

# Azure Spring Cloud : Microsoft et VMware poussent les microservices Java

Le PaaS [Azure Spring Cloud](#) aura passé un peu moins d'un an en préversion.

Microsoft et VMware assurent conjointement l'ingénierie et l'exploitation du service. Ils viennent d'en [annoncer](#) la disponibilité générale, [dans 10 régions Azure](#) dont l'Europe de l'Ouest et l'Europe du Nord.

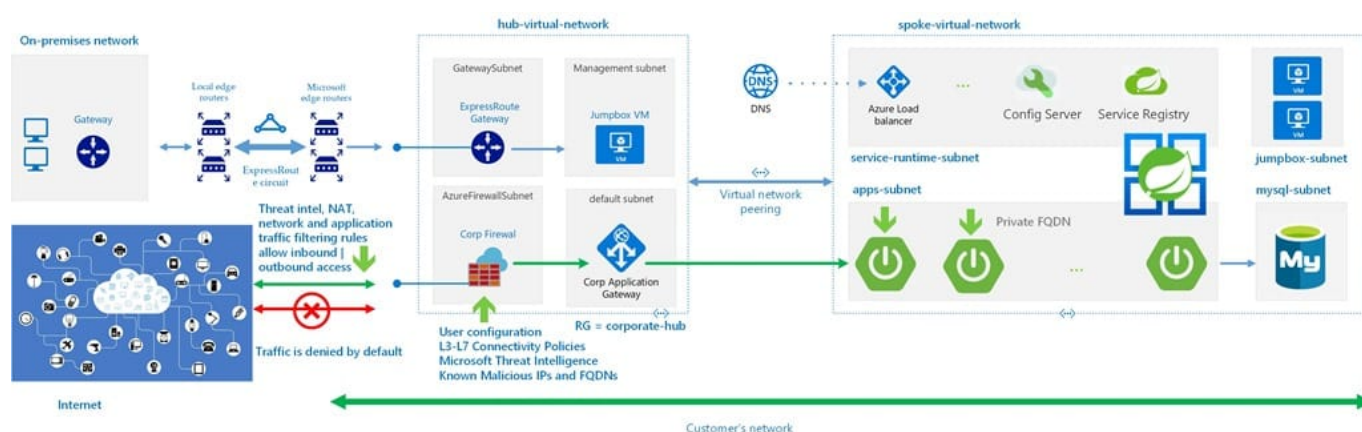
Leur promesse : une offre managée qui facilite le déploiement en cloud des applications Java développées avec [Spring](#). Ou plus exactement avec un *framework* dérivé voulu plus simple à l'usage : [Spring Boot](#).

Pour implémenter les patterns spécifiques aux systèmes distribués, le *middleware* [Spring Cloud](#) a été mis en place. Il inclut notamment un registre de services, un équilibreur de charge côté client et des *circuit breakers*.

Azure Spring Cloud en est une distribution managée. Axée sur les applications architecturées en microservices, elle repose sur le service d'orchestration de conteneurs [Azure Kubernetes Services](#). Et se nourrit de [kpack](#), bibliothèque de contrôleurs Kubernetes destinée à automatiser la création et la mise à jour d'images de conteneurs.

## **L'autoscale, mais en preview**

Microsoft propose un portefeuille de connecteurs à intégrer dans les applications Java pour faire la jonction avec des services Azure. Parmi eux, Cosmos DB (base de données), Event Hubs (monitoring) et Azure Key Vault ([gestion de secrets](#)).



Azure Spring Cloud embarque une fonctionnalité de [mise à l'échelle automatique](#), qui reste toutefois en préversion. Même chose pour l'option « Managed Virtual Network », qui permet l'utilisation du PaaS en environnement privé.

The screenshot displays the Azure portal interface for configuring a scale rule. On the left, the 'Autoscale setting' page shows 'Custom autoscale' selected. The 'Scale rule' configuration on the right includes the following details:

- Metric source:** Current resource (default)
- Resource type:** Azure Spring Cloud
- Resource:** petclinic-microservices
- Criteria:**
  - Time aggregation:** Average
  - Metric namespace:** Azure Spring Cloud standard metrics
  - Metric name:** system.cpu.usage
  - Time grain:** 1 minute
- Dimension Name, Operator, Dimension Values:**

Dimension Name	Operator	Dimension Values
App	=	1 selected
Deployment	=	default
Instance	=	2 selected
- Scale out rule:** When petclinic-microservices (Average) system.cpu.usage > ... Increase count to 4
- Instance limits:** Minimum: 2, Maximum: 8, Default: 2
- Schedule:** This scale condition is executed when none of the other scale condition(s) match

A line graph at the bottom right shows the average system CPU usage over time, with a current value of 6.0297%.

La [tarification](#) se fait sur deux niveaux :

- Basic (sans SLA, avec un maximum de 25 instances d'applications ; recommandé en dev/test)
- Standard (avec [SLA à 99,9 %](#) et un maximum de 500 instances d'applications)

Chaque instance d'application inclut des capacités mémoire et processeur. En l'occurrence, 10 Go et 5 vCPU en Basic, 32 Go et 16 vCPU en Standard. Lorsqu'on dépasse l'une ou l'autre, on est facturé en « utilisation supplémentaire » (Go/h et vCPU/h).

*Illustration principale © Macrovector – shutterstock.com*