Virtualized Test Data for DevOps and TDM via Windocks and Voracity

Smart Test Data Production Meets Modern Database Provisioning

Abstract:

Windocks is a leader in database virtualization (cloning) for delivery of writable database environments in seconds, each consuming less than 40 MB on delivery. This white paper primarily introduces Windocks capabilities, including SQL Server, PostgreSQL, and MySQL containers, database and folder virtualization. It also introduces -- and summarizes the integration of -- IRI Voracity test data creation jobs (data scrubbing, subsetting, and synthesis), and compares the Windocks-IRI bundle against Delphix on technical and commercial grounds.

Windocks

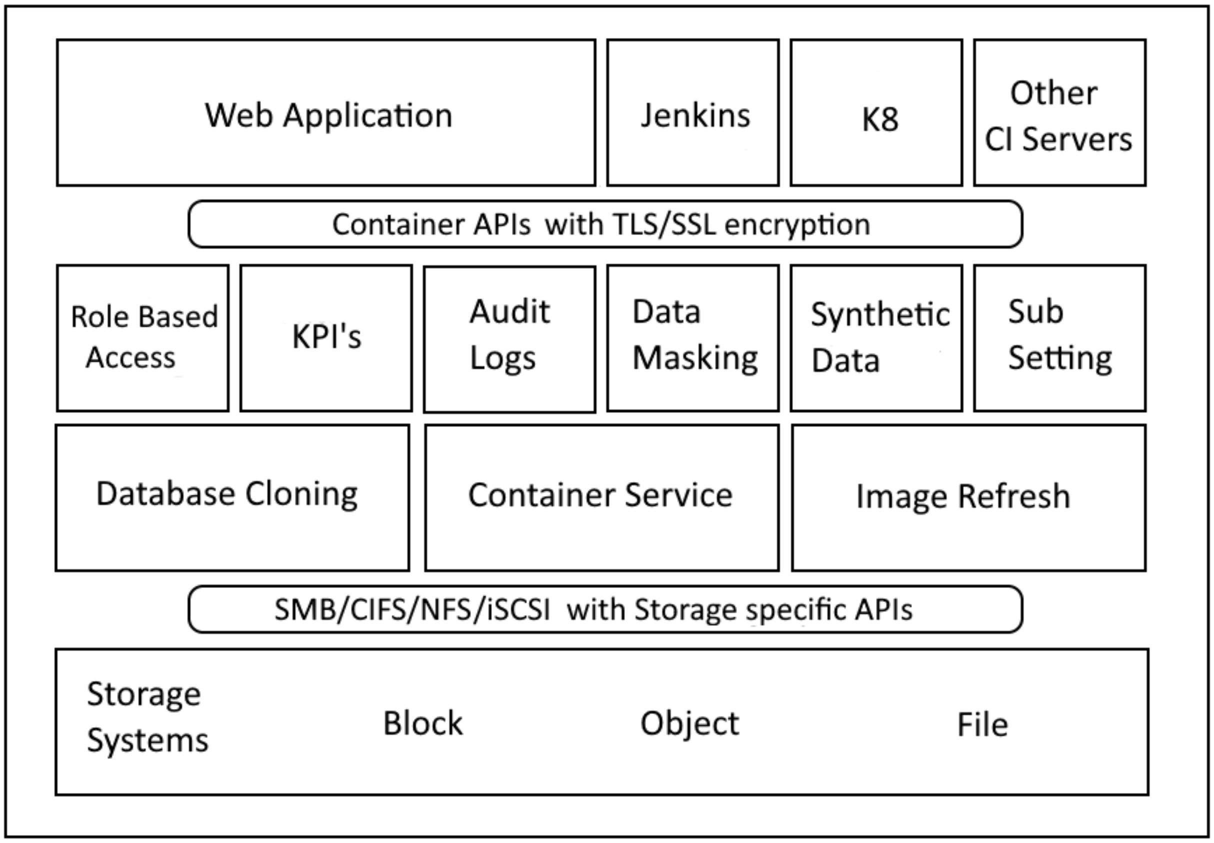
IRI Voracity
An Insatiable Appetite for Data

Introduction

Database virtualization is increasingly included in enterprise plans for test data management (TDM), along with requests for database containers. A primary challenge in virtual TDM is integrating the disparate tools and capabilities needed to pull it off, ranging from database virtualization, to data masking, subsetting, and synthetic data generation, to operational automation and governance.

Poor interoperability between tools often forces TDM architects into expensive or incomplete solutions, and steep learning curves. Database virtualization has also been complicated by expensive, proprietary, client/server storage. A consequence is that TDM continues to rely on sole-sourced megavendor approaches which lack support for containers and Kubernetes, offer fewer de-ID and smart test data features, and limit access to metadata and auditing options.

Windocks and IRI have partnered to break down these barriers with a comprehensive but affordable TDM solution bundle that works with existing enterprise and cloud storage, and delivers on-demand compliant databases for containers, Kubernetes, ordinary instances, and workstations. The goal is modern TDM that can be configured in hours and deployed in minutes.



Windocks Containers

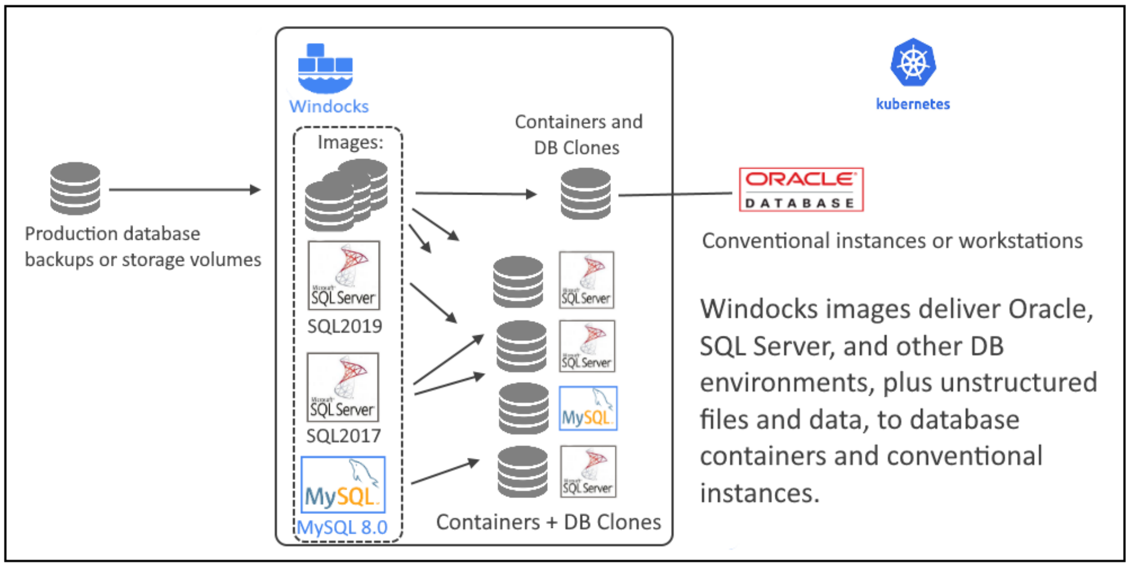
Windocks launched in 2015 as an independent port of Docker's open source project to Windows. Windocks decision to compete with an ostensibly free Docker Windows implementation from Microsoft proved prescient, as Microsoft's SQL Server division now focuses exclusively on Linux containers.

Windocks is the market and technical leader in Windows SQL Server containers, supporting all editions and releases of SQL Server 2008 to 2019, with full SQL Server services (DB engine, Reporting, Analysis, Integration, and Agent). Windocks also supports Postgres and MySQL containers.

An automated Windocks installer supports installation on Windows Server 2012 R2, 2016, and 2019. Locally installed SQL Server instances serve as images, and are cloned and delivered as SQL Server containers. This approach delivers several benefits:

1. Windocks SQL Server containers are identical to conventional installed instances, and compatible with Active Directory, SQL Writer, and all other infrastructure.
2. Windocks containers preserve the trusted SQL Server security model and avoid Docker image security concerns.
3. These SQL Server containers require no new SQL Server licensing as Microsoft licenses include the right to install or clone "unlimited SQL Server instances" with per-core or per-server licenses.

Windocks database virtualization is unique in supporting containers as well as conventionally installed instances and workstations. The ability to install any on premise infrastructure or cloud, and deliver SQL Server containers that are identical to conventional instances simplifies adoption. SQL Server, Postgres, and MySQL containers quickly become the preferred approach for development, testing, and other uses.


Windocks
Images:
Production database backups or storage volumes
SQL2019
SQL2017
MySQL 8.0
Containers and DB Clones
Containers + DB Clones
kubernetes
ORACLE DATABASE
Conventional instances or workstations
Windocks images deliver Oracle, SQL Server, and other DB environments, plus unstructured files and data, to database containers and conventional instances.

Database Virtualization

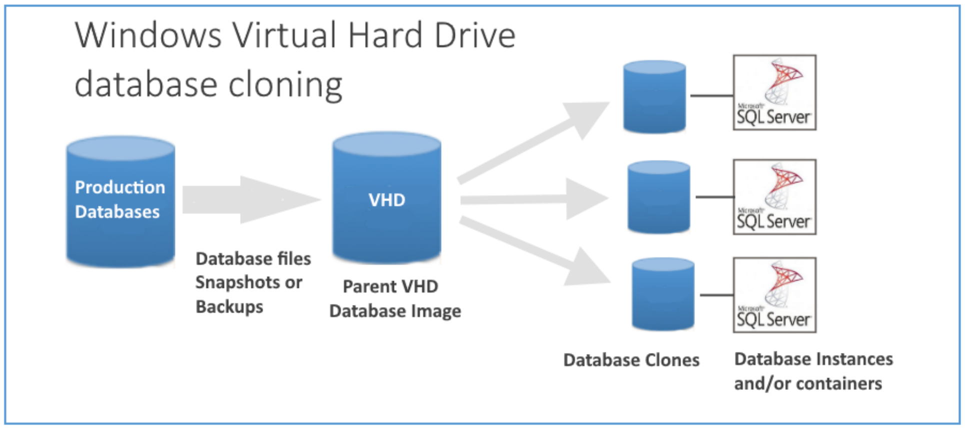
Database virtualization delivers writable databases in seconds, regardless of size, with each clone consuming only 40 MB on delivery. Here, Windocks can deliver any RDB environment. For example, Oracle test environments can be delivered to Linux/Oracle instances on the LAN.

Two technologies are used:

1. Windows Virtual Hard Drives (VHDs), a file format representing a hard disk.
2. Volume snapshots and volume clones on NetApp, Pure Storage, and other storage systems.

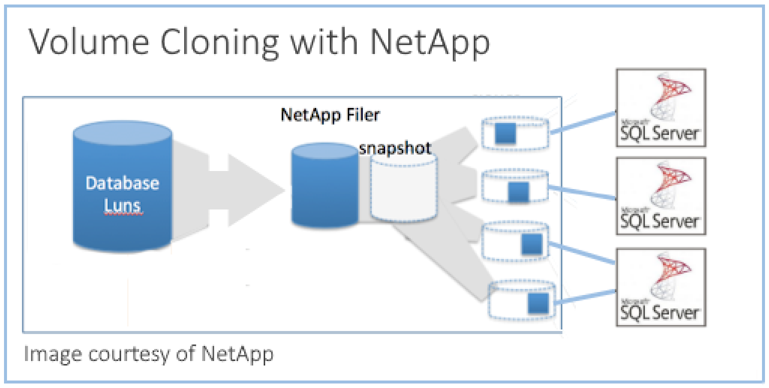
VHDs are an industry standard file format that leverages commodity server-attached storage. Databases are copied or restored to a VHD, creating a full byte copy VHD image that is cloned by creating a child VHD (a differencing disk).

Cloned databases are attached to database instances or containers. The VHD image is read-only, while each clone supports writes based on a Copy on Write from the image.


Windows Virtual Hard Drive database cloning
Production Databases
Database files Snapshots or Backups
VHD
Parent VHD Database Image
SQL Server
SQL Server
SQL Server
Database Clones
Database Instances and/or containers

Volume-based database virtualization begins with a snapshot of a source (typically production) database environment. The snapshot is "thin volume" cloned, and databases are mounted and attached to the target database container or instance.

Most block or volume storage systems include these capabilities, and support restful APIs, but existing arrays may be limited to scripting for creation of snapshots and thin volume cloning.



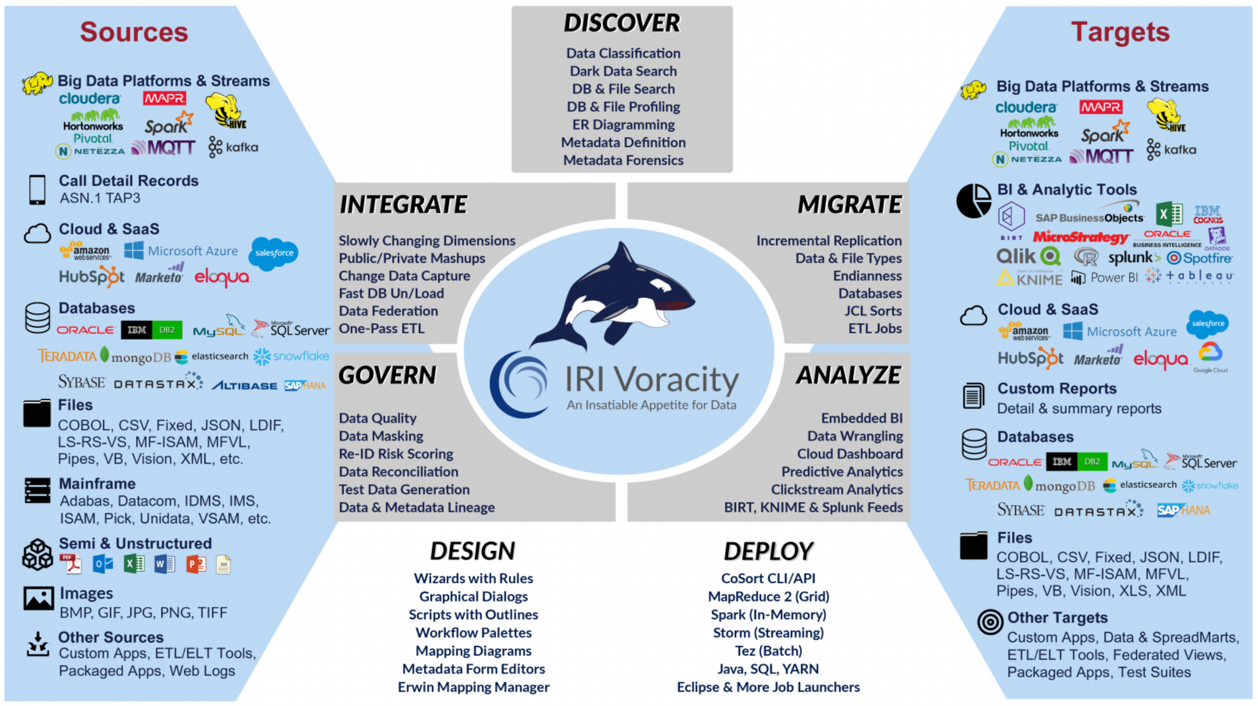
Volume-based database virtualization creates inherent storage dependencies, which is a recognized anti-pattern in DevOps success, and is expensive to purchase and operate. Storage-based cloning also generally involves client-side software or agents, and staging servers are needed for applying data masking, subsetting, or synthetic data.

Directory cloning of files and unstructured data

Git is an increasingly popular tool for cloning directories. Both Windocks and IRI support the ability to clone and deliver virtualized files and folders of text and non-relational data. In the latter case for example, the EGit provider in IRI Workbench can serve as a secure repository for both the synthesized, subsetted or masked targets created in standalone file formats, as well as the metadata assets associated with each project that were used to generate those test sets.

The IRI Workbench environment, built on Eclipse, is a free graphical integrated development environment (IDE) where IRI Voracity jobs that cleanse, mask, subset, or synthesize relational and flat-file test data are built. Workbench users have a variety of job design and deployment options within that, and other venues, because its metadata framework is common and open.

Note that Voracity users can also leverage IRI DarkShield functionality via Workbench or API to search and mask NoSQL databases as well as semi- and unstructured text, document, and image files. DarkShield shares the same data classes and masking functions with IRI FieldShield (for atomic RDB and flat-file data), to maintain enterprise data integrity.

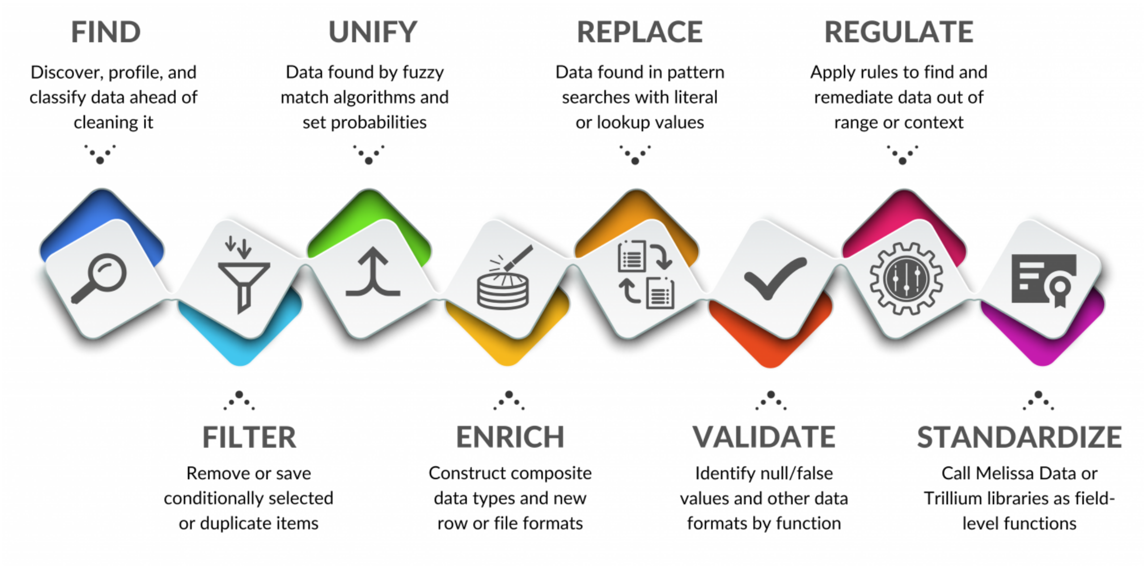


The point is that database and file targets from any Voracity production job, or test data creation job (further summarized below), can be containerized and virtualized in Windocks.

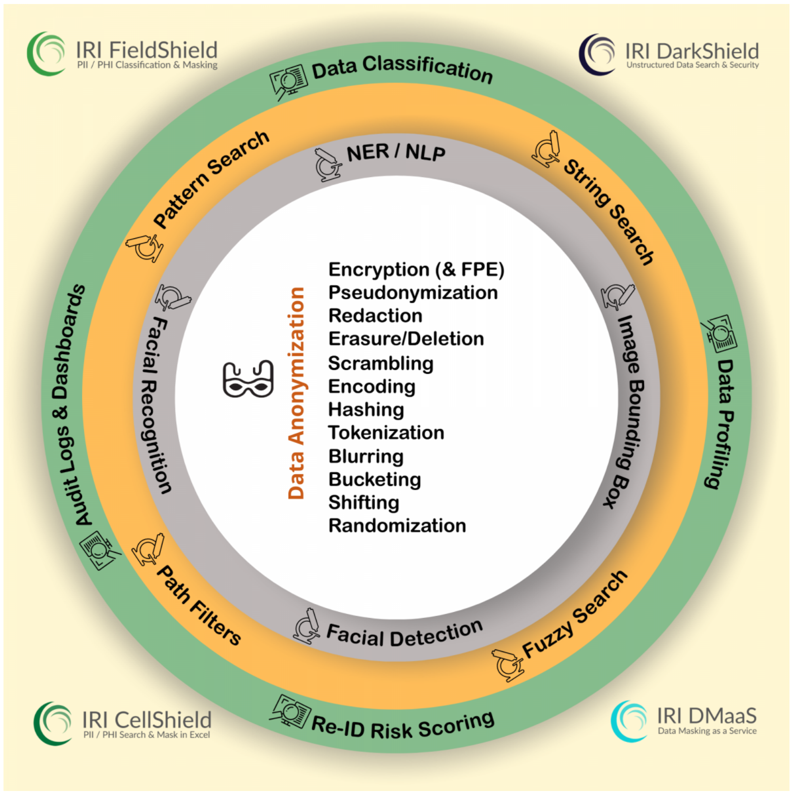
IRI Data Scrubbing, Subsetting and Synthesis

Windocks database virtualization begins by building an immutable image that supports delivery of test-ready environments. A plain text dockerfile specifies each database image which is built contemporaneously with execution of an IRI batch script defining a multi-table Voracity data cleansing, masking, subsetting or synthetic data generation jobs built in IRI Workbench.

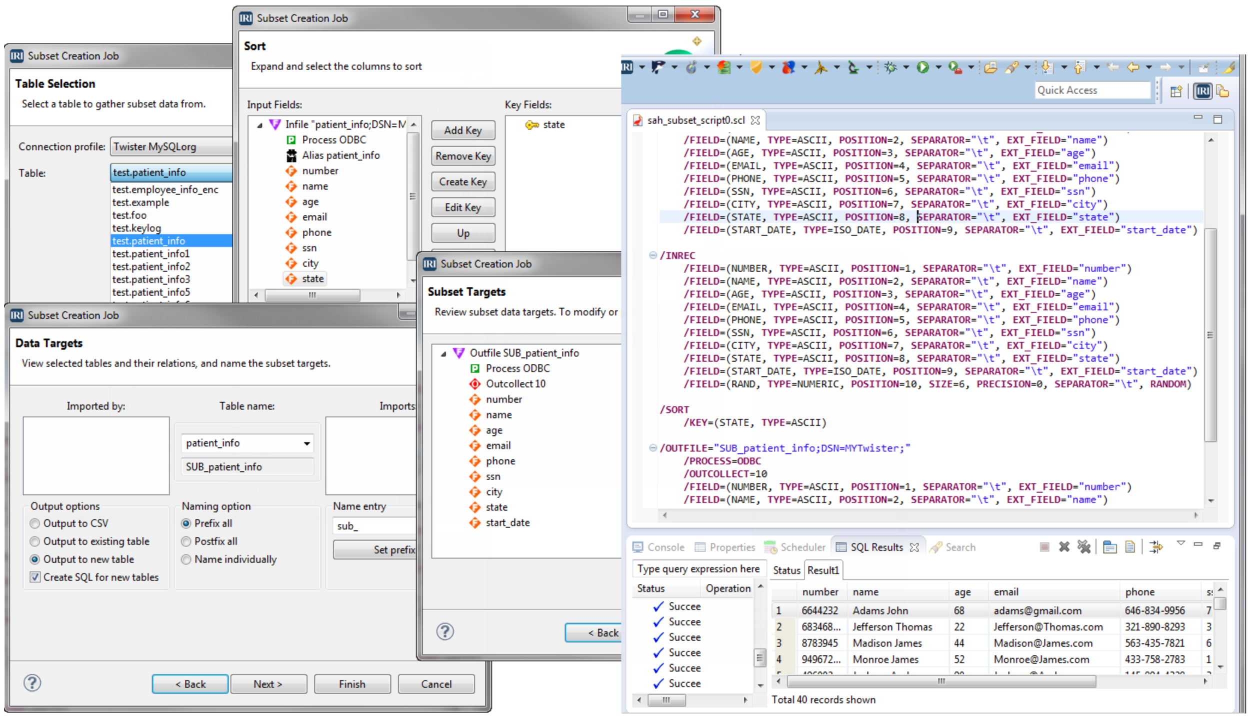
Here are the IRI CoSort-powered data cleansing capabilities in Voracity:


FIND
Discover, profile, and classify data ahead of cleaning it
UNIFY
Data found by fuzzy match algorithms and set probabilities
REPLACE
Data found in pattern searches with literal or lookup values
REGULATE
Apply rules to find and remediate data out of range or context
FILTER
Remove or save conditionally selected or duplicate items
ENRICH
Construct composite data types and new row or file formats
VALIDATE
Identify null/false values and other data formats by function
STANDARDIZE
Call Melissa Data or Trillium libraries as field-level functions

FieldShield and DarkShield data masking functions in Voracity:



Relational database subsetting (with optional rule-based masking) in Voracity:



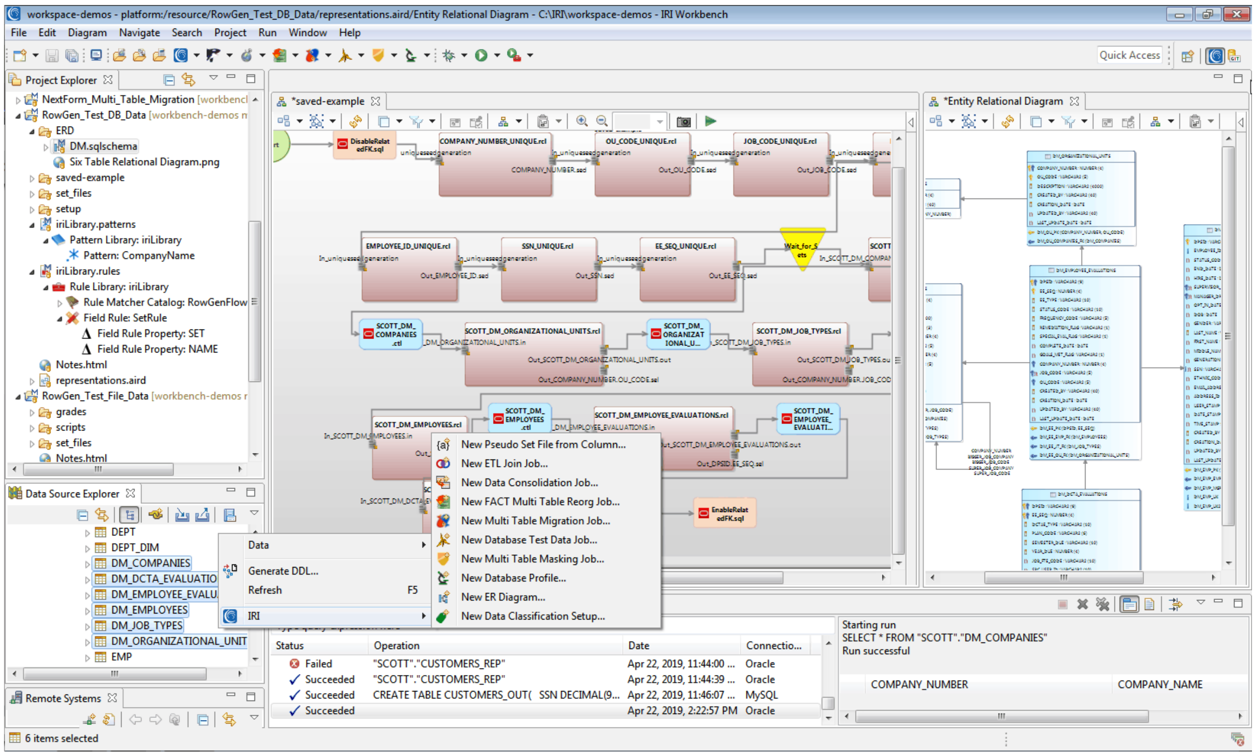
```
    /FIELD=(NAME, TYPE=ASCII, POSITION=2, SEPARATOR="\t", EXT_FIELD="name")
    /FIELD=(AGE, TYPE=ASCII, POSITION=3, SEPARATOR="\t", EXT_FIELD="age")
    /FIELD=(EMAIL, TYPE=ASCII, POSITION=4, SEPARATOR="\t", EXT_FIELD="email")
    /FIELD=(PHONE, TYPE=ASCII, POSITION=5, SEPARATOR="\t", EXT_FIELD="phone")
    /FIELD=(SSN, TYPE=ASCII, POSITION=6, SEPARATOR="\t", EXT_FIELD="ssn")
    /FIELD=(CITY, TYPE=ASCII, POSITION=7, SEPARATOR="\t", EXT_FIELD="city")
    /FIELD=(STATE, TYPE=ASCII, POSITION=8, SEPARATOR="\t", EXT_FIELD="state")
    /FIELD=(START_DATE, TYPE=ISO_DATE, POSITION=9, SEPARATOR="\t", EXT_FIELD="start_date")

/INREC
    /FIELD=(NUMBER, TYPE=ASCII, POSITION=1, SEPARATOR="\t", EXT_FIELD="number")
    /FIELD=(NAME, TYPE=ASCII, POSITION=2, SEPARATOR="\t", EXT_FIELD="name")
    /FIELD=(AGE, TYPE=ASCII, POSITION=3, SEPARATOR="\t", EXT_FIELD="age")
    /FIELD=(EMAIL, TYPE=ASCII, POSITION=4, SEPARATOR="\t", EXT_FIELD="email")
    /FIELD=(PHONE, TYPE=ASCII, POSITION=5, SEPARATOR="\t", EXT_FIELD="phone")
    /FIELD=(SSN, TYPE=ASCII, POSITION=6, SEPARATOR="\t", EXT_FIELD="ssn")
    /FIELD=(CITY, TYPE=ASCII, POSITION=7, SEPARATOR="\t", EXT_FIELD="city")
    /FIELD=(STATE, TYPE=ASCII, POSITION=8, SEPARATOR="\t", EXT_FIELD="state")
    /FIELD=(START_DATE, TYPE=ISO_DATE, POSITION=9, SEPARATOR="\t", EXT_FIELD="start_date")
    /FIELD=(RAND, TYPE=NUMERIC, POSITION=10, SIZE=6, PRECISION=0, SEPARATOR="\t", RANDOM)

/SORT
    /KEY=(STATE, TYPE=ASCII)

/OUTFILE="SUB_patient_info;DSN=MYTwister;"
    /PROCESS=ODBC
    /OUTCOLLECT=10
    /FIELD=(NUMBER, TYPE=ASCII, POSITION=1, SEPARATOR="\t", EXT_FIELD="number")
    /FIELD=(NAME, TYPE=ASCII, POSITION=2, SEPARATOR="\t", EXT_FIELD="name")
```

Status	Operation
✓ Succee	
✓ Succee	
✓ Succee	
✓ Succee	
✓ Succee	
✓ Succee	

	number	name	age	email	phone	s
1	6644232	Adams John	68	adams@gmail.com	646-834-9956	7
2	683468...	Jefferson Thomas	22	Jefferson@Thomas.com	321-890-8293	3
3	8783945	Madison James	44	Madison@James.com	563-435-7821	6
4	949672...	Monroe James	52	Monroe@James.com	433-758-2783	1

Total 40 records shown

RowGen structurally and referentially correct Data Vault and RDB test data synthesis in Voracity:


workspace-demos - platform:/resource/RowGen_Test_DB_Data/representations.aird/Entity Relational Diagram - C:\IRI\workspace-demos - IRI Workbench
File Edit Diagram Navigate Search Project Run Window Help
Quick Access
Project Explorer
NextForm_Multi_Table_Migration [workbench
RowGen_Test_DB_Data [workbench-demos m
ERD
DM.sqlschema
Six Table Relational Diagram.png
saved-example
set_files
setup
iriLibrary.patterns
Pattern Library: iriLibrary
Pattern: CompanyName
iriLibrary.rules
Rule Library: iriLibrary
Rule Matcher Catalog: RowGenFlow
Field Rule: SetRule
Field Rule Property: SET
Field Rule Property: NAME
Notes.html
representations.aird
RowGen_Test_File_Data [workbench-demos r
grades
scripts
set_files
Notes.html
Data Source Explorer
DEPT
DEPT_DIM
DM_COMPANIES
DM_DCTA_EVALUATIO
DM_EMPLOYEE_EVALU
DM_EMPLOYEES
DM_JOB_TYPES
DM_ORGANIZATIONAL_UNIT
EMP
Remote Systems
6 items selected
*saved-example
*Entity Relational Diagram
DisableRelatedFK.sql
COMPANY_NUMBER_UNIQUE.rcl
OU_CODE_UNIQUE.rcl
JOB_CODE_UNIQUE.rcl
EMPLOYEE_ID_UNIQUE.rcl
SSN_UNIQUE.rcl
EE_SEQ_UNIQUE.rcl
Wait_for_Sets
SCOTT_DM_COMPANIES.ctl
SCOTT_DM_ORGANIZATIONAL_UNITS.rcl
SCOTT_DM_ORGANIZATIONAL_U...
SCOTT_DM_JOB_TYPES.rcl
SCOTT_DM_EMPLOYEES.rcl
SCOTT_DM_EMPLOYEES.ctl
SCOTT_DM_EMPLOYEE_EVALUATIONS.rcl
SCOTT_DM_EMPLOYEE_EVALUAT...
EnableRelatedFK.sql
Data
Generate DDL...
Refresh F5
IRI
New Pseudo Set File from Column...
New ETL Join Job...
New Data Consolidation Job...
New FACT Multi Table Reorg Job...
New Multi Table Migration Job...
New Database Test Data Job...
New Multi Table Masking Job...
New Database Profile...
New ER Diagram...
New Data Classification Setup...

Status	Operation	Date	Connectio...
Failed	"SCOTT"."CUSTOMERS_REP"	Apr 22, 2019, 11:44:00 ...	Oracle
✓ Succeeded	"SCOTT"."CUSTOMERS_REP"	Apr 22, 2019, 11:44:39 ...	Oracle
✓ Succeeded	CREATE TABLE CUSTOMERS_OUT(SSN DECIMAL(9...	Apr 22, 2019, 11:46:07 ...	MySQL
✓ Succeeded		Apr 22, 2019, 2:22:57 PM	Oracle

```
Starting run
SELECT * FROM "SCOTT"."DM_COMPANIES"
Run successful
```

COMPANY_NUMBER	COMPANY_NAME

Windocks supports the implementation of IRI Voracity projects during the initial image build, as well as at “run time” depending on user needs. A sample joint build command would simply be:

```
FROM mssql-2019
SETUPCLONING FULL Windocks C:\Windocks\DBBackups\Windocks.bak
RUN C:\Windocks\IRI\Voracity\RunBatch.bat
```

Selecting a Database Virtualization Solution

Evaluating and selecting database virtualization systems varies according to organizational needs. Some need only SQL Server support, while others need broad database support. Most require multi-database images, while some are satisfied with single database cloning.

The following evaluation criteria can be weighted according to organizational needs, to identify and value differences between different database virtualization solutions.

1) **Volume or VHD cloning, or both?** Database virtualization requires on-premise or cloud volume, object, or file storage. Historically, solutions involved expensive volume storage. Windocks supports VHD cloning and also automates volume cloning with Pure Storage, NetApp, Cohesity, and other storage arrays. An open solution untied to vendor-specific storage, or limited to one type of storage, improves flexibility and affordability.

2) **Database platforms.** Some organizations need a range of databases, while others require only PostgreSQL, or SQL Server, making this an easy criteria to evaluate.

3) **Heavy client vs client-less.** Client software or agents installed on target database instances adds complexity to the database virtualization system, increasing instance maintenance and need for staging servers, and precludes container support. Cloud native and modern enterprise servers operate without client-side agents.

4) **Restful APIs.** Industry standard REST APIs are preferred over scripting, and support a broad range of programming, including Curl, Powershell, C#, Java script, and others. Systems with REST API support are favored by DevOps and automation engineers.

5) **Multi-database environments.** Some organizations are satisfied with individual database cloning, while most require images that can scale to deliver a handful or even dozens of databases.

6) **Directory and file cloning.** Not all systems support file and folder/directory level cloning.

7) **Containers, conventional instances, and workstations.** Application DevOps have adopted containers and industry standard APIs for automation and interoperability with Kubernetes, Jenkins, and other services. Identical containers are created in seconds, and simplify team collaboration and bug reproduction. A single VM can host up to 100 simultaneous containers and virtualized databases, and drives efficiency and infrastructure consolidation. Use of containers typically results in a 5-10x reduction in the number of VMs and standalone database instances.

   While containers deliver speed, automation, and testing efficiency, most organizations favor support for both containers and conventional instances, and workstations. Flexibility to support a complete range of runtime environments is important.

8) **Near real-time database cloning.** Delivery of production databases within minutes is important for production support and debug. Some virtualization systems are limited to images built from full backups, and cannot deliver near real-time database environments.

9) **Data cleansing, masking, subsetting, and synthesis**. Industry standard metadata and APIs simplify the integration of IRI test data engineering, including data quality, de-identification, and generation, enabling customers to build a best-of-breed test data management solution.

10) **Pricing.** Reseller and end-user pricing are important considerations too. The pre-discounted Windocks-IRI bundle provides partners and their customers the freedom to deliver compelling price advantages over competitors in data classification, masking, integration, migration, wrangling, replication, and virtualization, all while growing revenues and savings.

These criteria capture key considerations, but are not exhaustive. Other considerations include role-based access controls, audit logs, and immutable images that can be archived, audited, and restored. Real time visibility to database environments, and automated image creation and updates are also important. Avoiding complexity with staging servers is also important, as are dashboards and real time feedback.

Tip: avoid staging servers.

Most TDM systems require dedicated "staging" servers to deal with backups, or to create/apply masking.

Comparing Database Virtualization Solutions

Delphix

Delphix version 6.0.5.0 was released in October 2020, and has evolved over the past decade from UNIX System V to being a fork of the Illumos project (1). The Delphix OS is a virtual appliance available on VMware, AWS, Azure, and GCE, and Hyper-V, and delivers database virtualization with Delphix OS volume based cloning. The system architecture involves staging servers, and recommended configurations start with 16 vCPU cores, and 122 GB of RAM (2).

Public pricing on the AWS Marketplace for 10 TB starts at $34.00/hour or $293,000 annualized (3). The system requires target server connectors, also referred to as a “toolkit”, and either NFS or iSCSI connections (4). Data masking is available as a separate product, starting at $5.14/hour on AWS for 3 TB, or $44,000/year (3).

Links:
1) https://github.com/delphix/delphix-os
2) https://docs.delphix.com/docs/deployment/standard-deployment-architecture
3) https://aws.amazon.com/marketplace/seller-profile?id=44cc8643-3ad6-4209-b326-9e92a9a9621b
4) https://docs.delphix.com/docs53/delphix-administration/unstructured-files-and-app-data/data-management-toolkits-an-overview/design-build-and-install-a-toolkit

Windocks-IRI

Windocks version 5.X is based on Docker's container technology and API, and features a Restful API, and is an open platform supporting any volume, object, or file system storage, for any database. Windocks does not require client software or agents, and supports volume based cloning with any storage array including Pure Storage, NetApp, and others, as well as Windows Virtual Hard Drives. Windocks VHD cloning includes incremental updates with transaction log backups, for near real-time production database cloning (1, 2, 3, 4).

The entry-level USD $99K annual cost of a Windocks-IRI bundle includes 10TB support and give Voracity-executing hostname licenses (5), along with unlimited deployment of IRI Workbench and all base Voracity components and capabilities (6).

Links:
1) https://windocks.com/test-data-management
2) https://windocks.com/lps/doc
3) https://windocks.com/lps/docsan
4) https://windocks.com/lps/restapi
5) https://www.iri.com/products/voracity/platforms-pricing
6) https://www.iri.com/products/voracity/technical-details

Evaluation summary

Windocks is the clear technical winner in virtualization and includes a compelling price advantage. It is also an easily downloadable solution that can be setup in minutes, and proven in just days. Finally, Windocks compatibility with existing enterprise infrastructure and IRI jobs makes it uniquely suited for "any new infrastructure."

For relative IRI advantages over Delphix and other data masking solutions, see this FieldShield page. For relative IRI advantages in test data synthesis (via RowGen), see this page.

If you are interested in learning more about Windocks or IRI, or want to run a free supported POC, register for a free pilot at www.windocks.com or www.iri.com.

	Delphix	Windocks
Open Storage?	No	Yes
Databases?	Yes - all	Yes - all
Client-less?	No	Yes
Restful API?	Yes	Yes
File/Folder Cloning?	Yes	Yes
Multi-database Environments?	Yes	Yes
Containers & Instances?	No	Yes
Real-time Cloning?	Yes	Yes
Third party masking, subset, and Synthetic?	No	Yes

Windocks

IRI Voracity
An Insatiable Appetite for Data