Robot Auto Healing powered by UiPath IT Automation

Documentation: Deployment / Configuration / User-Guide

Robots Auto-Healing powered by UiPath IT Automation

Andrei Oros

Release Version: 1.5

Publish Date: 2021.10

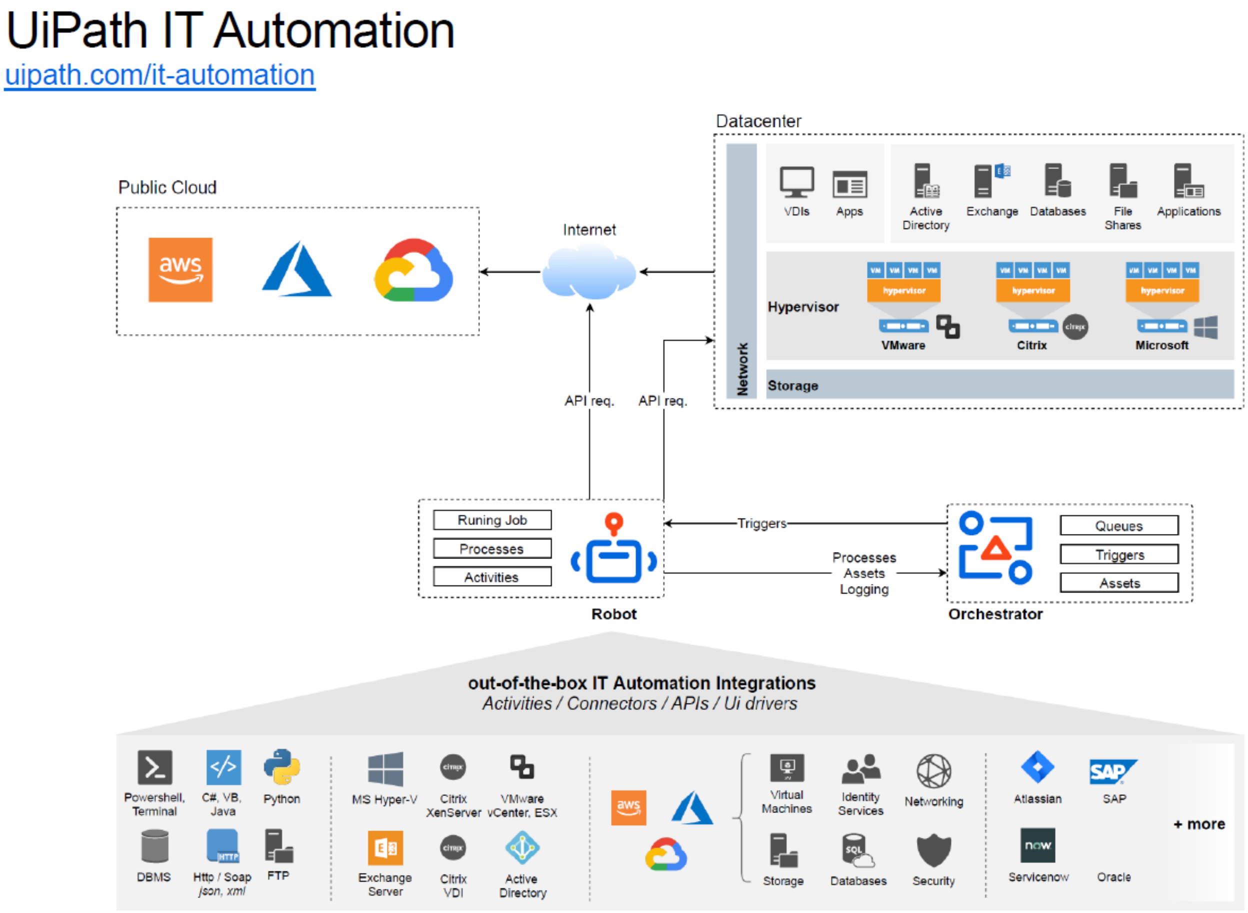
Author: andrei.oros@uipath.com

Table of Contents

Introduction

UiPath Robots Auto Healing (RAH) is a workflow based solution that helps optimize RPA Deployments by minimizing robot downtimes.

The RAH solution is implemented with the help of the official UiPath IT Automation activities - they empower us to automate use cases from all areas of our IT Ecosystem, in both on-premises datacenters and public clouds:



What is UiPath IT Automation?

UiPath's platform has builtin capabilities that enable us to automate IT processes. The official IT Automation activities are background running, implemented on top of the official SDKs from vendors such as Microsoft, Amazon, VMware, Citrix and more.

The UiPath IT Automation activities can be included in your projects at no cost from UiPath Studio (any edition) > Manage Packages.

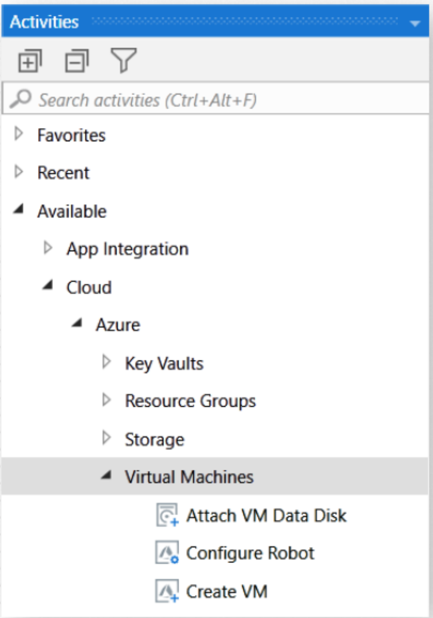
Official Activities:
Azure, Amazon Web Services (AWS), Active Directory, Azure Active Directory, VMware vCenter ESXi, Citrix Hypervisor, Hyper-V, Exchange Server, Microsoft System Center
.. and more.

Server Virtualization

Cloud Infrastructure & Services

User Management

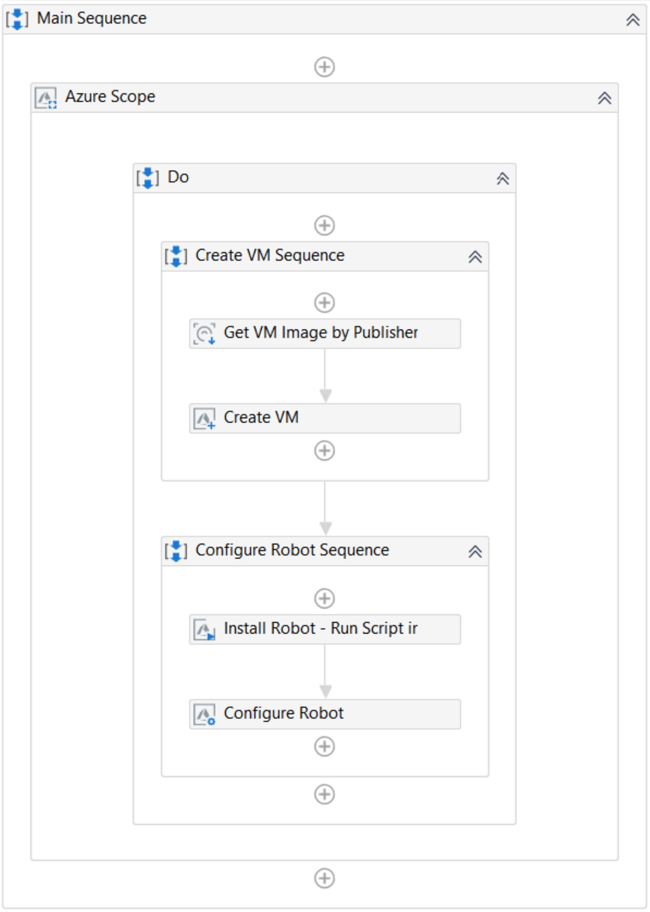
Networking & Security



Activities

Search activities (Ctrl+Alt+F)

- Favorites
- Recent
- Available
  - App Integration
  - Cloud
    - Azure
      - Key Vaults
      - Resource Groups
      - Storage
      - Virtual Machines
        - Attach VM Data Disk
        - Configure Robot
        - Create VM



Main Sequence
- Azure Scope
  - Do
    - Create VM Sequence
      - Get VM Image by Publisher
      - Create VM
    - Configure Robot Sequence
      - Install Robot - Run Script ir
      - Configure Robot

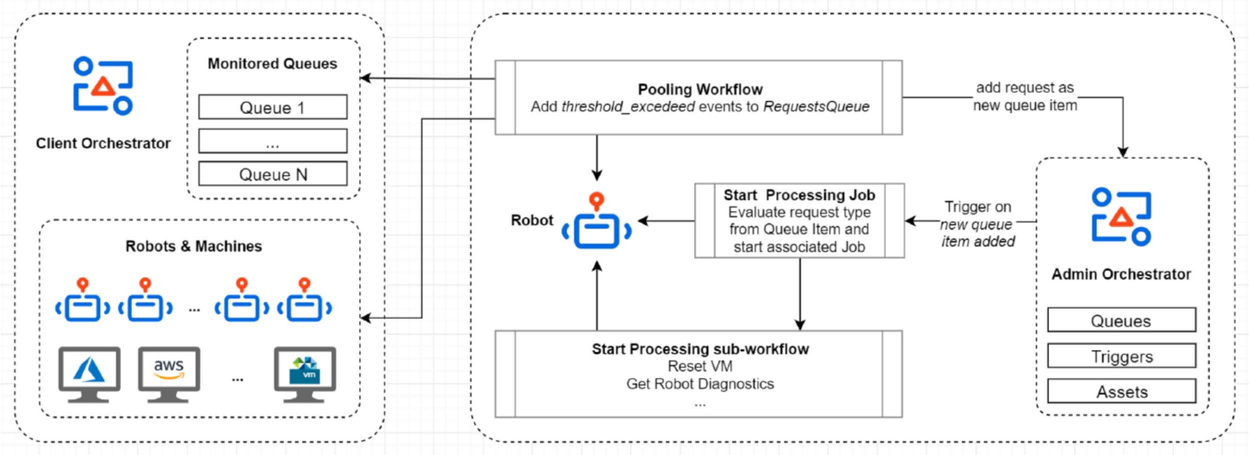
500+ IT activities

easy to use with drag & drop
robust background execution via APIs
secure Veracode certified
supported by UiPath

Simple. Elegant

powerful workflow automation for IT Tasks

RAH Components Overview


Client Orchestrator
Monitored Queues
Queue 1
...
Queue N
Robots & Machines
...
aws
...
vm
Pooling Workflow
Add threshold_excedeed events to RequestsQueue
add request as new queue item
Robot
Start Processing Job
Evaluate request type from Queue Item and start associated Job
Trigger on new queue item added
Admin Orchestrator
Queues
Triggers
Assets
Start Processing sub-workflow
Reset VM
Get Robot Diagnostics
...

Management Orchestrator
Most of the RAH solution items are to be configured in an Orchestrator tenant, in a dedicated folder: *Processes* (eg RSH_RobotsServiceCheck, ..), *Assets* (DB connection, infrastructure access), *Triggers*. The RAH solution can be deployed in the same Orchestrator tenant as one of the managed clients.

Client Tenant
RAH communicates with the monitored clients via the Orchestrator REST API: in the solution's assets you define the monitored Orchestrator tenant, folders and the API auth info.

Database
The SQL Server database where the monitored robot machines data is stored (performance counters, app versions, OS updates). The connection to the DB is done with the *UiPath.Database* official activities – the connection string is retrieved from the *RSH_DbConnectionString* text Orchestrator asset (from the **Management Orchestrator**).

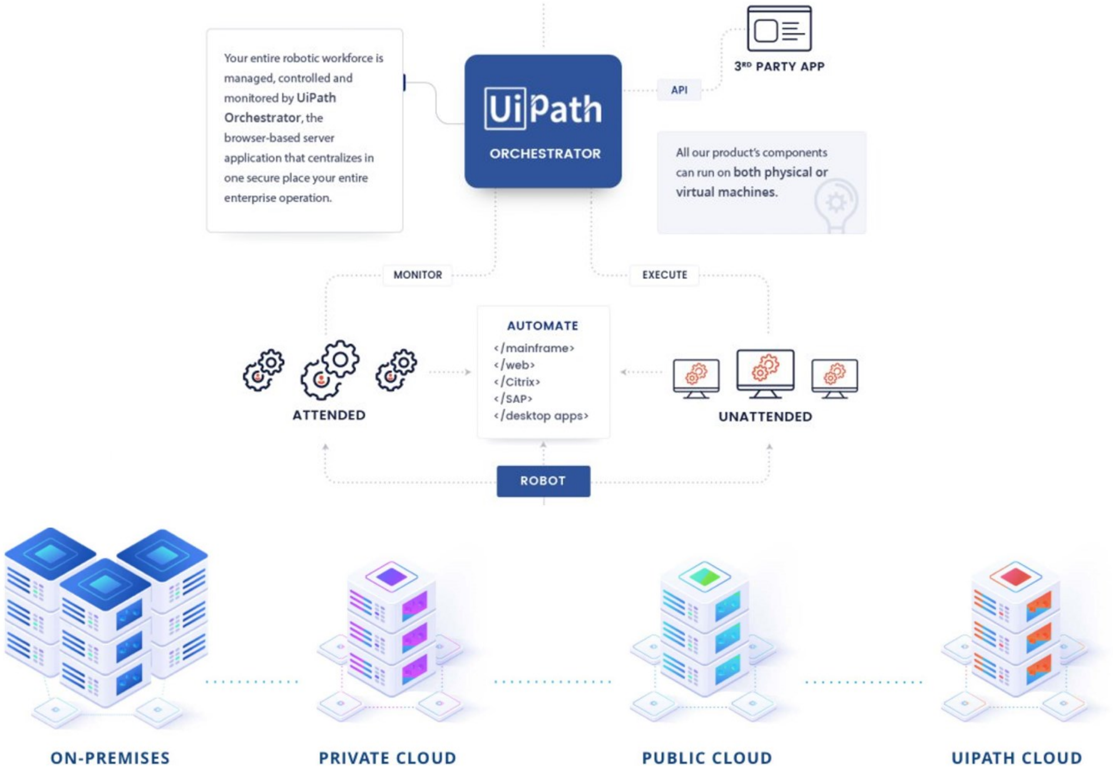
Processes

- RSH_RobotServiceCheck
- RSH_QueueItemInProgressLimit
- RSH_JobFailed (for chrome extension install)
- RSH_GetMachineData
- RSH_AnalyzeMachineData
- RSH_ClearMachineData

Solution Deployment & Configuration

The Robots Auto Healing solution is workflow based, so it runs on top of UiPath’s RPA Platform.
The implications of this dependency are that:

1. it can be deployed in all environments compatible with UiPath’s platform
2. It can monitor robot machines deployed in on-prem environments, in private & public clouds, in UiPath’s RPA Cloud



Your entire robotic workforce is managed, controlled and monitored by **UiPath Orchestrator**, the browser-based server application that centralizes in one secure place your entire enterprise operation.

All our product’s components can run on **both physical or virtual machines.**

Database

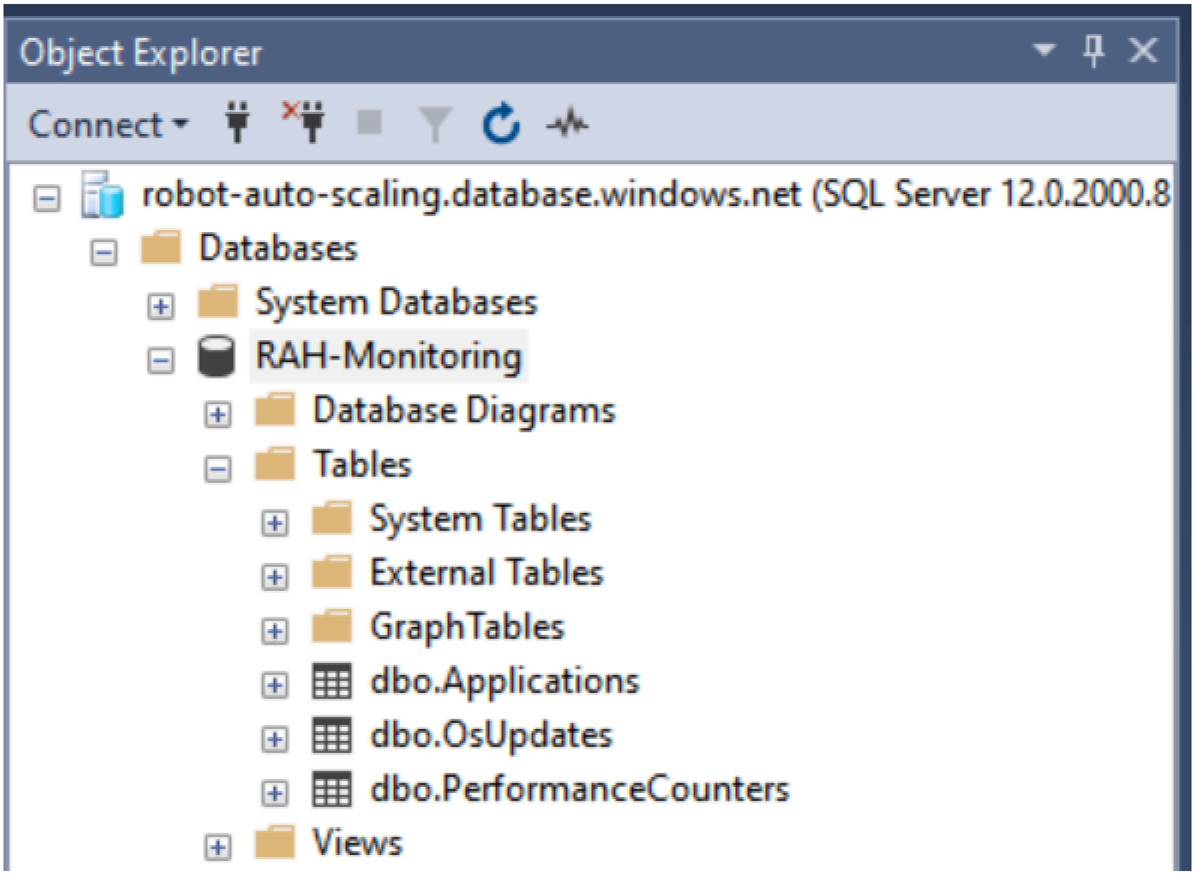
The database is used for monitoring robot machines for application version updates, OS updates, resources usage (cpu, ram, free disk space). The processes associated with this usecase are: *RSH_GetMachineData, RSH_AnalyzeMachineData, RSH_ClearMachineData.*

Note: If you don't plan to use RAH monitoring, you don't need to deploy the database or the 3 processes listed above.

Database Deployment

Create a Database for on a SQL Server and run the *RAHdb.sql* script(file provided in the RAH solution .zip archive) to create the solution tables.

Save the DB Connection String in the *RAH_DbConnectionString* asset. Example value:
Server=tcp: YOUR-SERVER,1433; Initial Catalog=YOUR-DATABASE; Persist Security Info=False; User ID=YOUR-USER; Password=YOUR-PASSWORD; MultipleActiveResultSets=False; Encrypt=True; TrustServerCertificate=False; Connection Timeout=30;



Management Orchestrator

The RAH solution components (process packages, assets) need to be deployed in an Orchestrator folder.

Basic Setup

In the Orchestrator where the RAH solution is to be deployed you will need to have the following:

- create a new Folder for the solution (any name)
- add Robot(s)
    - the solution's processes are background running and not resource intensive, so 1 unattended robot should be enough to handle the RAH load
    - it is recommended to have at least 2 robots for High Availability reasons

RAH Solution Packages

The RAH Solution .zip archive contain both the process *.nupkg* file(s) as well as the UiPath Studio workflow project files (the xaml sources). There are to options for adding the RAH processes:

1. import the .nupkg package in your Orchestrator tenant
2. you can publish it from Studio (open the RAH projects from the workflow sources)

After the packages are in the Management Orchestrator, create the processes.

Orchestrator Assets

Add the following assets to your Management Orchestrator RAH Folder.

Note:
Not all infrastructure related assets need to be added: the ones that don't apply to your managed clients can be skipped (eg. skip the Azure assets if all your robot machines are in AWS).

Core

RAH_InfrastructureType (Text)
Specifies the robot machines infrastructure type – one of the values: *aws*, *azure*

Machines Monitoring

If you don't plan to use RAH monitoring, you don't need to deploy the database related assets.

RAH_DbConnectionString (Text)
SQL connection string used to connect to the RAH database

RAH_DbMonitoringDataClear (Integer)
Number of days to keep monitoring data in RAH database.

RAH_DbMonitoringThresholdCPUpercentage (Integer)
Average CPU % usage limit – notification is sent when average value from database logs is above it.

RAH_DbMonitoringThresholdRAMpercentage (Integer)
Average RAM % usage limit – notification is sent when average value from database logs is above it.

RAH_DbMonitoringThresholdCdriveFreeGb (Integer)
C: drive free spage limit (in Gb) – notification is sent when min value from database logs is below it.

Orchestrator API

RAH_Orchestrator_IsOnPrem (Bool)
Specifies if the Orchestrator deployment you are connecting to is on-prem or in the UiPath cloud.

RAH_Orchestrator_AuthUrl (Text)
Orchestrator API authentication url. Example value: *https://account.uipath.com/oauth/token*

RAH_Orchestrator_Url (Text)
Orchestrator API url. Example value: *https://platform.uipath.com/*

RAH_Orchestrator_TenantLogicalName (Text)
Orchestrator API tenant logical name.

RAH_Orchestrator_Client (Credential)
Add to the asset's *Username* the *ClientID / username* of the Orchestrator *API*
Add to the asset's *Value* the *ClientSecret / password* of the *Orchestrator API*

RAH_Orchestrator_TenantService (Text)
Orchestrator API tenant service. Set value to "-" for on-prem Orchestrator deployments.

Notifications. Email

EmailNotification_Account (Credential)
Username and password used for connecting to the email server.

EmailNotification_Server (Text)
SMTP email server address. Example: *smtp.office365.com*

EmailNotification_Server (Integer)
SMTP email server address. Example: *587*

EmailNotification_To (Text)
Address(es) that receive RAH notifications.

Infrastructure Access. Azure

If you robot machines are not in Azure, you don't need to deploy the Azure related assets.

AzureAuth_AppRegistration (Credential)
Add to the asset's *Username* the *ClientID* of the *AppRegistration* used to connect to Azure.
Add to the asset's *Value* the *ClientSecret* of the *AppRegistration* used to connect to Azure.
Documentation: UiPath Azure Scope activity

Azure_SubscriptionID (Text)
Azure Subscription ID where the resource group(s) from the *Azure_VMs_RG* asset are located.

Azure_TenantID (Text)
Azure Tenant ID associated with the *AppRegistration*.

Azure_VMs_RG (Text)
Name of the Azure resource group where the VMs associated with the robot machines are located (for multiple RGs, use a csv value – eg: *RG1,RG2*)

Infrastructure Access. Amazon Web Services

If you robot machines are not in AWS, you don't need to deploy the AWS related assets.

AmazonAWS_IAM_Role (Text)
IAM Role to be used for connecting to Amazon Web Services.

- can be used only if the robot(s) running the RAH jobs are deployed in AWS virtual machines

If value is not "-", this asset will take precedence over the *AmazonAWS_AccessKey* asset.

AmazonAWS_AccessKey (Credential)
Add to the asset's *Username* the access key used to connect to Amazon Web Services.
Add to the asset's *Value* the secret key used for connecting to Amazon Web Services.
Documentation: UiPath AWS Scope activity

AmazonAWS_Region (Text)
The AWS Region to connect to.

Processes

RSH_RobotServiceCheck

Process logic:

- gets via the Orchestrator API the machine hostnames associated with disconnected or unavailable robots assigned to the folder specified by the FolderID workflow inargument
- checks if the machines are running
- for the running machines we check if the UiPath Robot Service is running; if it's not, we try to restart it

Mail Notification sample:

Running Machines with disconnected robots:

VM [RAH-Monitored1] UiRobotSvc had State =Stopped and RSH successfuly restarted it - robots should be now available in Orchestrator.

RSH_QueueItemInProgressLimit

Process Logic:

- gets all *InProgress* queue items for all Queues specified with the *QueuesCSV* inargument; Queues location is configured with the *QueueOrchestratorFolderPath* workflow inargument
- sends an email notification for all the queue items that are in progress for more hours than the threshold set with the *MaxHours* workflow inargument

Trigger

- add in Orchestrator a scheduled trigger (eg run job 1 time / day)
- 1 trigger can monitor multiple queues that you specify with the *QueuesCSV* argument
- the monitored queues need to be located in the same Orchestrator tenant as the RAH solution deployment

Mail Notification sample:

Queue Items exceeding the <in progress> time limit:

[RAH \ rsh3] started 09/01/2021 22:52:06 exceeds defined InProgress time limit - please check.
[RAH \ rsh2] started 09/01/2021 22:52:05 exceeds defined InProgress time limit - please check.
[RAH \ rsh1] started 09/01/2021 22:52:03 exceeds defined InProgress time limit - please check.

RSH_JobFailed (for chrome extension install)

Process Logic:

- gets via the Orchestrator API the failed jobs (last day) from the folder specified by the *FolderID* workflow inargument and filters them by error message *"Cannot communicate with the browser, please check the UiPath extension."*
- gets via the Orchestrator API the machine hostnames associated with the failed jobs
- for each resulting machine it tries to reinstall the UiPath Chrome browser extension in order to fix the cause of the failed jobs

Trigger:

- add in Orchestrator a scheduled trigger (eg run job 1 time / day)

Mail Notification sample:

> *Re-Installed Chrome Browser Extension in VMs: RSH-MONITORED1*
>
> *If problem persists, please rdp into the machines and check if the UiPath extension is enabled in the browser.*

Monitoring. RSH_GetMachineData

Process Logic:

- adds machine monitoring data in the RAH database for the category specified by workflow inargument *PoolingType*, for the running robot machines associated with the Orchestrator folder specified by the *FolderId workflow inargument*
    - data: installed apps, installed system OS updates, cpu + ram + C:\ free disk space
    - *PoolingType* possible values: apps, os, cpuramcdrive

Trigger

- add in Orchestrator scheduled trigger(s) (eg run job 1 time / day)

Monitoring. RSH_AnalyzeMachineData

Process Logic:

- analyzes monitoring data (installed apps, system OS updates, cpu & ram & free C disk space) from the RAH database for the category specified with *PoolingType* argument
- *PoolingType* possible values and associated checks:
    - *apps* - check if there are multiple versions logged for the same application
    - *os* - check if the system updates count differs (daily level)
    - *cpuramcdrive*
        - check if average cpu, ram usage are above configured threshold
        - check if C:\ drive free space is less than configured threshold
- send alert email (if the case)

Trigger:

- add in Orchestrator scheduled trigger(s) (eg run job 1 time / day)

Mail Notification sample:

RAH \ Servers Monitoring \ Machine AdminRobot1 \ OS updates were installed - updates count by date: [12/06/2021 00:00:00 - 15], [12/13/2021 00:00:00 - 14]

RAH \ Servers Monitoring \ Machine AdminRobot1 \ Apps with version changes: [Microsoft Monitoring Agent]

RAH \ Servers Monitoring \ Machine T204F5031-R1 \ CPU average usage 67% above threshold 60%
RAH \ Servers Monitoring \ Machine T204F5031-R1 \ RAM average usage 72% above threshold 70%
RAH \ Servers Monitoring \ Machine T204F5031-R1 \ C: drive free space 18Gb below threshold 20Gb

Monitoring. RSH_ClearMachineData

Process Logic:
- deletes from database monitoring data older than number of days specified in the Orchestrator asset *RAH_DbMonitoringDataClear*

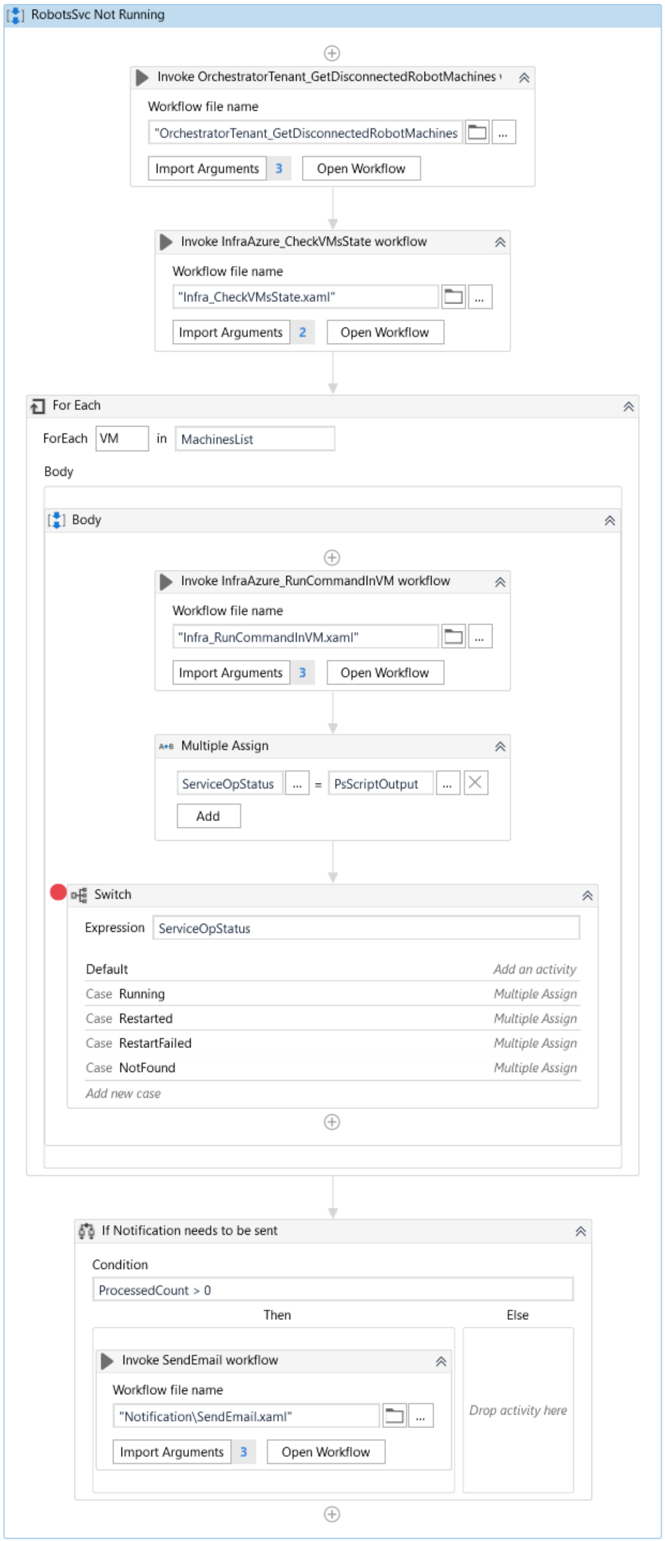
Trigger:
- add in Orchestrator a scheduled trigger (eg run job 1 time / day)

Mail Notification sample:

RAH \ Servers Monitoring - executed Database cleanup for entries older than 7 days.

Solution Customization

The Robots Auto Healing solution is workflow based and all the project's sources (workflow xaml files) are provided in the .zip archive. The workflows can be modified to accommodate customizations, new usecases and scenarios:



RobotsSvc Not Running

Invoke OrchestratorTenant_GetDisconnectedRobotMachines

Workflow file name: "OrchestratorTenant_GetDisconnectedRobotMachines

Import Arguments 3

Invoke InfraAzure_CheckVMsState workflow

Workflow file name: "Infra_CheckVMsState.xaml"

Import Arguments 2

For Each

ForEach VM in MachinesList

Body

Invoke InfraAzure_RunCommandInVM workflow

Workflow file name: "Infra_RunCommandInVM.xaml"

Import Arguments 3

Multiple Assign

ServiceOpStatus = PsScriptOutput

Switch

Expression: ServiceOpStatus

Default: Add an activity

Case Running: Multiple Assign

Case Restarted: Multiple Assign

Case RestartFailed: Multiple Assign

Case NotFound: Multiple Assign

If Notification needs to be sent

Condition: ProcessedCount > 0

Then

Invoke SendEmail workflow

Workflow file name: "Notification\SendEmail.xaml"

Import Arguments 3

Else: Drop activity here

Release Notes

Version 1.5

- Infrastructure types: Azure, AWS
- Processes
    - RSH_RobotServiceCheck
    - RSH_QueueItemInProgressLimit
    - RSH_JobFailed
        - chrome extension install
    - RSH_GetMachineData
    - RSH_AnalyzeMachineData
    - RSH_ClearMachineData