



# Automation Anywhere Enterprise V11.1 Performance Guide

---

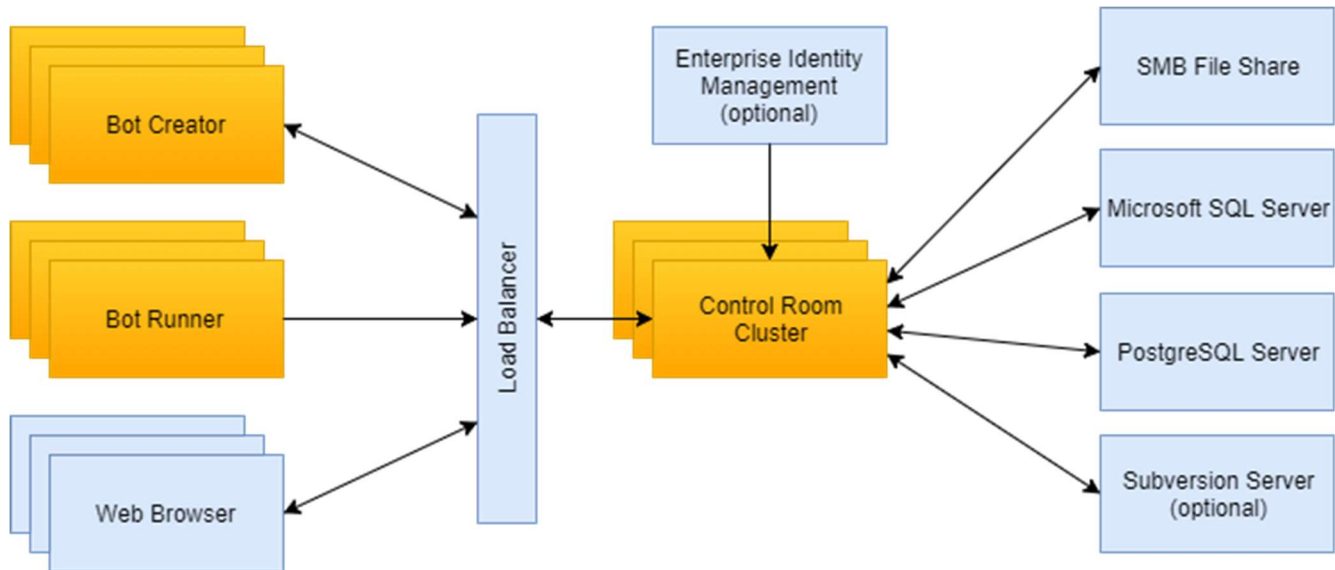
Revision 1

## Table of Contents

<b>1</b>	<b>Architecture .....</b>	<b>3</b>
1.1	Overview .....	3
<b>2</b>	<b>Reference Hardware Specifications .....</b>	<b>4</b>
<b>3</b>	<b>Entity Types and Counts .....</b>	<b>4</b>
<b>4</b>	<b>Quality of Service .....</b>	<b>5</b>
4.1	Request Originator Type.....	5
4.2	Repository Requests from Devices.....	5
4.3	Other Bot Deployment and Execution Requests from Devices .....	5
4.4	Security Token Refresh Requests from Devices .....	5
<b>5</b>	<b>Deployment and Repository Access.....</b>	<b>6</b>
5.1	Tuning Parallel Repository Operations .....	6
5.2	Tuning Other Parallel Deployment Operations .....	7
<b>6</b>	<b>RDP-based Auto-Login.....</b>	<b>7</b>
<b>7</b>	<b>Reactive Progress Processing.....</b>	<b>8</b>
<b>8</b>	<b>Workload Management .....</b>	<b>8</b>
8.1	Importing CSV Data .....	8
<b>9</b>	<b>Concurrent Schedules .....</b>	<b>9</b>

# 1 Architecture

## 1.1 Overview



COMPONENT	FUNCTION
<b>Control Room Cluster</b>	A group of servers that are used to manage, monitor and deploy bots.
<b>Bot Creator</b>	Software used to author and upload bots.
<b>Bot Runner</b>	Software used to run bots that have been authored via Bot Creator. Bot Runners download bots from the Control Room.
<b>Device</b>	A Bot Creator or a Bot Runner
<b>Web Browser</b>	Used by users and administrators to access the Control Room.
<b>Enterprise Identity Management</b>	The Control Room can optionally use an enterprise identity management system such as Active Directory, or a SAML identity provider, to authenticate users.
<b>SMB File Share</b>	An SMB file share is used by all members of the Control Room cluster to store or cache bots and configuration information. It must be accessible by every Control Room server.
<b>Microsoft SQL Server</b>	Primary database used to store all critical data related to Control Room operation including: bot, access control and analytics information.
<b>PostgreSQL Server</b>	Database used to store metadata related to analytics dashboards.
<b>Subversion Server</b>	When the optional bot versioning feature is enabled, bots are stored and versioned in an external Subversion server.
<b>Load Balancer</b>	Network Load Balancer that distributes HTTP(S) requests between Control Room cluster members

## 2 Reference Hardware Specifications

	<b>BOT RUNNER</b>	<b>BOT CREATOR</b>	<b>CONTROL ROOM</b>	<b>SQL SERVER</b>	<b>POSTGRESQL SERVER</b>
<b>Processor</b>	Intel Core i5 2.6 GHz	Intel Core i5 2.6 GHz	8 core - 3.0 Ghz Intel Xeon Platinum Processor (Turbo Boost to 3.5 Ghz)	4 core Intel Xeon Processor	2 core Intel Xeon Processor
<b>RAM</b>	8 GB	8 GB	16GB	8GB	4GB
<b>Storage</b>	32 GB	32 GB	500GB	500GB	10GB
<b>Network</b>	1GbE	1GbE	10GbE	1GbE	1GbE

## 3 Entity Types and Counts

The following entity type quantities are supported:

<b>Entity Type</b>	<b>Count</b>
Users	5000
Roles	2000
Schedules	1500
Audit Entries	5,000,000
Lockers	100
Credentials	5000
Repository Files	2500
Repository Folders	1250
Queues	10
Device Pools	10
Work Items	1,000,000

## 4 Quality of Service

To maintain high operational availability the Control Room is designed around the concept of Quality of Service (QoS).

Each incoming request is examined to identify:

1. Whether the request originated from a device or not
2. What high-level Control Room function it is for (Repository, Configuration, etc)

Based on these characteristics requests are prioritized.

### 4.1 Request Originator Type

API requests that do not originate from a bot creator or bot runner, such as a user accessing the Control Room via a web browser, are assigned a high priority. This allows administrative functions, such as viewing the status of a device or disabling an automation, to be performed even when the Control Room is fully utilized.

### 4.2 Repository Requests from Devices

By default, the number of device requests processed in parallel by the Control Room Repository are limited. Any requests that are received above the limit are fairly queued first in, first out with very little overhead. This allows the CPU and operating system thread consumption of repository operations to be controlled.

### 4.3 Other Bot Deployment and Execution Requests from Devices

A bot runner will make infrequent requests to other services on the Control Room related to the deployment and execution process. Unlike repository requests completion time for individual requests of this type are not influenced by network capacity.

### 4.4 Security Token Refresh Requests from Devices

Security token refresh requests from Devices are prioritized ahead of other device requests.

## 5 Deployment and Repository Access

Action	Maximum Concurrent Operations
Deployment and Execution	1000
Repository Upload from Bot Creator	100
Repository Download from Bot Creator	100

Number of Devices	Bot and Dependencies Total Size	Approximate Deployment Time
1000	10MB	1 minute
1000	50MB	1 minute
1000	100MB	5 minutes

Devices and the Control Room co-ordinate to implement a fair queuing strategy for download and uploads of chunked data to the Repository.

As the number of devices simultaneously downloading and uploading increase, the time taken for the Control Room to start processing a chunk request increases. A device will wait for up to two minutes for a response before timing out a request to upload or download. With the default limit of 10 parallel processed repository requests and simultaneous deployment and execution to 1000 devices, the average time to queue and process a chunk is approximately 10 seconds.

**Note:** Downloads and uploads from bot creators are subject to the same QoS policies as deployments.

### 5.1 Tuning Parallel Repository Operations

If the network connectivity between devices and the Control Room is slower than the reference configuration, or the CPU is under-utilized during deployment, the number of parallel repository requests can be increased for better throughput.

If the CPU speed of the Control Room is lower than the reference configuration or network speed is higher it may be necessary to lower the number of parallel repository requests allowed to avoid loss of availability due to excessive CPU utilization throughout deployment.

To adjust the number of parallel Repository requests:

1. Create a `boot.server.properties` file in the Program Files\Automation Anywhere\Enterprise\config directory if not already present.
2. Add a line:  
`requests.max.client.concurrent.repo=<number>`

## 5.2 Tuning Other Parallel Deployment Operations

If during execution the Control Room or database server experience a loss of availability due to excess CPU usage, the default of 10 parallel processed requests can be adjusted down.

If the Control Room server has more than 8 cores, or if latency is high between the Control Room and devices you can increase the number of parallel processed requests. Use the following procedure:

1. Create a `boot.server.properties` file in the Program Files\Automation Anywhere\Enterprise\config directory if not already present.
2. Add a line:

`requests.max.client.concurrent.other=<number>`

## 6 RDP-based Auto-Login

If the option *Run bot runner session on control room* is selected when running a bot, the Control Room opens an RDP session to the devices. The number of simultaneous sessions supported can be determined by dividing the unallocated Control Room memory by the memory usage of an individual RDP client, which is 75MB.

**Note:** It is necessary to manually increase the amount of memory available for the non-interactive desktop heap size to support more than 35 simultaneous RDP sessions. This can be done by editing the Windows registry value that exists at:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session  
Manager\SubSystems

Within the various options specified in the Windows registry value is the `SharedSection` option that should be modified.

SharedSection=A,B,C

Section	Description	Default Value (KB)	New Value (KB)
<b>A</b>	Maximum size of the system-wide heap	1024	1024
<b>B</b>	Size of each interactive desktop heap	20480	20480
<b>C</b>	Size of non-interactive desktop heap	768	<b>20480</b>

## 7 Reactive Progress Processing

To prevent overload when concurrently executing a large number of bots, processing of status update messages from bot runners is rate limited on a per-Control Room node basis. Additionally, the rate limit is adjusted dynamically based on the number of unprocessed status update messages.

No critical status update messages that indicate start, stop or error are ever rate-limited. If reactive rate-limiting is activated the progress reported on the Activity page will be updated at a lower frequency than normal.

## 8 Workload Management

### 8.1 Importing CSV Data

By default, Workload Management (WLM) has a conservatively configured CSV import rate to minimize resource utilization.

For a more aggressive import strategy the number of lines imported per batch, and the interval between each batch can be configured.

Number of CSV rows	Number of CSV columns	Approx. time to upload with default configuration	Approx. time to upload with a more aggressive strategy
<b>50,000</b>	15	25 minutes	2 minutes
<b>100,000</b>	15	50 minutes	3.5 minutes
<b>300,000</b>	15	2.5 hours	10 minutes
<b>1,000,000</b>	12	8.5 hours	35 minutes



Strategy	Number of lines per batch	Interval between each batch
Default	1000	30 seconds
Aggressive	10,000	10 seconds

To change strategy use the following procedure:

1. Create a wlm.properties file in the Program Files\Automation Anywhere\Enterprise\config directory if not already present.
2. Add the lines:

```
workOrder.execution.job.interval.seconds=<interval in seconds>
workOrder.max.execute.lines=<number of lines per batch>
```

## 9 Concurrent Schedules

When configuring bots to run repeatedly on a schedule it is important to make sure that the time between runs does not drop below the total time for deployment and execution of the bot. Otherwise, sequential executions of the bot may overlap leading to unexpected behavior.

With the reference specification, it is possible to successfully configure:

Bot Size	Concurrent Schedules	Devices per Schedule	Total Concurrent Bots	Overall Execution Time
1MB	100	10	1000	4 minutes
10MB	100	10	1000	9 minutes