



Thomas, avec tous les offres de Cloud Public qui existent, pourquoi les clients ne migrent pas dessus ?

Excellente question. Comme tu le sais aujourd'hui, on peut héberger ses services/applications dans un **Cloud privé**, on parle alors d'**On-Premise**.

On peut aussi utiliser **plusieurs fournisseurs Cloud public**, c'est ce qu'on appelle le **multi-cloud**.

Effectivement, mais cela doit être compliqué de gérer des ressources déployées un peu partout ?!

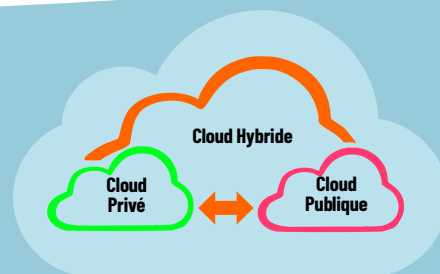


Tout à fait, et c'est là que le service **Azure Arc** prend tout son sens, car il devient un point d'entrée unique pour la gestion des ressources déployées à **l'extérieur d'Azure**.

Il permet donc de gérer les **environnements hybrides et multi-cloud**, mais en plus, ces ressources peuvent être intégrées à d'autres services Azure comme pour la sécurité ou le monitoring.



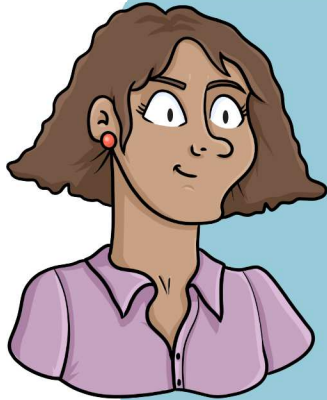
Un service pour gérer toutes ses ressources depuis Azure cela paraît super cool !



Et en plus, c'est super simple.



Avant d'aller plus loin, revenons sur les **raisons** qui poussent les entreprises à avoir des **environnements hybrides** c'est à dire faire cohabiter de l'**On-Premise** et du **Cloud public** ou bien encore du **multi-cloud**.



La 1ère, peut être liée à une **contrainte technique, métier ou légale** qui oblige à héberger les données dans le pays dans lequel est implantée la société, à l'image des banques.

La 2nde, est la **latence**. Certains services/applications nécessitent d'avoir des temps de réponses très bas. C'est pour cela qu'est apparu le **Edge Computing**, c'est-à-dire exécuter les charges de travail, au plus près de sources de données.

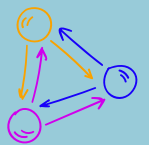


La suivante peut être simplement une volonté de ne pas avoir qu'un seul fournisseur pour **éviter le vendor locking**. L'idée est de ne pas mettre tous ses œufs dans le même panier.

Une autre raison est de pouvoir **basculer automatiquement** ses charges de travail ailleurs en cas d'incident sur l'un des fournisseurs ou dans son **DC**.



Evidemment cela doit être prévu lors de la définition de l'architecture surtout si ton application doit être **hautement disponible**.



Ce qui entraine forcément un prix plus élevé car certaines ressources sont déployées sans forcément être utilisées.

Exactement

Et dernières raisons, **des frais engagés** dans la modernisation de son DC ou simplement le fait de ne **pas avoir de temps** à consacrer à une **migration** dans le Cloud.



Tu comprends que toutes ces raisons, ajoutent de la complexité à la gestion des ressources.

Oui sans aucun doute.

Voilà pourquoi **Azure Arc** a été conçu, pour **simplifier** leur gestion.

Commençons par le plus simple avec la gestion des **serveurs sur Azure Arc**.

Cette fonctionnalité permet de gérer des **VMs** ou des **serveurs physiques** déployés en dehors d'Azure, comme si ils étaient déployés dans Azure. Et cela marche aussi pour ceux qui hébergent du SQL.

Comment cela fonctionne ?

Il suffit simplement d'installer l'agent, "**Connected Machine Agent**" sur les serveurs ou VMs, afin de les remonter dans **Azure Arc**.

Il faut aussi, que les **flux sortants** de ta VM soient **ouverts** en direction d'Azure pour envoyer certaines informations, qu'on va évoquer plus bas.

Et une fois dans **Azure Arc**, qu'est-ce que tu peux faire avec ?

Assieds toi avant que je commence.

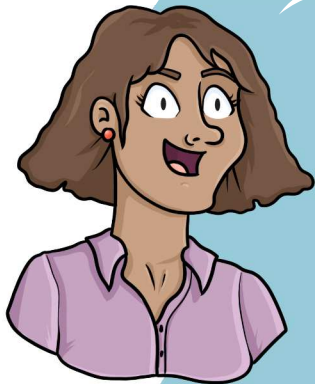
Tu peux auditer les configurations au sein des VMs avec **Azure Policy**, tu peux les protéger avec **Microsoft Defender for Cloud**, et ainsi remonter les vulnérabilités, et détecter les menaces.

L'intégration avec **Azure Automation** te permet d'installer les patches de sécurité, d'exécuter des **runbooks** pour les tâches récurrentes, ou de suivre les modifications avec la fonctionnalité **Change Tracking and Inventory ...**

Mais aussi remonter les logs et les métriques de performances directement dans **Azure Monitor**.

Et en plus, **Azure Private Link** est supporté !

suivante



Continuons avec **Kubernetes (K8S) sur Azure Arc**.



Cette fonctionnalité permet de gérer directement dans Azure Arc les **clusters K8S** certifiés Cloud Native Computing Foundation (**CNCF**) qui sont déployés On-Premise, ou chez d'autres fournisseurs comme AWS ou GCP.

Cela nécessite également l'installation d'un agent ?



Non le fonctionnement est un peu différent. Cela fonctionne avec l'**installation de modules et d'extensions**.

Il faut d'abord inscrire les fournisseurs Microsoft qui vont être utilisés. Ce sont des modules qui permettent d'effectuer des tâches sur le cluster. Ensuite, après la création d'un **Resource Group (RG)** qui va héberger le cluster **K8S**, il suffit d'exécuter une commande pour connecter le cluster à Azure Arc.



Cela n'a pas l'air très compliqué ?!

Non c'est super simple !

Ensuite, l'installation d'extensions à ton cluster te permettront de déployer des applications, d'**activer le monitoring** avec Azure Monitor, d'**activer la protection** contre les menaces avec Microsoft Defender for Kubernetes ou bien encore Azure Policy.

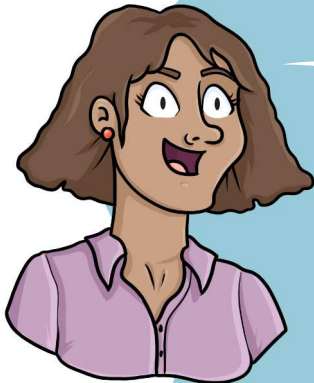
C'est génial, qu'on puisse intégrer les clusters **K8S** avec les **services natifs Azure** !



C'est effectivement un **vrai point fort** de Azure Arc.

suivante

Mais ce n'est pas tout parce qu'on peut également gérer les ressources déployées dans **Azure Stack HCI** directement depuis Azure Arc.



Tu peux nous rappeler ce qu'est **Azure Stack HCI** ?

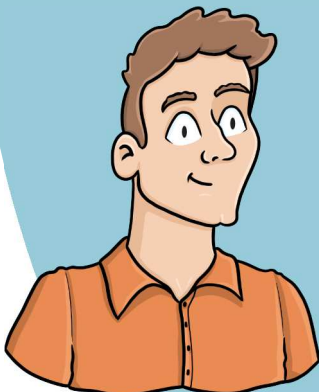
Oui, c'est l'**offre Azure** qui te permet de déployer des ressources Azure sur un équipement **hyperconvergé**, que tu vas pouvoir installer directement au sein de ton **DC**.

Très utile pour ceux qui ont des contraintes légales à utiliser le Cloud Public, ou alors ceux qui ne sont pas encore prêt à franchir le pas.

C'est plutôt bien résumé. **Azure Stack HCI** offre une interface qui ressemble fortement au portail Azure, histoire que les clients ne soient pas perdus si un jour ils décident de basculer sur Azure.



**Azure Arc** permet de gérer les ressources déployées dans **Azure Stack HCI** comme si elles étaient déployées dans Azure. On va retrouver les mêmes services Azure, comme Azure Monitor pour le monitoring, Azure Policy pour la conformité, le patching via l'Update Management ... Et en plus, il prend en charge **Azure Stack HCI single-node**.



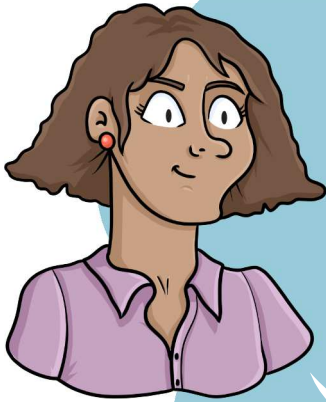
Pour le moment, entre les **VMs**, les **serveurs**, **Kubernetes** et **Azure Stack HCI**, on a adressé depuis Azure Arc uniquement les composants **d'infrastructure**. mais il est également possible de gérer, et même de déployer des services PaaS de bases de données ou d'applications directement dans Azure Arc.

Je vais me répéter mais c'est très ingénieux !

suyvante

Parlons maintenant des **services data sur Azure Arc**.

Cette fonctionnalité permet aujourd'hui, de déployer des **SQL Managed Instances** ou **PostgreSQL Hyperscale** en utilisant K8S comme **socle d'hébergement**.



Pour gérer ce type de services, Microsoft propose 2 modes différents : le **mode connecté** (Directly Connected Mode), ou le **mode déconnecté** (Disconnected mode ou Indirectly connected mode)

Quelle est la différence entre ces 2 modes ?



La différence se situe au niveau du mode de connexion entre le service déployé et Azure, pour la **transmission** de certaines **informations vers Azure Arc**.

Comme les metadata, que tu vas pouvoir stocker dans la région de ton choix, mais aussi les données liées à la facturation, les données d'inventaire, les logs, les métriques de performance, ...

Le **mode connecté** s'effectue par une connexion **internet**, une connexion **VPN** ou via un **ExpressRoute**.

Le **mode déconnecté**, est utilisé par les clients qui ont des environnements très sécurisés **sans possibilité d'établir de connexions sortantes vers Internet**, soit des clients qui disposent d'une **mauvaise connexion Internet**.

Sans connexion comment fais-tu pour déployer les services sur ton cluster **K8S** ?



Il te suffit de récupérer depuis les registry publics de Microsoft, les images prêtes à l'emploi, pour les déployer localement.

La **transmission des informations** se fait via des **exports de fichiers**, manuels ou automatiques, que tu charges ensuite sur Azure.

On vient de parler de la brique Data, mais il est aussi possible de déployer des **services applications sur Azure Arc**.

Comme des services Web IIS ?

Encore mieux que cela !

Tu vas pouvoir **déployer en dehors d'Azure**, des services très populaires comme des **App Service**, des **Functions**, des **Logic Apps**, des **API Management** ou bien encore des **Event Grid**.

Au lieu de les déployer dans des régions Azure, ces services sont déployés dans ce qu'on appelle des **emplacements personnalisés (Custom Locations)** qui correspondent à des environnements On-Premise, AWS, GCP, ...

Comme la brique Data, les services sont déployés sur un **socle K8S**, grâce aux extensions dont nous avons déjà parlé tout à l'heure. L'extension va générer un script à exécuter sur ton cluster K8S. Tu pourras après cela directement **depuis Azure Arc déployer l'un des services évoqués plus haut**.

**AMAZING**

Je suis désolé de le dire mais Azure Arc est un service de la mort qui tue !

Effectivement et **Azure Arc continuera à s'enrichir** dans les prochaines semaines avec de nouveaux services Azure qui feront leur apparition, mais aussi avec **l'intégration d'autres services natifs Azure**.

C'est l'utilisation combinée de l'hybride et du multi-cloud qui a permis l'éclosion de Azure Arc, pour simplifier la vie des utilisateurs en gérant plus efficacement leurs ressources déployées sur différents environnements.

suivante

C'est là qu'on se rend compte de la puissance d'Azure Arc

Et encore on a pas pu tout voir.

Si tu souhaites approfondir le sujet, tu peux consulter la documentation officielle ou aller sur **Microsoft Learn**.

Et si tu veux pratiquer, Microsoft a mis à disposition le site **azurearcjumpstart.io** avec plein de contenus et des cas d'usages.

Super, et peut-on tester Azure Arc ?

Oui, il te suffit simplement d'une souscription Azure.

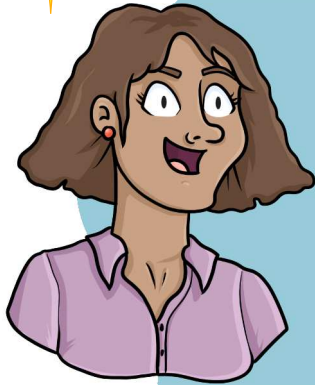
Tu trouveras aussi sur le site Jumpstart, la solution **ArcBox** pour déployer un environnement de tests complet pour commencer à jouer avec.

Donc si je résume ce que nous avons vu :

**Azure Arc** est un **point d'entrée unique** qui te permet de **gérer les ressources déployées en dehors d'Azure**.

Des briques d'infrastructure comme des VMs, des serveurs physiques, et bien évidemment des clusters K8S, mais aussi des briques d'applications, et même des briques data que l'on trouve en mode connecté ou déconnecté selon son contexte.

C'est un excellent résumé !







Si vous souhaitez continuer à **apprendre**, de façon ludique, sur **l'écosystème Azure**, et ne rater aucune de nos illustrations ...

... N'hésitez pas à vous abonner sur :



<https://aka.ms/grow-una>



<https://tinyurl.com/youtube-growuna>

Et si le contenu vous plaît, partagez-le ;o)

A très vite !

