LECTURE 12. ORCHESTRATOR

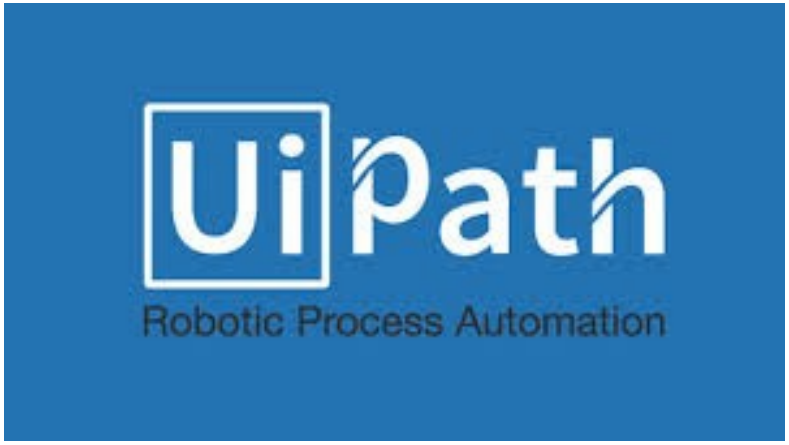
Robotic Process Automation
[19 December 2023]

Elective Course, 2023-2024, Fall Semester

Camelia Chisăliţă-Creţu, Lecturer PhD
Babeş-Bolyai University

Acknowledgements

This course is presented to our Faculty with the support of UiPath Romania.



Contents

Demo 1. Workflow Publishing and Execution

- **Use the workflow defined during Lecture 11 to retrieve e-mails from the gmail.com (using IMAP, where the server is imap.gmail.com, port 993, username: rpaubb, password:** *16charspassword*);
  1. ***Publish*** **the workflow to a local folder;**
  2. ***Run*** **the process from the local Robot.**

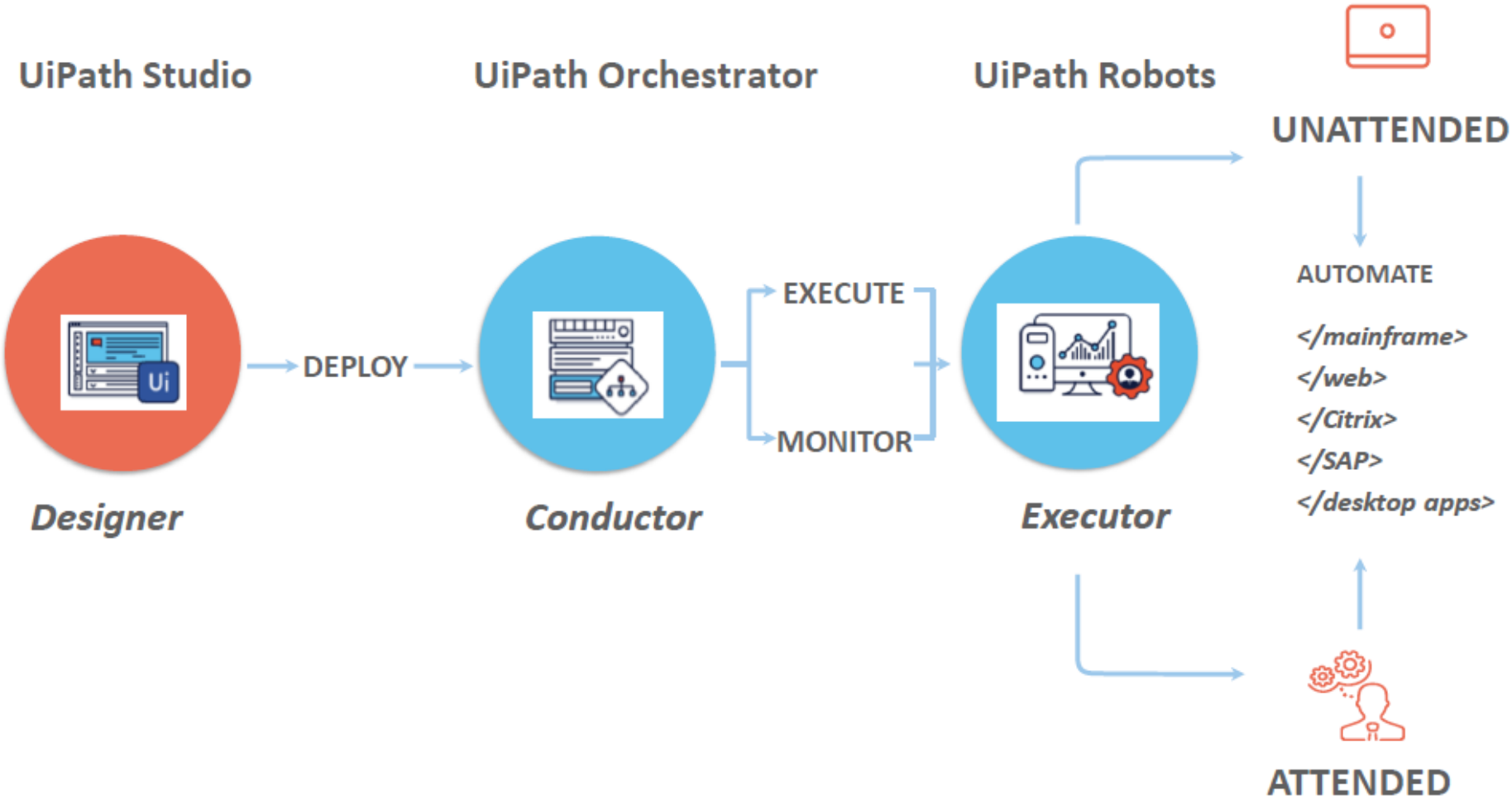
  Publish to Location:
  - **Assistant (Robot Defaults) option – the folder C:\ProgramData\UiPath\Packages**
    - **the results of execution are available in folder**
      - **c:\Users\<user>\.nuget\packages\<project folder>**
  - **Custom option – the folder C:\CustomActivities**
    - **the results of execution are available in folder**
      - **c:\Users\<user>\.nuget\packages\<project folder>**

- **For this demo the Assistant (Robot Defaults) option is used. The UiPath Robot (the local Robot) is not connected to the Orchestrator, yet.**

see **Demo1-WorkflowLocalRobot**

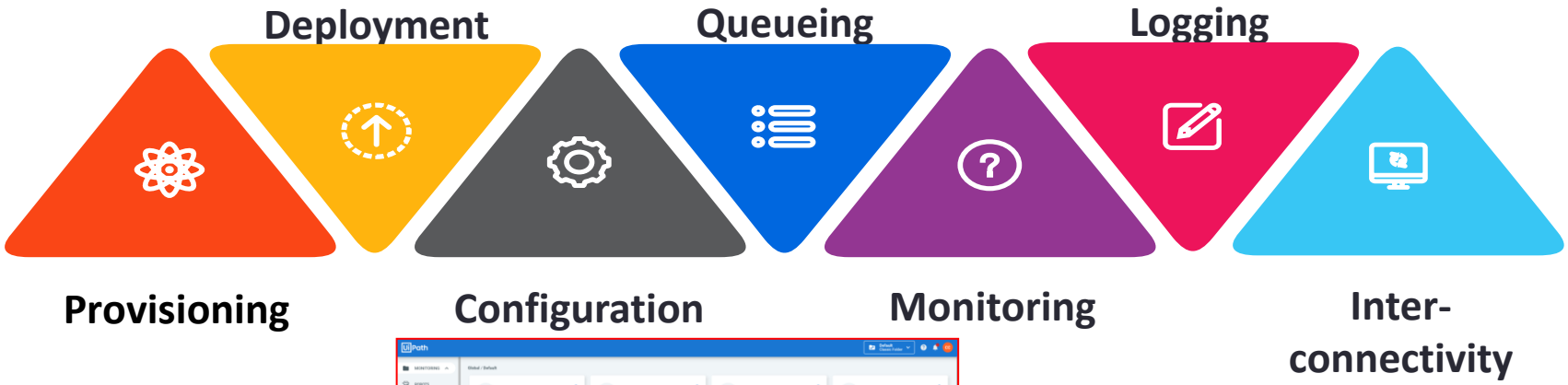
UiPath

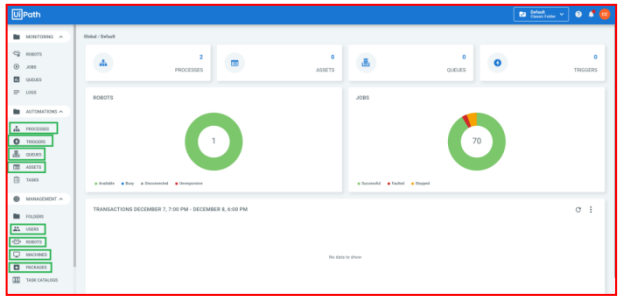
UiPath Platform. Components



Orchestrator. Details

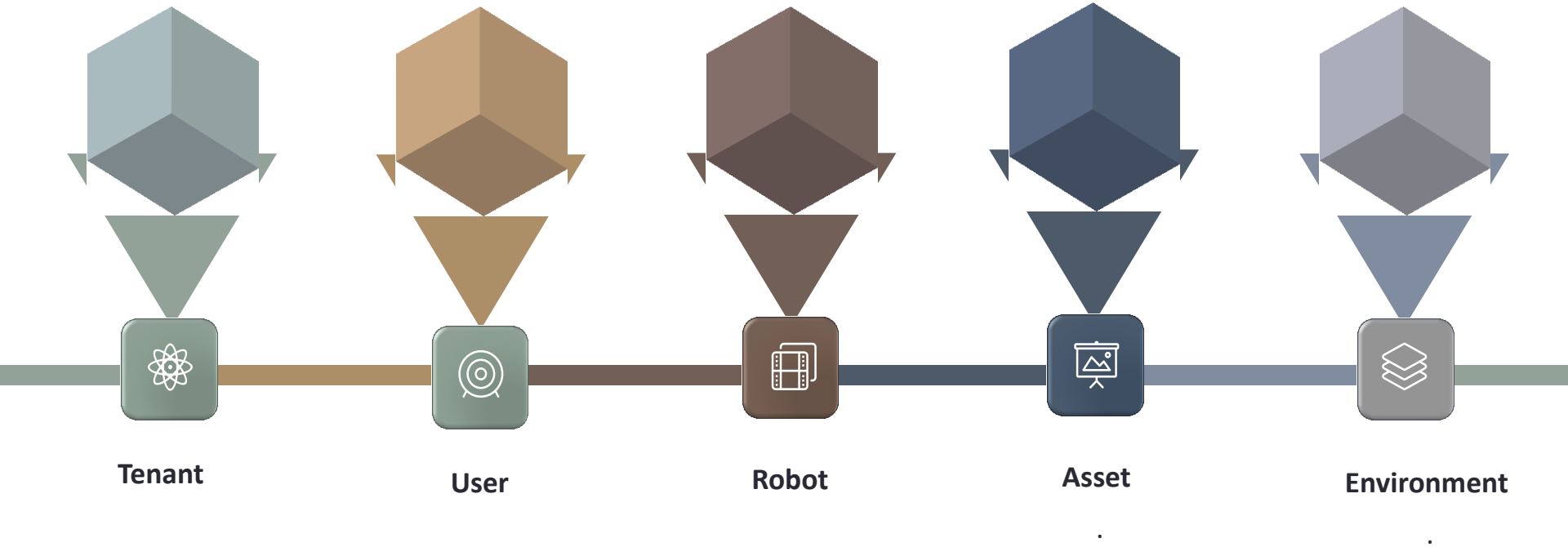
- **Orchestrator** is
  - a web-based key component of the UiPath enterprise RPA platform;
  - used to **deploy, control and monitor** the productivity of **UiPath robots**;
  - a server-based application which is used to regulate and monitor the activities of the **Robot** attached within the network;
- primary functions are follows:





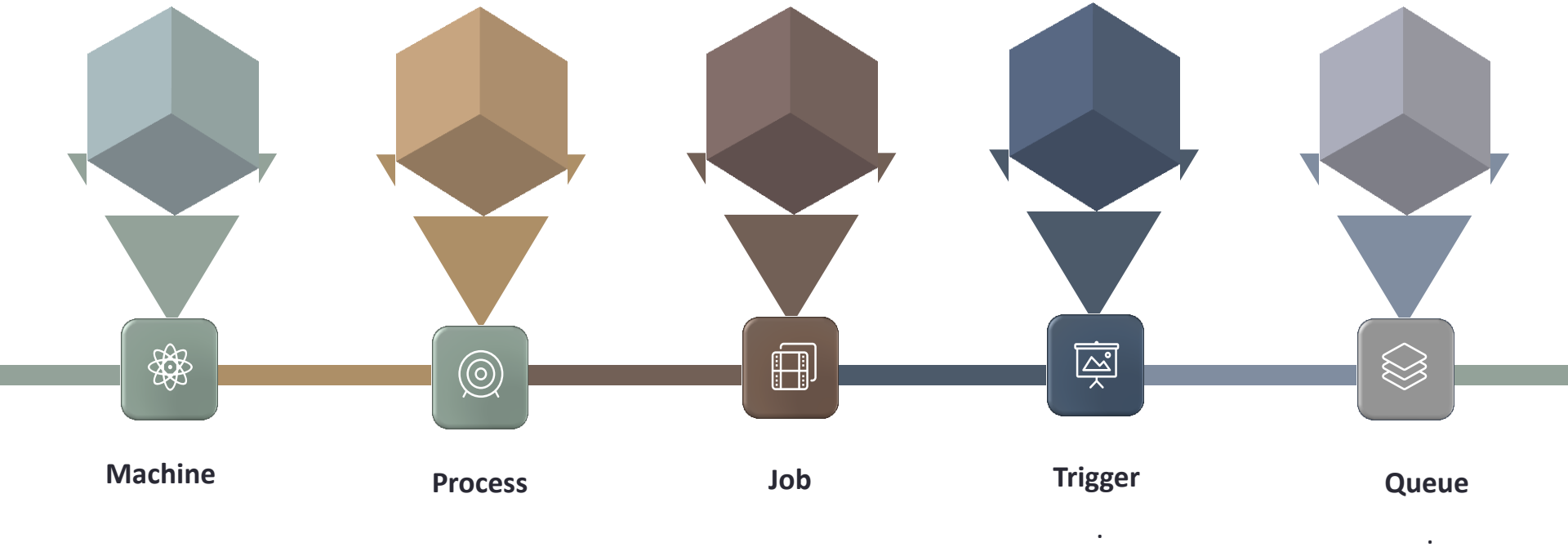
Orchestrator. Components (1)

- Orchestrator consists of several components:



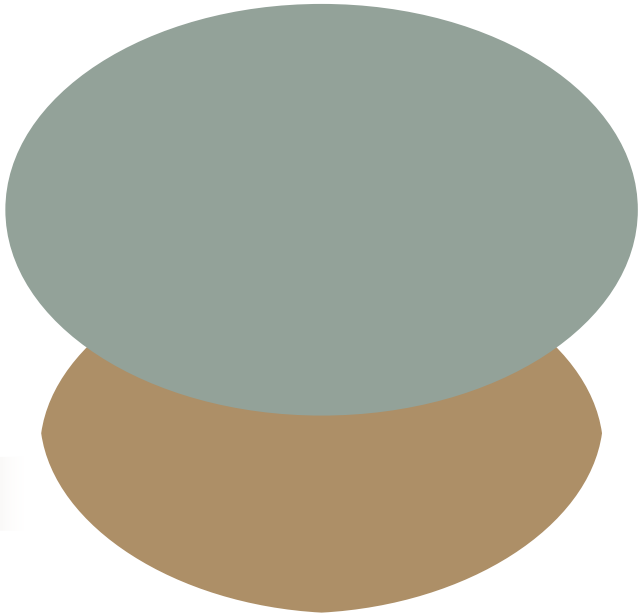
Orchestrator. Components (2)

- Orchestrator consists of several components:



Orchestrator. Functions

- **Orchestrator** acts as an integral part of UiPath platform and has **multiple functions** to perform which makes it versatile in nature.



Functions of Orchestrator

- Deploying a Process
- Using Orchestration Server to *control* Robots
- Using Orchestration Server to *deploy* Robots

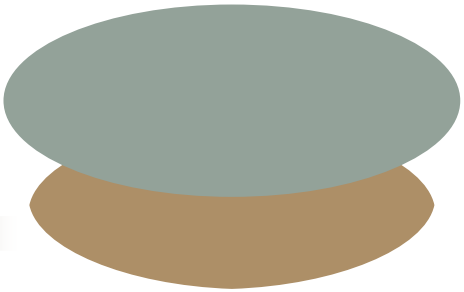
Logical Components of Orchestrator

- User Interface Layer
- Web Services Layer
- Persistence Layer

Orchestrator. Layers

- **Orchestrator** has **logical layers** as follows:

1. **User Interface Layer:**
   - Web Application
2. **Web Services Layer:**
   - Monitoring Service
   - Logging Service
   - Deployment Service
   - Configuration Service
   - Queues Service
3. **Persistence Layer:**
   - SQL Server
   - ElasticSearch



Logical Layers of Orchestrator

- User Interface Layer
- Web Services Layer
- Persistence Layer

Functions of Orchestrator

- Deploying a Process
- Using Orchestration Server to Control Robots
- Using Orchestration Server to Deploy Robots

UiPath

Tenant. Details (1)

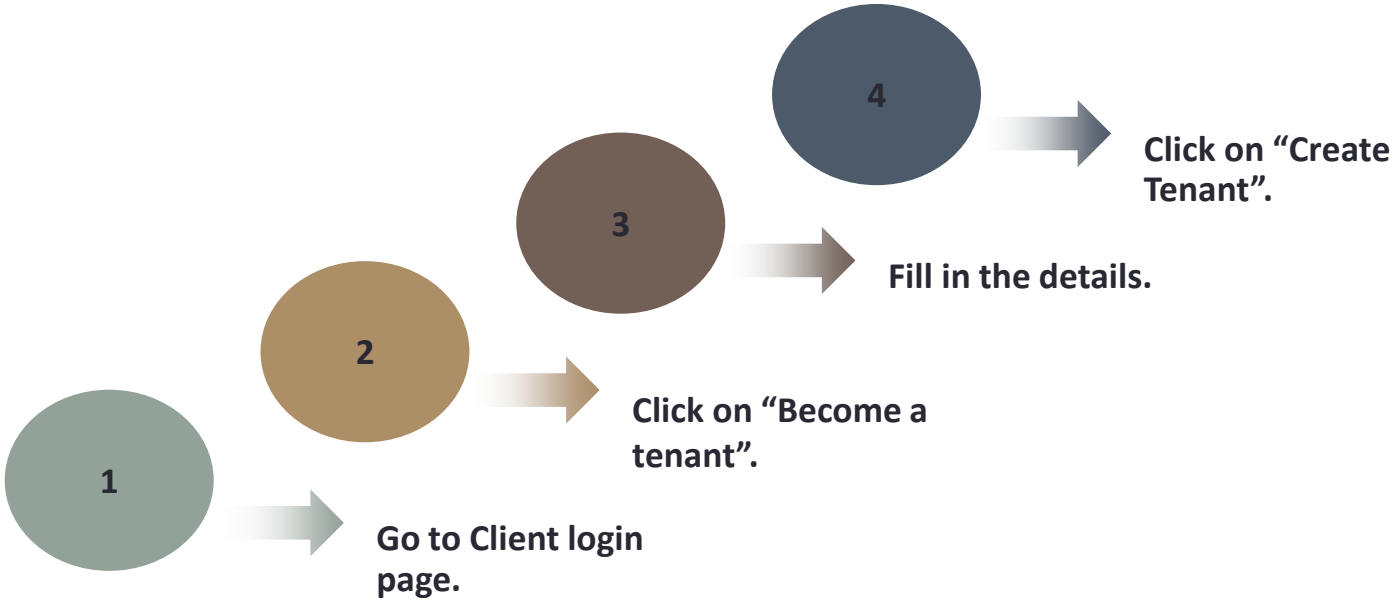
- **Tenant** is
  - a component in UiPath Orchestrator;
  - defined as that user who signs up first with their details in order to use the **Orchestrator**;
- multitenancy allows the user to customize and access information on a single instance on **Orchestrator** where multiple data is stored in the equivalent database.

- **Tenant** registration can be done while installing the Orchestrator; this can be enabled by setting the parameter `Tenant.Registration.Enabled = true` in the `web.config` file; then, Orchestrator will display a login page.

Tenant. Details (2)

- The steps to create a new tenant are:

1. Client **login** page
2. Click on **Become a tenant**
3. Fill in the details required like **Tenancy name, Name, Surname, email address, and admin password.**
4. Click on **create tenant** this will reflect the newly created tenant.



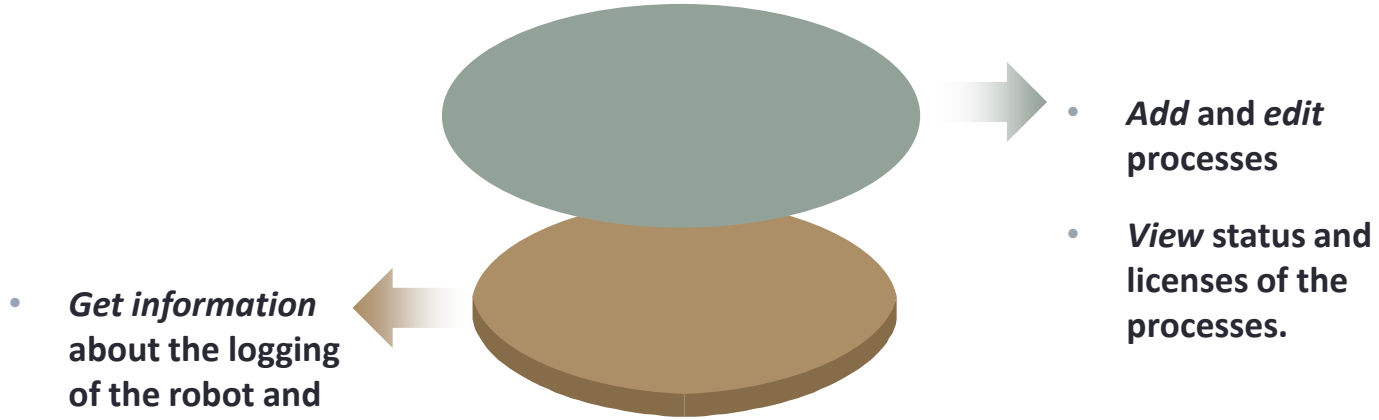
User. Details

- **User** is
  - an entity that stores the assigned role(s), email settings and enables to login to **Orchestrator**;
- there are three different types of users:
  - **Robot**:
    - gets created automatically;
    - can communicate and display the configuration;
  - **User**:
    - is imported from the **Active Directory** or created directly;
    - is used to log into **Orchestrator;**
  - **Admin**:
    - comes predefined with **Orchestrator**;
    - cannot be deleted;
    - can activate, deactivate, and remove other users.

Robot. Details

- **Robot** is
  - a piece of software which can be taught how to execute workflows that contain multiple steps in various applications;
  - capable to mimic any process as well as human actions, e.g.: to move files, copy and paste the data into any folder, to extract structured and semi-structured data.

- when the **Robot** is busy in executing a **process**, we cannot edit or make any changes in username, machine and type.



see **Demo2-WorkflowOrchestrator**

UiPath

Robots. Details

- **Robots** in Orchestrator are of two types:

Attended

- **These robots act as a helping hand to its users;**
- **also known as assistance robot which requires user assistance (user interactions) during the process;**
- **E.g.: the Robot requires the user to provide credentials or displays some message or dialog where the user is required to give the response otherwise further processes will not work.**

Unattended

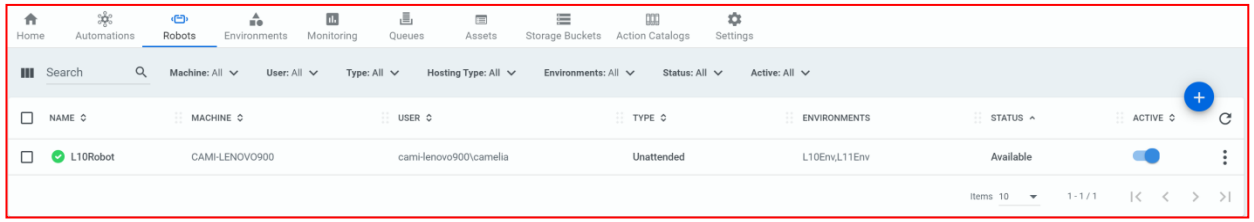
- **These robots run the automated process in unattended mode;**
- **can be started with the help of Orchestrator;**
- **can be scheduled or can be run manually using UiPath Robot.**

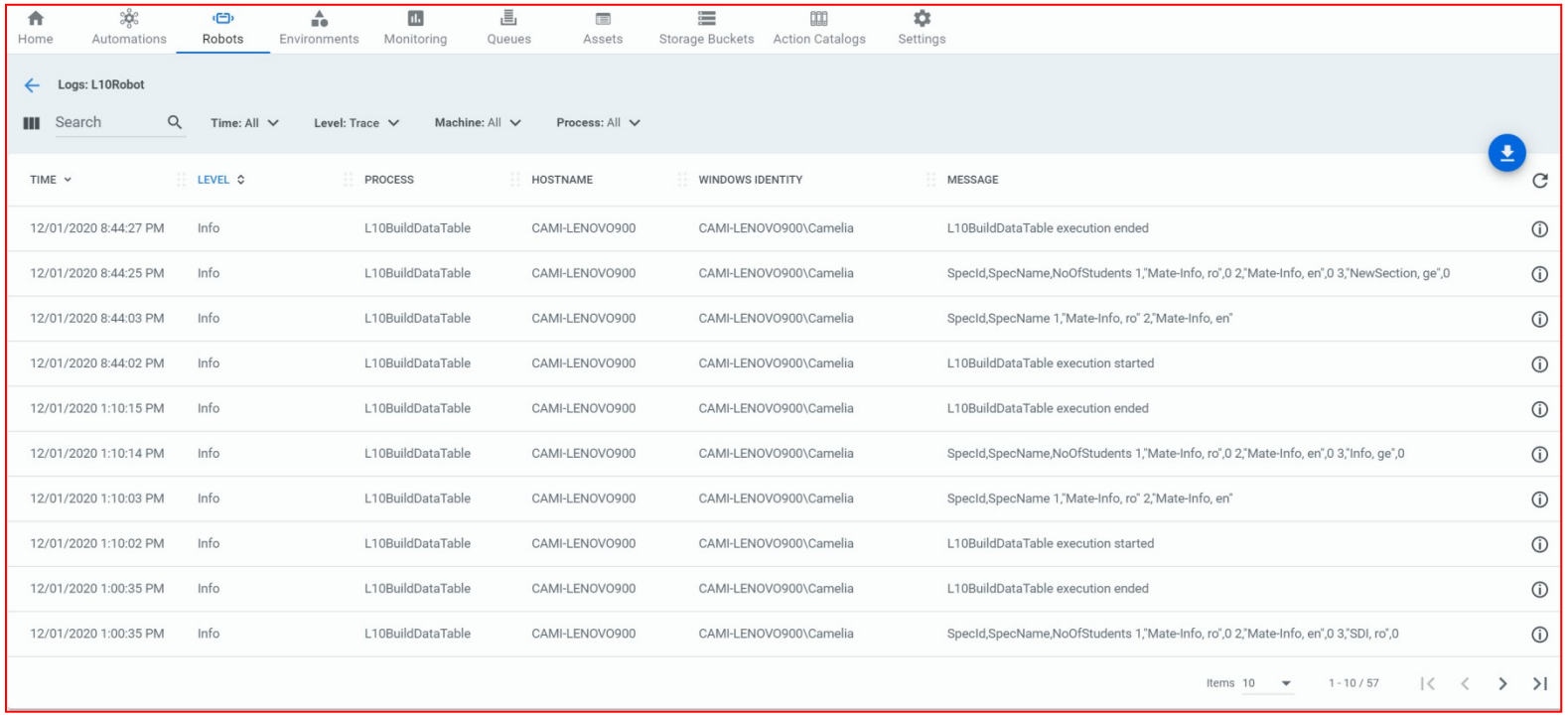
UiPath

Robots. Viewing Logs

- Steps to view the **Robots**' logs:

1. go to the **Robots** tab;
2. search for desired Robot.
3. click on **More Actions**;
4. click on **View logs**.





Robots. Deploying in Orchestrator

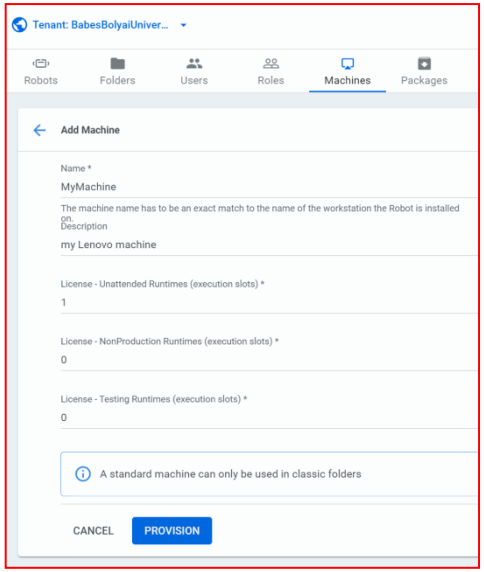
- **deployment** ensures the correct delivery of the package version to the assigned robot for executions;
- to deploy **Robots** to **Orchestrator**, we need to configure a **Machine** with **Orchestrator**;
- Steps:
  1. provisioning a **Machine**;
  2. provisioning a **Robot**;
  3. connecting **UiPath Assistant** to **Orchestrator**;
  4. deploy the **Robot**.

see **Demo2-WorkflowOrchestrator**

UiPath™

Machine Provisioning

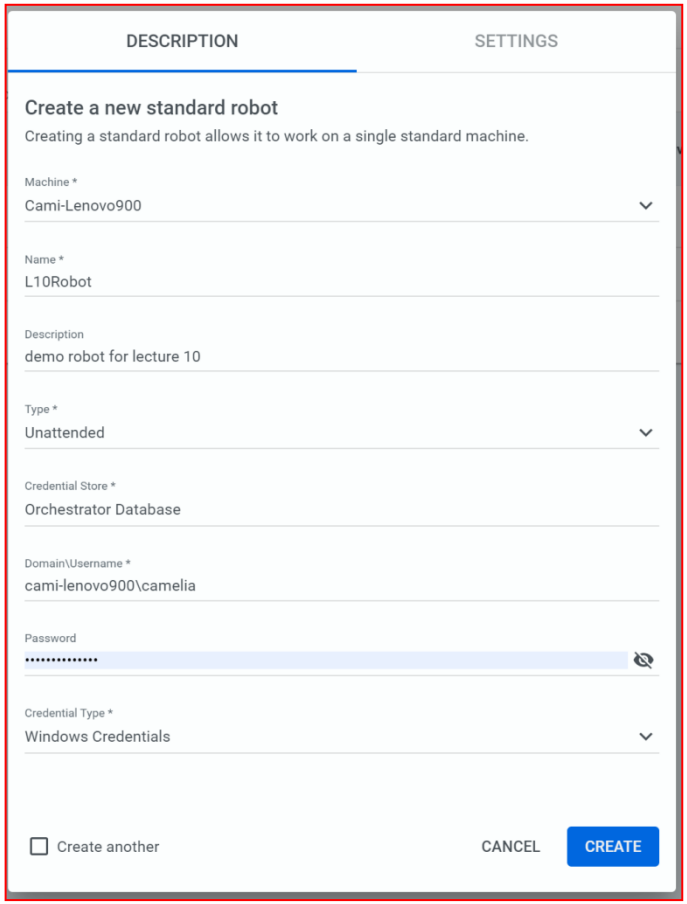
- **Machine** allows
  - to connect various type of robots to specific entities the users work on;
- types of machine:
  - **Template Machine**, tied to **Floating Robots**;
    - useful and suitable when working on a non-persistent VDI, where different computer is used every day;
  - **Standard Machine**, tied to **Standard Robots**;
    - useful when there is a unique user-machine combination in **Orchestrator**;
- Steps:
  1. go to **Machines** tab;
  2. click on **+** sign;
  3. choose **Add Standard Machine** option;
  4. enter the machine name and description;
  5. click on **Provision**.



see **Demo2-WorkflowOrchestrator**

Robot Provisioning

- types of robots:
  - **Floating Robots** connected to **Template Machine**;
  - **Standard Robots** connected to **Standard Machine**;
- Steps:
  1. go to **Robots** tab;
  2. click on **+** sign;
  3. Fill in all the mandatory information:
     - **Machine, Name, Type**
     - **Domain\Username**
       - Taken from Command window, after entering command **`whoami`**
     - **Password**
  4. click on **Create**.



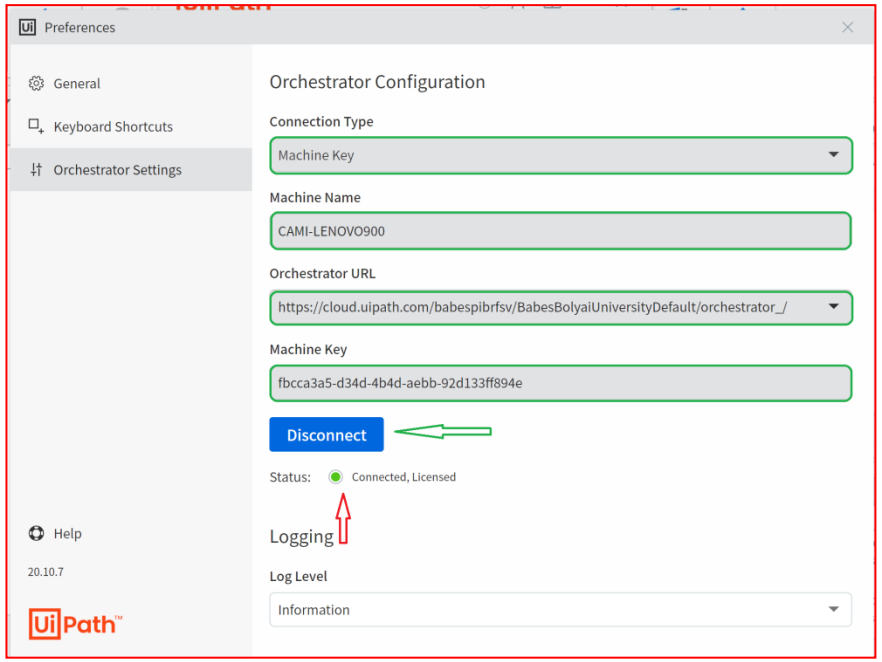
https://docs.uipath.com/orchestrator/v2019/docs/managing-robots

https://docs.uipath.com/orchestrator/v2019/docs/about-robots

see **Demo2-WorkflowOrchestrator**

Connect Robot to Orchestrator (1)

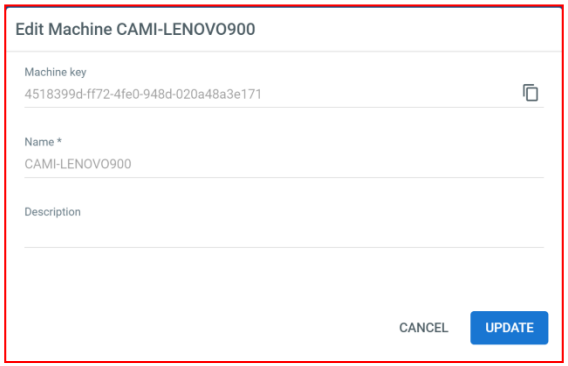
- Steps:
  1. opening **UiPath Robot:**
     - search UiPath Robot in Start Menu or
     - open UiPath Studio folder and then UiPath Robot executable file;
     - open UiPath Robot from System tray;
  2. go in the **Settings** option;
  3. Fill in **Orchestrator URL** with the address taken from the **Orchestrator** web browser;
     - this should be similar to https://cloud.uipath.com/babespibrfsv/BabesBolyaiUniversityDefault/orchestrator_/

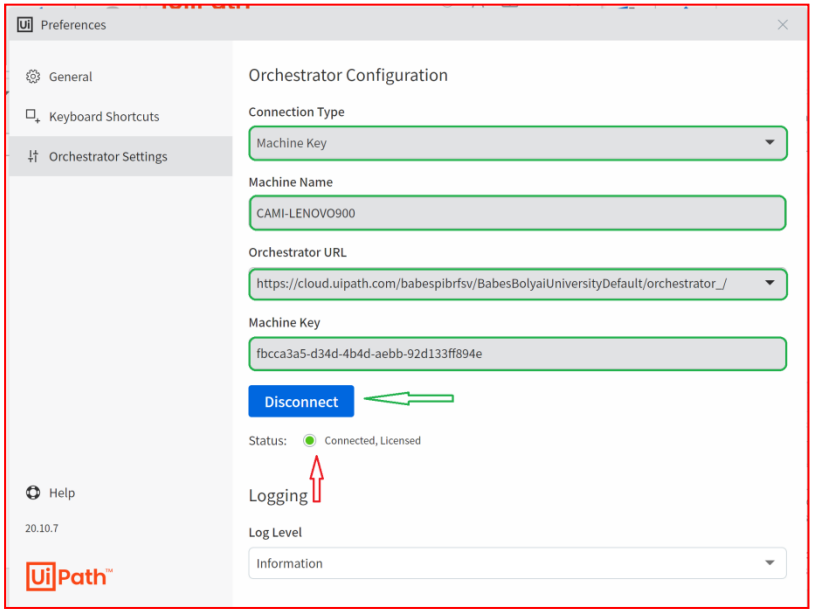


see **Demo2-WorkflowOrchestrator**

Connect UiPath Robot to Orchestrator (2)

- Steps:
  4. and **Machine key** taken from the **Orchestrator** on the **Machine Page**, **Edit window**;
  5. in local UIPath Robot window click **Connect**;
     - the **Status** is changed from ***Offline*** to ***Connected, licensed***.
  6. The **Standard Machine** installed attribute is changed to the local Robot version, e.g., 19.4.5.





see **Demo2-WorkflowOrchestrator**

Demo 2. Process Execution from Orchestrator

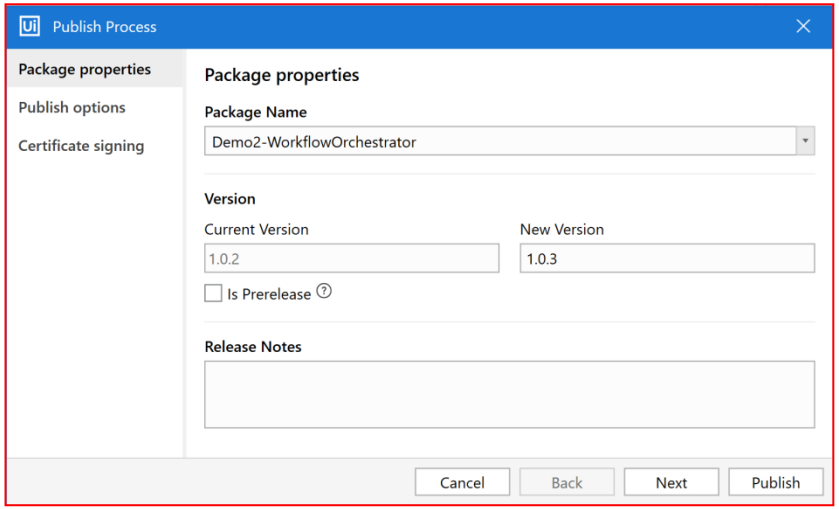
- **Use the workflow defined during Lecture 09 to retrieve e-mails from the gmail.com (using IMAP, where the server is imap.gmail.com, port 993, username: rpaubb, password:** *16charpassword*);
  1. ***Publish*** **the workflow to the Orchestrator;**
  2. ***Define*** **appropriate components in Orchestrator (Machine, Process and Job);**
  3. ***Run*** **the published workflow.**

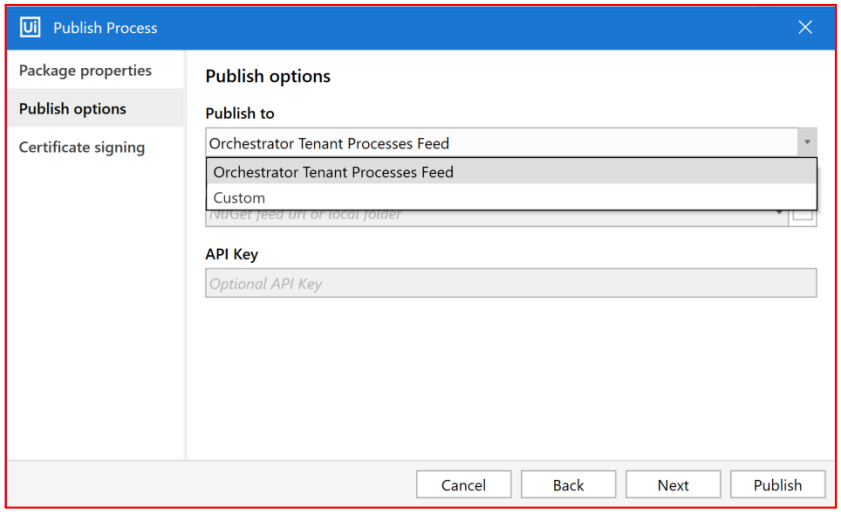
see **Demo2-WorkflowOrchestrator**

UiPath™

Workflow. Publishing in UiPath Studio

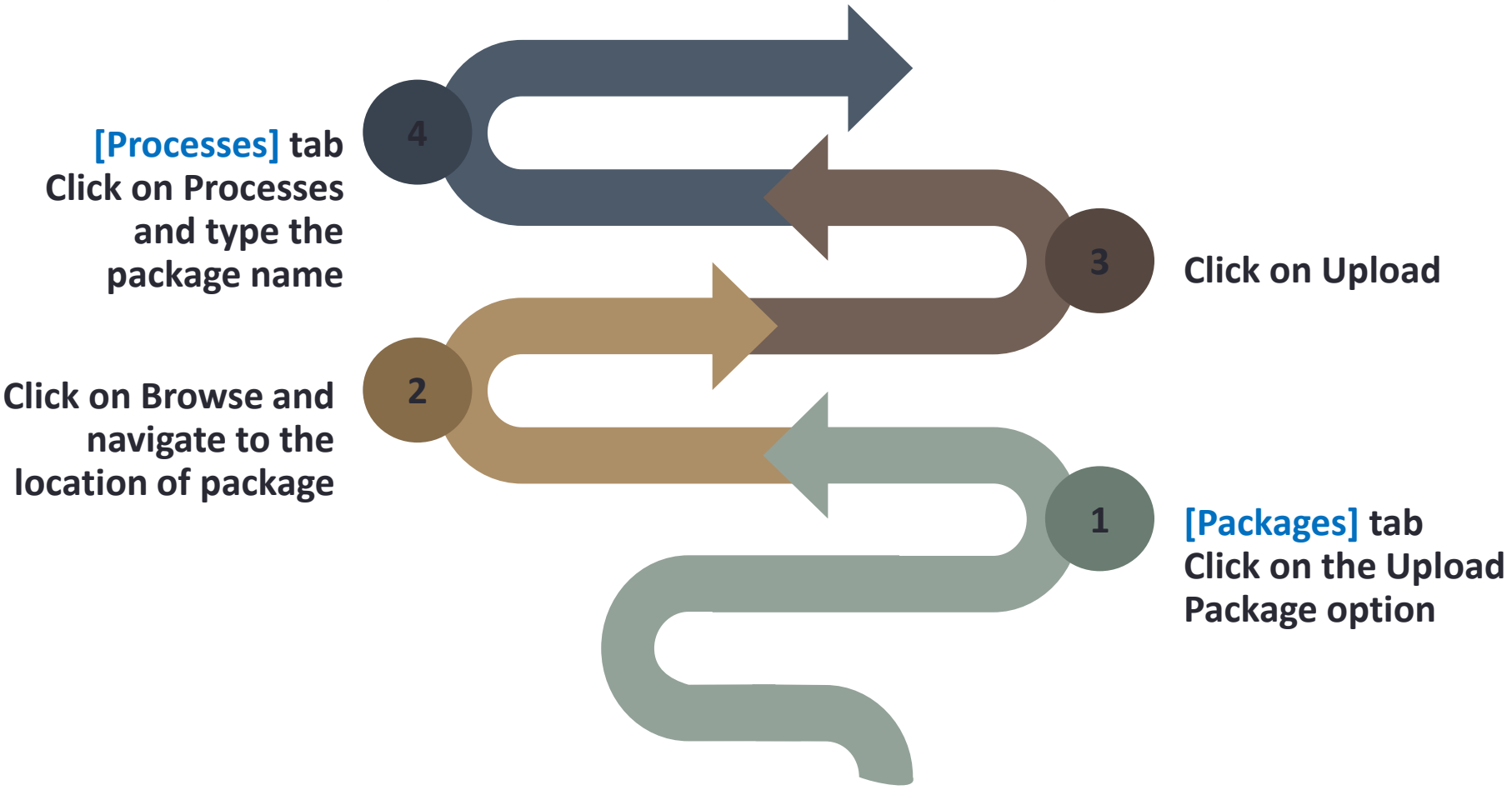
- **publishing** a workflow in UiPath enables the user to run it using **UiPath Robot** or **Orchestrator** as a **package**;
- Steps:
  1. create a **process** type project and give it appropriate name;
  2. go to Setup/Design ribbon and click on **Publish** and an Info dialog box displays:
     - URL of the Orchestrator;
     - name of the package that has been published;
     - version of the package published to **Orchestrator** (**Orchestrator Personal Workspace Feed**);
  3. click **Publish** button.





Workflow. Add a Package Manually

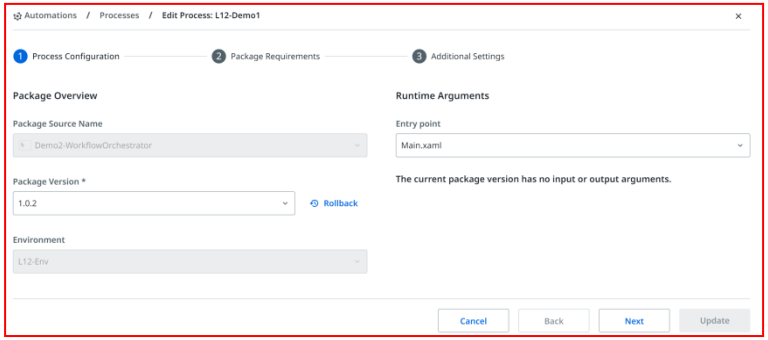
- Steps to add a package manually to the **Orchestrator** in **Packages** tab and **Processes** tab:

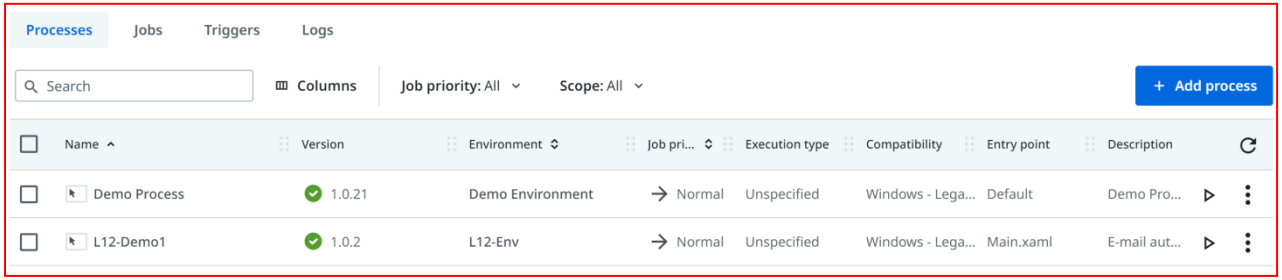


Processes. Details

- **Process** is
  - used to deploy a published **Package**, it keeps the packages updated;
- Steps:
  1. go to the **Automations** tab;
  2. go to the **Processes** section, click on **+** sign;
  3. select the package name, its version, and its description.
  4. click on **Create**;
- as a result, a new process is being created

which can be executed in the **Jobs** section or by clicking the ***run*** button.

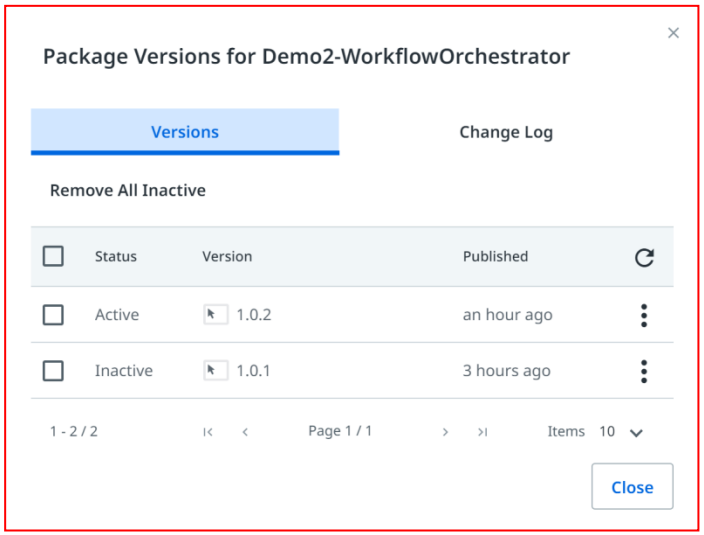


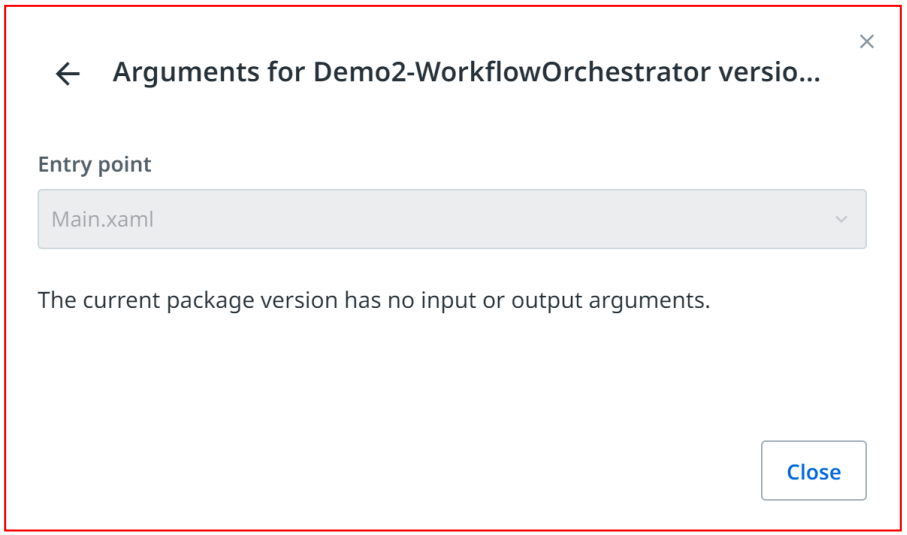


see **Demo2-WorkflowOrchestrator**

Processes. Parameters

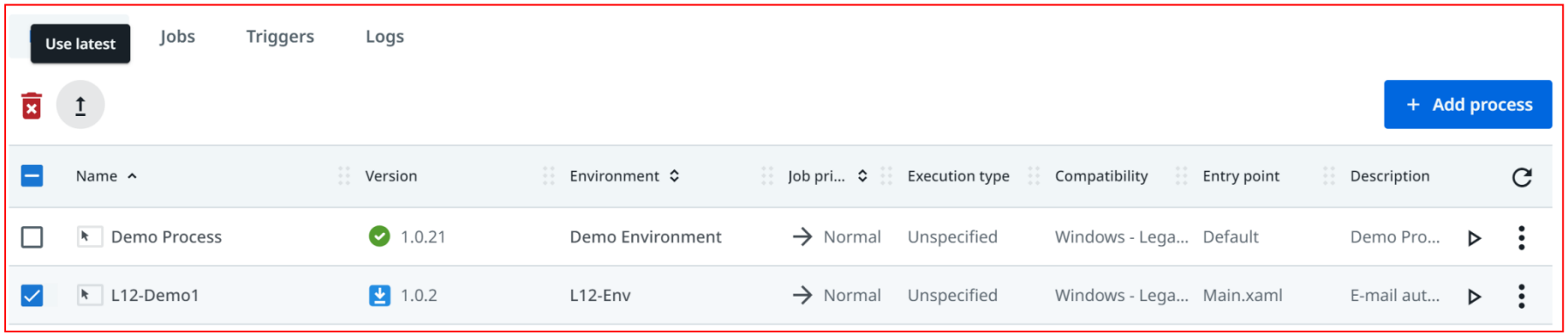
- when the workflow is successfully completed, we cannot afford to open UiPath Studio again and again to run the workflow;
  - to use a workflow directly from the **UiPath Assistant**, we have to publish the workflow and then schedule it through **Orchestrator**;
  - when the workflow is published, it can be directly run a **UiPath Assistant**from **Orchestrator**.
- if the published workflow has **arguments**, they are displayed in the **Show Arguments** option in the *3 dots menu* of the selected package.





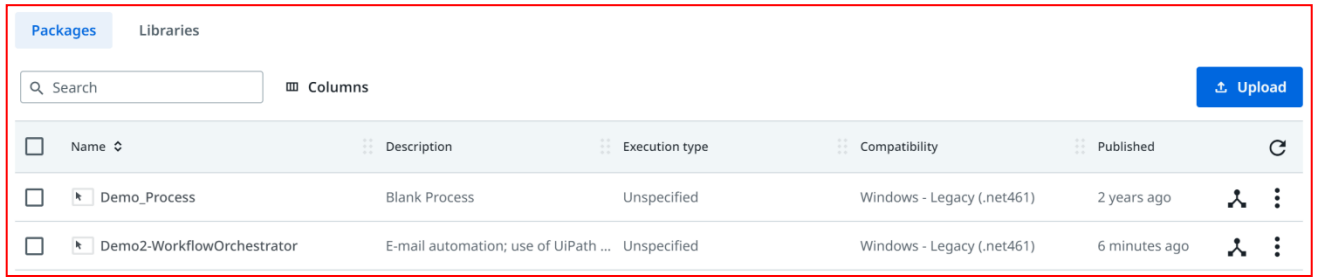
Processes. Workflow Version Management

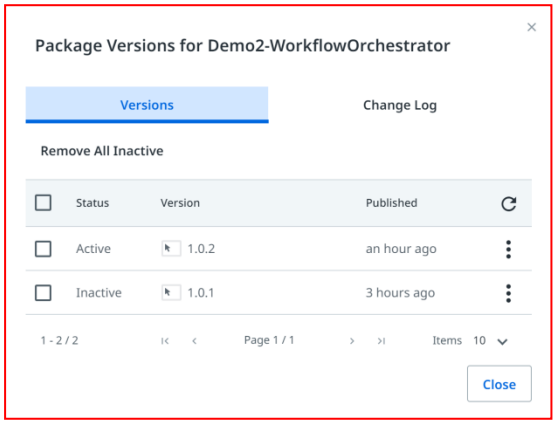
- when publishing the same workflow from the UiPath Studio, a **new version** is added, and a **blue symbol** is indicated corresponding to the **Process**;
  - hence, the **process version** can be *updated* or *rolled back* as desired, e.g., Use latest option allows to update to the last version, the packages that correspond to the selected processes.
- at every stage a **Package** is linked to the **Environment** and it is automatically distributed with each **Robot** machine that belongs to that **Environment**;
- all activities that are used in UiPath Studio are stored in the **NuGet feed** that **Orchestrator** has access to.



Packages. Details

- after a **Package** has been created on the **Orchestrator Server**, it can be viewed in the **Packages** tab;
- details on the versions of selected **Package** are available by choosing **View versions** option;
- types of packages:
  - **Active**: meaning that the version is currently in use;
  - **Inactive**: meaning that the version is not in use, but the user can restore the system to use it.

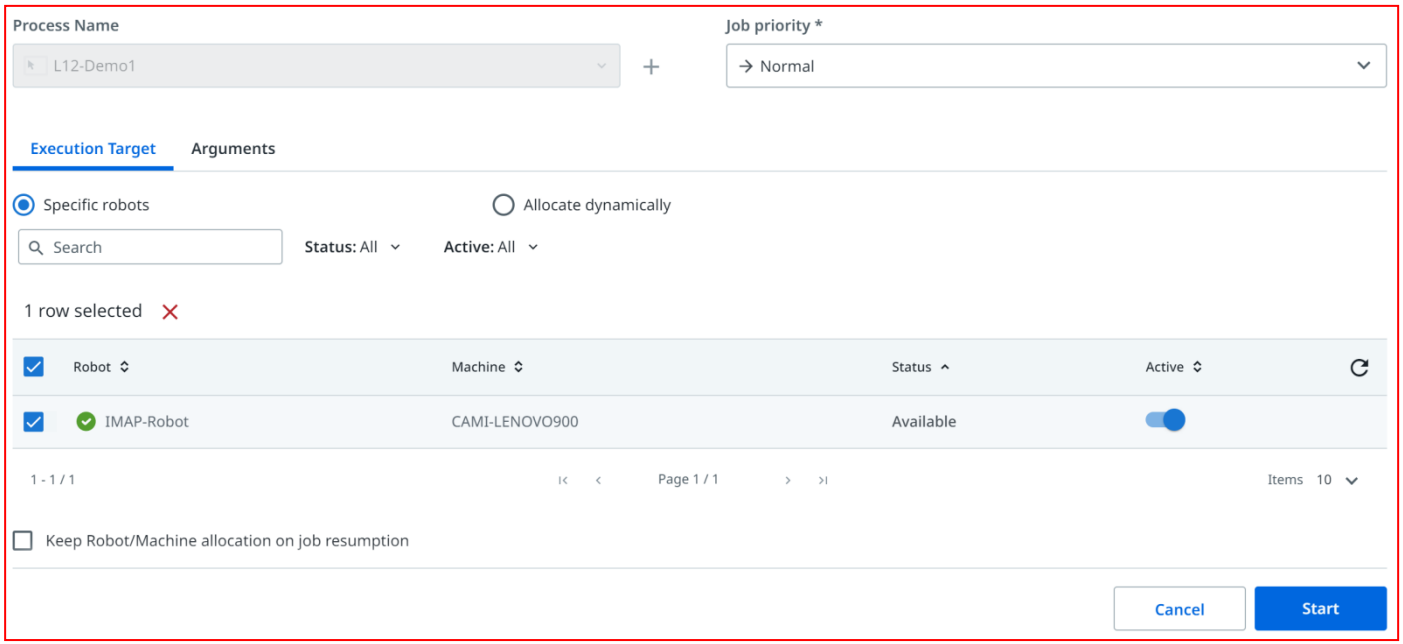




see **Demo2-WorkflowOrchestrator**

Jobs. Details

- **Jobs** are
  - entities assigned to **Processes** in order to execute **Packages**;
- they are assigned to **Processes** from the **Jobs Panel** available **Automations** option of the **Orchestrator.**
- **Processes** assist in distributing packages over the **Robot** machine, which makes execution faster.

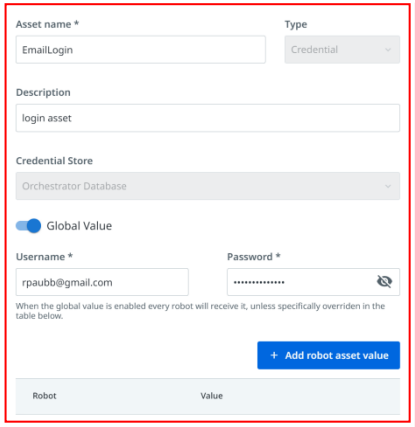


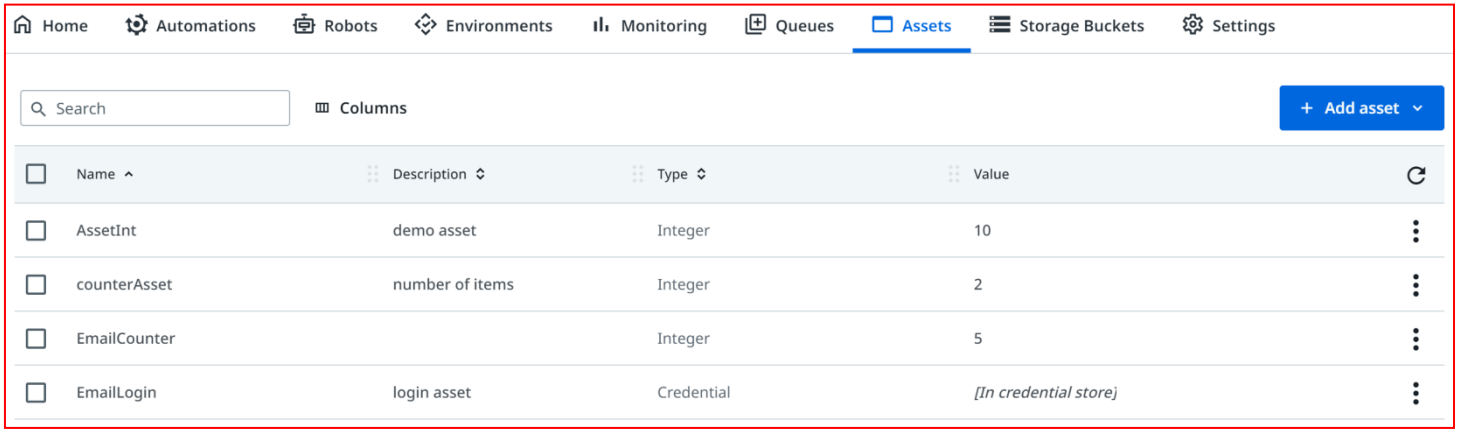
see **Demo2-WorkflowOrchestrator**

Assets. Details

- **Assets**
  - work as **variables** or **credentials** and can be used in distinct automation projects.
- it yields the possibility to operate the specific information, that can be easily obtained by the **Robot** in the **Activities Panel**;

- **Assets Page** shows all the previously created assets that can be removed or edited;
- data credentials are encrypted using the **AES 256** encryption algorithm;
- when an RPA developer is developing a process the asset can be invoked in any activity to from **Orchestrator;** the information will be hidden.



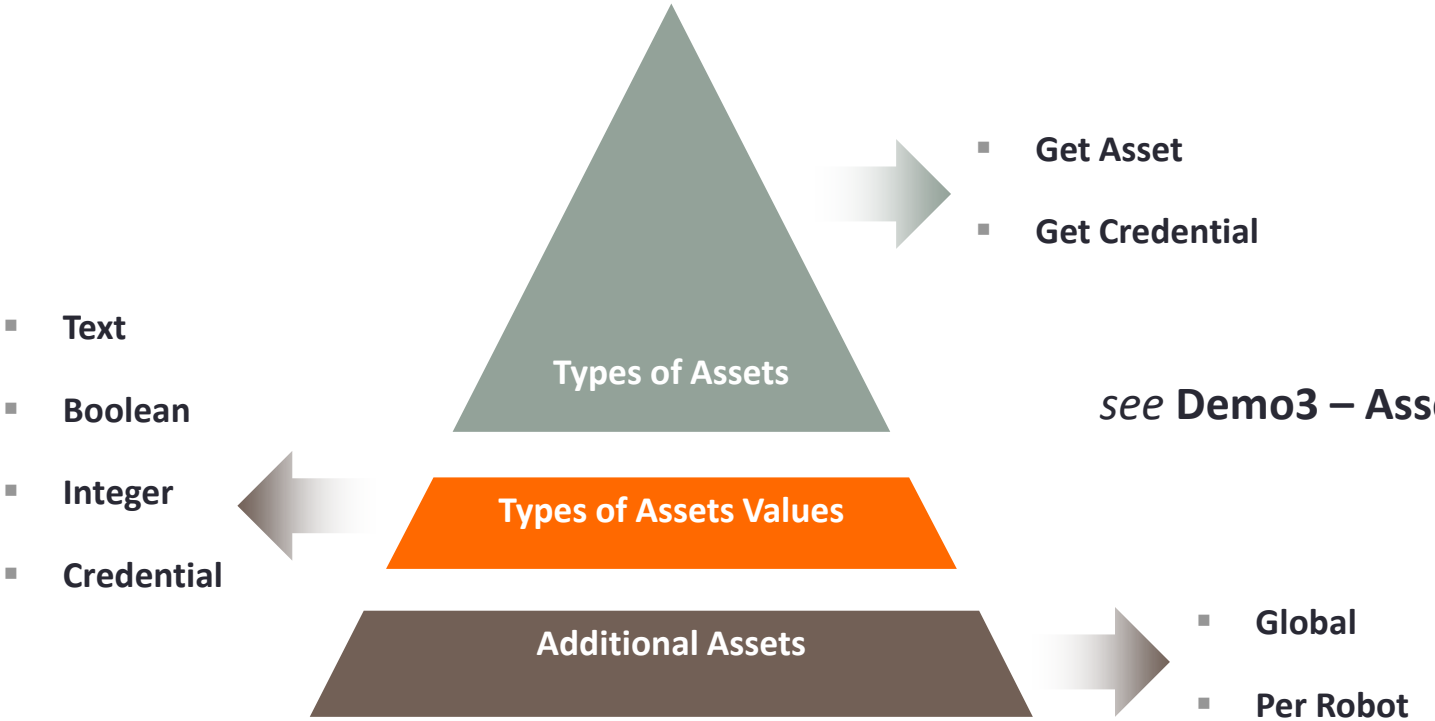


see **Demo2-WorkflowOrchestrator** *see* **Demo3 – AssetsTriggers**

Assets. Types

- **Assets** work as **variables** or **credentials** and can be used in distinct automation projects.

- it has types:
  - **Global**: it can be accessed and used by all available **Robots**.
  - **Per Robot**: it can be accessed only by a specified **Robot**.

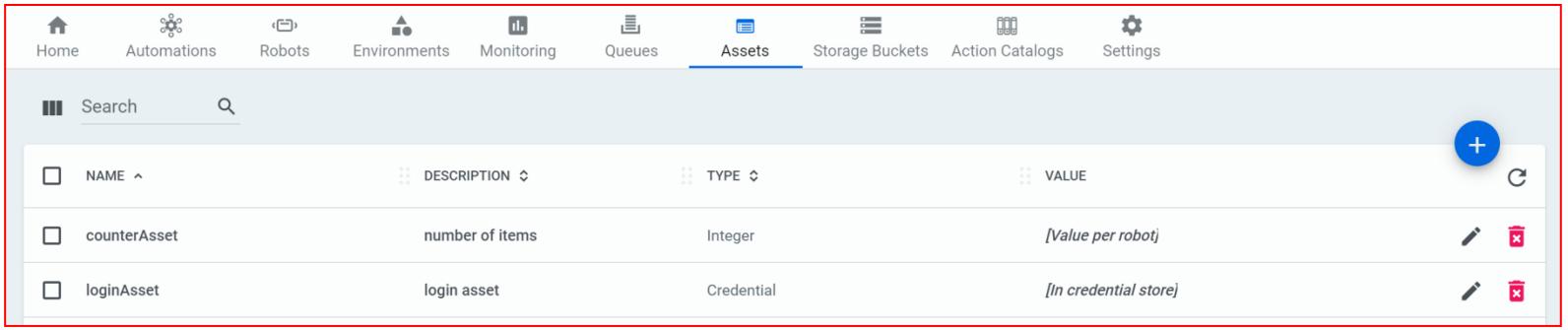


see **Demo3 – AssetsTriggers**

Assets. Activities

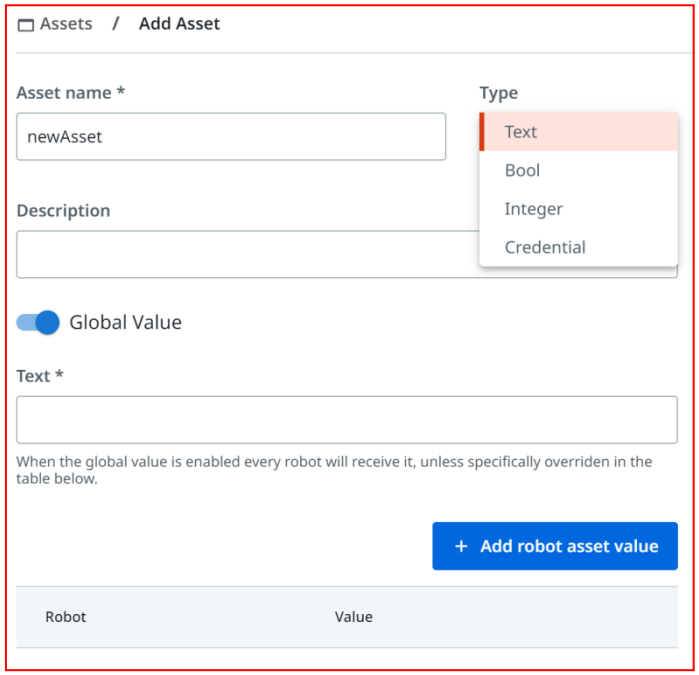
- **Get Asset** and **Get Credential** activities are used in Studio to request information from **Orchestrator** about a specific **Asset**, according to a provided **AssetName**.

- **AssetName** is required for an already stored asset in the **Orchestrator** database so that the **Robot** can access the information stored in **Asset**; the **Robot** needs permission to retrieve information from that particular asset to be used in the automation project, by using the **Get Asset** and **Get Credential** activities.



Assets. Asset Value Types

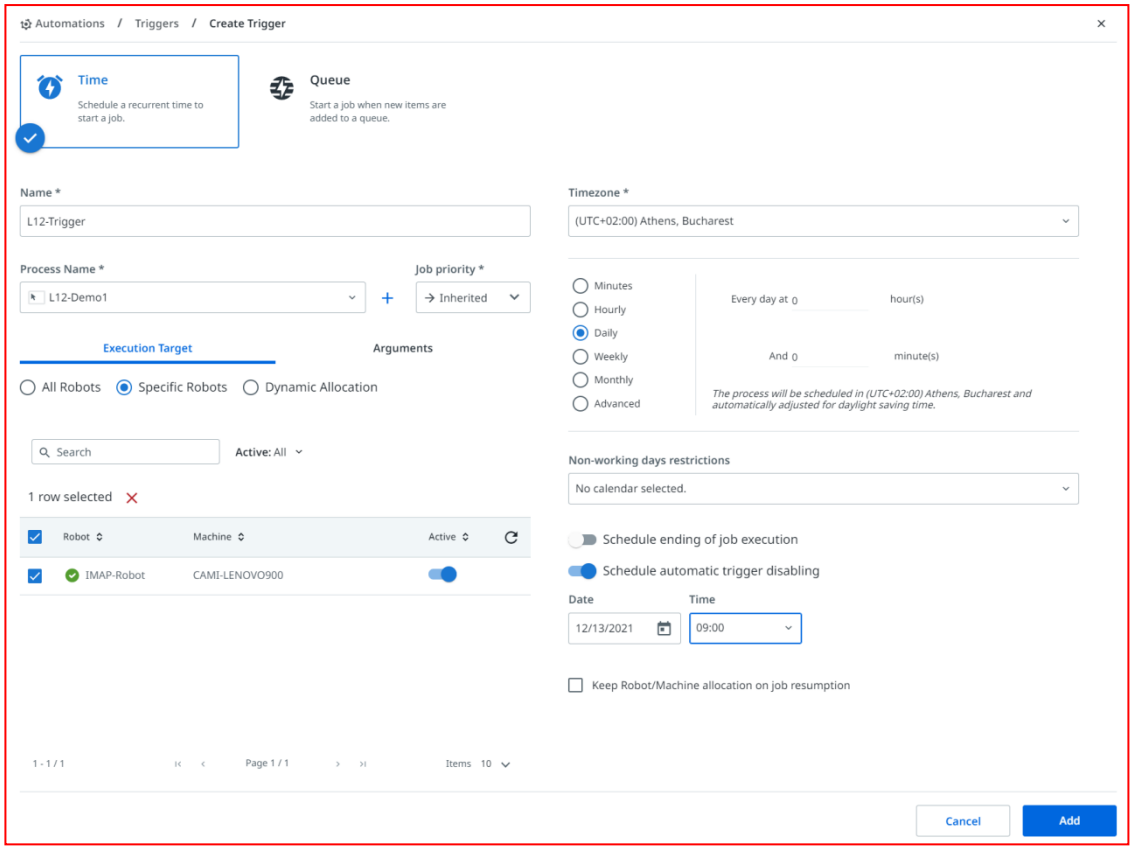
- there are four types of asset values:
  - **Text**: it holds **string** values.
  - **Boolean**: it supports only **true** or **false** values.
  - **Integer**: it stores **integer** values.
  - **Credential**: it holds **usernames** and **passwords** that are needed by the **Robot** to execute specific actions, such as login actions.



see **Demo3 – AssetsTriggers**

Schedulers. Details

- **Schedulers** or **Triggers** are used when the execution of **Jobs** can be planned in such a manner that it starts its execution following a desired pattern, repeats at regular intervals and in a desired time.
- Steps:
  1. go to the **Automations** tab;
  2. go to the **Triggers** section;
  3. click on **+** sign;
  4. Fill in the data with:
     - **Name** of the trigger;
     - **Process** from the drop down list;
     - configure the desired time zone and the repetitive pattern (days, minutes, **Every**…);
  5. if desired, choose a **Robot**;
  6. click on **Add**.

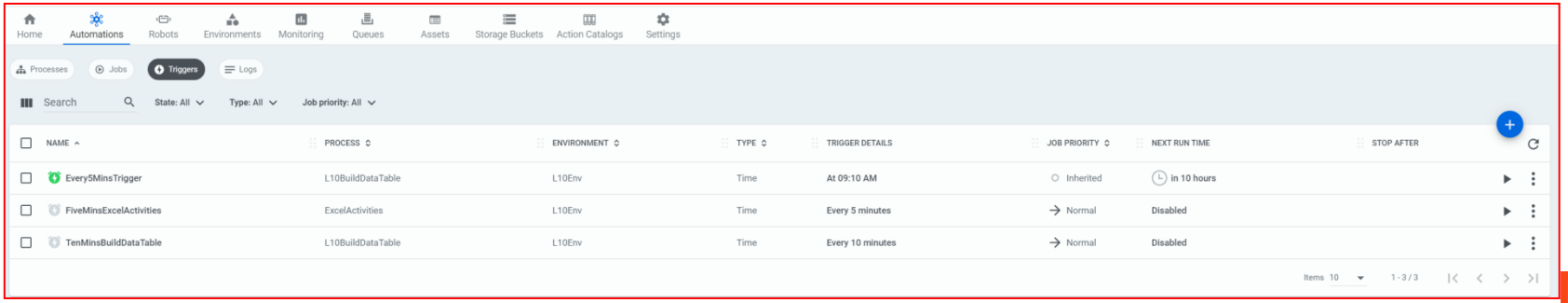


see **Demo2-WorkflowOrchestrator**

see **Demo3 – AssetsTriggers**

Schedulers. Options

- while creating a trigger, various other options are also available:
  - restrict the schedule on **non-working days** (which can be defined in the **Settings Page**);
  - repeat the schedule on minutely, hourly daily or weekly;
  - assign the schedule to ***all*** robots, ***specific*** robots or ***dynamic allocation*** from the **Execution Target Tab**;
  - **Stop** or **disable** the schedule after some specific date or time period;
  - give the ***input*** or ***output parameters*** to the process from the **Parameters Tab**;
- if a single robot is assigned to more than one scheduler, then they will form a queue and they are performed in chronological order.
- the non-executing **Scheduler** goes into a pending state until it is executed by the **Robot**.



Demo 3. Assets and Triggers in Orchestrator

- **Use the workflow defined during Lecture 09 to retrieve e-mails from the gmail.com (using IMAP, where the server is imap.gmail.com, port 993, username: rpaubb, password: *16charpassword*);**
  1. **Adapt workflow such that credentials to access the Gmail account are received from Orchestrator's Assets;**
  2. ***Add* a Trigger for the process to be executed every 5 minute for the next 20 minute, then disable the Trigger.**

  - **Check whether the actual Robot in Orchestrator has the privileges set accordingly ---> to set/update the Asset values (Tenant| Manage Access|Robot tab| Folder Permissions tab| check Edit privilege for Assets).**
  - **Define an output parameter for the number of e-mails; check the Job log for the output parameter value.**

see **Demo3 – AssetsTriggers**

UiPath

Queues. Activities

- **Queue** items can also be added from **UiPath Studio** by using several activities:

Add Queue Item

- to add a new item to the queue in Orchestrator and then we can get the new item status;

Add Transaction Item

- to add an item to the queue to begin transaction and set the status as **In Progress;**

Get Transaction Item

- to get an item from the queue to process it and set its status as **In Progress;**

Postpone Transaction Item

- to define time parameters between which transaction should be processed;

Set Transaction Progress

- to assist and create custom progress statuses for **In Progress** transactions;

Set Transaction Status

- to modify the status of the transaction item; whether it *fails* or is *successful*.

see **Demo4 – Queues**

Demo 4. Queues in Orchestrator

- **Use the workflow defined during Lecture 09 to send e-mails from the gmail.com (using SMTP, where the server is smtp.gmail.com, port 465, username: rpaubb, password: *16charpassword* to the e-mail address rpaubb@gmail.com);**
  1. ***Publish* the workflow to the Orchestrator;**
  2. ***Define* appropriate components in Orchestrator (Machine, Robot, Environment, Process and Job);**
  3. ***Run* the published workflow;**
  4. ***Define* a Queue to store the pairs (recipient, recipient e-mail) and use them to send custom e-mail.**
     - **(Anna, rpaubb@gmail.com);**
     - **(George, rpaubb@gmail.com);**
     - **(Alex, rpaubb@gmail.com).**

see **Demo4 – Queues**

UiPath

References

- UiPath Docs - https://docs.uipath.com/
- UiPath Studio Docs - https://docs.uipath.com/studio/standalone/2023.4
- UiPath Forum - https://forum.uipath.com/
- UiPath Academy - https://academy.uipath.com/