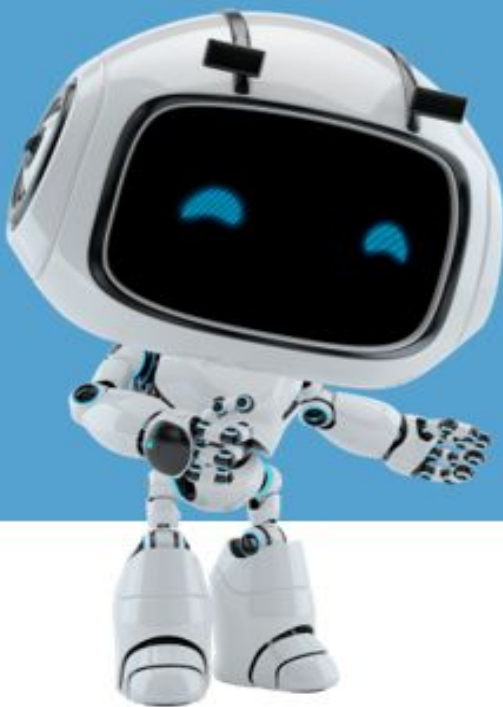




Robotic Process Automation



## UiPath Orchestrator Architecture

Information contained herein is confidential and may be disclosed only with the written permission of UiPath. All rights reserved.

UiPath Robotic Process Automation ®

## Revision History

Date	Version	Author	Description
28 <sup>th</sup> November 2016	2016.2	S.A	First Version
24th Jan 2017	2016.2	O.B.	Added DR option, updated disc space requirements

# Table of Contents

<b>1. Logical Decomposition</b>	<b>3</b>
1.1. Physical Deployment Options	6
1.1.1. Option 1 – Web Server on a Single Machine	6
1.1.2. Option 2 – High Performance and High Availability	7
1.1.3. Option 3 – Deployment in the cloud	8
1.1.4. Disaster Recovery Option	9
1.1.5. Disaster Recovery with Two Active Data Centers	10
1.2. Security Considerations	11
1.3. Infrastructure Requirements	12
<b>2. Orchestrator Hardware Requirements</b>	<b>13</b>
2.1. For Demo, Dev and Test environment (10 – 50 robots)	13
2.2. Production environment (10 – 200 robots)	15
Network Load Balancer	17
2.3 TCP Ports	17
<b>3. Client Hardware Requirements</b>	<b>18</b>

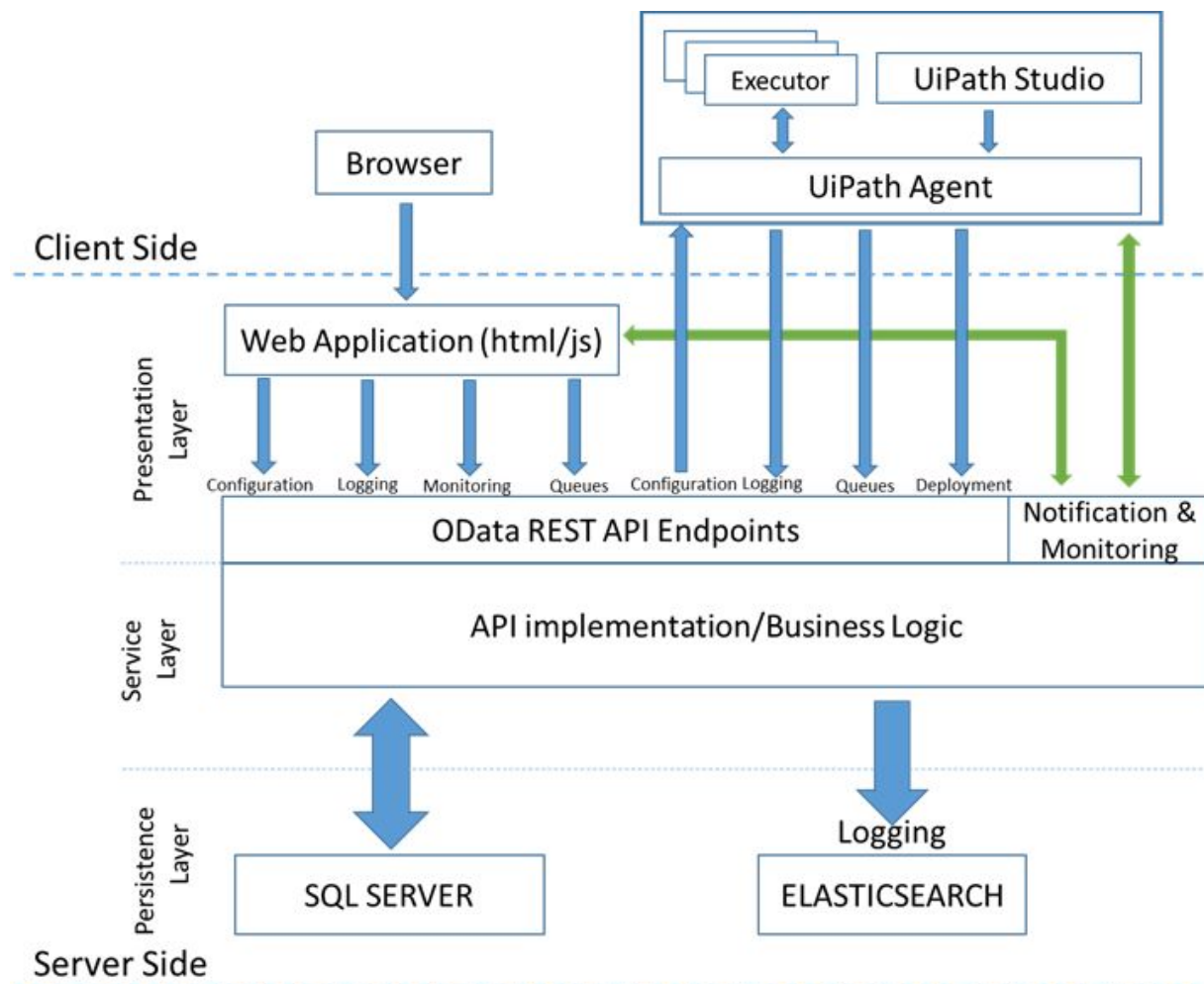
# Server Platform Architecture

UiPath Server is designed to monitor, log and control the execution of Robots on client computers. It is called Orchestrator.

## 1. Logical Decomposition

The UiPath Server Platform has the following logical components, grouped in three layers:

1. Presentation Layer
  - Web Application
  - OData REST API Endpoints
  - Notification API
2. Web Service Layer
  - REST API implementation
3. Persistence Layer
  - SQL Server
  - ElasticSearch



The **Web Application** is the visual layer of the Server Platform. The user interacts with its web pages for controlling and monitoring the robots - creating robot groups and assigning robots to them, assigning processes to groups, analyzing logs per robot or per process, starting and stopping the robots. Besides the web pages, Orchestrator contains also a service layer which exposes a REST API consisting mainly in OData endpoints.

The REST API is consumed by both the Web Application and the Agents. The Agent is the supervisor of one or more robots on the client computer.

The REST API covers all the Orchestrator functionalities:

- Configuration - REST endpoints used to define and configure application users, permissions, robots, assets, releases and environments
- Monitoring and Notification - REST endpoints used for registering the Agents, delivering configuration settings to the Agents and also used for sending or receiving notification from Server and the Agent. The Notification API uses also a WebSocket communication.
- Logging - REST endpoints used to log different information like errors, explicit messages sent by the robots and other environment specific information etc.
- Deployment - REST endpoints used by the robots to query the package version that needs to be started as a result of Start command from Orchestrator
- Queues - REST endpoints responsible for queues and queue items management – adding data to the queue, obtaining a transaction item from the queue, setting the status of a transaction item

Persistence Layer is composed of:

1. SQL Server, used:
  - On one hand, to store the configuration of robots, robot groups and associated processes – information managed through the Web Application component.
  - On another hand, to manage the queues and the queue items.
  - Optionally, to store the messages logged by the robots (in lieu of Elasticsearch or in addition to Elasticsearch).
2. Indexer Server (Elasticsearch) whose role is to store and index the information logged by robots. The use of Indexer Server can be disabled by configuration.

For clarification, messages logged by the robots can be stored either in SQL Server or in Indexer Server or in both or in none.

The current Orchestrator version was designed with regards to multi-tenancy. The multi-tenancy feature is in beta version and it will be available in the next release. The actual approach is “Shared Schema”, meaning that multiple tenants are sharing the same database and the same database schema. All tenant related data is unambiguously defined in each table by its own TenantId.

The SQL Server versions supported are SQL Server 2008 R2, SQL Server 2012, SQL Server 2014 and SQL Server 2016, including the SQL Server Express edition. If the SQL Server is not installed on a separate machine, we recommend **to install it along with the Web Application Component**. This leaves the Indexer Server alone, on a separate machine, due to resources requirements.

The **Indexer Server** uses the [Elasticsearch](#) (an open source project) full-text search engine. All the messages logged by the robots (using activities like Log Message, Write Line) are sent through the Logging REST endpoint to the Indexer Server where they are indexed for future utilization.

On the client computer, a running process is represented in the above diagram as an **Executor**. There can be several business projects running simultaneously, each project having a corresponding Executor. The **UiPath Agent** (a Windows service) is the single point of contact for all the Executors, through which all the messages are logged to the Orchestrator Service, which persists them further (**Indexer Server** or **SQL Server Database** or both).

A **robot** represents an association between machine name and username. A robot can manage multiple executors at the same time. On systems which support multiple interactive sessions running simultaneously (e.g. Windows Server 2012) there can be multiple robots running in parallel, each in a separate Windows session using different usernames.

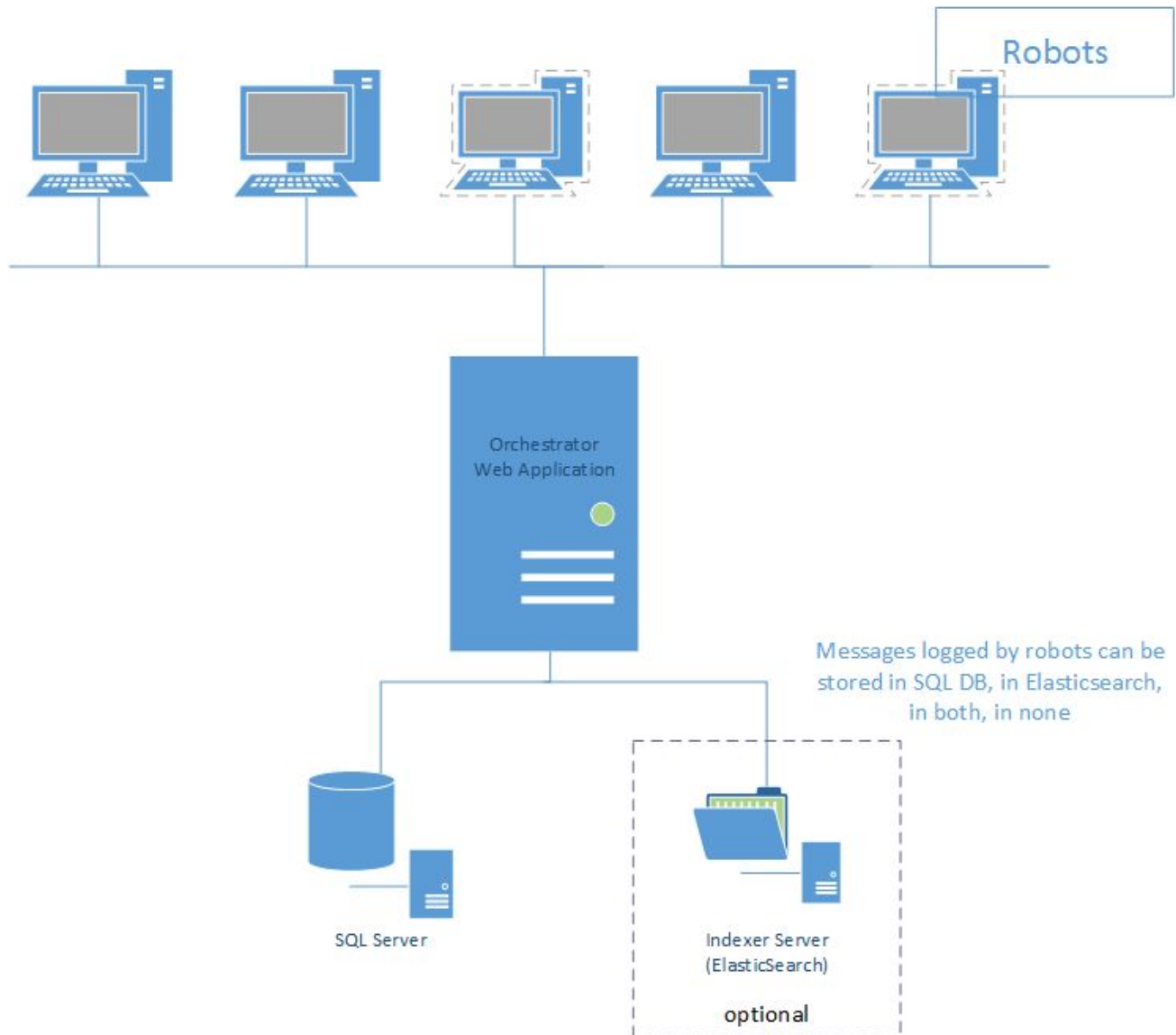
The **UiPath Agent** is also responsible for sending the status of the robot (e.g. **SubmitHeartBeat** endpoint) and downloading the required version of the package to be executed.

Communication between the Agent and Orchestrator is always initiated by the Agent. In the notification scenario, the Agent opens a WebSocket channel which is later used by Orchestrator to send commands to the Robot (start, stop, etc.).

## 1.1. Physical Deployment Options

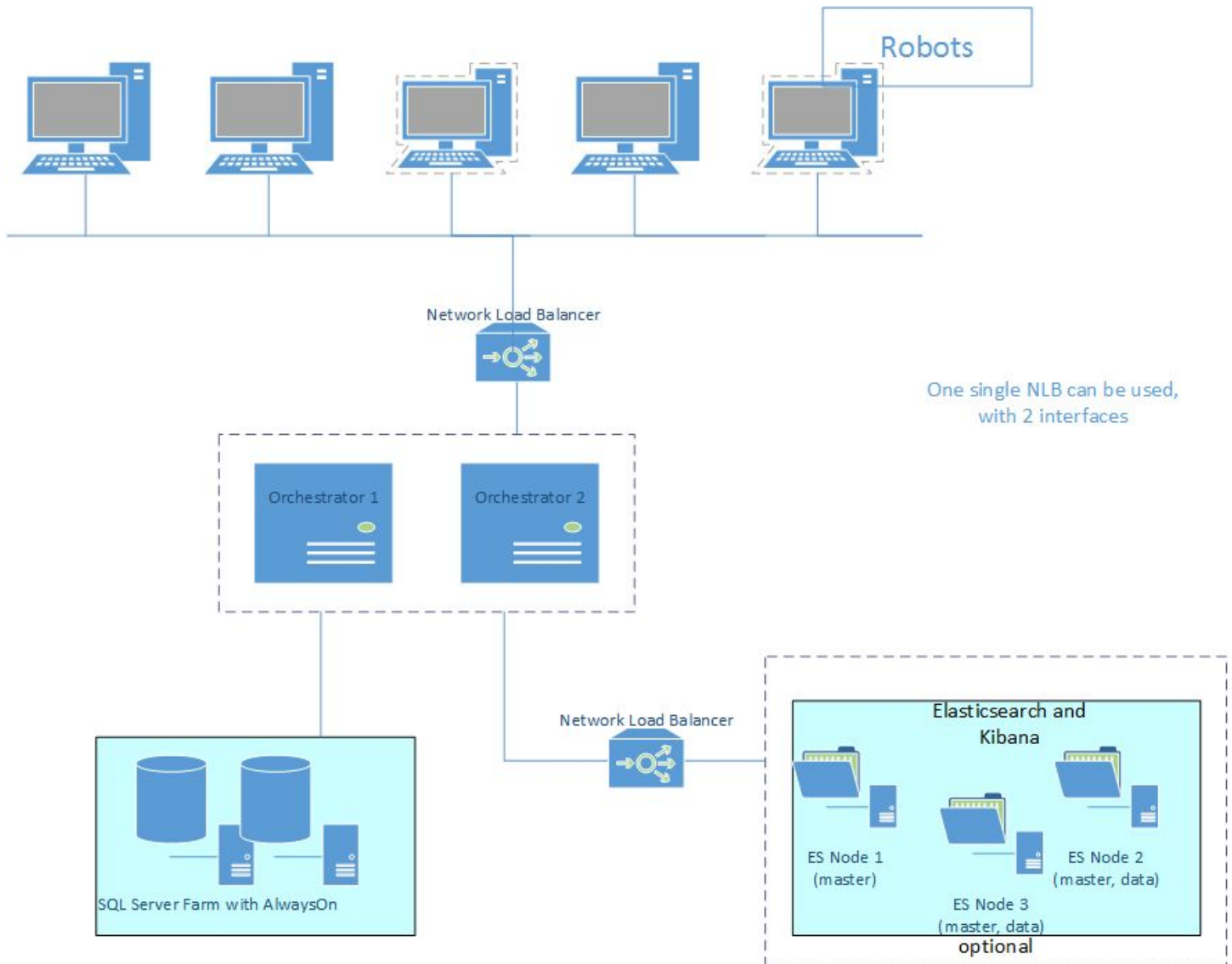
### 1.1.1. Option 1 – Web Server on a Single Machine

Recommended for medium scale deployment of Robots (10-50 Robots)



## 1.1.2. Option 2 – High Performance and High Availability

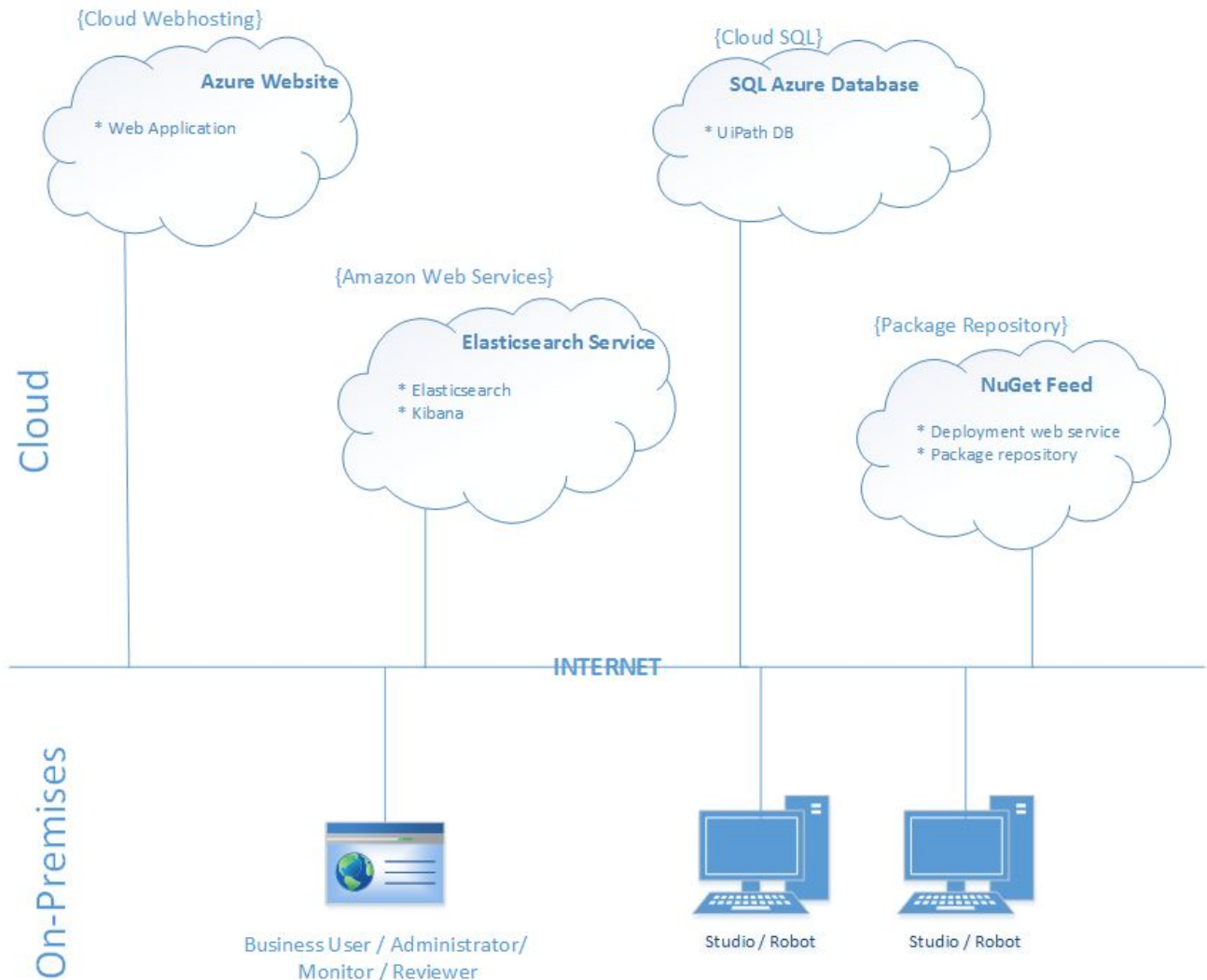
Recommended for large scale deployments of the robots (100 – 200 robots). Offers both horizontal scalability and high availability, due to multiple Orchestrator nodes.





### 1.1.3. Option 3 – Deployment in the cloud

Several combinations are available. The diagram below shows a combination of 3 cloud services – Azure, Amazon and MyGet.



Orchestrator is a Web Application which exposes several REST API endpoints. Communication (between robots and Orchestrator, between Orchestrator and the Database Engine and between Orchestrator and Elasticsearch) requires the web service URL, which makes the physical location irrelevant.

**Azure Website** – deploy the web application on the Microsoft Azure Website

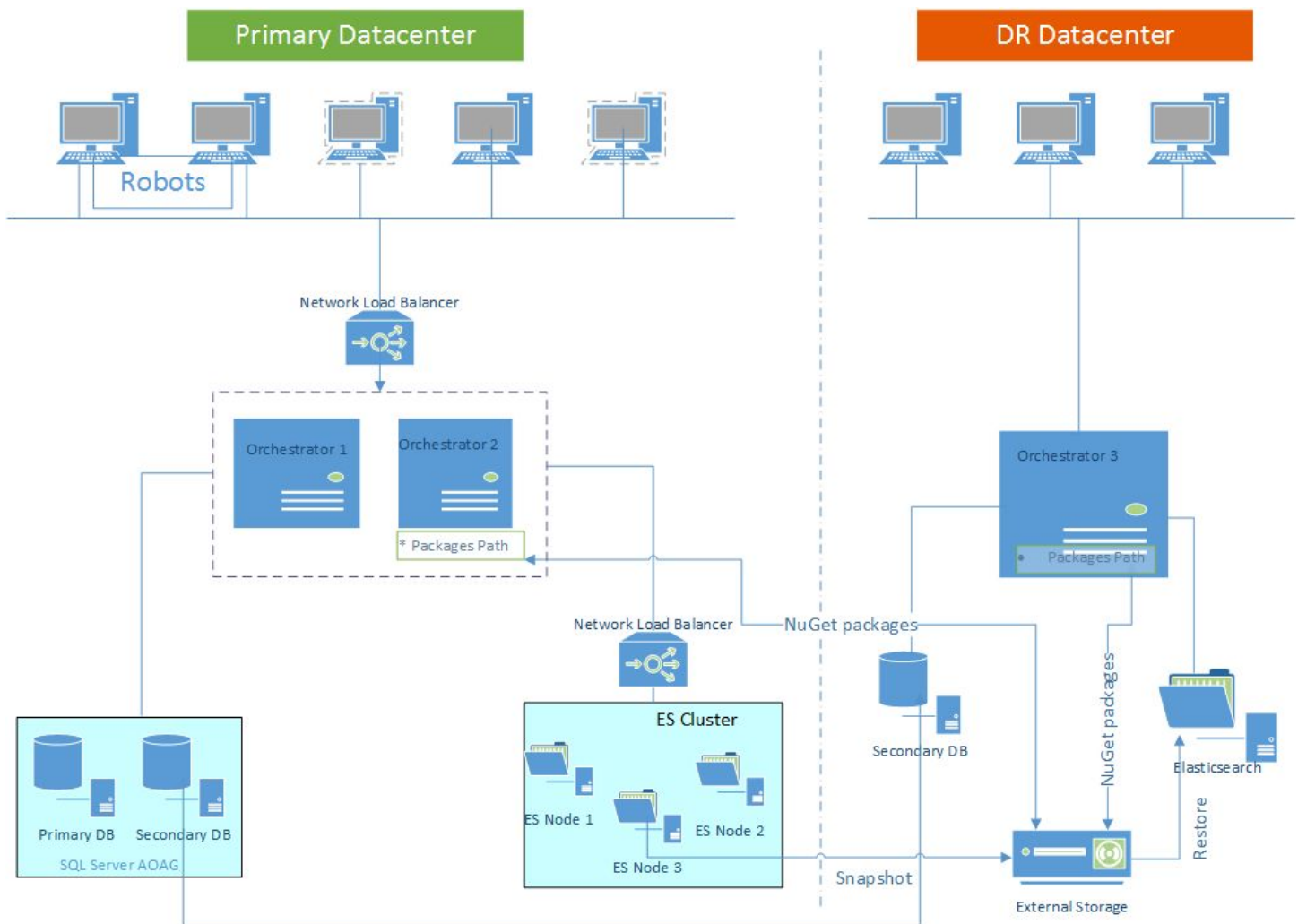
**SQL Azure Database** – use the SQL Server as a service to host the necessary databases

**Elasticsearch Service** – use the Elasticsearch service offered by AWS (Amazon Web Services)

**NuGet Feed** – there are several NuGet feed repositories, use the web service URL provided by the hosting company

## 1.1.4. Disaster Recovery Option

The following deployment configuration extends the High-Availability configuration in the Primary Datacenter with a Disaster Recovery option in a secondary Datacenter. A reduced number of machines is envisioned for DR, considering that the secondary Datacenter is provisioned for temporary use, until the Primary Datacenter is rebuilt.



The following conditions must be met:

- At least one machine from the SQL Server Farm with Always On Availability Group is physically located in the DR Datacenter
- There is network connectivity between Primary Datacenter and DR Datacenter
- External storage is provided, located in DR Datacenter, to store the Elasticsearch snapshots created by "Snapshot" in Primary DC. These snapshots are read and applied by "Restore" to the Elasticsearch indexes located in DR DC. "Snapshot" and "Restore" are tools provided by Elasticsearch.
- Automation packages (artifacts), having the NuGet format, are stored on External storage and each Orchestrator points to this storage using the "PackagesPath" configuration setting.

- Optional (not figured in the diagram) the External storage is mirrored.

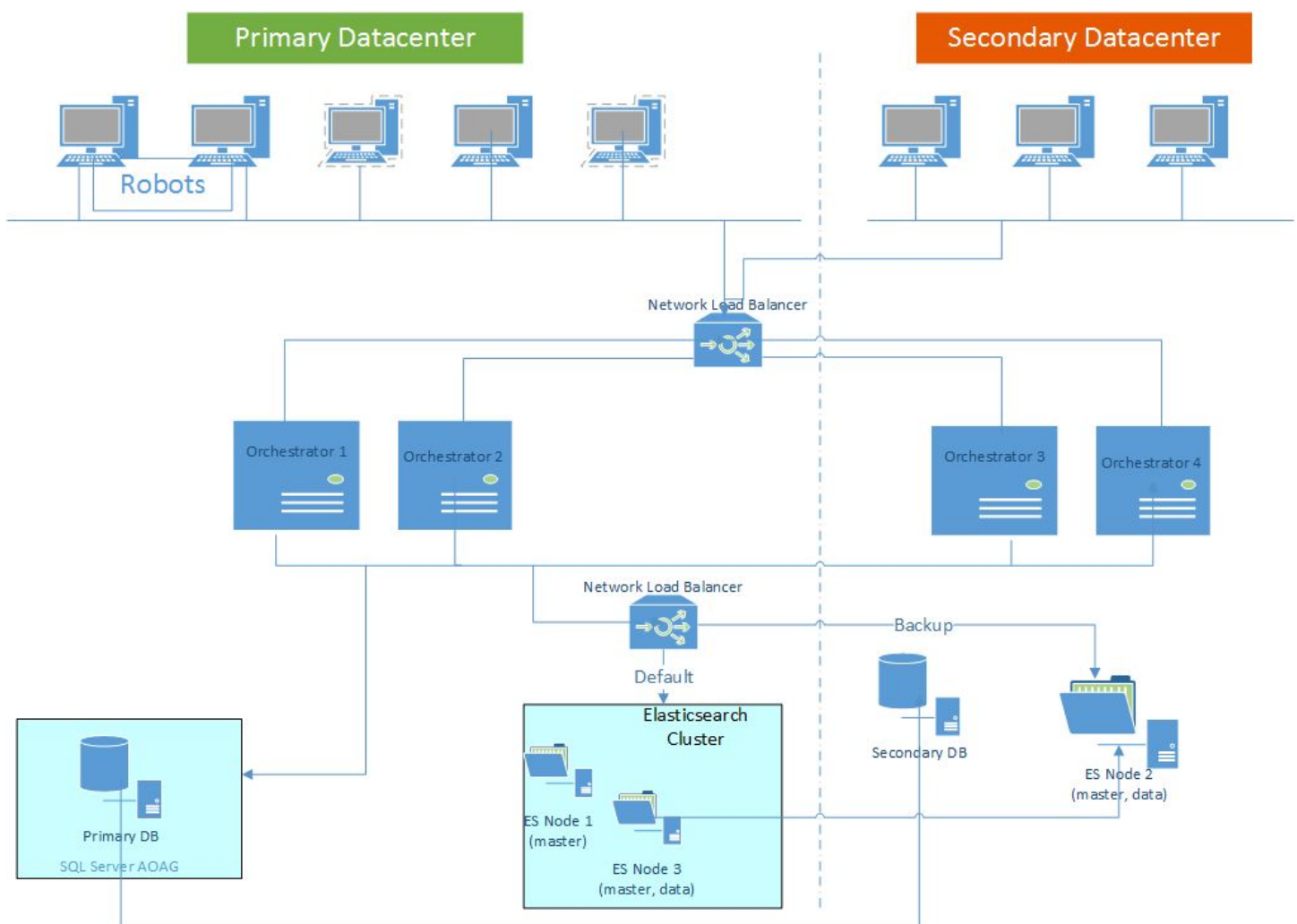
## 1.1.5. Disaster Recovery with Two Active Data Centers

The following deployment model can be implemented to ensure both High Availability and Disaster Recovery.

This model assumes there is a good network communication between the datacenters that are located in different geographical areas.

This deployment model can be primarily implemented on premises. For the cloud, you need to pick different regions for the primary and for the secondary cloud locations.

All the Orchestrator nodes are **active**, the Load Balancer will direct the traffic to all of them in a Round Robin mechanism.



SQL Server farm can be composed of minimum 2 machines - Primary DB in the first Data Center and Secondary DB in the second Data Center. For Elasticsearch minimum 3 nodes are required, 2 of them will be placed in the first Data Center and the third in the second Data Center.

## 1.2. Security Considerations

The following authentication types are supported

- Standard using Username and Password
- AD Username using Windows Authentication
- Google Account using OAuth 2.0
- Token based authentication and authorization for robots

The authorization token is stored for the first three types in a cookie and for the last one in a HTTP header.

Orchestrator is SSL enabled. In order to have a fully secured communication is critical to have HTTPS configured at the IIS level.

**The communication between Agent and Orchestrator is always initiated by the Agent.**

## 1.3. Infrastructure Requirements

The minimum requirements for the server platform OS are:

- Windows Server 2008 R2 SP3 or
- Windows Server 2012 R2
- Windows Server 2016

The SQL Server supported versions are:

- SQL Server 2008 Standard Edition with at least SP3
- SQL Server 2012 Standard Edition or Enterprise Edition
- SQL Server 2014 Standard Edition or Enterprise Edition
- SQL Server 2016 Standard Edition or Enterprise Edition

The Customer is responsible for the licenses of the Windows Operating System and SQL Server.

## 2. Orchestrator Hardware Requirements

### 2.1. For Demo, Dev and Test environment (10 – 50 robots)

These hardware requirements apply to [Option 1](#).

V1 – 3 servers

#### Web Application Server

10-50 robots		
	Min	Recommended
<b>CPU</b>	2 X 1.8Ghz cores	4 X 2.4Ghz cores
<b>RAM</b>	4GB	8GB
<b>HDD</b>	50GB	100GB
<b>OS</b>	Windows Server 2008 R2 SP3 OR Windows Server 2012 R2 OR Windows Server 2016	

#### SQL Server

10-50 robots		
	Min	Recommended
<b>CPU</b>	2 X 1.8Ghz cores	4 X 2.4Ghz cores
<b>RAM</b>	6GB	12GB
<b>HDD(*)</b>	80GB	150GB
<b>OS</b>	Windows Server 2008 R2 OR Windows Server 2012 R2 OR Windows Server 2016	
<b>Relational DB</b>	SQL Server 2008 R2 Standard Edition w/ Service Pack 3 OR SQL Server 2012 R2 Standard Edition OR SQL Server 2014 Standard Edition OR SQL Server 2016 Standard Edition	

(\*) Disc space requirements highly depend on:

- whether work queues are used or not; if work queues are used, it depends on average number of transactions added daily/weekly and size (number of fields, size of each field) of each transaction
- the retention period for successfully processed queue items (the customer should implement their own retention policy)

- whether messages logged by the robots are stored or not in the database; if they are stored, a filter can be applied to only store in the DB specific levels of messages (for example, store in the DB the messages with log level “Error” and “Critical”, and store in Elasticsearch messages with log level “Info”, “Warn” and “Trace”)
- frequency of logging messages - the robot developer uses the “Log Message” activity at will, whenever they consider a message is worth to be logged;
- the retention period for old logged messages (the customer should implement their own retention policy)
- logging level value set up in the robot; for example, if logging level in the robot is set to “Info”, only messages with levels “Info”, “Warn”, “Error” and “Critical” are sent to Orchestrator; messages with levels “Debug”, “Trace” and “Verbose” are ignored, they will not reach Orchestrator.

### ElasticSearch Server

10-50 robots		
	Min	Recommended
<b>CPU</b>	2 X 1.8Ghz cores	4 X 2.4Ghz cores
<b>RAM</b>	6GB	12GB
<b>HDD(**)</b>	100GB	250GB
<b>OS</b>	Windows Server 2008 R2 OR Windows Server 2012 R2 OR Windows Server 2016 RedHat Linux Ubuntu 16	

(\*\*) Disc space requirements depend on:

- the retention period (the customer should implement their own retention policy)
- frequency of logging messages - the robot developer uses the “Log Message” activity at will, whenever they consider a message is worth to be logged;
- logging level value set up in the robot; for example, if logging level in the robot is set to “Info”, only messages with levels “Info”, “Warn”, “Error” and “Critical” are sent to Orchestrator; messages with levels “Debug”, “Trace” and “Verbose” are ignored, they will not reach Orchestrator.

V2 – 2 Servers

### Web Application Server and ElasticSearch

10-50 robots		
	Min	Recommended

<b>CPU</b>	2 X 2.4Ghz cores	4 X 2.4Ghz cores
<b>RAM</b>	8GB	16GB
<b>HDD(**)</b>	150GB	250GB
<b>OS</b>	Windows Server 2008 R2 OR Windows Server 2012 R2 OR Windows Server 2016	

HDD(\*\*) - see the same remark above

#### SQL Server

10-50 robots		
	Min	Recommended
<b>CPU</b>	2 X 1.8Ghz cores	4 X 2.4Ghz cores
<b>RAM</b>	6GB	12GB
<b>HDD(*)</b>	80GB	150GB
<b>OS</b>	Windows Server 2008 R2 OR Windows Server 2012 R2 OR Windows Server 2016	
<b>Relational DB</b>	SQL Server 2008 R2 Standard Edition w/ Service Pack 3 OR SQL Server 2012 R2 Standard Edition OR SQL Server 2014 Standard Edition OR SQL Server 2016 Standard Edition	

HDD(\*) - see the same remark above

## 2.2. Production environment (10 – 200 robots)

These hardware requirements apply to [Option 2](#).

#### Web Application Server

10-200 robots		
	Min	Recommended
<b>CPU</b>	4 X 2.4Ghz cores	8 X 2.4Ghz cores
<b>RAM</b>	8GB	16GB
<b>HDD</b>	80GB	150GB



**OS** Windows Server 2008 R2 SP3 OR Windows Server 2012 R2 OR Windows Server 2016

## SQL Server

10-200 robots		
	Min	Recommended
<b>CPU</b>	4 X 2.4Ghz cores	8 X 2.4Ghz cores
<b>RAM</b>	12GB	32GB
<b>HDD(*)</b>	200-300GB	600-800GB
<b>OS</b>	Windows Server 2008 R2 OR Windows Server 2012 R2 OR Windows Server 2016	
<b>Relational DB</b>	SQL Server 2008 R2 Standard Edition w/ Service Pack 3 OR SQL Server 2012 R2 Standard Edition OR SQL Server 2014 Standard Edition OR SQL Server 2016 Standard Edition	

(\*) Disc space requirements highly depend on:

- whether work queues are used or not; if work queues are used, it depends on average number of transactions added daily/weekly and size (number of fields, size of each field) of each transaction
- the retention period for successfully processed queue items (the customer should implement their own retention policy)
- whether messages logged by the robots are stored or not in the database; if they are stored, a filter can be applied to only store in the DB specific levels of messages (for example, store in the DB the messages with log level "Error" and "Critical", and store in Elasticsearch messages with log level "Info", "Warn" and "Trace")
- frequency of logging messages - the robot developer uses the "Log Message" activity at will, whenever they consider a message is worth to be logged;
- the retention period for old logged messages (the customer should implement their own retention policy)
- logging level value set up in the robot; for example, if logging level in the robot is set to "Info", only messages with levels "Info", "Warn", "Error" and "Critical" are sent to Orchestrator; messages with levels "Debug", "Trace" and "Verbose" are ignored, they will not reach Orchestrator.

## ElasticSearch Server

10-200 robots		
	Min	Recommended
<b>CPU</b>	4 X 2.4Ghz cores	8 X 2.4Ghz cores

RAM	16GB	32GB
HDD(**)	200-300GB	500-600GB
OS	Windows Server 2008 R2 OR Windows Server 2012 R2 OR Windows Server 2016 RedHat Linux Ubuntu 16	

(\*\*) Disc space requirements depend on:

- the retention period (the customer should implement their own retention policy)
- frequency of logging messages - the robot developer uses the “Log Message” activity at will, whenever they consider a message is worth to be logged;
- logging level value set up in the robot; for example, if logging level in the robot is set to “Info”, only messages with levels “Info”, “Warn”, “Error” and “Critical” are sent to Orchestrator; messages with levels “Debug”, “Trace” and “Verbose” are ignored, they will not reach Orchestrator.

## Network Load Balancer

A Network Load Balancer (hardware or even software NLB, like NGINX, Zen Load Balancer, HAProxy) is required when Orchestrator is installed on multiple servers in a High Availability configuration.

## 2.3 TCP Ports

Communication between robots and the server takes place on the default web port – 80 for HTTP or 443 for HTTPS. The communication port can be specified at installation time, but it can also be changed after installation.

Orchestrator is communicating with Elasticsearch by default on port 9200 (configurable after Elasticsearch installation). If Elasticsearch and Orchestrator are installed on different computers, port 9200 should be opened for inbound access on the computer where Elasticsearch is installed. The firewall rule can use filters - to allow access from Orchestrator only.

The Kibana web plugin listens by default on port 5601 (configurable after Kibana installation). This port needs to be opened if Kibana needs to be accessed from other computers in the network, not only from the server where Orchestrator is installed.

The SQL Server port (by default 1433) needs to be opened to give access to the web application.

### 3. Client Hardware Requirements

	Minimum	Recommended
<b>Hardware</b>	PC (physical or virtual)	PC (physical or virtual)
CPU	2GHz 32-bit (x86)	4 core 64-bit (x64)
RAM	2GB	4GB
<b>Software</b>		
<i><b>UiPath Studio</b></i>		
OS	Windows 7 (update KB2533623)	Windows 7 (update KB2533623) and above
.NET	4.6	4.6
<i><b>UiPath Robot</b></i>		
OS	Windows 7 (update KB2533623)	Windows 7 (update KB2533623) and above
.NET	4.6	4.6

Setup Size on Disk*	Archived	Installed
	80MB	200MB

\* For both **UiPath Studio** and **UiPath Robot** (one single installer).