RPA Design and Development

v4.0

UiPath

Lesson 20 Orchestrator

Exam Topics - Orchestrator

1. Describe Orchestrator Entities (Robot, Folder, Package, Process, Job, Heartbeat), Tenant Entities (User, Machine, License, Webhook, Alerts) and Folder Entities (Assets, Storage Buckets, Queues, Triggers, Credential Stores)

2. Use Tenant Entities (User, Machine, License) and Folder Entities (Assets, Storage Buckets, Queues)

3. Provision Robots.

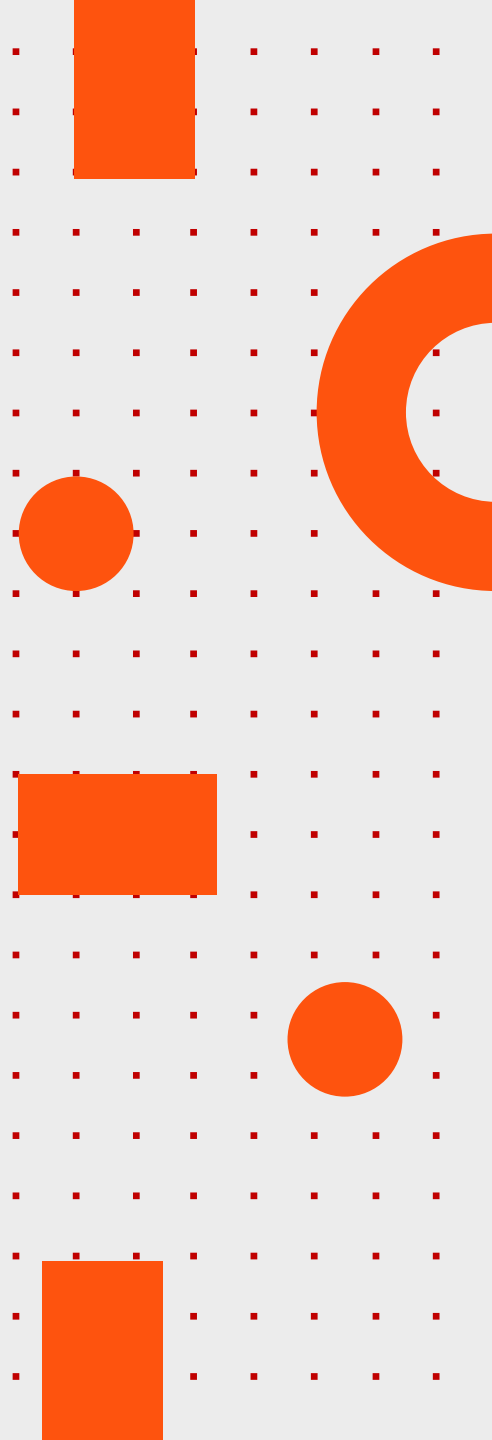
4. Use Personal Workspaces.

5. Define Roles and Permissions.

6. Use Orchestrator Logging features.

Orchestrator Overview

- Introduction to Orchestrator
- Orchestrator Capabilities



Orchestrator Overview

www.cloud.uipath.com

- Orchestrator is the component of the UiPath Platform for managing automations, robots, and the related entities.
- Although it comes with different cloud and on-premises delivery options, including persistence, high availability, and disaster recovery, users can access it through a simple web interface.
- Orchestrator offers role-based access control and a structure of tenants and folders to replicate organizational structures.
- Users can run the automation workflows developed in Studio and published to Orchestrator using the unattended robot workforce.
- Orchestrator is used to manage and distribute licenses, as well as to store automation resources.

Orchestrator Main Capabilities

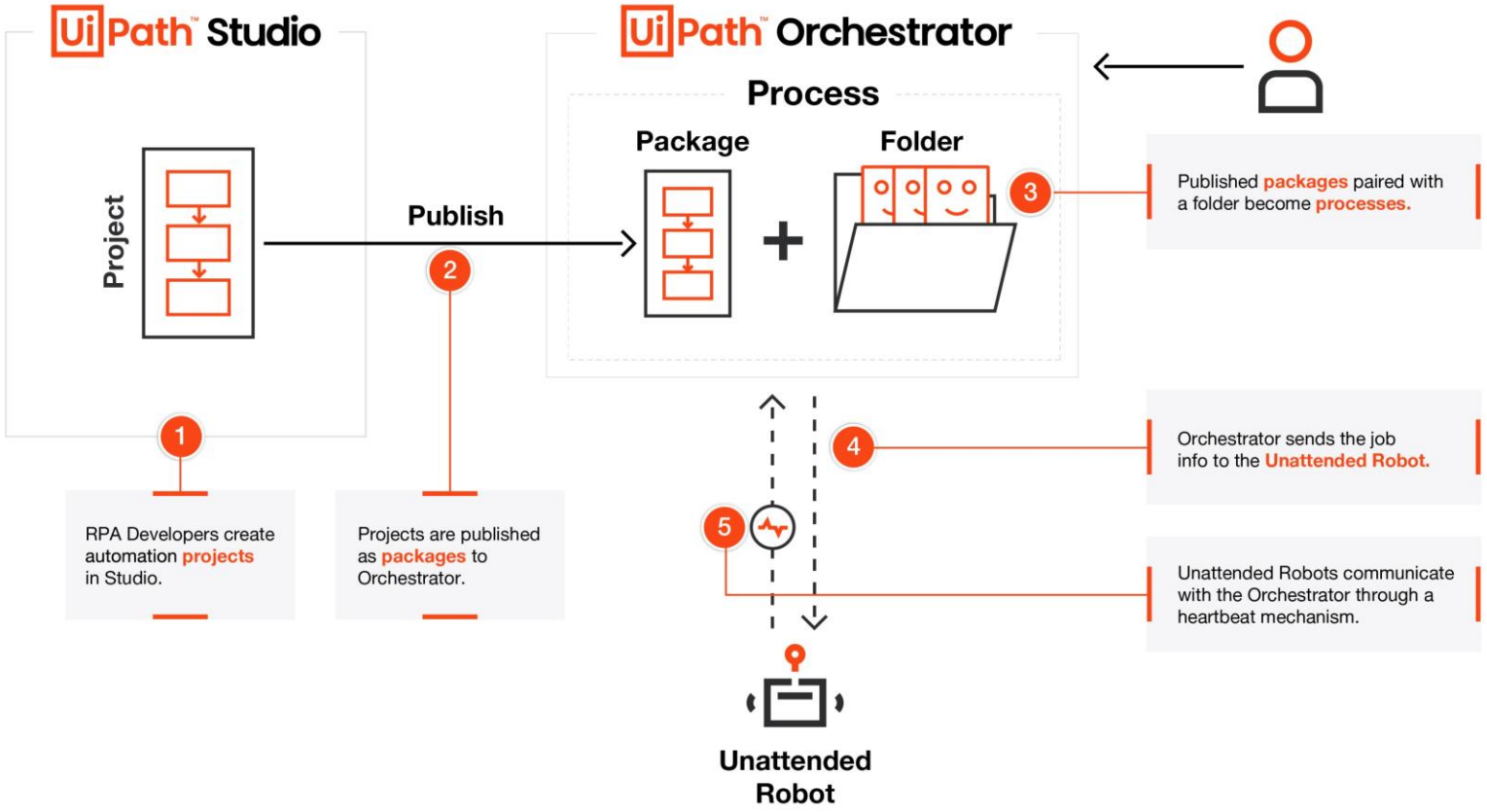
- Orchestrator creates and maintains the connection with robots and attended users
-
- Control and license distribution: Orchestrator also enables the creation, assignment, and maintenance of licenses, roles, permissions, groups, and folder hierarchies
- Running automation jobs in unattended mode: as presented earlier, it enables the creation and distribution of automation jobs in various ways, including through queues and triggers
- Automation storage and distribution: Orchestrator is the environment enabling the controlled storage and distribution of automation projects, assets, and credentials, as well as large files used in automations
- Monitoring: with Orchestrator, admins are able to monitor jobs and robots. It also stores logs for auditing and analytics
- Inter-connectivity: finally, Orchestrator acts as the centralized point of communication for third-party solutions or applications.

Orchestrator Main Capabilities

- Orchestrator creates and maintains the connection with robots and attended users
- Control and license distribution: Orchestrator also enables the creation, assignment, and maintenance of licenses, roles, permissions, groups, and folder hierarchies
- Running automation jobs in unattended mode: as presented earlier, it enables the creation and distribution of automation jobs in various ways, including through queues and triggers
- Automation storage and distribution: Orchestrator is the environment enabling the controlled storage and distribution of automation projects, assets, and credentials, as well as large files used in automations
- Monitoring: with Orchestrator, admins are able to monitor jobs and robots. It also stores logs for auditing and analytics
- Inter-connectivity: finally, Orchestrator acts as the centralized point of communication for third-party solutions or applications.

Orchestrator Overview

Core Components: **Unattended Robots**



UiPath Academy

Chapter 13:

Orchestrator Overview for Automation Developers

Introduction to Orchestrator

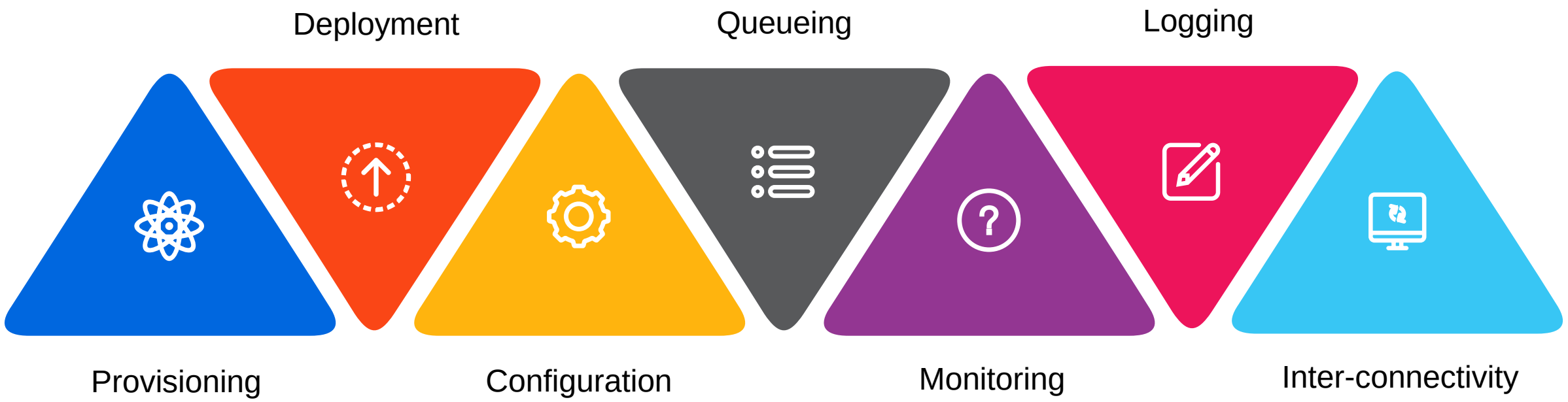
Introduction to Orchestrator

Orchestrator is used to manage the creation, monitoring, and deployment of resources in an environment. It is a web application that:

- Controls and monitors the productivity of robots
- Deploys the workflows to the robots
- Manages the robots by scheduling them at any time

Orchestrator Capabilities

The important capabilities of Orchestrator are:



Resources

Topic	Link
Orchestrator Standalone User Guide	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/introduction

Orchestrator Entities: Tenants and Folders

- A single Orchestrator instance can be split into multiple Tenants.
- Each tenant in an organization can be further subdivided and organized into Folders.
- Tenants are designed for the purpose of complete isolation of all Orchestrator entities (i.e., Robots, Assets, Queues, etc.) between these segregated instances of your deployment, all without having to maintain multiple Orchestrators.
- Modern folders provide multiple features such as automatic robot management, hierarchical structures, and fine-grained role assignment for users.

Consider a typical large company, in which both the data and the business processes are typically separated between divisions like Sales and Finance. But then, the subdivisions would have some of the data or some of the processes separated, at the same time, sharing others.

Processing_Bot_Line_Left.png

In Orchestrator, some of the entities exist in the tenant context, while others exist in the folder context.

Orchestrator Concepts
Robot (Orchestrator)

- This is an execution host that runs automation processes published in Orchestrator, as jobs

- In Orchestrator, a robot entity represents an image or the Robot component, controlling its connection and capabilities

- The robot entity exists only if it is defined in relation to a user/ robot account in Orchestrator

Orchestrator Concepts : Folder

- Folders enable the separation and hierarchical organization of automation entities (processes, queues, assets) and the fine-grained configuration of roles and permissions

- You can create as many sibling folders as you want on a given level. The maximum hierarchy depth is 7, meaning you can have a maximum number of 6 nested subfolders under a first-level folder

- Folders help replicate the organizational hierarchies, with the separation of automated processes between teams, segregation of process data, and access control for users. At the same time, when it makes sense, they allow sharing of the resources and assets.

Orchestrator Concepts : Package

- A project developed in UiPath Studio that is published to Orchestrator as a **NuGet package**.
- Multiple versions of the same project can be stored and used
- Packages can also be manually uploaded to Orchestrator
- Additionally, by viewing the versions for a package you can download it from Orchestrator

Orchestrator Concepts : Process

- It is a version of a package that's been allocated to a certain folder

- Given that most processes use a queue, asset, or storage bucket, the Package Requirements tab, for when adding a new process, makes it easy to identify which entities your package is using and if they are missing from your folder

Orchestrator Concepts : Job

- A job represents the **execution of a process on a UiPath Robot**
- You can launch the execution of a job in either attended or unattended mode
- You cannot launch a job from Orchestrator on attended robots, unless for debugging purposes using personal workspaces, and they cannot run under a locked screen.
- Attended jobs can be triggered from the UiPath Assistant or the Robot Command Line Interface.
- Unattended jobs are launched from Orchestrator, either directly on the spot from the Jobs or Processes page or in a preplanned manner through triggers on the Triggers page
- The Jobs page represents the jobs control center, where you can monitor launched jobs, view their details and logs, and stop/kill/resume/restart a job

Orchestrator Concepts : Heartbeat

- Attended and unattended robots send a heartbeat to Orchestrator every **30 seconds**

- This signals to Orchestrator that the connection is working

UiPath Academy

Chapter 13:

Orchestrator Overview for Automation Developers

Orchestrator Entities, Tenants and Folders

Tenant entities

- **Robots are tenant entities**. This means that they can be allocated to multiple folders in that tenant
- Using roles and permissions, the way robots work with each of the folders can be customized
- Packages are published to Orchestrator using feeds
- The feeds can be configured to be at tenant level, or at folder level
- A package published to the tenant feed can then be used in a process in any of the folders
- If a package is published using a folder feed, it can't be used for processes in other folders

Tenant entities - User

- Both human users and robots are uniquely identified with users in Orchestrator

- On the Accounts & Groups page, in Automation Cloud, you can define local user accounts, robot accounts, and local groups for your organization

- The level of access and the actions that your users can perform is controlled using two elements:
  - accounts, which establish the identity of a user and are used to log in to your UiPath applications
  - roles, which are assigned to accounts in order to grant them certain permissions within the UiPath ecosystem.

- Accounts are not created or managed in Orchestrator, only roles and their assignments are

Tenant entities – Machine (Orchestrator)

- Machines are Orchestrator entities corresponding to the workstations where human users and robots work

- Using API keys, they enable the connection between the physical or virtual machines and Orchestrator

Tenant entities – License

- The right to use Studio and/or Robots, both attended and unattended, is done through licenses.

- Licenses exist at tenant level, from where they get distributed to users, and consumed when the machines connect to Orchestrator

Tenant Entities – Webhook

- Webhooks facilitate the communication between Orchestrator and other applications at API level

- These are mapped at tenant level, which means they cannot be differentiated between folders and will provide information for the entire tenant

Tenant Entities – Alerts

Alerts are real-time notifications related to robots, queue items, triggers, and more.

You can receive alerts for all the folders to which you have access.

You can view alerts quickly in the notification panel or review them in detail on the Alerts page.

Alerts can refer to:

- Different severities of events (info or warning, for example)
- Different components (Triggers, Actions, and more)

UiPath Academy

Chapter 14:

Working with Orchestrator Resources

Working with Orchestrator Resources in Studio

Folder Entities

- A folder is a storage area that helps keep your projects separate.
- From the entities defined at the beginning of the lesson, processes and jobs are folder entities.
- Packages depend on feed configuration.

Folder Entities - Assets

- An asset is a piece of data stored in Orchestrator for the use of robots. There are four types of assets:
  - **Text** - stores only strings (it isn't required to add quotation marks).
  - **Bool -** supports true or false values.
  - **Integer** - stores only whole numbers.
  - **Credential** - contains usernames and passwords that the Robot requires to execute particular processes, such as login details.

- Assets can have a global value or a value per user

- For a value per user ,only the specified user will access a certain value stored in that asset

Folder entities - Storage Buckets

Storage buckets are entities used for storing files which can be used in automation projects

UiPath Academy

Chapter 14:

Working with Orchestrator Resources

Working with Storage Buckets in Orchestrator and Studio

Folder entities - Queues

- Queues are containers that can hold an unlimited number of items, storing different types of data

- The process of feeding items to a queue is typically different from the process of processing the queue items, and is handled by different robots

UiPath Academy

Chapter 14:

Working with Orchestrator Resources

Configuring Populating and Consuming Queues

Folder Entities - Triggers

Triggers enable the execution of jobs in a preplanned manner:

- **Time triggers**—they instruct the automation to start at regular intervals
- **Queue triggers**—they instruct the automation to be activated whenever new items are added to your queues
- **Event triggers**—they instruct the automation to start whenever a specified event occurs

Connected Triggers

Orchestrator provides you with the means to configure the queue and time triggers in your processes via the Package Requirements tab. This allows you, for example, to create the missing queues and time triggers.

Orchestrator accommodates trigger-based automations from Studio via several personal workspace adjustments such as:

- When you publish a project to Orchestrator, the package becomes available in your personal workspace. Then, Orchestrator automatically creates a process in the workspace so you can start executing it right away
- When the automation project is republished to Orchestrator, the queue trigger properties are overwritten
- At publish time, Orchestrator chooses from the available personal workspace's runtimes (Serverless, Production, and Nonproduction) to execute the job

This functionality reduces the amount of required coordination between the Automation Developers and the Orchestrator Admin.

It reduces the chance of misconfigurations and makes maintenance easier.

Folder Entities - Credential Stores

A credential store is a named location within a secure store, such as CyberArk, in which you can store sensitive data, such as Robot credentials and credential assets.

Orchestrator supports the use of multiple credential stores at the tenant level and provides built-in support for

- Azure Key Vault
- CyberArk CCP
- HashiCorp Vault
- Thycotic Secret Server
- BeyondTrust
- CyberArk
- And the architecture enabling you to develop a plugin for other secure stores

Personal Workspaces

A personal workspace is a modern folder available for the dedicated use of a particular attended user.

Personal Workspaces make it easy to deploy automations to your own robot, for easy regular execution, with the organizational benefits of logging, visibility, and potential reuse.

Personal Workspaces come with the following entities:

- package feed
- machine template
- resources (jobs, assets, logs, queues, etc.)

Personal Workspaces

A Personal Workspace is a special kind of folder, hosted in Orchestrator and accessible only to its owner.

As with any other type of Orchestrator folder, a Personal Workspace acts as a storage area for your projects, allowing you to control your automations and their intrinsic entities, such as processes, queues, assets, and storage buckets

The essential difference between a regular Orchestrator folder and a Personal Workspace has to do with permissions:

- Regular folders ensure collaboration across an organization. To this end, they are set up by Orchestrator admins with fine-grained control in mind and are shared between multiple users

- Personal Workspaces, on the other hand, are designed to serve as an automation playing ground for its owner. Once it's set up, you'll be the only person with access to it and you'll be able to use it at any time

UiPath Orchestrator

Working with Personal Workspaces in StudioX and Orchestrator

Roles

Roles are sets of permissions used to control the access of human users and robots to tenant and folder entities

Each permission is defined from the combination of at least an action type (view, edit, create, and delete) and an entity, be it at the tenant or folder level.

For example, you can set up the right to view only the queues in a certain folder, but not the queues from other folders

Logs in Orchestrator

Logs are time-stamped files that contain informational events, errors, and warning messages relevant to the application.

The UiPath Platform has logging capabilities for all of its main components.

Logs are created locally of every robot and automation action, then sent to Orchestrator from where they can be filtered, viewed, and analyzed.

There are two types of Orchestrator logs:

1. **Diagnostic logs** generated by UiPath Orchestrator regarding its behavior
2. **Execution logs** are logs generated by process execution. The Logs page displays logs generated by Robots in all folders you have access to, including logs generated for jobs started through remote debugging sessions. To access it, navigate to the Automations tab from a folder context, and select Logs tab

UiPath Academy

Chapter 13:

Orchestrator Overview for Automation Developers

Orchestrator Entities, Tenants and Folders

Resources

Topic	Link
Organization Modeling in Orchestrator	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/organization-modeling-in-orchestrator
About the Tenant Context	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/about-the-tenant-context
About the Folders Context	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/about-the-folders-context
Automation Best Practices	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/automation-best-practices
About Roles	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/about-managing-user-access
Orchestrator Logs	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/orchestrator-logs

Provision an Unattended Robot to Orchestrator

Robot provisioning refers to the process of setting up and configuring robots within Orchestrator.

You can connect your attended and unattended robots to Orchestrator in different ways.

On a high level it involves

- adding robots to Orchestrator
- establishing the necessary connections
- configuring their properties to enable effective management and execution of automation processes

User Robots & Accounts

In Orchestrator, both human users with attended licenses (Robot or Studio) and unattended robots need to have a corresponding Orchestrator user.

Depending on the deployment type and the organizational setup, users are added and managed in different ways:

- Users can be added locally in on-premises, standalone Orchestrator.
- Users have to be added in Automation Cloud, then in cloud Orchestrator to grant them a license.
- Users can be added from Active Directory for both on-premises and cloud Orchestrator if the integration was configured beforehand.

Robot accounts have to be created in Automation Cloud, and they behave like user accounts regarding permissions.

The only differences compared to user accounts are:

- Robot accounts aren't allowed to have any interactive-related process configuration.
- No email address is required to create a robot account.

Automatic Robot Creation

To simplify the attended and unattended robot creation, as well as the license provisioning, the automation robot creation can be enabled at user level, for both attended and unattended robots, and at group level for attended robots.

Basically, you enable the Attended Robot or Unattended Robot toggle at the account or group level, configure the various settings (robot execution settings, machine login credentials, if applicable), and a floating robot with those attributes is created.

Note: You can only enable attended robot auto-provisioning for user groups.

Unattended robot auto-provisioning is not possible.

Machine Templates

Robots run on physical or virtual workstations.

These are mirrored in Orchestrator by entities called machines.

The machines in Orchestrator work as API key generators, authorizing the connection between the robots and Orchestrator.

There are two types of machines in Orchestrator:

- **Machine templates**: this allows the connection to multiple workstations with a single API key.
- **Standard machines**: this allows the connection between Orchestrator and a single machine. This is suitable for scenarios in which robots need to run on specific machines.

License Distribution

In Orchestrator, licenses are **also called runtimes**.

They're allocated at machine template level, **under Machines in the Tenant menu**.

The number of runtimes allocated there should be matched with the maximum number of users which can run on a single machine connected using that machine template.

On a regular Windows machine, only one user can run.

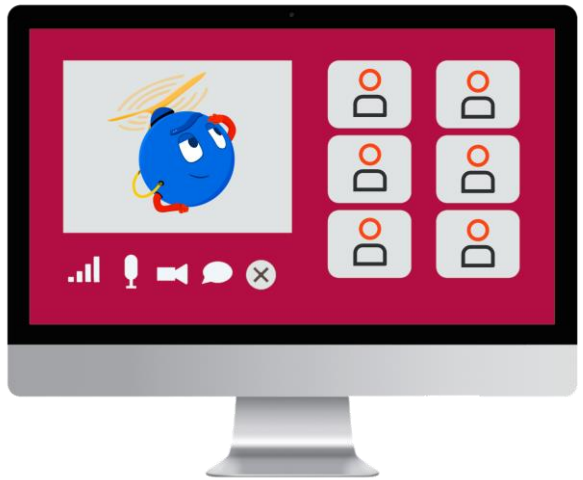
But on a Windows server machine, multiple users can run simultaneously.

Licenses are consumed as soon as a machine is connected to Orchestrator, no matter the number of users running on it.

Resources

Connecting Robots to Orchestrator	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/connecting-robots-to-orchestrator
Managing Robots	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/managing-robots-modern-folders#unattended-setup
Service licensing	https://docs.uipath.com/overview/other/latest/overview/service-licensing

Classroom Exercise



Use-case/ problem statement

It’s time to take your environment further by provisioning an unattended robot and connecting it to the same Orchestrator using the Client ID in the Assistant. You can set it up on your current machine, but bear in mind that in a real scenario you'll need a different machine (virtual or physical).

In the previous lessons, you saw how you can connect an unattended robot to Orchestrator using the Machine Key in the Assistant, revert to it if needed.

A few important things to look after

- Accounts and Groups.(opens in a new tab)
- Managing Machines(opens in a new tab).
- Connecting Robots to Orchestrator(opens in a new tab).

UiPath Academy

Chapter 13:

Orchestrator Overview for Automation Developers

Provisioning an Unattended Robot to Orchestrator

Unattended Automation with Folders

Functionally, the purpose of attended automation is to have the robots ready to take over the undesirable tasks when the human users need it, in their cycle of work and during work hours.

When it comes to unattended automation, the purpose is quite different: the robot needs to be busy as much as possible, with as little human input as possible.

Use of Folders & Job Priorities

By accurately reflecting the business hierarchies with the help of folders, roles and permissions, we can control the access to automation processes and make sure the effort is spent where it brings the most return.

Job priorities can ensure that business priorities are well reflected in the automation process.

You can control which job has precedence over other competing jobs through the Job Priority field, either

- when deploying the process or
- when configuring a job/trigger for that process

The separation of processes based on Ui Interaction

Processes in Orchestrator are now differentiated between

- foreground processes : those that interact with the user interface and
- background or headless processes : those that don't

This has a significant impact on the way automation jobs are executed.

An unattended robot can **simultaneously run a foreground job and as many headless jobs as available runtimes on a machine**.

License allocation per machine

Allocating licenses (runtimes) per machine makes sure that their consumption is optimized.

For example, on a Windows Server machine, multiple robots can open sessions and run unattended jobs up to the maximum number of runtimes.

Custom Job allocation strategy

Unattended automation was designed in Orchestrator to ensure resource optimization and effectiveness.

But there are cases in which business logic is more important.

Orchestrator offers a couple of features and options to allow the customization of the job allocation process so that certain jobs or resources are available only to certain users.

UiPath Academy

Chapter 13:

Orchestrator Overview for Automation Developers

Unattended Automation with Folders

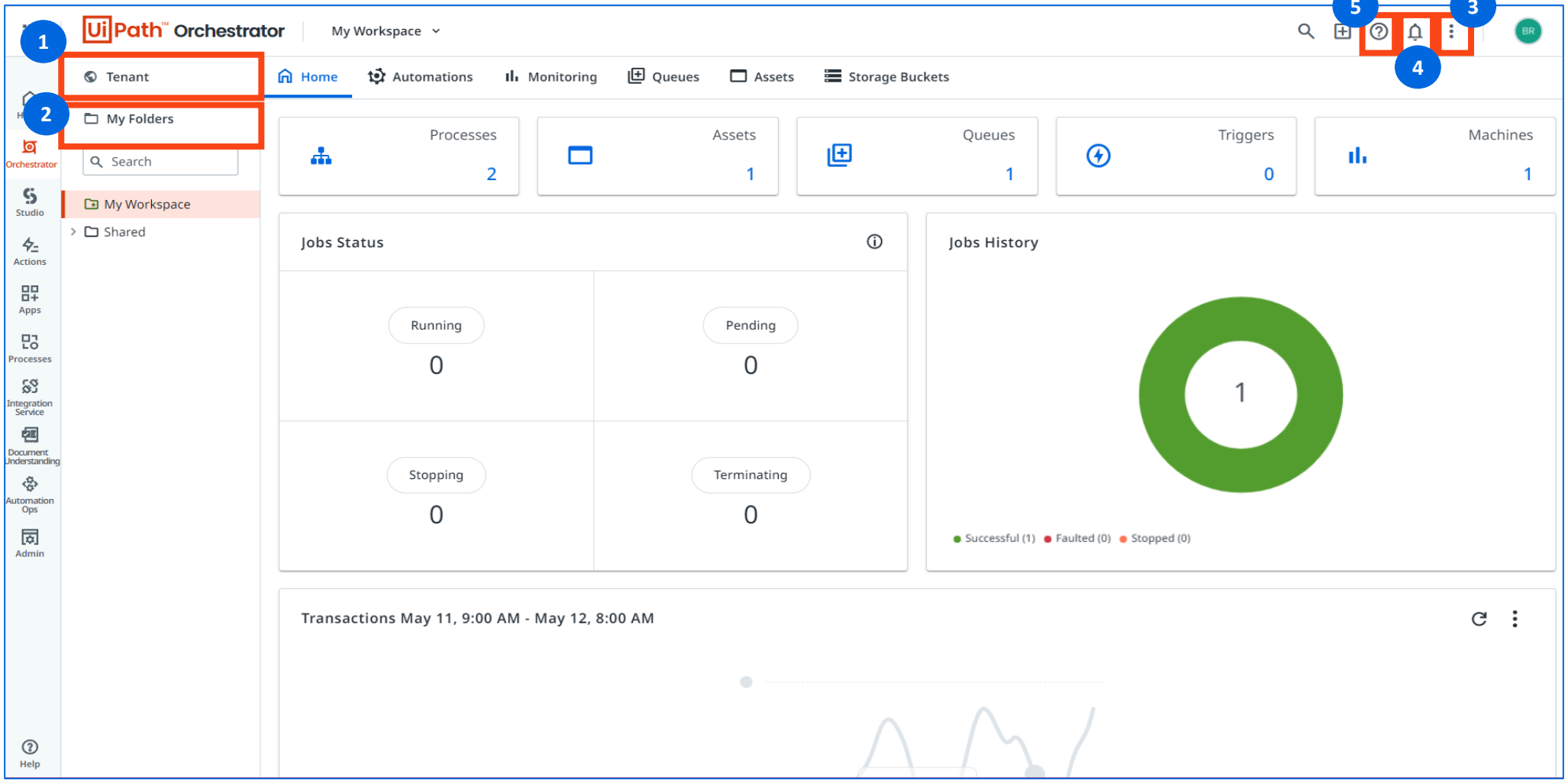
Resources

Useful concepts in unattended automation	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/useful-concepts-in-unattended-automation
Managing Large Deployments	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/managing-large-deployments
Background and **Foreground process automation**	https://docs.uipath.com/robot/standalone/2023.4/user-guide/background-process-automation
Background Vs Foreground Processes	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/about-processes#background-vs-foreground-processes
About Jobs	https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/about-jobs

Orchestrator Functionalities

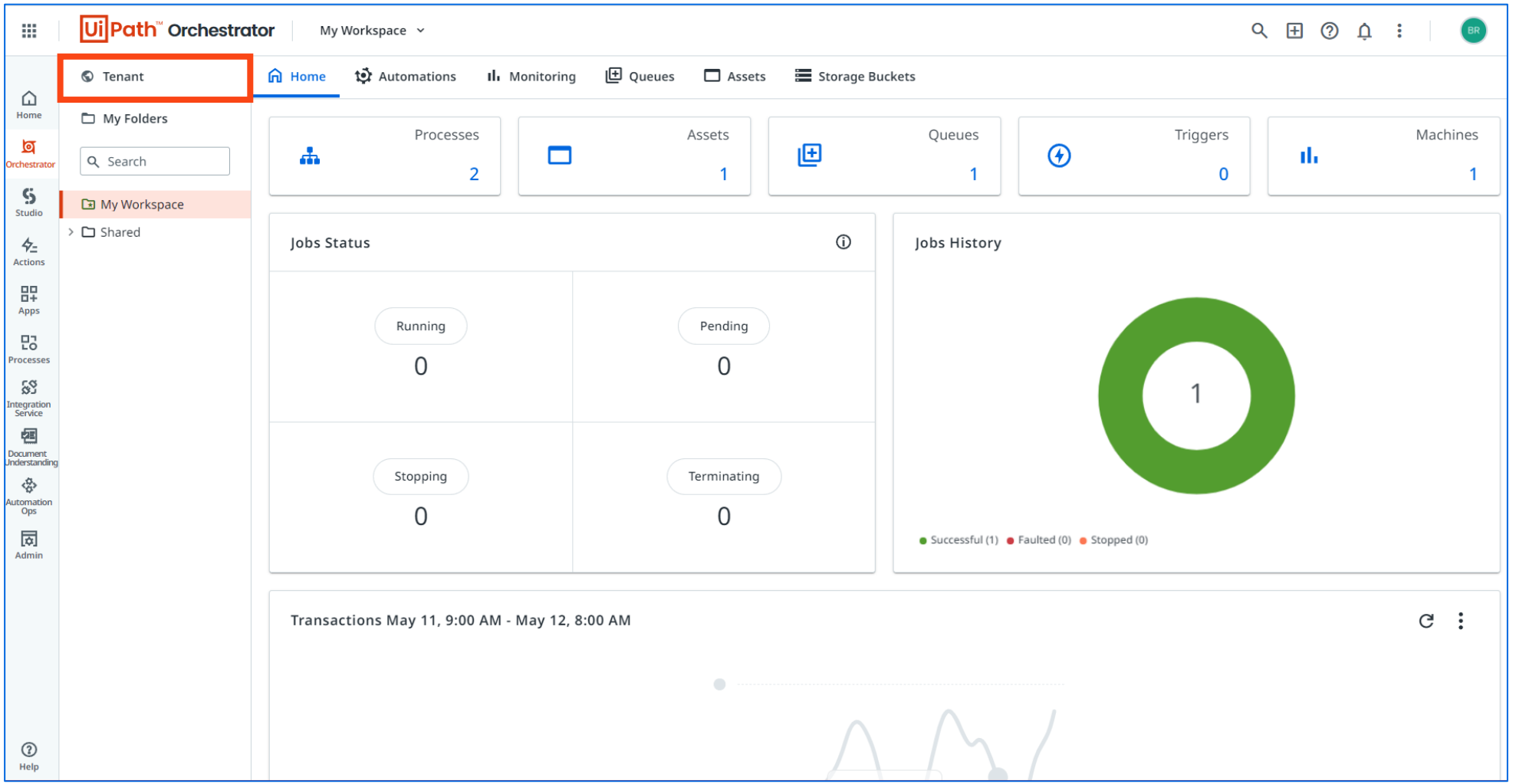
- Orchestrator User Interface
- Contexts of Orchestrator
  - Tenant
  - Folder

Orchestrator User Interface



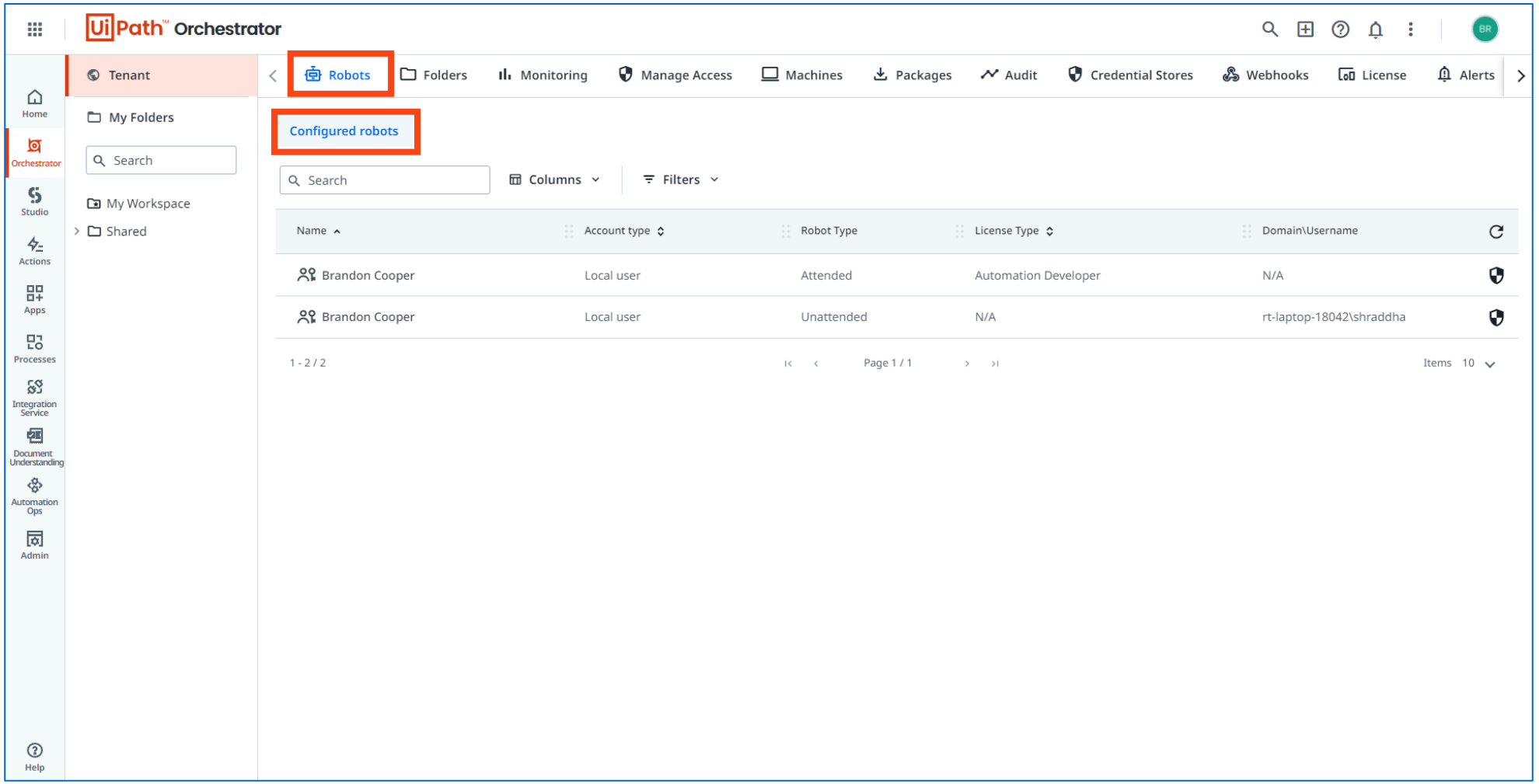
Tenant

The Tenant context is used to manage tenant-level entities of Orchestrator deployment.



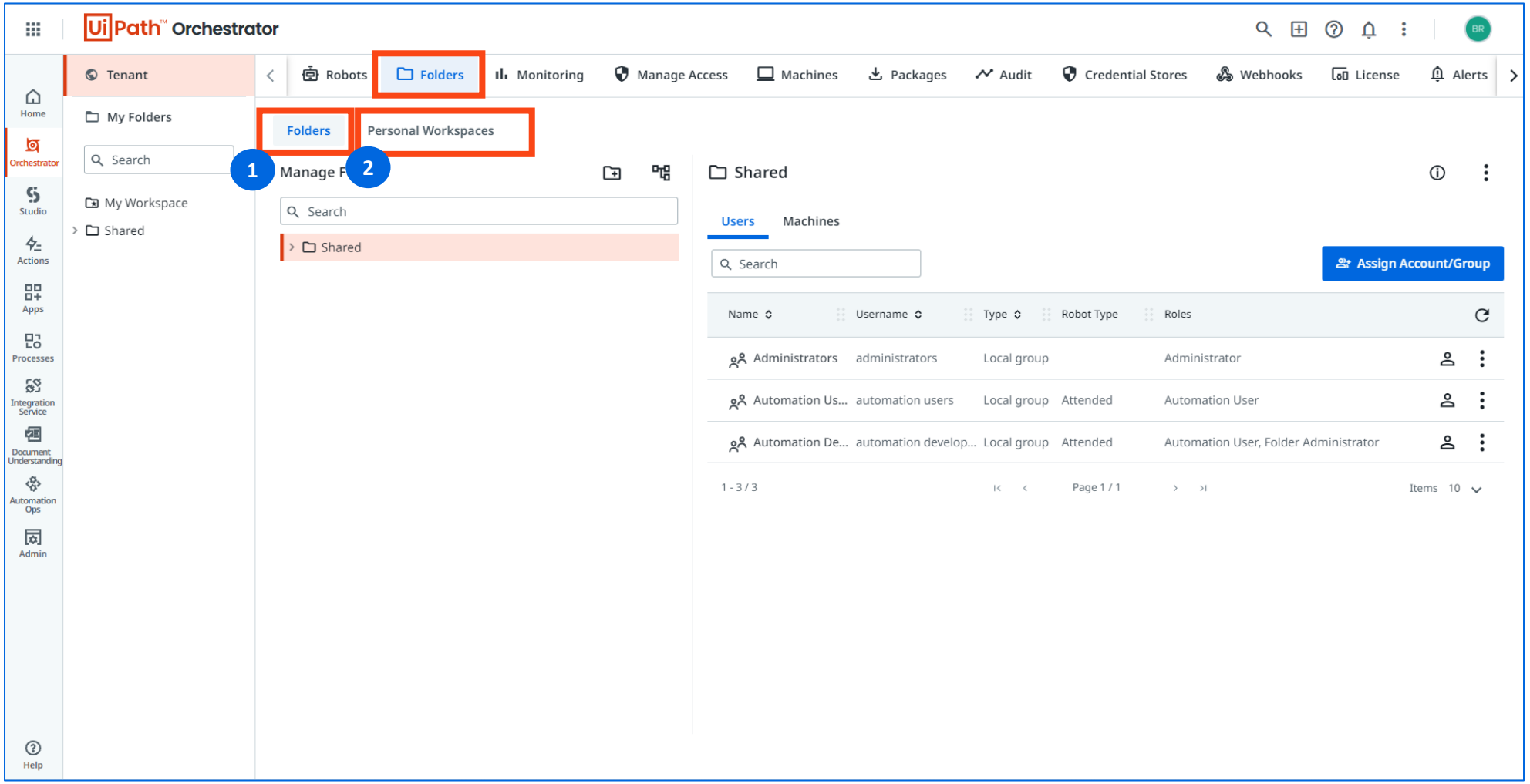
Tenant ⟶ Robots

The Robots page at the tenant level illustrates the robot configuration done in Orchestrator.



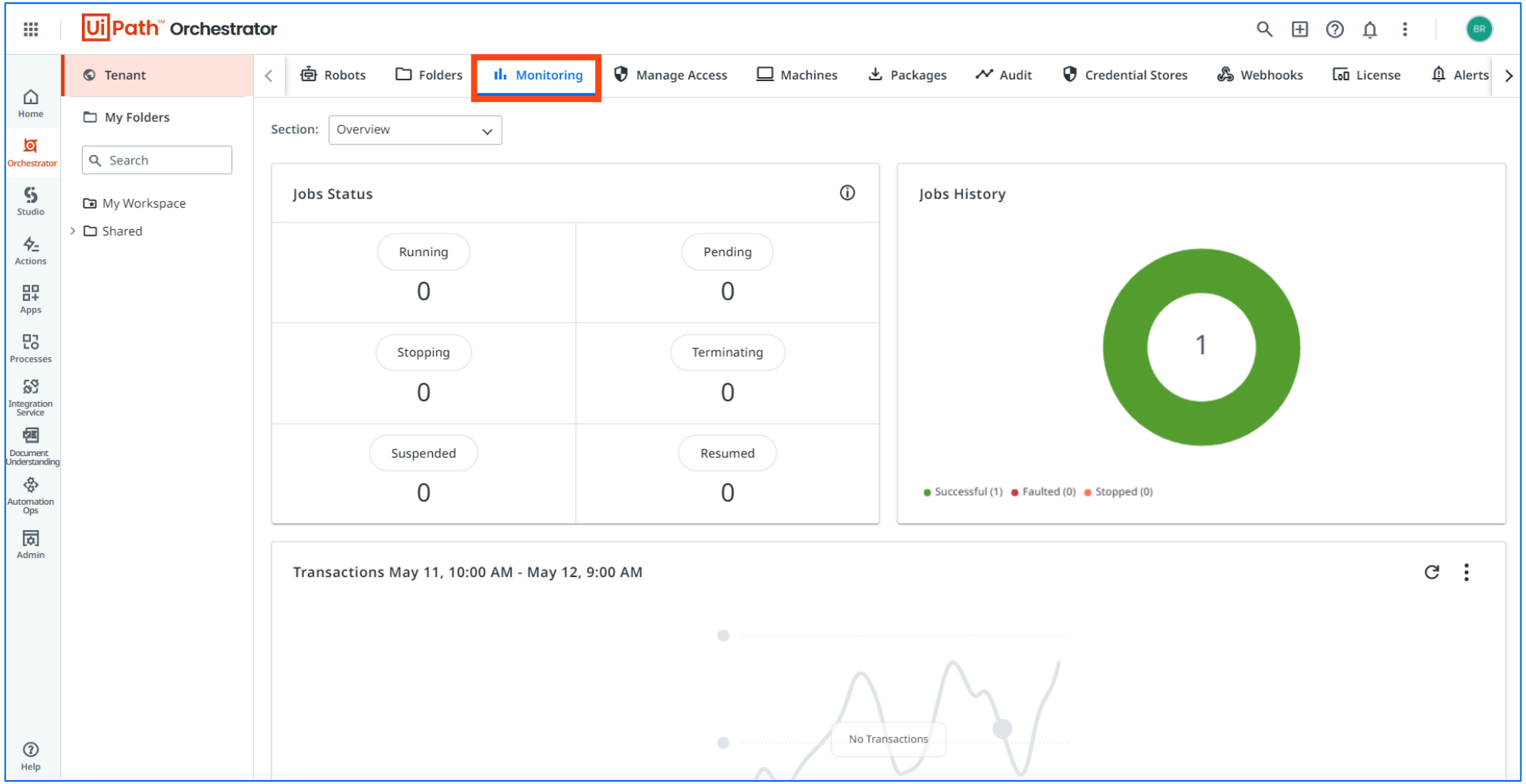
Tenant ⟶ Folders

The Folders page shows the folders present in Orchestrator.



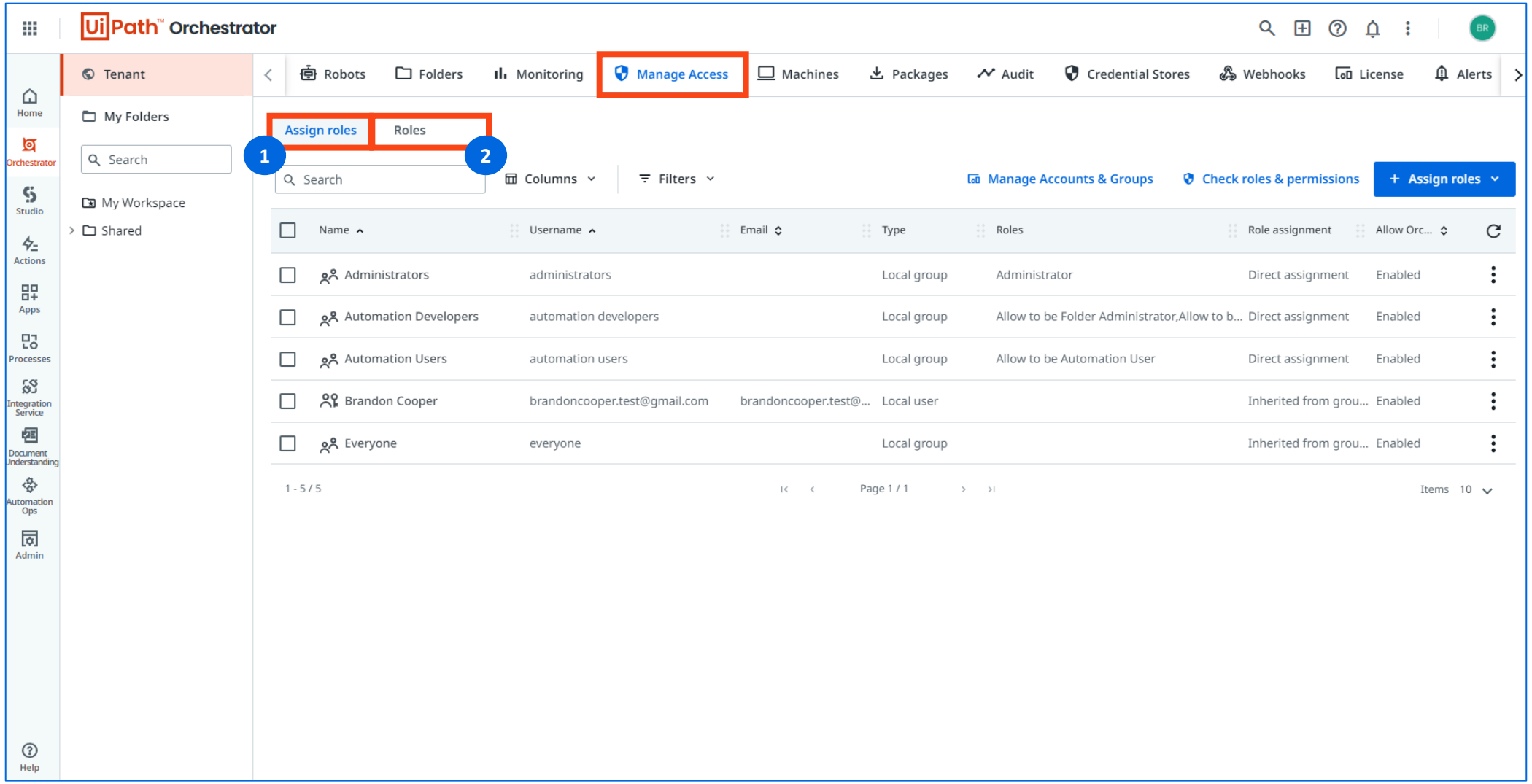
Tenant ⟶ Monitoring

The monitoring page displays aggregated data from all the accessible folders and personal workspaces.



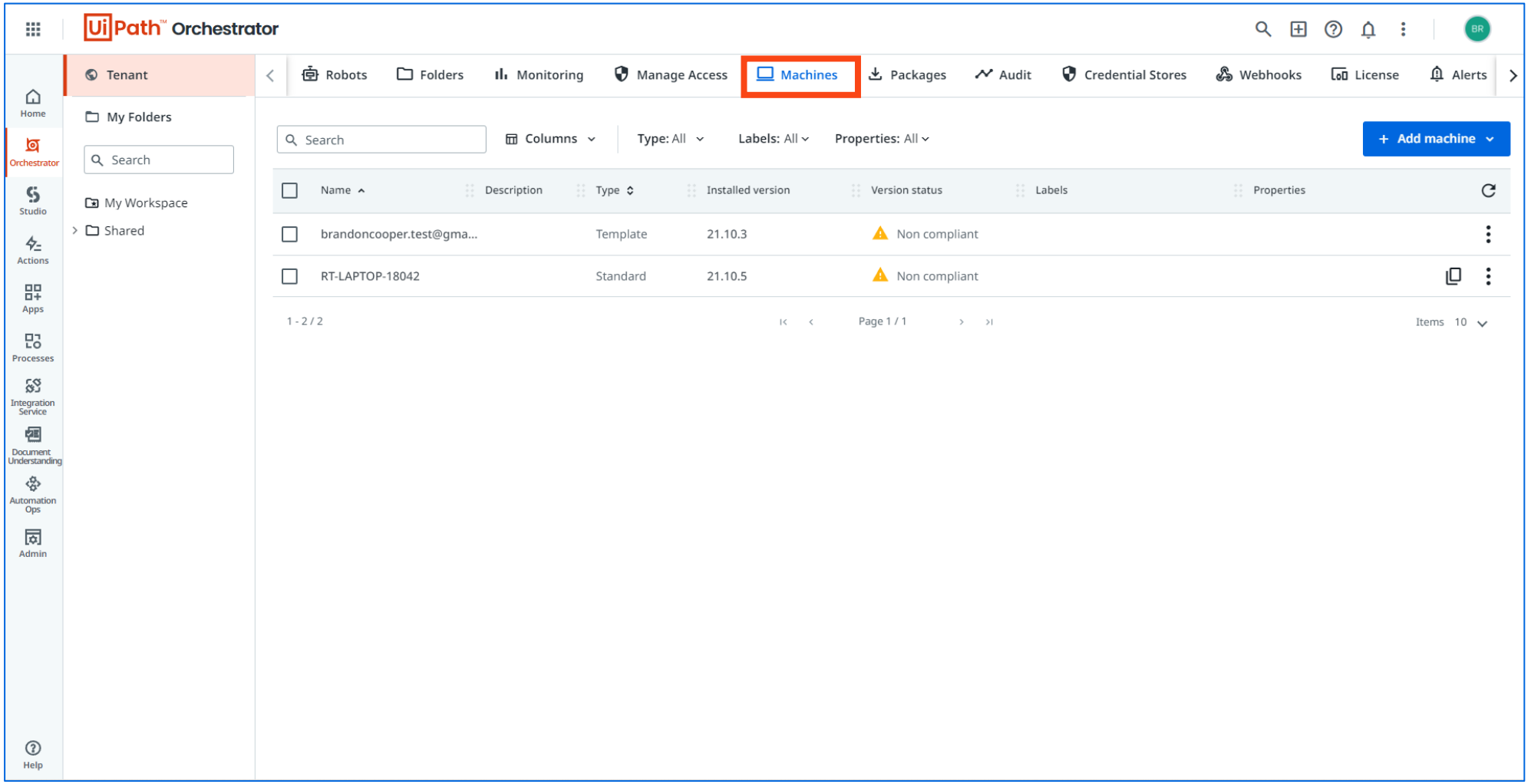
Tenant ➟ Manage Access

Manage Access is the page where user accounts are added and managed by the System Admin.



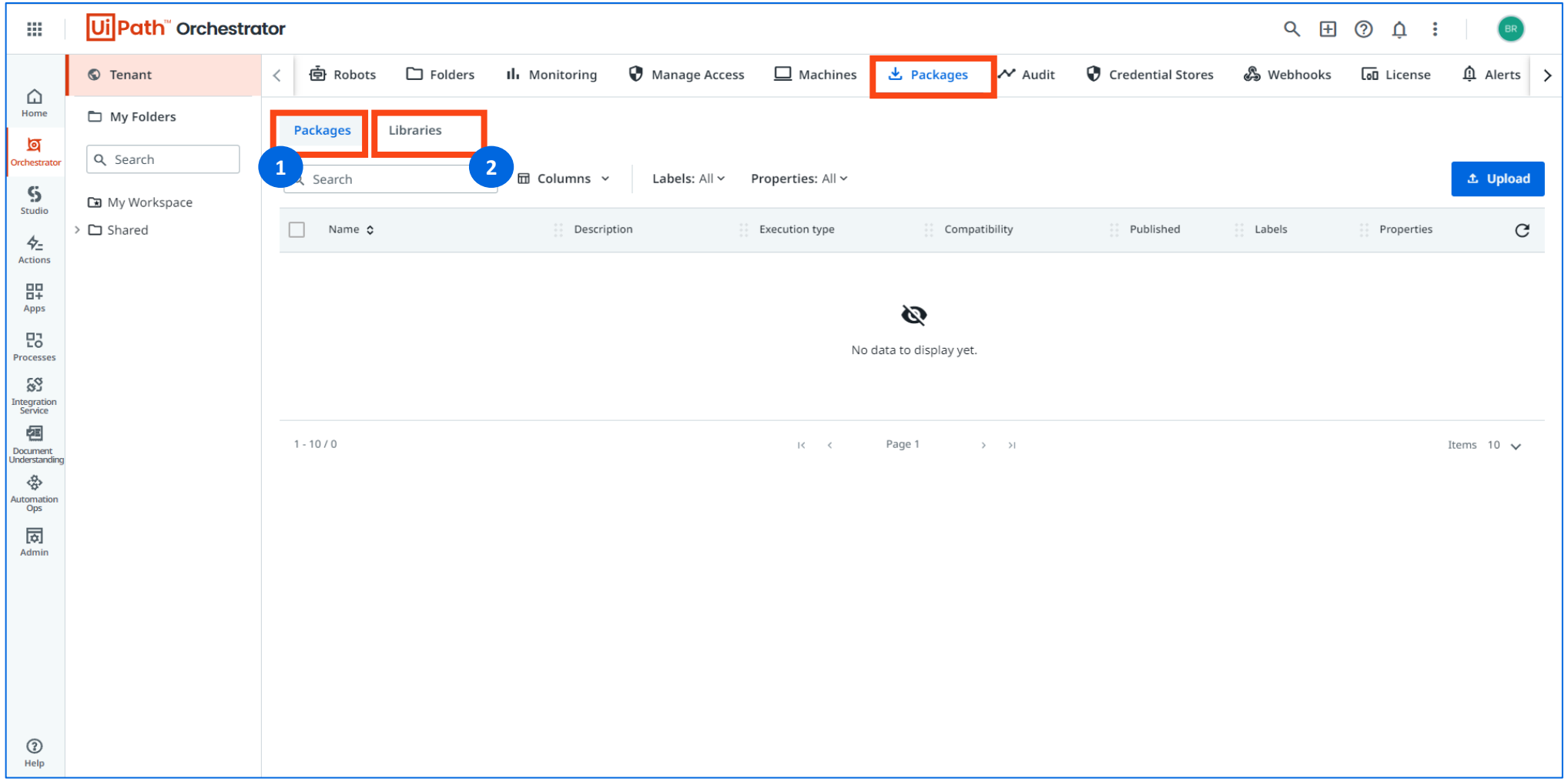
Tenant ⟶ Machines

The Machines page allows you to provision and manage machine entities to further use them for connecting Robots to Orchestrator.



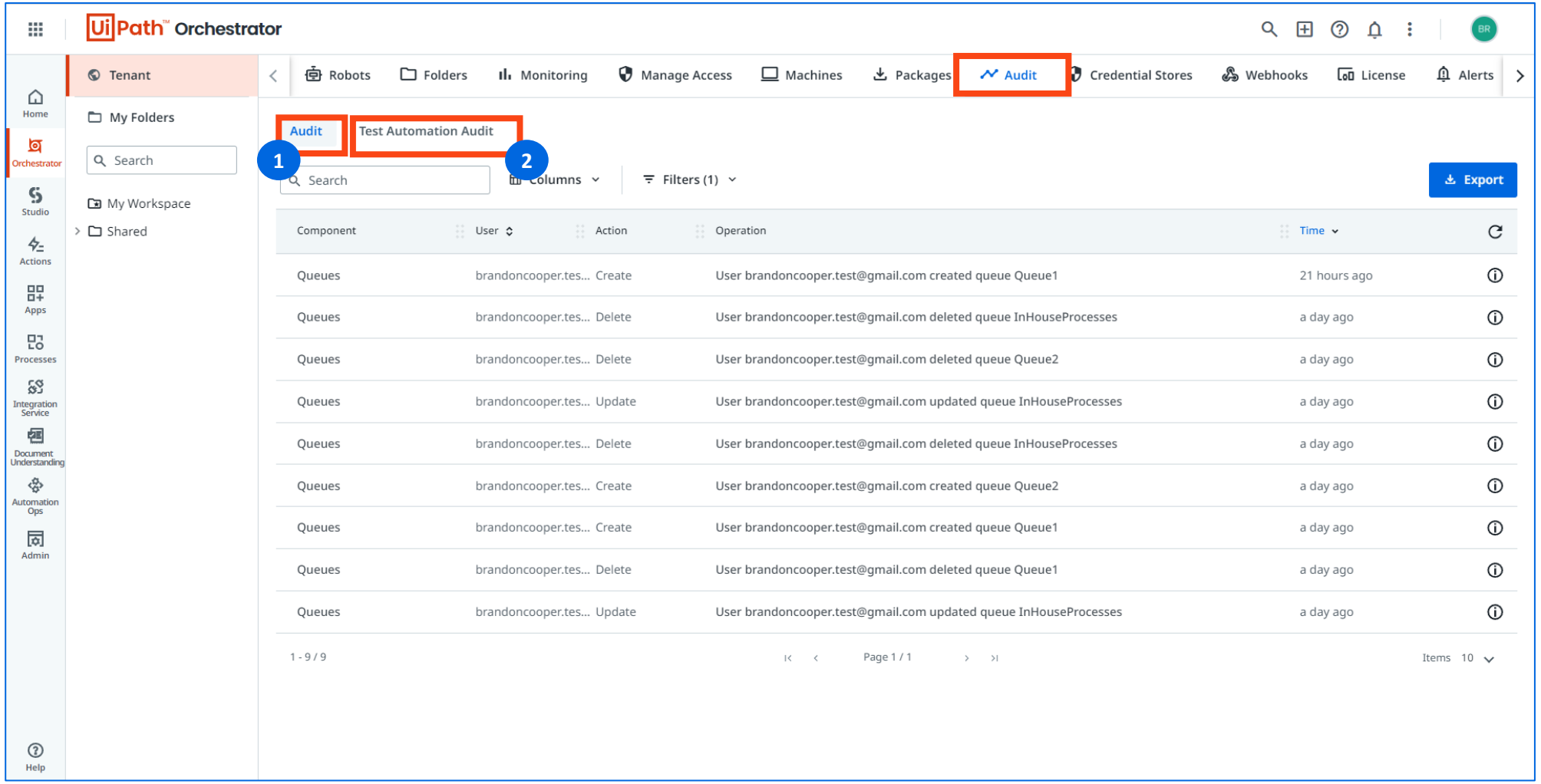
Tenant ⟶ Packages

The Packages page displays all the projects published from Studio as well as the ones that were manually uploaded.



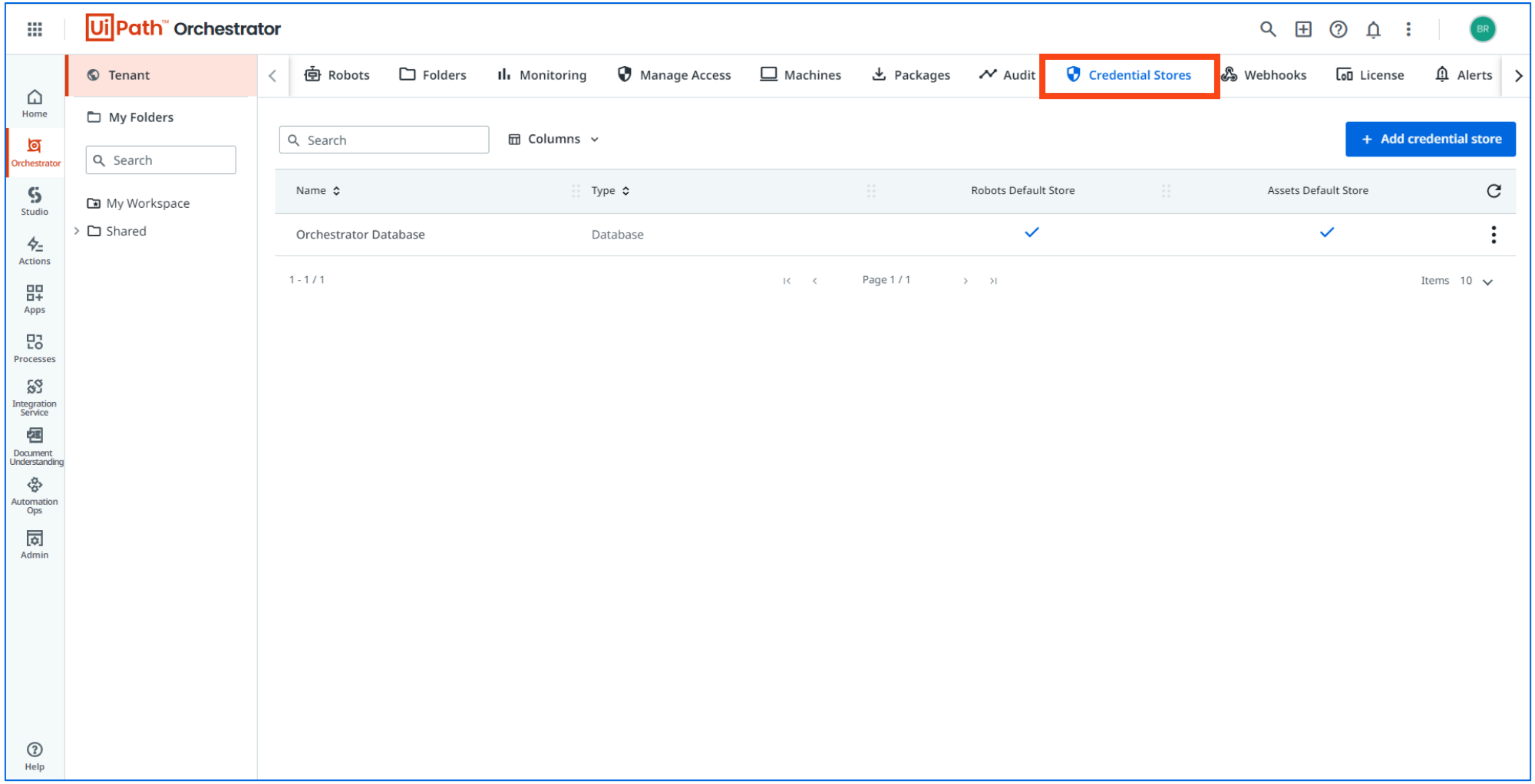
Tenant ⟶ Audit

The Audit page displays the audit trail for actions performed by the Orchestrator users.



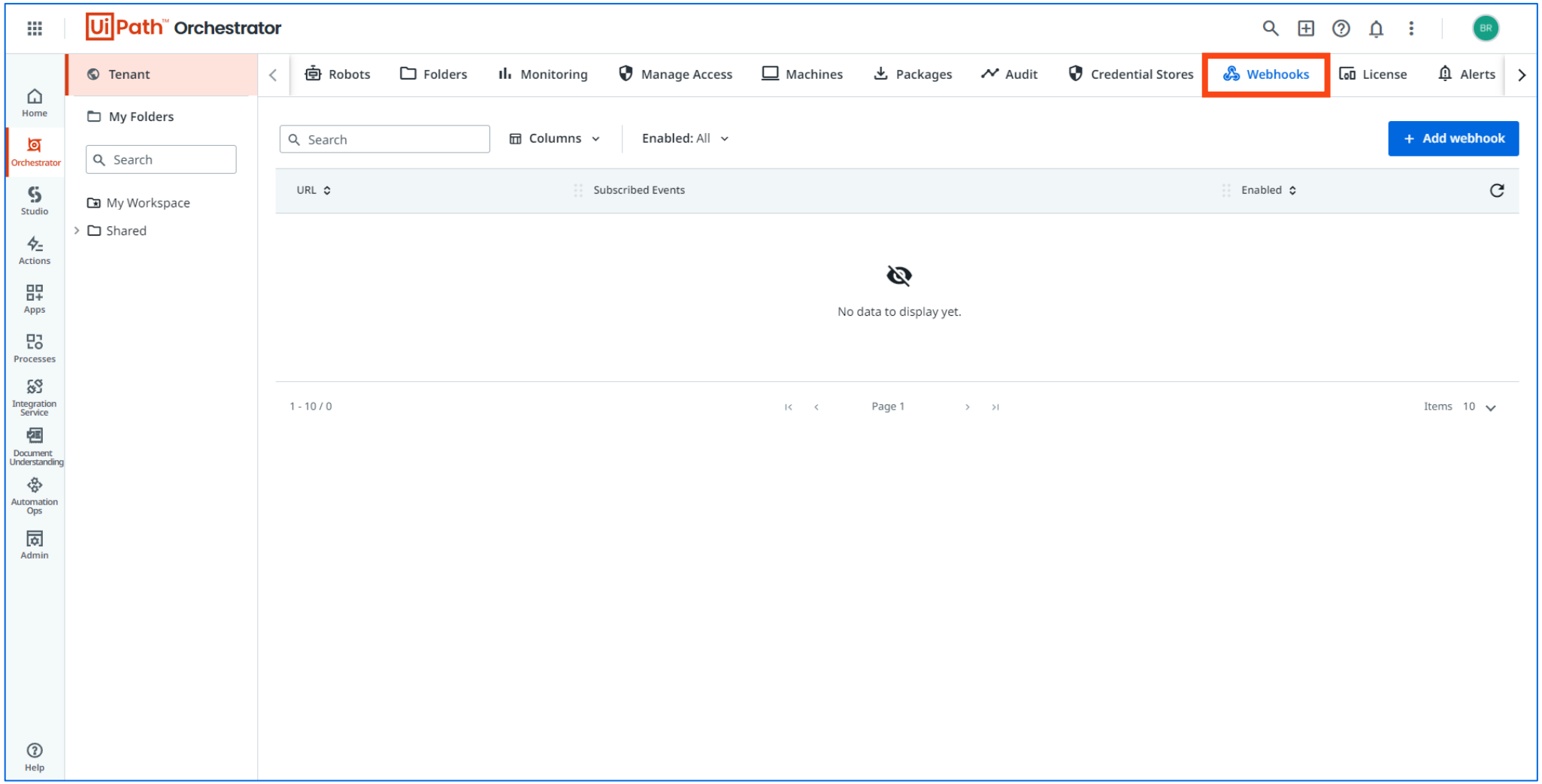
Tenant ⟶ Credential Stores

The Credential Stores page allows you to access and manage per tenant credential stores.



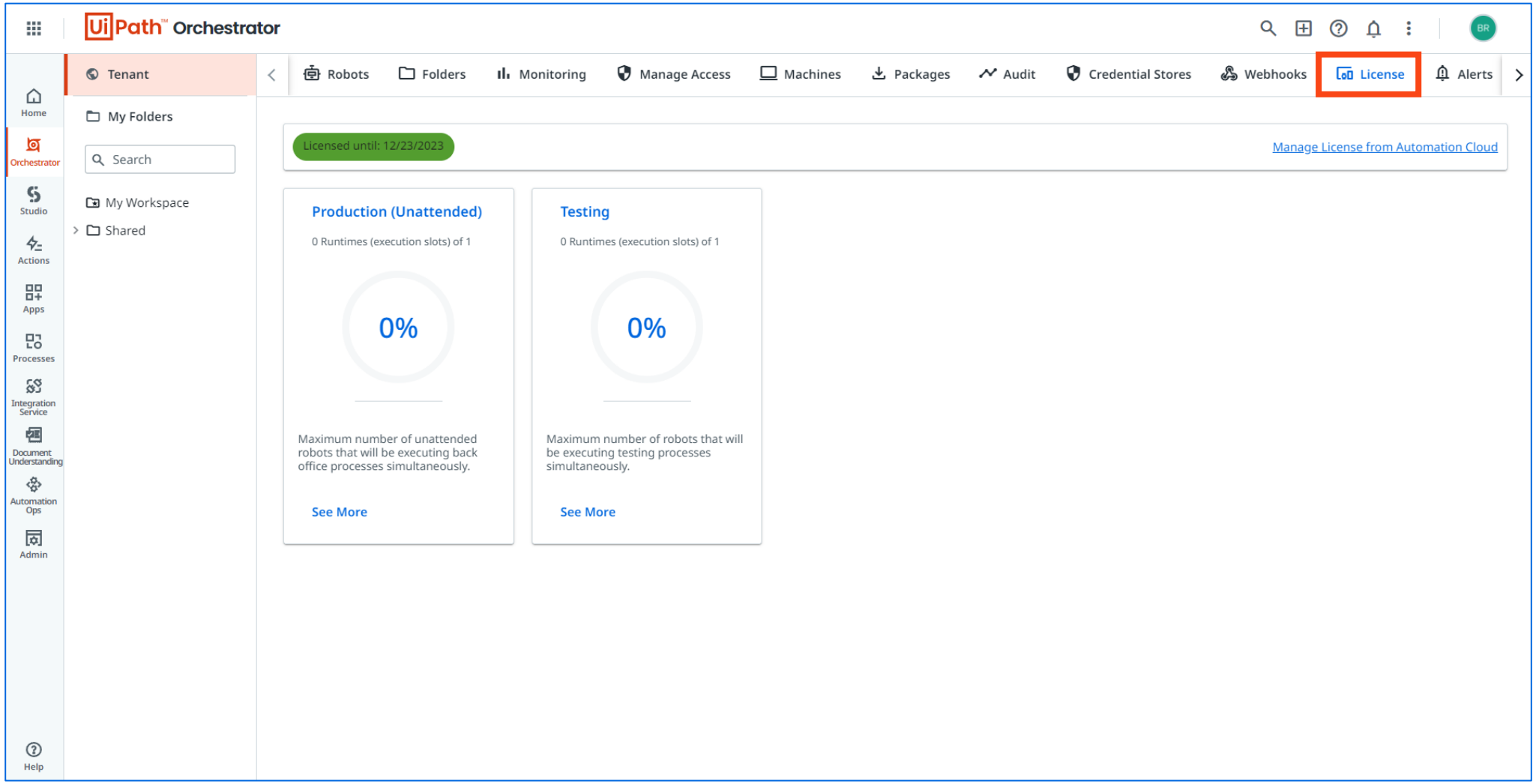
Tenant ⟶ Webhooks

The Webhooks page allows you to integrate the UiPath automation with the entire application ecosystem.



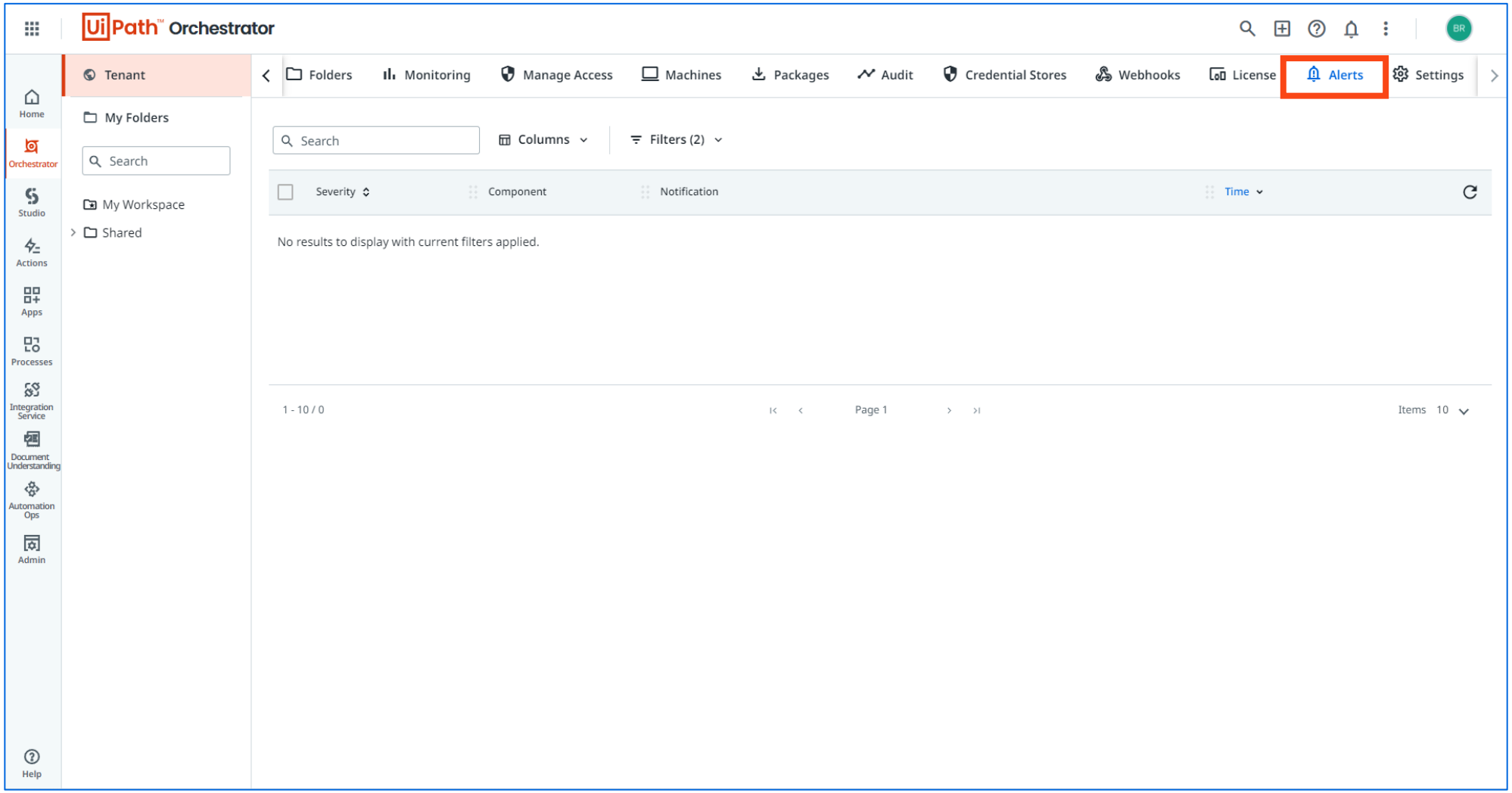
Tenant ⟶ License

The License page allows you to manage the licenses across the tenant.



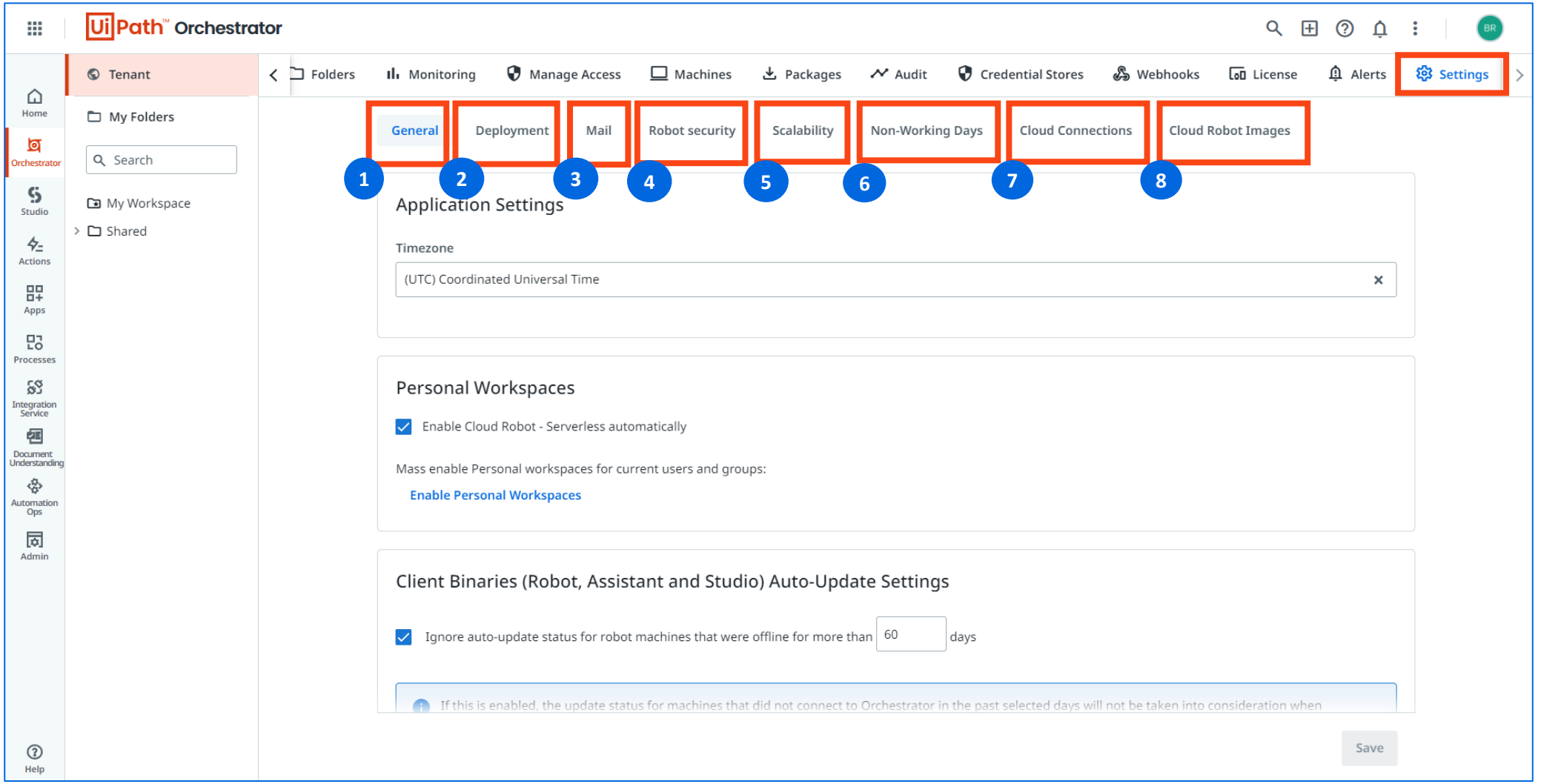
Tenant ⟶ Alerts

The Alerts page displays notifications for robots, queue items, triggers, jobs, processes, and actions.



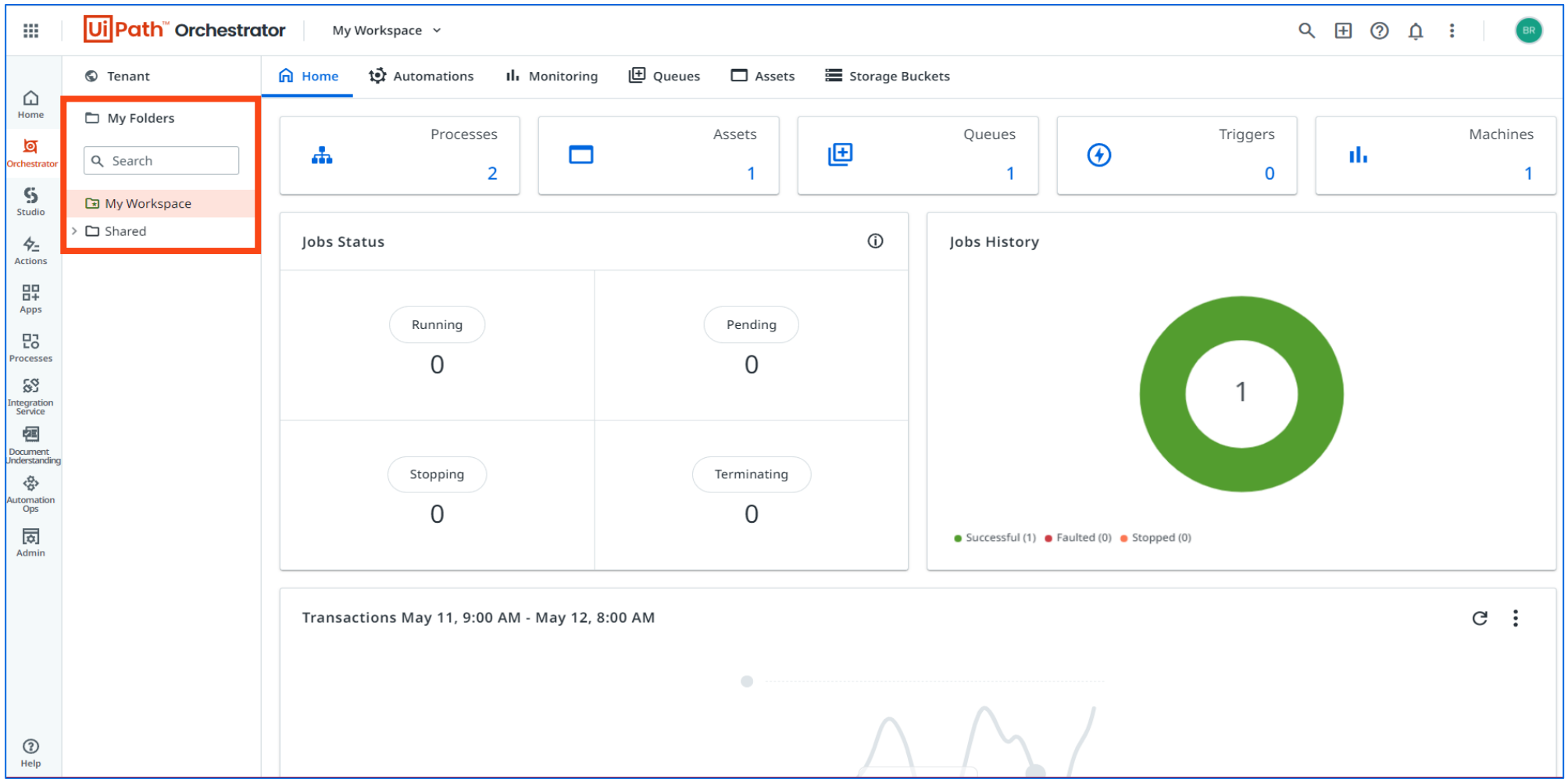
Tenant ⟶ Settings

The Settings page allows you to configure Orchestrator application.



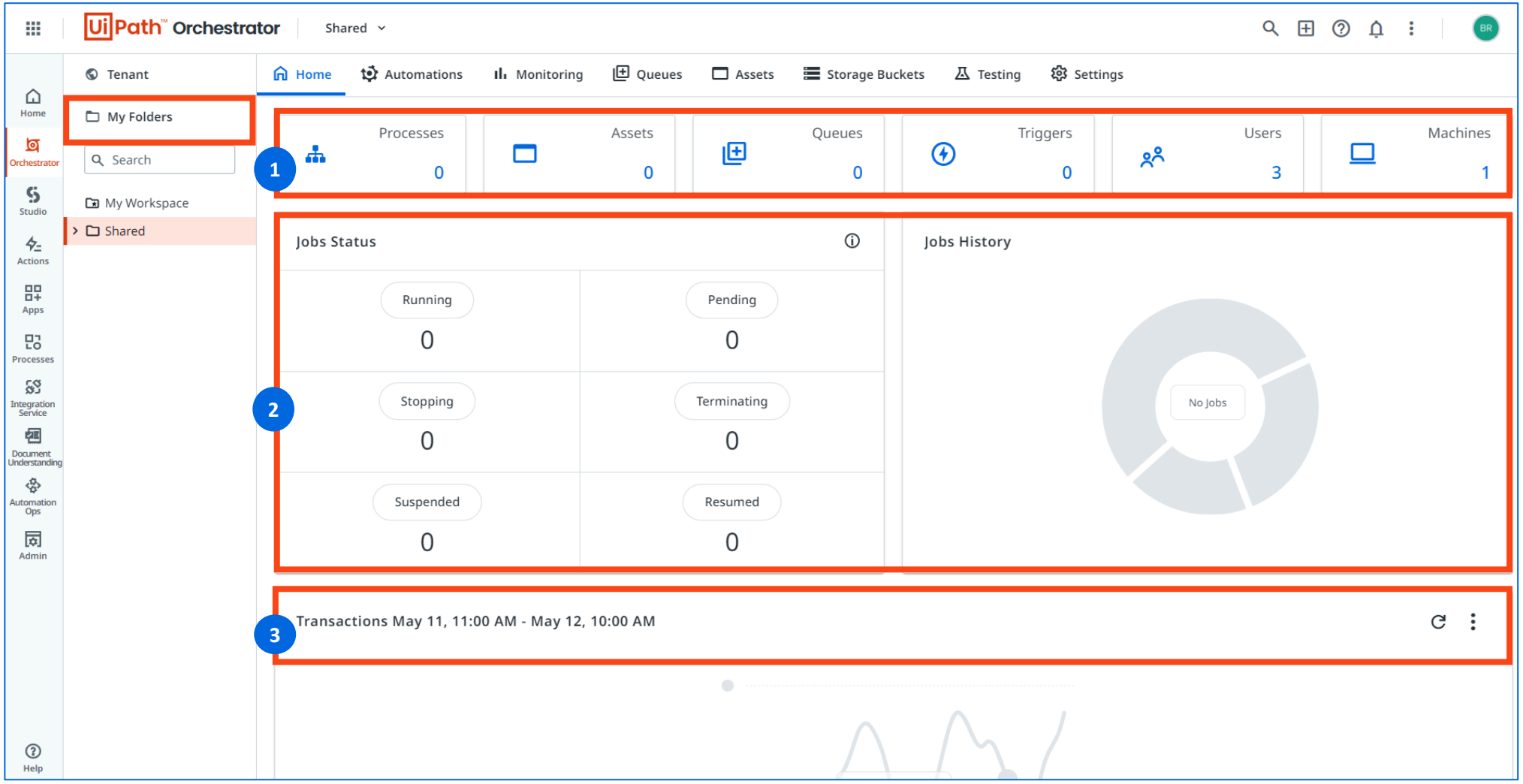
Folders

The Folders context allows you to select any available folder from the sidebar menu to view and manage that folder's entities.



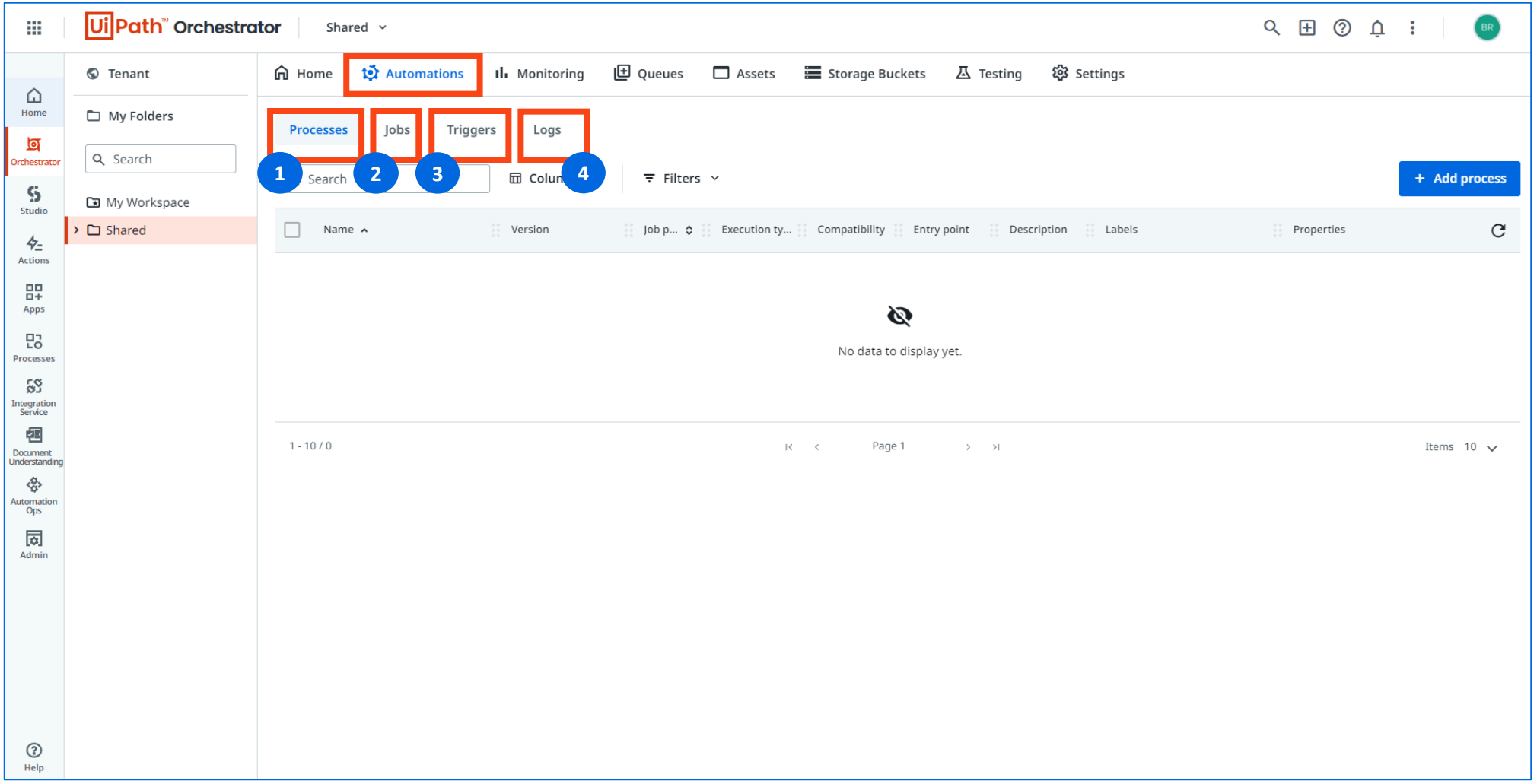
Folders ⟶ Home

The Home page gives a quick overview of Orchestrator entities in the folder.



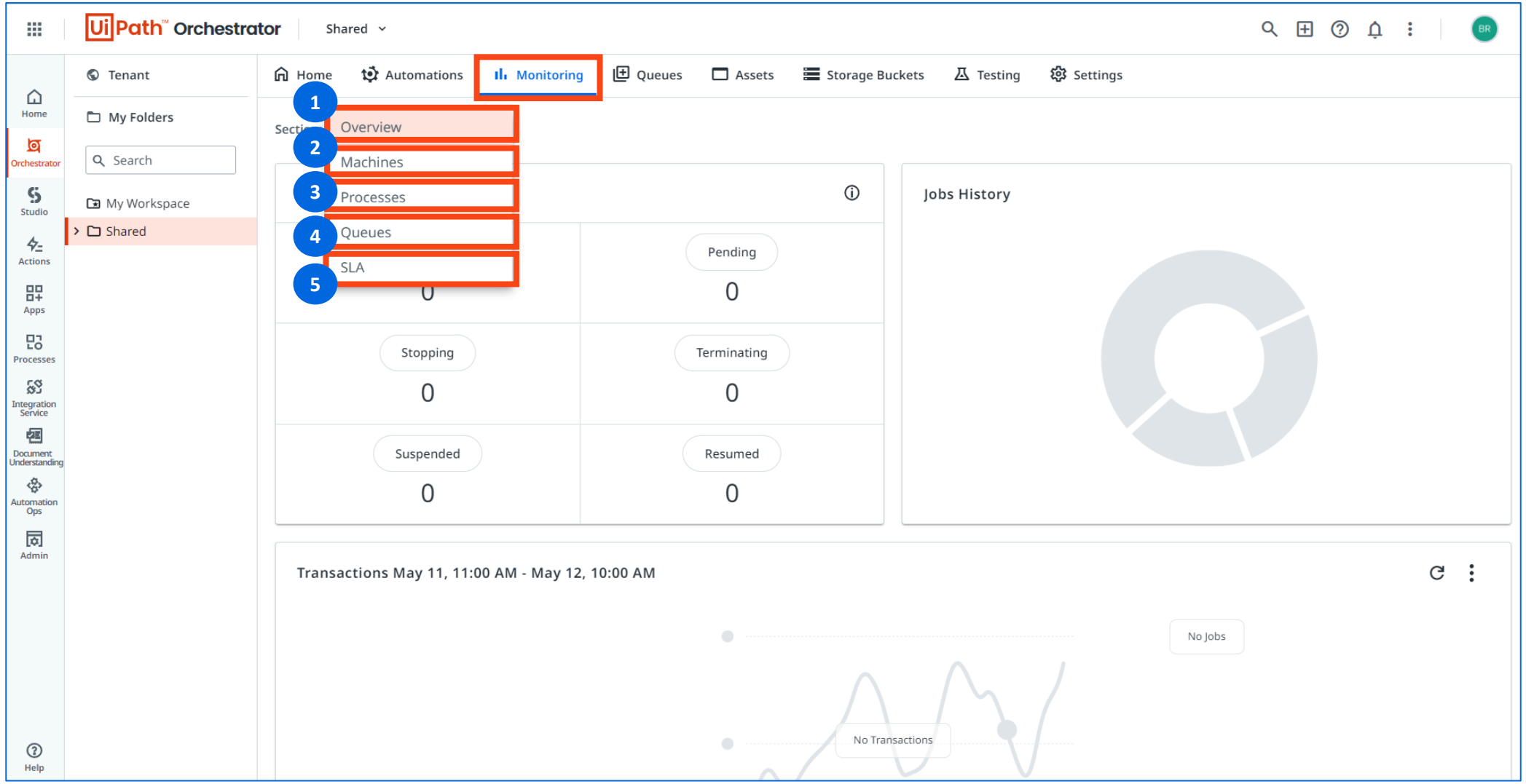
Folders ⟶ Automations

The Automation page allows you to run processes, deploy packages, and manage jobs, triggers, and logs.



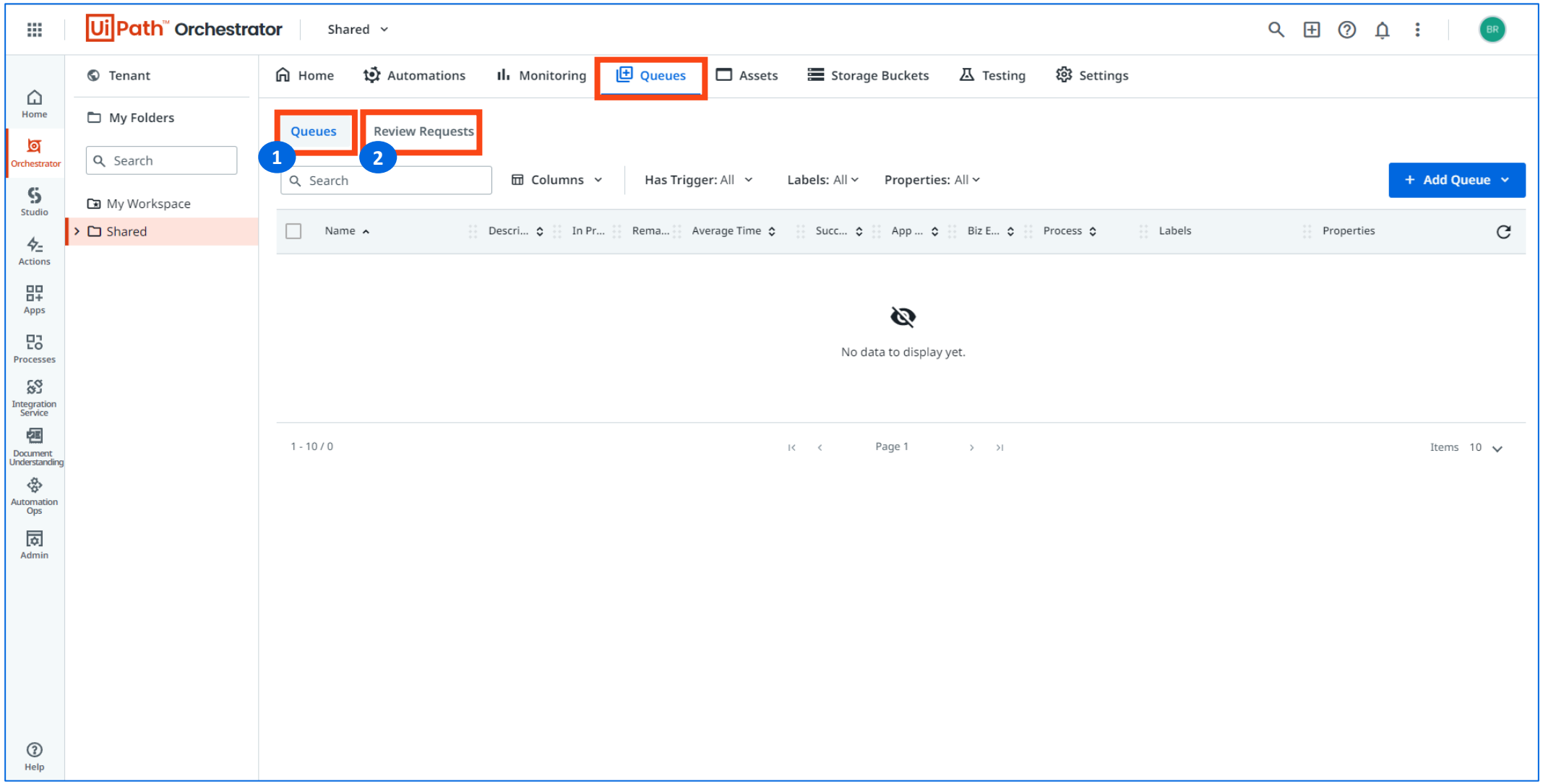
Folders ⟶ Monitoring

The Monitoring page allows real-time monitoring of Orchestrator entities.



Folders ⟶ Queues

The Queues page allows you to create new queues.

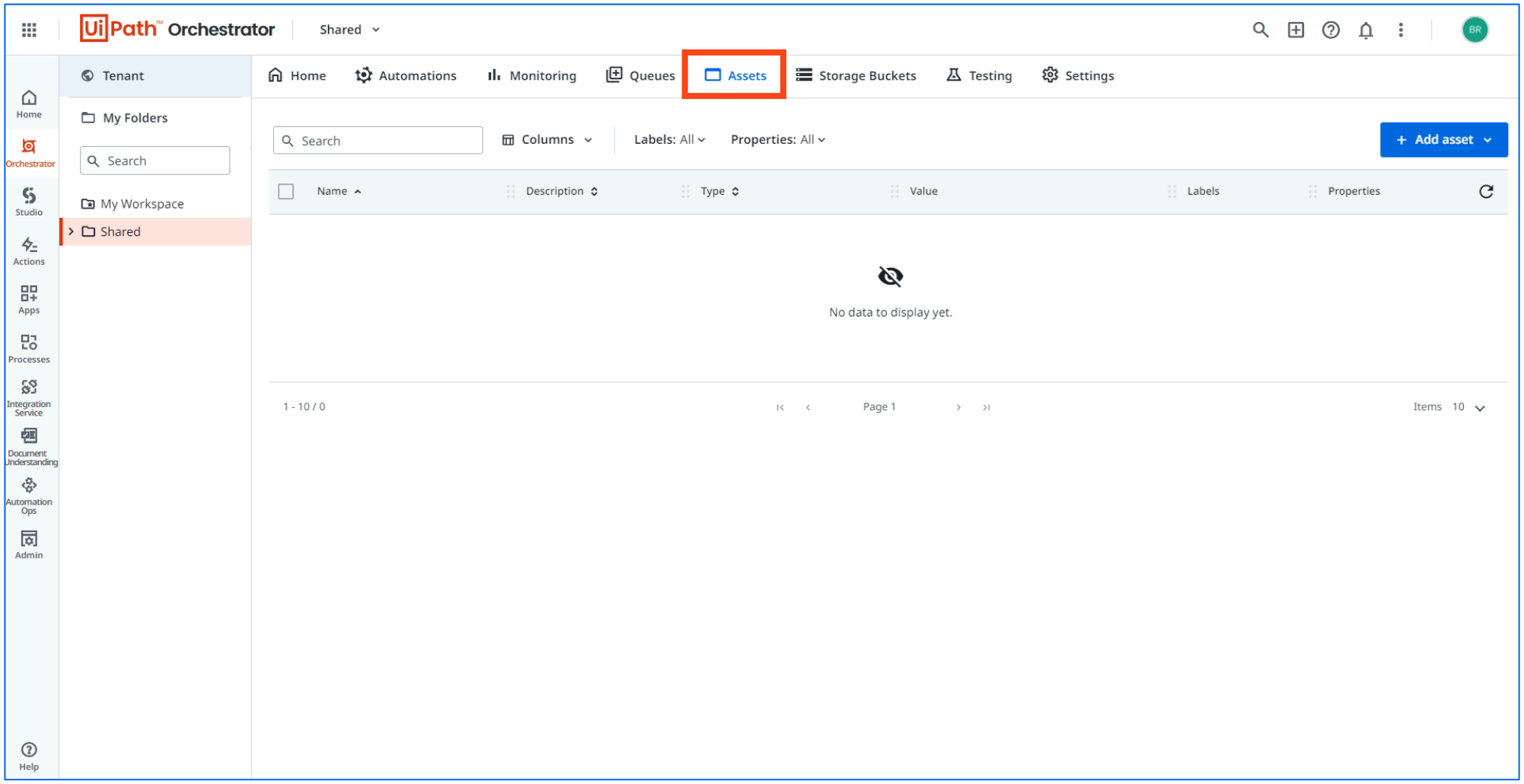


Classroom Exercise

Demonstrate the steps to create a Queue in Orchestrator.

Folders ⟶ Assets

The Assets page allows you to manage shared credentials or variables used in processes.

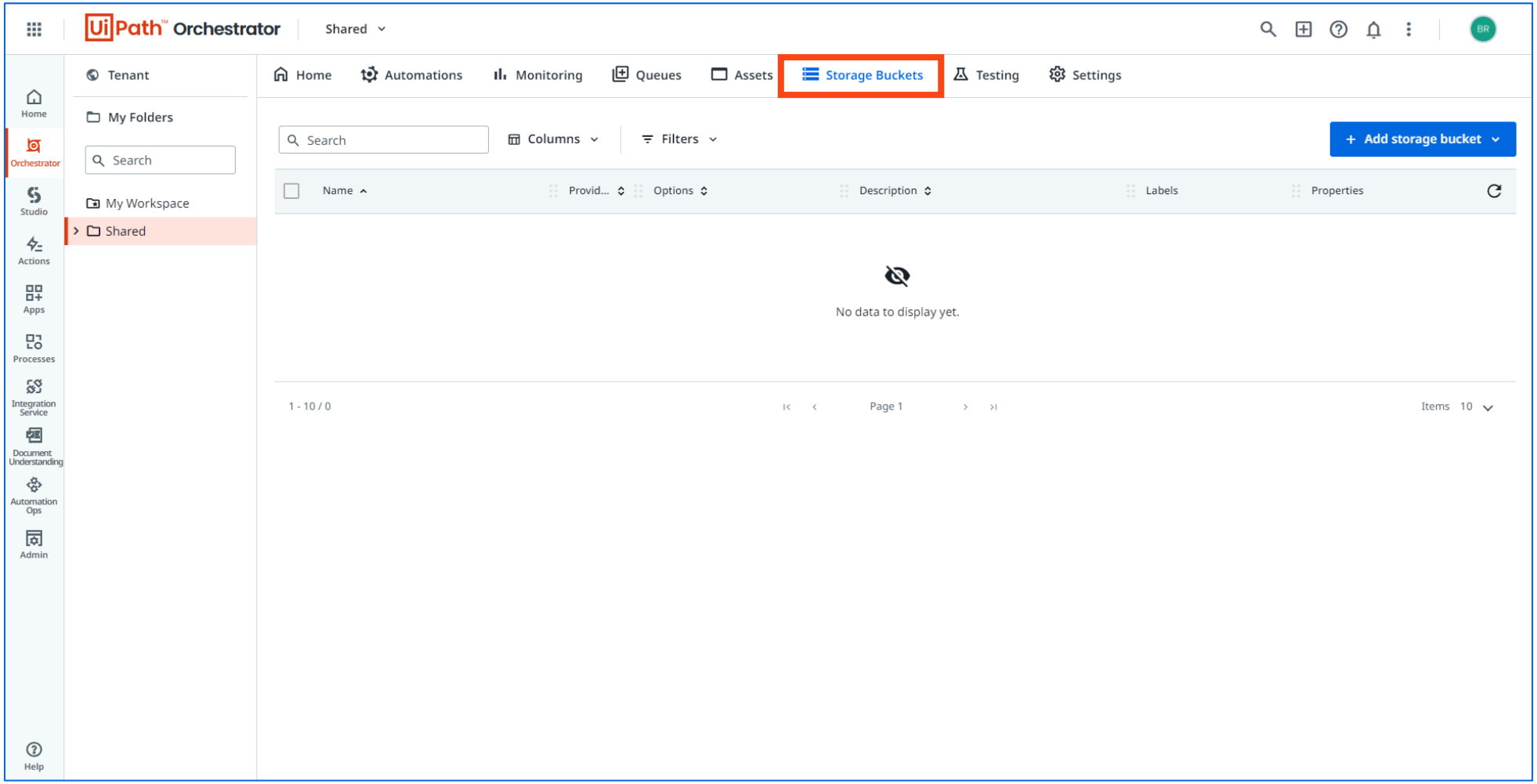


Classroom Exercise

Demonstrate the steps to create an Asset in Orchestrator.

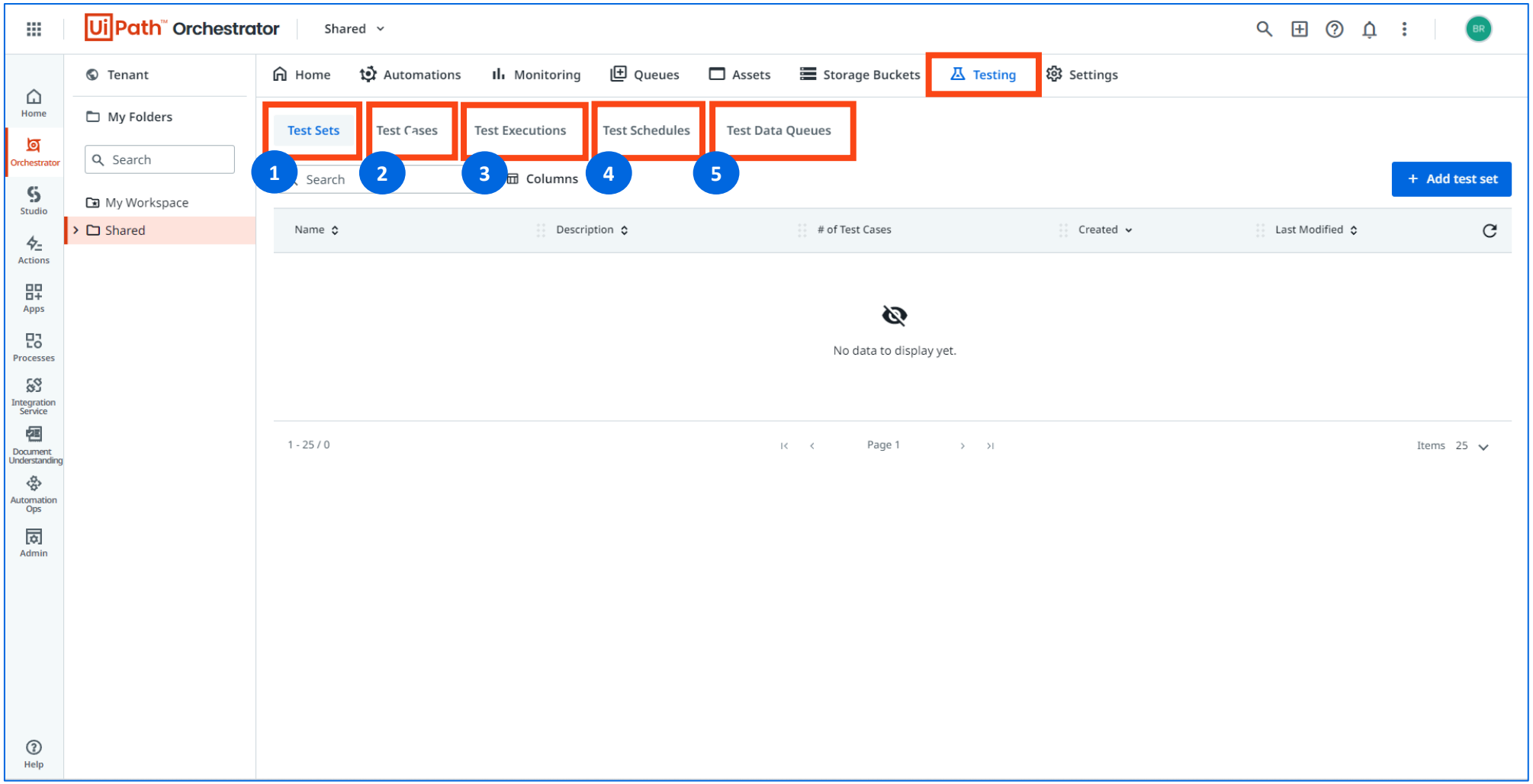
Folders ⟶ Storage Buckets

The Storage Buckets page allows you to manage different types of storage solutions integrated with automation.



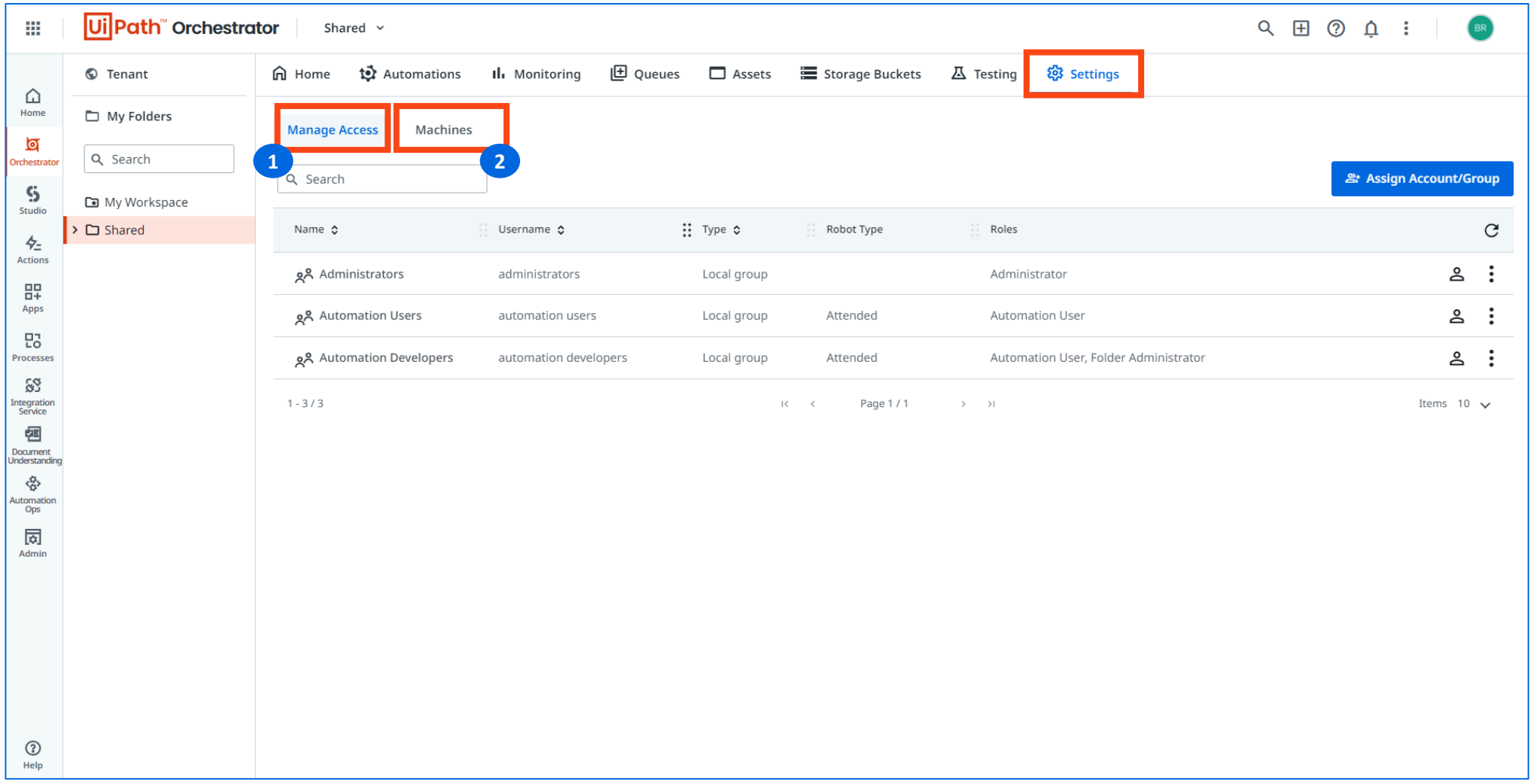
Folders ⟶ Testing

The Testing page allows you to conduct testing.



Folders ⟶ Settings

The Settings page enables you to manage access and machines assigned to the folder.



My Workspace

It is a modern folder available for the dedicated use of a particular attended user.