SUCCESS STORY

Multinational Bank achieves Security Compliance using DROPS to Deploy across 'Segregated' Hybrid Clouds

One of the largest banking and financial services corporations in Canada sought to deploy multi-platform in a high security environment

Challenge

The digital banking division operates in a **diverse IT landscape**, including IBM i (aka iSeries, AS/400) as database server, a core RPG application, and a front-end UI developed in .NET in Microsoft Azure.

Deployment on IBM i was slow and difficult, as teams struggled with traditional change management tools and a highly fragmented separation of duties. For security reasons, an external company was employed to promote release packages, while another set of internal users performed the actual install. This externalized operation was costly and delayed the delivery of new software features.

Meanwhile, on the .NET side, all front-end work was full CI/CD using Azure DevOps.

DevOps Transformation

The core banking application on IBM i is highly strategic. So, to modernize methods and rationalize costs, the architecture team needed to bring IBM i development into the same Azure DevOps pipeline methodology as their front-end apps. The goal was to use **existing cloud-based Azure tools to pilot all deployment,** spanning **native IBM i objects, .NET code, and a Java application** residing in the IFS (Integrated File System).

Hybrid Cloud & Security

All IBM i systems at the bank run on premise in the same data center, while the Azure front-end is managed and served by cloud-based resources.

To comply with strict security regulations, the private environments (development, test, and pre-production) had to be isolated from the customer-facing production environment.

So, in a true hybrid cloud infrastructure, the bank configured two **independent and isolated cloud instances**: a private development cloud and a partially public production cloud. The IBM i development server makes use of a cloud-based development tool stack, while the IBM i production server presents a cloud-based UI to customers via a public interface.

The development IBM i server has no connection to the outside world, and no communication is permitted between the two cloud instances, except through a highly controlled enterprise-level mechanism.

At the start, this **strict network isolation presented certain challenges** during the deployment phase of the application lifecycle.

Deployment between 'segregated' cloud environments

Complete network segregation was facilitated using ARCAD's DROPS. To deploy from the development cloud to the production cloud, DROPS manages a deployment package (in the case of IBM i, a savefile) containing tested application objects. DROPS retrieves this package from Azure Artifacts and pushes it out to the production cloud, from where it is deployed on demand from Azure out to the production server.

As no tool could span both development and production environments, Azure Artifacts is replicated on each environment and a separate DROPS server interacts with each of the Azure Artifacts instances.

An Azure pipeline now automates the build, packaging and deploy of the savefile from Dev to QA, populating the Azure Artifacts instance in Dev. A second Azure pipeline in the production domain then uses the replicated Azure Artifacts instance together with ARCAD metadata repository knowledge to validate the incoming release and activate the deployment to production.

This ensures a universal, secure, and compliant deployment process: **all RPG, .NET, and Java components are deployed synchronously across isolated cloud environments, with automated rollback on error.**

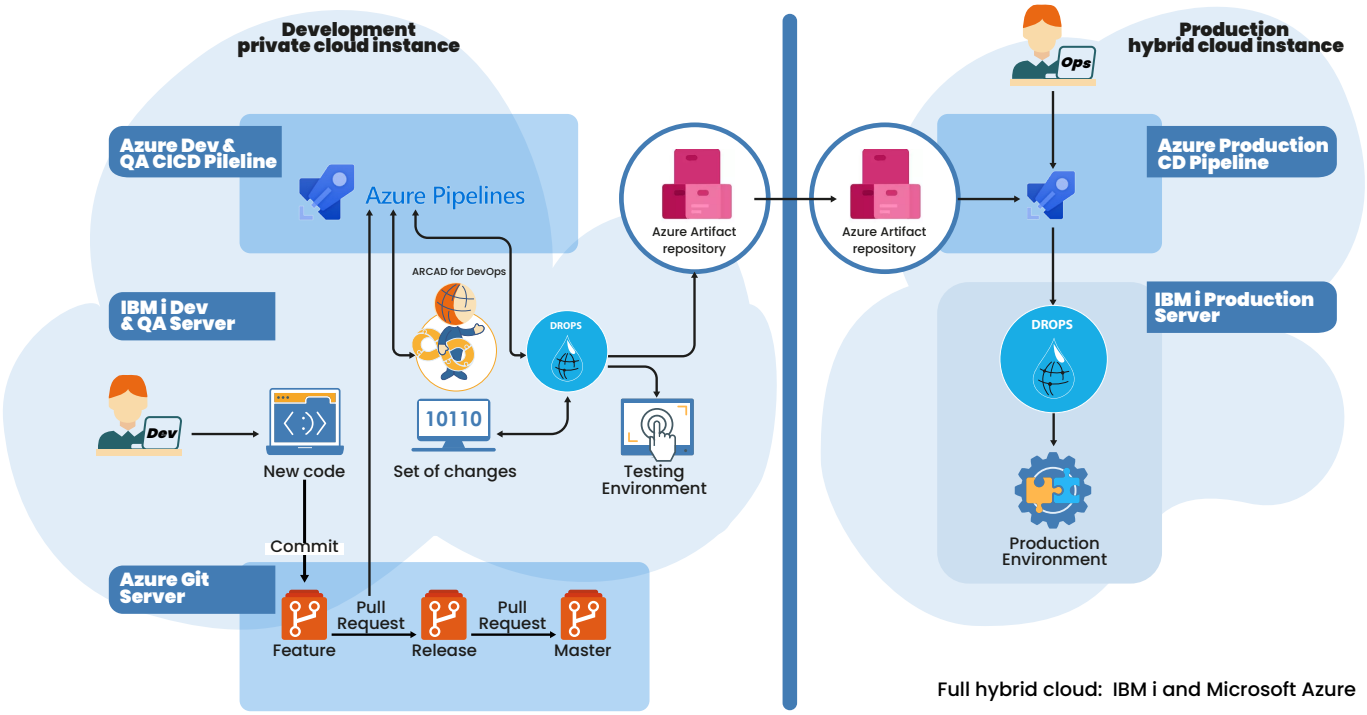
Change tracking and Compliance

When integrating IBM i within Azure, a key challenge was the ability to capture a deployment package and identify it correctly.

DROPS captures the unique package identifier externalized from metadata exposed during the development process to ensure that objects deployed to production are the **exact same objects that were tested in the QA environment**. This identifier then 'travels' with the changed objects, enabling change tracking throughout the application life cycle and out to production.

This means the **entire lifecycle can now be tracked from Azure DevOps.** It is easy to view which developer made the change request in the first place, against which work item in Azure Boards, and even which specific source lines have been changed.

This gives a high level of **transparency and accountability of changes**, with clear **segregation of duties and visibility** at all levels. Auditors can be provided with complete lifecycle information about changes that were made months ago.



Full hybrid cloud: IBM i and Microsoft Azure

"The progress we've made is incredible. The team's feedback has been all positive. We now deploy ALL artifacts securely across our hybrid cloud environment. ARCAD's DROPS has led to a reduction in work, and time savings", Development Manager, Digital Banking