

Understanding UiPath Platform Constructs and Setup

Robotic process automation (RPA) adoption is expected to grow at a much faster rate in upcoming years. As UiPath Platform is one of the market leaders in this space, the demand for UiPath Platform will grow, and so will the importance of UiPath support and administrator roles in the companies. Let's get started with the first chapter.

The first chapter will introduce UiPath Platform products and explain the core Orchestrator architecture components. It will also cover the basic setup of Orchestrator, Robots, and the UiPath program organization.

Here is what you will learn as part of this first chapter:

- Get an overview of UiPath Platform
- Understand UiPath Orchestrator and Robots setups
- Difference between on-prem and cloud setup
- Learn about Robot types and licenses in detail
- Real-life environment setup

UiPath Platform overview

UiPath is an RPA platform company that was started back in 2005. They began to get real traction in 2015, and as of 2022, they are the leader in the RPA vendor landscape. The core product lineup is made up of the three products listed here, as any basic RPA use cases can be automated, deployed, and monitored with these core products:

- **UiPath Studio:** The developer platform to create the automation scripts
- **UiPath Orchestrator:** The core platform to run and monitor the Robot's jobs
- **UiPath Assistant:** Robot client software installed in runtime environments to execute the jobs

All UiPath customers need to set up this three-core software to start their automation journey. As UiPath evolved into a hyper-automation platform, it started to add new products to complement its core process automation offering. It's important for any UiPath support and administrator to have high-level knowledge of all the UiPath products. They might need to set up and support them as and when required.

Note

The products can be installed separately on-prem and subscribed as a cloud package in the UiPath automation cloud. Recently, the Automation Suite offering combines all these different products and can be leveraged as a Kubernetes container deployed and used on-demand.

There is a whole suite of UiPath products to support the entire automation journey of any organization. The product inclusion is solely dependent on the UiPath Enterprise customer needs, and the list of available choices is listed in the following figure:

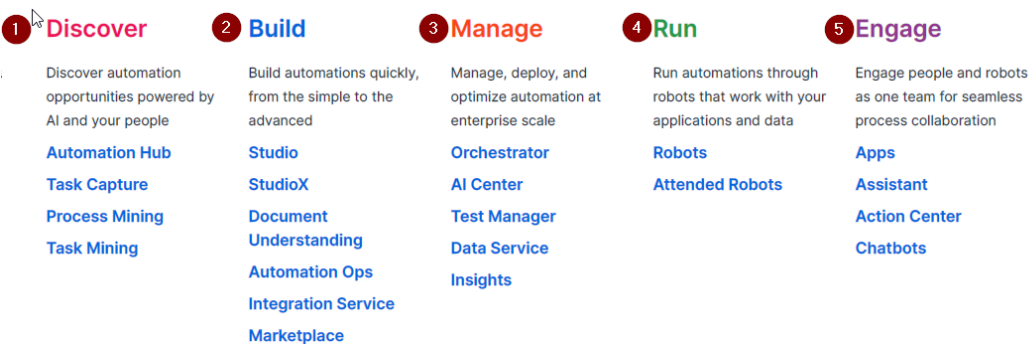


Figure 1.1 – UiPath product portfolio (as of January 2022)

Note

Reference information for all the UiPath products can be found at this link: <https://docs.uipath.com>.

Starting from the 2021.10 LTS release, the UiPath suite supports Linux-based OS, and the new offering: Automation Suite is preconfigured with Kubernetes containers and management tools.

UiPath Enterprise customers can choose cloud or on-prem versions of the products based on their needs.

In a few cases, they might even switch from an on-prem version of the products to a cloud version or the other way around. As all the products are supported in both editions, the UiPath support and administration team must be educated and trained to handle both scenarios.

Suppose a new product, say process mining or test manager, is added to the UiPath environment. In that case, the UiPath support and administration team needs to be equipped to start supporting these additional products along with the core software such as Orchestrator, Studio, and Robots/Assistant.

Let's consider two UiPath support administrators named as follows:

- Patty, who works at XYZ Bank, needs to set up Orchestrator on-prem
- Candace, who works at ABC Insurance Corporation, needs to enable the cloud Orchestrator in her organization to explain the concepts in the following sections

On-prem UiPath Orchestrator

Learning about UiPath Orchestrator's inner details is mandatory for any successful UiPath support and administration professional. If Orchestrator is installed on-prem, it is even more important to learn about it. There are two ways Orchestrator can be configured:

1. **Single node setup** – One instance of UiPath Orchestrator is installed
2. **Multi-node setup** – Multiple instances of UiPath Orchestrator are installed

Before Patty can start the actual installation, she must understand the underlying architecture of these two options.

Let's see them in detail in the next section.

Single node UiPath Orchestrator architecture

As the name suggests, a single node will have Orchestrator installed in one instance. As per UiPath documentation, a single node can support small or medium-scale deployment of Robots (i.e., 1 - 250 unattended Robots or 1 - 2,500 attended Robots). There is no failover plan, hence, downtime is expected during maintenance windows or server outages.

This is the preferred model for any proof of concept and pilot run since the time and cost to set this up are low:

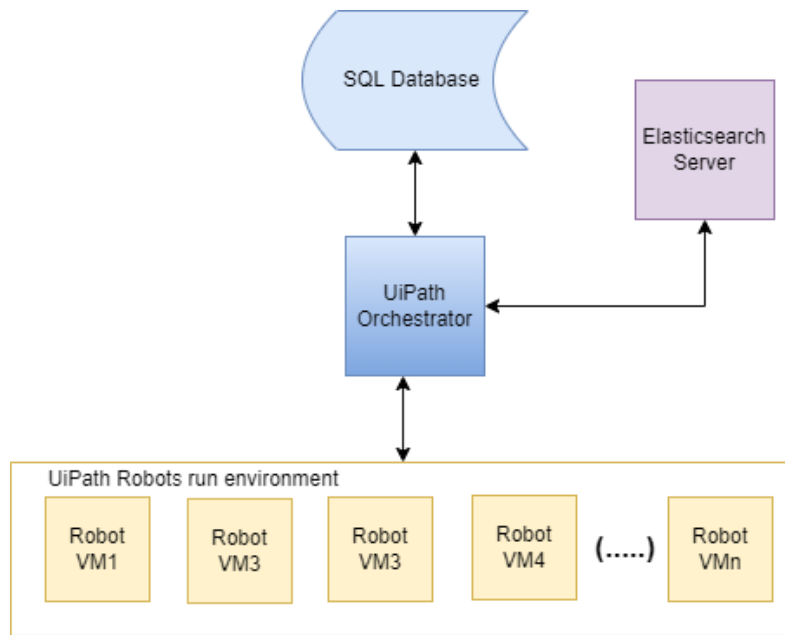


Figure 1.2 – UiPath single node architecture

Components list

- **UiPath Orchestrator:** It is recommended to install the Orchestrator application on a Windows server (physical or virtual). It will be a web application accessible from the **Internet Information Services (IIS)**.
- **SQL database (DB):** A SQL database is recommended to be set up in a separate SQL Server, or you can even leverage the cloud services provided by popular cloud service providers. This database will be the primary application database where the data can be queried later after installation. The connection must always be alive between the DB and UiPath Orchestrator.
- **UiPath Robots (digital workers):** This software is set up in virtual or physical machines that can be remotely accessed by Robot accounts. Multiple devices can be registered and connected with Orchestrator to execute the automation.

- **Elasticsearch (ES) and Kibana (optional):** Orchestrator logs can be directed to the SQL DB and ES. Hence, ES is an optional setup. ES and Kibana can be set up on the same Orchestrator Windows server or a separate server based on the need. ES is an open source application that can optimize UiPath Orchestrator logs access from Orchestrator. We can redirect the execution logs away from default DB tables and into the ES indexes by updating the information in `UiPath.Orchestrator.dll.config`. Kibana is an optional open source monitoring dashboard application that can also be installed on this server for monitoring needs.

Elastic Search and Kibana are part of Elastic Stack and you can learn more about Elastic Stack here: <https://www.elastic.co/elastic-stack/features>.

Note

UiPath Insights is the official business analytics and monitoring tool offering from UiPath. It provides complete monitoring and alerting capabilities for a UiPath program. This product can also be installed when we set up the Orchestrator nodes. If Insights is in use then ES is not needed to be setup.

Patty has gathered enough knowledge to set up a single node Orchestrator now.

Note

Please refer to the software requirements to install the supported version of the software from here: <https://docs.uipath.com/installation-and-upgrade/docs/orchestrator-software-requirements>.

Multi-node UiPath Orchestrator architecture

As the name suggests, multi-node will have multiple instances of Orchestrator installed. UiPath documentation can support large-scale UiPath Robot deployments with better performance and fault tolerance, as multiple Orchestrator nodes are available. If one node fails, other nodes will process the transactions and bots.

In addition, horizontal scalability is possible as we can add additional nodes to the Orchestrator as the Robot needs increase. Starting with two Orchestrator nodes that support up to 7,000 Robots, we can scale to 15 Orchestrator nodes, where we can have 150,000 Robots on the platform.

Patty and her team can start building a single Orchestrator instance. Then, as they harden things up for production, they can move ES off the single server, add additional DB server ES shards, and bring up a second node as an active/standby configuration. In other words, single to multi-node is more of a progression for any organization who are willing to mature their automation journey.

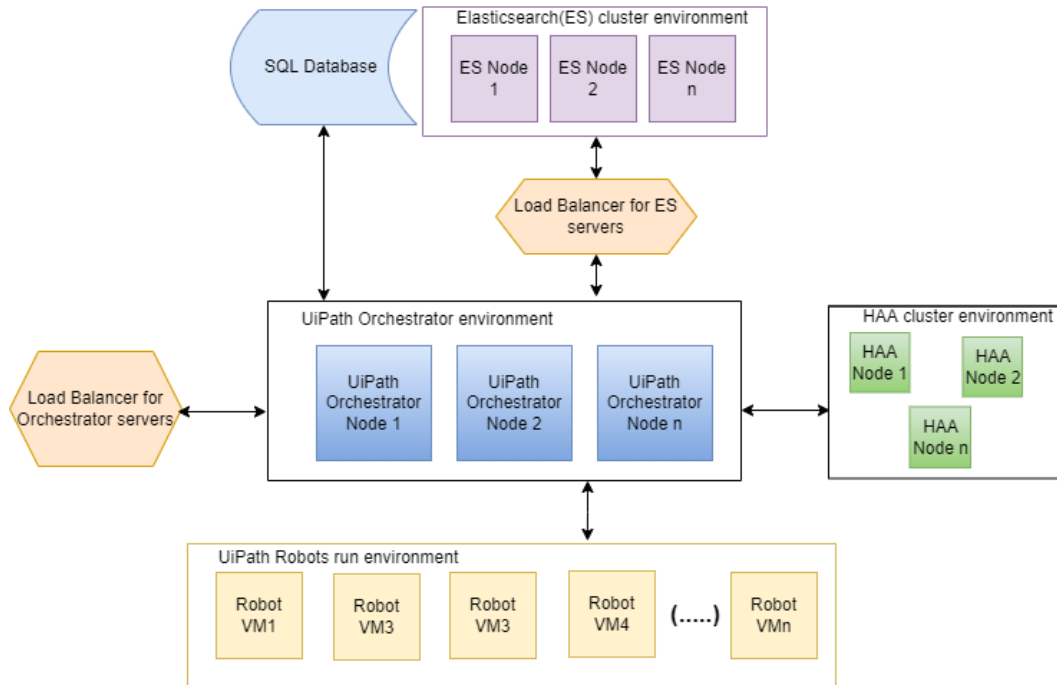


Figure 1.3 – UiPath multi-node architecture

Let's look at the components one by one:

- **UiPath Orchestrator scale set:** It is recommended to install the individual Orchestrator applications on a dedicated Windows server(s). All the Orchestrator nodes should have a similar IIS configuration, access rights, and security policies. It is recommended to have a service windows user account configured to this UiPath Orchestrator scale set. This can be used to set up the DB connection.
- **Scalable SQL DB:** The database server should be configured based on the number of Orchestrator nodes in production. Most cloud DBs are now scalable, and it would be good to consult the DB administrator to set up the best practices and proper access privileges.

DB maintenance activities need to be put in place from day one because the performance of Orchestrator is directly related to DB health.

- **UiPath Robots (digital workers) run environment:** These are set up in virtual or physical machines, which can be remotely accessed by Robot accounts. The only difference here is that this environment should be set up to scale faster to accommodate the new Robots quickly. A Kubernetes container is a popular option recently in this environment.
- **ES cluster environment:** It is recommended to set up a separate Windows server and host the ES and Kibana applications to build this cluster. The main idea is to have a scalable option when the Robots' logs start to increase with the Robots' jobs being executed. Logs are critical to maintaining UiPath operational health, and hence maintaining this cluster is crucial to support and monitor.
- **High Availability Add-on (HAA) cluster environment:** HA provides redundancy and stability for your multi-node Orchestrator deployment through failure resistance. In an HAA configuration, as long as multiple Orchestrator and HAA nodes are available, the other nodes will be activated if one fails; if any nodes fail or are taken down on purpose, processing will "failover" to the rest of the nodes in the cluster. Moreover, horizontal scalability is also possible that means you can add another node whenever your Robot counts needs to grow. UiPath HAA is just Redis **Original Equipment Manufacturer (OEM)** version for UiPath. Redis is used for in-memory data structure storing used for caching, which will improve the application's performance. The UiPath support contract supports only the multi-node setup with UiPath HAA. Hence, it is more than a nice-to-have component. Although there are other ways to set up multi-node failover and horizontal scaling, UiPath HAA is the only method supported by UiPath.

Please refer to the documentation for setup: <https://docs.uipath.com/installation-and-upgrade/docs/haa-installation>.

- **Load balancer (LB):** A load balancer is an appliance (which encompasses hardware load balancers, such as F5, as well as software load balancers) that will automatically distribute incoming web traffic across various endpoints. All the cloud service providers have a load balancer offering that can be leveraged, for example, F5 load balancing on AWS. UiPath Orchestrator supports Layer 4 load balancing only, not Layer 7, and no SSL offloading:
 - **LB for Orchestrator servers:** Load balancers are critical to redirect traffic to different Orchestrator nodes originating from various client requests. It is recommended to have a load balancer URL for Orchestrator login so if a node is down, we can still access the application.
 - **LB for ES servers:** The same concept applies to redirect connections from Orchestrator to ES servers. If there are multiple ES servers, then a load balancer is functional. There can also be scenarios where there are numerous ES shards on several different servers without a load balancer.

- **UiPath Identity server:** This provides a centralized authentication service to all UiPath products, and it is included in the on-prem Orchestrator installer. It is recommended that the enterprise **Identity and Access Management (IAM)** team configure the Identity server component to meet the existing enterprise standards. Most of the configurations are done on `appsettings.json` (please refer to <https://docs.uipath.com/installation-and-upgrade/docs/orchestrator-is-appsettings-json>).

The recommended hardware setting for scaling up the operation is mentioned here: <https://docs.uipath.com/installation-and-upgrade/docs/orchestrator-hardware-requirements>.

Note

NuGet packages are stored in a shared location which all the Orchestrator nodes can access. SQL Server will have redundancy and have the **Always-on Availability group**. If UiPath Insights is used, it is good to have an Insights application server added to this architecture.

Please refer to this link for hardware requirements based on the number of Robots in production: <https://docs.uipath.com/installation-and-upgrade/docs/orchestrator-hardware-requirements>.

I hope this was interesting. Patty and her team know to set up a multi-node Orchestrator in their organization. Now, let's look at the Orchestrator setup details in the next section.

Orchestrator setup

As introduced in the previous section, UiPath Orchestrator is the critical component used for the creation, monitoring, and deployment of Robots in the defined runtime environments.

There are three main options of UiPath Orchestrator setup possible in an enterprise:

1. The first one is a standalone installation which can be on-prem or cloud
2. The second one is the UiPath Automation Cloud
3. The third option is to set up the UiPath Automation Suite (starting November 2021)

It is important for UiPath Support and Administrator professionals to overview the UiPath Orchestrator setup. Let's start with this setup of a single node on-prem Orchestrator.

On-prem UiPath Orchestrator (single node)

Patty can now proceed with the installation detailed in this section. On a high level, the UiPath Orchestrator setup can be charted down into five different steps, as shown in the following diagram:



Figure 1.4 – UiPath single node setup steps

All the 5 steps are discussed in this section, let's start with the first step.

1. Set up a Windows Server on a machine or a cloud infrastructure. Ensure this web application server configuration (CPU, memory and security groups, and so on) is set up as per the recommendation from the UiPath documentation.

Refer to this link for more information: <https://docs.uipath.com/installation-and-upgrade/docs/orchestrator-hardware-requirements>.

2. Configure a SQL server and set up a DB. The next important step is to set up the DB server. Again, this can be set up on a separate machine or subscribed as a cloud service. Once the server is set up, ensure the DB is created and credentials noted.
3. Install and set up the prerequisites. The prerequisites (URL Rewrite, .NET, certificates, and so on) need to be installed on the application server before starting the UiPath Orchestrator setup. Please refer to the complete checklist from this link: <https://docs.uipath.com/installation-and-upgrade/docs/orchestrator-prerequisites-for-installation>.
4. Install UiPath Orchestrator; this is the primary step of running the UiPath Orchestrator installer and updating the mandatory information such as the DB information, Identity server, certificate, and so on). The installation will be completed successfully when all the prerequisites and access are correctly configured. Otherwise, the installer will roll back the installation.
5. Verify the UiPath Orchestrator installation and active license. The last step is to verify the installation by logging into Orchestrator and activating the license.

Note

The ES server setup mentioned in the architecture diagram is optional, and we will cover it in a different chapter. Please refer to *Appendix A* at the end of the book to get a step-by-step walkthrough of this setup.

Next, let's look at some of the essential considerations for a multi-node setup for Patty and her team.

Multi-node on-prem Orchestrator considerations

Based on the customer's need, a multi-node setup can be made. In the case of Patty and her team, once a single node orchestrator was set up, they decided to add more nodes and grow into a multi-node resilient platform. Multi-node setup is the most common Orchestrator setup in any organization that is still using an on-prem environment. Two or three-node setup is typical for mid to large-scale Robot setups. There are a few considerations to set up this environment:

- The Orchestrator setup needs to replicate to Orchestrator cluster setup. Multiple servers will act as nodes for UiPath Orchestrator. Single node setup can be scaled to multi-node with appropriate infrastructure if needed.
- Similarly, the ES setup needs to be scaled up to the ES cluster environment. Multiple ES servers may need to be set up to support the scaling needs of Robots.
- SQL servers need to be configured to be scalable and available. It is recommended to have a redundancy setup in effect for the servers.
- In-memory HAA clusters need to be configured in the multi-node setup to maintain the platform's performance.

- Load balancers need to be configured to redirect traffic to Orchestrator, and an additional load balancer is recommended if there is an ES cluster setup.
- The Robots farm needs to be set up to scale up operations on demand.

The entire setup can be made on physical machines or cloud infrastructure based on companies' IT policies. After following the steps, Patty and her team set up multi-node Orchestrator.

Cloud – UiPath Orchestrator

In recent years, a cloud UiPath Orchestrator has been gaining traction, and as many enterprises move toward cloud-first architecture, let's cover this aspect.

UiPath will maintain the complete infrastructure for Orchestrator, and all the resources will be held in the cloud. As UiPath introduces new products regularly, it becomes easy for customers to quickly subscribe to them and not install them as separate applications in on-prem environments.

Let's consider ABC Insurance Corporation, an insurance service provider and a UiPath Enterprise customer who is setting up the cloud UiPath platform. Candace and her team at ABC Insurance Corporation will work on this assignment.

Automation Cloud setup

Now it's Candace's turn to set up the cloud Orchestrator for her team. The UiPath account team will provide the initial cloud administrator for UiPath Automation Cloud. She can add new users from the admin and access the Orchestrator default tenant, and the rest of the setup can be made once we log in to cloud Orchestrator.

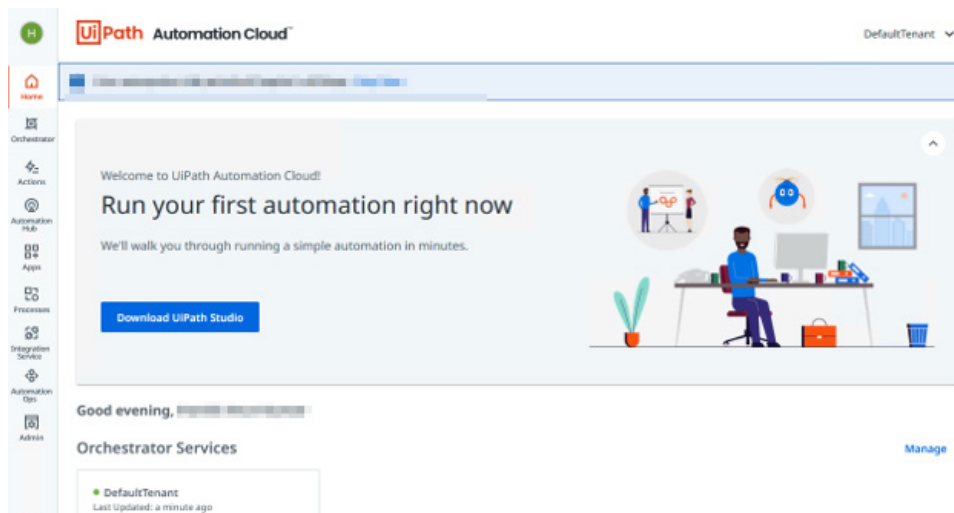


Figure 1.5 – UiPath Automation Cloud

This is the primary offering for all new customers. Many existing customers are interested in converting to Cloud as it reduces the overhead of maintaining the Orchestrator platform and easy scalability.

Even though there are plenty of advantages, there are also a few disadvantages, such as not having access to the DB. Custom monitoring with Kibana or any internal platform is also not possible in this setup.

Candace and her team could access the UiPath cloud Orchestrator from the Automation Cloud within minutes. Next, she and her team need to understand Robots and options provided by UiPath before setting up.

Robots setup

As per UiPath documentation, UiPath Robots is the client program that runs processes developed in UiPath Studio or Studio Pro. Robots information is stored and controlled by UiPath Orchestrator.

There are different ways Robots can be configured in an Enterprise setup based on business needs. Before we can configure Robots, we need virtual machines to host these Robots' client software.

Before moving forward, let's understand folder concepts, as Robot types are tied to folder concepts.

UiPath folders

Orchestrator resources, such as processes, jobs, and assets, can be logically grouped into folders. Folder concepts provide an excellent way to organize the entire UiPath program. There are two types of folders currently supported by UiPath:

- **Modern folder:** This is the recommended option that will enable the UiPath program to utilize better all the latest features available in the UiPath Platform. Users and roles are assigned at the folder level, and greater flexibility to move Robot licenses between folders. Automation workflow executed in a folder can access resources in another folder, too.
- **Classic folder:** User roles are assigned at the tenant level, and Robots can be only part of a single folder. If the Robot needs to be used in a different classic folder, then the Robot needs to be deleted and recreated in the new folder.

Note

Classic folders will not be supported from UiPath Orchestrator v2022; all the existing customers need to migrate to modern folders to upgrade to this version.

UiPath Robot machine setup

Machines are runtime environments (physical or virtual) where the Robots execute the process. There are four options of machines in the latest version of Orchestrator (v2022.10):

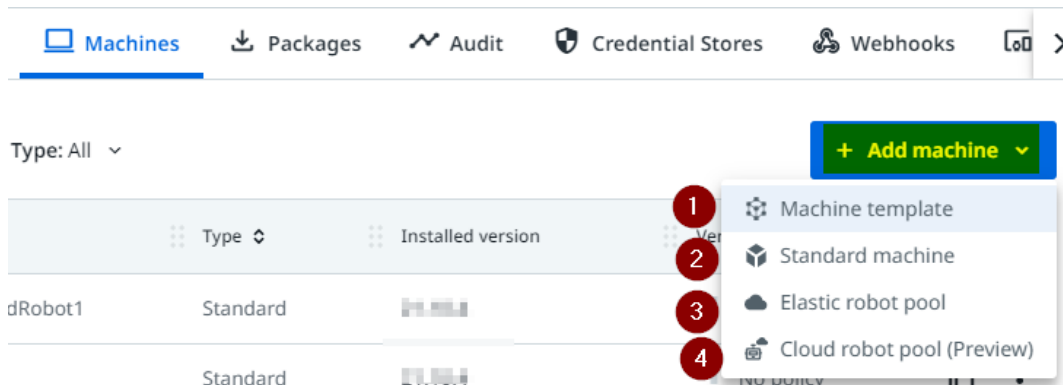


Figure 1.6 – Machine options

Note

The software needed to run the automation (including UiPath Assistant) must be installed once the machine is set up. Access to resources, such as network shared location, needs to be provided.

All the Robot machine options are discussed in this section ,lets start to understand the first option.

- **Machine template:** Robot machines can be logically grouped as per an organization's functional or IT needs. Let's assume 10 AWS EC2 virtual machines are available to execute HR and supply chain processes. We can build two different templates for HR and the supply chain. The HR template might have Version 10 of App A and App B installed, and the supply chain process needs Version 12 of App A and App C. Best practice here is to create a "Gold Image" with all the applications you will need for your automation. Failing that, you need to plan out and create machine templates for the combinations of applications you need. Adding a machine template key to the Robots will break the machine mapping.

Note

As the Robot and machine mapping is broken, any program can be executed in any Robot tied to the machine template.

To build a template, fill in the template name and runtime licenses needed, for example, a **Production (Unattended)** 5 runtime license. Additional options are to mention whether the machines are Windows or Linux-based once under the process compatibility settings:

Machines / Machine template

General details

Allows floating robots to run on any machine linked with the generated access key.

Template name *

Description

Client binaries (Robot, Studio, Assistant etc) auto-update policy *

None

The auto-update policy applies to client binaries (Robot, Assistant, Studio, etc) versions. Version downgrades are not supported. [Learn more](#) about the update process or see [release notes](#).

Runtime license (execution slots)

Production (Unattended) *

0

NonProduction *

0

Testing *

0

Supported usage

Process type *

All

Process Compatibility *

All

Windows only

Cross-platform only

Cancel Provision

Figure 1.7 – Machine template options

In addition to this, we can also customize it to run foreground or background jobs.

- **Standard machine:** This is the default option available, and runtime licenses are needed. Most of the classic folder Robots are running on standard machines, and it is one of the most popular machine types used currently in organizations. The standard machine is supported in the modern folders concept, too. This will be a recommended type with a robust bot machine mapping, for example, a dedicated bot running on email processing on a machine.

Machines / Standard machine

Backwards compatibility configuration for classic folders setups, with limited functionality. For modern folder setups, we recommend using [Machine Templates](#) instead.

General details	Runtime license (execution slots)
Name * <input type="text"/> <small>Has to be an exact match to the name of the workstation the Robot is installed on.</small>	Production (Unattended) * <input type="text" value="0"/>
Description <input type="text"/>	NonProduction * <input type="text" value="0"/>
Client binaries (Robot, Studio, Assistant etc) auto-update policy * <input type="text" value="None"/>	Testing * <input type="text" value="0"/>

The auto-update policy applies to client binaries (Robot, Assistant, Studio, etc) versions. Version downgrades are not supported. [Learn more](#) about the update process or see [release notes](#).

Cancel Provision

Figure 1.8 – Standard machine options

- Primary Robot Pool:** This is one of the recent machine types added at the start of 2021. This is a scalable auto option provided by popular cloud vendors, which will allow to automatically add new Robot machines in the cloud as per the configured parameter such as cost, maximum number, and time.

We need to add a Cloud provider connection before using this feature.

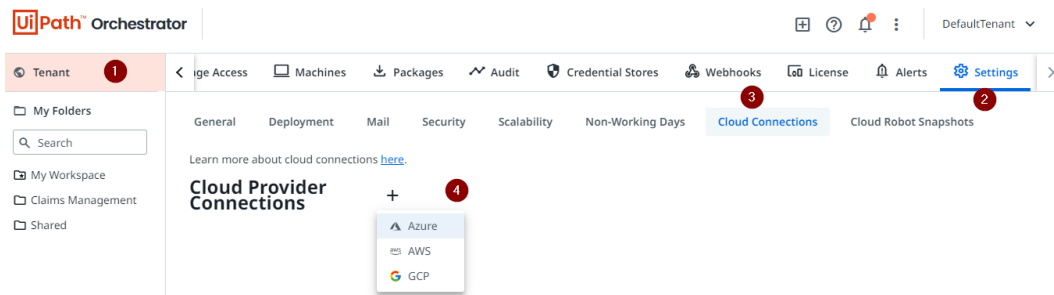


Figure 1.9 – Cloud machine options

We just need to provide a name and associate the cloud connection to enable these auto-scalable machines:

Machines / Add Elastic Robot Pool

Name *

Description

Machine Details

Cloud Connection

Cancel Provision

Figure 1.10 – Robot pool

Note

A Robot pool can be a way to reduce your license and infrastructure costs significantly. If your organization has a mature cloud practice or is trying to start one, it is highly recommended to consider using a Robot pool as part of your architecture plan. Please refer to this documentation to set up the cloud connection: <https://docs.uipath.com/orchestrator/v0/docs/elastic-Robot-orchestration>.

- **Cloud Robot Pool (Preview):** The last one is the latest feature in preview mode as of Jan 2022. This is a pure-play attempt by UiPath to enable Robots as a service offering. The Robot VMs will also be managed by the UiPath cloud platform and can be provisioned at will if this option is enabled. Reduction in risk and cost of maintenance are significant benefits in this approach, but lack of control, customization options, and Azure cloud dependency are a few drawbacks to be highlighted.

A new license type called **Automation Cloud Robots** is needed to leverage this feature. The user needs to enter a name and choose a machine image to set up a VM on UiPath Cloud. Once the virtual machine is enabled, the remote desktop needs to be enabled, and when we log in, UiPath Studio and Assistant are already preconfigured:

☐ Machines / Add Cloud Robot Pool (Preview)

Name *

Description

Machine Image *

☒ Standard ☐ Customized

UiPath + Supported Browsers on Windows Server 2019 ▾

Advanced Settings ^

IP Address

☒ Dynamic ☐ Static

Execution slots per machine *

Cancel Provision

Figure 1.11 – UiPath Cloud Robot Pool

Note

I predict that this will be the most popular machine of choice in a few years as UiPath will handle the VM maintenance, and clients have one less maintenance function. Please refer to the official documentation here: <https://docs.uipath.com/orchestrator/v0/docs/automation-cloud-Robots>.

UiPath Robot categories

Now that we have seen different machine options, let's look at some Robot categories.

Supervised versus autonomous Robot

The most common differentiation of Robots is based on the human-in loop concept, where the Robot executes a job with or without assistance from humans.

Supervised Robots run under human supervision and need humans to interact to complete a transaction.:

- **Attended** – Bots operate along with the user and can be invoked via user events
- **RPA developer bots** – This is used to connect your Studio/StudioX or StudioPro to UiPath Orchestrator, and it is used when developing the UiPath scripts

On the other hand, autonomous Robots do not require human supervision to execute jobs:

- **Unattended** – Unattended bots are executed as batch jobs in the background, typically in a virtual environment, and are used to automate processes designed by the developers. It is the most popular bot license type being used in many organizations.
- **Non-production/testing bots** – Works in unattended/test mode for development purposes only.

Note

Choosing the right mix of Robot licenses is also an essential consideration for the UiPath program to succeed.

Standard versus floating Robot license

A standard Robot is configured in a dedicated virtual or physical environment, that is, there is a tight bot-to-machine mapping, and the same bot cannot be used in a different machine at the same time. The standard machine is the only machine type that works with a standard Robot. Standard Robots are the current standard in most organizations.

The floating Robots concept allows the Robot to be in multiple virtual or physical environments. The Robot is not associated with any machine. If a machine template is used and the Robot is configured to run on it with multiple machines, it becomes a floating Robot.

Note

Once the classic folder concept is depreciated in UiPath Orchestrator v2022, all the Robot licenses will be floating in nature by default on modern folders.

Normal versus high-density Robots

The subsequent differentiation of Robots is based on the hosting machine. If the Robot is hosted in individual VMs or desktops, it is considered a normal Robot.

If multiple Robot IDs are configured to run on a server environment, for example, Windows Server 2019 with five Robots accounts, it is a **high-density Robot**. The main advantage is that we can run multiple automation simultaneously in separate user sessions on the same machine. For this, some configurations must be done on the Windows Server machine first; then, you need to set up the environment in Orchestrator. Please refer to the following documentation: <https://docs.uipath.com/installation-and-upgrade/docs/setting-up-windows-server-for-high-density-Robots>.

There are a few advantages and disadvantages of high-density Robots. One of the main advantages is to reduce the Robot machine maintenance downtime and easy monitoring of machine health. Still, the major drawback is if the server goes down with an issue, then multiple bot accounts will be stopped. One way to mitigate this risk is to have a machine template and provide additional runtime licenses to the machines in the template.

There are a few more types of Robots worth mentioning:

- **Test Robot:** If the UiPath Test Suite is enabled, we can procure this test non-production license type with a testing runtime associated. Converts a non-production Robot to a testing Robot.
- **AI Robot:** This is a particular type of license for the AI Center. An AI Robot license is used for **machine learning (ML)** training jobs, and a single license can also be used for executing any two ML skills available in the AI Center.

This section has covered all the major Robot types, and next, let's see how to set up a Robot. Candace and the team are ready to set up unattended Robots using the machine template option in the ABC Insurance Corporation's cloud Orchestrator.

UiPath Robot setup

On a very high level, both attended and unattended Robots can be set up on any of the four supported machine types, and they follow a structured flow as depicted in the following diagram:

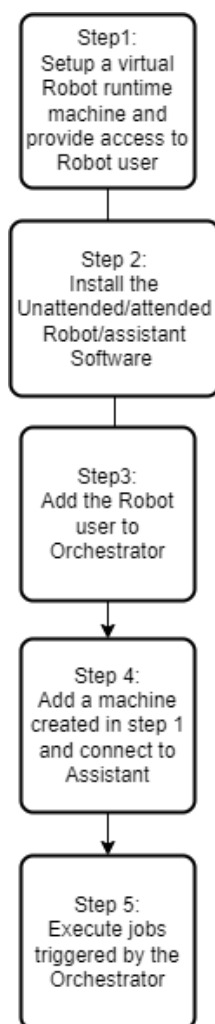


Figure 1.12 – Robot setup steps

Details of all the steps depicted in the diagram are detailed in this section, let's get into the details of the first step.

1. Set up a VM and install all the required software for executing the process as needed, such as an Office suite, SAP client, and so on. We must add the root user to the right user group and domain that has access to this machine. Usually, the organization's infrastructure group will be responsible for setting up the machines as needed, as per the policy agreed with the Enterprise architecture group.
2. Install the UiPath Assistant software and choose whether you wish to run attended or unattended automation on this machine. The Enterprise support team will be able to install the software with elevated privileges.
3. Next, we need to add a Robot user in Orchestrator with runtime licenses configured and provide correct access privileges to the folders and their resources.
4. Establishing the connection between Robots and Orchestrator is mandatory, and again, the Enterprise support team needs to get involved as admin privileges are required to update the Orchestrator settings.
5. Once the connection is established, trigger a test job, and see the execution in action.

Note

Please refer to *Appendix B* at the end of this book for step-by-step instructions to connect an unattended Robot to Orchestrator. In the real world, once a new Robot is set up on a machine, prerequisite checks are needed, such as whether the Robot has all the access rights enabled, software access, and licenses provided, such as Office 365 Acrobat, and so on. A trial job run is highly recommended in an attended mode before the actual runs take effect.

This completes the information on Robots. Candace and the team have followed the previous steps to set up the Robots.

Now that we've set up the UiPath environment, let's talk about organizing the UiPath program.

Program organization

On a high level, there are different ways to organize the UiPath RPA program to build for scale. Various organizations deploy different strategies based on agreed policies.

In this section, let's talk about two strategies to define a UiPath program organization. Program sponsors and architects usually characterize them, but UiPath support and Administrator teams should also be consulted when designing the structure. The UiPath Administration team will be the responsible party for the actual set of this model in UiPath environments.

We will use the ABC Insurance Corporations UiPath RPA program as a reference to explain the concept better.

Organizing tenants and folders

When we install UiPath Orchestrator to an on-prem or set up on the cloud, we will get a **Default** tenant visible when we log in for the first time. Every organization starts with one tenant for the proof of concept and pilot. Once they get requirements, they mature and organize the UiPath resources in a multi-tenant environment.

It is essential to understand the meaning of the terms and their significance:

- **Tenant:** This is the highest level of grouping UiPath Orchestrator resources such as Robots, processes, machines, and so on. It is always recommended to have multiple tenants, usually at the organization function level such as finance and HR. You can log in to Orchestrator and learn more by viewing how the context changes:

The screenshot shows the 'Add Tenant' dialog box. It has a title bar with the text 'Add Tenant' and a close button. Below the title bar, there are two input fields: 'Tenant Name' and 'Tenant Region'. The 'Tenant Name' field contains the text 'Finance'. The 'Tenant Region' field is a dropdown menu showing 'United States'. Below these fields is a section titled 'Provision Services' with a list of services and checkboxes. The services listed are: Orchestrator (checked), Actions (checked), Processes (checked), Test Manager (unchecked), Insights (unchecked), Integration Service (checked), Data Service (unchecked), AI Center (unchecked), Document Understanding (checked), and Automation Hub (checked). At the bottom of the dialog are two buttons: 'Cancel' and 'Save'.

Figure 1.13 – Tenant setup

- **Folders:** This is the next level of grouping resources such as processes, assets, and so on, and it is usually at the business process level. The Robot's and user's rights can be defined at the individual folder level or inherited from the Level 1 folder. IT service management is a good example of a folder that will contain processes such as incident management, problem management, and so on, that is, UiPath Service Now integrations jobs. Modern folders are used in ABC Insurance Corporations's UiPath setup.

Note

We can add a subfolder as well to have more control at the subprocess level.

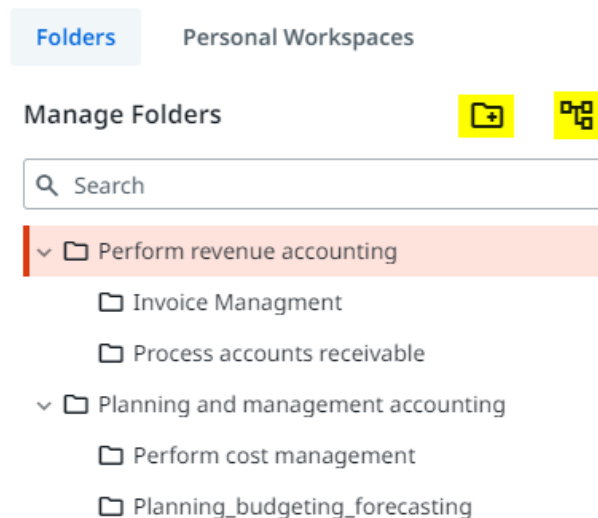


Figure 1.14 – Folders setup

Note

We have one more segregation called **Environment** supported in classic folder setups only, and it is very similar to the subfolder concept. We can group Robots and processes to dedicate them to a partial environment.

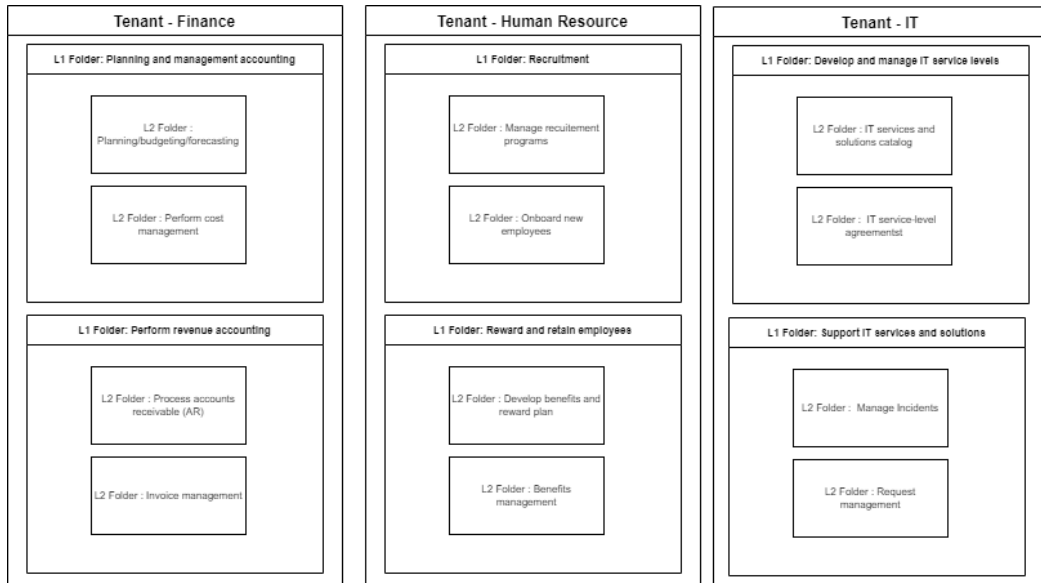


Figure 1.15 – ABC Insurance tenant/folder structure example

This was how the ABC Insurance UiPath program was organized, and there are many other approaches to manage the Robots, jobs, and other assets. The principal rule in many firms is that the team that pays for the license decides the strategy.

Organizing the RPA environments

The next level of organization is to replicate the above tenant/folder (along with all the resources) structure in different environments, namely test, **user acceptance testing (UAT)**, and production.

The ABC Insurance Corporation's UiPath program setup is depicted in the following diagram; let's discuss the concept in detail:

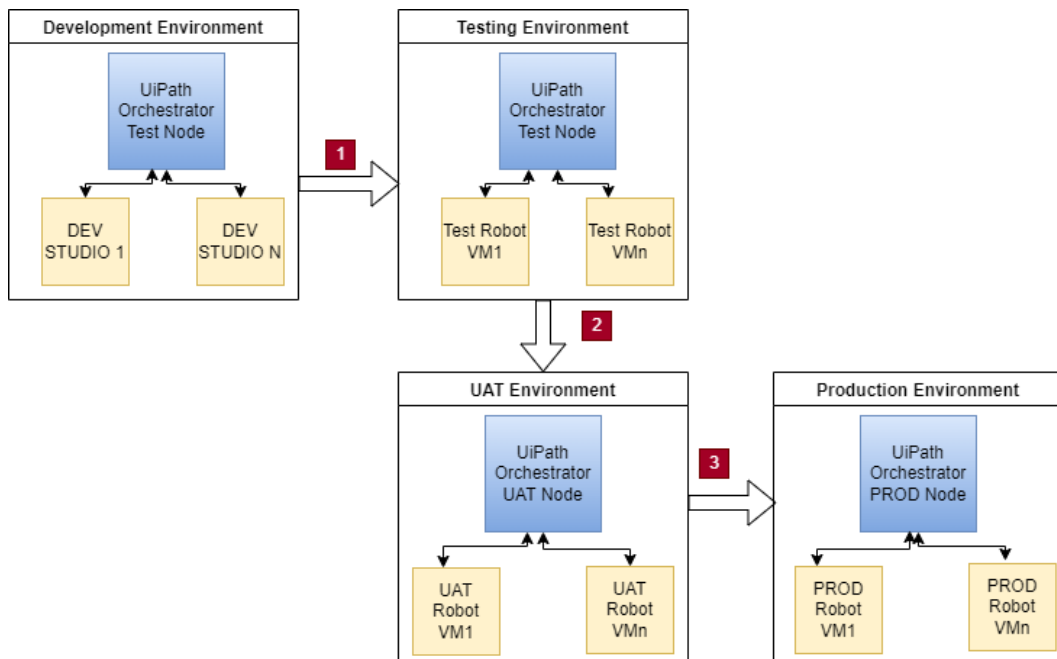


Figure 1.16 – ABC Insurance Corporation's UiPath environments

The details of the four environments depicted in the diagram are explained in the following sections, lets start with the development environment.

- **Development environment:** This is the primary environment where ABC Insurance Corporation's UiPath developers use Studio/Studio Pro or Studio X to create the automation. Ideally, individual developers will have separate licenses allocated to them. The Studios are usually connected to the UiPath test Orchestrator node. The developers also perform unit testing in this environment; then, the packages can be published in the test Orchestrator once the connection is established.
- **Testing environment:** This is the primary environment where ABC Insurance Corporations UiPath testers perform their system and integration tests. The UiPath test Orchestrator (non-production license type) and non-production Robot are usually configured in this environment. The tenant/folder setup and access rights for different UiPath resources need to sync in the test/UAT and production environment. The jobs that are created from the package from the Studio can be trigged in the test environment. Multiple test bots can be configured to perform the test as per the business need. The testers will usually use this environment to run functional, integration, performance, and security tests.

The test environment access for the target application(s), such as Salesforce and Mainframe, needs to be provided to the bots in this UiPath environment to perform the tests.

Note

If you use the UiPath Test Automation Suite, then testing Robot licenses are needed to execute the test automation.

- **UAT environment:** This is the primary environment where ABC Insurance Corporation's UiPath business users and user acceptance testers perform UAT. Once the package passes the test gateway, the same package can be uploaded to the UAT environment. This environment again has a non-production license Orchestrator and non-production Robots configured. The setup should ideally reflect the production environment as the test will be equivalent to dry runs in production.

The UAT environment access for the target application(s) that are automated such as SAP, Salesforce, and Mainframe, needs to be provided to the bots in this UiPath environment to perform the UAT tests.

This environment will be used by the business analysts/product owner, who will provide the UAT signoff. Once the UAT signoff is provided, the package will be promoted to the production orchestrator.

- **Production environment:** This is the primary environment where ABC Insurance Corporation's UiPath business validators provide final validation and where the bots will execute the business transactions. This environment is the final runtime environment for the bots. This will have a production Orchestrator setup (single/multi-node), and production licensed Robots (unattended/attended) on VMs or servers. The automation packages that business stakeholders signed off on will be run in this environment.

The production environment access for the target application (automated, such as SAP, Salesforce, and Mainframe) needs to be provided to the bots executing the process in this UiPath production environment to perform the actual business transactions and processes.

Note

Based on the maturity of the UiPath programs, one or more of the environments may be missing in your current setup.

If the production is a multi-node Orchestrator setup, it is recommended that both test and UAT are multi-node setups. Risk reduction during Orchestrator upgrade is the primary reason behind this recommendation. The test and UAT UiPath Orchestrator nodes will be upgraded to a newer version before the production orchestrator is upgraded. The issues can be identified upfront and will improve the success rate of the production upgrade.

We mentioned the manual movement of the UiPath NuGet packages between environments by the ABC Insurance Corporation's RPA team. It doesn't always have to start that way, and the whole deployment process can be automated from the start of the program. Automated package movement through a DevOps pipeline has an extensive list of benefits to the UiPath RPA program. We will look at this UiPath DevOps concept in a later chapter.

Summary

UiPath RPA Platform is a robust process automation platform with many products in its portfolio. I hope you got a quick overview of the core UiPath products currently shipped as part of UiPath Platform and on-prem and cloud offerings. We used two personas from XYZ Bank and ABC Insurance Corporation in this chapter to explain the concepts so that you can relate to the idea of real-life personas.

In addition to this, you would have understood the on-prem and cloud architecture of UiPath Orchestrator. This is one of the most important concepts for any UiPath support and administrator role.

Next came the different types of Robot types and licenses offered. Having complete knowledge of Robots will give you an edge as a UiPath support professional, to provide valuable inputs during the UiPath program execution.

Finally, you learned how a typical RPA organization would be set up in a UiPath program. This is just the tip of the iceberg, and there is plenty of information about setup and configurations in UiPath documents; please deep-dive and learn as and when the need arises.

Now that you have a basic understanding of UiPath Platform, it's time to get moving to the next topic. Let's learn about the UiPath support strategy, framework, and models in the next chapter.

