citrix

Optimizing EHR Logon with Citrix

Delivering faster, more consistent EHR access across complex healthcare environments

Citrix Healthcare Solutions

Configuration Guide

Contents

Executive summary

In healthcare, logon performance is a technical metric with clinical consequences. Every clinician's interaction with the Electronic Health Records (EHR) begins with authentication, and delays are repeated dozens of times a shift across thousands of devices. Whether the endpoint is a shared clinical workstation, an exam-room device, a cart, a thin client, or a personal workstation, slow logons erode clinician productivity and shape how the EHR itself is perceived.

Clinicians often report that the EHR takes too long to open, but the root cause is rarely the EHR itself. Most of the wait occurs during logon and session initialization, before the application is available. Profile loading, Group Policy processing, logon scripts, startup applications, and resource contention account for the majority of that time.

This guide targets a specific outcome: To reduce the time between authentication and productive EHR access from a typical 30- to 40-second baseline to under 5 seconds, while maintaining consistency, security, and scalability across the environment.

Experience from Citrix healthcare deployments shows this is achievable through two complementary strategies.

Strategy A: Prepare or reuse the session

Rather than making the logon faster, this approach either prepares sessions and applications in advance or reuses existing ones, so the clinician experiences near-instant access. The following technologies deliver this:

- Citrix DaaS session remote start (prepares in advance)
- Citrix DaaS session pre-launch (prepares in advance)
- Citrix DaaS session lingering (reuses an existing session)
- Third-party tool e.g. Epic Slingshot (prepares in advance)

Strategy B: Optimize the logon process

When pre-staging is not possible, the focus shifts to the logon itself. The primary areas of optimization are:

- User profile loading
- Group Policy processing
- Logon scripts
- Startup applications
- Active Directory dependencies
- Image hygiene and OS configuration
- Citrix DaaS Workspace Environment Management (WEM)

The goal is to remove unnecessary work from the critical path, and to accelerate the work that remains by running it in parallel, so the desktop or application becomes available as quickly as possible.

Understanding the scope of the guide

The guide focuses specifically on Windows and Citrix logon, and is organized into four sections:

- **The logon journey**: Where Windows logon fits within the clinician experience and the phases that occur during session launch.
- **Measurement and diagnosis**: How Citrix DaaS Monitor identifies which phases consume time.

- **Session preparation and reuse strategies**: When to use Citrix DaaS session remote start, Citrix DaaS session pre-launch, Citrix DaaS session lingering, and third-party tools such as Epic Slingshot.
- **Logon optimization strategies**: Image hygiene, Active Directory, Citrix DaaS Profile Management, Group Policy optimization, and Citrix DaaS WEM.

Post-logon topics such as Citrix HDX graphics, audio, webcam, clipboard, and session recording do not materially affect logon time and are covered separately as a post-logon performance initiative.

The logon journey: From endpoint authentication to EHR access

When clinicians describe an EHR as "slow," the delay is often attributed to the application itself. Much of the waiting occurs before the EHR becomes available. Windows logon, Citrix DaaS session initialization, profile loading, policy processing, logon scripts, and startup tasks all execute before the clinician can begin working. Understanding where these activities occur in the workflow is the first step toward reducing logon time and improving the overall user experience.

This guide focuses on the portion of the workflow that typically contributes most to delays: the Windows and Citrix logon process.

The three stages of the clinician logon experience

From the clinician's perspective, accessing the EHR consists of three distinct stages.

Stage 1: Authentication at the endpoint device

The process begins when the clinician interacts with the endpoint device. The authentication experience varies depending on the type of workstation being used.

Shared Clinical (Kiosk) Workstations

A common example is a shared clinical workstation in a workroom or exam room. Two sub-patterns are common. In the first, the device runs under a persistent generic Windows account with the EHR client (for example, Epic Hyperdrive) pre-launched, and clinicians authenticate only to the EHR through tap-and-go. In the second, a badge-driven solution such as Citrix UniconOS signs the clinician into the device, Citrix DaaS StoreFront, and the EHR in a single tap. Endpoints in either pattern are not always Windows-based. In both, the OS logon is kept off the clinician's path, so access remains immediate across rapid user turnover.

Private Workstations

Other environments use workstations assigned to individual users for extended periods. In these scenarios, the user authenticates once to Windows on the endpoint, then launches the EHR and other applications required throughout the shift. Credentials are commonly passed through downstream applications to minimize additional authentication prompts.

Regardless of endpoint type, the next stage is typically the same: establishing a Citrix DaaS session.

Stage 2: Citrix DaaS session launch

Once endpoint authentication is complete, the clinician launches the published application or desktop. This initiates a Citrix DaaS session and begins the most important part of the workflow from a performance perspective.

During session launch, Windows must set up the user's operating system environment before the EHR can be used. Activities commonly performed during this phase include:

- User authentication to the DaaS VDA
- User profile loading
- Group Policy processing
- Printer and drive mapping
- Logon script execution
- Registry and environment customization
- Application startup activities
- Security and compliance checks

Every clinician experiences this process, whether it takes 5 seconds or 40 seconds. Across hundreds or thousands of concurrent users, even small inefficiencies become highly visible. A single misconfigured policy, oversized profile, or slow logon script can significantly affect the user experience.

The remainder of this guide focuses primarily on optimizing this stage.

Stage 3: EHR access

Once the Citrix DaaS session is ready, the EHR application becomes available.

Depending on the deployment model, the clinician may:

- Authenticate explicitly to the EHR
- Use pass-through authentication
- Use badge-tap authentication
- Resume an already running application session

Only after this point can the clinician begin charting, reviewing patient information, placing orders, or performing other clinical tasks.

While EHR optimization remains important, many of the delays that users perceive as poor EHR performance originate in Stage 2 rather than the application itself.

Inside stage 2: The Windows logon process

To optimize logon performance, it is important to understand what Windows is actually doing while the user waits.

The Windows logon process consists of five major phases:

- Sign-in
- Profile Load
- Group Policy
- Scripts
- Startup

These five phases form the framework used throughout this guide.

At the operating system (OS) level, Windows performs seven detailed steps. Several of these steps are grouped into the five phases above because they tend to be analyzed and optimized together.

Phase 1: Sign-in

The sign-in phase consists of credential collection, authentication, and access-token construction.

Step 1: Credential collection

The logon process begins when Windows collects credentials.

winlogon.exe manages the interactive session and hands control to LogonUI.exe. Credential providers collect passwords, smartcard PINs, Windows Hello credentials, or authentication data from third-party identity providers before passing them to the Local Security Authority.

Issues at this stage are uncommon but can occur when poorly performing authentication, or multi-factor authentication providers introduce delays before authentication even begins.

Step 2: Authentication

Authentication runs inside lsass.exe, which selects Kerberos in a domain and falls back to NTLM. It first locates a domain controller through Active Directory Sites and Services, DNS service records, then performs the ticket exchanges that produce all further needed Kerberos tickets. Every exchange is a network round-trip, which is why this is the first place where network distance to the domain controller becomes visible in real time.

Step 3: Access-token construction

With the user authenticated, lsass.exe assembles the access token, and to do that it has to walk every security group the account belongs to, nested groups and security identifier history included. In a large directory that can run to hundreds of groups, which inflates the Kerberos PAC and the token. It earns a closer look when group nesting has grown unchecked.

Phase 2: Profile load

Once authentication completes, the User Profile Service prepares the user's environment.

This includes loading the user's registry hive and making user-specific settings available to the session. The profile approach used by the organization has a significant impact on logon time.

Local profiles

Local profiles reside on the machine itself and typically load quickly because no network transfer is required. However, they are generally unsuitable for non-persistent VDI and shared healthcare environments.

Roaming profiles

Windows copies a **roaming profile** from a central file share during sign-in. The larger and more cluttered the profile becomes, the longer the wait. This is one of the oldest causes of a slow logon. In Citrix and VDI estates, the usual answer is Citrix DaaS Profile Management, which can stream only required files or can mount a virtual hard disk (VHDX) instead of fully copying files. That is a real improvement, but mount time, registry replay, and storage contention during a shift-change logon storm can still add up to seconds.

Mandatory profiles

A **mandatory profile** is a roaming profile that has been deliberately frozen. The administrator takes a customized default profile and publishes it to a network share or puts a copy within the master image. Any change the user makes during the session is held in memory, written to the local profile copy on the DaaS VDA, and discarded the moment they log off.

> Note:
>
> More on the recommendation to use a roaming profile will follow later in this document.

Phase 3: Group Policy

The Group Policy Client (gpsvc.exe) determines which policies apply, filtering by organizational unit, site, security group, and Windows Management Instrumentation (WMI) query, then runs each client-side extension in turn. The slow ones are predictable: drive maps, printer mappings (installing a single printer can cost 10 seconds or more), folder redirection, and software installation. Two design choices turn this into the most common bottleneck:

Synchronous foreground processing. Windows waits for Group Policy processing to finish before it displays the desktop.

Loopback processing. Loopback supports Merge and Replace modes:
Merge requires more processing. User-side extensions run once against the Group Policy objects (GPOs) on the user's own OU, then again against the user settings attached to the host's OU, with the host taking precedence where the two disagree. That second pass is why slow extensions hurt.
Replace drops the user's own GPOs entirely and applies only the host's user settings, so the extensions run once. On a hardened session host, that single set is usually carrying the bulk of the configuration anyway.

Phase 4: Scripts

Logon scripts run at this point, and normally they do so synchronously. Windows waits for each one to return before moving on. A script that maps drives, queries the directory, or reaches out to a server runs each call in sequence. If one of those calls hits a resource that is not responding, the script waits out the full timeout. A single hung mapping with a 30-second timeout can account for most of a 40-second logon on its own.

Phase 5: Startup

With policy and profile out of the way, userinit.exe finishes the environment and launches the shell, either explorer.exe or the published application.
Everything wired to start at logon then fires together: Run keys, the Startup folder, scheduled tasks on logon triggers, and Active Setup on the first run. This is where the agents stack up — security and endpoint detection and response (EDR), OneDrive, Teams, Office add-ins, and monitoring tools — each competing for CPU and disk. On shared hosts, it is also where logon storms do the most damage. When an entire shift signs in within a few minutes, the host saturates and every logon slows, even when no individual machine is misconfigured.

The logon timeline - where do 40+ seconds come from?

Across healthcare environments, the majority of logon time is typically concentrated in:

- **Group Policy processing** leads, especially when it is synchronous and carries slow extensions like printer and drive mapping.

- **Logon scripts** come next, usually when one is blocked on a resource that is slow or simply gone.
- **Profile load** extends the delay, driven primarily by oversized roaming profiles or container mount and contention if hosted on a slow storage repository. It is important to ensure that the storage systems are not creating bottlenecks.
- **Startup applications and logon-storm contention** take much of the rest on shared hosts.
- **Authentication and token construction** trail the list and only become the story when the directory or DNS is unhealthy.

Network and storage latency amplifies each of these phases, because most of them depend on both.
Because the expensive steps are synchronous, the logon moves only as fast as its slowest blocking dependency, and any unreachable resource becomes a fixed timeout paid on every sign-in.

The timeline widths are illustrative rather than measured, but the shape is the point. Profile Load and Group Policy occur in the middle of the chain and usually account for most of a 40-second logon. Sign-in is normally fast, but climbs when a domain controller is distant, DNS is unhealthy, or the access token is bloated. Scripts are marked moderate because that is the common case, but they are the likeliest source of a sudden spike.

Running these steps synchronously in the foreground degrades the user experience, as the screen shows nothing usable until the last of it clears. That is the mechanism behind a single blocked script swallowing the entire budget. Once the shell launches at the right edge, the clinician can interact, though startup agents continue to load in the background.

.

Figure 1. Logon phases on a time axis. Width approximates elapsed time, color shows severity, and the bracket marks the synchronous foreground chain.

Phase	Typical time	Severity	Usual culprit
Sign-in (auth + token)	~1 to 3 s	Usually fast	Distant or slow DC, DNS, token bloat
Profile load	~5 to 15 s	Major	Profile size and type, container mount and contention

Group Policy	~10 to 20 s	Major (often #1)	Synchronous client-side extensions (CSEs): printers, drive maps, loopback
Logon scripts	~3 to 10 s	Moderate (can spike)	Scripts blocked on unreachable resources
Shell + startup apps	~3 to 8 s	Moderate	Startup agents, logon storms on shared hosts

Indicative timing breakdown. The figures are illustrative, not measured.

Measuring and diagnosing logon performance

Before attempting to optimize logon performance, it is important to understand where the time is actually being spent. Slow logons rarely originate from a single root cause. In most healthcare environments, several components contribute to the overall delay. Practitioners need to determine which phase of the logon process is consuming time before changes can be prioritized and validated.

Start with Citrix DaaS Monitor

For Citrix DaaS Cloud and Citrix DaaS Local environments, Citrix DaaS Monitor is typically the first place to investigate logon performance. Citrix DaaS Monitor provides visibility into logon duration at both an aggregate and individual-session level, allowing administrators to determine whether delays are isolated incidents or systemic issues affecting entire user populations. The overall logon duration displayed in Citrix DaaS Monitor is useful as a high-level indicator, but it should be treated as a starting point rather than a diagnosis.

Understanding the Citrix DaaS Monitor logon breakdown

The logon duration chart, found in the Session Logon tab of the User Details view, can be used to analyze and troubleshoot the logon process. Citrix DaaS Monitor measures the following phases. The Citrix documentation describes each phase in detail:

- **Brokering**: Time taken to assign a desktop to the user.
- **Machine Start-up**: Time taken to start the virtual machine
- **HDX Connection**: Time taken to establish an HDX connection between the client and the virtual machine.
- **Authentication**: Time taken to complete authentication to the remote session.
- **GPOs**: Time taken to apply group policy objects. Note that the GPO drilldown displays CSE processing time only. The total GPO time is retrieved from Event Viewer logs, which can be overwritten depending on memory allocation.
- **Logon scripts**: Time taken for the logon scripts to run.
- **Profile Load**: Time taken to load the user profile. Note that the profile drilldown doesn't include redirected folders, and that some hidden files (e.g. NTUser.dat) may be included even if they are not visible to the user.

Additionally, some hidden files in the AppData folder are not included, and the data may not match the Personalization panel due to Windows limitations.

Note:

If you are leveraging Citrix Experience Insights, the insights on your logon performance can be used in conjunction with data available in Citrix DaaS Monitor to provide a more comprehensive view.

Logon behavior varies by use case, so it pays to look at individual delivery groups rather than trusting the average. A task-worker group and a power-user group can sit a long way apart, and the average quietly hides that gap.

Citrix DaaS Monitor splits each session's logon duration into its component phases, including GPO processing, profile load, and the HDX connection itself. The GPO and profile phases are where most of the useful detail lives. You can drill into them to see which policy is consuming the most time, how large the profile has grown, and how many files are sitting in folders like Desktop and Documents that get pulled at logon.

Once you have that baseline, the result becomes a set of distinct, addressable phases. The sections below take each one in turn.

Mapping Citrix DaaS Monitor to the Windows logon process

One source of confusion for many administrators is that Citrix DaaS Monitor and Windows describe the same process using different terminology.

The five logon phases introduced earlier remain the primary framework used throughout this guide. Citrix DaaS Monitor simply provides additional visibility into the session-delivery activities that occur before Windows begins processing the user's environment.

The colors carry the same severity meaning as in Figure 1.
The purple blocks in Figure 2 mark the phases that only exist once a session-delivery layer sits in front of the OS: Brokering, Machine Start-Up, and HDX Connection.

This is why the same operating system feels heavier through a remote session than at the physical desk. Part of that cost is generic to remoting since even a direct Remote Desktop Protocol connection also performs a protocol handshake. However, Brokering and Machine Start-Up exist only in a brokered environment. Below that line the phases map one to one to the OS view, and the two bottlenecks are the same ones the timeline flagged: Profile Load and Group Policy.

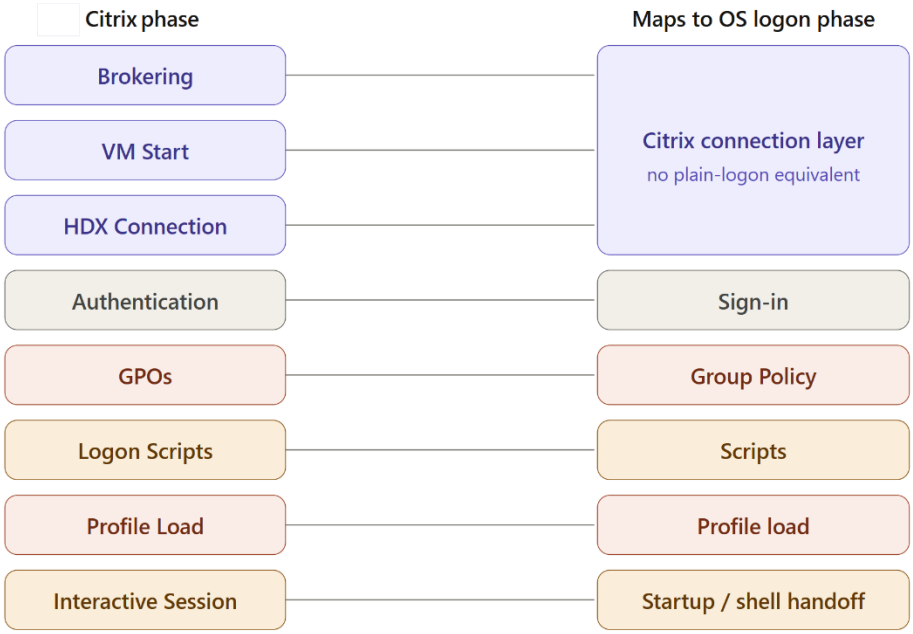


Figure 2. Citrix DaaS Monitor phases on the left, aligned to the OS phases from Figure 1 on the right.

Three things are worth keeping in mind when you read the live logon duration panel in Citrix DaaS Monitor. First, VM Start usually reads zero and only appears when the broker had to power a machine on. Second, the phases do not always add up to the total, because Group Policy, scripts, and profile load overlap in practice. A session might report 12 seconds of GPO, 8 of profile load, and 4 of scripts while total logon reads 18 seconds, because portions of the three run concurrently. Treat Citrix DaaS Monitor's numbers as attribution rather than a strict sum. Third, Authentication here means authenticating the remote session on the DaaS VDA, which is why it sits after HDX Connection rather than at the front.

Interactive Session is the one to watch, because its name hides the most. It covers the handoff to the user, the shell launching, and the startup items, so a high number points at startup applications and a contended host, not at the directory or the profile.

Reducing clinician wait time: two strategies

There are two ways to reduce the time a clinician spends waiting for the EHR:

- Automate the launch through session pre-staging
- Optimize the logon process

The first removes much of the logon process from the clinician's path through session pre-staging. The second focuses on reducing the cost of the logon process itself. We will start with session pre-staging technologies and examine where each approach fits within a healthcare workflow.

Strategy A: Prepare or reuse the session

In a basic user scenario, the clinician opens their workspace, double-clicks the application, and waits for every step of the logon chain to complete before being able to use the EHR system. Several Citrix technologies automate that

launch, so the wait happens before the clinician arrives.

Citrix DaaS session remote start

Citrix DaaS session remote start addresses one of the most common healthcare complaints: a clinician waiting 30 seconds or more before the EHR becomes available. It exposes APIs that let scripts or third-party services enumerate, launch, and log off Citrix DaaS sessions, so a session can be triggered by events like a badge scan at the hospital entrance or a scheduled task at an ambulatory clinic.
By the time the clinician reaches the patient, the session and application are already running.

The result is a near-instant feel at launch, fewer help desk tickets about slow logons, and a cleaner way to keep sessions ready for reconnect without rebuilding state.

Review the Citrix DaaS session remote start documentation for configuration requirements and implementation guidance.

Citrix DaaS session pre-launch

Citrix DaaS session pre-launch silently establishes a blank session in the background. When the clinician opens the EHR, the pre-launched session provides near-instant access. It runs in one of two modes.

Just-in-time fits private workstations: as the clinician signs into the endpoint, Citrix Workspace app begins the session pre-launch in the background.

Scheduled fits kiosks: sessions are pre-launched shortly before a shift change, spreading logon load across the window instead of concentrating it in a five-minute storm.
Either way, the heaviest part of the logon happens before the clinician needs the application.

Review the Citrix DaaS Session Pre-Launch documentation for configuration options.

Citrix DaaS session lingering

Unlike the technologies above, which prepare a session before the clinician arrives, Citrix DaaS session lingering reuses a session the clinician has already used. It keeps recently used app sessions alive in a disconnected state instead of fully logging them off, so a relaunch reconnects to the existing session rather than starting from scratch. The result is faster subsequent access, reduced backend load from repeated logons, and a more responsive experience for frequently used applications.

Epic Slingshot

Powered by the Citrix Workspace app StoreBrowse component, it streamlines Epic access by handling authentication and session launch programmatically in the background. Clinicians land directly in a ready Epic instance and only need application-level authentication (manual or tap-and-go). Slingshot also auto-relaunches Epic if the session was previously closed.
The result is fewer clicks, instant EHR access, and a consistently frictionless experience.

Citrix DaaS session remote start	

Best fit:	• A clear requirement for pre-staged session and application exists - the session must be ready before the user even reaches the device, triggered by a real-world event like a badge tap, a mobile app or a scheduled task. • Ability to integrate with a trusted third-party orchestration layer like badge system (e.g. Imprivata), identity vendor that can call the SRS APIs to enumerate, launch, and log off sessions seamlessly. • Citrix DaaS StoreFront Local is available to pre-launch the sessions. Clinicians can then launch or reconnect through DaaS StoreFront Local, DaaS StoreFront Cloud, or Citrix NetScaler Gateway.
Not best fit:	• No DaaS StoreFront Local is deployed, either because the environment is fully on DaaS StoreFront Cloud or because the customer chooses not to deploy one. • Deploying and managing Citrix DaaS Federated Authentication Service and certificate authority infrastructure is not possible

Citrix DaaS session pre-launch	
Best fit:	• The first application launch of the day is feeling slow because of profile load, GPOs, scripts, agents or environmental configurations. • Connections are from Windows endpoints running the native Citrix Workspace app and configured against a known DaaS StoreFront Local store • Using DaaS Published Applications from a multi-user Windows Server OS
Not best fit:	• Connecting from non-Windows endpoints • When delivering DaaS Published Desktops

Citrix DaaS session lingering	
Best fit:	• Clinical staff repeatedly open and close the EHR or other published applications and need fast access.

	• Applications are delivered from a multi-user Server OS. • Simple solution that doesn't require additional hardware or software
Not best fit:	• Lingering only helps with subsequent launches within the linger window, not the initial session/application launch. • Delivering DaaS Published Desktops, as it is geared to support DaaS Published Apps.

Epic Slingshot	
Best fit:	• Publishing Hyperdrive as a virtual app with the requirement for Epic virtual channel support for Epic integrations (including integrations with locally installed apps on the endpoint such as picture archiving and communication system. • Persistent connection on shared / kiosk workstations where there is a need to automatically relaunch Hyperdrive when it closes and keeps a session waiting at the login screen for the next clinician. • Double-hop architecture for delivery of Epic (VDI → published Hyperdrive) • Use cases with the need for alternative authentication approaches
Not best fit:	• Lingering only helps with subsequent launches within the linger window, not the initial session/application launch. • Delivering DaaS Published Desktops, as it is geared to support DaaS Published Apps.

Example from the field - 1:

A large healthcare provider deployed Citrix DaaS session remote start in their environment to pre-launch sessions by using an automated mechanism. With it, users saw an approximately 85% to 90% improvement in logon experience when accessing their EHR session.

Example from the field - 2:

Another healthcare provider runs Epic on Citrix and rolled out Epic Slingshot alongside Citrix Workspace app to cut friction out of clinical workflows. Sessions stay pre-launched, so a badge tap drops the clinician straight into Epic. The team sized the pool of hot sessions against available capacity, which kept resource contention in check without

sacrificing readiness. Clinicians now reach Epic in under five seconds, satisfaction is up, and the environment still leans on Citrix for the resiliency, security, and flexibility they built the platform on in the first place.

Strategy B: Optimize the logon process

The second path to a faster logon is to shorten the logon itself. That means the configuration, environment, and operational details that determine how quickly a session and its applications come up: profile load, Group Policy processing, scheduled tasks, services, image hygiene. Individually, each looks small. Together, they account for most of what the clinician experiences as waiting time.

Citrix Solution Architects and Citrix Consulting have refined these recommendations across a range of healthcare deployments. The following example shows their cumulative effect on one environment.

Optimizing the operating environment

Before introducing new technologies, organizations should ensure the underlying environment is not introducing avoidable delays. In many engagements, significant improvements are achieved simply by addressing long-standing configuration issues that have accumulated over time.

Antivirus exclusions

Citrix has observed that security tool scanning during logon can lead to longer than desired logon times. You can find actual recommendations in this Citrix TechZone article.
If it is not possible to implement some or all the exclusions, testing with security products temporarily disabled in a controlled validation environment can help isolate whether security tooling is contributing to delays.

Active Directory and Group Policy optimizations

Start with site topology, since it's the piece most people skip. To see which domain controller authenticated the session, run echo %logonserver% at a command prompt. The Domain Controller that authenticates you and the DC that serves your GPOs aren't always the same. Run gpresult /r as well and check the server Group Policy actually came from. If either one is geographically distant, the fix is usually site association: confirm the client's IP subnet is mapped to the correct site in AD Sites and Services, so the client locates and binds to a nearby DC. Getting this right can pull time out of authentication, GPO download and processing, and logon scripts all at once.

Group Policy itself is the other half. Moving settings into Citrix DaaS WEM is one route, but plenty of environments won't want to, and there's real headroom in tuning Group Policy directly before you reach for anything else:

- **Reduce the GPO count**. Consolidate related settings into fewer, well-scoped Group Policy objects. Every linked GPO with applicable settings adds download, parsing, and client-side extension processing.
- **Select the appropriate loopback mode**. Merge evaluates policies linked to the user's organizational unit and then evaluates user settings linked to the host's organizational unit. Replace applies only the host-linked user settings and usually performs better on locked-down session hosts.
- **Use WMI filters sparingly**. Every filtered GPO requires the client to run a query at logon. Prefer organizational unit placement or security-group filtering when either method can meet the requirement.

- **Review drive maps and logon scripts**. Group Policy Preference mappings can slow when they use extensive item-level targeting. Synchronous logon scripts can also block the desktop while they wait for network resources.

Tip:

But even with a lean GPO count, the right loopback mode, and WMI filters under control, you're still bound by synchronous client-side processing at logon. Citrix DaaS WEM shifts that user-side work out of the critical path, applying settings agent-side instead of making the session wait on CSEs.
So tune Group Policy where it sits first, then decide what's worth handing to Citrix DaaS WEM on top.

RunOnce, Active Setup and AutoRuns

RunOnce, Active Setup, and other autorun locations are often treated as small cleanup items, but in a Windows logon they sit in a very sensitive part of the logon chain and could lead to delays.

- **Active Setup** deserves special attention because it is per-user initialization driven from machine-level registration. The components registered under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\Installed Components and HKEY_LOCAL_MACHINE\Wow6432Node\SOFTWARE\Microsoft\Active Setup\Installed Components are evaluated for the user during logon. If Windows determines that a component has not yet run for that user, or that the machine version is newer than the user version, the component's command is executed in the user context.

- **RunOnce** has a similar impact, but for a different reason. RunOnce entries are intended to be executed once, often after software installation, image preparation, upgrade activity, or application registration. In a VDI or published-app image, stale RunOnce entries can become expensive if they are captured in the master image or repeatedly regenerated by applications, agents, or update mechanisms.

- SysInternals **Autoruns tool** is useful in logon analysis because it shows far more than the obvious startup applications. It enumerates the Windows locations where programs, drivers, services, shell extensions, browser components, toolbars, Winlogon notification packages, and other autostart entries are registered. These entries can come from the Startup folder, Run, RunOnce, Active Setup, and many other registry-based or system-level launch points.

File type association

File type associations (FTA) are processed during logon to determine which application handles which file type. If Windows must evaluate, repair, reset, or prompt for default handlers during logon, this adds foreground work before the user can fully interact with the published desktop or application. If the user profile does not retain the expected association state cleanly, Windows may repeat association evaluation or repair work for every new session. For non-persistent DaaS VDAs, mandatory profiles, and pooled multi-session hosts, the effect can be more noticeable. Modifying the FTA for PDF (e.g., via a DefaultAssociations.xml deployed through GPO) pre-sets the association, so Windows doesn't spend time during logon evaluating, prompting, or resetting the PDF handler. This is especially relevant when replacing a built-in handler with a third-party reader like Foxit.

AppX / MSIX

AppX and MSIX packages add measurable work to the logon path because provisioned packages staged in the Windows image are registered per user at first logon. To optimize AppX package load times during Windows logon, consider the following strategies:

- **Remove Unnecessary AppX Packages**: Use PowerShell commands like Get-AppxPackage -AllUsers | Remove-AppxPackage to uninstall unused packages, reducing processing time.
- **Disable First Sign-In Animation**: Disabling this feature can significantly decrease logon delays caused by animations and timed delays.
- **Use Citrix Optimizer**: This tool can streamline AppX-related processes and improve overall logon performance.
- **Optimize User Profiles**: On non-persistent hosts, a full profile container persists work that would otherwise repeat every login like the AppX/MSIX registrations as these are normally discarded at logoff and rebuilt from scratch at a cost of 10 to 15 seconds per session. Folder redirection or a partial profile won't carry that registration, so this saving is specific to a full container that roams the whole profile and includes the registry hive.

Caution:

Because mandatory profiles are read-only, any AppX package that isn't already registered in it after the template was captured, tries to register itself into the user's profile at logon and fails, since there's nowhere to write. Windows retries before giving up, and that alone adds 10 seconds or more to every single login.

Rebuild the mandatory profile template so the current set of packages is registered in it, which means re-capturing whenever the package inventory shifts. Or set the **Allow deployment operations in special profiles** policy, which permits writes in those locations instead of failing.

The table below shows the cumulative effect on a real customer environment.

Example from the field - 3:

Change Set	Configuration	Description	Director Average Time	Reduction in Logon Time
Change Set 1	Baseline Logon Time	No changes made to the environment	100%	
Change Set 2	Control Server + AV exclusions + Citrix Optimizer	Implemented the latest AV exclusions to reduce the impact of security tool scanning.	94%	6%

		Ran the baseline Citrix Optimizer to the image.		
Change Set 3	Citrix DaaS WEM + GPOs	Configured with DaaS Workspace Environment Management, adding a configuration set for the DaaS VDA. Within Citrix DaaS WEM migrated a set of group policies from AD.	81%	19%
Change Set 4	Citrix DaaS WEM + AV Exclusions + Folder Redirection + Citrix Optimizer	Added Folder Redirection for Documents, which was needed as a requirement for some clinicians.	74%	26%
Change Set 5	Citrix DaaS WEM + GPO + Template Profile as Mandatory + Async GPO	Configured Citrix DaaS Profile Management, where a template profile is configured as a mandatory profile.	69%	31%
Change Set 6	Citrix DaaS WEM + GPO + Template Profile as Mandatory + Async GPO + Modified FTA + Deleted specific AppData parts	Optimized File Type Associations by pre-populating into the user profile rather than having to define upon login. Also cleaned up default profile by removing app-specific data that is automatically added from installation but not needed.	65%	35%

Important:

The optimization results can vary for customers following the same part.

Tip from the field:

Review all the optimizations mentioned above as part of image hygiene. Back up the current configuration, disable only what is demonstrably unnecessary for the clinical workflow, and validate the result with before-and-after logon measurements.

Using Citrix DaaS Workspace Environment Management (WEM) to change the logon process

Citrix DaaS WEM provides user environment and resource management for DaaS Cloud and DaaS Local environments. It performs two primary functions.

First, the Citrix DaaS WEM agent applies mappings, registry settings, environment variables, shortcuts, and other user-environment actions based on filters such as Active Directory group, client name, IP range, or time of day. The agent evaluates these settings locally and can move them outside the synchronous logon path.

Second, Citrix DaaS WEM manages CPU, memory, and I/O resources while sessions are running. These capabilities reduce resource contention and help maintain consistent performance on multi-session hosts.

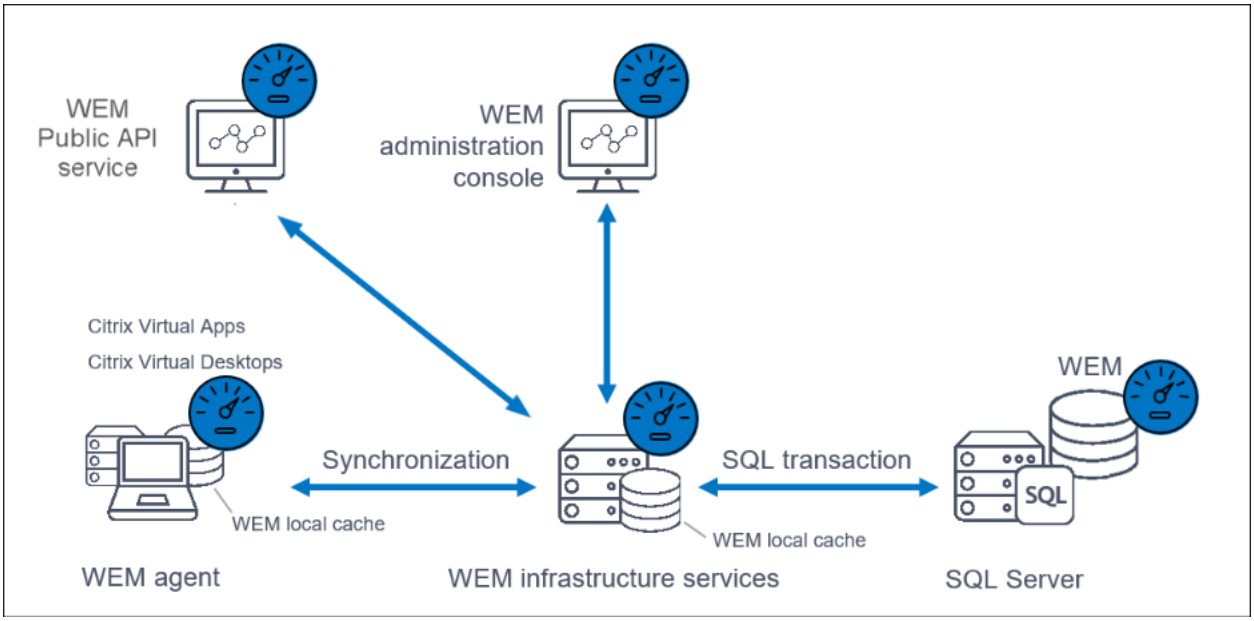


Figure 3. Citrix DaaS Workspace Environment Manager (WEM) architecture.

The control plane is a central console that pushes configuration to a lightweight agent on each DaaS VDA or physical endpoint. The Citrix DaaS WEM agent evaluates settings locally, so there's no round trip to a domain controller in the critical path of the logon. That is where the logon-time wins come from, and it's also why Citrix DaaS WEM scales cleanly into multi-session hosts where density and predictability matter more than per-machine tuning.

Citrix DaaS WEM leads to faster logons and offers a single place to manage the settings that used to live across a dozen GPOs, scripts, and login agents.

Citrix DaaS WEM addresses all phases of the logon process. The map below shows in detail where Citrix DaaS WEM acts and where it does not:

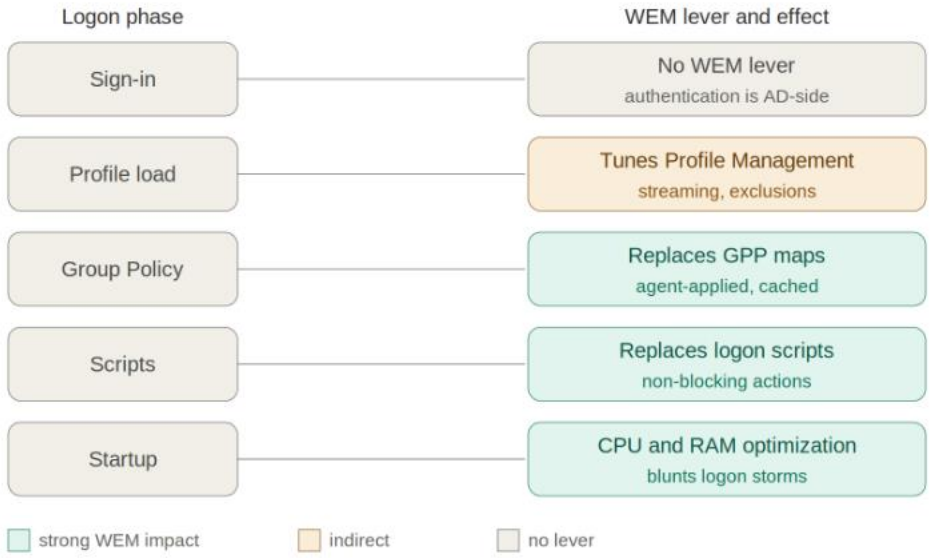


Figure 4. Where Citrix DaaS WEM acts on the logon chain. Color shows the strength of Citrix DaaS WEM's impact at each phase.

Citrix DaaS WEM concentrates on the phases that account for most of the logon duration and leaves the rest to the platforms that actually control them.

Citrix DaaS WEM does not touch authentication, the directory, or DNS, so the first stretch of the chain sits outside its reach. Citrix DaaS WEM's value is concentrated in the phases that consume the most time: Profile Load, Group Policy, Scripts, and Startup. It reaches each of them differently.

Phase	Citrix DaaS WEM lever	How it helps
Sign-in / authentication	None	Citrix DaaS WEM does not change DC location, Kerberos, or token build. Those belong to Active Directory and DNS health, not to Citrix DaaS WEM.
Profile load	Citrix DaaS Profile Management configuration	Centralizes and tunes Citrix DaaS Profile Management (streaming, redirections, exclusions) so the profile that loads is leaner. The mount itself stays with CPM.
Group Policy	Agent-applied settings and maps	Moves drive maps, printer mappings, registry settings, and environment variables out of synchronous GPP and into the Citrix DaaS WEM agent, which works from a local cache instead of live AD round trips.

Logon scripts	Actions and external tasks	Replaces synchronous batch and PowerShell logon scripts with agent actions and external tasks that run without holding up the desktop.
Shell + startup	System Optimization	Citrix DaaS WEM lowers resource usage, ensures unused resources are available, and prevents simultaneous logons from saturating a shared host. This is the core defense against logon storms and overall session degradation by a runaway process.

Citrix DaaS WEM concentrates on the phases that account for most of the logon duration and leaves the rest to the platforms that control them.

Two effects compound in practice:
It shrinks the synchronous foreground window by pulling drive maps, printers, and scripts out of the part of the chain that blocks the desktop, so users reach a usable session sooner.
On shared hosts, it limits resource contention during a shift change, so later logons are not slowed by the resource demand from sessions already starting or running.

Management is also easier. As one console replaces a scatter of GPOs, preference items, and login scripts, which makes the configuration far easier to understand, audit, and change. Citrix DaaS WEM is a valuable level for the phases of logon it owns. It moves the work out of the path the clinician is waiting on, and it keeps a shared host from becoming saturated during a concentrated period of logons.

Citrix DaaS Profile Management

Every time a clinician taps into a session, the user profile gets assembled. Without DaaS Profile Management, using a **roaming profile** means copying the entire profile from the profile share to the DaaS VDA before the desktop becomes interactive. The user waits for the copy to be completed.

The bigger the profile, the longer they wait. There's no exclusion mechanism or granular control to keep the profile from accumulating files that have nothing to do with the workload at hand. In a healthcare environment, where the same clinician may launch sessions dozens of times a day across shared workstations, that overhead compounds in a way that generates friction and dissatisfaction.

DaaS Profile Management (CPM) takes a different approach if **roaming profiles** are used in your environment. Instead of copying the whole profile up front, it copies only what's essential to start the session and streams the rest on demand. The shell appears quickly because application settings load when an application requests them.

The initial logon becomes largely independent of profile size. CPM adds the controls that roaming profiles never had: exclusion policies to prevent bloat, granular configuration through Citrix DaaS WEM, and built-in handling for the merge conflicts that occur when the same user runs multiple sessions simultaneously.

Using profile containers represents the next step up from roaming profiles. Instead of synchronizing files between the user store and the DaaS VDA, DaaS Profile Management stores the entire profile in a VHDX file on a network share. When the user logs on, the OS mounts that VHDX as if it were a local disk. The applications read and write to it in real time over the network. Windows does not copy the full profile at logon. Mounting the container makes the profile data available to the session.

If roaming profiles are not an option, **mandatory profiles** come into play. Citrix recommends to use a mandatory profile in healthcare, as user settings don't need to persist across sessions. Epic stores its user preferences server-side, not in the Windows profile, so no profile needs to be synchronized, merged, or stored. The profile that loads today is the same as yesterday. CPM has a documented method for configuring a mandatory profile through its template mechanism, which keeps you inside the Citrix-managed model rather than relying on legacy Active Directory mandatory profile plumbing.

Optimizing the Group Policy preferences

Moving GPPs to Citrix DaaS WEM policies isn't a swap of mechanisms. It's a change in when and how the user environment gets built. The Citrix DaaS WEM Group Policy Migration Tool converts existing GPOs and preferences into Citrix DaaS WEM actions, so modernization isn't a manual rebuild.

GPPs run inside the synchronous logon pipeline. Citrix DaaS WEM moves that work off the critical path. The Citrix DaaS WEM agent executes actions after authentication is completed and processed asynchronously, so the desktop renders while the environment continues to configure in the background.

GPPs build the environment before the user can work. Citrix DaaS WEM builds it while the user is already productive.

Example from the field - 4:

The figures below are a striking example of how migrating to GPPs helps reduce login times:

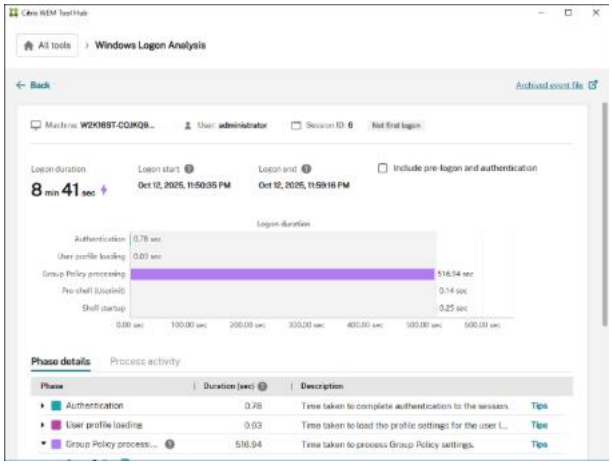
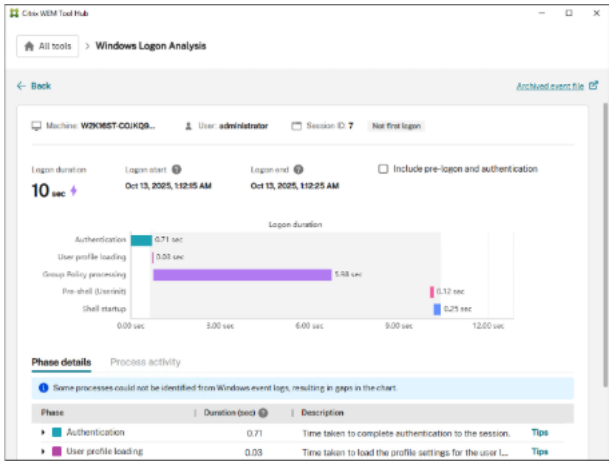
Before GPP migration to Citrix DaaS WEM	After GPP migration to Citrix DaaS WEM
	

Figure 5. Before and after GPP migration to Citrix DaaS WEM

Learn more on how to migrate settings from Group Policy to Citrix DaaS WEM in the following Citrix documentation.

Optimizing logon scripts

Citrix DaaS WEM doesn't make logon scripts run faster. It stops them from running while the user is waiting. In a traditional setup, batch and PowerShell scripts execute inside the synchronous sign-in pipeline. Citrix DaaS WEM

lifts that work out of the critical path by re-registering it as External Tasks, which the agent runs in the user's session after sign-in is complete, with full access to profile and network context.

Conditional drive mappings, printer assignments, shortcuts, and registry tweaks move into native Citrix DaaS WEM Actions with filter rules.

The result is measurable rather than asserted. The Windows Logon Analysis tool in the Citrix Environment Tool Hub gives you before-and-after timings on the logon-script phase, so the value of each migration step can be quantified in seconds saved.

For step-by-step configuration of drive mappings, printer assignments, shortcuts, and registry actions in Citrix DaaS WEM, see the Citrix documentation on Citrix DaaS WEM Actions. Drive mapping specifically is covered under Network Drives.

Optimizing startup and shell

Citrix DaaS WEM's system optimization capabilities are designed to lower resource usage on the machine. They make sure that freed-up resources are available for other applications and increase user density by supporting more users per server.

Five different settings help with getting the best optimizations for using EHR:

CPU management

Processes can run across all cores and can use up as much CPU as they want. In Citrix DaaS WEM, the CPU management feature lets you limit how much CPU capacity individual processes can use.

- **CPU spike protection** is designed to reduce the impact on user experience by processes that consume an excessive percentage of CPU usage.
  CPU spike protection examines each process through a brief snapshot. If a process reaches a specified threshold, Citrix DaaS WEM automatically lowers the priority of the process for a certain time. If the average load of a process exceeds the specified usage limit for a specified sample time, its priority reduces immediately.
  After a specified time, the process' CPU priority returns to its previous value.
- **CPU priority** optimizes the CPU priority level of specific processes, so that processes that are contending for CPU processor time do not cause performance bottlenecks. When processes compete, processes with lower priority are served after other processes with a higher priority.
- **CPU affinity** defines how many "logical processors" a process uses.

Step-by-step configuration for each of the settings below is covered in the Citrix documentation on Citrix DaaS WEM System Optimization.

Memory management

Most applications request and hold onto memory they don't actively use. On a multi-session host, that leads into silent contention. Citrix DaaS WEM changes that behavior. It continuously scans for idle processes and forces them to release unused memory back to the operating system, without closing the application or interrupting the user. The reclaimed memory immediately becomes available to active sessions on the host and reduces the overall footprint of each session.

Review the Citrix DaaS WEM system optimization documentation to configure memory management.

I/O management

At its core, Citrix DaaS WEM treats disk and network I/O like a shared, finite resource that needs active arbitration. By default, every process competes with equal resource needs. In a multi-session environment, a single background task can monopolize the storage subsystem and degrade every active session on that host. Citrix DaaS WEM assigns I/O priority based on process activity: The process the user is interacting with gets precedence, while everything else is pushed out of the way.

Learn more on how to configure I/O management in the Citrix documentation on Citrix DaaS WEM System Optimization.

Citrix Optimizer

Citrix Optimizer is the part of the Citrix DaaS WEM stack that does its work before anyone logs on. It leans the OS by removing unneeded Windows components, telemetry processes not needed for a clinical workload, scheduled tasks that do not support the clinical workload, Universal Windows Platform apps that the clinical workload does not require. All of these carry a cost: CPU cycles, memory, disk I/O, and time during session initialization. Citrix Optimizer runs a quick scan of user environments and then applies template-based optimization recommendations.

Review the Citrix Optimizer documentation.

Multi-session optimization

Multi-session optimization in Citrix DaaS WEM addresses a problem that quietly dominates EHR environments: the disconnected session that nobody is using, but which still ties up server resources. Applications keep running, memory stays allocated, background processes continue to consume CPU and disk shares. After a session has remained disconnected for one minute, Citrix DaaS WEM lowers the CPU and I/O priorities of every process in that session. Citrix DaaS WEM imposes strict memory limits and reclaims the headroom for active users. The moment the EHR user reconnects, the original priorities are restored.

Review the Citrix DaaS WEM multi-session optimization documentation.

Summary

In healthcare, clinicians rarely judge technology by architectural decisions, infrastructure design, or optimization techniques. They judge it by responsiveness. Every interaction with the EHR begins with authentication and application access, making logon performance one of the most visible measures of the overall user experience. A clinician who can access the EHR immediately can focus on patient care. A clinician who repeatedly waits through logon delays experiences friction from the very first interaction of the workflow.

While users often attribute delays to the EHR itself, the majority of waiting typically occurs before the application is fully available. Authentication, profile loading, Group Policy processing, logon scripts, startup activities, and session-delivery overhead all contribute to the time between user authentication and productive access. Understanding where that time is spent is the foundation of every successful optimization initiative.

Healthcare organizations generally have two approaches available to improve their experience.

The first approach prepares sessions in advance or reuses existing sessions. Citrix DaaS session remote start and DaaS session pre-launch prepare sessions before clinicians need them. Citrix DaaS session lingering reconnects clinicians to existing sessions. Third-party tools such as Epic Slingshot can further streamline Epic-specific workflows.

The second approach is logon optimization. This focuses on reducing the amount of work Windows and Citrix must perform during session launch. Improvements to Active Directory configuration, profile management, Group Policy design, startup processing, image hygiene, antivirus configuration, and operating system optimization all contribute to a shorter and more consistent logon experience. Citrix DaaS WEM further enhances this process by moving many traditionally synchronous operations outside the critical path, allowing users to become productive sooner while the environment continues to configure in the background.

Effective logon optimization combines both approaches. Session preparation and reuse improve perceived performance by reducing the time clinicians spend waiting. Logon optimization improves actual performance by reducing or parallelizing the work required to establish the session.

Administrators should begin with Citrix DaaS Monitor and Citrix Experience Insights to identify the phases contributing to the delay. They can then select the appropriate combination of session preparation, profile management, Group Policy optimization, image hygiene, and Workspace Environment Management.

The objective is to create a predictable clinical workflow across shared workstations, carts, thin clients, and private workstations. Clinicians should be able to authenticate, access the EHR, and begin patient care without avoidable delay.