

# Login Enterprise powered by NVIDIA nVector

Installation Guide (version 1.0.4)

September 2025

## Table of Contents

Summary.....	3
Getting Started.....	3
Pre-requisites.....	3
nVector and Login Enterprise integration .....	3
Downloads.....	5
Setup & Deployment Steps .....	5
Login Enterprise Virtual Appliance setup .....	5
Login Enterprise Launcher Machine(s) setup .....	6
Login Enterprise Session Metrics Configuration .....	7
Configuring your Login Enterprise Web UI to see Platform metrics when using a Login Enterprise version less than 6.4 .....	10
Testing.....	12
Reporting.....	14
Examples of Latency Results in Continuous Testing Results.....	14
Examples of Latency Results in Load Testing Results.....	15
Best Practices .....	17
Appendix A – Script Overview .....	18
nVector_Client_Prepare.ps1 (for Login Enterprise Launcher machine) .....	18
nVector_Startup_Script.cmd (for Login Enterprise Launcher) .....	19
nVector_Desktop_Prepare (C# Workload) .....	20
get_nVectorMetrics.ps1 .....	21
Additional Information.....	23

---

### Revision History:

1.0.0 - May 20, 2025 - Release for GA

1.0.1 - May 23, 2025 - Fixed how to download scripts

1.0.2 - June 13, 2025 - Fix instructions on uploading Desktop Prepare script

1.0.3 - July 20, 2025 - Fix for timezone and LE v 6.1.8 API issue

1.0.4 - Sept 2025 - Updated *get\_nVectorMetrics.ps1* section for v1.0.1 (adds *-ApiVersion*, *-ImportServerCert*, *-KeepCert*, *PS5/PS7* behavior, security notes, examples).

---

## Summary

Login VSI, in collaboration with NVIDIA, proposes a unique extension to Login Enterprise that leverages NVIDIA's GPU and virtual GPU (vGPU) performance testing and analysis technology. This integration aims to enhance graphical responsiveness and quality in virtual environments using NVIDIA's nVector agents and performance metrics for scalability testing and VDI production monitoring. This integration enables customers to optimize GPU-based configurations, ensuring optimal performance and cost-efficiency for their solutions.

## Getting Started

### Pre-requisites

1. A proper functioning VDI Solution based on Citrix, VMware, Omnicast, or Microsoft technologies.
2. A proper functioning Login Enterprise virtual appliance and launcher(s)
  - a. **NOTE:** Login Enterprise version 6.1.8 has a known issue, please use a later version.
3. Admin access to Login Enterprise virtual appliance
4. Knowledge of NVIDIA virtual GPU technology
5. An NVIDIA GPU or vGPU enabled launcher machine
6. Downloaded nVector-agent, and enablement scripts

### nVector and Login Enterprise integration

This solution contains a suite of integration scripts that wrap NVIDIA's **nvector-agent.exe** for use with the Login Enterprise platform. These scripts enable you to collect and upload round-trip graphics latency data—measured via NVIDIA nVector's “click-to-pixel-change” mechanism—to the Login Enterprise API as platform metrics. The metrics can then be visualized in the Login Enterprise web interface or retrieved via an API. In addition, the solution also adds NVIDIA suggested, VM-based session metrics for greater visibility into the performance of the system.

#### Important:

The **nvector-agent.exe** and its associated components are developed and distributed by **NVIDIA**. Until the **nvector-agent.exe** is made generally available, you must obtain this executable from

NVIDIA and deploy it in your environment. In the future the nvvector-agent.exe will be made available by Login VSI.

## Downloads

This solution requires four key files to integrate nVector functionality with Login Enterprise. Please request these from [b.parkhill@loginvsi.com](mailto:b.parkhill@loginvsi.com) - he will send you a link to download the files. We will eventually get a proper location to download the files.

1. Extract the files from their respective zip files into a temporary directory. The files you will need for the integration are as follows.
2. **nVector-agent.exe**
3. Login Enterprise launcher scripts (client prepare & startup script)
  - a. **nVector\_Client\_Prepare.ps1**
  - b. **nVector\_Startup\_Script.cmd**
  - c. More information about these scripts can be found in [Appendix A](#)
4. Login Enterprise desktop prepare script (virtual user application)
  - a. **nVector\_Desktop\_Prepare.cs**
  - b. More information about this script can be found in [Appendix A](#)

## Setup & Deployment Steps

Use the following checklist to record your steps

- ☐ Upload nVector agent to virtual appliance
- ☐ Create nVector environment on virtual appliance
- ☐ Create nVector virtual user startup app on virtual appliance
- ☐ Setup launcher with nVector agent
- ☐ Create session metrics and session metrics group
- ☐ Create load test
- ☐ Create continuous test
- ☐ Enable appliance WebUI for nVector platform metrics

## Login Enterprise Virtual Appliance setup

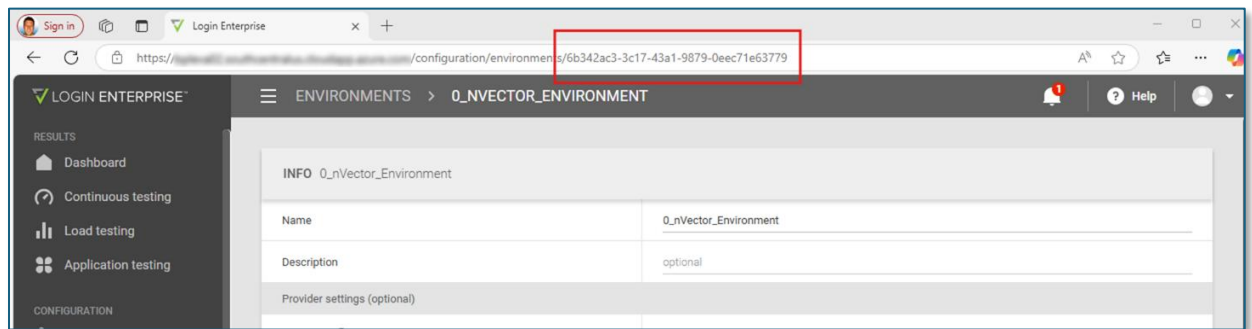
### Upload nvector-agent.exe to the Login Enterprise Virtual Appliance

1. Place the `nvector-agent.exe` executable on the Login Enterprise virtual appliance.  
Use SSH to access the virtual appliance or use SFTP tools like WinSCP if necessary to transfer the file. The location on the virtual appliance Linux filesystem is  
`loginvsi/content/scriptcontent/nvector-agent.exe.`

### Create an environment

1. On the virtual appliance, navigate to “Environments”

2. Click on “Add environment” in the top right corner of the UI
3. Give the environment a name (e.g., nvector environment)
4. Save
5. After clicking Save, select the environment and record the UUID from the page’s URL



## Create the nVector virtual user startup app

(for help see [Importing Applications](#))

1. On the virtual appliance, navigate to “Applications”
2. Click on the green plus sign in the top right corner of the UI to add the application
3. Select “IMPORT APPLICATION”
4. Browse to the “nvector\_Desktop\_Prepare.cs” script and select “Open”
5. Click “SAVE”

## Login Enterprise Launcher Machine(s) setup

1. If you don’t have a launcher yet, download installer from the virtual appliance and install launcher software
2. Copy `nVector_Client_Prepare.ps1`, `nVector_Startup_Script.cmd`, and `nvector-agent.exe` file to a directory on the launcher (e.g., `c:\temp\nvidia\`)
3. On the virtual appliance, [create a Public API configuration access token](#) and record it for use in the next step
4. Edit the `nVector_Client_Prepare.ps1` file to match the required parameters to your environment (see [Key Variables & Parameters](#)). The parameters you are required to set or confirm are: `CsvFilePath`, `NvectorScreenshotDir`, `NvectorLogFile`, `TranscriptFile`, `NvectorAgentExePath`, `ConfigurationAccessToken`, `BaseUrl`, and `EnvironmentId`.  
Here is an example.

```
# Full paths for CSV and logs
$CsvFilePath           = "C:\temp\nvidia\latency_metrics.csv"
$NvectorScreenshotDir = "C:\temp\nvidia\SSIM_screenshots"
$NvectorLogFile        = "C:\temp\nvidia\agent.log"

$TranscriptFile        = "C:\temp\nvidia\${Timestamp}nVector_Agent_Client.log"

# nVector-agent executable path
$NvectorAgentExePath = "" # Place full path to nvector-agent.exe here

# API configuration
$ConfigurationAccessToken = "abcd1234abcd1234abcd1234abcd1234abcd1234abc"
$BaseUrl                  = "https://myDomain.LoginEnterprise.com/"

$EnvironmentId          = "abcd1234-abcd1234-abcd1234-abcd1"
```

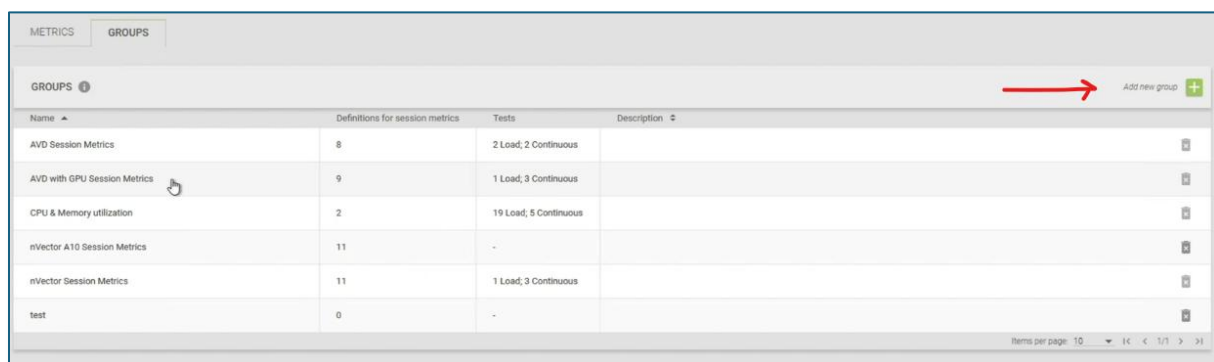
Note: As of nVector\_Client\_Prepare.ps1 version 1.1, TimeOffset is no longer required to be set.

5. Modify the nVector\_Startup\_Script.cmd file to add the full path to the nVector\_Client\_Prepare.ps1 (e.g., ... -File "c:\temp\nvidia\nVector\_Client\_Prepare.ps1")
6. Copy the nVector\_Startup\_Script.cmd to Windows Startup Folder (e.g., C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup). This will make sure that when the launcher is restarted, that the nVector agent will automatically start and log new data. For more information on how to start a script on logon see [this article for examples of other ways to automatically start a script](#).

## Login Enterprise Session Metrics Configuration

Session metrics allow us to capture performance metrics from the VM being tested. Both PerfMon counters and WMI counters can be added. See this [KB article](#) for more information.

1. Create a new session metrics group for nVector.



2. Create the session metrics you want to add to the group. For more information on creating session metrics see this [Login Enterprise Session Metrics KB article](#). Create NVIDIA GPU session metrics for GPU Usage, GPU Memory Usage and Frame Buffer Usage. You should open Perfmon on the target VM to get the information you need. Note that "Counter

Category”, “Counter Name”, and “Counter Instance” must exactly match the values you see in Perfmon. See the example below.

INFO ⓘ	
General	
Name	NVIDIA A10 % GPU Util
Description (optional)	
Source for session metrics	PerformanceCounter
Settings	
Counter Category	NVIDIA GPU
Counter Name	% GPU Usage
Counter Instance (optional)	#0 NVIDIA A10-4Q (id=1, NVAPI ID=131072)
Display Name	% A10 GPU Util
Unit ⓘ	%

**NOTE:** The instance name will change as you change the vGPU profile being used.

3. Add Protocol Framerate session metrics. Because there can be many instances to collect it is best to use WMI counters. You will want to collect Input and Output Framerate.

INFO ⓘ	
General	
Name	RemoteFX Input Framerate
Description (optional)	
Source for session metrics	WmiQuery
Settings	
WMI Query	SELECT InputFramesPerSecond FROM Win32_PerfFormattedData_Counters_RemoteFXGraphics
Name Space ⓘ	root\cimv2
Instance Field ⓘ	Name
Measurements ⓘ	
▼ InputFramesPerSecond	
Property Name ⓘ	InputFramesPerSecond
Summarize Operation ⓘ	Avg
Display Name	RemoteFX Input FPS
Unit ⓘ	FPS

- a. To discover WMI counters try the following in PowerShell
  - i. To list all counters run: `Get-cimclass | select-string PerfFormattedData`
  - ii. Once you find the counter set you'd like to find objects within, run:  
`Get-CimInstance Win32_PerfFormattedData_NameOfCounterSet (e.g., Get-`



CimInstance Win32\_PerfFormattedData\_Counters\_RemoteFXGraphics)

```
PS C:\Users\b.parkhill> get-cimclass | select-string PerfFormattedData_Counters_R
ROOT/cimv2:Win32_PerfFormattedData_Counters_RDMAActivity
ROOT/cimv2:Win32_PerfFormattedData_Counters_ReFS
ROOT/cimv2:Win32_PerfFormattedData_Counters_ReFSBucketizedPerformance
ROOT/cimv2:Win32_PerfFormattedData_Counters_ReFSdedupMinstorePerfCounters
ROOT/cimv2:Win32_PerfFormattedData_Counters_ReFSdedupPerfCounters
ROOT/cimv2:Win32_PerfFormattedData_Counters_RemoteFXGraphics
ROOT/cimv2:Win32_PerfFormattedData_Counters_RemoteFXNetwork
```

- iii. From that result you can find the counters you'd like to use for your session metrics.

```
PS C:\Users\b.parkhill> Get-CimInstance Win32_PerfFormattedData_Counters_RemoteFXGraphics
Caption :
Description :
Name : rdp-sxs250105600 0
Frequency_Object :
Frequency_PerfTime :
Frequency_Sys100NS :
Timestamp_Object :
Timestamp_PerfTime :
Timestamp_Sys100NS :
AverageEncodingTime : 0
FrameQuality : 100
FramesSkippedPerSecondInsufficientClientResources : 0
FramesSkippedPerSecondInsufficientNetworkResources : 0
FramesSkippedPerSecondInsufficientServerResources : 0
GraphicsCompressionratio : 6
InputFramesPerSecond : 3
OutputFramesPerSecond : 3
SourceFramesPerSecond : 0
PSComputerName :
```

4. Add individual session metrics to the nVector Session Metrics Group created above
  - a. CPU Utilization
  - b. Memory Utilization
  - c. GPU Usage
  - d. GPU Memory Usage
  - e. GPU Framebuffer Usage
  - f. Input FPS
  - g. Output FPS
  - h. Any others you've created and would like to have in this collection

Here is an example:

GROUP INFO ⓘ

Name	nVector Session Metrics
Description	optional

SESSION METRIC DEFINITIONS

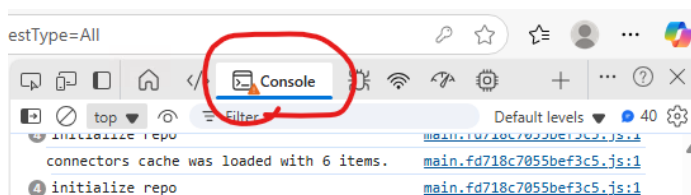
Name	Source	Description
CPU utilization	Builtin	
Memory utilization	Builtin	
Committed Bytes	PerformanceCounter	
Available MBytes	PerformanceCounter	
Avg. Disk sec/Transfer	PerformanceCounter	
NVIDIA GPU FB Usage	PerformanceCounter	
NVIDIA GPU Usage	PerformanceCounter	
NVIDIA GPU Memory Usage	PerformanceCounter	
Frame Quality	WmiQuery	
RemoteFX Input Framerate	WmiQuery	
RemoteFX Output Framerate	WmiQuery	

## Configuring your Login Enterprise Web UI to see Platform metrics when using a Login Enterprise version less than 6.4

As of Login Enterprise version 5.13.6, the virtual appliance is capable of logging 3<sup>rd</sup> party performance data to a time-series database. To see this data the browser used for the appliance WebUI must have a feature flag enabled.

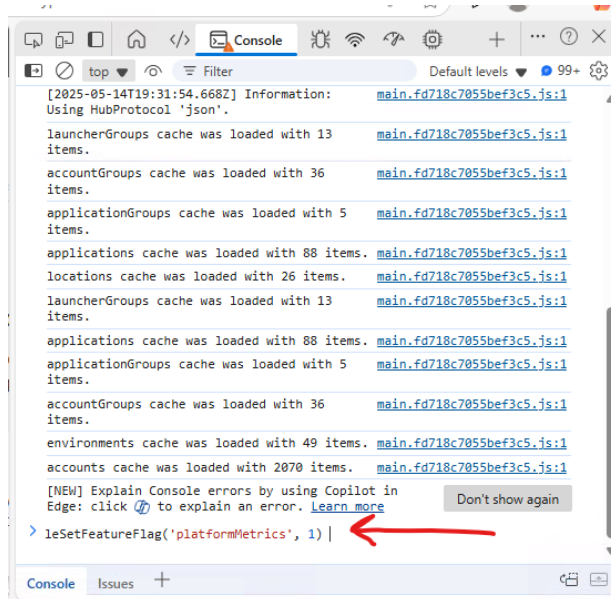
### Feature flag instructions:

1. Open the browser's developer mode. (e.g., hit F12) and select the "Console" view.

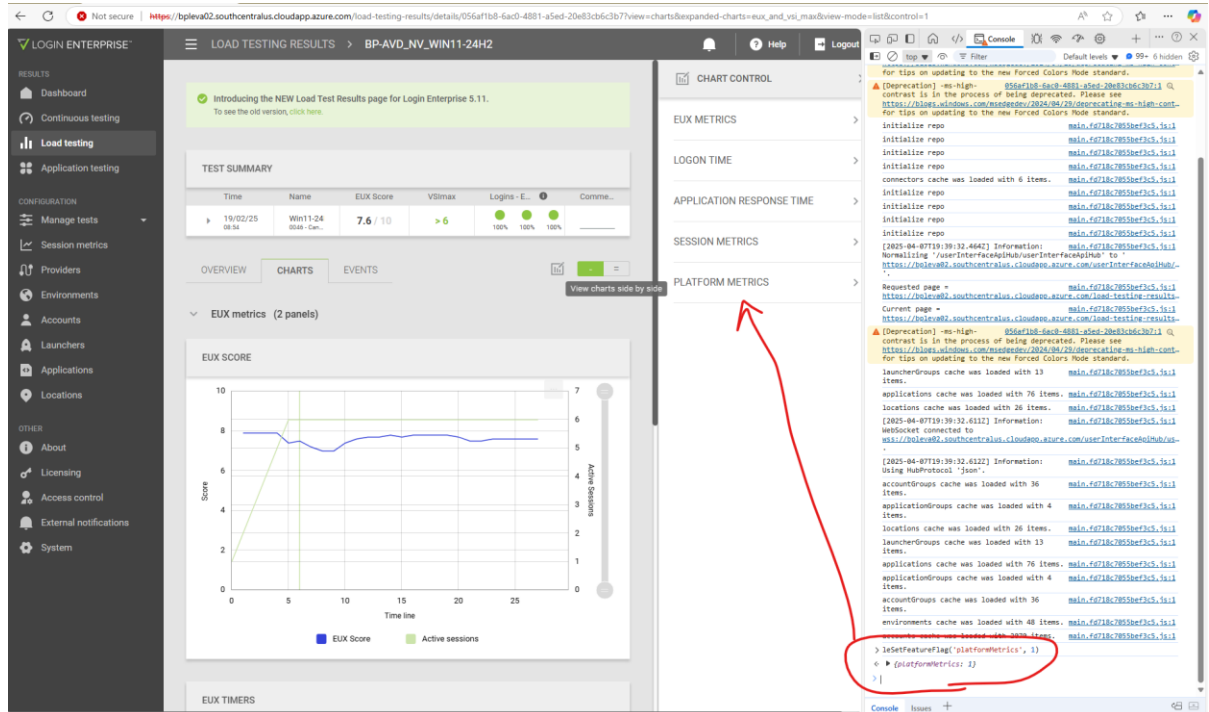


2. Enter the following command at the prompt on the bottom of the page to enable the feature:

```
leSetFeatureFlag('platformMetrics', 1)
```











3. Press **Enter** to execute the command. Platform Metrics should now be visible.



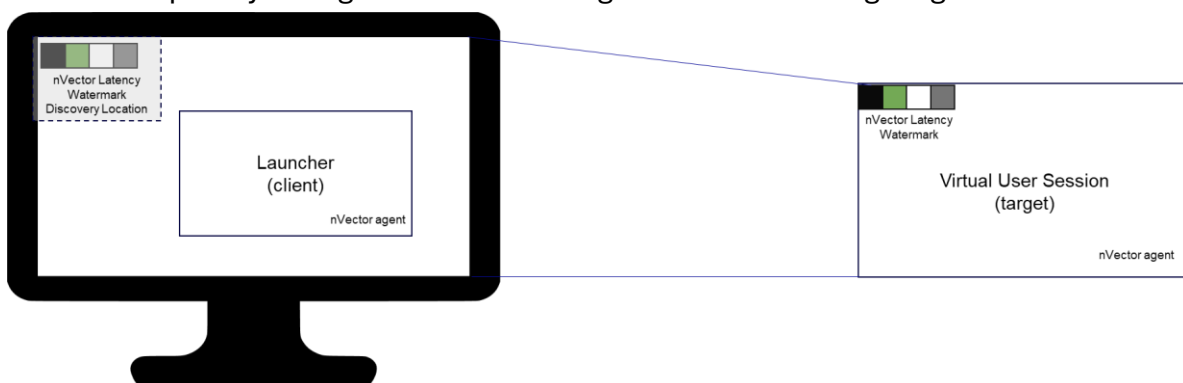
4. Please note that the feature will remain hidden under this flag until further notice.

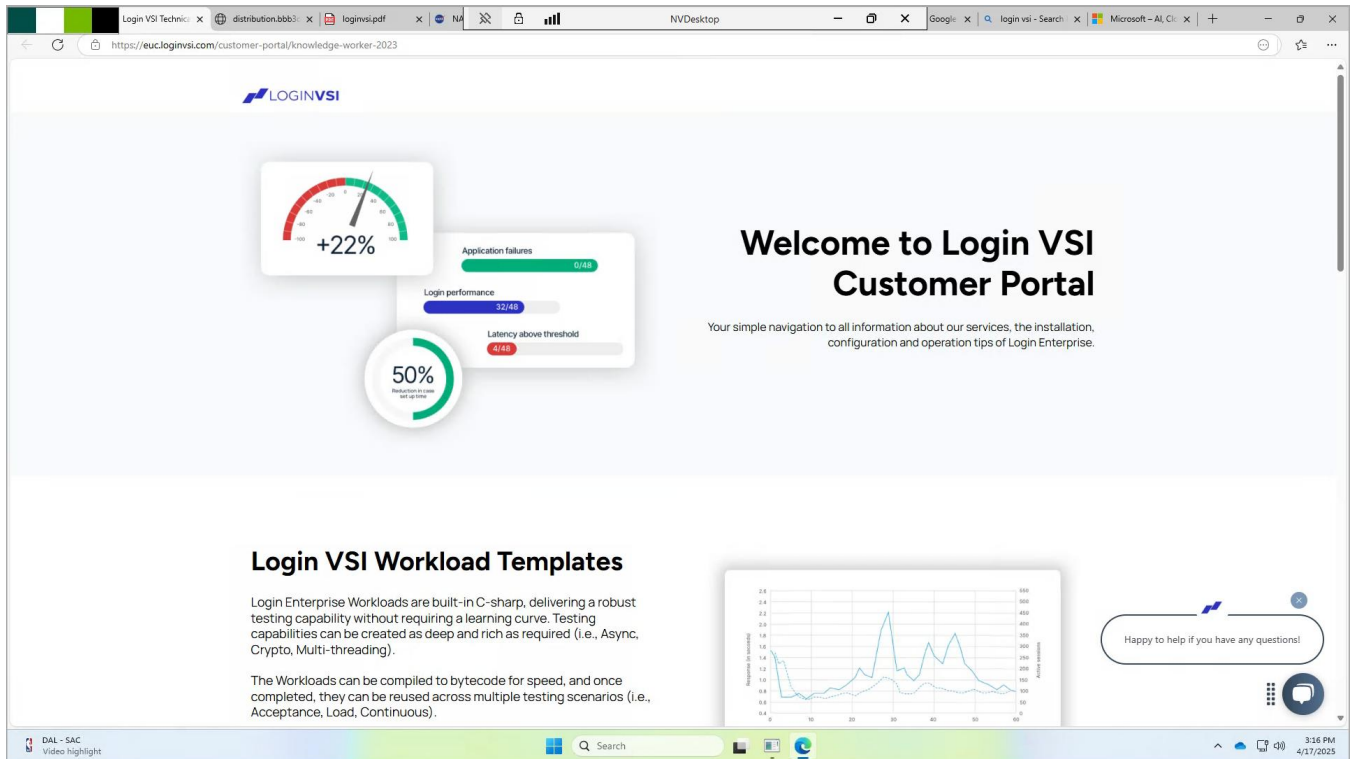
## Testing

- 1) Create a load test. See this [KB article](#) for creating your first load test
  - a. In addition to setting up the general load test here are a few extra tasks to add nVector
    - i. Make sure the connector is set to “Full Screen” mode
    - ii. Add the Environment you created [in the step above](#)
    - iii. Set the session metrics group to the [group you created above](#)
    - iv. In the list of actions add a 30s wait to the beginning of the workload
    - v. After the “Wait” add the nVector\_Desktop\_Prepare application and set it to “Run Once”

ACTIONS		
List of actions		Settings
	Session starts	
	Wait	 30 second(s)
	Run nVector Desktop Prepare	 "Run Once" is enabled
	2022 M365 Knowledge Worker Application group	
	Repeat all actions	

- vi. Feel free to add other applications and actions to your workload
- 2) Create a continuous test. See this [Creating a Continuous Test article](#) for creating your first continuous test.
  - a. A continuous test can be used to monitor a vGPU enabled VM. Apply the same settings as your load test.
- 3) Run test
- 4) To know if the agent is running successfully during the test you should see the nVector watermark on each launcher running the agent. The watermark is a small series of squares that will frequently change if the launcher agent detects the target agent.





The screenshot shows the Login VSI Customer Portal interface. At the top, there's a navigation bar with the Login VSI logo. Below it, a large section titled "Welcome to Login VSI Customer Portal" features a subtitle: "Your simple navigation to all information about our services, the installation, configuration and operation tips of Login Enterprise." To the left of this section are several performance metrics: a gauge showing a +22% increase, a bar chart for "Application failures" with a value of 6/168, a bar chart for "Login performance" with a value of 32/40, and a circular gauge showing a 50% reduction in costs. Below the welcome section, there's a section titled "Login VSI Workload Templates" with a description: "Login Enterprise Workloads are built-in C-sharp, delivering a robust testing capability without requiring a learning curve. Testing capabilities can be created as deep and rich as required (i.e., Async, Crypto, Multi-threading)." and a note: "The Workloads can be compiled to bytecode for speed, and once completed, they can be reused across multiple testing scenarios (i.e., Acceptance, Load, Continuous)." To the right of this text is a line graph showing "Response time (seconds)" and "Active sessions" over time. At the bottom right, there's a chat bubble that says "Happy to help if you have any questions!" and a "DAL - SAC Video highlight" video player.

# Reporting

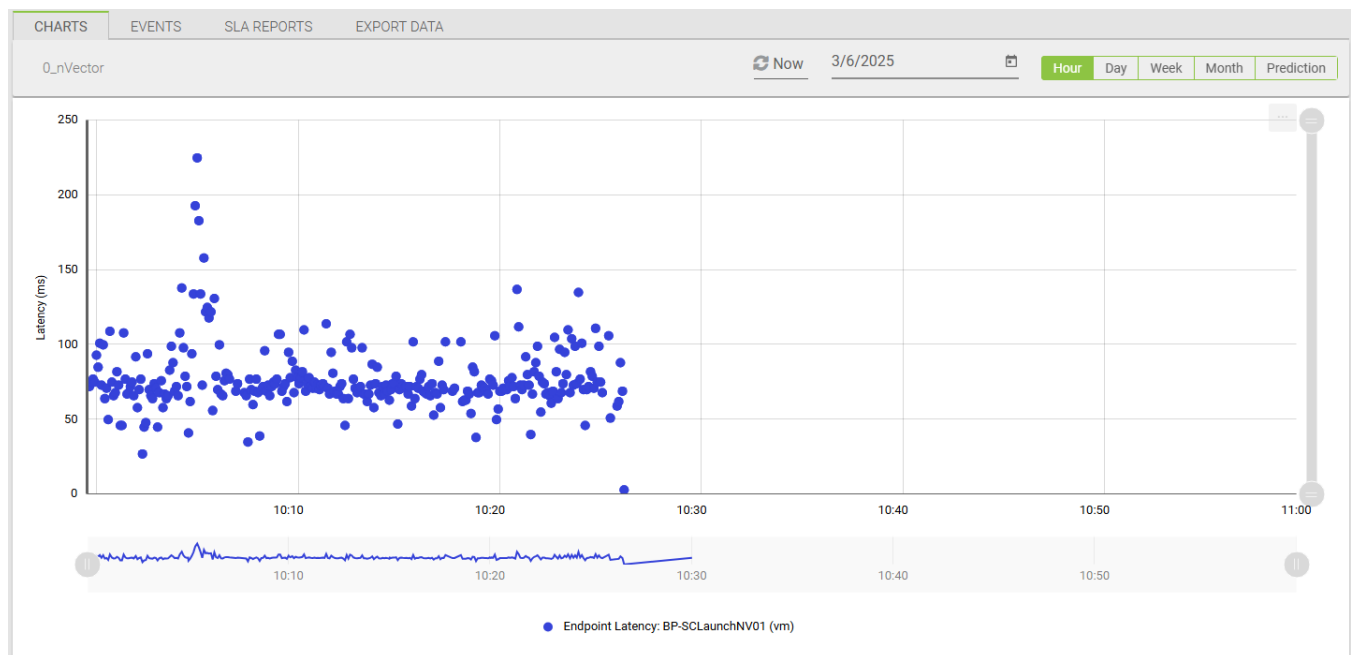
## Examples of Latency Results in Continuous Testing Results

Note: you can watch real-time test results of a load test using a continuous test view if they both use the same “environment” ID in their test configuration settings.

### Hourly View

#### Screenshot

The following example graph visualizes nVector latency measurements over an hourly timeframe in the Login Enterprise web interface. (depending on LE version, you may have to turn on Platform Metrics with a feature flag in the browser)



#### Explanation of Graph Elements:

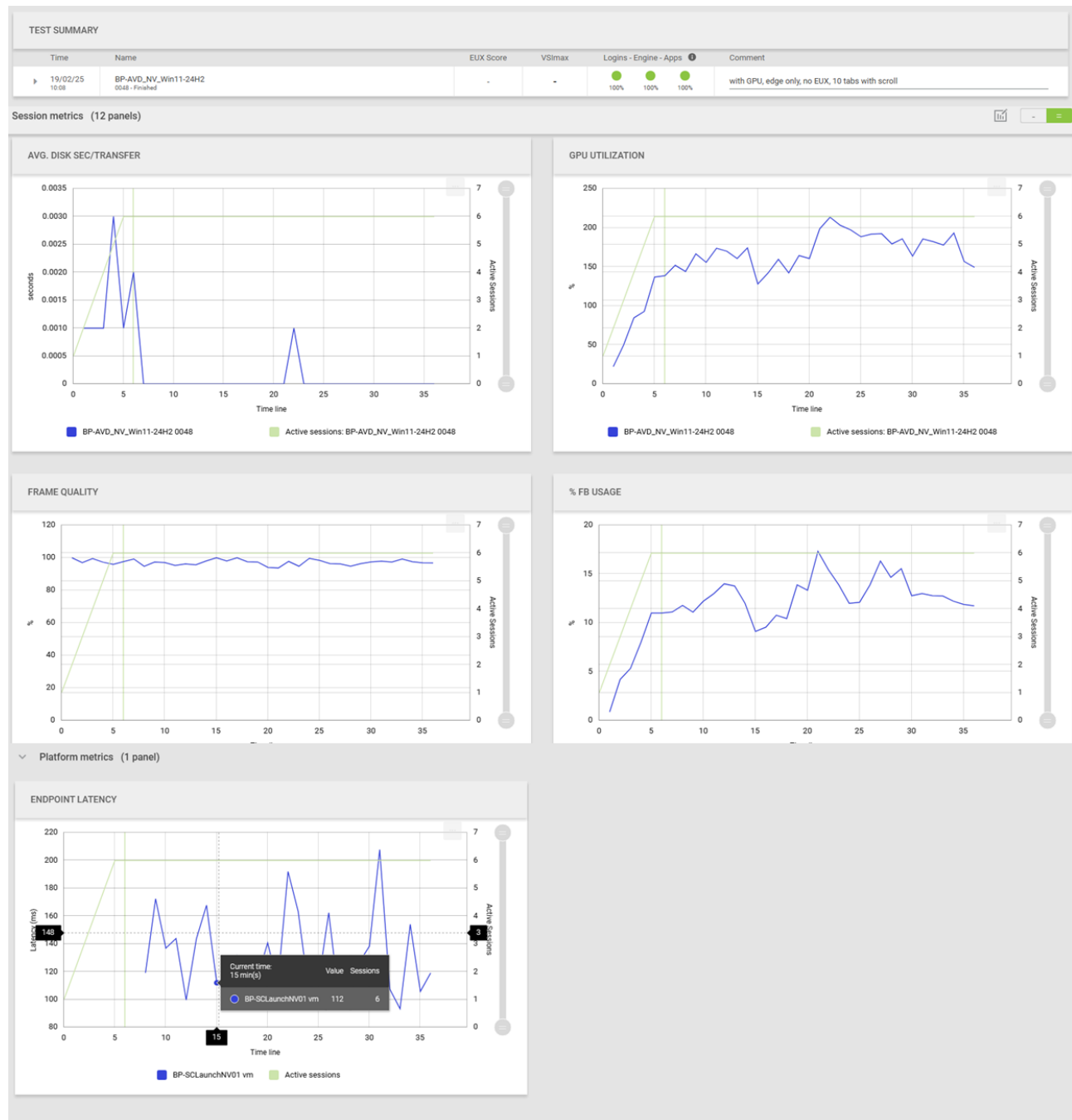
- **Y-Axis (Latency in ms)** → Each **datapoint** represents a **latency measurement in milliseconds**.
- **X-Axis (Time in Hourly Span)** → Represents the **time range** for **nVector latency measurements**.
- **"Latency (ms)"** → This corresponds with the **unit key** in the API datapoint.
- **"Endpoint Latency"** → This corresponds with the **displayName key** in the API datapoint.
- **Hostname (e.g., BP-SCLaunchNV01)** → This corresponds with the **instance key** in the API datapoint.

- **"vm" (Component Type)** → This corresponds with the **componentType** key in the API datapoint.

These labels align **directly with the API response structure** and are displayed in the **Login Enterprise web interface graphs**.

## Examples of Latency Results in Load Testing Results

To see load test results you must wait for the load test to complete. You can compare load tests. You can create a PDF report of the results as well.







## Best Practices

- 1) Best practices
  - a. It is recommended to use only one session per launcher. If you would like to run a large load test and only have a limited number of launchers please contact support and we can help you with that.
  - b. Always run sessions in full screen on the launcher
- 2) You can use an Application test in Login Enterprise with a Desktop connector to make sure the virtual user successfully downloads and runs nvector-agent.

## Appendix A – Script Overview

### nVector\_Client\_Prepare.ps1 (for Login Enterprise Launcher machine)

#### Purpose

- Ensures the **Login Enterprise Launcher** process is running and starts it if it's not already active.
- Invokes nvector-agent.exe with the **client role** to measure round-trip latency.
- Terminates any running instances of nvector-agent.exe before restarting it with user-defined parameters.
- Monitors a CSV file for new latency results and uploads them to the Login Enterprise API as **Platform Metrics**.

#### Key Variables & Parameters

##### *Required settings*

- **CsvFilePath** - Path to the CSV file where the nvector-agent writes latency metrics.
- **NvectorScreenshotDir** - Directory for screenshots (if required by nVector).
- **NvectorLogFile** - Log file path for the nvector-agent.
- **TranscriptFile** - Path for logs produced by this PowerShell script.
- **NvectorAgentExePath** - Full path to the nvector-agent.exe on the Login Enterprise Launcher.
- **ConfigurationAccessToken** - Login Enterprise API token with **Configuration** access level.
- **BaseUrl** - Base URL of the Login Enterprise instance. (e.g., <https://myDomain.LoginEnterprise.com/>)
- **EnvironmentId** - Unique environment identifier in Login Enterprise (obtained from your appliance).

##### *Optional settings*

- **NvectorAgentCheckIntervalMs** - How often (in milliseconds) the nvector-agent checks latency.
- **PollingInterval** - How frequently (in seconds) this script checks the CSV for new entries.
- **MaxLatencyThreshold** - Maximum allowed latency (in ms) before a reading is discarded as spurious.
- **ApiEndpoint** - Endpoint for posting the latency metrics (default: publicApi/v7-preview/platform-metrics).

## Usage Example

### 1. Edit the Script Variables:

At the top of nVector\_Client\_Prepare.ps1, set paths, thresholds, time offsets, and API configuration details to match your environment.

### 2. Invoke the Script:

- Manually via PowerShell:

```
.\nVector_Client_Prepare.ps1
```

- Or automatically through nVector\_Startup\_Script.cmd or a scheduled task.

### 3. Verification:

- Check the PowerShell console or the ScriptLogFile to confirm that:
  - The **Login Enterprise Launcher** process is running.
  - The **nvector-agent.exe** was started in **client mode**.
  - New lines in the CSV are being uploaded to the **Login Enterprise API**.

## nVector\_Startup\_Script.cmd (for Login Enterprise Launcher)

### Purpose

- A simple **Windows CMD script** that invokes nVector\_Client\_Prepare.ps1 on the **Login Enterprise Launcher**.
- Ensures that the **client script** runs automatically at user login without requiring manual execution.
- Can be placed in the **Windows Startup folder** or configured as a **scheduled task**.

### Key Variables and Parameters

- **File Path to PowerShell Script** - Update the CMD file with the correct path to nVector\_Client\_Prepare.ps1 to match its location on the **Login Enterprise Launcher**.

## Usage Example

To automatically start nVector\_Client\_Prepare.ps1 at user login, create the following **CMD script**:

```
powershell.exe -ExecutionPolicy Bypass -File "C:\Path\To\nVector_Client_Prepare.ps1"
```

- Save the CMD file and place it in the Windows Startup folder:  
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

### Automatic Invocation:

- When a user logs in, the CMD file runs and starts the **client integration** automatically.
- Alternatively, configure it as a **Windows Task Scheduler job** for more control over execution timing.

## nVector\_Desktop\_Prepare (C# Workload)

### Purpose

- Runs on the **Target/Desktop** machine in Login Enterprise, often configured as a **workload**.
- Ensures the **nvector-agent.exe** is available by downloading it from the Login Enterprise virtual appliance if necessary.
- Starts nvector-agent.exe with the **desktop role** (-r desktop) to generate a watermark for latency measurement.
- The **client role** nvector-agent.exe running on the Login Enterprise Launcher detects these watermarks to calculate round-trip latency.

### Key Variables & Parameters

- **tempDir**  
Location (commonly %TEMP%) where the agent will be temporarily stored.
- **logFilePath**  
Path for the desktop agent's log file.

### Usage Example

#### 1. Upload the Workload:

- Use the **Login Enterprise web interface** to upload nVector\_Desktop\_Prepare.cs as a **custom workload**.
- Configure the workload settings (e.g., set to **Run Once** if applicable).

#### 2. Adjust Variables as Needed:

- Change logFilePath if needed to specify a different location for the log output.

#### 3. Run the Workload:

- When executed, the workload downloads and runs nvector-agent.exe in **desktop mode** (-r desktop).
- The **client role** nvector-agent.exe on the Login Enterprise Launcher will detect the watermark changes and use them to compute round-trip latency.

## get\_nVectorMetrics.ps1

### Purpose

- Retrieves raw **Platform Metrics** data (including nVector latency) from the Login Enterprise API.
- Queries the `/publicApi/{ApiVersion}/platform-metrics` endpoint (defaults to `v7-preview`).
- Exports results to **CSV** and **JSON** for analysis.
- Optional server certificate import flow to handle self-signed/privately signed appliance certs.

### Key Parameters

Update defaults (paths, URL, token) in the script if needed. The following parameters are supported:

- StartTime, EndTime (Mandatory)
  - Time range for metrics.
  - ISO 8601 Zulu format, e.g. `2025-02-07T00:00:00.000Z`.
- EnvironmentId (Mandatory)
  - Login Enterprise Environment UUID to filter metrics.
- ApiAccessToken (Optional)
  - Overrides the script's default API token.
- BaseUrl (Optional)
  - Overrides the script's default Base URL, e.g. `https://mydomain.LoginEnterprise.com`.
- ApiVersion (Optional)
  - API version segment. Default: `v7-preview`.
  - Final URL format: `https://<BaseUrl>/publicApi/<ApiVersion>/platform-metrics?...`
- OutputCsvFilePath, OutputJsonFilePath (Optional)
  - Output locations. Defaults:
    - CSV: `C:\temp\get_nVectorMetrics.csv`
    - JSON: `C:\temp\get_nVectorMetrics.json`
- LogFilePath (Optional)

- Script log file. Default: `C:\temp\get_nVectorMetrics_Log.txt`.
- ImportServerCert (Optional switch)
  - Fetches the server's leaf + chain certs and imports them into `CurrentUser\Root` before the request.
- KeepCert (Optional switch)
  - Used with `-ImportServerCert`. Keeps imported certs in `CurrentUser\Root` after the run.
  - If omitted, any newly imported certs are removed at the end.

## Security & Compatibility Notes

- **PS 5.x path:** Uses `HttpWebRequest` (certificate validation **not** skipped).  
Use `-ImportServerCert` for appliances with self-signed/private CA certs.
- **PS 7.x path:** Uses `Invoke-RestMethod -SkipCertificateCheck` (validation is **bypassed**).  
Use only on trusted networks. `-ImportServerCert` can still be used to trust the cert for other tools.
- Importing into **CurrentUser\Root** affects trust for the current user only. Use `-KeepCert` sparingly.

## Usage Examples

- 1) Standard (most secure on PS5, default API version)
 

```
.\get_nVectorMetrics.ps1 -StartTime "2025-02-07T00:00:00.000Z" -EndTime "2025-02-07T23:59:59.999Z" -EnvironmentId "abcdef1234" -ApiAccessToken "MY_TOKEN" -BaseUrl "https://mydomain.LoginEnterprise.com"
```
- 2) Handle self-signed cert temporarily (imports then auto-removes)
 

```
.\get_nVectorMetrics.ps1 -StartTime "2025-02-07T00:00:00.000Z" -EndTime "2025-02-07T23:59:59.999Z" -EnvironmentId "abcdef1234" -ApiAccessToken "MY_TOKEN" -BaseUrl "https://appliance.local" -ImportServerCert
```
- 3) Keep imported cert for future sessions
 

```
.\get_nVectorMetrics.ps1 -StartTime "2025-02-07T00:00:00.000Z" -EndTime "2025-02-07T23:59:59.999Z" -EnvironmentId "abcdef1234" -ApiAccessToken "MY_TOKEN" -BaseUrl "https://appliance.local" -ImportServerCert -KeepCert
```
- 4) Use a different API version explicitly
 

```
.\get_nVectorMetrics.ps1 -StartTime "2025-02-07T00:00:00.000Z" -EndTime "2025-02-07T23:59:59.999Z" -EnvironmentId "abcdef1234" -ApiAccessToken "MY_TOKEN" -BaseUrl "https://mydomain.LoginEnterprise.com" -ApiVersion "v7-preview"
```

## Expected Behavior

- Builds the request URL with **System.UriBuilder** for safe path/query encoding.
- Saves the raw **JSON** response to *OutputJsonFilePath*.
- Parses JSON and exports flattened **CSV** rows to *OutputCsvFilePath* with:
  - *timestamp, value, metricId, environmentKey, displayName, unit, instance, group, componentType*.
- Writes detailed progress and errors to *LogFilePath*.
- When *-ImportServerCert* is used without *-KeepCert*, any newly imported certs are removed after execution.

## Troubleshooting

- **Empty CSV / “No parsed JSON to convert to CSV”:**
  - Verify the **time range** contains data and **EnvironmentId** is correct.
  - Confirm the **API token** has sufficient scope and the **BaseUrl** is correct.
- **Certificate errors on PS5:**
  - Re-run with *-ImportServerCert*. If you trust the cert long-term, add *-KeepCert*.

## Additional Information

### Documentation Links

- [Login Enterprise Application Upload](#)
- [Login Enterprise Script Editor](#)