

IMPROVE DEVOPS

Quickly deliver relevant data to the right people to improve DevOps

CHALLENGE

As the number of projects for developing applications increases, organizations struggle with quality assurance, release management, and infrastructure support. Provisioning an instance of a production database for a development team can take days to weeks. This time lag from the point of request to the point of provisioning depends on the size of the database, the overhead of the process, and the availability of storage. This data drift can prevent developers from using current database schemas and data in their work, which can result in suboptimal results. These inefficiencies can be significant when processes cross over organizational boundaries, and for a large number of applications and development teams. Delayed projects can lead to delays in organizational acceptance and translate into missed due dates for releases of products.

Time lags in the overall process and testing against dummy data can introduce application issues, which can negatively affect customer satisfaction. Releasing applications directly into production may not eliminate such delays. The reason is that the processes for change management can experience similar delays as each request for change can trigger legacy processes for approval and provisioning.



DEVOPS

DevOps is a methodology for the development of applications. It emphasizes the collaboration between developers and infrastructure teams. DevOps strives for the rapid development of applications. It also aims for continuous integration through industry-accepted development methodologies, process management, and automation.

For application development teams, the goals for DevOps are self-service, point-in-time bookmarking, fast access, protection of data, and meeting due dates for development. However, organizations may face the challenge of improving automation to support the full lifecycle for the development of applications efficiently. In many organizational data environments, slow access to data and manual, time-intensive processes may still be prevalent.

When organizations are slow or fail to implement DevOps initiatives correctly, this can negatively affect their time to market, the growth of revenue, customer satisfaction, and customer retention. Consequently, organizations are looking for better ways to help them achieve organizational goals and to keep up with the changing organizational needs.

VIRTUAL DATABASES

A virtual database improves the accessibility and cost of providing the relevant people access to the appropriate data in the correct format. A virtual database employs advanced database virtualization technology and automation. It provisions quickly virtual copies of test data for applications while supporting large numbers of concurrent developers. A virtual database enables development teams to access virtual copies of any size database quickly. As such, it can improve service levels dramatically and cost-effectively without compromising performance or control. With a virtual database, keep development data current, provision rapidly, automate workflows, and enable self-service.

Minimize the impact on production databases by accessing the database backup files that database administrators create for availability and disaster recovery. Such backup files include backup set files, and full, differential and transaction log backups. Enforce data security by encrypting the backup files. Share the backup files with the development teams. Quickly gain read and write access to any point-in-time copy of the production database in the backup files by setting up a virtual database. Access the database schema and the data in a virtual database from any relevant software for application development including quality assurance. Refresh the virtual database using updated backups. Integrate a virtual database into automated workflows for continuous integration for the development of applications with its command line interface.

A unique recovery technology that allows backup files behave like regular physical databases underlies a virtual database. This technology creates virtual databases by directly attaching backup files to an instance of Microsoft SQL Server without requiring a restore operation. Access all objects and data in the virtual databases via T-SQL scripts, stored procedures, SQL Server agent jobs, and native, third-party and custom applications.

BENEFITS

With a virtual database, the lifecycle for application data improves significantly. Benefits for application development include faster self-service access to copies of production databases, read and write copies available quickly to authorized users, point-in-time access for faster reset, a quicker refresh of test databases with updates from production, reduced database administration, and workflow automation. These benefits result in shorter time to market by removing provisioning bottlenecks and thus shorter time to complete projects for developing applications.

IMPACT

A virtual database moves data management out of the data path. A virtual database transforms the way organizations manage their data and accelerate the development cycles of applications. It offers a simple, cost-effective implementation path for organizations currently struggling with issues concerning copy data.

Reducing the time to provision instances of production applications for development, and decreasing the lag between synchronizing live production instances and test instances can help address the challenges that organizations face concerning quality assurance, release management, and infrastructure support.



IDERA'S SOLUTION

IDERA's SQL Safe Backup contributes to DevOps by providing faster self-service access to copies of production databases, read and write copies available quickly to authorized users, point-in-time access for quicker reset, a faster refresh of test databases with updates from production, reduced database administration, and workflow automation. This results in shorter time to market by removing provisioning bottlenecks and thus shorter time to complete projects for developing applications. Its command line interface (in addition to its graphical user interface) enables automation and integration into DevOps workflows.

Start a free, fully-functional, 14-day trial today!

Start for FREE

The screenshot displays the IDERA SQL Safe web interface. The top navigation bar includes 'HOME', 'POLICIES', 'OPERATION HISTORY', 'INSTANCES', 'DATABASES', 'SQL SAFE AGENTS', 'VIRTUAL DATABASE', and 'ADMINISTRATION'. The user 'konoha|administrator' is logged in, with 'Administration' and 'Help' options available. The left sidebar shows a 'FILTERING' menu with expandable sections for STATUS, POLICY TYPE, POLICY NAME, DATABASES COVERED, INSTANCES COVERED, LAST OPERATION, LAST OPERATION WITH FAILURE, INSTANCE, DATABASE, and BY CUSTOM FILTER. The main content area is titled 'MANAGED POLICIES' and features a 'Policies' section with buttons for 'Add instance', 'Create policy', 'Edit policy', 'Copy policy', 'Properties', 'Remove/delete', and 'Export'. Below these buttons is a table with the following data:

<input type="checkbox"/>	Status	Policy Type	Policy Name	Databases Cov...	Instances Cove...	Last Operation	Last Operation With Failure
<input type="checkbox"/>	Succeeded	Log Shipping	Northwind Log Shipping	2	2	Thu Mar 09 02:53:11 GMT	
<input type="checkbox"/>	Wait	Backup	Full Backup Policy	3	2		
<input type="checkbox"/>	Succeeded	Log Shipping	AdventureWorks Log Shipping	2	2	Thu Mar 09 02:55:36 GMT	

At the bottom of the table, it indicates '3 total rows' and '10 items per page'. A pagination control shows '1 / 1'.