Liquibase

WHITE PAPER

How Database Release Automation Fits into the Cloud Application Toolchain

How Database Release Automation Fits into the Cloud Application Toolchain

The cloud has changed data management forever. It's easy to spin up new environments, allowing development teams to move fast. There is a catch, though. The proliferation of new cloud instances means that each new environment requires an additional cloud database instance that needs to be managed. With the increase in databases across distributed cloud instances, it's critical, yet much harder, to keep application and database changes in sync.

Managing Data in the Cloud

There's a lot to love about the cloud. Users enjoy on-demand access to applications and services. Business leaders can get apps to market faster, without worrying about underlying infrastructure costs and maintenance. The hidden challenge of the cloud lies in the ability to quickly scale cloud database services, like managing database deployments across an ever-increasing set of cloud database instances and enforcing database standards across different teams. These tasks take valuable time away from software development teams as they work on housekeeping instead of building compelling new features for customers.

Simplify Managing Cloud Infrastructure

The beauty of a database release automation tool, like Liquibase, is that it is system-agnostic. No matter where your databases reside, in private cloud, a datacenter, or public/hybrid cloud, such as Amazon, Azure, or Google, Liquibase automatically enforces rules and standards across all databases, ensuring bad changes are never deployed.

Cloud Applications

Whether software development teams are tasked with rearchitecting legacy applications to run on top of the cloud or building greenfield cloud-native applications, teams still face the challenge of the data layer. Managing the data layer is the most painful part of developing cloud apps since applications are stateless and the database remains stateful. This means that teams can't rely on application release automation or cloud developer tools alone to address database change and deployment. These automation tools don't track database changes. In order to minimize risk, database changes are reviewed by DBAs, slowing the whole process down.

Database Release Automation is Key to Unleashing the Full Power of the Cloud

One of the primary drivers of moving to the cloud is to go faster and cut maintenance costs. In order to truly realize the promise of the cloud, teams need to address the database change and deployment process.

Liquibase partners with many cloud technology vendors, such as Pivotal, to validate, automate, and track database changes through the release pipeline. Liquibase creates a simple, repeatable process that synchronizes with your cloud application releases so they take minutes instead of days or weeks. True continuous delivery!

Inviting the DB Team to the Cloud App Party

Developers have reaped the benefits of cloud application tools like Pivotal Cloud Foundry. They can't even imagine a world without automated builds, tests, and deployments of their application code. Sadly, the people who work on database code have not been invited to the party.

A recent report[1] reveals that manual database rework is the norm, even in organizations that have adopted DevOps. In fact, the faster the application release cycle, the more database professionals had to rework database changes. 93% of survey respondents reported reworking changes multiple times for daily or weekly release cycles.

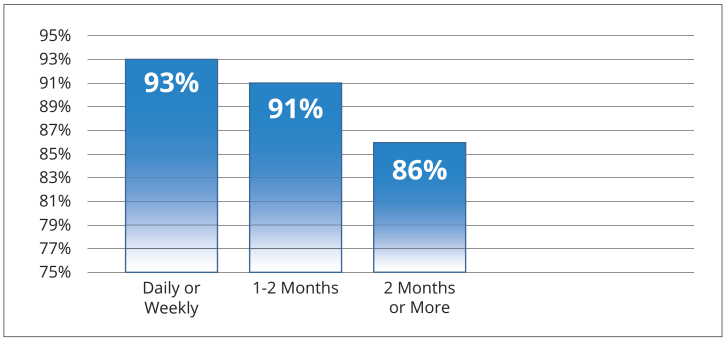


Figure 1: The percentage of respondents reworking database changes based on release cadence.

Given the amount of wait time and task switching database changes are forcing on these teams, applying DevOps principles makes perfect sense. Developers and database professionals need an effective process and the right tools to address this costly problem.

Database Release Automation

Implementing a fully automated, continuous appli-cation delivery process requires ALL code, including database code, to be checked in to version control and deployed as part of the application release process.

[1] 2019 State of Database Deployments in Application Delivery. Dimensional Research. https://www.liquibase.com/resources/reports/database-deployments-2019?utm_source=white-paper&utm_medium=pdf&utm_campaign=wp-2019-survey-state-of-deployment-in-application-delivery

Database release automation (DRA) refers to the process of packaging and deploying database changes for an application from development, across various environments, and ultimately to production. Best practices dictate that DRA solutions combine deployment automation, environment modeling and release coordination.

If your company isn't using a database release automation process yet, you're not alone. Many companies that implement continuous delivery for their application code aren't doing the same for their database code. Yet.

More companies are discovering that DRA eliminates bottlenecks and speeds application deliveries.

A Growing Sea of Cloud Platform Development Tools

There's a growing sea of tools that need to integrate and fit together in the cloud application toolchain. It's vitally important that you understand how any new tool, no matter how amazing and helpful it seems on paper, fits in. With so many variations in toolsets and team make up, there's no one size fits all.

A Typical Cloud Application Release Process

In order to understand how Liquibase's database automation tools fit into cloud application releases, first, let's set the stage with a typical cloud application release process that we see from customers and the tools they are leveraging.

Quick Stats on Database Changes

84% of application stakeholders had serious production issues due to database change errors.

57% of all application changes require a corresponding database schema change.

88% report taking more than an hour to resolve these schema change issues.

Benefits of DRA

Faster application releases

Worry-free database deployments

Painless database audits

Happier, more productive teams

Serenity Co. currently uses the following tools in their cloud application development and release process:

- Visual Studio for application code and database code development
- Git for source control
- ServiceNow for IT ticket tracking
- Jenkins for CI/CD
- Artifactory for artifact repository
- Pivotal Cloud Foundry for application development and deployment to the cloud

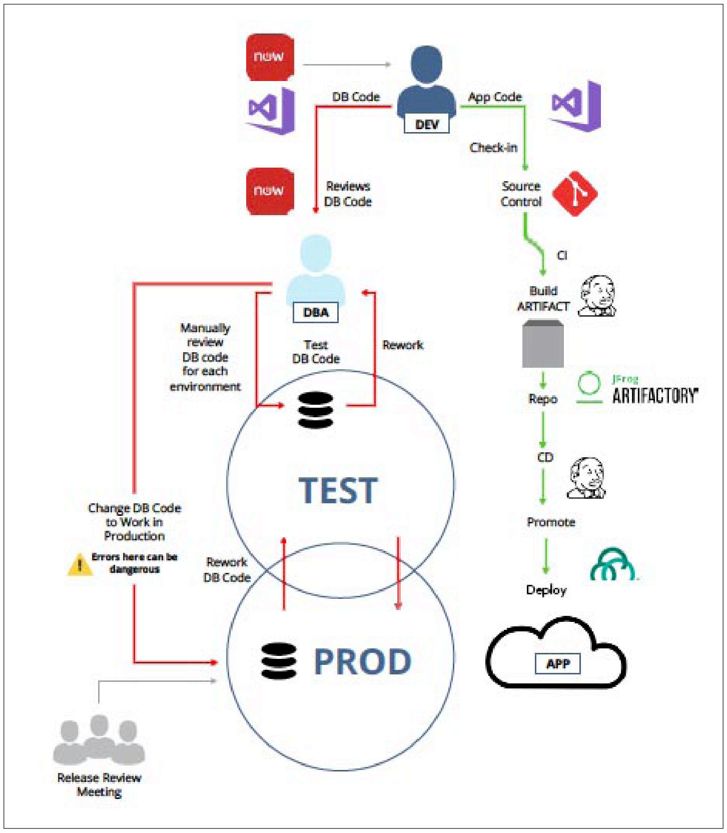
On the application code side, they are using automation. Although their process is faster than it was last year, they're experiencing a bottleneck with database changes.

Serenity Co. knows that database code is different than application code.

Different Environments and Manual Work

For Serenity Co, the same database change script won't work in each environment because each database environment is different. They must make their database code very specific to different cloud environments.

When developers write their code, they can just drop it into any app server and it would work. This isn't the same with database code. It is very common that a database change script will deploy successfully in a DEV environment, but then it won't work in Test and may not work in Production. The environments are often not in sync. This happens because people are making database changes manually, often without a defined process or tooling to support a modern database process.



This makes life difficult for operational DBA teams. They have to take the time to write custom scripts for each environment. This rework takes extra time. It also makes it challenging to know and track exactly what was run in different environments as these reworked scripts and manual tweaks often don't find their way into source control or change management systems.

When You Do Things Manually, Bad Things Happen

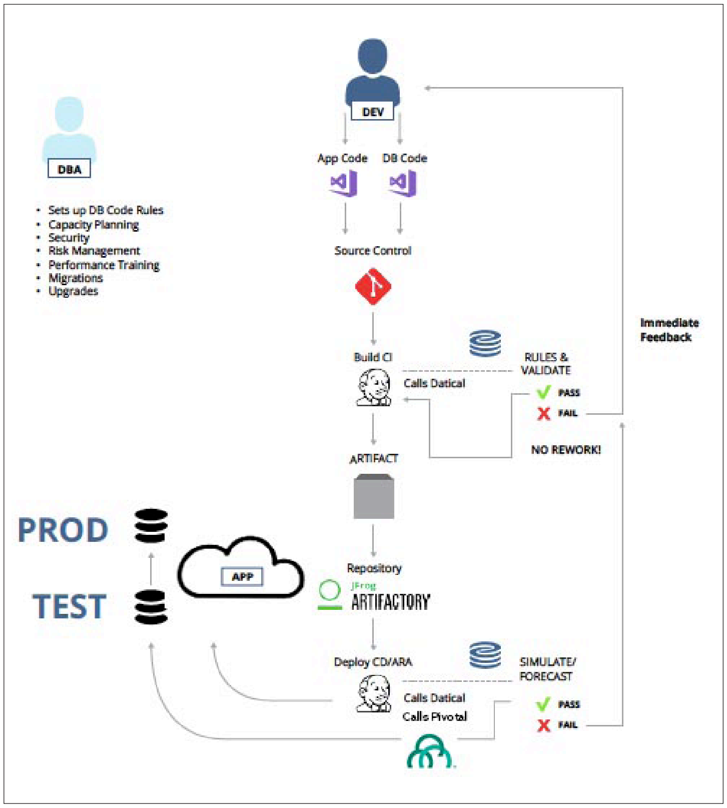
When Serenity Co. pushes a bad database script through, it can drop thousands of rows of customer data. They do backups so they can recover. A bad database change can be much more destructive than an app code change. This is why they haven't considered automating the database yet. They think it's just too risky.

The Application Release Process to the Cloud with Liquibase

Now there is a way to eliminate risk and automate database changes and database releases.

Serenity Co. can continue using all the tools in their application release process. Liquibase just adds smart automation to their current manual processes of database changes.

- **Visual Studio** for application code and database code development
- **Git** for source control for BOTH application code and database code
- **ServiceNow** for IT ticket tracking (with way fewer tickets for database errors and rework)
- **Jenkins** for CI/CD for BOTH application code and database code
- **Artifactory** for artifact repository for BOTH application code and database code
- **Liquibase** for database change and release automation
- **Pivotal Cloud Foundry** for application development and deployment to the cloud



With Liquibase, Serenity Co. is able to produce twice as many application releases as they did before. Their developers aren't waiting around for the database code they wrote a week ago to blow up. Instead, the developers get automated reports on their database code once it has been committed to source control and evaluated by Liquibase. Their DBAs are no longer constantly cleaning up bad scripts. Instead, they're focused on important tasks, like performance tuning, database cloud migrations, security, and all of the other important things the DBA team needs to do besides reviewing and deploying SQL scripts.

Want your company to reap the benefits of database release automation?

Sign up for a 1-1 demo. We'll show you exactly how your team can apply DevOps to your database process.

Schedule a Demo >

Liquibase

www.liquibase.com | info@liquibase.com