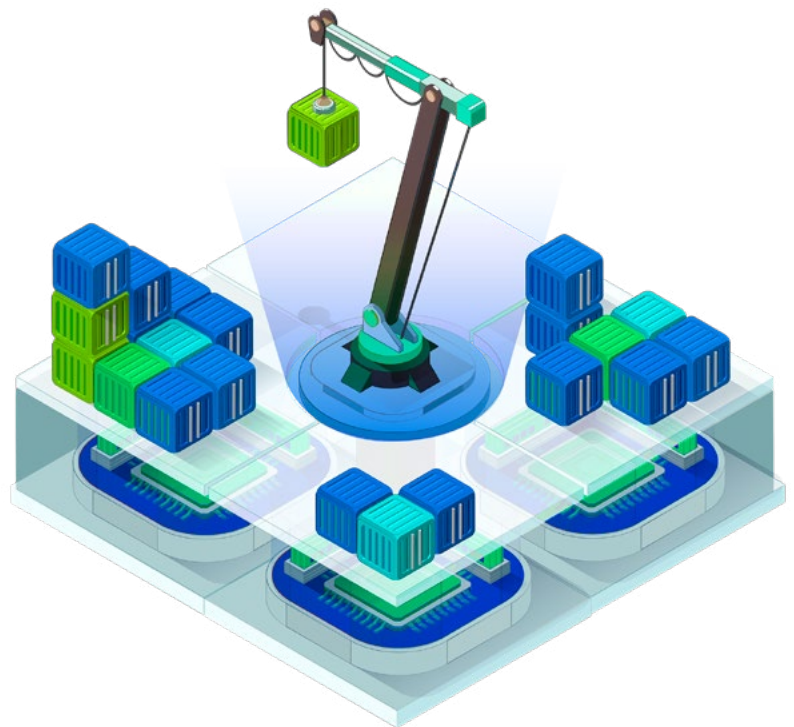
Containerization is a computing practice in which applications run in isolated spaces, known as containers, all on the same OS. This is a fundamental difference from virtual machines, where applications can run on multiple operating systems.

Containerization has become popular because of the many benefits it offers, such as:

- ▶ Portability between different platforms and clouds. It is the write once, run anywhere approach to running applications.
- ▶ Efficiency that results from using fewer resources than virtual machines.
- ▶ More flexibility while integrating with the existing DevOps environment.
- ▶ Quicker delivery enhancements.
- ▶ The ability to work on virtualized infrastructures or bare metal servers.
- ▶ Faster app start-up and easier scaling.
- ▶ Easier management due to built-in install, upgrade and rollback processes within the Kubernetes platform.

Companies can implement containerization to promote more efficient resource management and encourage the reduction of their costs. Furthermore, Kubernetes works by automating, deploying, scaling and managing the container applications by using microservices.

Companies can implement containerization to promote more efficient resource management and encourage the reduction of their costs.



Why is Kubernetes important for microservice architecture?

One of Kubernetes' primary benefits is its ability to increase a company's infrastructure utilization with efficient computing resource sharing across multiple processes. In other words, it has the ability to strategically deploy computing resources to fit the demands of your organization.

When companies migrate from monolithic architectures to microservices, they may face certain challenges, including:

- ▶ Predicting the computing resources each service needs.
- ▶ Understanding and adapting to how each service changes under load.
- ▶ Learning how to carve out infrastructure partitions and divide resources between microservices.
- ▶ Enforcing resource restrictions.

Kubernetes alleviates the challenges of microservice architecture with its common framework that describes, inspects and

allocates resource sharing. For this reason, companies considering the switch from monolithic application architecture to a microservice architecture should evaluate the benefits offered by Kubernetes.

Kubernetes solves the challenges of microservice architecture with its common framework that describes, inspects, and allocates resource sharing.



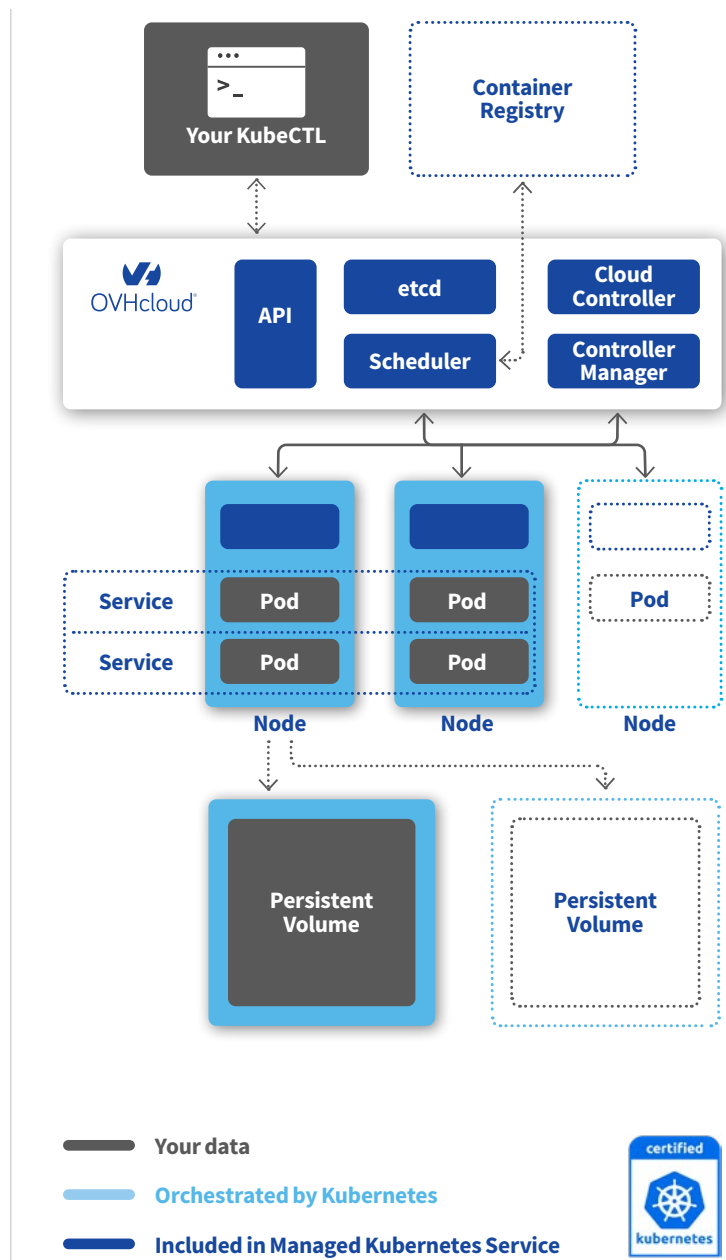
Kubernetes.

Kubernetes on public cloud allows for a cloud-native approach to developing applications and services. By utilizing this advanced approach, developers can focus on building, deploying and maintaining applications. Kubernetes manages containers for applications and services and allows DevOps teams to easily merge code, automate the deployment of services and applications, automate operations, and automate the scaling of their containers.

Because Kubernetes facilitates the use of microservices and serverless architecture, it improves the time to market for software and applications. It streamlines the containerization process and makes developers' jobs easier by:

- ▶ Giving the ability to self-heal containers with pre-defined user health checks and replacing containers that are no longer operational.
- ▶ Fitting containers to nodes with pre-defined CPU and RAM requirements.
- ▶ Managing passwords by storing OAuth tokens, SSH keys and more without having to rebuild container images.
- ▶ Load balancing and distributing network traffic when requests become overloaded.
- ▶ Auto-scaling containers when applications need more RAM or CPU usage.

Managed Kubernetes facilitates both declarative configuration and automation, allowing you to manage your host clusters easily and more efficiently. Because Kubernetes clusters are able to be spawned in public, private or hybrid clouds, they are well-suited for cloud-native hosting, which contains apps that require rapid scaling.



Why is cloud-native important today?

Cloud-native is a method for building and running applications that leverages the cloud computing delivery model. This type of architecture helps businesses take advantage of the public cloud's distributed, flexible and scalable nature. Companies can bring new ideas to market faster while also responding to customer demands quicker and more efficiently.

When businesses opt for cloud-native, they strip away infrastructure layers such as networks, servers and operating systems. With cloud-native technology, all the infrastructure DevOps teams need automatically scales with a quick script.

Write Once, Deploy Infinitely

One of the primary advantages of cloud-native technology is that it can simplify deployment. For example, before cloud-native, if a company had a mobile e-commerce application, they would have had to consider their networking infrastructure and other infrastructure details before focusing on what mattered: the code and how it helped their customer.

With cloud-native, developers may write the code once and deploy it on whatever container they want, simplifying the application deployment.

For example, switching from a monolithic to a microservice architecture using cloud-native technology can result in multiple teams no longer having to coordinate their coding, allowing them to deploy several features at once that don't require complex integration and testing. Additionally, developers may use any language they want for their code.

Deployment Speed

On top of the simple deployment and the coding freedom cloud-native applications offer, they also offer speed. In fact, a Kubernetes case study found that one company's deployment time dropped from four hours to fifteen minutes.² With increased deployment speed also comes decreased repair time. Cloud-native applications can restore your site quickly, keeping your customers engaged.

² <https://kubernetes.io/case-studies/appdirect/>

What are the benefits of Kubernetes to DevOps teams?

1. Developing code in a per-service model means the potential for faster time to market, increased productivity and more accurate code testing.

Because Kubernetes manages microservices by predicting computing resources and adapting service changes under load, DevOps teams may speed up their application development time to market. DevOps teams can worry less about altering application details and may instead focus solely on deployment and management.

Kubernetes also facilitates end-to-end testing for both developers and testers. By working together with the more accurate tools Kubernetes offers, testers may log bugs and developers may access the application deficiency. This results in increased testing capabilities and improved DevOps correspondence.

2. Deploying code in a per-service model can utilize DevOps resources better for less.

Executing deployment on a per-service basis means you may efficiently use resources and save money with hourly public cloud billing.

Additionally, each service has an independent lifecycle when you use Kubernetes. The independent lifecycles mean the DevOps team may independently test and deploy individual applications. The ability to have pre-determined configurations also allows for end-user input.

3. Service Discovery Pattern and Load Balancing

Service discovery is a critical component of Kubernetes' load-balancing capabilities. Kubernetes deploys applications through pods. Different hosts manage these pods, scaling down by killing excess or moving pods from one node to the next and scaling up by increasing pods.

The service discovery pattern promotes the application remaining reachable at all times throughout this process. It's also instrumental in load balancing the process of distributing network traffic to multiple backend services.

Load balancing is one of the critical components of Kubernetes' maximized scalability and availability. Kubernetes has a variety of choices for load balancing external traffic to pods.

4. Auto-Scaling

Kubernetes can respond to end-user requests and ship software extremely fast. But what happens when you make an app that becomes more popular than anticipated? Auto-scaling is Kubernetes' solution to this frequent issue. Kubernetes automatically scales your cluster up or down using their service discovery pattern.

5. Storage Orchestration

Kubernetes' storage orchestration lets you build application services across containers, schedule containers across clusters, scale containers and manage your containers' health over time.

6. Automated Rollouts and Rollbacks

Kubernetes automatically rolls out changes to your application or its configuration. It does this while monitoring the application's health to ensure it doesn't kill your instances. If something goes wrong with the application, Kubernetes is able to roll the change back expediently.

7. Automatic Bin Packing

Kubernetes automatically places containers in the correct place based on their required resources and restrictions. It can do this without impairing availability.

8. Self-Healing

Kubernetes automatically restarts containers that fail, replaces containers, and kills containers that don't respond to the user-defined health checks. It also doesn't advertise these containers to the user until they are ready to serve.

The independent lifecycles mean the DevOps team may independently test and deploy individual applications.

9. Secret Management

Kubernetes allows you to store and manage sensitive information such as passwords, OAuth tokens and ssh keys. Doing so in secret is safer and more flexible than using a pod definition or a container image.

10. Environmental Consistency

Managed Kubernetes lets you build a standard image for every place you deploy your container. Doing so can give you a consistent environment, so it looks and functions the same whether you use Windows, Mac or a test server.

11. Cloud and OS Distribution Portability

Kubernetes runs on Ubuntu, RHEL, CoreOS, on-premises, major public clouds and more.

12. Resource Isolation and Utilization

Kubernetes' resource isolation can offer you predictable application performance while its resource utilization can offer high efficiency and density.

Why choose Kubernetes with OVHcloud?



Fully Managed by OVHcloud®

We keep our operations in-house. OVHcloud deploys, hosts and maintains all the necessary Kubernetes components. We maintain the core components on your nodes and provide updates with security patches. Simply download your Kubernetes' config file and access all the features of a certified Kubernetes service.



Scalability and High Availability

We can help your business grow with Kubernetes container deployment. Kubernetes enables you to expose a service across multiple worker nodes, launch containers and configure the load balancer, and add new computing nodes. Adapt and grow by redefining the health conditions for any service and Kubernetes will automatically relaunch any containers that don't meet the updated criteria.



Health Data Hosting Compliance

Rest easy knowing your healthcare data is secure with OVHcloud. Our certified cloud infrastructures and services confirm that our information security management system (ISMS) will manage risks and vulnerabilities and encourage business continuity. Our infrastructures and services are ISO/IEC 27001, 27017, 27018 and 27701 certified.



Competitive Pricing

We know your bottom line is essential, which is why we offer competitive pricing on all of our products and services. Our Kubernetes offering has no fees on clusters, and our Public Cloud Services are billable by the month or hour to save you money depending on your usage.

OVHcloud US is a subsidiary of OVHcloud, a global player and Europe's leading cloud provider operating more than 400,000 servers within 43 data centers across four continents. For over 20 years, the company has relied on an integrated model that provides complete control of its value chain, from the design of its servers to the construction and management of its data centers, including the orchestration of its fiber-optic network. This unique approach allows it to independently cover all the uses of its 1.6 million customers in more than 140 countries. OVHcloud now offers latest generation solutions combining performance, price predictability, and total sovereignty over their data to support their growth in complete freedom.



us.sales@us.ovhcloud.com



x.com/OVHcloud_US



us.ovhcloud.com

