IBM **TechXchange**

Bringing DevOps to the Database with IBM DevOps and DBmaestro

Hands-on lab guide

Randall Langehennig
rlange@ibm.com
Product Manager, IBM DevOps

Daniel Cappon
dcappon@ca.ibm.com
Product Manager, IBM DevOps

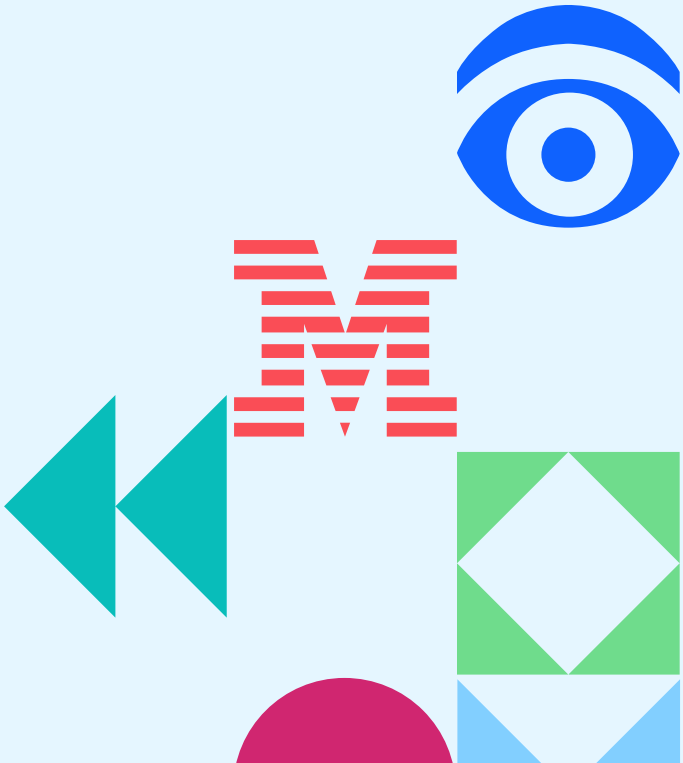


Table of Contents

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled,
isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

Notices and disclaimers (Continued)

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at

Learn more →

1 Introduction

DBmaestro is a DevSecOps platform and organizational database management and automation solution. DBmaestro deploys everything, anywhere, to all databases. Its purpose is to assist users with implementing and executing organizational policies and best practices for seamless database deployment automation.

DBmaestro enables companies to adopt DevOps best practices in database deployments to improve the frequency, quality, and speed of deployments. DBmaestro is a DevSecOps solution for all your databases that offers:

- Dev: Database source control, compilation automation, self-service.
- Sec: Security and compliance, policy enforcement, role management.
- Ops: Deployment automation, risk mitigation, auditing and KPIs (key performance indicators).

DBmaestro helps our clients in:

- Creating an end-to-end CI/CD pipeline for database deployments.
- Have a consistent automated process, eliminating quality issues and boosting the speed of deployments.
- Ensure a secure and compliant deployment process.
- Build once, deploy many times.
- Deploy in the same way to all environments.
- Always have complete visibility of who made changes and what changes were made, where and why.
- Maintain a complete, easy-to-access record of all changes for compliance and audit reporting, including an in-depth audit trail that describes all database activities and changes.
- Manage versioning risks, identify and alert about unexpected configuration changes.
- Avoid unauthorized changes in your databases that are unrelated to your policies.

DBmaestro focuses on strengthening team collaboration through version control, which is crucial for maintaining a single source of truth for all database changes. This approach helps minimize errors and time-consuming rework, which is essential for automating database security, compliance, and governance. By managing and enforcing organizational policies and standards, DBmaestro ensures your database remains secure, with role-based access and delivery.

In terms of integration with CI/CD pipelines and compatibility with a wide range of database management systems, DBmaestro excels in offering support for Oracle, SQL Server, PostgreSQL, MySQL, MariaDB, Amazon RDS, among others.

DBmaestro stands out for its flexibility, allowing teams to adapt the platform to their specific needs without the need for complex programming. This facilitates integration with existing processes and promotes effective collaboration between developers, DevOps teams and DBAs.

DOP Release Automation

Release faster while improving quality with smart automation for your database. Accelerate feedback loops between developers and DBAs to save time and eliminate costly rework.

Its main responsibility is to manage database release pipelines:

- Package, verify, deploy, and promote the database delivery pipeline
- Dry run the next release on a provisional environment to prevent errors

- Identify, flag, and manage configuration drifts, conflicts, and errors to ensure a successful release every single time
- Revalidate the database's final state post-deployment and audit all changes made

1.1 About this hands-on lab

The main objective of this hands-on lab is to present an end-to-end overview of what it takes to bring DevOps to the Database using DBmaestro. A secondary objective is to show how the solution can be easily integrated into existing pipelines you may be using today from GitLab, Jenkins, Azure DevOps, and IBM DevOps Deploy as examples. This lab is focused on GitLab as the pipeline orchestrator, however, you will see how this would apply to other investments you may have in place today.

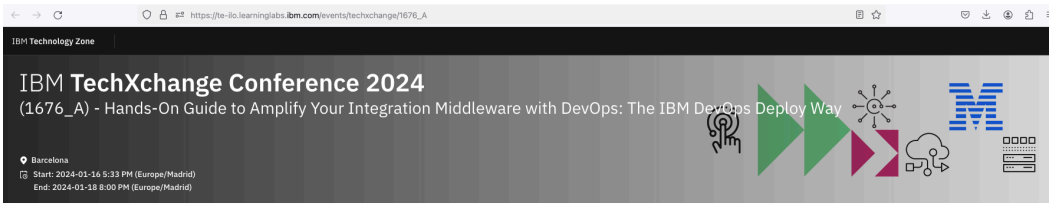
You do not have to be a DBA to get started with this solution and see the value it brings. This solution helps to reduce a DevOps velocity gap that exists between application and the database allowing your teams to move with greater agility. Finally, you will see that DBmaestro has excellent governance in place that will ensure that quality SQL (both DDL and DML) updates are applied that follow your teams' requirements and will not break your production environment. Excellent guard rails are provided to ensure this as you bring DevOps to your database environments.

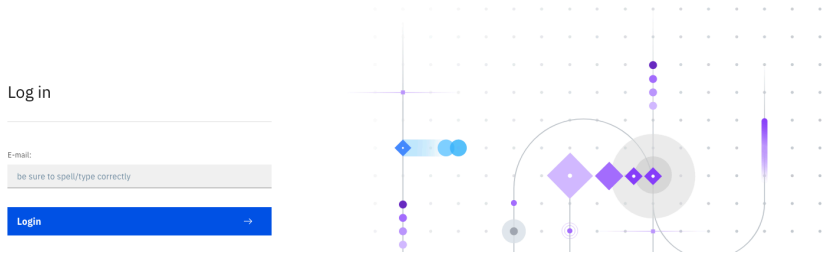
2 Getting Started

2.1 How to Connect to the Lab

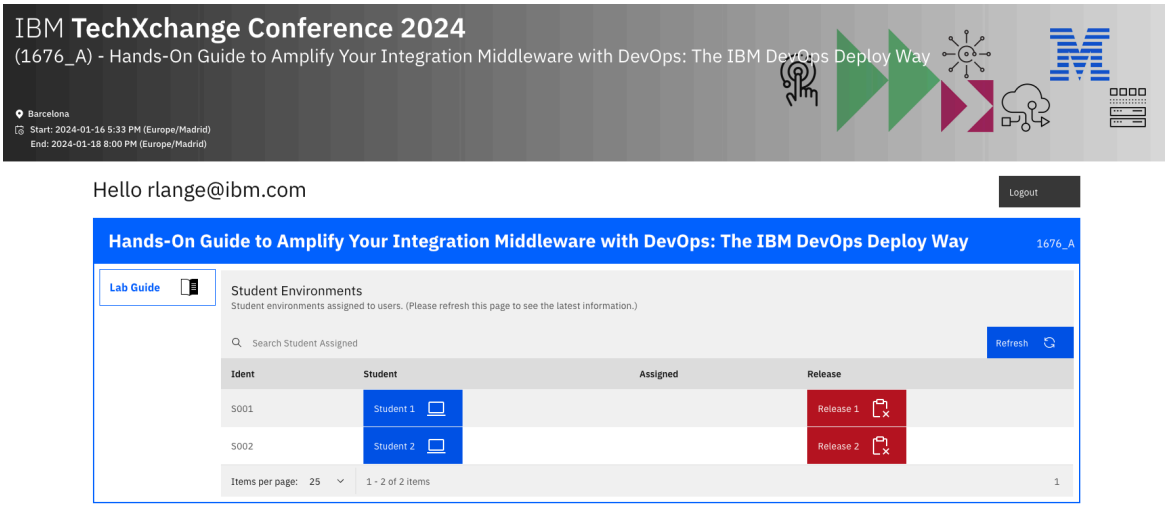
Follow these steps to connect to the lab environment

1. On your computer, start your Firefox browser.
2. Click the link to your environment that has been assigned to you. For example, you will see a screen like this:

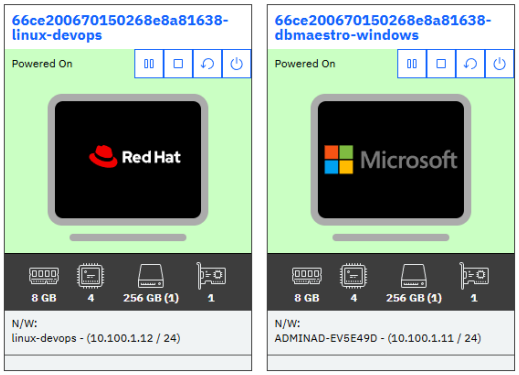




3. Enter your email address and click the Login button (as shown above). You should see a screen that looks like this:



4. Click on the “Launch Lab” button. You will see the following screen:



5. Click on the “Microsoft” screen as shown above. This will launch the console we will be using for the labs.
6. Click on the screen and you see a login for the ‘admin’ user. Provide a password of “IBMDem0s” and press Enter.
7. You are good to proceed to the first lab!

3 Lab Exercises

During the lab, we will showcase DBmaestro capabilities for implementing a Secure DevOps process for databases.

3.1 Deployment of Database DDL scripts package

In this section, we will demonstrate how to deploy a DDL script package to the *Integration* and *QA* environments. We will use a CI/CD pipeline implemented with both DBmaestro and GitLab.

The package upgrade script contains the table creation:

CreateTableUsers.sql 70 B

```
1  CREATE TABLE IF NOT EXISTS "Users"
2  (
3      "FirstName" character(50)
4  )
5
```

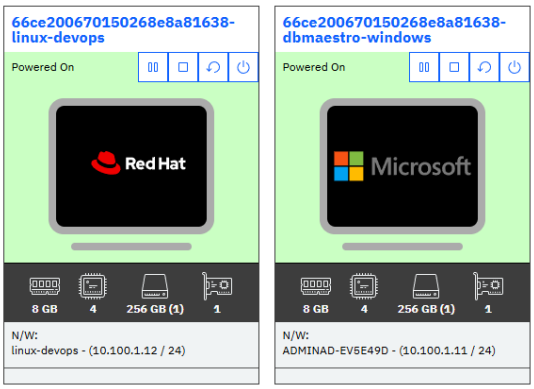
The downgrade script contains the table drop as shown below:

DropTableUsers.sql 18 B

```
1  DROP TABLE "Users"
```

Let’s get started.

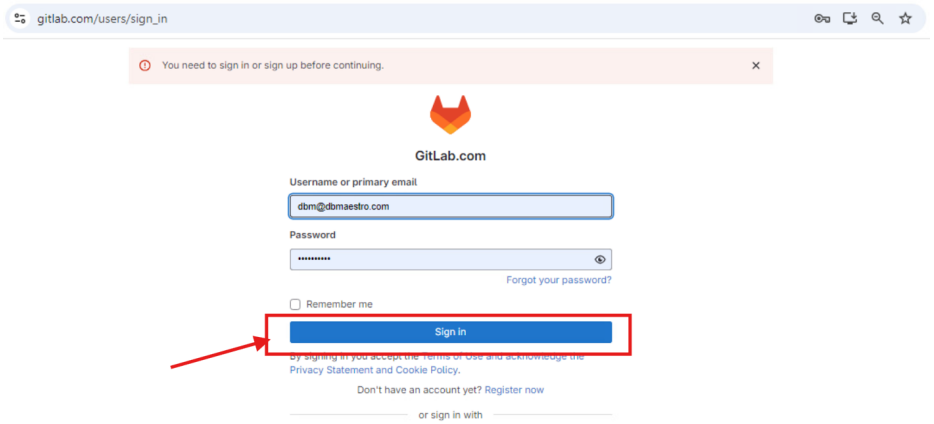
1. Click on the Microsoft image found in the middle of the virtual machines to open this image. Login as the user ‘admin’ and the pasasword of ‘IBMDem0s”. The 0 is a zero.



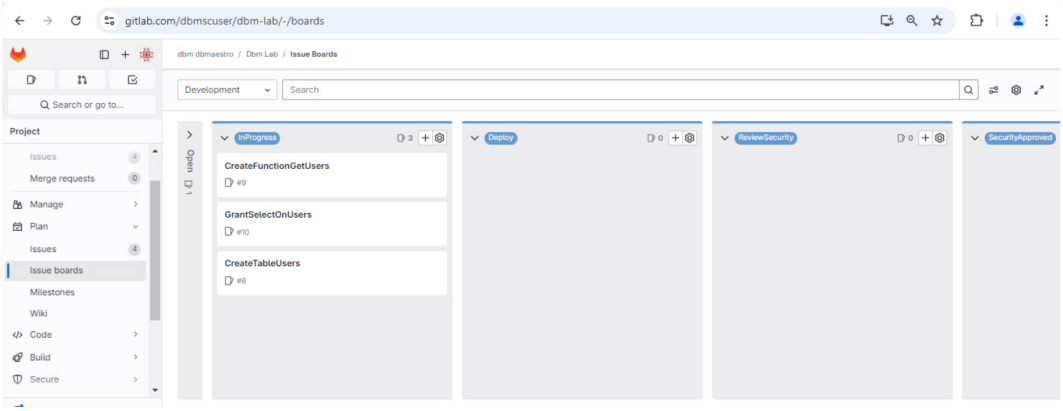
2. Execute the GitLab board preparation script clicking on the Powershell icon anchored to the task bar. This step creates the GitLab board cards required for this lab.

3. Open the Chrome browser using the anchored icon from the task bar.

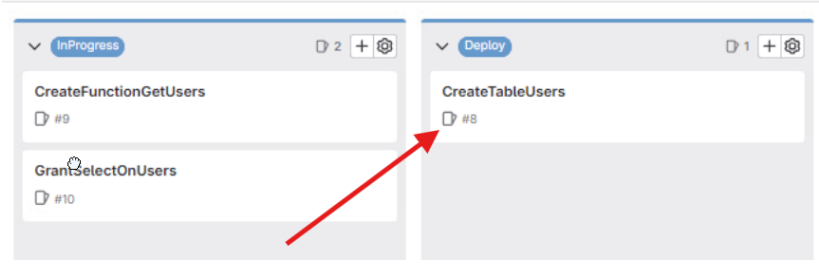
4. If required, login to GitLab using the stored credentials. Click the ‘Sign In’ button.



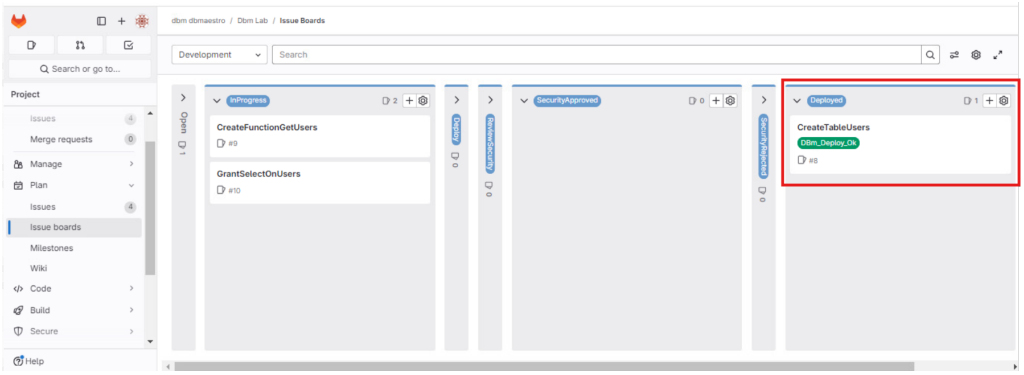
5. You should see the GitLab Project Board as shown below:



6. Drag the card ‘CreateTableUsers’ work item from InProgress to Deploy state. This step triggers the DBmaestro deployment process to Integration and QA environments.



7. Click the browser refresh button after 30 seconds. Repeat until you see the card was moved to the ‘Deployed’ state and the label ‘DBM_Deploy_Ok’ was added to the card.



8. At this point, the DDL scripts were executed in the Integration and QA environments. Congratulations in bringing your first automated deployment for the Database!

3.2 Scripts deployment with security approval process

In this section, we will showcase DBmaestro capabilities to perform static code analysis against your SQL code which can help you quickly identify potential security problems. The result is a very fast feedback loop for your developers.

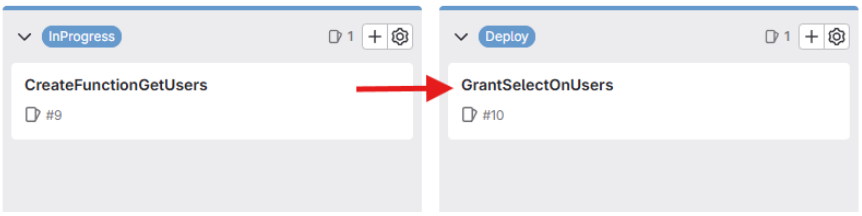
In this sample scenario, the deployed package script contains a GRANT sentence, which is detected during the DBmaestro ‘Pre-Check’ process. This will trigger a security review approval step before the deployment can take place. After the scripts are reviewed and approved, the developer can deploy the changes successfully.

DBmaestro provides a three step CI process (called 'Pre-Check'), to automate code review, perform a dry-run release, and assist with issues.

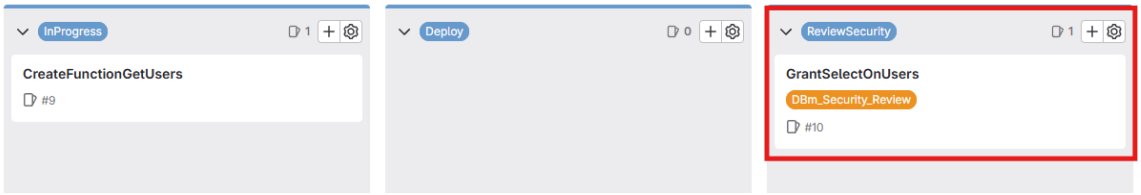
- Static code analysis. This includes 100+ automated code tests provided out of the box. You can extend this to cover additional needs your organization requires as well. This step can automate code review and prevent security or standards violations.
- Dry-Run will test all pending changes together, flushing out any runtime issues or conflicts between change contributors.
- AI assistant will provide explanation of any DB errors that arise and will provide resolution details to assist your development teams.

The goal of DBMaestro Pre-Check is to provide an automated short-feedback loop to prevent bad code being introduced into an environment and to help achieve self-service for developers.

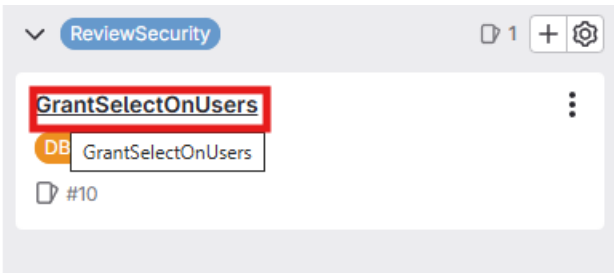
1. Drag the card 'GrantSelectOnUsers' from the list 'InProgress' to the list 'Deploy' as shown below:



2. This step will trigger the DBmaestro PreCheck/Deployment process to the Integration and QA environments.
3. Click the browser refresh button after 30 seconds and repeat this until you see the card has moved to the 'ReviewSecurity' state with a label of 'DBM_Security_Review' being added to the card.

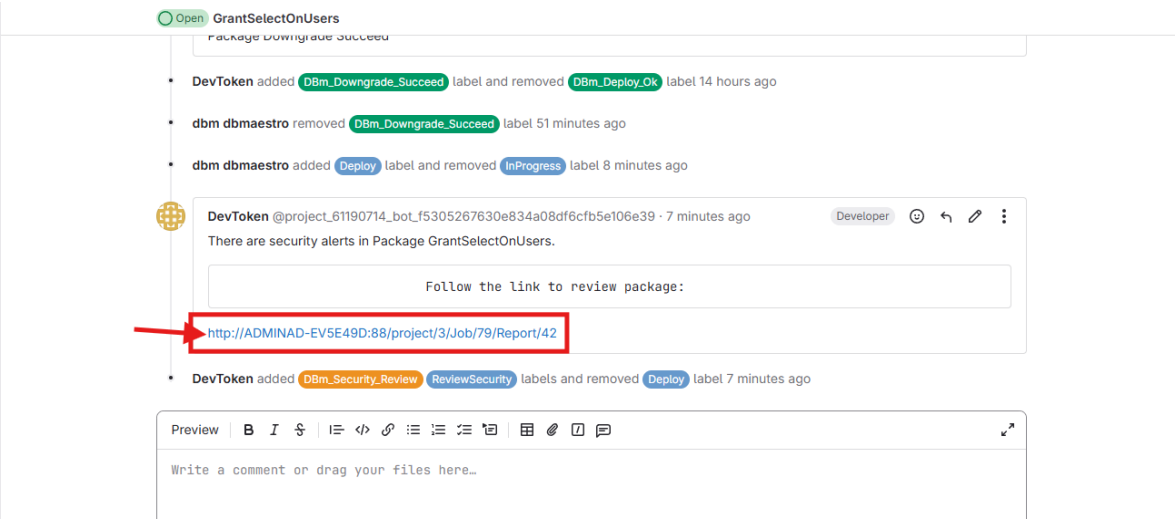


4. At this point, the DDL script deployment was blocked and it requires the security team to review and approve or reject the changes.
5. Click on the GrantSelectOnUsers card title as shown below to go to the card details and comments history.



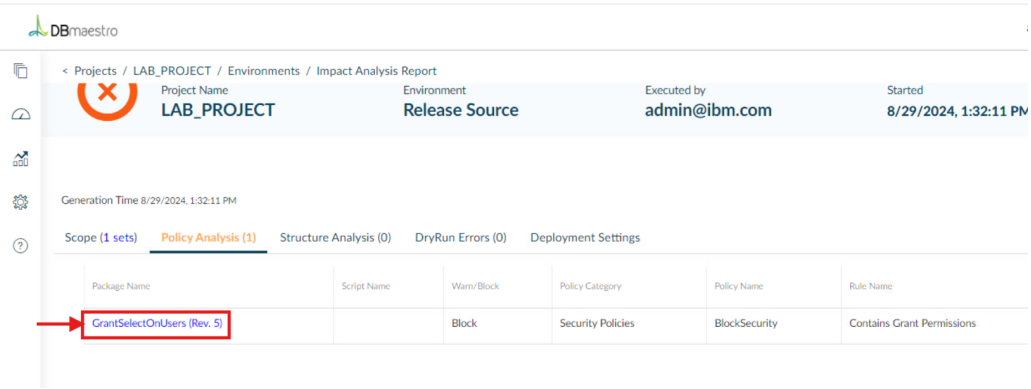
6. Scroll down until the end of the card activity history. You will see the comment:

"There are security alerts in Package GrantSelectOnUsers"



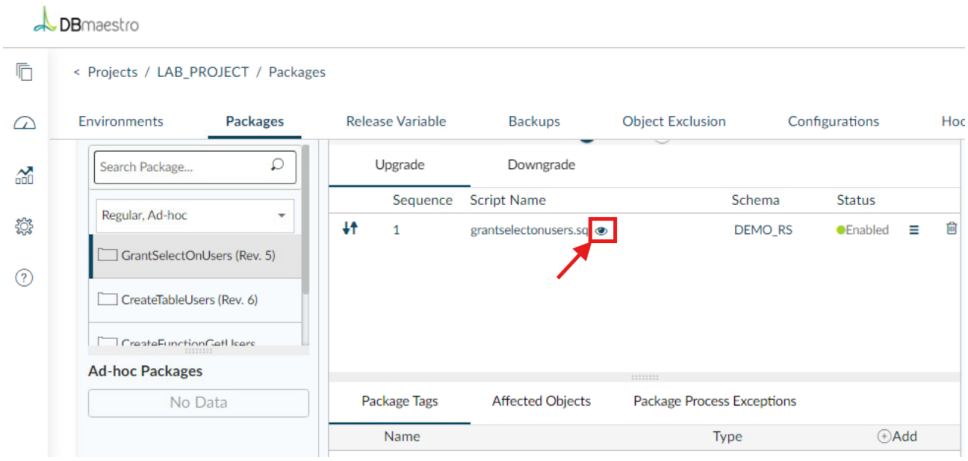
7. Click the link (as highlighted above) and it will open the DBmaestro UI where you will be able to review and approve the package.

8. A new chrome tab will be opened showing the DBmaestro UI with the Policy Analysis report. This report displays the list of violated Policies and Rules.

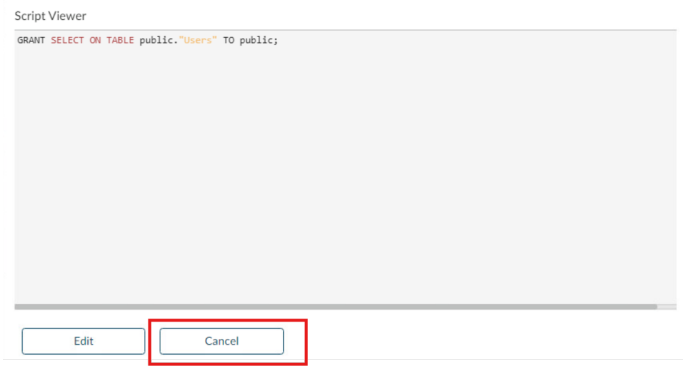


9. Click the package name (as highlighted above)

10. DBmaestro shows the package list of SQL scripts. Click the eye icon to view the script content as shown below:

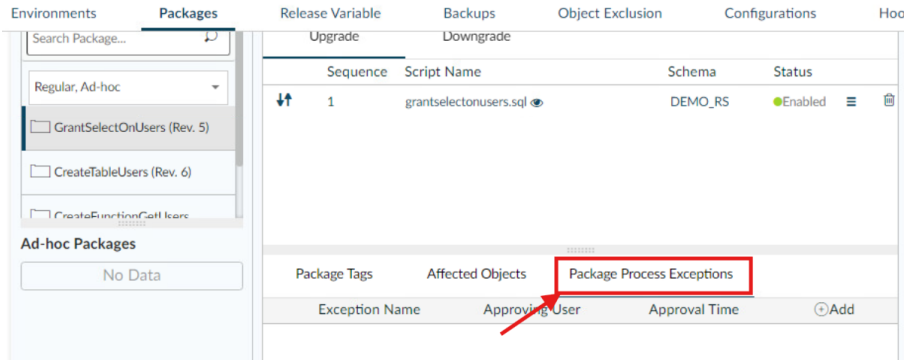


11. DBmaestro will show the script. Click the cancel button after you have reviewed the code and determined it is a safe SQL script.

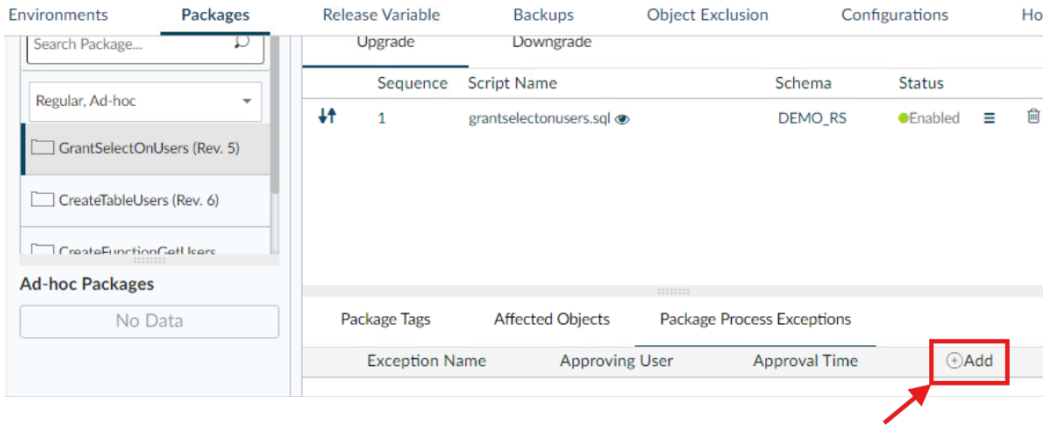


12. We will now approve the package by following these steps:

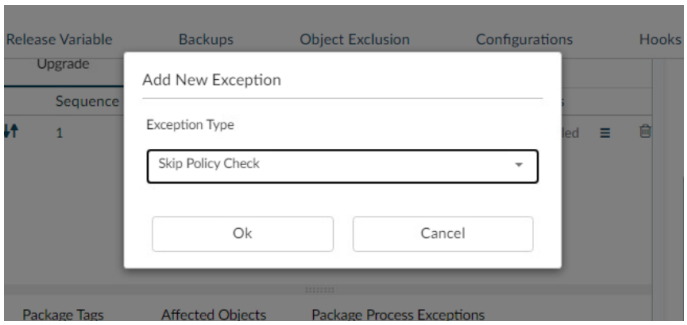
    a. Click on the Package Process Exceptions tab



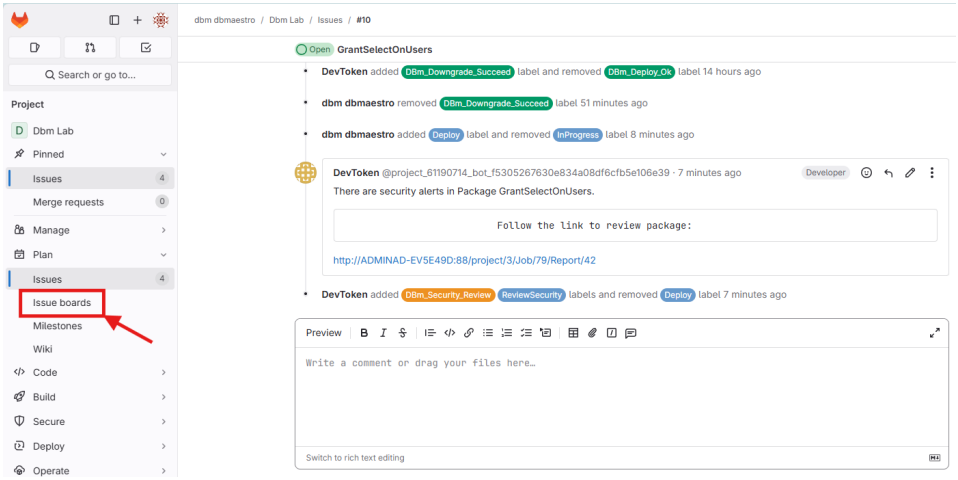
    b. Click the Add button



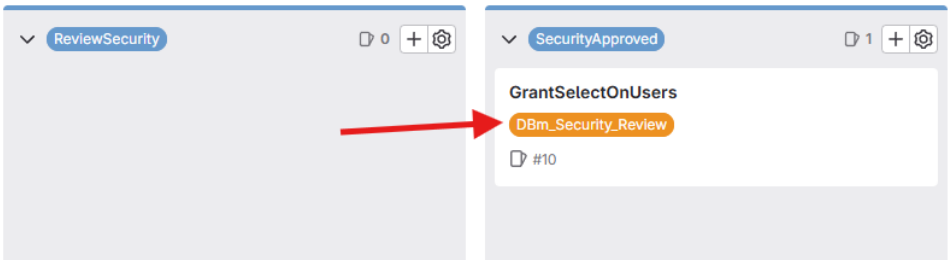
c. Select the option ‘Skip Policy Check’ from the pulldown menu and press Ok.



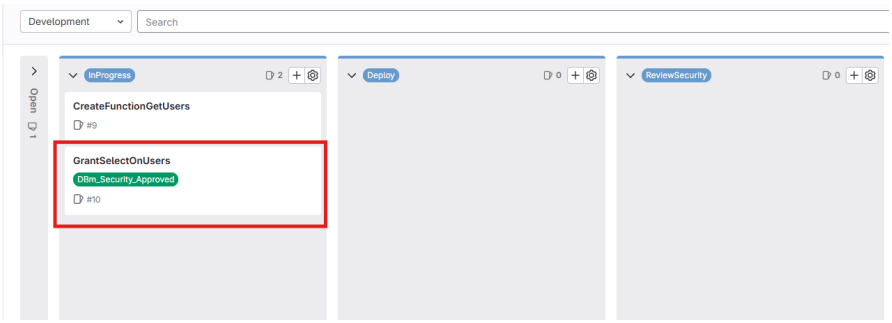
13. Once approved, go back to the GitLab UI tab and select the ‘Issue boards’ view as shown below:



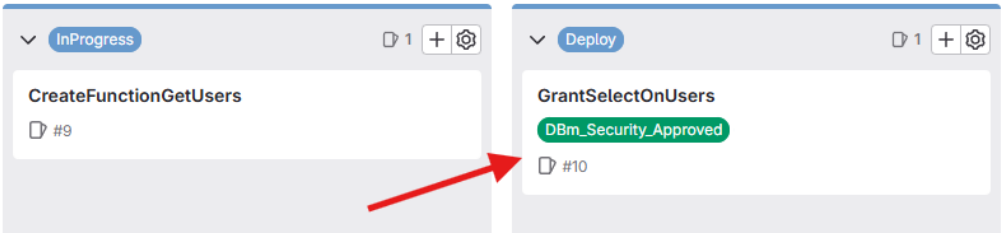
14. Drag the card ‘GrantSelectOnUsers’ from the ReviewSecurity state to the ‘SecurityApproved’ state.



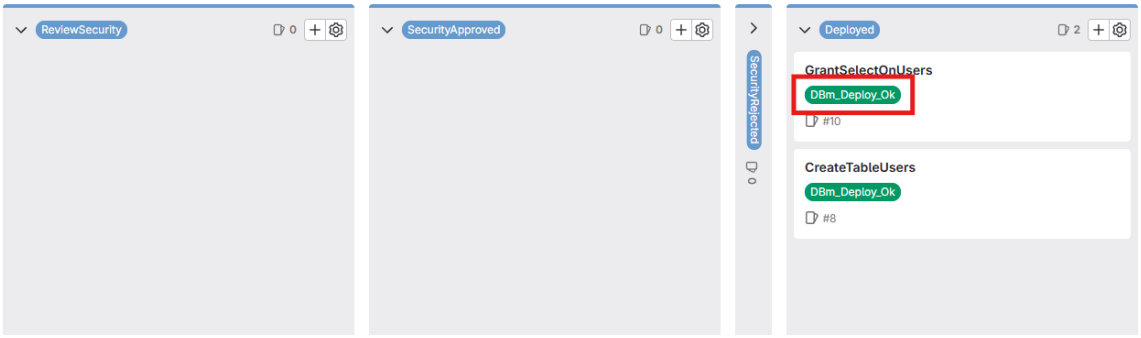
15. Click the browser refresh button after 30 seconds and repeat this until you see the card has moved to ‘InProgress’ (returned to the Developer) state and the label ‘DBm_Security_Approved’ tag was added to the card.



16. To deploy the approved Package, drag the card ‘GrantSelectOnUsers’ from ‘InProgress’ to ‘Deploy’ as shown below. This will trigger the deployment process.



17. Click the browser refresh button after 30 seconds and repeat until you see the card was moved to the ‘Deployed’ state and the label ‘DBm_Deploy_Ok’ was added to the card.



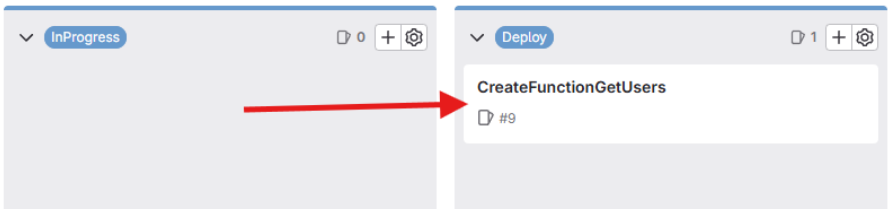
18. This means that the grant script was successfully executed in the Integration and QA environments. Congratulations, you have completed this section of the lab!

3.3 Block deployment due to violation of coding standards

In this section, we will review DBmaestro's capabilities in code review and detection of coding standards violations. This allows developers to receive immediate feedback on code quality eliminating the need to wait for code review comments.

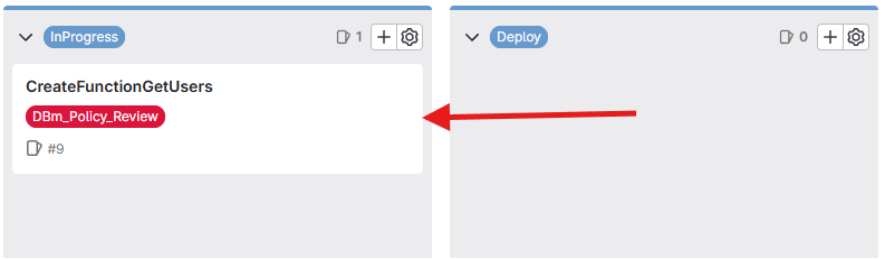
In this scenario, deployed package script contains a 'WHERE ... LIKE '%' statement, which is detected during the DBmaestro Pre-Check process as a coding standard violation. This will block the deployment of this script until the code is fixed.

1. Drag the card 'CreateFunctionGetUsers' from the list 'InProgress' to the list 'Deploy' as shown below:



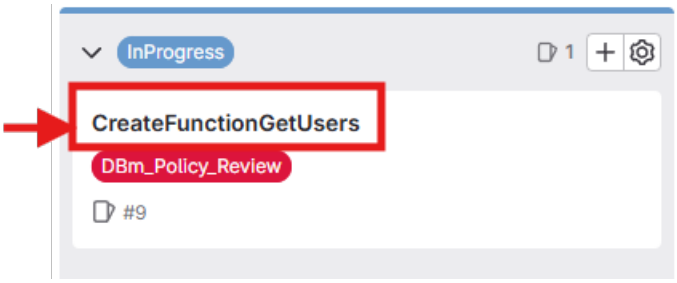
2. This step will trigger the DBmaestro Pre-Check/Deployment process to the Integration and QA environments.

3. Click the browser refresh button after 30 seconds and repeat until you see the card was returned to the 'InProgress' state and the label 'DBm_Policy_Review' is added to the card.



4. At this point, the DDL scripts deployment was blocked and requires that the developer fix the offending code.

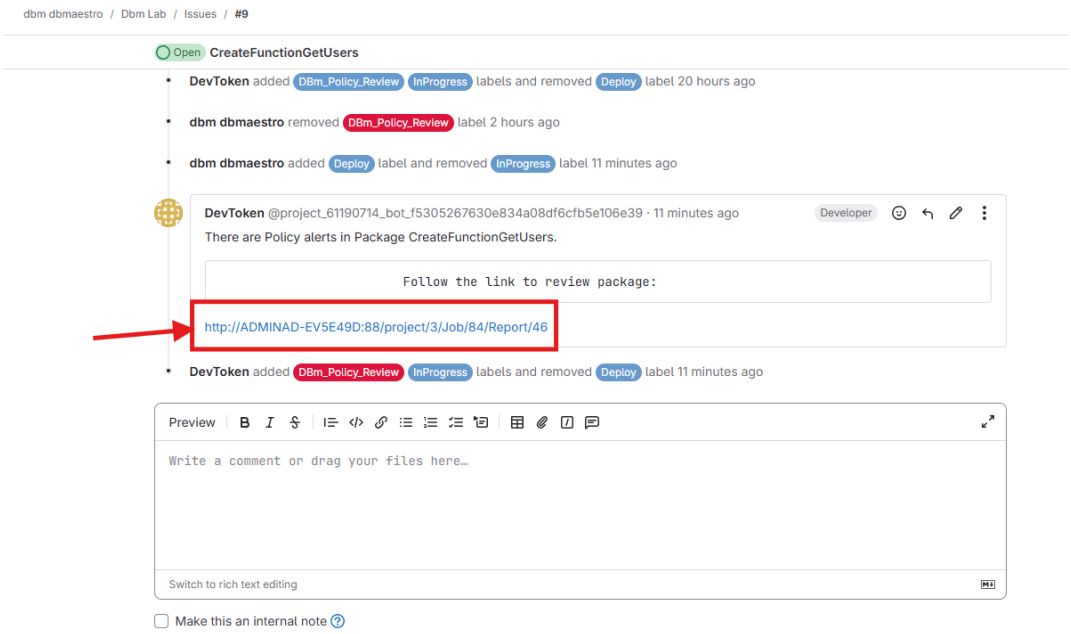
5. Click on the 'CreateFunctionGetUsers' card title to go to the card details and comments history.



6. Scroll down until the end of the card activity history. You will see the comment:

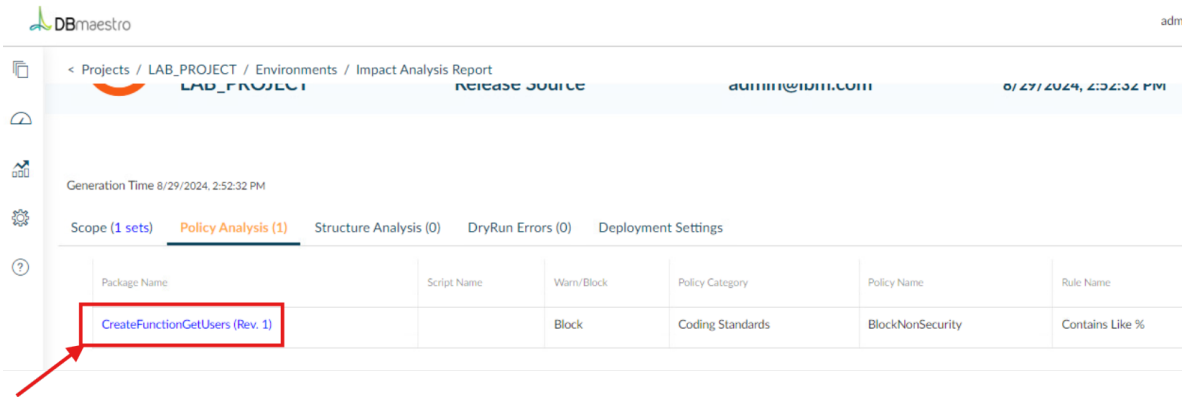
   "There are Policy alerts in Package CreateFunctionGetUsers"

Click the link that will open the DBmaestro UI, where you will be able to review and edit the package.



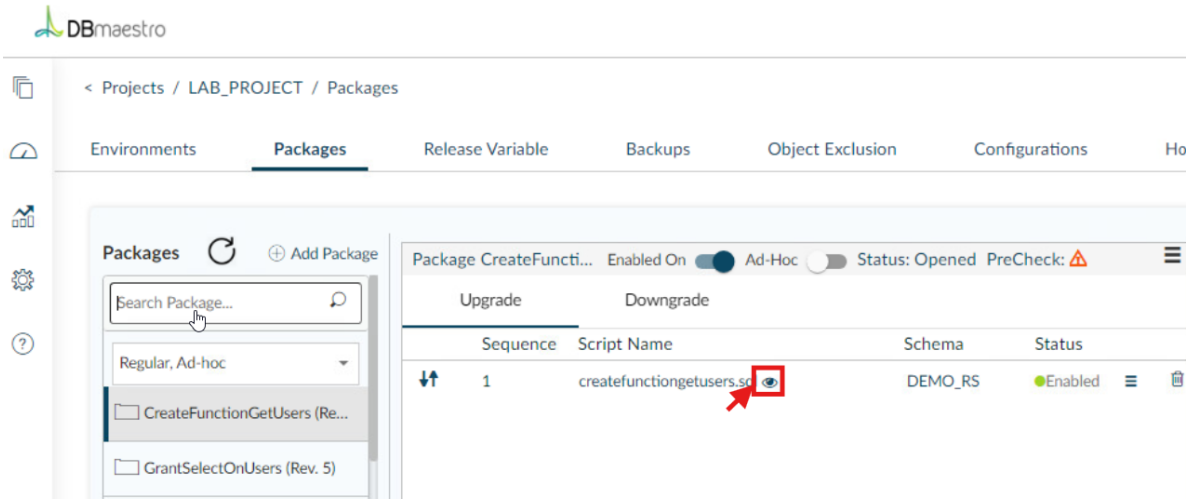
7. A new chrome tab will open showing the DBmaestro UI with the Policy Analysis report.

   This report displays the list of violated Policies and Rules.

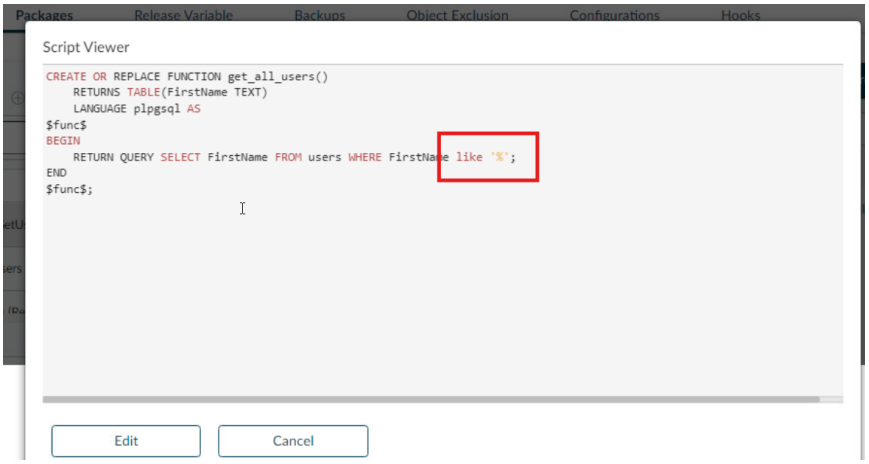


8. Click the package name ‘CreateFunctionGetUsers’ as shown above and this will take you to the package details.

9. DBmaestro will show the package list of SQL scripts. Click the eye icon to view the script content as shown below:



10. The script will be displayed in a Script Viewer. Notice the like '%' code as highlighted below. This violates our standard policy.



11. Click the cancel button after you have reviewed the script code to close the Script Viewer.

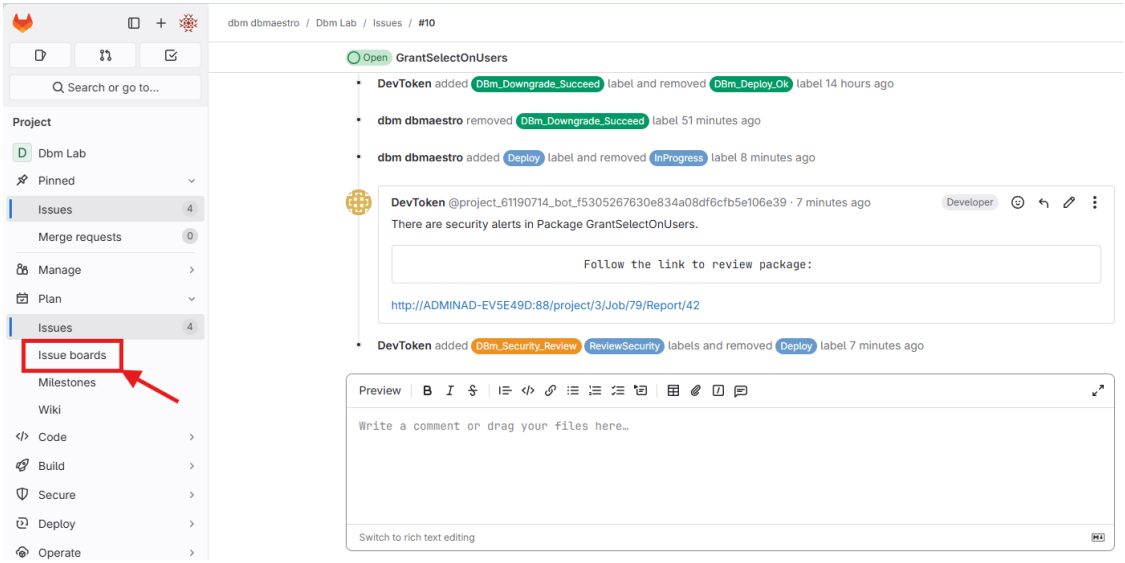
12. The package will remain blocked and will not be deployable until the script is fixed.

13. This concludes this lab to highlight how rules and policy enforcement can be achieved with DBmaestro. The process of fixing the script is beyond the scope of this lab, however, a developer would use this feedback to adjust and correct his code and then the process to deploy the code will be followed as you have seen.

3.4 Downgrade deployed changes

In this section, we will show how to revert the changes deployed in previous steps by executing the Downgrade scripts associated with each package that is created.

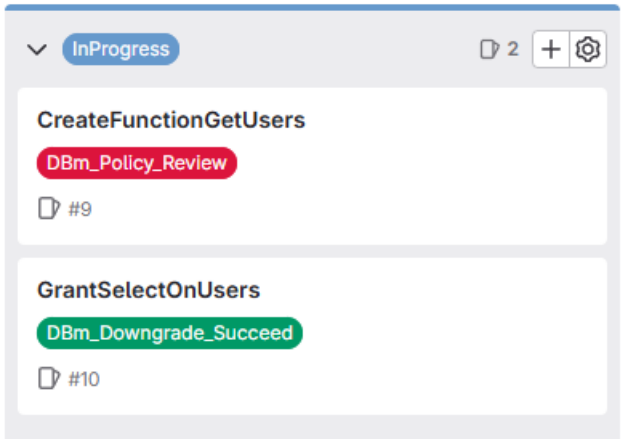
1. Return to the GitLab UI tab in your browser and select the 'Issue boards' view as shown below:



2. Drag the card ‘GrantSelectOnUsers’ from the list ‘Deployed’ to the list ‘InProgress’ as shown below:

3. This will trigger the DBmaestro downgrade process in the QA and Integration environments by executing the downgrade script that reverts the changes that were applied with the upgrade script.

4. Click the browser refresh button after 30 seconds and repeat until you see the card is moved to ‘InProgress’ (returned to the developer) state and the label ‘DBm_Downgrade_Succeed’ is added to the card.

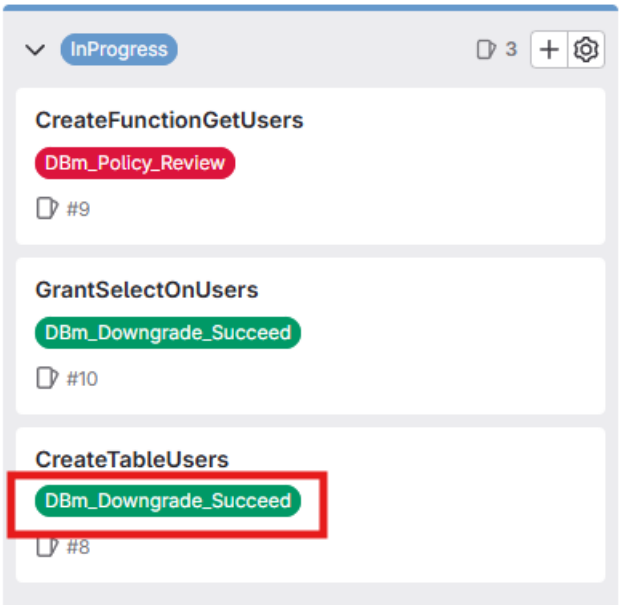


5. At this point, the grant permissions were revoked from the Integration and QA environments.

6. Drag the card ‘CreateTableUsers’ from the list ‘Deployed’ to the list ‘InProgress’ as shown below:

7. This step will trigger the DBmaestro downgrade process in the QA and Integration environments by executing the downgrade scripts that will revert the changes applied by the upgrade scripts.

8. Click the browser refresh button after 30 seconds and repeat until you see that the card has moved to ‘InProgress’ (returned to the developer) state and that the label ‘DBm_Downgrade_Success’ is added to the card as shown below:



9. At this point, the table users were dropped from the Integration and QA environments.

10. This concludes this lab showing the ability to rollback changes when required using DBmaestro.

4 Getting help and troubleshooting

This section provides information about getting help with your lab and some common troubleshooting topics.

4.1 Common troubleshooting tips

Use this section to document common troubleshooting issues that users may experience.

4.2 Getting help

Customize this section as needed, keeping in mind that some individuals will be doing this lab in the self-paced room without an instructor and would need guidance on where to go for assistance.