

# AZURE STACK HCI: TRUSTED ENTERPRISE VIRTUALIZATION

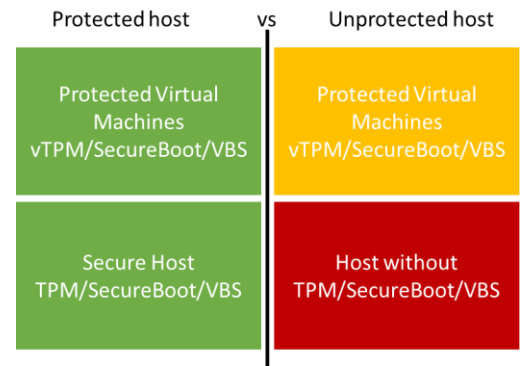
Leverage your Azure Stack HCI investment to run workloads on a highly secure infrastructure by choosing the hardware designed for the Trusted enterprise virtualization scenario, with unparalleled levels of operating system security enabled with virtualization-based security (VBS) and hybrid cloud capabilities made easy through Windows Admin Center and Azure portal.

Below, you will find a how-to guide for building an infrastructure for the Trusted enterprise virtualization scenario on Azure Stack HCI.

## Overview of Trusted enterprise virtualization scenario

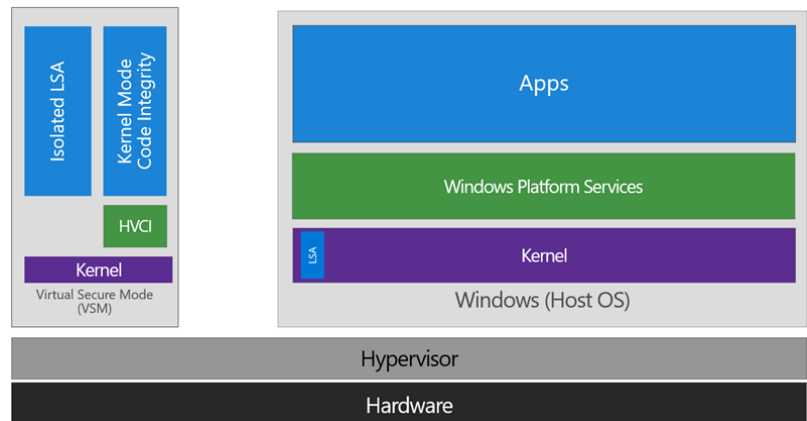
Virtualization-based security (VBS) is a key component of the [security investments in Azure Stack HCI](#) to protect hosts and virtual machines from security threats.

For example, the [Security Technical Implementation Guide \(STIG\)](#) is published as a tool to improve the security of Department of Defense (DoD) information systems, and lists VBS and hypervisor-protected-code-integrity (HVCI) as general security requirements. It is imperative to use host hardware that is VBS and HVCI enabled, in order for the protected workloads on virtual machines to fulfil their security promise because protection of virtual machines is not guaranteed on a compromised host.



VBS uses hardware virtualization features to create and isolate a secure region of memory from the normal operating system. Windows can use this "virtual secure mode" to host a number of security solutions, providing them with greatly increased protection from vulnerabilities in the operating system, and preventing the use of malicious exploits which attempt to defeat protections.

VBS uses the Windows hypervisor to create this "virtual secure mode", and to enforce restrictions which protect vital system and operating system resources, or to protect security assets such as authenticated user credentials. With the increased protections offered by VBS, even if malware gains access to the operating system kernel the possible exploits can be greatly limited and contained, because the hypervisor can prevent the malware from executing code or accessing platform secrets.



One such security solution example is HVCI, which uses VBS to significantly strengthen code integrity policy enforcement. Kernel mode code integrity checks all kernel mode drivers and binaries before they are started and prevents unsigned drivers or system files from being loaded into system memory.

HVCI leverages VBS to run the code integrity service inside a virtual secure mode, providing stronger protections against kernel viruses and malware. The hypervisor, the most privileged level of system software, sets and enforces page permissions across all system memory. Pages are only made executable after code integrity checks inside the virtual secure mode have passed, and

# AZURE STACK HCI: TRUSTED ENTERPRISE VIRTUALIZATION

executable pages are not writable. That way, even if there are vulnerabilities like buffer overflow that allow malware to attempt to modify memory, code pages cannot be modified, and modified memory cannot be made executable.

## How to deploy VBS and HVCI-enabled Azure Stack HCI

### 1. Plan Hardware Deployment

All the Azure Stack HCI solutions by **Thomas-Krenn.AG** are certified for the Hardware Assurance Additional Qualification, which tests for [all the functionality needed for VBS](#). However, VBS and HVCI are not automatically enabled in Azure Stack HCI and Step 2 will guide you on how to enable them.

**Warning:** Hypervisor-protected code integrity (HVCI) may be incompatible with devices not listed in the Azure Stack HCI catalog. Microsoft strongly recommends using an Azure Stack HCI validated solution from our hardware partners for the Trusted enterprise virtualization scenario.

- Thomas-Krenn.AG Azure Stack HCI Systems
  - [Azure Stack HCI Rack-Series Variants](#)
  - [Azure Stack HCI Micro-Cluster Variants](#)
- Plan your Azure Stack HCI environment with [Azure Stack HCI Solutions from Thomas-Krenn.AG](#)
  - Maximum flexibility and scalability
  - 2x AMD 3<sup>rd</sup> Generation EPYC CPUs each Node
  - Up to 4 TB RAM each Node
  - Up to 2x 200 GbE RDMA-Network each Node
  - Individual calculated Storage-Capacity available <sup>(1)</sup>
    - NVMe-Storage

(1) Nested two-way Mirror is highly recommended for 2-Node-Azure-Stack-HCI-Clusters

Please consider backup and restore times for large-capacity configurations. Most data protection solutions are not able to ingest and restore at speeds businesses expect.

- Network and Switch Connectivity
  - [RoCE/RDMA](#) technology for Storage Spaces Direct Traffic.  
With RoCE there is additional DCB and PFC configuration required. Network devices have to support this capabilities.
  - Available as switch-connected or direct-attached solution <sup>(2)</sup>
    - One single/standalone switch
    - Two redundant/stacked switches
    - Direct-attached for RDMA (storage & cluster) traffic


(2) Thomas-Krenn.AG recommends a switched configuration. The switch helps the cluster determine whether loss of node connectivity is due to node or network failure, ensuring more predictable failover behavior.

### 2. Deploy VBS-Enabled Azure Stack HCI

**Step by Step guide** to [deploy Azure Stack HCI](#). Also install [Windows Admin Center \(WAC\)](#) for managing Azure Stack HCI.

[Enable virtualization-based protection of code integrity](#)

## AZURE STACK HCI: TRUSTED ENTERPRISE VIRTUALIZATION

3. From Windows Admin Center (WAC), set up Azure Security Center to add threat protection and quickly assess your security posture of your workloads.
  - You can also setup additional  Azure hybrid services such as Backup, File Sync, Site Recovery, Point-to-Site VPN, Update Management, and Azure Monitor in WAC.

### Summary

With the completion of the Azure Stack HCI Trusted enterprise virtualization deployment and the configuration of VBS / HVCI, you now have a platform with the highest security standards for protecting security sensitive workloads on both physical and virtual machines.