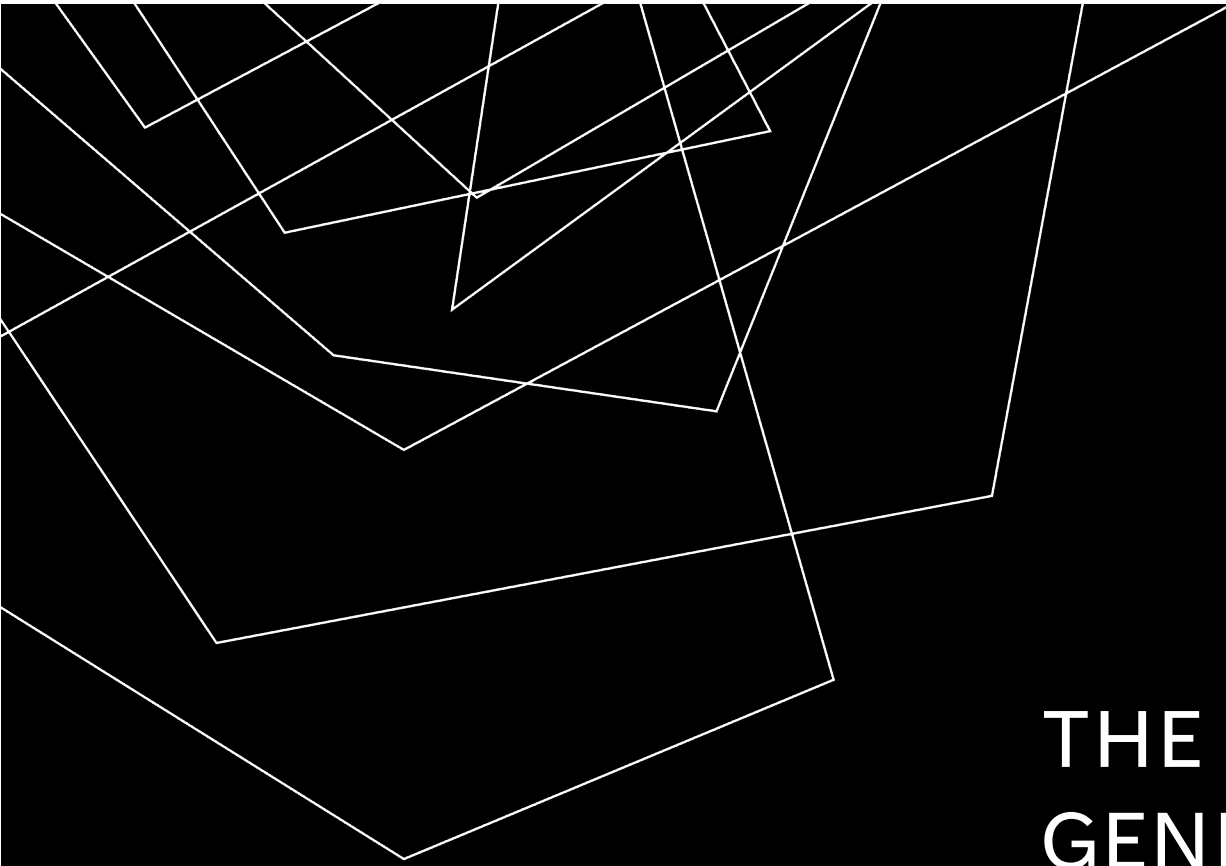
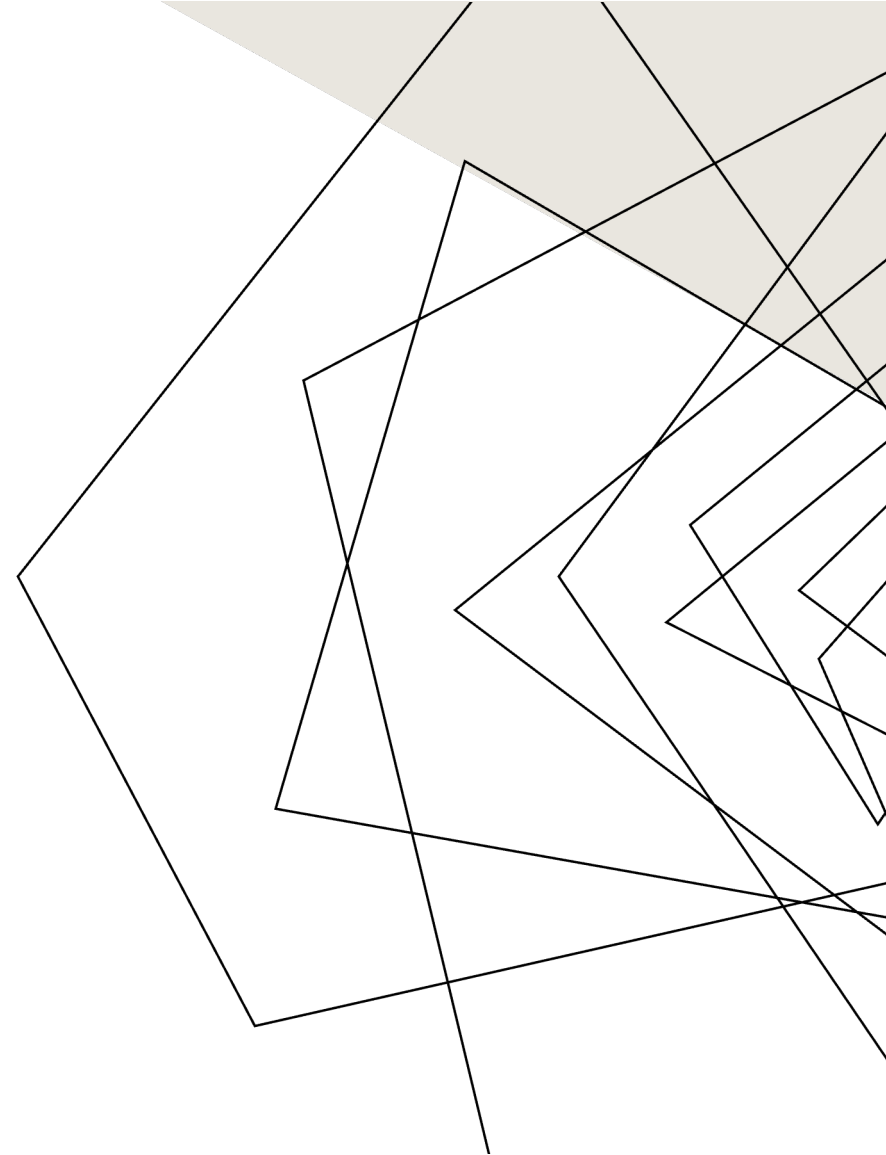


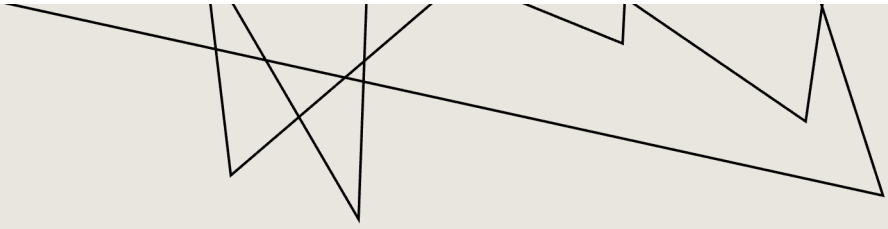
An abstract graphic consisting of several thin, white, irregular lines that intersect and overlap on a solid black background. The lines form a complex, geometric pattern that resembles a stylized, fragmented shape or a network of connections. The pattern is located in the upper left portion of the slide.

THE NEXT GENERATION OF CONTAINERIZED DB2

AGENDA

- Brief history of containerized Db2
- Motivation for the next generation
- Table stakes
- Exceeding expectations
- Beyond



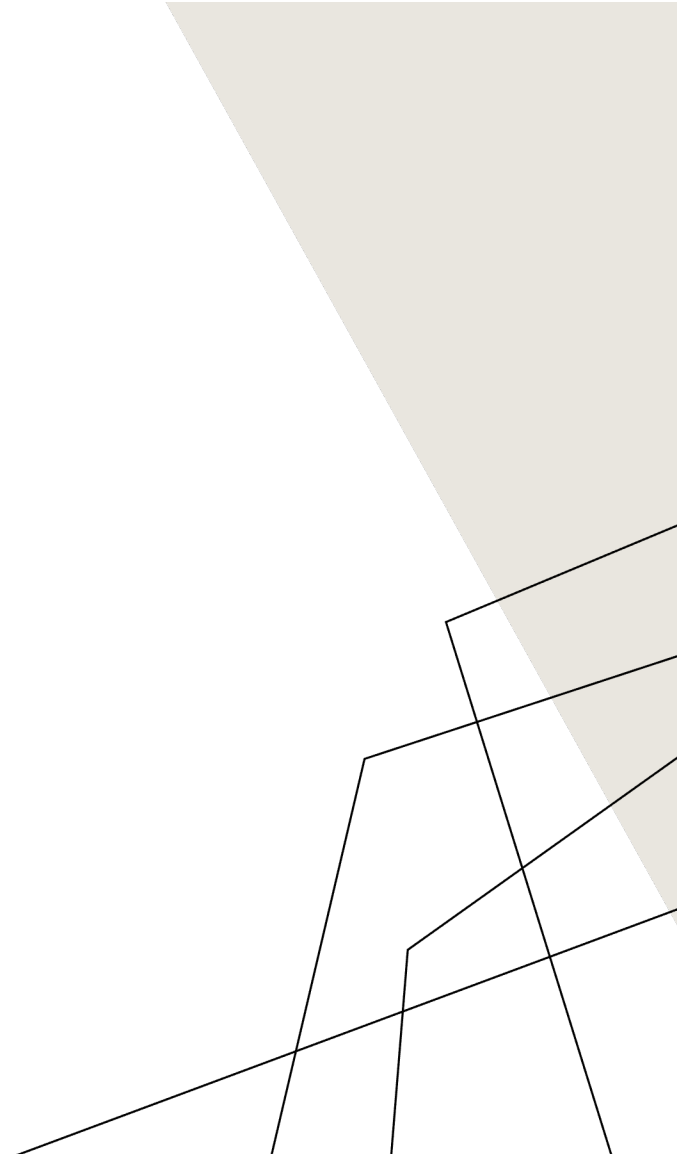
Abstract geometric lines in the top right corner, consisting of several intersecting black lines forming a series of triangles and polygons.

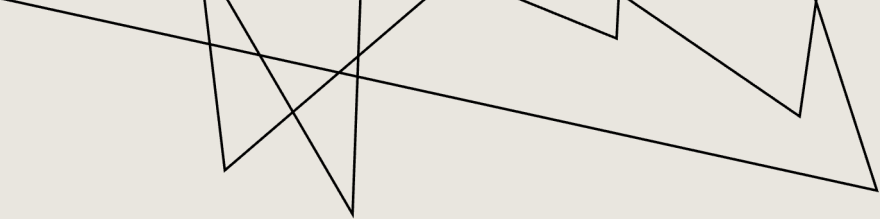
A BRIEF HISTORY

Who are we?

DB2 CONTAINERIZATION TEAM

- Docker released in 2013
- A crazy idea is born: can we stick Db2 into a container?
- Project “BluHelix” started
- Various developers with different skillsets brought together
- IBM’s “dashDB” deployed with IBM UCD
 - Lots of automation, lots of redundant overhead!
- Move this automation to a Dockerfile
 - Build once, deploy often!
- IBM takes its first major step into the container space: dashDB local/IBM Db2 Warehouse



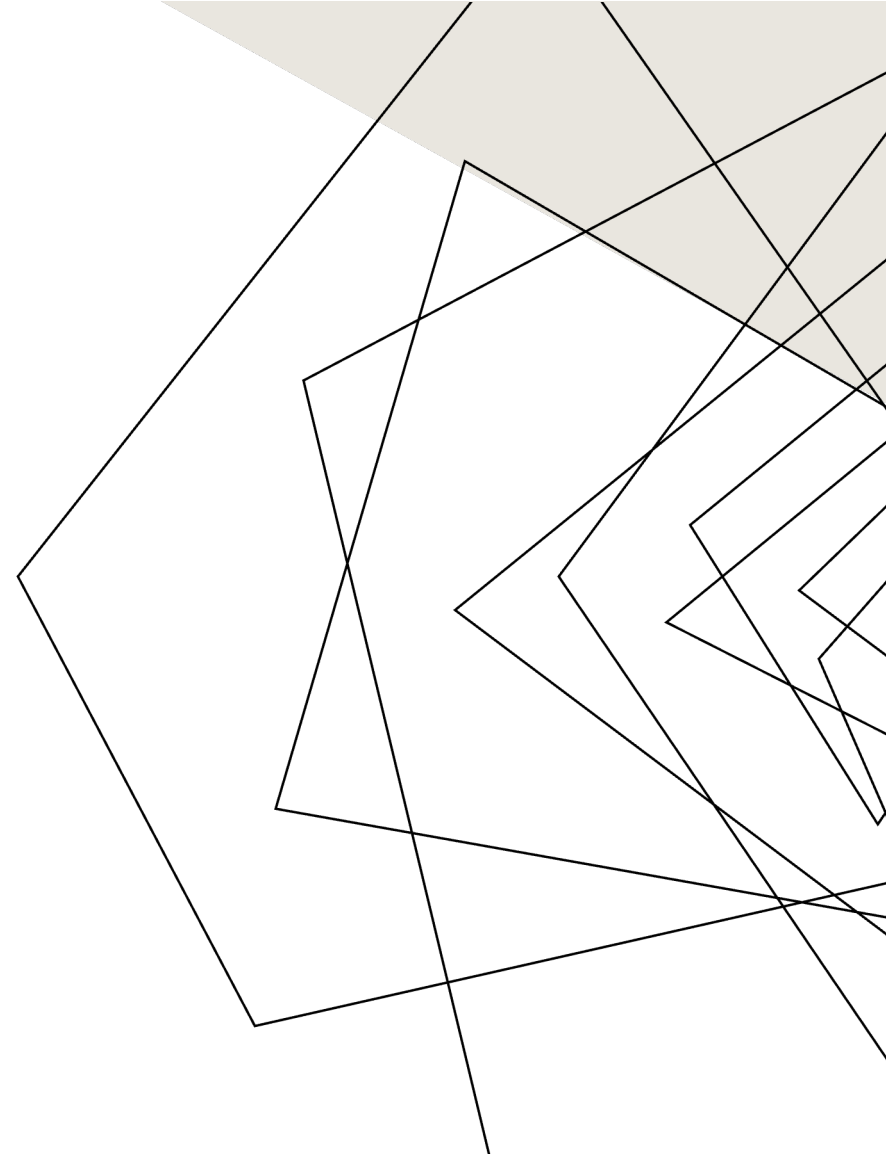


A BRIEF HISTORY

How did we get here?

A HUGE SUCCESS!

- Db2 Warehouse proves to be a hit with customers
- Containers gain widespread adoption in the industry
- Customers are demanding: they want more from Db2 Warehouse
- Our image is getting massive
- Enter Kubernetes and Cloud Pak



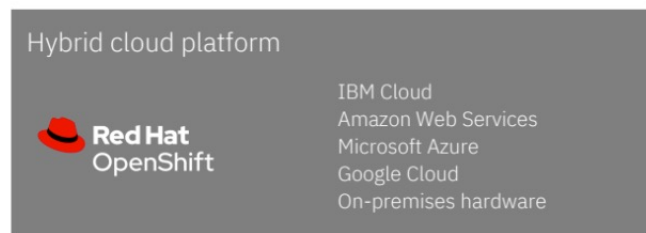
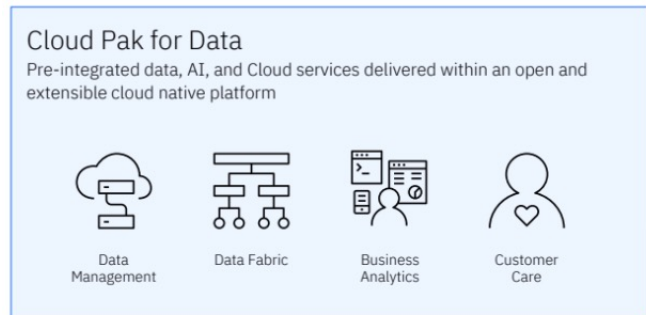
MARKET TRENDS...

- Kubernetes and microservice architecture gain massive popularity
- Us: single container, single image
- This is a natural progression!
- Design containerized Db2 for the Kubernetes environment
- This lines up perfectly with IBM's partnership with RedHat



IBM CLOUD PAK FOR DATA

- Debut the new architecture in IBM's latest platform
- We provide a critical path for both customers and IBM services
- Adoption grows, services integrate
- There is demand outside of this use case!



MEETING CUSTOMER DEMAND

- Just give me Db2!

Platforms



Cloud Pak
for Data



OPENSIFT

RHOS Container
Platform



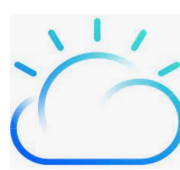
Native
Kubernetes



AWS EKS/ROSA



Azure AKS/ARO



IBM Cloud



IBM Power
Infrastructure
(P10 CR)

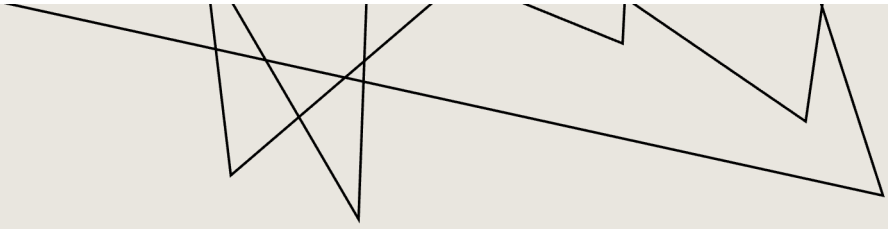
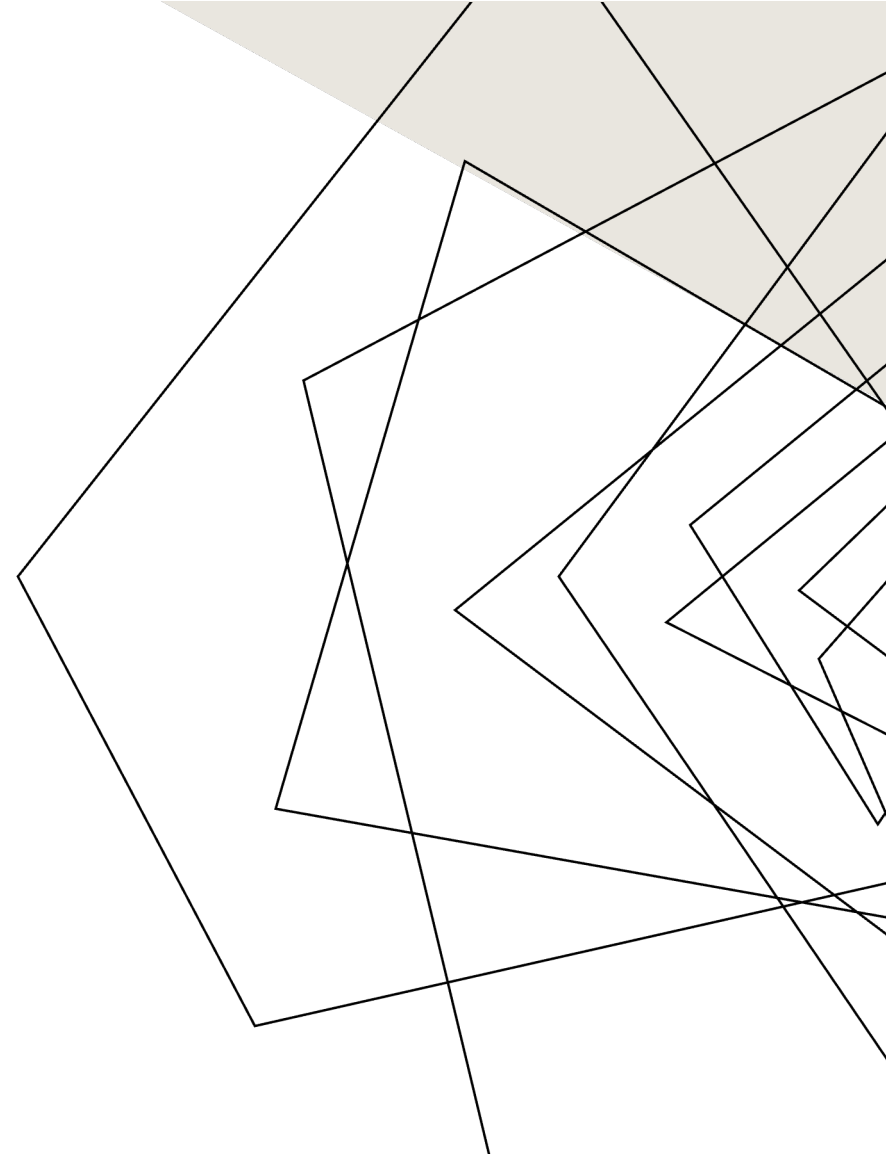


TABLE STAKES

Next Gen Non-Negotiables

OUR PRIORITIES

1. Install & Deployment 
2. Update/Upgrade 
3. Security 
4. Resiliency 
5. Usability 



1) INSTALL & DEPLOYMENT

- Move to helm-based install of our operator
- Overrides
- Templates
- GitOps





2) DEPLOYMENT

- **Db2uInstance**

- Next gen uses `Db2uInstance` custom resource (CR) for deployment
- **Numerous enhancements** to `Db2uInstance` CR in Next Gen

- **Enhancement examples**

- Instead of specifying the `version`, in next gen the Db2 image is taken as input and the version is specified in the tag of `spec.image`.
- Annotation `db2u.databases.ibm.com/maintenance` for disruptive operations
- Integrate with Kubernetes-native certificate manager
 - Specify the certificate secret in the `ssl` section of the CR
- Next Gen uses GSS Plug-in for AD/LDAP auth
 - LDAP settings provided via configmap `db2u-ldap-settings` in section `IBMIAMauthldap.ini`.
 - <https://www.ibm.com/support/pages/setting-ldap-db2u-next-generation>



2) DEPLOYMENT

- **Semantic versioning**

- Next gen follows Db2 semantic versioning scheme for the container images.
- Image format is:
 - `spec.image: icr.io/db2u/db2u.<DBTYPE>.VRMF-CSB-architecture`
- Where V is *version*, R is *release*, M is *Modpack*, F is *Fixpack*.
- For example:
 - `icr.io/db2u/db2u.db2wh:v12.1.5.0`
- Versioning scheme allows separate upgrade of the operator and Db2U containers, within limits/ranges enforced by the webhook.

- **Independent Image & Operator Updates**

- Db2 images can be updated independently of the operator, enhancing security and simplifying upgrades.
- Operator updates are only required for new features or security fixes affecting the operator.



DEMO

Install and Deploy

3) USABILITY

Highlights:



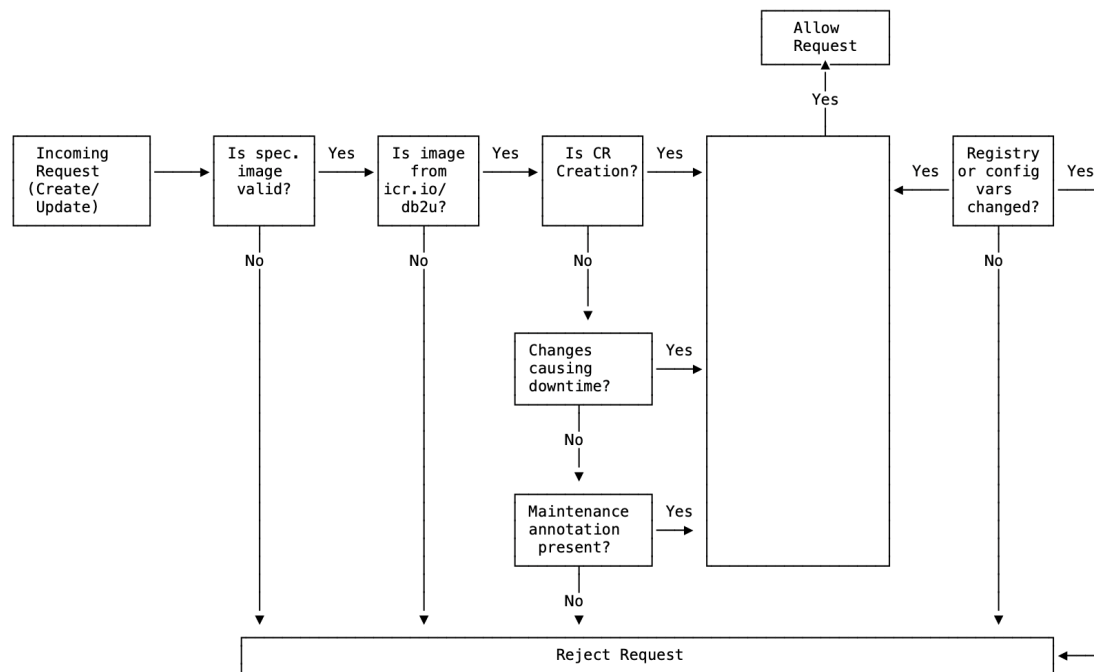
Category	Current Gen	Next Gen
Admission Webhooks	NO CR validation	CR validation, prevents misconfiguration and enforces supported setups
Prechecker	NO day-0 config validation	Precheck validates configuration as step 0 of deploy sequence
Diagnostics	Longer lead-time to detect failure and triage due to sparse operator logging	Clearer and meaningful operator logging

3) USABILITY – ADMISSION WEBHOOK

- **What is it?**
 - Kubernetes admission controller (*validating* webhook and *mutating* webhook)
 - Intercepts requests to Kubernetes API server before action is taken
 - Allows validation of the incoming requests
 - Rejects if do not meet certain criteria or support statement
 - Enforcing supported features and configs
 - Ensures only valid instances are created or updated
- **Validates the image provided in the Db2uInstance CR**
 - checks the `Db2uInstance` CR to ensure the image specified in the `spec.image` field is compatible with the db2 operator installed
- **Enforces that a Maintenance annotation is set**
 - Before allowing disruptive operation e.g. scaling, upgrade
 - Checks that `db2u.databases.ibm.com/maintenance: "true"` annotation is set



3) USABILITY – ADMISSION WEBHOOK



3) USABILITY – THE PRECHECKER

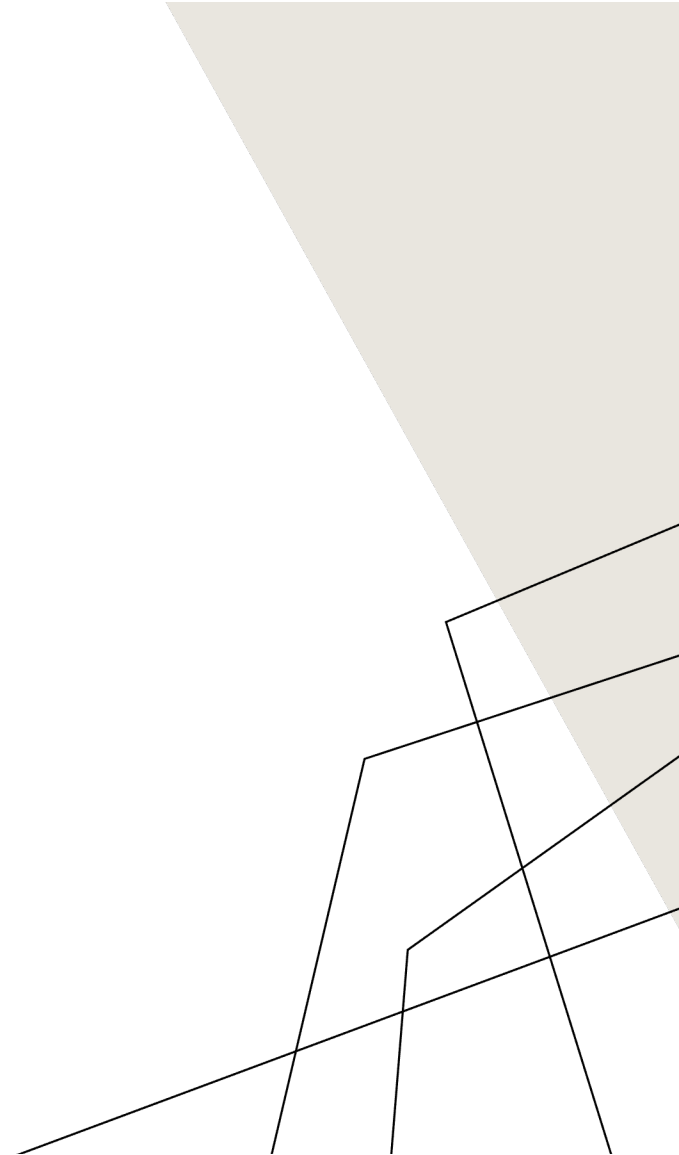
- **Why the prechecker?**
 - More than 50% cases from install, upgrade and BnR
 - Many of those are due to misconfiguration / storage
- **What does it do?**
 - Prechecker can validate 6 sensitive points in 40s
 - Automatically adapt to the deployment environment
 - Saving 20min-30min against a manual diagnostic
- **How does it integrate in next-gen?**
 - Run as a step 0 when a db2u pod is created
 - Prevent unhealthy deployment to process / report warning

```
INFO: Executing storage precheck...  
WARNING: [NFS_VERSION] - NFS is configured with a version which is not fully supported: 4  
INFO: [WRITE_SPEED] - Storage write speed: 79.3 MB/s (BS=1M count=1024)  
INFO: [WRITE_SPEED_LIGHT] - Storage write speed (lite): 3.3 MB/s (BS=4K count=1024)  
INFO: [WRITE_SPEED_FIO] - Storage IOPS speed read: 28.223512, write: 30.044384  
INFO: Storage prechecks executed in 36.481964s  
INFO: Prechecker passed
```



4) RESILIENCY

- HA within container
 - Replacing Wolverine with a new framework
 - Based on a streamlined runit daemon and native Kubernetes probes
 - Removes dependency on ETCD
 - Node level recovery
 - Improved recovery times from planned and unplanned outage events
- HA with replicas
 - Copy/paste from slide 24



5) UPDATE

1. Update the operator

- `helm upgrade db2-operator ./db2-operator -f overrides.yaml --set image.tag=1.0.1`

2. Set maintenance mode

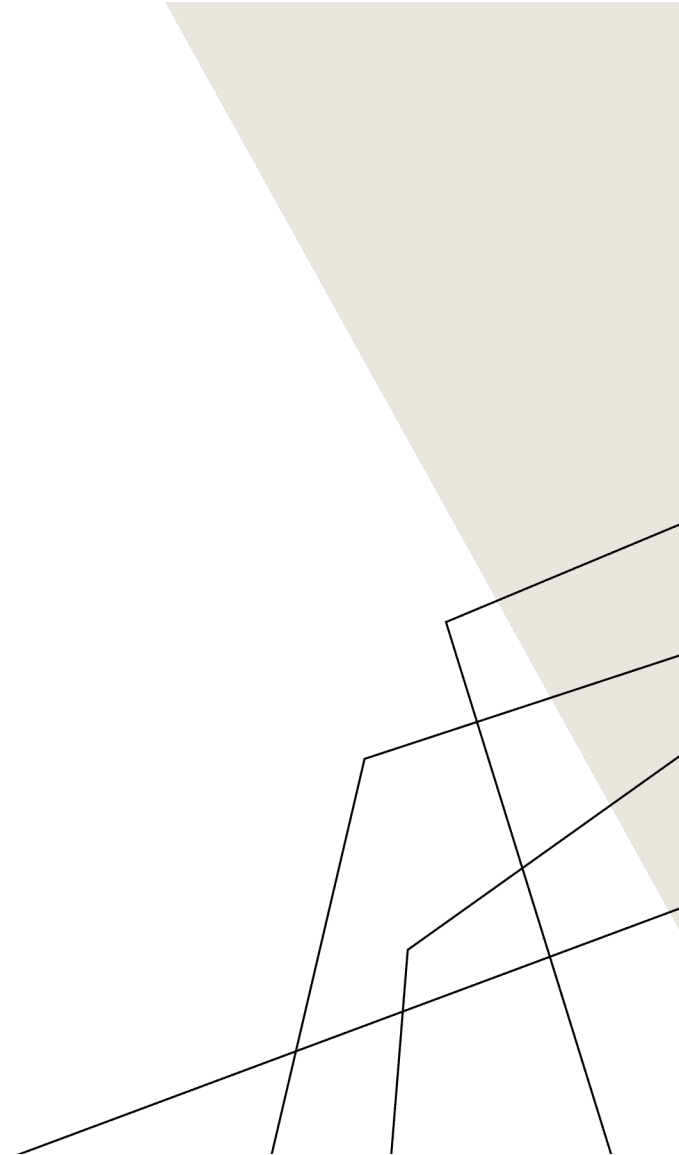
- `oc annotate db2 db2-wh-next-gen db2u.databases.ibm.com/maintenance=true --overwrite`

3. Update the image in Db2uInstance

- `oc patch db2 db2-wh-next-gen --type merge -p '{"spec":{"image":"icr.io/db2u/db2u.db2wh:12.1.6.0"}}'`

5) UPDATE – WHAT’S HAPPENING?

1. Operator reconciles
 1. CR status changes to "updating"
2. Instance Shutdown
3. K8s resources reconverged to the new version
4. New pods with new version come up
5. Operator triggers Db2 update procedure
 1. On Success: "Ready"
 2. On Failure: "UpdateFailure"



6) SECURITY

What's changed?	Improvement
Prior used <i>minimal</i> UBI, next gen uses <i>micro</i> UBI	Reduced attack surface, minimized image size, enhanced performance
MPP hostIPC requirement – removed	Enhanced node-level security and isolation
OLTP - ssh - removed	Improved network security & reduced vulnerability exposure
SCC + accountability	Restricted-v2 scc supported OOTB
Namespace-scoped resources with Helm	Enhanced resource isolation, permission management, and easier maintenance
Reduced number of images	Reduced surface area & vulnerability
Leveraging K8s Native Security features (secret, cert managers)	Reducing direct exposure and centralizing credential security
Resource reduction	25% reduction in attack surface exposure

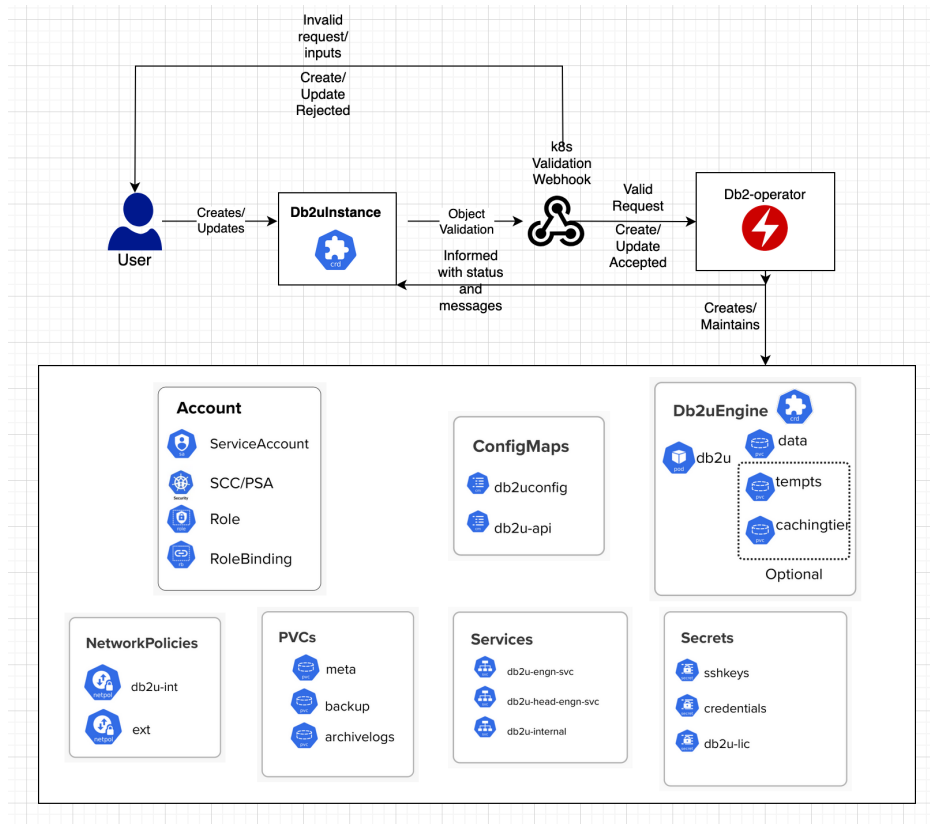


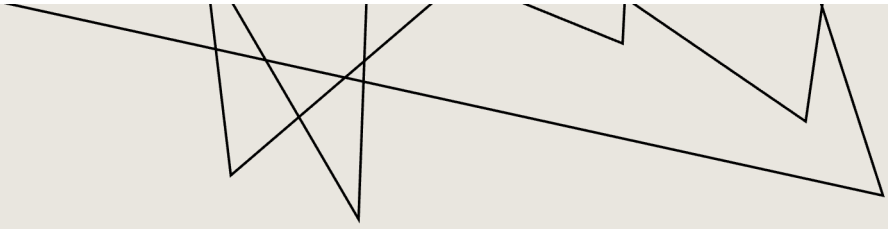


6) SECURITY – KYVERNO POLICY COMPLIANCE

- **Validate**
 - Blocks or audits resources that don't meet your rules (e.g., "No root containers").
- **Mutate**
 - Automatically edits resources to add defaults (e.g., injecting sidecars or labels).
- **Generate**
 - Creates new resources automatically (e.g., setting up a NetworkPolicy when a new namespace is made).
- **Verify**
 - Checks image signatures to ensure only trusted code runs.
- **Cleanup**
 - Deletes stale resources based on a schedule or TTL.
- <https://github.com/kyverno/policies/tree/main/best-practices>
- <https://github.com/kyverno/policies/tree/main/pod-security>

WHAT IT LOOKS LIKE



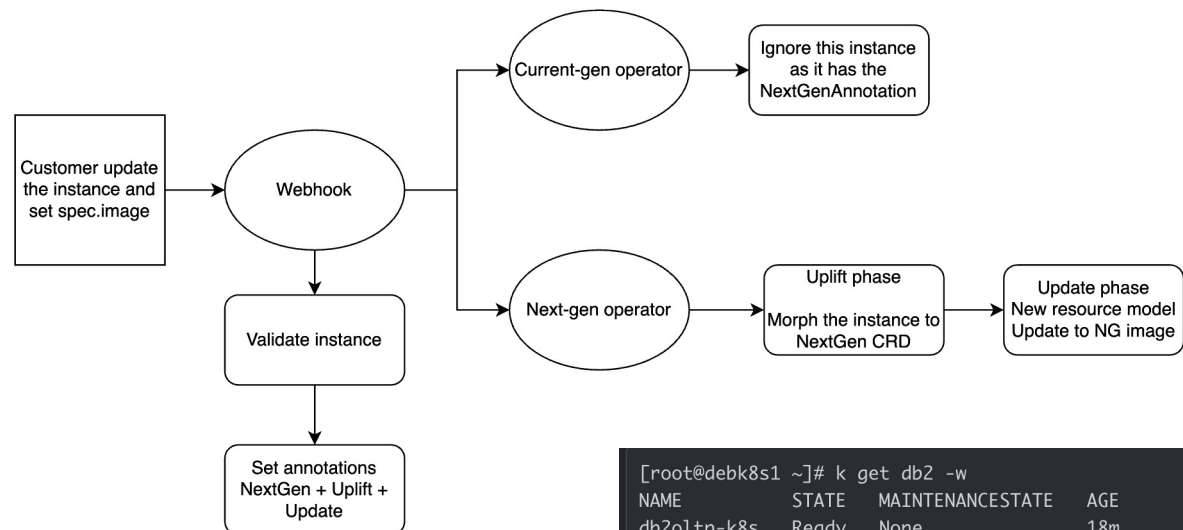


GETTING TO NEXT GEN

A simple path for existing users

UPLIFT

- A seamless transition into Next Gen
- A one-time, quick procedure



```
[root@debk8s1 ~]# k get db2 -w
NAME          STATE      MAINTENANCESTATE  AGE
db2oltp-k8s   Ready      None               18m
db2oltp-k8s   converge   None               26m
db2oltp-k8s   Transition None          26m
db2oltp-k8s   uplift     None               27m
db2oltp-k8s   Transition None          28m
db2oltp-k8s   update     None               29m
db2oltp-k8s   Transition None          29m
db2oltp-k8s   Ready      None               29m
```



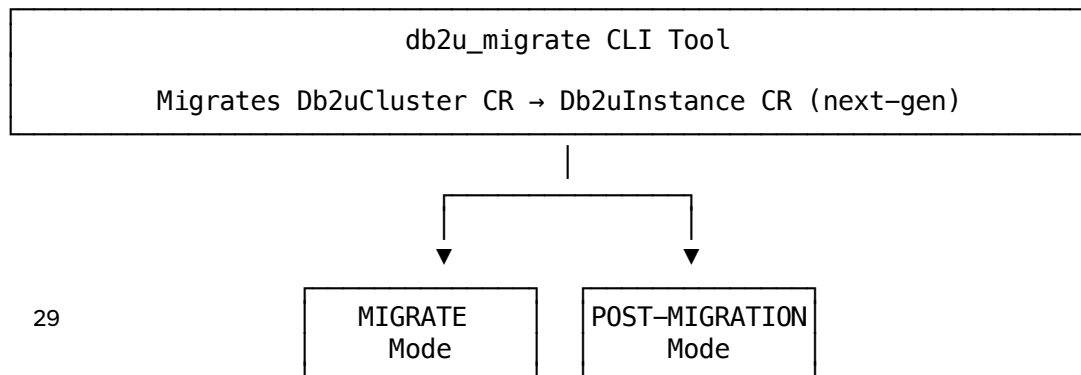
DEMO

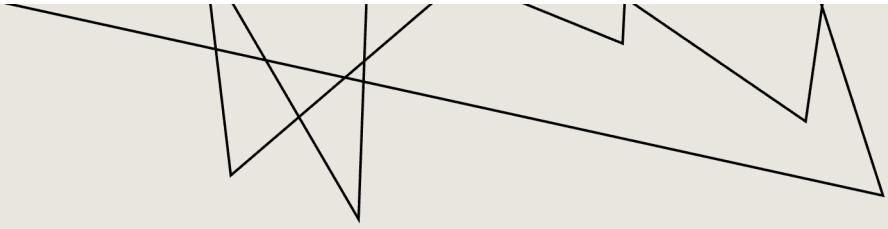
Uplift

MIGRATION

Db2uCluster is no longer supported in Next Gen.

We've developed a standalone migration tool to get your Db2uCluster to Db2uInstance:





EXCEEDING EXPECTATIONS

Measurable gains across the board

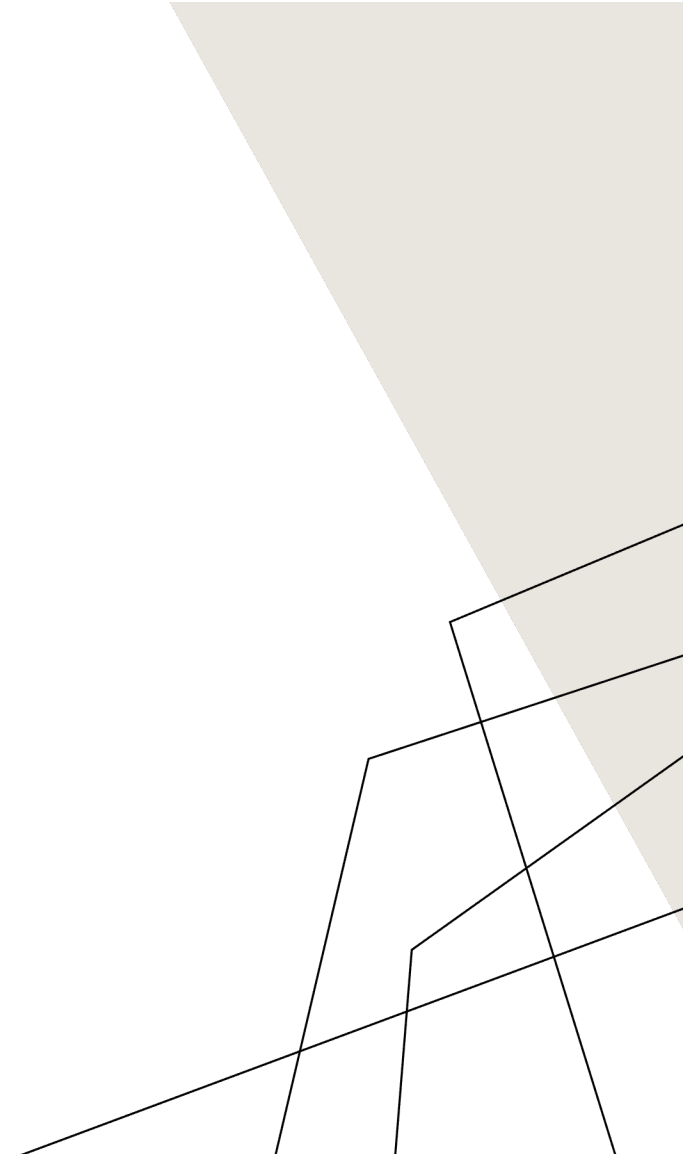
BY THE NUMBERS

- Kubernetes resource reduction

Category	Current Gen	Next Gen	Reduction
Secrets	13	3	↓ 77%
ConfigMaps	7	2	↓ 71%
Jobs	2	0	↓ 100%
Auxiliary Components	4	0	↓ 100%
Other	13	13	-
Overall	39	18	↓ 50%

TOP 5 ENHANCEMENTS

- Install
 - Helm-based install support
- Deploy
 - Deployment time **down 54%**
 - Image footprint **down 86%**
- Resiliency
 - Brand new HA design
- Security
 - CVEs **down 89%**
- Usability
 - Prechecker
 - Maintenance mode annotation for disruptive admin ops





DEMO

Current Gen and Next Gen
Side-by-Side

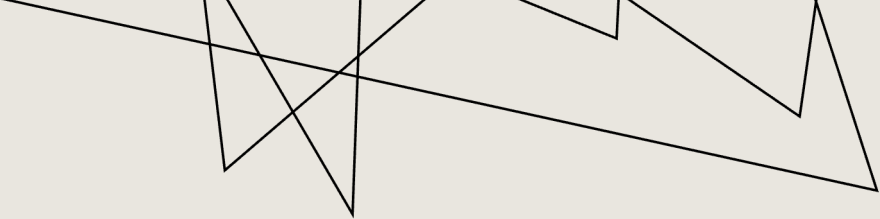


THANK YOU!

Want to give it a try? Reach out!

tonyaps@ca.ibm.com

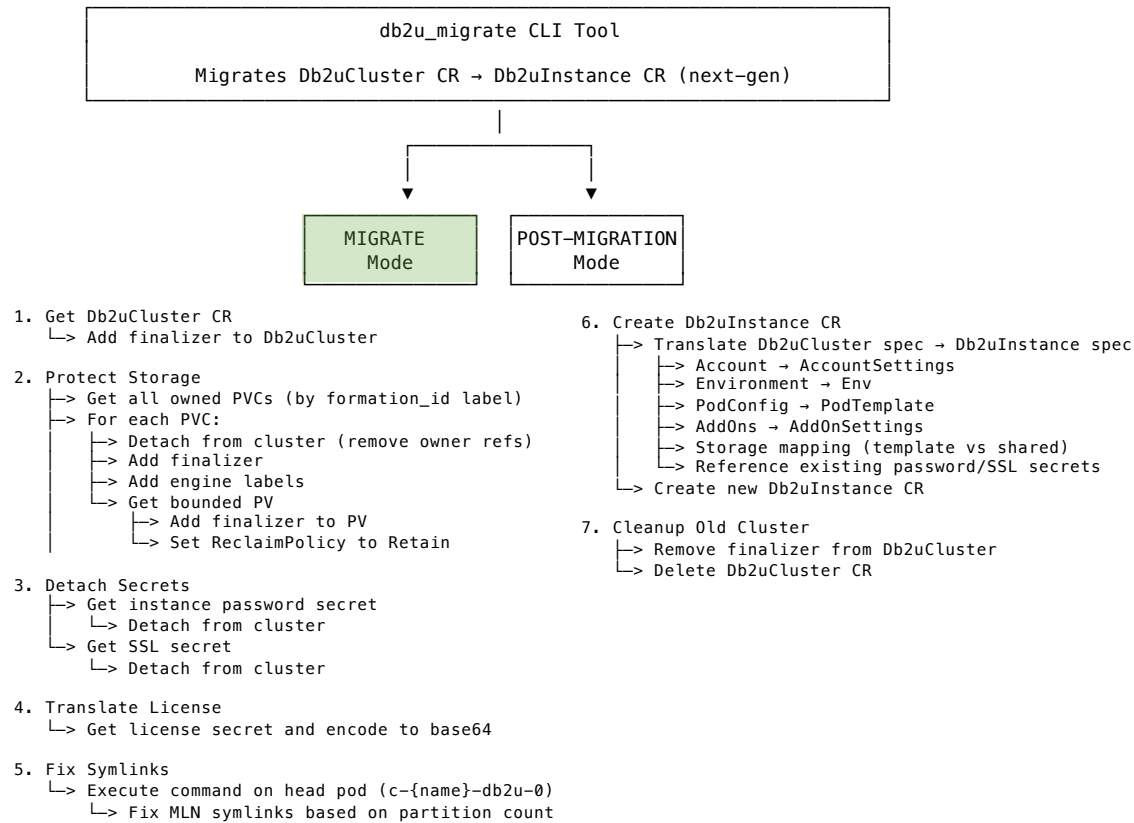
janpreet.chandhok@ibm.com



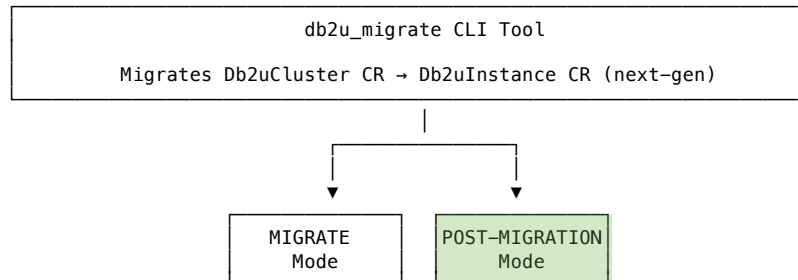
INDEX

Auxiliary content

MIGRATION: MIGRATE MODE



MIGRATION: MIGRATE MODE



1. Get `Db2uInstance` CR
 - ↳ Verify instance state is "Ready"
2. Unprotect Storage
 - ↳ Get all owned PVCs
 - ↳ For each PVC:
 - ↳ Remove finalizer from PVC
 - ↳ Get bounded PV
 - ↳ Remove finalizer from PV
 - ↳ Set `ReclaimPolicy` back to Delete
3. Unprotect Secrets
 - ↳ Remove finalizers from all owned secrets