







HVR's Data Integration Support for Azure Technologies & Services

Why HVR for Azure

HVR delivers efficient continuous data integration for the cloud: on-premises to cloud, cloud to on-premises and cloud to cloud environments. With HVR, you can set up once and deploy often, creating efficiencies across your team. Additionally the HVR platform enables you to feed multiple destinations continuously and also as a one-time load.

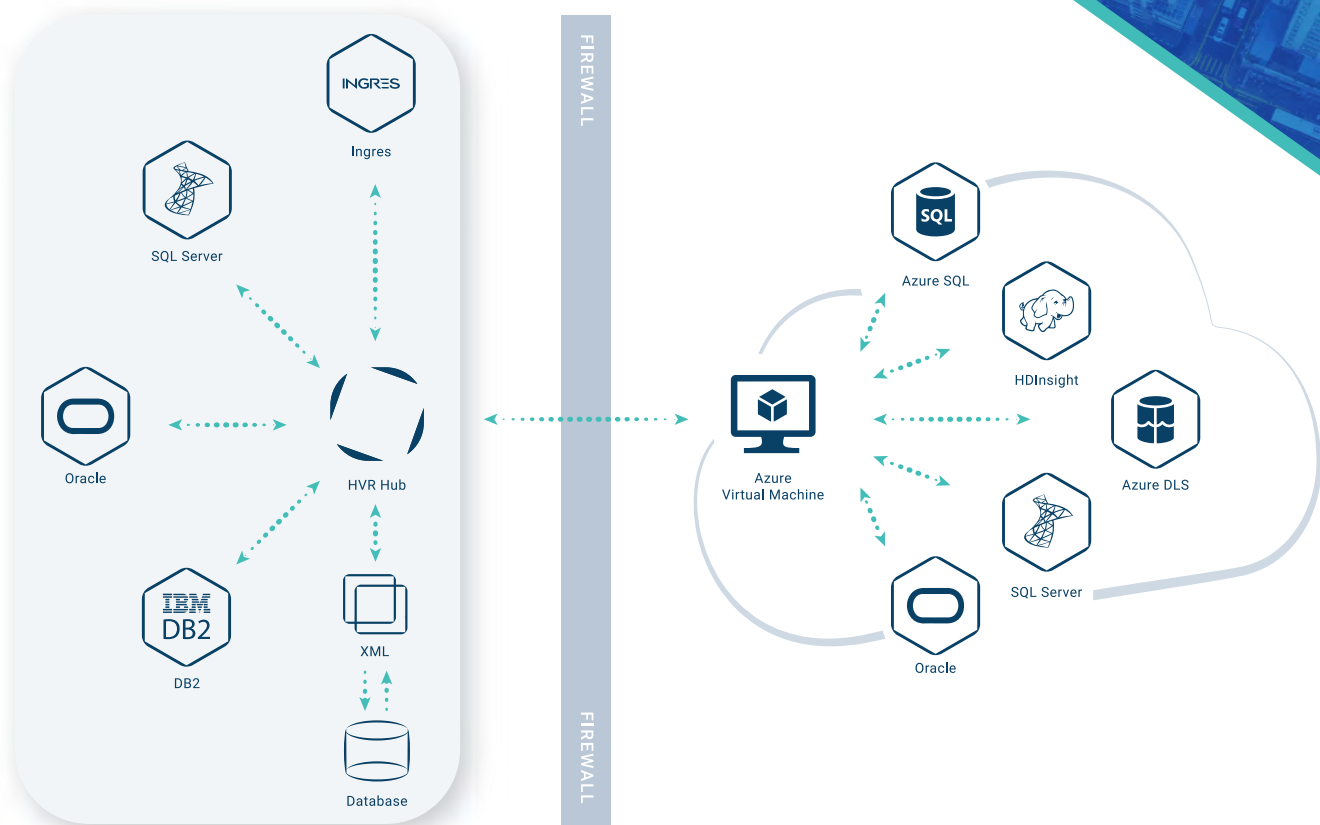
Starting with tables in an existing database or application, HVR enables you to:

-  Create target tables by automatically mapping your source system data types to compatible data types, ensuring no loss in precision in the Azure destination.
-  Perform a one-time load, referred to as refresh, integrated with continuous Change Data Capture (CDC).
-  Move data in real-time, typically via a log-based CDC method. You can achieve optimized continuous delivery to your Azure technologies and services.
-  Check your data with data validation and repair. This is a unique function that enables you to ensure the data you moved is correct and alerts you if there are any data discrepancies.
-  Setup the data flows and any automated alerts in the HVR management console.
-  Monitor your data integration statistics graphically to obtain insights in a browser-based environment, accessed through the management console.

HVR Distributed Architecture

HVR's modular architecture is designed to work well on relatively high latency, low bandwidth networks such as Wide Area Network (WAN) connections between on-premises and cloud environments, or between availability zones or regions in the cloud. Installing the HVR architecture in your environment is low impact and includes the use of optional agents and a hub (one of the installations in an HVR setup, which plays the roll of coordinator). The hub always initiates communication and hence determines in which direction the firewall must be opened. You can choose where to run the hub (e.g. many on-premises to Azure setups run the hub on-premises and only open the firewall to Azure).

An agent, which is an additional installation of software on or near the source or target database or data store, facilitates the movement of changes. Agents don't have to be installed on database servers, but to optimize efficiency agents should be used when communicating across a WAN.



When moving data using HVR, the following features ensure performance and security:

- Proprietary compression and large data block transfer ensure fast performance maximizing bandwidth utilization irrespective of latency
- Secure network communication using SSL/TLS encryption with the use of explicit certificates. Certificates can also be used for more secure two-factor authentication in addition to username/password validation.
- Use of a proxy is supported so that individual systems or database access is not exposed through the firewall but instead data is securely routed through a proxy. (e.g. in a DMZ (De-Militarized Zone))

Supported Services

In the Azure cloud, the majority of data storage services are available through Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). Some technology vendors like [Snowflake also leverage the Azure infrastructure to run their branded cloud services.](#)

To achieve optimum efficiency and fastest performance, we strongly recommend installation of an HVR agent in the availability zone of the source and/or destination data store. Agent sizing is dependent on the data store technology but can be very modest.

HVR is compatible with the infrastructure service Load Balancer to enable transparent scale out for agents in a high volume data delivery scenario.

IaaS SUPPORT

HVR supports all technologies installed on Virtual Machines in Azure using IaaS as if these were on-premises installations. For example, HVR customers use Oracle Databases directly running on Linux-based Azure Virtual Machines, as well as SQL Server on Windows VMs and various other databases and technologies.

The list of supported technologies continuous to expand. [Refer to the HVR website](#) for a current overview of supported technologies.

PaaS SUPPORT

Many Azure services can be used as a target for continuous data integration and some can also act as a source. Azure services can be the target for any of the source technologies HVR supports. The following Azure services are supported natively:



Azure SQL database as a service is supported as a target for change delivery, and can be used as a source through trigger-based capture. Other transactional database services, Azure database for MySQL and Azure database for PostgreSQL, can also serve as a target for continuous data integration.



The Azure SQL data warehouse solution is supported as a target. Changes are integrated most efficiently using direct path loads into staging tables, with integration running through micro-batches so that on aggregate, the Azure SQL data warehouse can keep up with the number of row changes arriving from one or more busy transactional systems.



Azure Data Lake Store (ADLS) is supported both as a destination and as a source. The use case of a data lake using ADLS as a destination is popular, with HVR supporting the management of files e.g. store files organized by table, per year/month/day, and data publication through the use of manifest files. Supported data formats include JSON, Parquet, Avro, CSV and XML with many options to fine-tune the format and control compression.



HVR supports the use of Hive tables on top of the data files in ADLS, both to enable direct access, and to support compare between the source database and target data in ADLS.

Beyond the natively-supported PaaS technologies, additional destinations can be utilized through the agent plugin framework. The agent plugin is a program or script that runs in a pre-defined environment with data available in files. Python is commonly used to write an agent plugin script, but other programming languages can be used as well. Plugin examples are included in the HVR installation directory \$HVR_HOME/lib/agent.

Get Started with HVR

[Contact Us](#)

ABOUT HVR

We accelerate data movement so that you can revolutionize your business. HVR is designed to move large volumes of data FAST and efficiently in hybrid environments for real-time updates.