IKO
InterSystems Kubernetes Operator

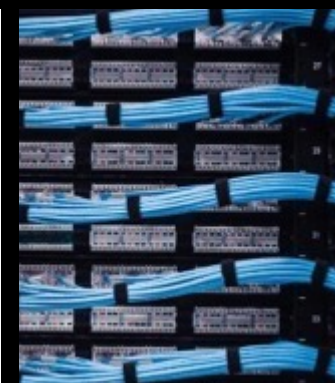**InterSystems®**
Creative data technology

Sylvain Guilbaud

Sales Engineer, InterSystems France

kubernetes

Orchestrateur
de container

Why would I use Kubernetes ?

Kubernetes is an open-source orchestration engine for **automating deployment**, **scaling**, and **management** of containerized workloads and services, and excels at orchestrating complex SaaS (software as a service) applications.

You provision a Kubernetes-enabled cluster and tell Kubernetes the containerized services you want to deploy on it and the policies you want them to be governed by; Kubernetes **transparently provides the needed resources** in the most efficient way possible, **repairs** or **restores** the configuration when problems with those resources cause it to deviate from what you specified, and can **scale automatically** or **on demand**.

In the simplest terms, Kubernetes **deploys** a multicontainer application in the configuration and at the scale you specify on any Kubernetes-enabled platform, and **keeps** the application **operating** exactly as you described it.

# Why do I need the InterSystems Kubernetes Operator?

In Kubernetes, a resource is an endpoint that stores a collection of API objects of a certain kind, from which an instance of the resource can be created or deployed as an object on the cluster. For example, built-in resources include, among many others, *pod* (a set of running containers), *service* (a network service representing an application running on a set of pods), and *persistentvolume* (a directory containing persistent data, accessible to the containers in a pod).

The **InterSystems Kubernetes Operator** (IKO) extends the resources built into the Kubernetes API with a [custom resource](#) called *IrisCluster*, representing an **InterSystems IRIS cluster**. An instance of this resource — that is, a sharded cluster, or a standalone InterSystems IRIS instance, optionally configured with application servers in a distributed cache cluster — can be deployed on any Kubernetes platform on which the IKO is installed and benefit from all the features of Kubernetes such as its services, role-based access control (RBAC), and so on.

The **IrisCluster** resource isn't required to deploy InterSystems IRIS under Kubernetes. But because Kubernetes is application-independent, you would need to create custom definitions and scripts to handle all the needed configuration of the InterSystems IRIS instances or other components in the deployed containers, along with networking, persistent storage requirements, and so on.

Installing the **IKO** **automates** these tasks. By putting together a few settings that define the cluster, for example the number of data and compute nodes, whether they should be mirrored, and where the Docker credentials needed to pull the container images are stored, you can **easily deploy** your InterSystems IRIS cluster exactly as you want it. The operator also adds **InterSystems IRIS-specific cluster management capabilities** to Kubernetes, enabling tasks like adding data or compute nodes, which you would otherwise have to do manually by interacting directly with the instances.

Petit retour en arrière

Traditional Deployment	**Virtualized Deployment**	**Container Deployment**
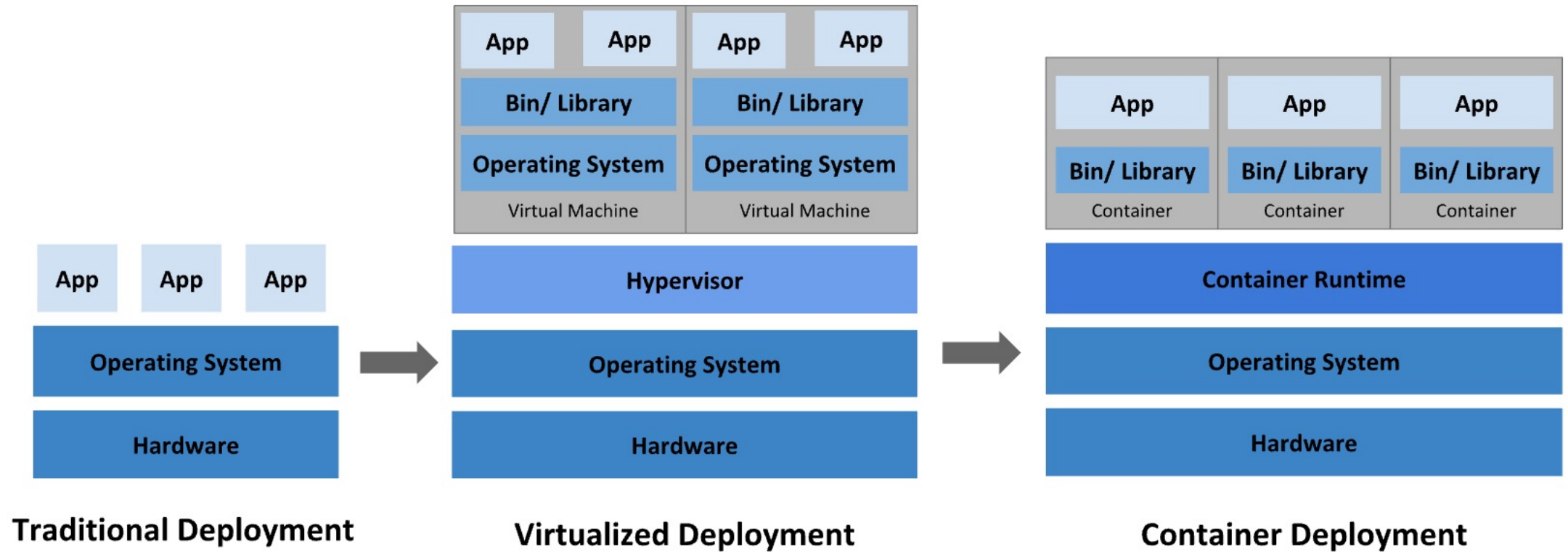
Le rôle de Kubernetes

Service discovery and load balancing
Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.

Storage orchestration
Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.

Automated rollouts and rollbacks
You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.

Automatic bin packing
You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
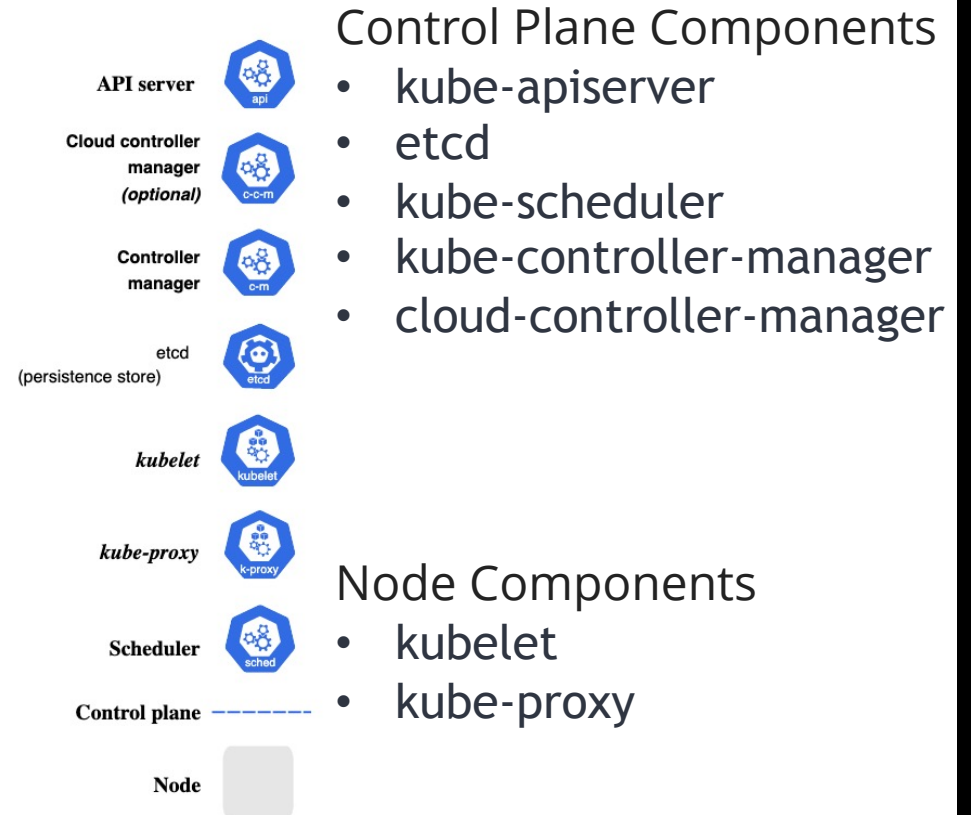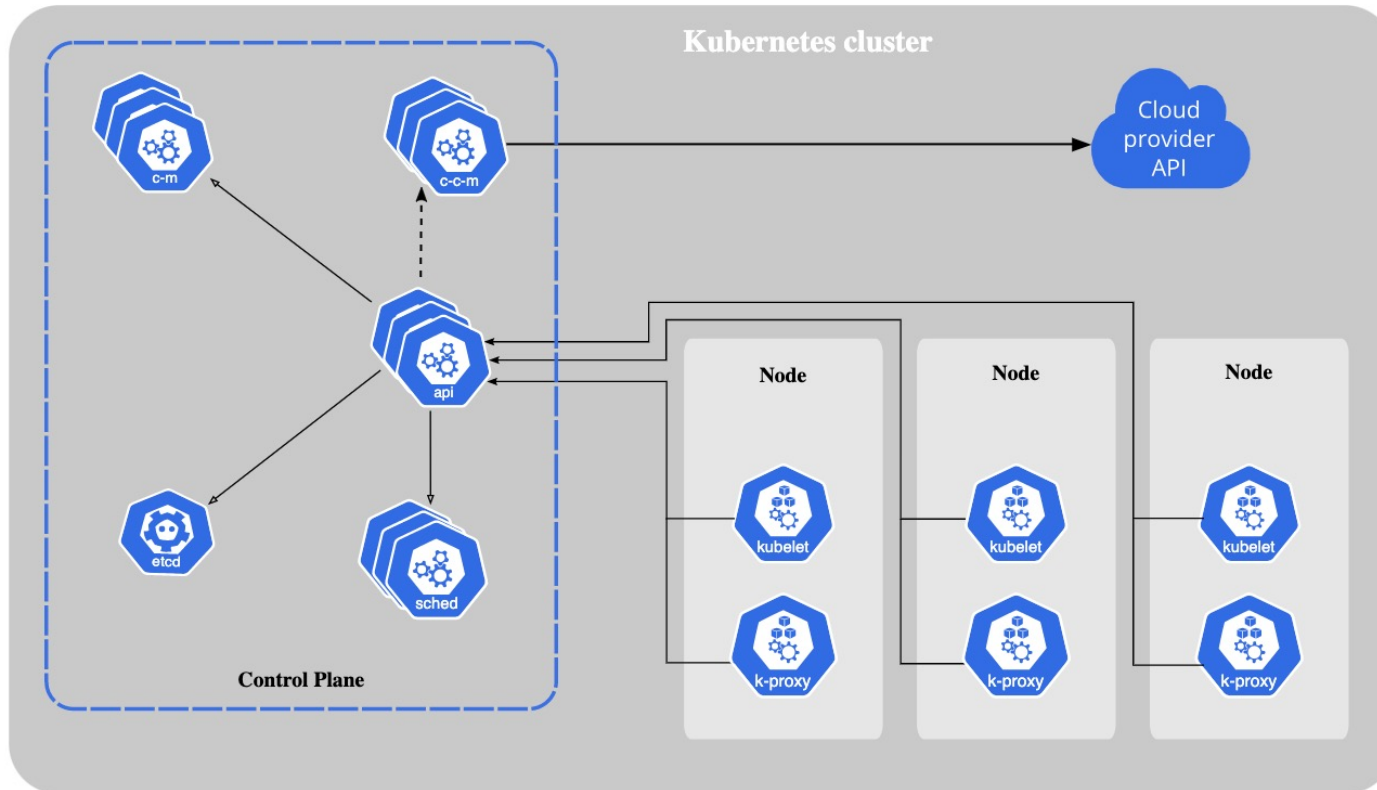
Self-healing
Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

Secret and configuration management
Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

kubernetes

Orchestrateur
de container

Composants Kubernetes

Control Plane Components
- kube-apiserver
- etcd
- kube-scheduler
- kube-controller-manager
- cloud-controller-manager

Node Components
- kubelet
- kube-proxy

https://kubernetes.io/docs/concepts/overview/components/

Extensions Kubernetes

Operator

- https://kubernetes.io/docs/concepts/extend-kubernetes/operator/

Motivation

- The Operator pattern aims to capture the key aim of a human operator who is managing a service or set of services. Human operators who look after specific applications and services have deep knowledge of how the system ought to behave, how to deploy it, and how to react if there are problems.
- People who run workloads on Kubernetes often like to use automation to take care of repeatable tasks. The Operator pattern captures how you can write code to automate a task beyond what Kubernetes itself provides.

Operators in Kubernetes

- Kubernetes is designed for automation. Out of the box, you get lots of built-in automation from the core of Kubernetes. You can use Kubernetes to automate deploying and running workloads, *and* you can automate how Kubernetes does that.
- Kubernetes' controllers concept lets you extend the cluster's behaviour without modifying the code of Kubernetes itself. Operators are clients of the Kubernetes API that act as controllers for a Custom Resource.

Fichiers de distribution de IKO

Vous pouvez obtenir IKO depuis 2 sources :

➤ **le référentiel containers.intersystems.com**

```
$ docker pull containers.intersystems.com/intersystems/iris-operator:2.0.0.223.0
```

➤ **le site Web de distribution du WRC dans la section Composants.**

https://wrc.intersystems.com/wrc/coDistGen.csp

InterSystems Components

Please contact InterSystems Support if the kit you are looking for is not present.

Click here to return to the main distribution page.

Components									
Name	**Product**	**Version**	**Maint**	**Build**	**Os**	**Arch**	**File Type**	**Size (Mb)**	**Download ...**
kub ✖									
InterSystems Kubernetes Operator	IRIS	2.0	0	223.0	Unix	x64	tar.gz	24	⬇

```
iris_operator-2.0.0.223.0
├── 111README
├── chart
│   └── iris-operator
│       ├── Chart.yaml
│       ├── README.md
│       ├── templates
│       │   ├── NOTES.txt
│       │   ├── _helpers.tpl
│       │   ├── apiregistration.yaml
│       │   ├── appcatalog-user-roles.yaml
│       │   ├── cleaner.yaml
│       │   ├── cluster-role-binding.yaml
│       │   ├── cluster-role.yaml
│       │   ├── deployment.yaml
│       │   ├── mutating-webhook.yaml
│       │   ├── service-account.yaml
│       │   ├── service.yaml
│       │   ├── user-roles.yaml
│       │   └── validating-webhook.yaml
│       └── values.yaml
├── image
│   └── iris_operator-2.0.0.223.0-docker.tgz
└── samples
    ├── compute.cpf
    ├── data.cpf
    ├── iris-gs2019-aks.yaml
    ├── iris-gs2019-eks.yaml
    ├── iris-gs2019-tke.yaml
    ├── iris-gs2019.yaml
    ├── iris-ssd-sc-aks.yaml
    ├── iris-ssd-sc-eks.yaml
    ├── iris-ssd-sc-gke.yaml
    ├── iris-ssd-sc-tke.yaml
    └── stash
        ├── gcs-repo.yaml
        ├── iris-gs2019-backupbatch.yaml
        ├── iris-gs2019-backupconfig.yaml
        └── iris-gs2019-backupsession.yaml

6 directories, 32 files
```

Tutorials

Using InterSystems Kubernetes Operator

https://docs.intersystems.com/irislatest/csp/docbook/Doc.View.cls?KEY=AIKO

Community : InterSystems Kubernetes Operator Deep Dive: Part 2
- https://community.intersystems.com/post/intersystems-kubernetes-operator-deep-dive-part-2

Developer Community

Ask questions and find answers! Sign up or use your InterSystems Learning / WRC credentials to join the conversation.

https://community.intersystems.com/

InterSystems
Developer Community

https://openexchange.intersystems.com/

https://docs.intersystems.com/

https://learning.intersystems.com

InterSystems®

Creative data technology

Merci