AZURE KUBERNETES SERVICE ON AZURE STACK HCI

Leverage your Azure Stack HCI investment to run containerized applications alongside traditional virtual workloads in highly available and resilient patterns. In this quick start guide, you will learn how to install and configure the Azure Kubernetes Service for Azure Stack HCI (AKS-HCI), using Windows Admin Center. From there, you will be able to explore additional scenarios around deployment, monitoring and management of containerized applications and workloads. You will perform the following tasks as part of this guide:

- Plan and configure Supermicro hardware and OS/tools to support the AKS-HCI deployment
- Step-by-step documentation to install and configure AKS-HCI including integration with Azure Arc

Introducing Azure Kubernetes Service on Azure Stack HCI

Azure Kubernetes Service on Azure Stack HCI (AKS-HCI) is a Kubernetes-based orchestrator that automates running containerized applications on clusters that use Azure Stack HCI. Orchestrators such as the open-source Kubernetes, automate much of the work involved with deploying and managing multiple containers. However, Kubernetes can be complex to set up and maintain. AKS-HCI helps simplify setting up Kubernetes on-premises, making it quicker to get started hosting Linux and Windows containers. AKS-HCI is currently in preview and has no added cost during preview. In order to use this guide, you'll need to first register for the preview.

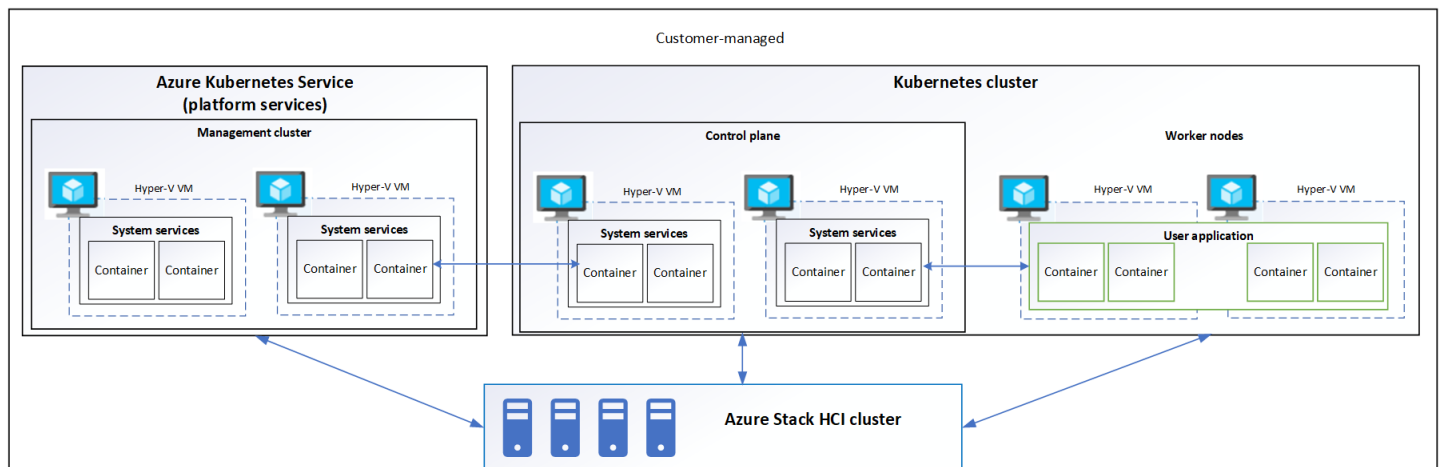**Azure Kubernetes Service on Azure Stack HCI**

Figure 1 - High-level AKS-HCI Architecture

From an architectural perspective, as shown above, AKS-HCI consists of several key components. At the base, is your Azure Stack HCI cluster, which not only serves as the platform for the Azure Kubernetes Service and containerized workloads, but also continues to be a powerful enterprise platform for traditional virtualized workloads.

On the left, you have a management cluster. When you create an AKS-HCI cluster, a management cluster is automatically created and configured. This management cluster is responsible for provisioning and managing target clusters where workloads run. It includes an API server, and a Load Balancer. On the right-hand side, you have your target (workload) clusters. The target cluster is a highly available deployment of Kubernetes using Linux VMs for running Kubernetes control plane components as well as Linux worker nodes. Windows Server based VMs are used for establishing Windows worker nodes. All of this can be quickly and easily deployed using the Windows Admin Center, or PowerShell.

AZURE KUBERNETES SERVICE ON AZURE STACK HCI

Why Azure Kubernetes Service on Azure Stack HCI?

Azure Stack HCI provides enterprise customers with a highly available, cost-efficient, flexible platform to run high-performance workloads. These workloads could run within traditional virtual machines, but also, within containers, ensuring you get the best utilization from your hyperconverged infrastructure. Azure Stack HCI provides a compelling platform for kubernetes for a number or reasons; firstly, it allows you to quickly scale virtual machines, including container hosts, to optimize workloads across your physical hardware resources, and it's quick and easy to add additional physical resources to increase capacity.

In addition, Azure Stack HCI allows VMs, including container hosts, to migrate across physical server hosts, to optimize utilization and account for infrastructure maintenance. It also provides automated VM failover, within a local cluster, or a cluster stretched across sites. Within the VMs themselves, containers and pods can move across the container hosts, offering an extra level of redundancy across physical and virtual. And all of this is running on a stack, completely supported by Microsoft.

Before you get started

The following summarizes, at a high-level what you will need to run Azure Kubernetes Service on Azure Stack HCI.

- An Azure Stack HCI version 20H2 cluster with 1TB storage capacity in your storage pool and at least 30GB of available memory for running Azure Kubernetes Service virtual machines.
- A system running Windows Admin Center. This should have at least 40GB free space, be registered with Azure, and will be in the same domain as the Azure Stack HCI cluster, once created.
- A network with IPv4 DHCP enabled

You can also review more detailed requirements information within our documentation.

Deploying Azure Kubernetes Service on Azure Stack HCI

The first step in the process is to obtain and setup your hardware that will support Azure Stack HCI, and subsequently, AKS-HCI.

Step 1: Hardware and OS configuration for Azure Kubernetes Service on Azure Stack HCI

Supermicro 1U 12 Bay Drawer X11 SSG-6119P-ACR12N4L offers the highest storage density in the least amount of rack space. Using onboard high-speed networking chipsets, it offers an optimized out-of-the-box solution for the AKS-HCI scenario. Below is the detailed tech spec.

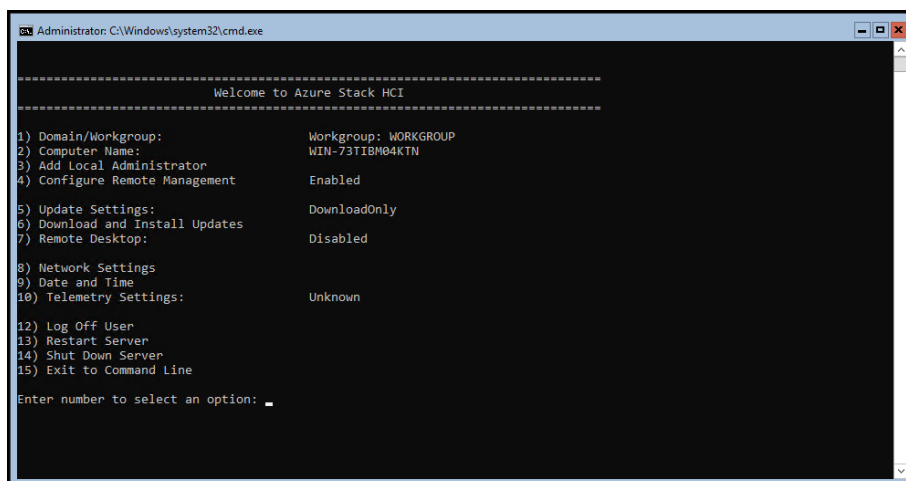# AZURE KUBERNETES SERVICE ON AZURE STACK HCI

Supermicro SSG-6119P-ACR12N4L-HCI

Scale
2 to 4 nodes

Single Node Data
- 16 to 56 cores (intel)
- 128GB to 3TB memory
- 4TB to 192TB storage
- NVMe + HDD
- Up to 100GbE

With your hardware racked, configured, and connected, you are ready to deploy the Azure Stack HCI OS. The first step in deploying Azure Stack HCI is to download Azure Stack HCI and install the operating system on each server that you want to cluster. You can deploy Azure Stack HCI using your preferred method – this could be via USB, network deployment, ISO boot over a dedicated OOB management port etc, Step through the simple Azure Stack HCI OS installation wizard, and once complete, you should be at the Server Configuration Tool (SCONFIG) interface. If you need to, make any simple changes here, but all that should be required is a single NIC with an IP address on your management network.

Figure 2 - SCONFIG in Azure Stack HCI

AZURE KUBERNETES SERVICE ON AZURE STACK HCI

Step 2: Deploy and Configure Windows Admin Center

With your Azure Stack HCI nodes deployed, and accessible over the network, the next step is to deploy the Windows Admin Center. If you haven't already, download the Windows Admin Center software. This should be installed on a Windows 10 or Windows Server 2016/2019 machine, and this machine should have at least 40GB free space. This machine should also be joined to your management domain. This should be the same domain that your Azure Stack HCI nodes will be joined to.

Step 3: Create Azure Stack HCI Cluster

With the Windows Admin Center installed, open the Windows Admin Center, and step through the process of creating an Azure Stack HCI cluster.
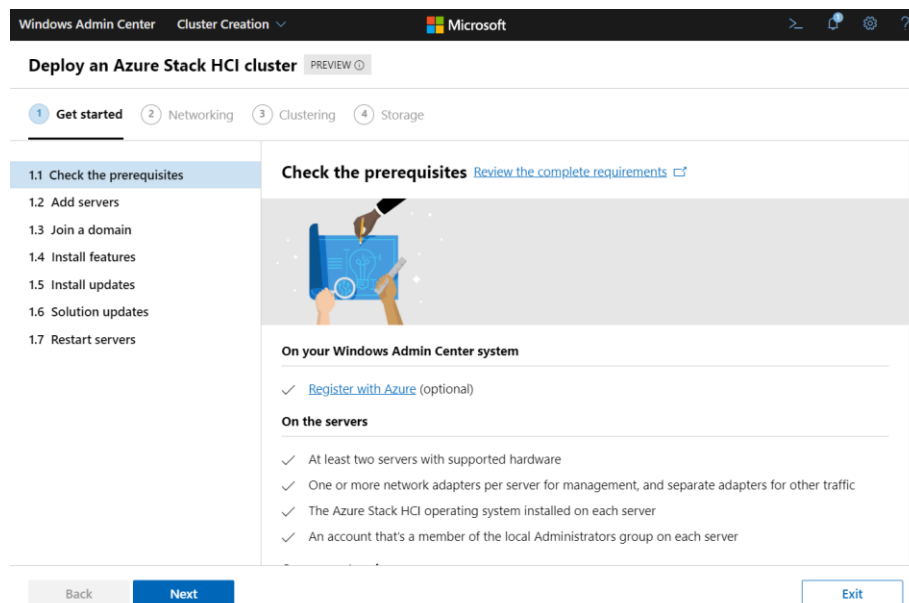

Figure 3 - Deploying an Azure Stack HCI Cluster in Windows Admin Center

The wizard will walk you through selecting your nodes, joining the nodes to the domain, installing required roles and features, and updates, before moving on to configuring the physical and virtual networks, clustering and software defined storage. When the wizard is complete, you should see your new cluster in your **All connections** view within Windows Admin Center.

Step 4: Prepare Windows Admin Center for AKS-HCI

With your Azure Stack HCI cluster under management of Windows Admin Center, the next step is to add the Azure Kubernetes Service on Azure Stack HCI extension. You would have received a link to download when you registered for the preview. Place the extension file in a folder on your Windows Admin Center machine, such as C:\AKS-HCI, or on an SMB share accessible to the Windows Admin Center machine. Once the extension file is downloaded, follow the steps to install the extension.
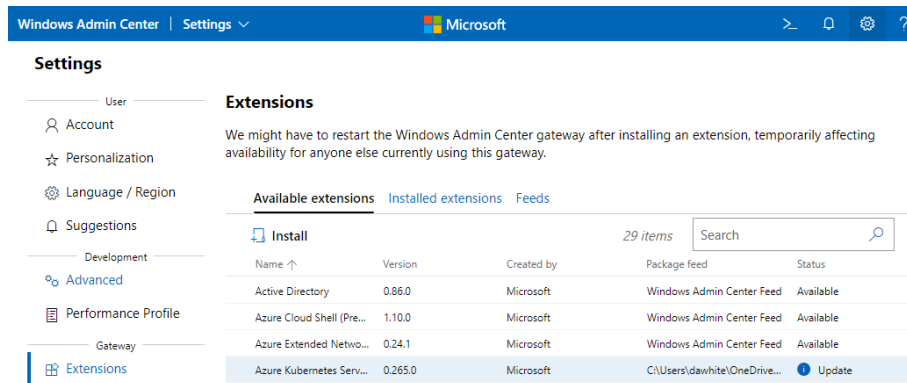
AZURE KUBERNETES SERVICE ON AZURE STACK HCI

Figure 4 - Adding the AKS-HCI Extension in Windows Admin Center

Step 5: Set up the Azure Kubernetes Service Host

There is one final step that should be completed before you create a Kubernetes cluster. You need to set up an Azure Kubernetes Service host on the system you want to deploy the Kubernetes cluster to. Firstly, within Windows Admin Center, make a connection to your Azure Stack HCI cluster, and once connected, in the bottom-left corner under **Extensions**, click on **Azure Kubernetes Service**, and in the central pane, click **Set up**, to launch the wizard.
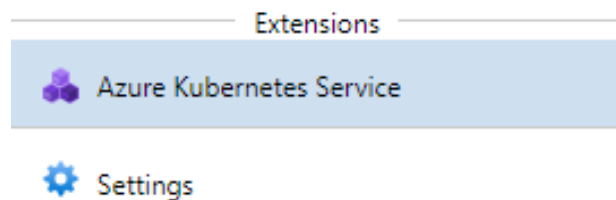

Figure 5 - Azure Kubernetes Service in Windows Admin Center

[Follow the documented steps](link) to complete the simple wizard. The wizard will check your cluster and your Windows Admin Center system, then allow you to configure the Azure Kubernetes Service host, defining storage and networking, along with Azure integration. **Note, there is no cost for setting up and using AKS-HCI in public preview**. However, the underlying Azure Stack HCI cluster may still be charged. Please refer to the Azure Stack HCI service for pricing. Upon completing the wizard, the deployment will begin, and once complete, you will be ready to deploy a kubernetes cluster.

Step 6: Creating a kubernetes cluster

Now that the Azure Kubernetes Service is set up on your Azure Stack HCI cluster, it works similarly to the hosted Azure Kubernetes Service: you use the service to create Kubernetes clusters that run your containerized applications. These Kubernetes clusters are groups of VMs that act as worker nodes, running your application containers. These workers could be Linux or Windows-based. The Kubernetes cluster also contains a control plane, which consists of Kubernetes system services used to orchestrate the application containers.

To start the process, on the homepage for Windows Admin Center, click **Add**, navigate to the Kubernetes clusters tile, and click **Create new** to launch the Kubernetes cluster wizard.
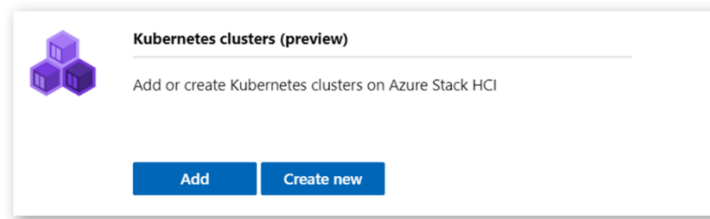
AZURE KUBERNETES SERVICE ON AZURE STACK HCI

Figure 6 - Create a new kubernetes cluster in Windows Admin Center

Walk through the wizard, providing information about your desired kubernetes cluster. You'll firstly optionally choose to integrate the new kubernetes cluster with Azure Arc, provide the details of your Azure Kubernetes Service host, created earlier, and define your primary node pool size, which will run key system services. From there, you will walk through creating additional workload pools, running Windows, or Linux. You will define some networking settings, persistent storage, then review your settings and create your cluster. Once completed, you can download your SSH key and kubernetes dashboard token, and if you chose to integrate with Azure Arc, click the link to be taken directly to the Azure portal for more configuration options.

Next Steps

With your kubernetes cluster deployed, you are at a point where you can explore some more advanced scenarios, including:

- Running Linux applications in AKS-HCI - Run a multi-container application that includes a web front end and a Redis instance. You will also scale and test your application.
- Running Windows applications in AKS-HCI - run an ASP.NET sample application in a Windows Server container to the cluster. You then see how to test and scale your application.
- Integrate your kubernetes cluster with Azure Arc – assuming you chose not to integrate during the Windows Admin Center wizard, this guide will walk you through integrating your kubernetes cluster with Azure Arc.
- Once integrated with Azure Arc, you can proceed to use Azure Policy to apply cluster configurations, deploy configurations using GitOps, and enable monitoring of your kubernetes cluster.

Summary

Following this guide, you have installed Azure Stack HCI, deployed Windows Admin Center and integrated the Azure Kubernetes Service extension. You have then deployed the Azure Kubernetes Service host onto your Azure Stack HCI cluster, and setup a Kubernetes cluster for your workloads. At this point, you are ready to continue your kubernetes learnings by exploring the resources covered in the next steps above.