



Automation Scheduling

Agenda

- Platform considerations
- On-premise vs Cloud-based environments
- Understanding triggers
- Schedule prioritization
- Scheduling considerations

Today's goals:

- Providing a framework for maximizing licensing costs and reducing errors with your selected automation platform

Platform considerations

What does your platform do for you?



On-premise vs Cloud-based

On-premise is a locally hosted version, running on your own infrastructure. Cloud-based is stored and managed through the web on the provider's servers.



Build Environment

Platforms either provide a script-based or visual-based (more popular) build environment. They are usually identical between Cloud-based and On-premise versions.



Scheduling

Tools leverage an Orchestrator or Control Room - which differ slightly between Cloud-based and On-premise.



Reporting

Most tools provide Reporting capabilities at an extra expense and/or additional service line. On-premise allows for more customized reporting power, where cloud is a standard offering.

On-premise vs. Cloud-based

Why would you choose one over the other?

The main distinction between Cloud-based and On-premise is where the infrastructure is hosted and managed. Cloud-based has additional benefits for scheduling and overall platform management, but there is a drawback on reporting and general “control”.

Management

Rationale:

Cloud-based has the ability to manage teams by quickly assigning/removing tenants from UserID's.

Cost Savings

Rationale:

Cloud-based is more desirable with it's acquired cost savings of having the platform host and manage the environment.

Efficiency improvements

Rationale:

With better scheduling methods available on the Cloud version, there can be efficiency improvements on automations compared to On-premise.

On-premise vs. Cloud-based, continued

Category	Cloud-based	On-premise
New feature availability		
Build capability		
Scheduling		
Reporting		
Product management		
Security		
Scalability		
Control		

Discussion:

What's most important for your organization when considering the connection type of your platform?

For example - is it important to keep automation records for longer than 120 days?

What is a “trigger”?

When scheduling an automation, how do you tell it “go”?

Trigger Types:

1. **Event Based** - an event can be defined as a report being made available, a system refresh occurring, an automation finishing and prompting another automaton to start, etc.
2. **Time Based** - run time is set in motion by a defined time or range, usually depending on system/data availability based on operational needs.
3. **Queue Based** - when a queue is generated by an SQL/export it can prompt an automation to run by indicating that the data is available.

Discussion:

Are there any different triggers leveraged by organizations today?

How to prioritize your schedule

Which automation runs first? How do I determine the sequence?

ROI & Operational Impact

- Automation has significant ROI and/or is cash generating
- Automation has direct impact on day-to-day operations and needs to be run before shift-start
- Automation has downstream statute/regulatory type requirements to meet

Data & Resource Availability

- PAS/EHR and/or internal system refresh times dictate when data is available
- PAS/EHR/External portals has designated refresh times when application is unreachable
- Server downtimes dedicated to maintenance, or re-run exceptions
- Department coordination if platform is shared

Discussion: How are your automations prioritized in the schedule?

Scheduling considerations

Once a “Trigger” and priority has been identified, are there any additional considerations?

Method	Description
Multi-Threading	Multi-threading is used to resolve large queues by assigning > 1 bot to 1 queue. For example, if there are 1000 total accounts, you could assign 500 accounts across 2 bots. <u>Cloud-based</u>: Use of available bot resources to assign multiple bots to one queue. <u>On-premise</u>: Splitting queues by Nth rows assignment to automations manually.
Credentialing conflicts	When running multiple automations for one organization/application/etc., credentialing conflicts can occur. For example, if your organization has 1 set of credentials, but 3 bots in the EHR, the bots can only 1 can run at a time.
Server congestion	Applications and website server performance can negatively impact the automations productivity when running automations during high-traffic hours. There may be a need to adjust the scheduled run-time.
Start-up time	Automations require time to “start-up” when initiated. This involves launching applications, resolving MFA, and logging into the system. You will need to allow for start-up time when scheduling automations. Each automation will have varying start-up times depending on the systems they are accessing.

Tell us how we did



1 - Least valuable

10 - Most valuable

We value your feedback!

Ending Conversation Topic:

Would it be helpful to go into these details in more depth? How to Multi-thread? How do I sequence automations by a Queue-based trigger?

Additional Resources from UiPath on Scheduling:

<https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/about-jobs>

<https://docs.uipath.com/orchestrator/automation-cloud/latest/user-guide/managing-triggers>



TARPON
H E A L T H

Benjamin Reigle | benjamin@tarpon.health

James Stanford | james@tarpon.health