# Deployment Tool User’s Guide

## Introduction

The Vantiq Deployment Tool is designed to simplify deployment tasks by focusing on the deployment of projects.

The Deployment Manager presents the developer with a graphical view in which to manage partitions, environments, deployments and deployment activities.

- *Project Partition* represents a collection of resources defined in a project that are deployed to the set of nodes in a target environment identified by the partition’s constraint(s).
- *Environment* represents a collection of nodes.
- *Nodes* represent access to a namespace within a Vantiq installation. An environment contains a subset of the nodes defined in the enclosing namespace.
- *Deployment* defines the binding between a project and an environment that enables the deployment manager to deploy the project into the environment.

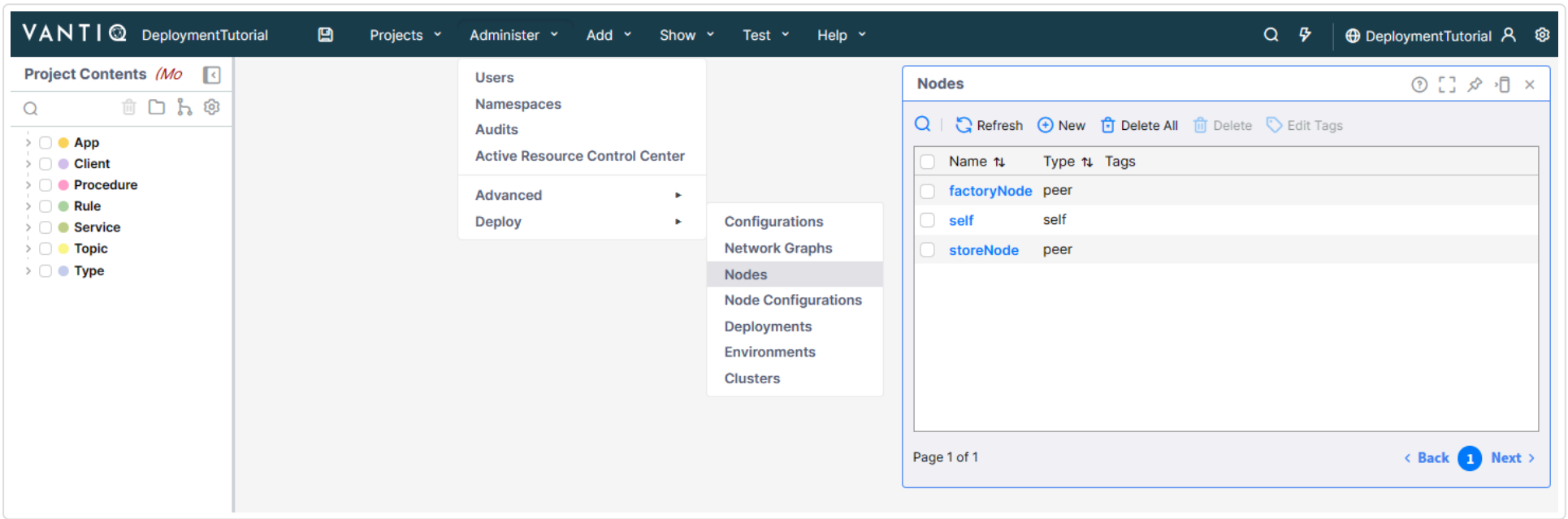
A project may be deployed to more than one environment satisfying the requirement to deploy to development, test and production environments. The tool uses a graphical notation inspired by the Project Resource Graph but optimized to support deployment activities.

This guide assumes you are familiar with building applications using the Vantiq IDE and will focus on creating Nodes, Environments and Deployments.

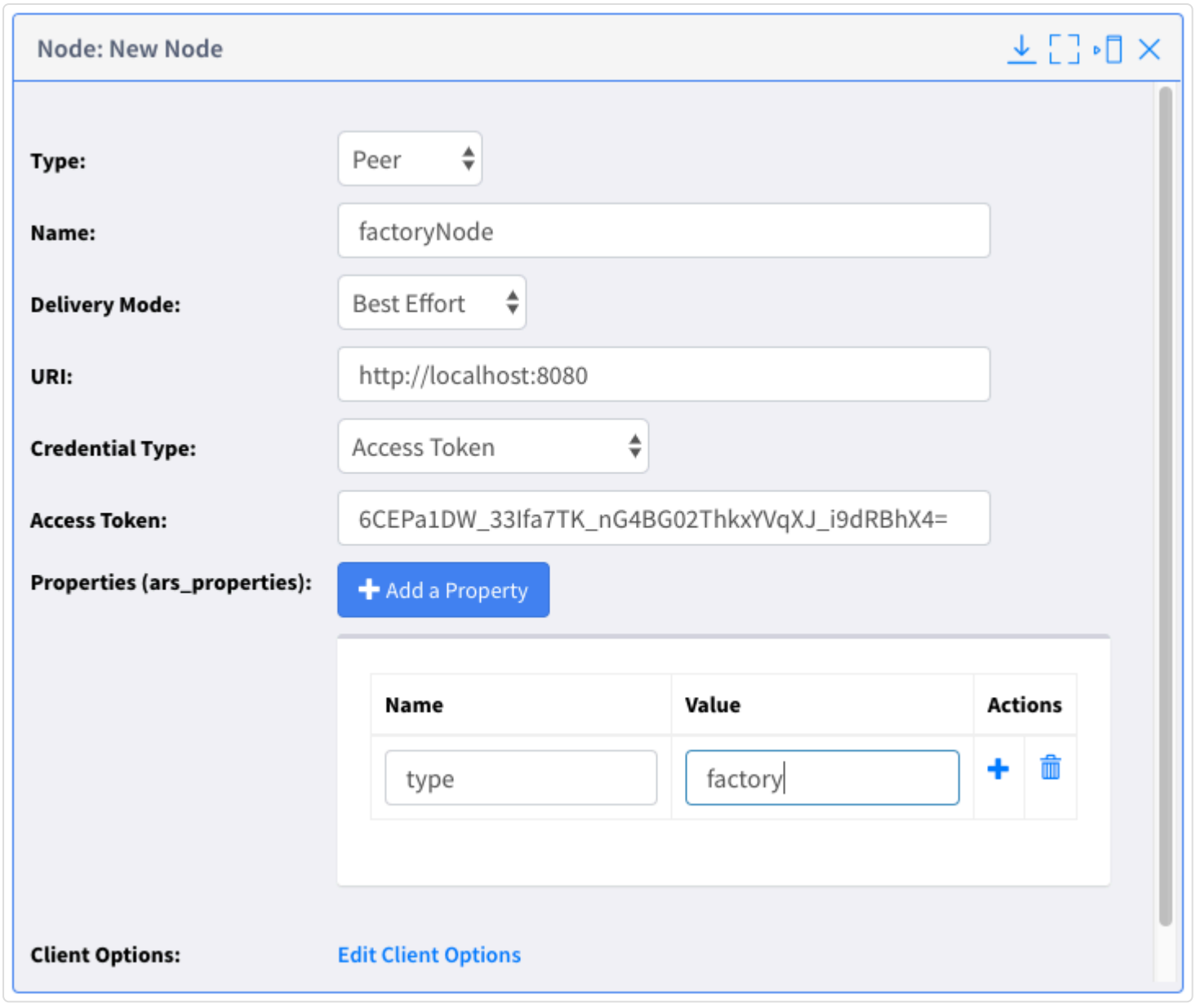
## Node

A Node in Vantiq defines access to a namespace within a Vantiq installation. In order for the Deployment Tool to deploy resources to a target namespace, a node must be created in the namespace where the deployment is executed, using an access token from the target namespace.

Go to **Administer** > **Deploy** > **Nodes** to bring up the Node List Pane.



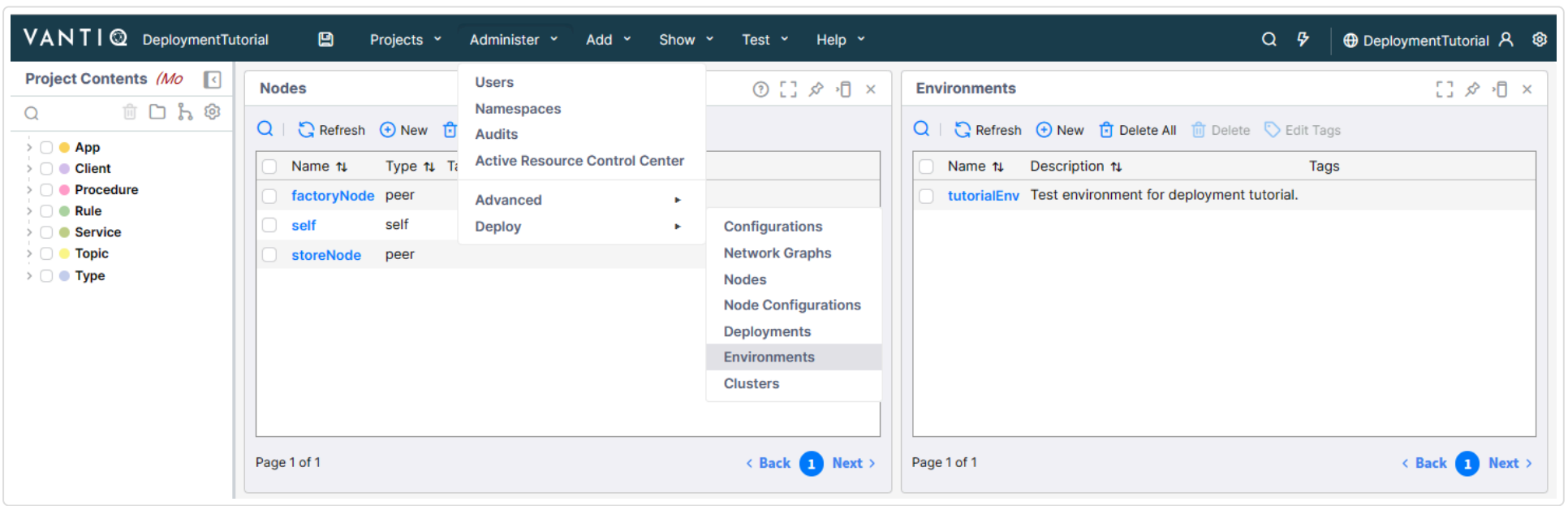

Click the **Create New** icon (“+” button) in the node list to create a new node. Or click on the name of a node to view and edit its properties.



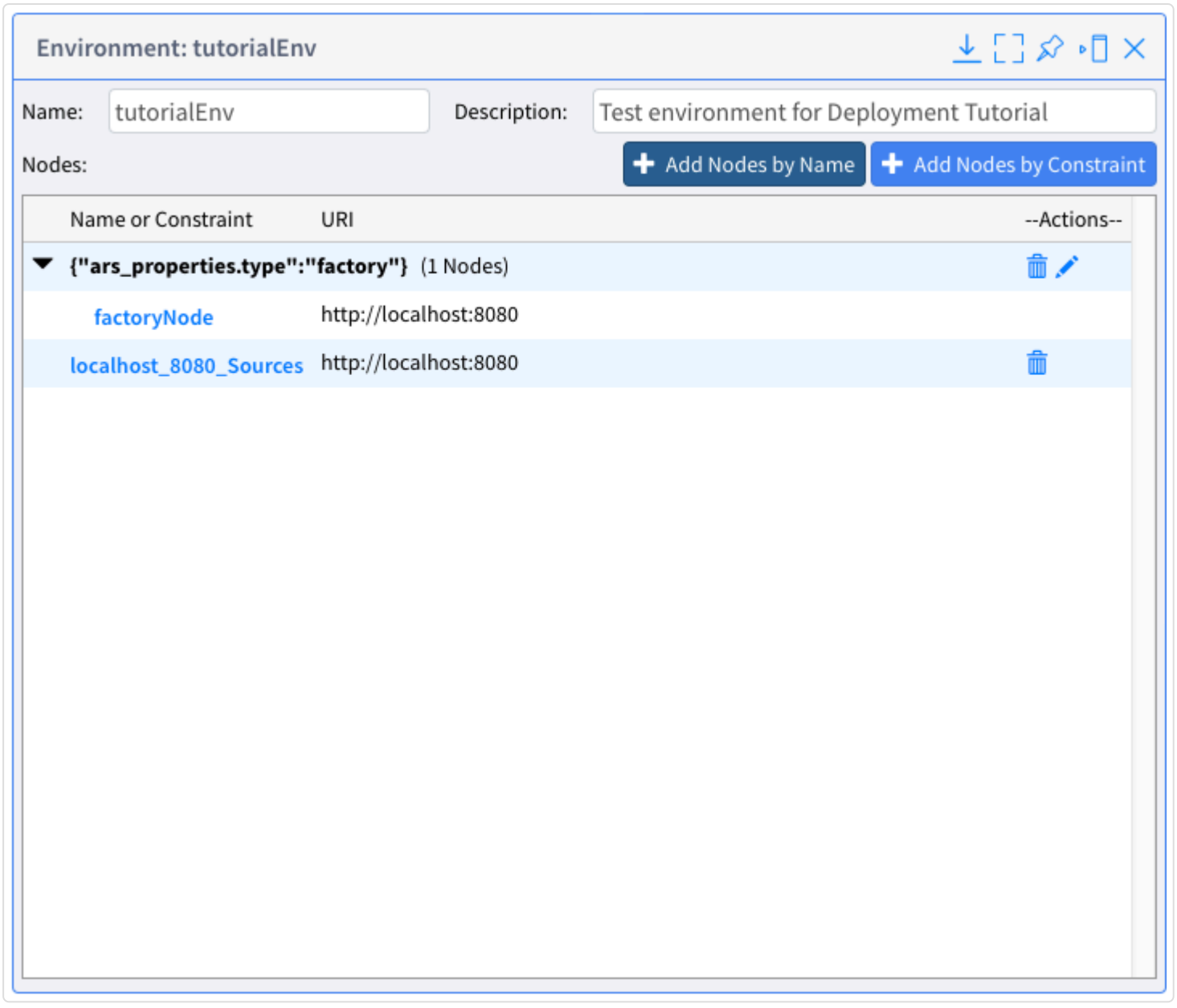

After the URI and Credential are provided, use the *Test Node Connection* button to verify the connection. Define as many properties as you want for the node. They are designed to help you place/group nodes into your Environment.

# Environment

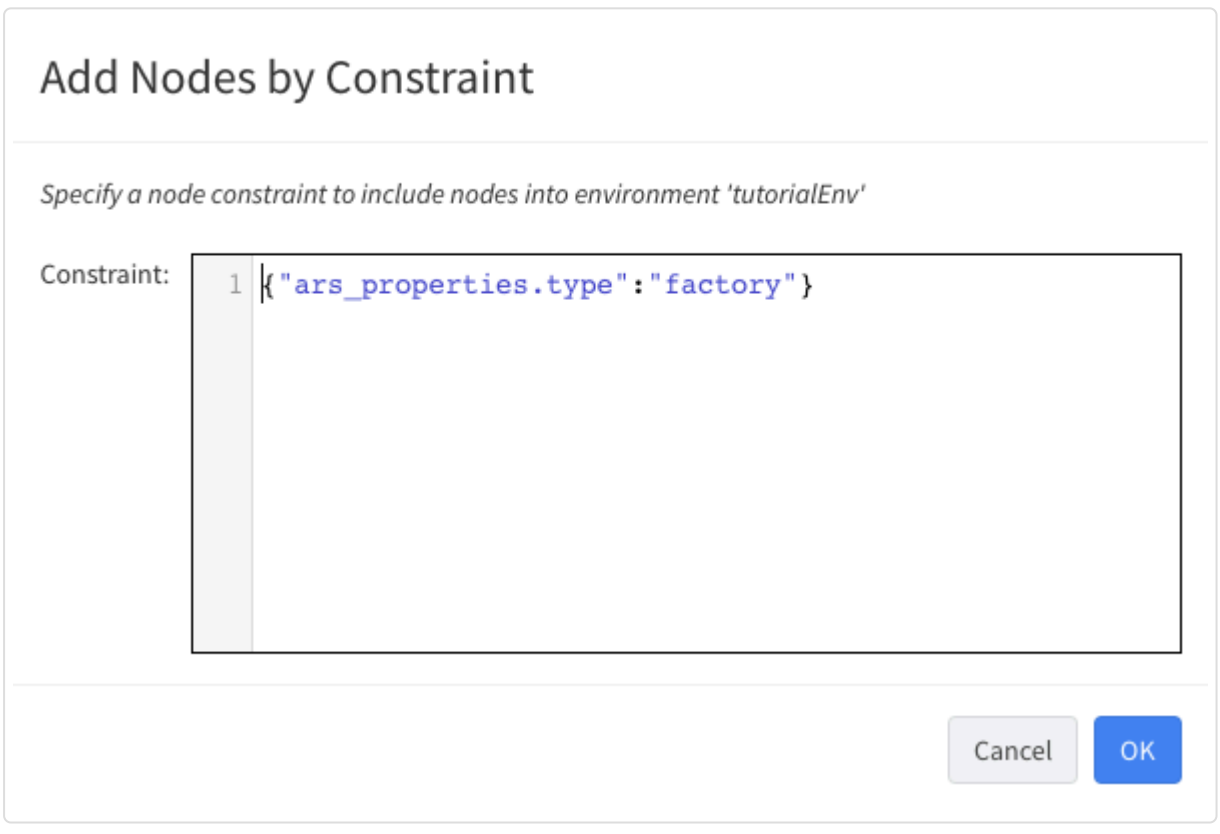
Go to **Administer** > **Deploy** > **Environments** to bring up the Environment List Pane.



Click the **Create New** icon (“+” button) in the environment list to create a new environment. Or click on the name of an environment to view and edit its definition.




Use the **Add Nodes by Name** or **Add Nodes by Constraint** button to select nodes into the environment.




Using a constraint to add nodes into an environment allows to automatically include any new node added to the same namespace.

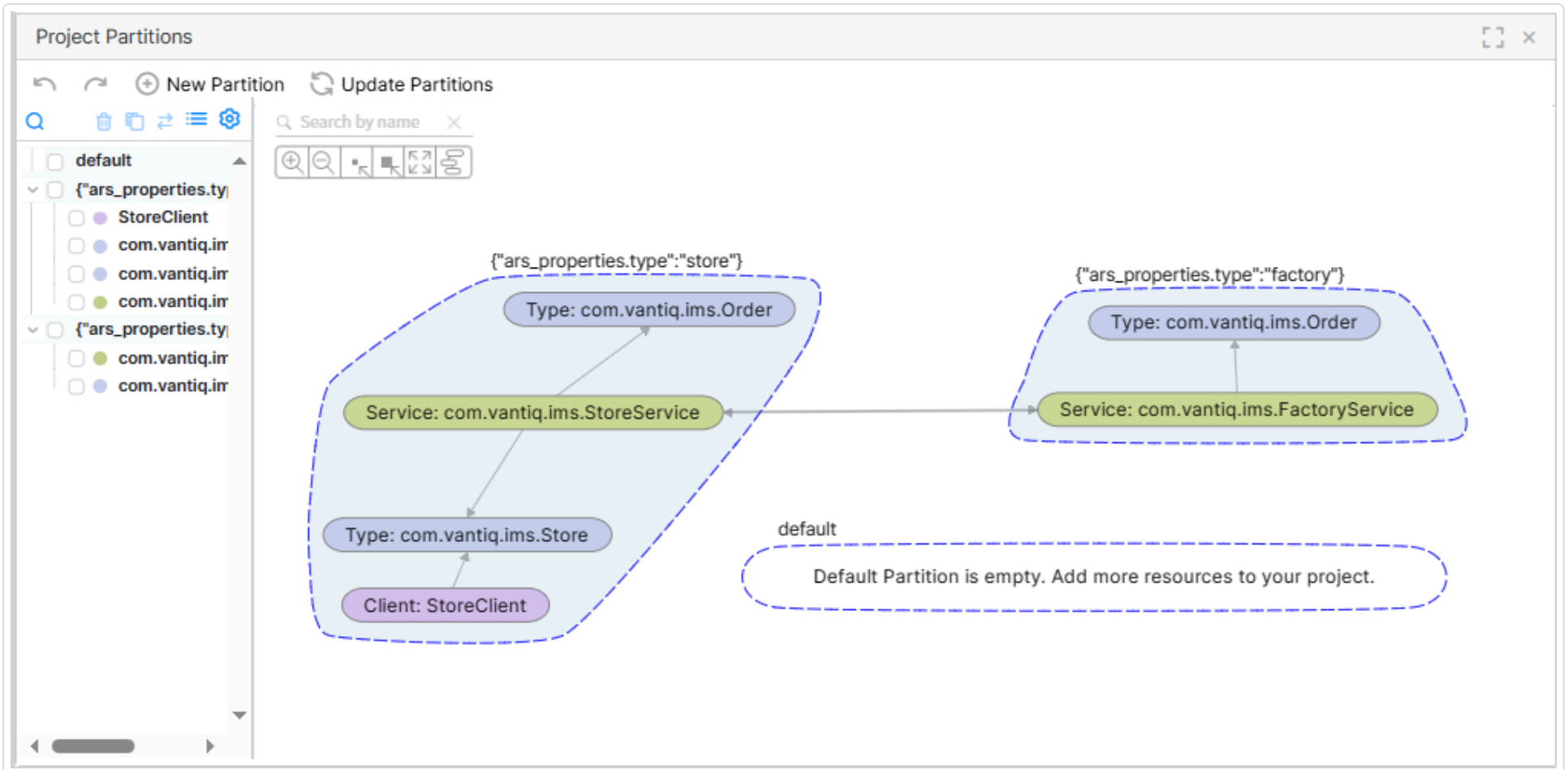
In the nodes table, use the *expand* icon button to open the constraint row to see nodes satisfying the constraint.

Use the action buttons to delete or edit node constraints.

# Project Partitions

Before you can deploy a project, you need to set up partitions. A *Partition* represents a collection of resources defined in a project that are deployed to the set of nodes in a target environment identified by the partition’s constraint(s).

Go to **Projects** > **Show Partitions** to bring up the Project Partitions pane.




Vantiq automatically analyzes all rules and procedures within the selected project and creates partitions based on the constraints used in *PROCESSED BY* statements. Related resources are automatically placed into partitions. Resources that are not related to any auto-created partition remain in the *default* partition. The same resource may appear in different partitions depending on how they were used by Rules and Procedures in your project.

If no resources use the *PROCESSED BY* statement, all resources are placed inside the *default* partition. In this case, all resources will be deployed to every node in the target environment.

You can also create partitions manually and move resources into them. If there are non-default partitions defined, then resources within the *default* partition will not be deployed.

If you create partitions manually, the same node may appear in multiple partitions. For example, if you have two partitions: `{"name": "store1"}` and `{"ars_properties.type": "store"}`, then a node named “store1” with the property type == “store” will receive resources from both partitions. Since all partitions are deployed simultaneously, resources with dependencies must be in the same partition to avoid compilation errors on the target node.

Partitions are stored in the project, so to save the partitions, save the project.

## The Graph View

The project graph draws a dashed boundary for each partition and titles the partition with its constraints shown as a string.

You can move resources within a partition by dragging and dropping them. The partition boundary will be redrawn to surround all resources within the partition.

You can also move the entire partition by dragging and dropping the partition body background.

To move a resource from one partition to another, hold down the ALT key while doing the drag and drop. The partition boundary is not redrawn while you are moving the resource (with ALT key down). During the drop phase, the resource and other resources that are linked to the moving resource are all moved to the target partition.

To copy a resource from one partition to another, hold down both SHIFT and ALT key while doing the drag and drop.

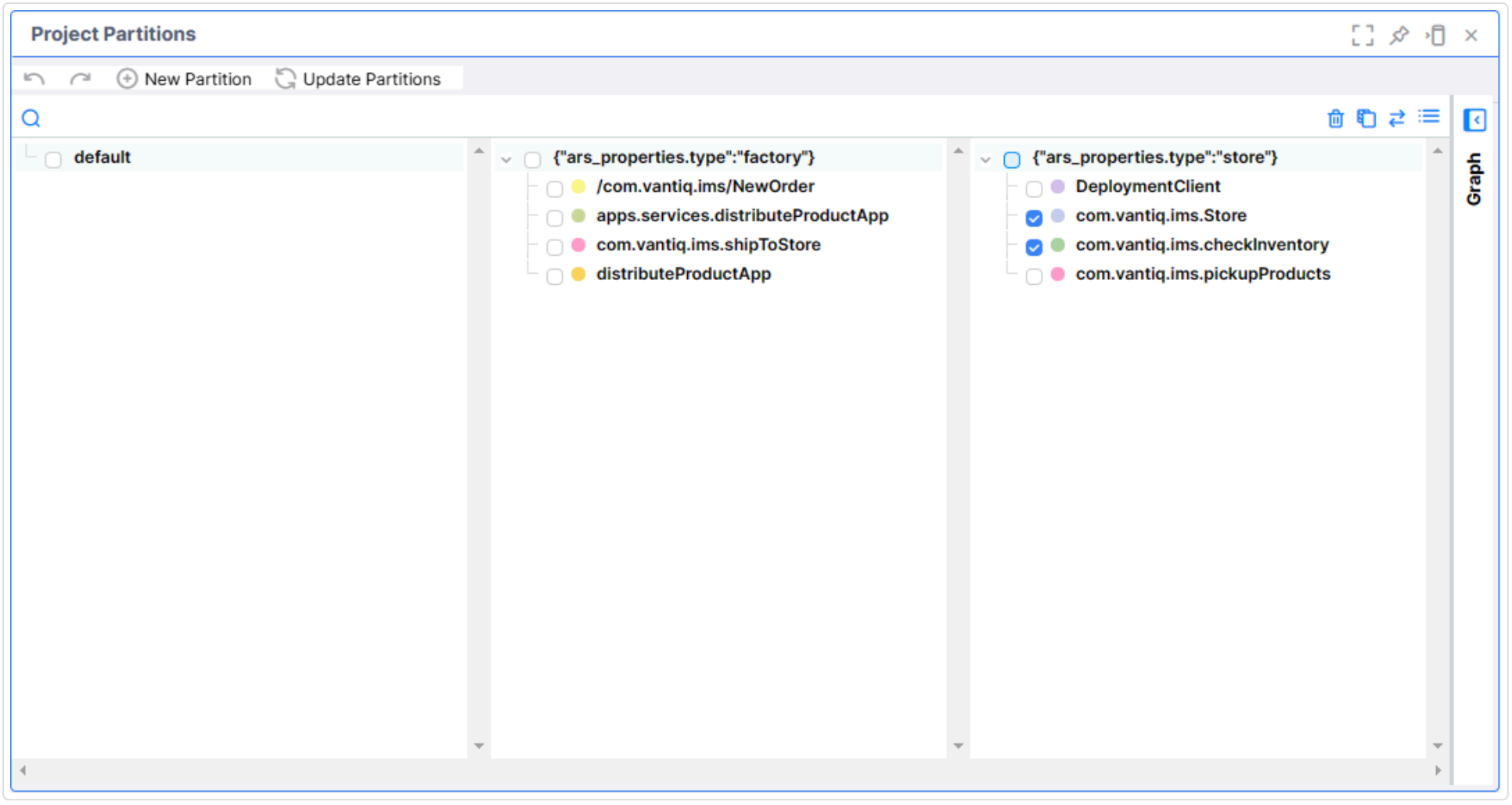
If new resources have been added to the project after the partitions are created, you will have to manually add the resource to the partitions if you want it to be deployed. Right mouse click on the partition that you want to add the new resource to in order to bring up the context menu.

Sometimes you may need to move all resources to a new partition. For example, you may have an application that runs on a single node only. You can do that by using the context menu to select “Move all resources to another partition”.

You can remove a single resource from a partition by using the context menu on the resource and selecting “Remove from partition”.

## The Tree View

The tree view list resources in a tree, grouped by partition. There is an option to group resources within the partition by resource type. When viewing tree and graph side by side, clicking a resource in the tree will make the graph center at the clicked resource.



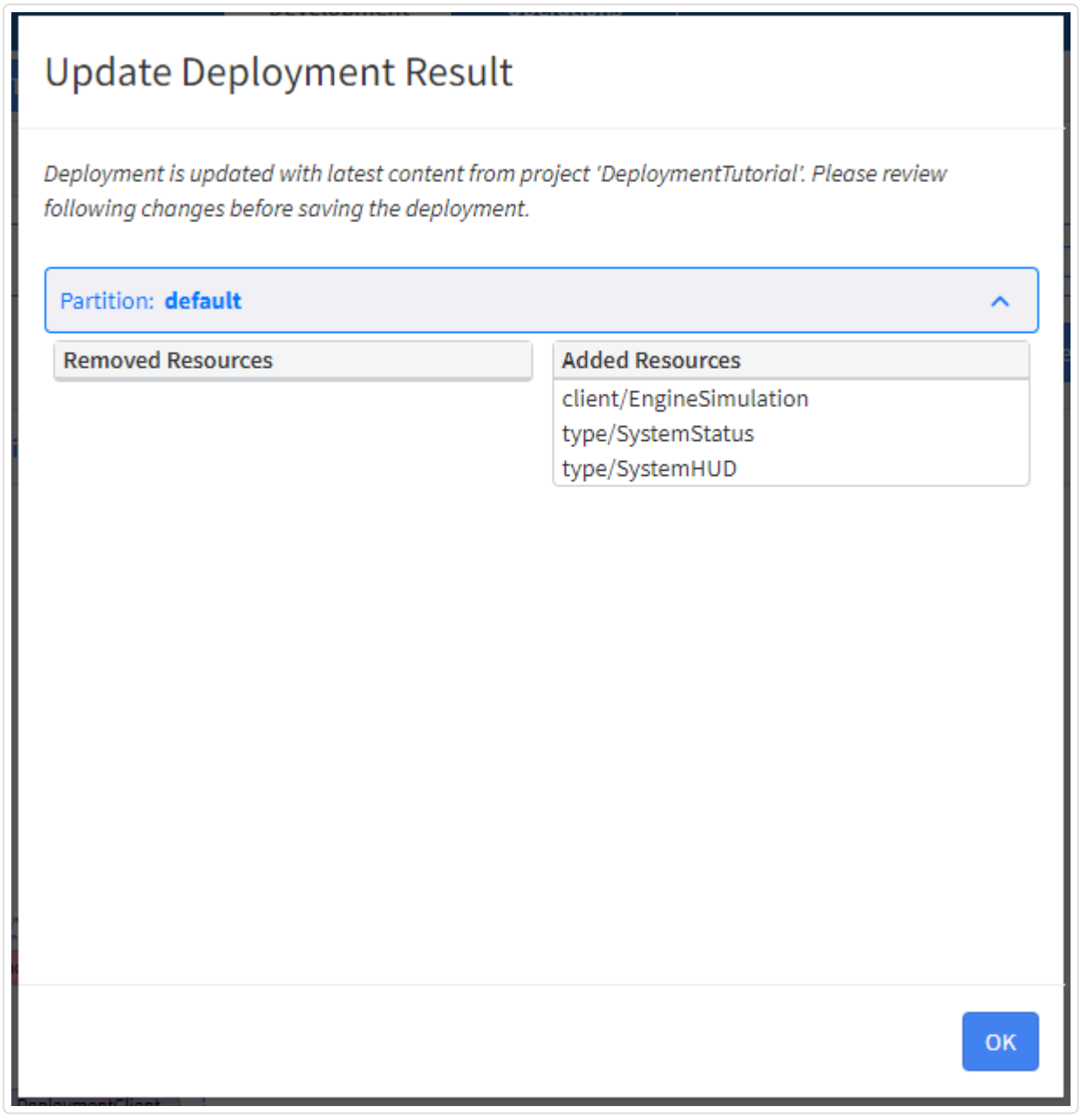

Resources within the same partition can be multi-selected so they can be moved or copied to other partitions together. Use context menu on resource nodes to perform the move/copy/delete action. You can also use toolbar buttons on top of the tree view for selected nodes. Use context menu on partition nodes to perform actions related to the partition (e.g. Add resource to partition, Edit/Delete/Move partition.)

## Update Partitions

If you have non-default partition defined, and if the project is modified, typically with new or removed resources, you will need to update the partitions using context menus on graph view or tree view. You may also use the “Update Partitions” button to update automatically.

An automatic update will perform the following actions:

- Remove resources from partitions if they are no longer in the development project.
- Perform auto-partitioning again to bring in new resources, placing them into partitions according to their relationship with existing resources in each partition.
- Display changes made to partitions in a popup dialog.




Carefully examine the changes and make any necessary adjustments before saving the Project.

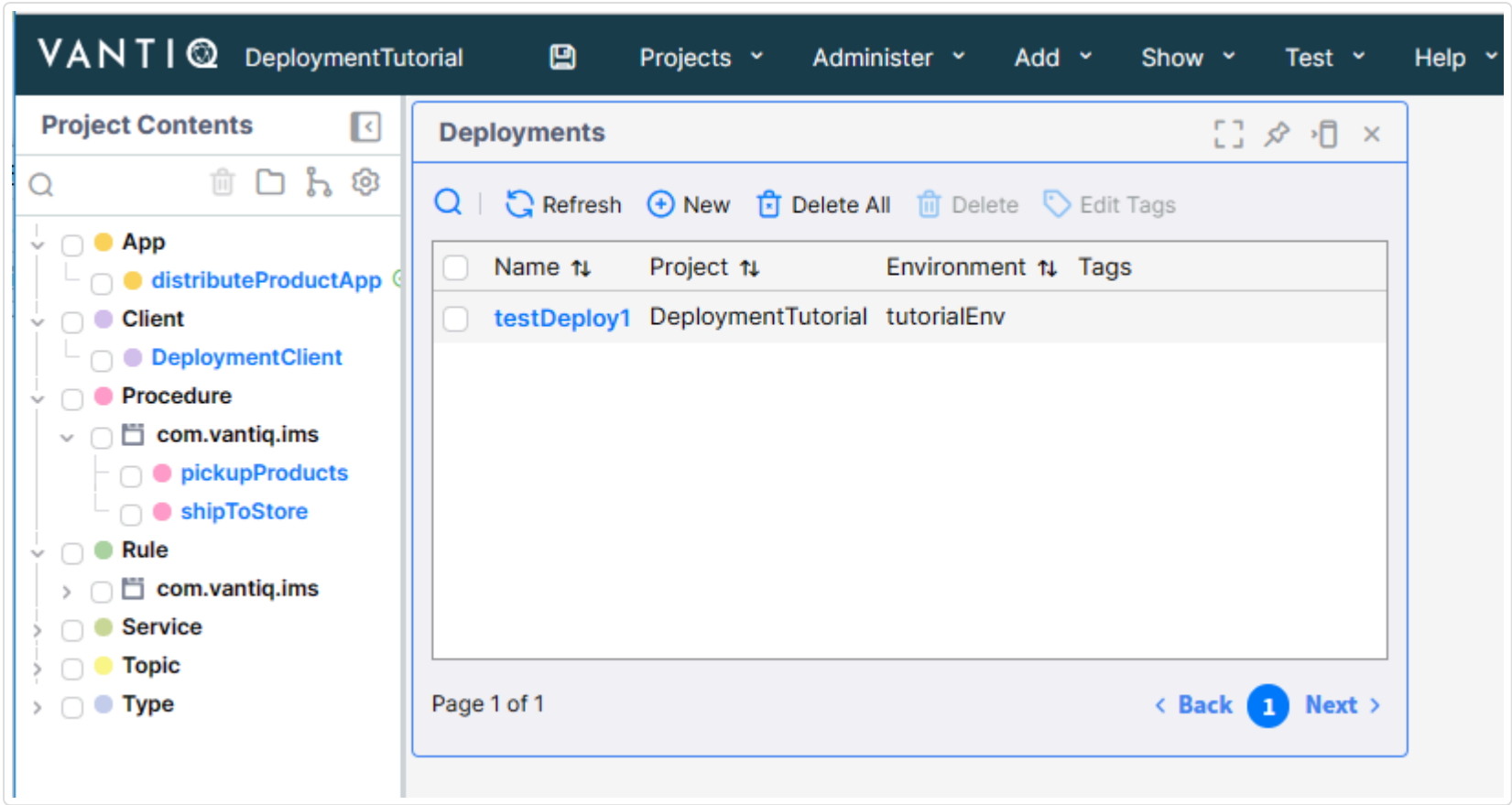
## Undo and Redo

You can undo and redo changes to partitions. The undo and redo buttons are located in the toolbar at the top of the Project Partitions pane. Actions that you can undo/redo are:

- create new partition
- delete a partition (including all nested sub-partitions)
- edit partition constraint
- move partition (change parent partition)
- update partitions
- add resources to a partition
- delete resources from a partition
- copy resources to another partition
- move resources to another partition

# Deployment

Go to **Administer** > **Deploy** > **Deployments** to bring up the Deployment List Pane.

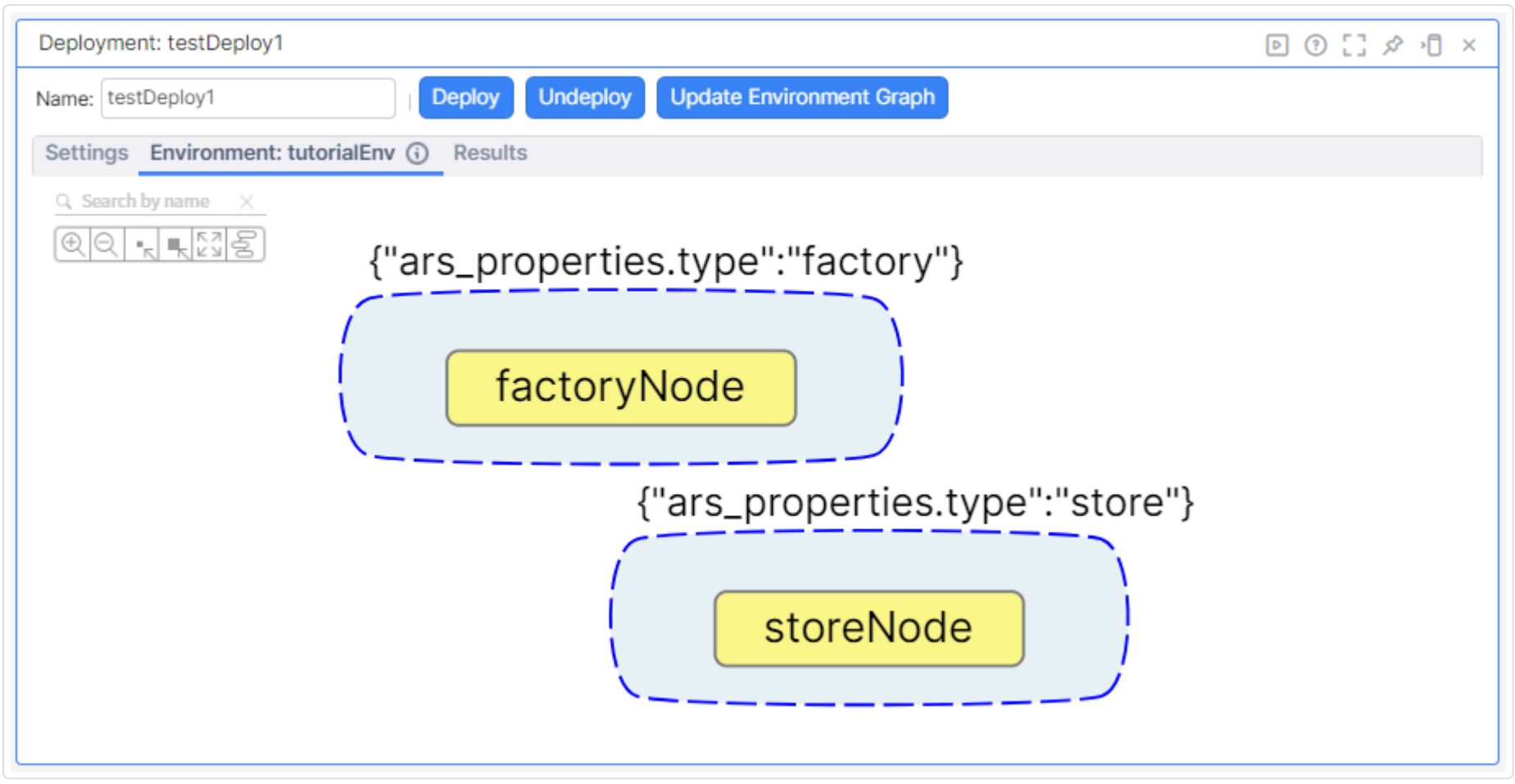



Click the **Create New** icon ("+" button) in the deployment list to create a new deployment. You must pick a project and an environment to create the deployment.

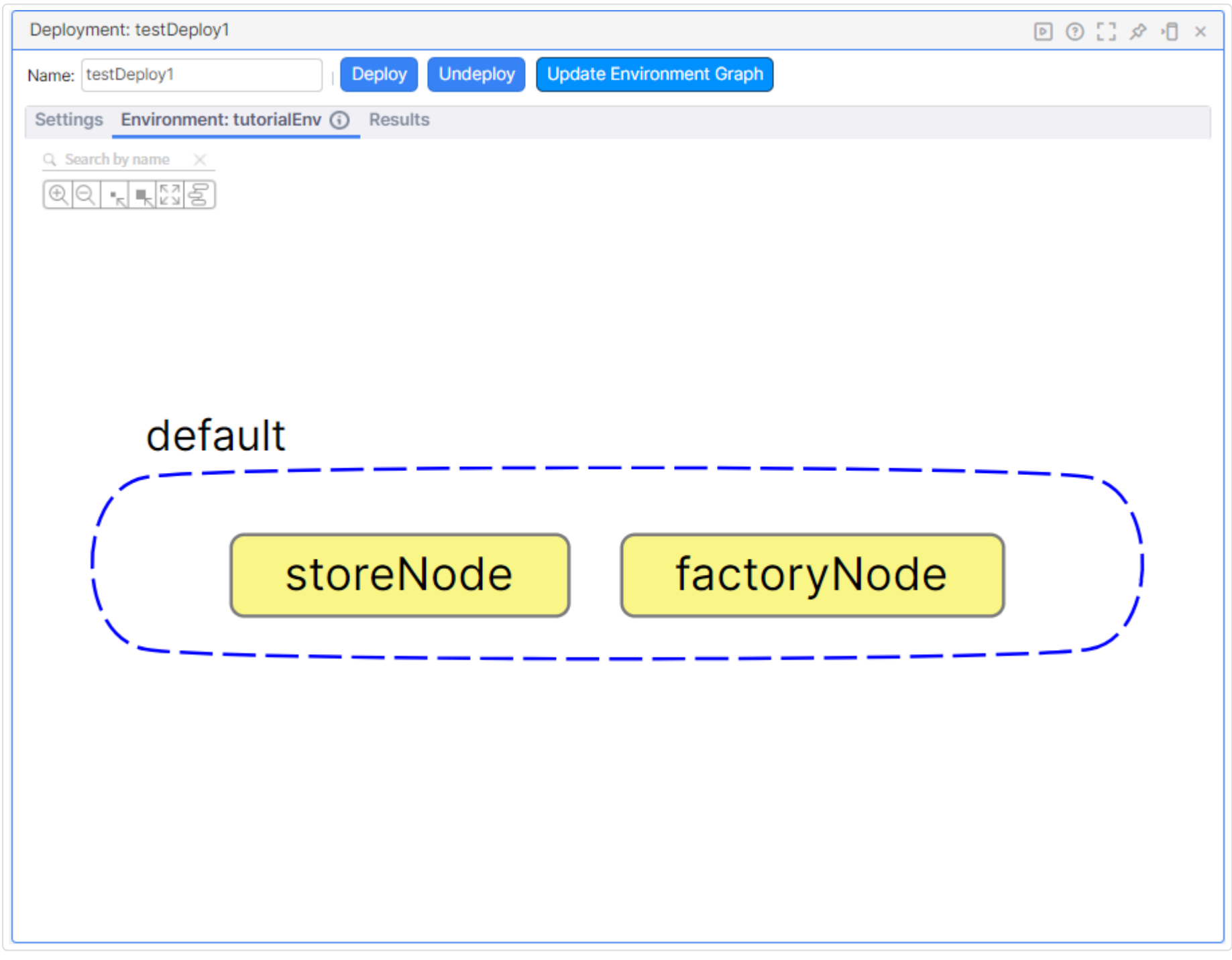

The Deployment detail pane contains 3 tabs: Settings, Environment and Results.

# Environment tab

The Environment tab shows another kind of partition graph where nodes are displayed in each partition. Only nodes belonging to the selected Environment are displayed in the graph.



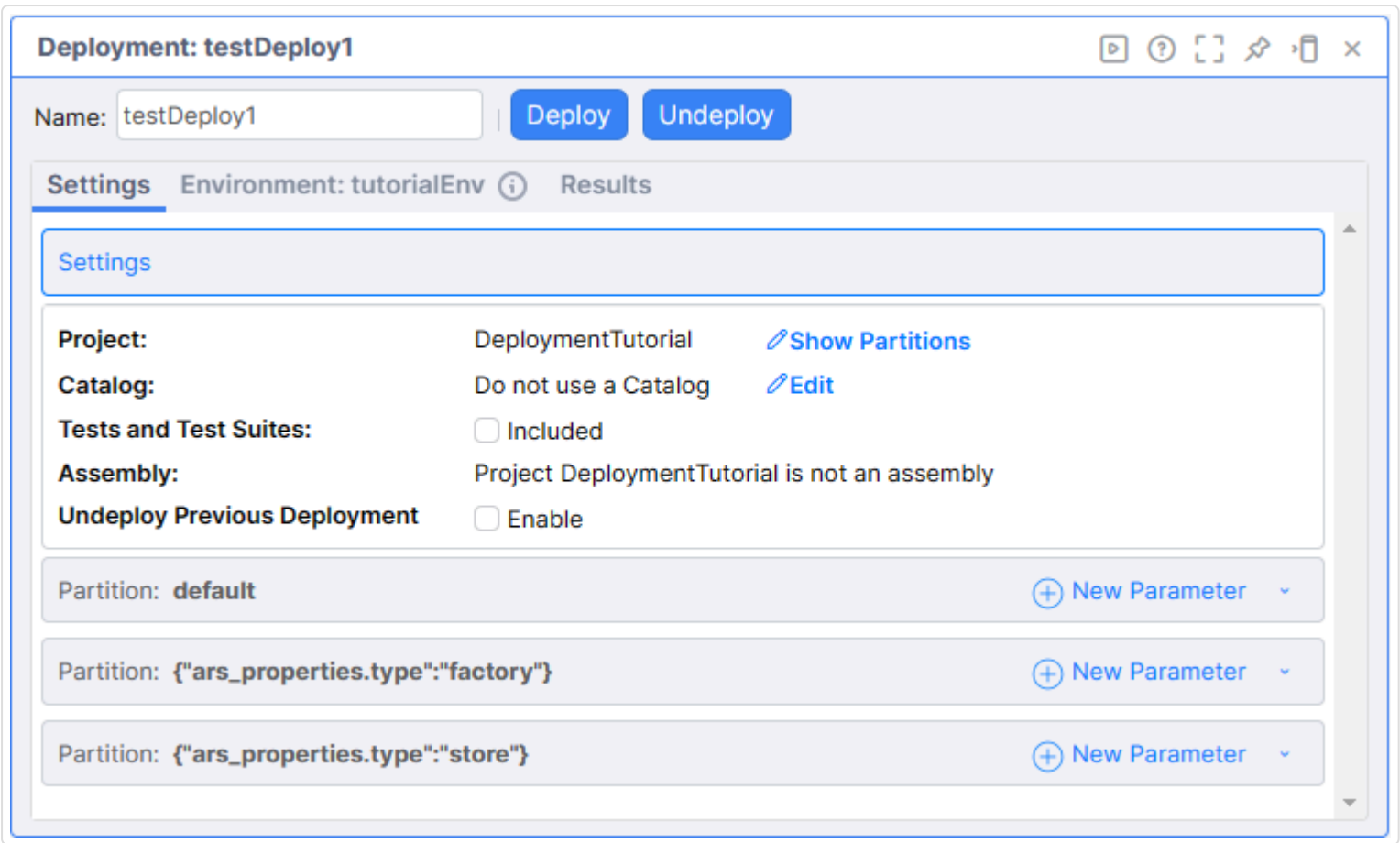

If there is only the default partition, then every node in the target environment will be shown inside the default partition. All resources of the project will be deployed to each node.




You can move partitions in the graph using drag and drop like the project graph. But you cannot add, remove or move nodes around in this graph. When the environment used by the deployment is changed, this graph is automatically updated to show the updated set of nodes. At deployment time, node definitions are also created at target nodes if deployed resources require access to those other nodes.

# Settings tab

This tab is where you specify deployment settings and create/edit partition parameters for customizing the deployment.




## Project Settings

This section displays the name of project to be deployed. If the project is currently opened in the IDE, a “Show Partitions” link can be used to view and edit the Project Partitions. If the project is not currently opened in the IDE, a “Switch project” link is provided so you can switch to that project quickly.

## Catalog Settings

When deploying a project that uses event types or services from a catalog, specify how the deployed system will use the catalog with one of the following three options:

- Do not use catalog
- Use original event catalog (access token for the event catalog is required)
- Migrate to an existing event catalog in deploy environment (both access token and URI for the event catalog are required)

After deployment, target nodes will connect to the specified event catalog and publish or subscribe to event types based on resources deployed to the nodes.

## Tests and Test Suites Settings

Use this checkbox to include/exclude Tests and Test Suites

## Assembly Settings

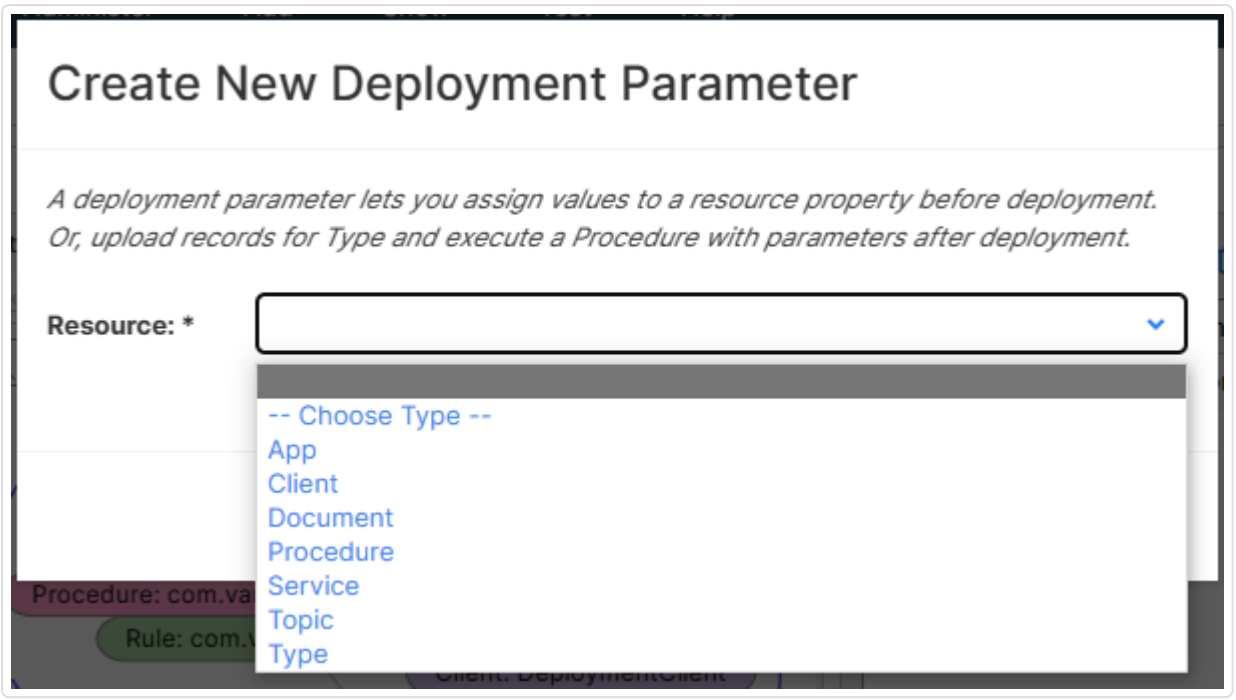
Use this checkbox to deploy the project as an assembly. If the assembly is included, there will be an extra Edit Assembly Configuration link available. Use the link to override assembly configuration during deployment.

## Undeploy previous Deployment

Use this checkbox to undeploy previous deployment before deploying the current deployment. This is useful if you have removed resources from the project or partition after it was last deployed. You can also use the “Undeploy” button to clean up the previous deployment.

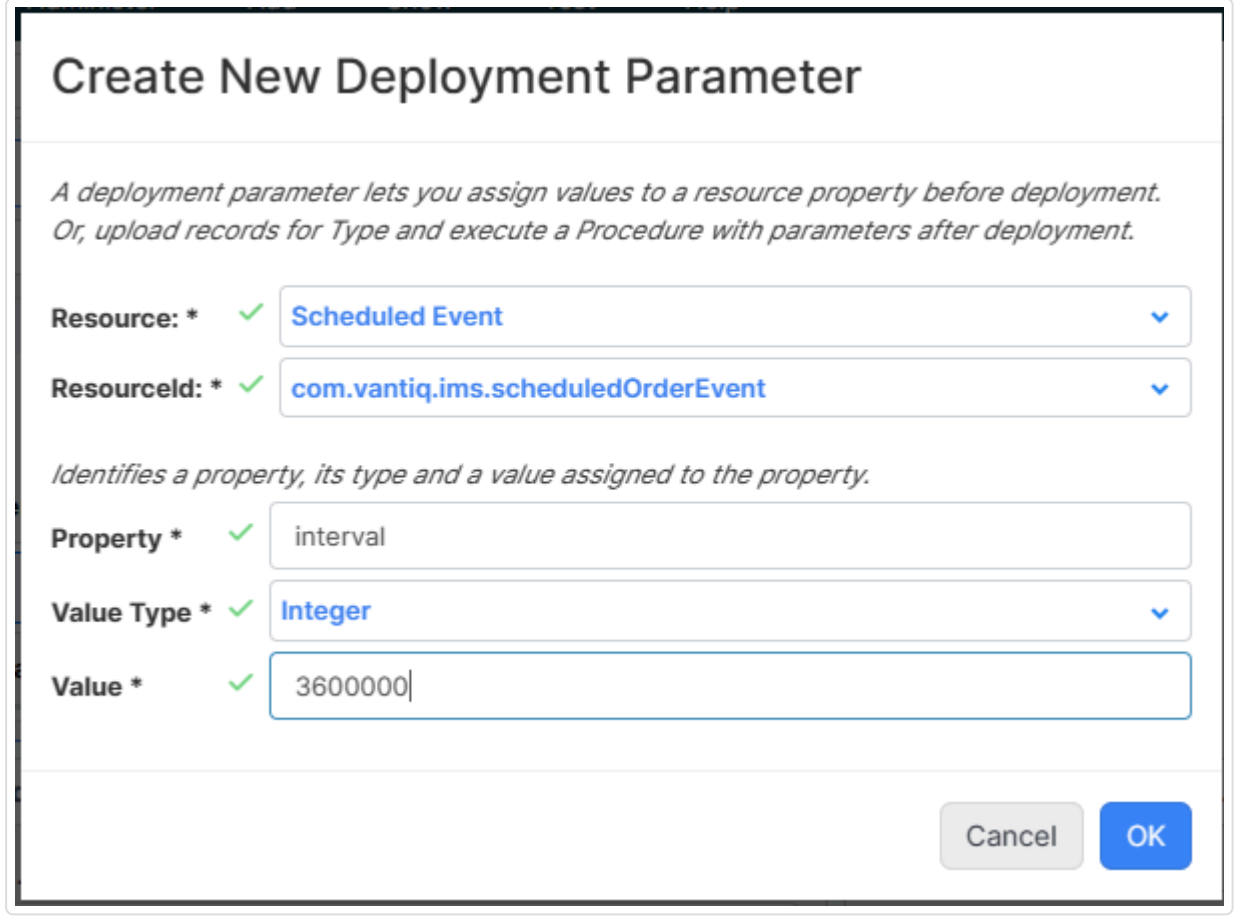
## Deployment Parameters

Parameters can be set on resources of each partition. The same resource may appear in different partitions and have different parameter values. Click the **New Parameter** icon (“+” button) on the partition title row to add a new parameter. For existing parameters, use the action buttons of each parameter to edit or remove the parameter.

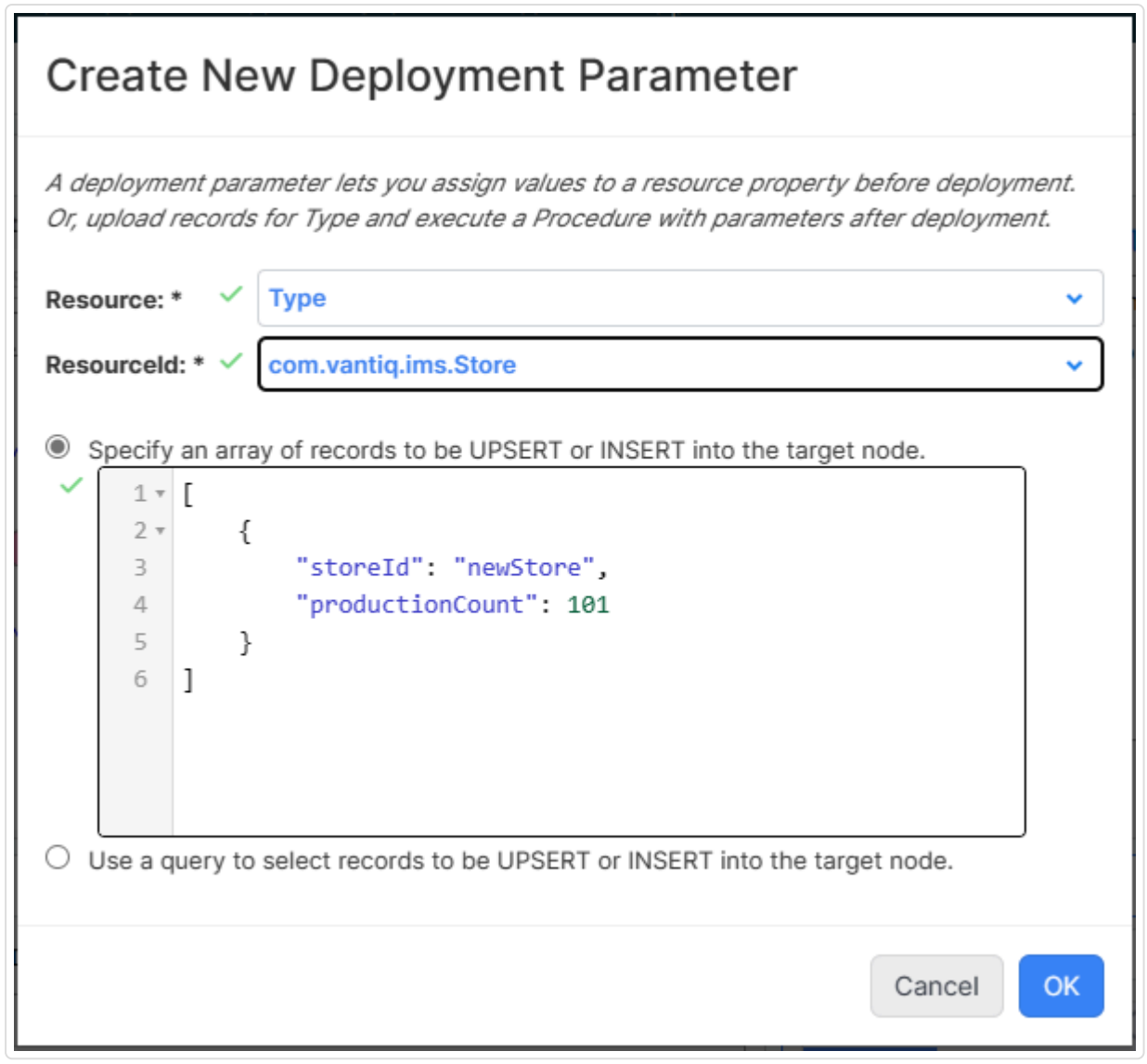



You can specify different kinds of parameter depending on the resource type.

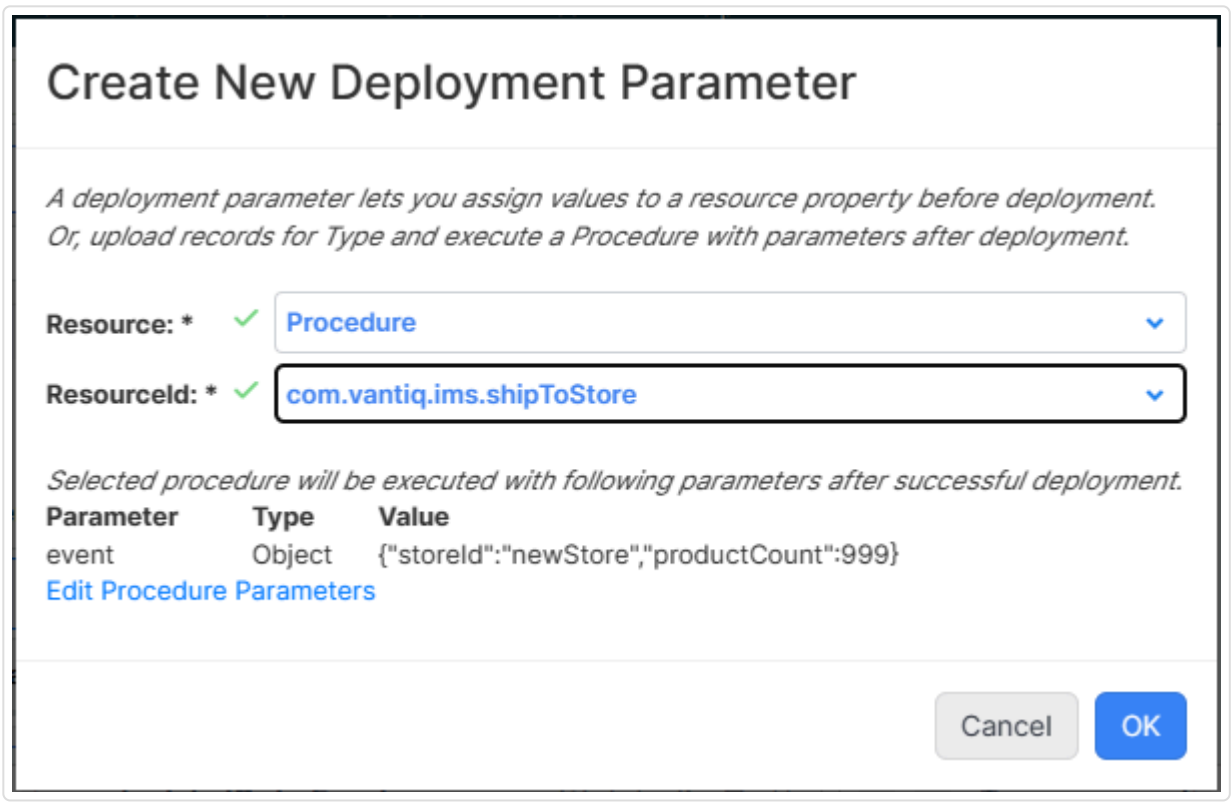
- You can set a property of a resource that is not a Type or Procedure. For example, the *interval* property of a ScheduledEvent resource.



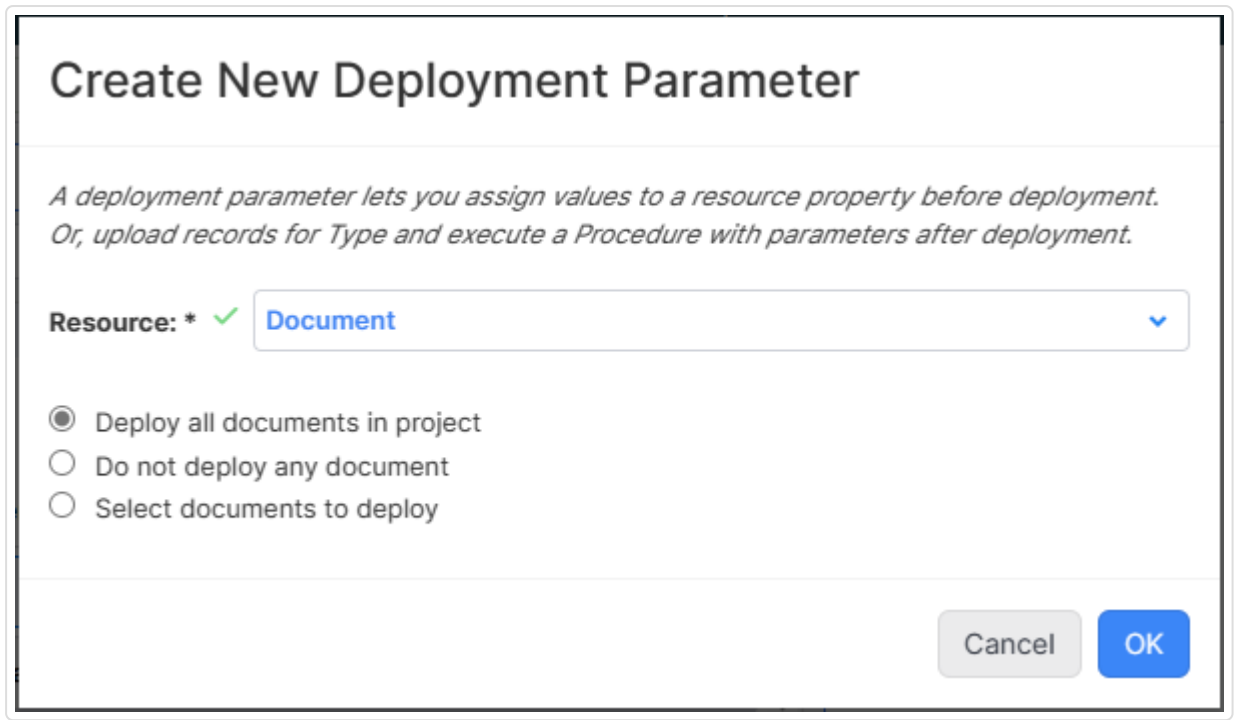

- For a Type resource, you cannot change the Type definition, but you can specify an array of records to be inserted on the target node after the Type is defined there. This is very helpful for initial application setup.




- For a Procedure resource, you can specify procedure parameters. The procedure will be executed on the target node **after** all resources within the partition have been deployed to the target node.



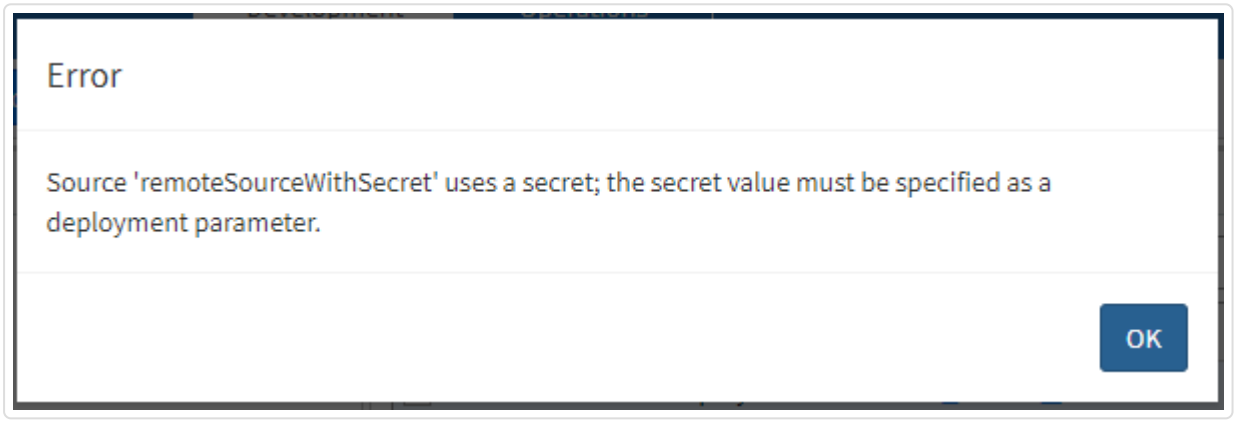
- For Documents, you can choose to deploy all documents included in the project or choose a specific list of Documents.



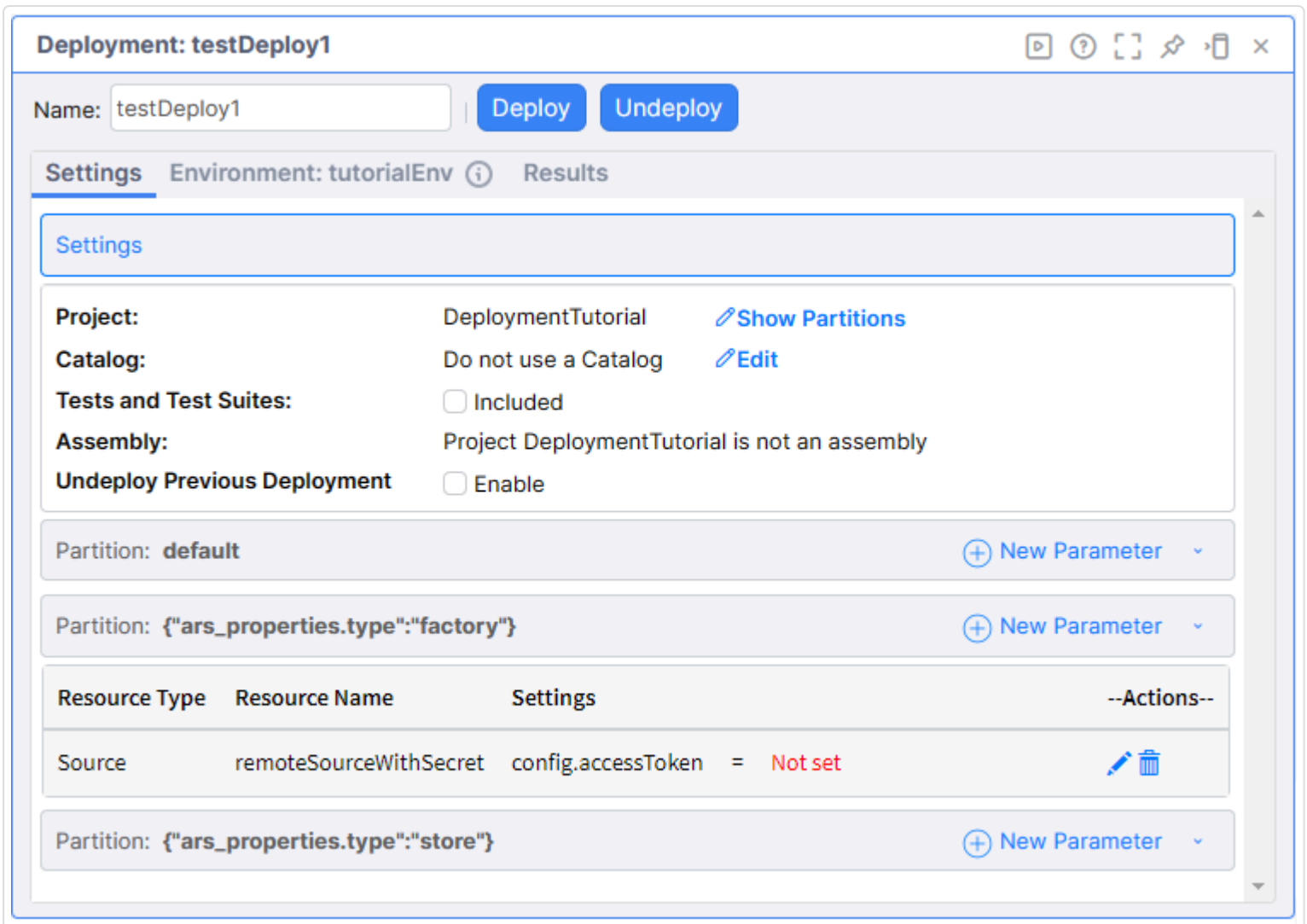

## Deployments that require a “Secret”

Secrets are storage containers for secure text that can be written once, and then never seen again by users in the namespace. Secrets can be used to configure certain properties in source configurations, which adds a layer of security to the source by hiding the secret value to users viewing the source configuration.

If your deployment uses a Secret for one of its sources then you must provide that value in order for your deployment to succeed. If you try to deploy your application and a secret is not provided then you will see the following message:

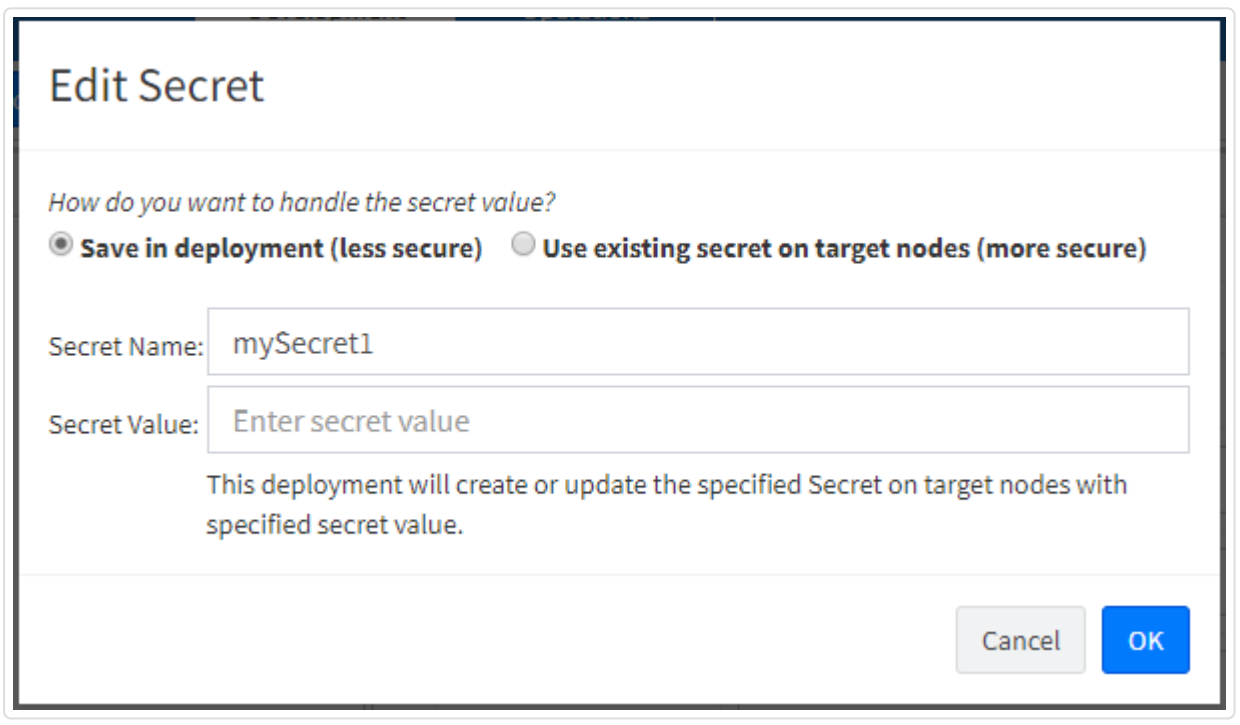



If you click on *Deploy Parameters* tab in the Deployment pane you will see a screen similar to the following:

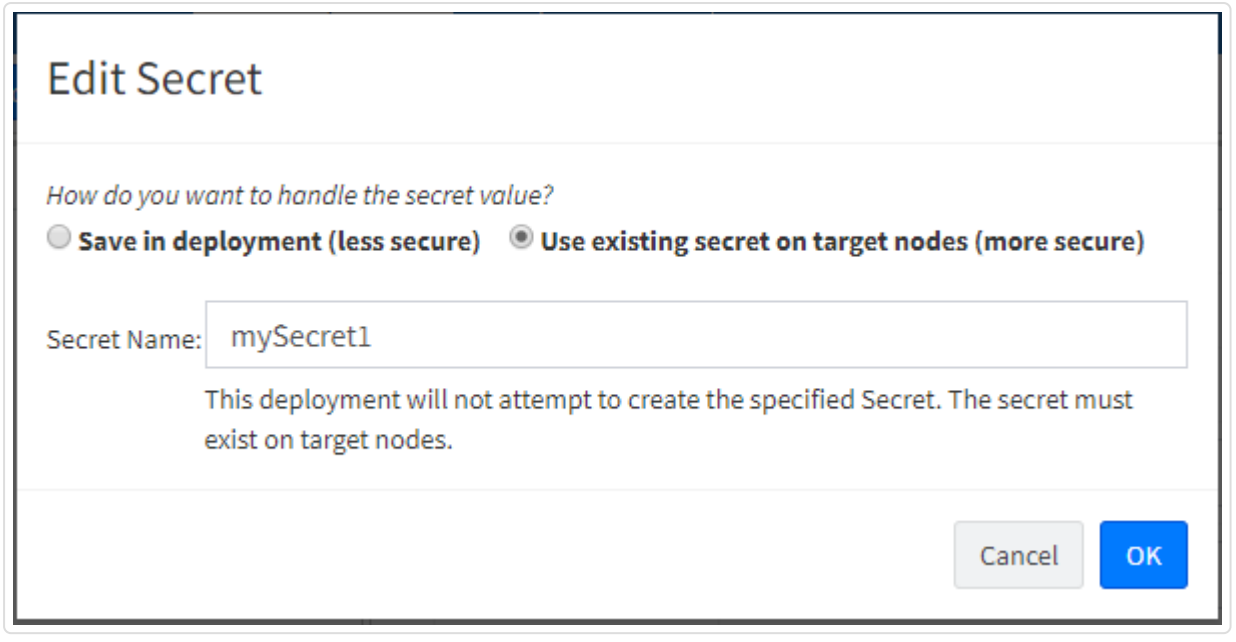



Click on the pencil icon to go to the “Edit Secret” window.
You have two options for providing the secret.

- Enter a value for the secret and save the value in your current deployment configuration. This is less secure because someone can view the secret value in the configuration. (For a way to convert this to the more secure option, see workaround below.)




- Specify the name of an existing secret from the target node. Because you are reusing an existing secret, no one will have access to the secret value. This is the more secure option.



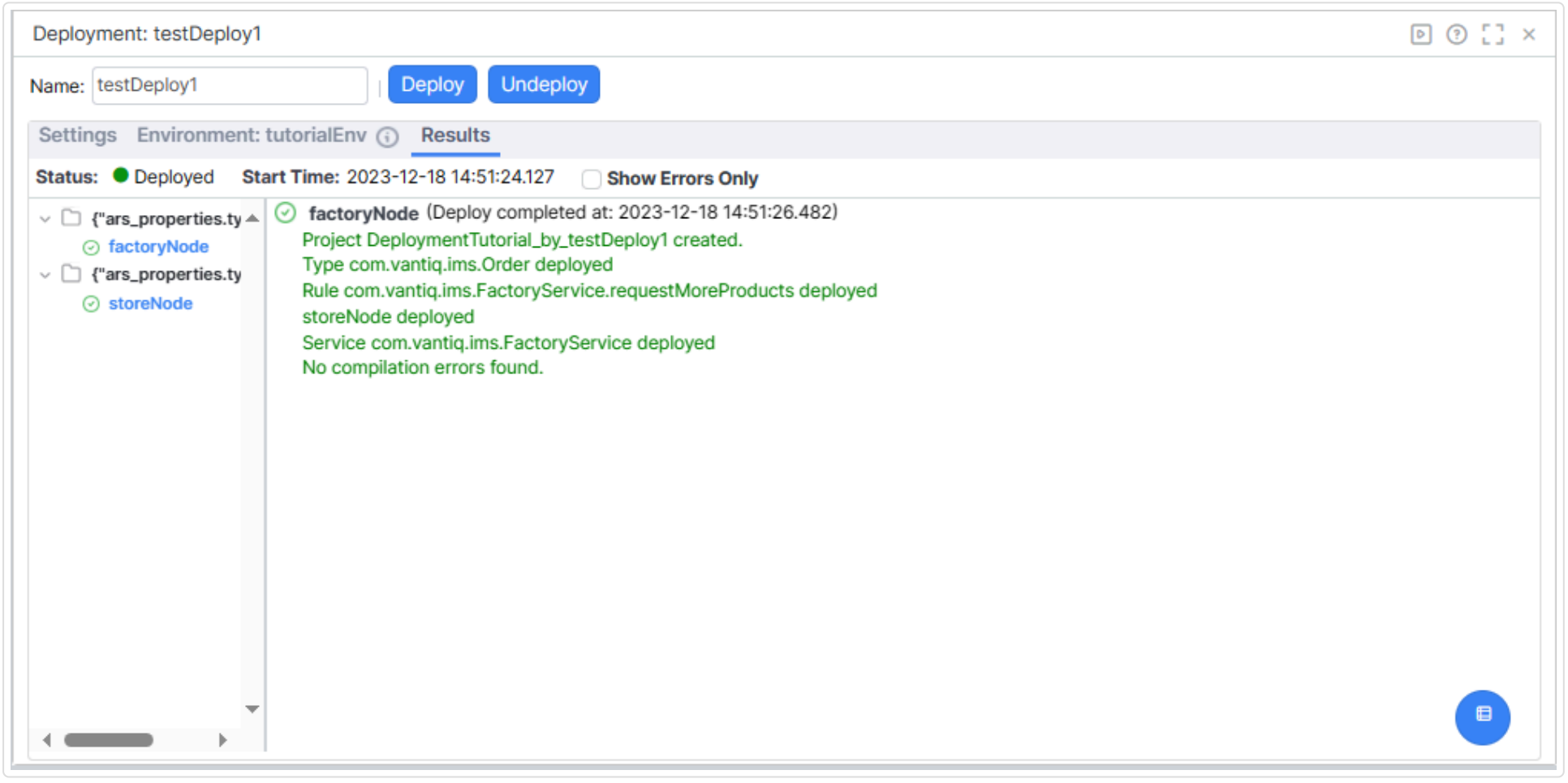

To convert the less secure option above to the more secure option, do the following workaround:

- Provide the “Secret Name” and “Secret Value” and hit “OK”.
- Deploy your application.
- Return to the “Edit Secret” window and select “Use existing secret on target node”
- Use the secret that you used above.
- Save your deployment.

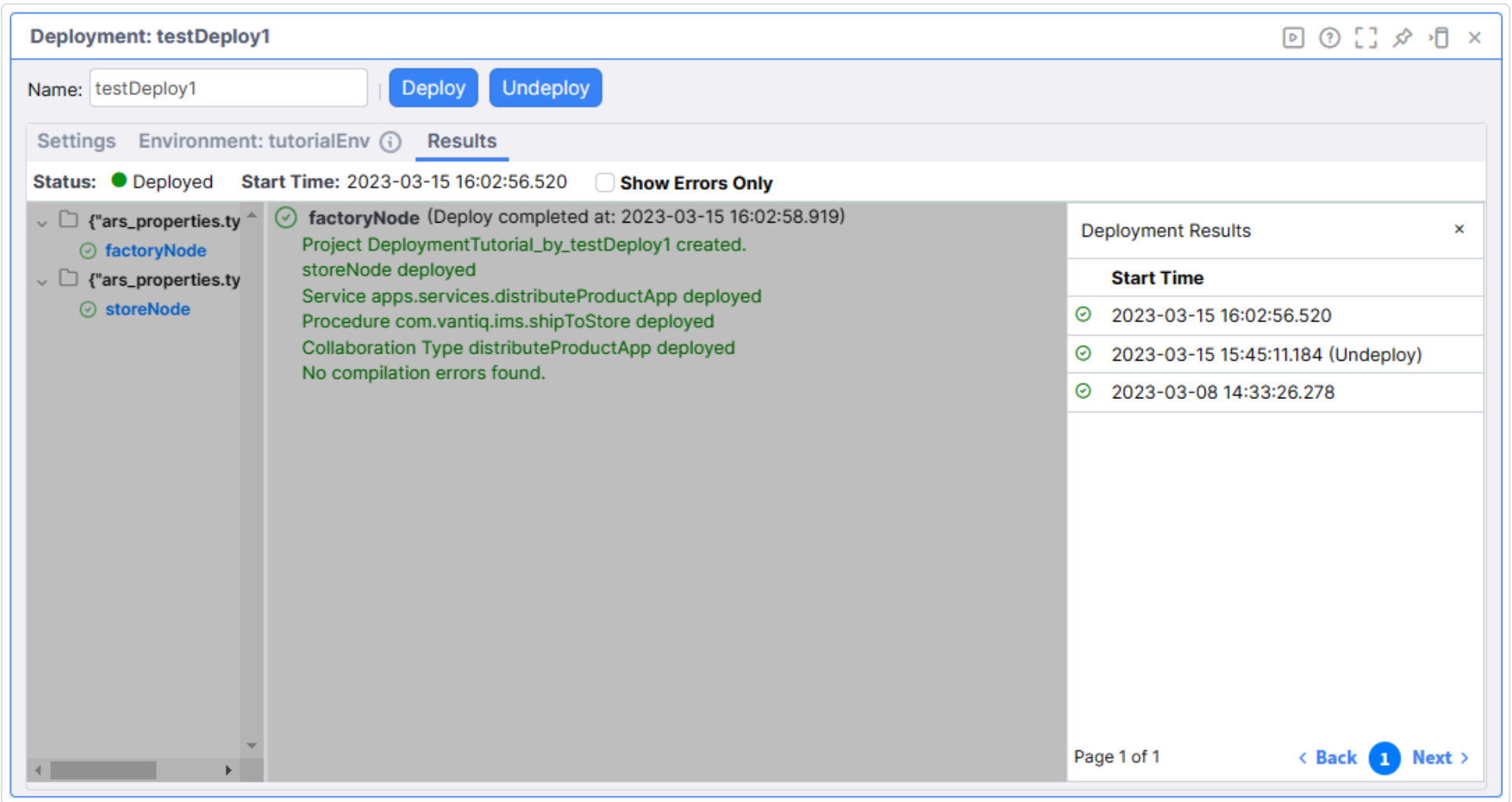
Your deployment will now be using a secret that is no longer visible when viewed through the Vantiq UI or viewed from a project export.

## Deploy Results tab

This tab is where you see the results of the deployment operation. It contains the overall status and the starting time of the deployment. For each partition, there is an individual status for each node. Detailed messages are displayed for each node. Use the **Show Errors Only** checkbox to filter the results by errors.



Click the lower right corner **Results List** button to look up historical results.




## Verify application after deployment

A successful deployment means that all resources have been copied to the target nodes. Note that users still need to verify how the application performs on each target node.

A project is automatically created on the target node to contain the deployed resources. The project name format is “{developmentProjectName}_by_{deploymentName}”. For example, “DeploymentTutorial_by_testDeploy1”.

# Deployment Advanced Features

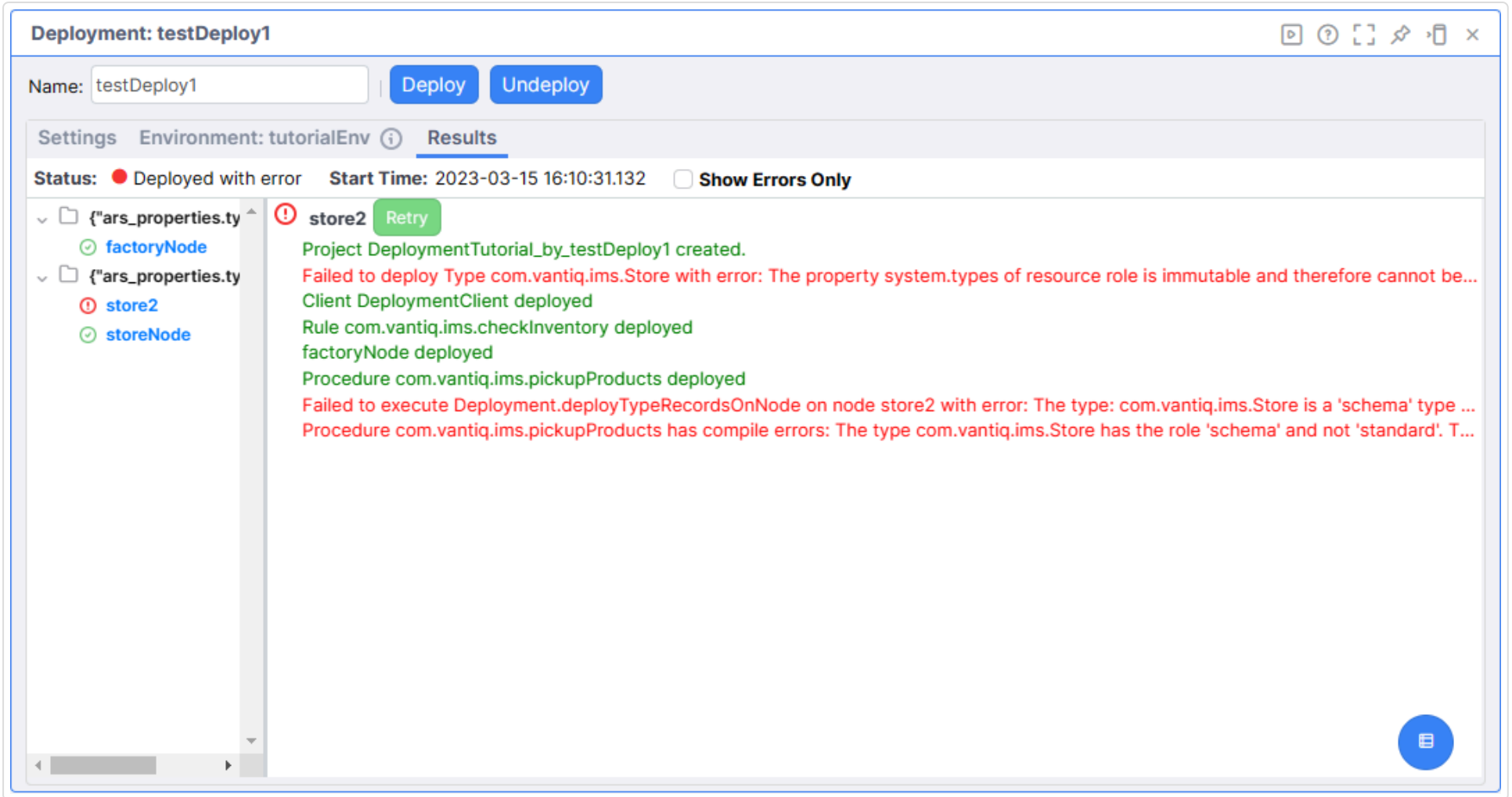
## Undeploy

To clean up resources from deployed nodes, use the “Undeploy” button. Resources from the previous deployment are deleted from the target node. Check the Results tab to see if there were any errors during undeploy.

## Redeploy on a failed node

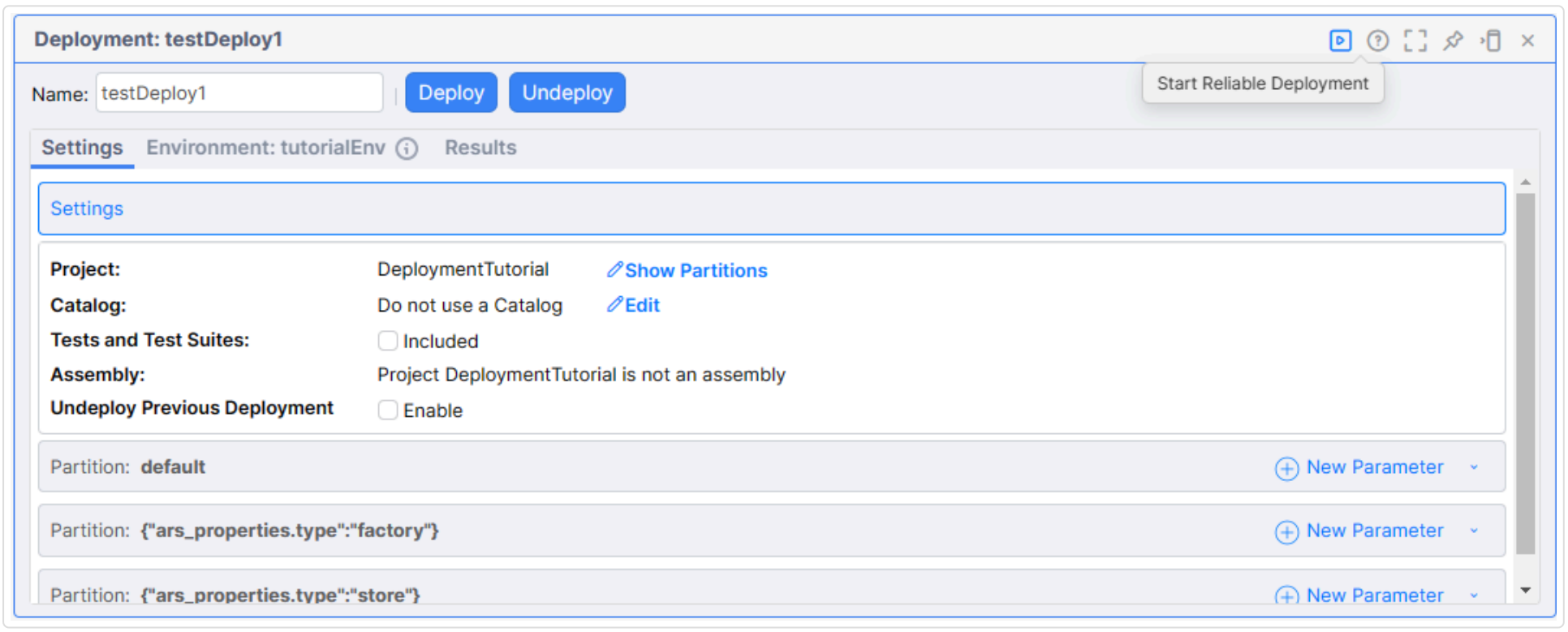
In an environment with multiple nodes, there may be errors deploying resources on one node, but not all nodes. If the Results Tab shows that a node had errors during deployment, you can use the “*Retry*” button to try deploying again to that specific node. “*Retry*” is only available on the latest deployment results.

Upon retry, only status for this node gets updated.




# Reliable Deployment

In a large environment, not all nodes are online at the same time and new nodes may be added to the environment at a different time. It would be simpler to issue deployment commands once and know they are going to eventually succeed. This can be done using the Reliable Deployment feature.
Once the deployment is in reliable deploy mode, it will try to deploy to failed and newly added nodes every hour. To start the reliable deployment, click the “Run” button on the deployment pane’s title bar.




Results of reliable deployment are also listed in the Results tab just like a normal Deploy operation.

A deployment in Reliable Deployment mode is read-only.

The deployment remains in Reliable Deployment mode even if the Deployment pane is closed.

Click the “Stop” title-bar button to stop Reliable Deployment.

# Deploying the same application to different environments

Once you have successfully deployed and tested your application on a test environment, you may want to deploy it to another environment.

To do that, you first define a new environment and select nodes into it. Then create a new deployment using the same project and the new environment. The Project Partitions graph is exactly the same. The Environment graph is similar but shows a new set of nodes. Deployment settings are set to use the default values. There is no deploy parameter defined on the new deployment.

If you have parameters defined in the original deployment, it will be easier to use the “Duplicate Deployment” context menu on the Deployments List because duplicating deployment copies settings including deploy parameters. Settings and parameters may need to be adjusted for the new environment.