Session Configuration

Session Settings, Application Mode, Frame Remoting Protocol, Session

- Session Settings
- Application Mode
- Frame Remoting Protocol

Session Settings

You can customize many aspects of how your users' sessions will behave when accessing their Frame session. The session settings page in your Account Dashboard was created to give you fine granularity over these features.

Account Session Settings

For Frame Account Session Settings, navigate to the “Settings” page listed on the left side of your Dashboard. Click on the "Session" tab. Any changes you make on this page will apply to all production/test instances, Utility servers, and your Sandbox.

image.png

Account Session Settings

> You may specify a separate set of session settings for your Sandbox/Utility server or test/production instances by clicking on the ellipsis next to your Sandbox (Dashboard>Sandbox), Utility Server (Dashboard>Utility Servers) or Launchpad (Dashboard>Launchpads), clicking on "Session settings," and disabling the "Use account settings" toggle. Edit this page as desired and click "Save" to apply your changes.

Features

image.png

Session Settings - Features

Feature	Description
Clipboard Integration	Enables clipboard functionality. Users can cut and paste text between their local device and their Frame session.

Feature	Description
Clipboard Direction	When the clipboard integration is enabled, use this drop-down menu to choose the clipboard direction policy for your users.
App switching	When **App switching** is enabled, end users can switch between two or more application windows by typing `Alt` + `~` (`Alt` + `~` is sent to the remote Windows VM as `Alt` + `Tab`).
Download	Enables downloading files from the remote session to the user's local device.
Upload	Enables uploading files from the user's local device to their Frame session.
Print	Enables printing files from the remote session to the Frame Virtual Printer.
Camera	Enables webcam support for sessions.*
Microphone	Enables audio input when using applications within the session. This feature is enhanced for sessions using FRP8.
USB Redirection	Enables locally-connected USB devices to be visible to the remote workload VM. USB-connected storage devices are automatically detected and populated as additional drives which are accessible from the file explorer.\\
FRP8	Enables Frame Remoting Protocol 8
4K Displays	Enables resolutions up to 4K (4096x2160) for CPU-only instances (GPU instances support up to 4K displays by default). Note: CPU-based encoding of 4K displays will require increased vCPU capacity (e.g., 4 vCPUs instead of 2 vCPUs). **Note: This feature should be enabled in scenarios where users will be accessing a CPU-only workload VM from ultra-wide or non-4k high resolution monitors.**
Audio playback on session start	This setting ensures audio playback is enabled on session start but can still be disabled within the session by the user if desired. Enabled by default.

* = Requires Frame Remoting Protocol 8 to be enabled.

\\ = Requires Frame Remoting Protocol 8 and Frame App 6.3 or above for Windows only.

Additional Documentation

Additional details for these features can be found in our Session Features End User Guide and other sections of our official Frame documentation.

Time Limits

The "Time Limits" section displays parameters which control how long sessions can run. See the corresponding sections below to learn more about each parameter.

image.png

Session Settings - Time Limits

User Inactivity Timeout

This is the maximum amount of time (in minutes) that Frame will keep a session connected when there is no user activity (no mouse/keyboard events). Frame will display a warning at the “1 minute left” mark (configurable) and then disconnect the session.

- **Default value**: 10 minutes
- **Minimum value**: 1 minute

Idle Timeout

For any sessions that are launched from the Launchpad, authenticated users can disconnect from a session and reconnect later to the same running session. The Idle Timeout setting refers to the amount of time (in minutes) that a session will be kept active after an authenticated user disconnects from the session by closing Frame App or their browser window (while in session), disconnecting from the Frame gear menu, or by getting disconnected due to a network issue. Once the Idle Timeout is reached and the user has not resumed their session, the session will be closed.

If the Idle Timeout value is set to 0, then when a Frame session is disconnected, the session will be closed immediately.

- **Default value**: 10 minutes
- **Minimum value**: 0 minutes

Idle Timeout and User Inactivity Timeout session setting values cannot exceed the maxmium value of the Max Session Duration as described below.

Max Session Duration

This is the maximum length (in minutes) of time that a session can run. The duration is shown on the status bar countdown timer in the session itself.

- **Default value**: 1 hour
- **Minimum value**: 1 minute
- **Maximum value**: 10,799 minutes

Max Session Duration timeout is now an optional setting and can now be enabled or disabled with the **Enable Max Session Duration** switch. By default, this feature is disabled for all new Accounts. When enabled, administrators can set a maximum duration for all sessions, after which the sessions will automatically close.

Reservation Timeout

The Reservation Timeout parameter refers to the amount of time (in seconds) that the client's browser will wait for a server to become available before displaying a timeout error. This value would typically be adjusted to accommodate slow-starting virtual machines. This timeout is less likely to be reached if a min or buffer is configured.

- **Default value**: 600 seconds
- **Minimum value**: 120 seconds
- **Max value**: 900 seconds

Session Preparation Timeout

This value specifies how long Frame will wait for session initialization to complete before automatically closing the session. This includes the time waiting for a user to log into their Windows domain in domain-joined Frame accounts. Admins can configure this value between 15 and 60 minutes (in 15 minute increments).

- **Default value**: 15 minutes
- **Minimum value**: 15 minutes
- **Max value**: 60 minutes

Network

The "Network" section of the Session Settings page is where you can set network and QoS settings for your users. Some organizations manage groups of users in remote areas which have limited bandwidth, high latency, and often encounter varying network conditions. The Frame Remoting Protocol responds to such circumstances by rapidly adjusting the visual properties, frame rate, image quality, and other key aspects to maintain a consistent user experience. Each

of the variables listed below applies to a different aspect of the session's QoS settings:

image.png

Session Settings - Network

- **Max Frame Rate**: Sets the maximum frame rate for a session. The frame rate is defined as the number of frames displayed per second.
- **Max Video Bit Rate**: Adjusts the maximum bandwidth (in Mbps) to be used for the session display.
- **Enable YUV444**: Use YUV444 encoding instead of YUV420 encoding.
- **Allow Users to Change These Settings**: Allows your users to define their own QoS settings from within their session by accessing the Frame gear menu.

Network QoS Settings Defined

Frame automatically adjusts the video frame rate in response to application activity and available bandwidth. Under normal circumstances, the default frame rate is 20 frames per second (fps). Limiting the maximum frame rate can reduce bandwidth requirements, but may cause choppiness and can make interactive editing tasks difficult. Administrators can set the maximum frame rate for production sessions from the Dashboard. If enabled by the admin, end users can adjust the frame rate of their session as they see fit.

Supported Ranges

- **GPU-enabled Instances**: 5 - 60 fps
- **CPU-only Instances**: 5 - 30 fps

Frame limits the maximum bandwidth to 32 Mbps. Lowering the bandwidth limits the overall bandwidth available to Frame, reducing both frame rate and image quality.

Supported Ranges

- **Video Bandwidth**: 256 kbps - 32 Mbps

By default, Frame's H.264 implementation uses YUV420 chroma subsampling to encode images. This takes advantage of the human eye's inability to recognize color differences to the same degree that it can recognize variations in brightness. By sending less information about color than it does about brightness, it is possible to reduce the amount of bandwidth required substantially without significantly compromising image quality. This does an excellent job of reducing the amount of bandwidth required, but in some situations, especially in apps where regions of strongly contrasting colors are displayed next to each other, chroma subsampling can result in colors “bleeding” into each other with undesirable results.

To support our customers who need absolute color fidelity, Frame also provides support for YUV444 encoding. This turns off chroma subsampling, sending the full depth of color information for every pixel. Since more color information must be sent, use of YUV444 will increase the required bandwidth between the remote VM and the end user.

> YUV444 is not supported on mobile web browsers.

End users accessing a YUV444-enabled session will notice HQ in the status bar

image.png

Keyboard Profiles

Keyboard profiles consist of custom keyboard shortcuts and language settings that allow you to map endpoint keyboard combinations to keyboard combinations within the remote VM. Refer to the Keyboard Profile section of our documentation for details on how to manage your end users' keyboard profiles.

Advanced Options

The Advanced Options section of Session Settings displays two editable fields, *Advanced Terminal Arguments* and *Advanced Server Arguments*. Here, you can enter configuration flags that will either affect the behavior of Frame Terminal or Frame Agent during a session. Advanced configuration flags should be separated using a space. Be sure to click **Save** to save your configurations.

Session Settings - Advanced Options

Terminal Build Version

At the bottom of the *Advanced Options* section, you will notice a dropdown menu listed under a section titled *Terminal Build*. This feature allows administrators to dictate which Terminal build version they would like any new sessions to utilize for the account they are accessing.

Terminal Build Version - Advanced Options

This setting can be adjusted on a per-Launchpad basis as well by navigating to the desired Launchpad from the Account Dashboard, clicking the kebab menu, and selecting *Session Settings*. The options are as follows:

- **Current**: The current GA (Generally Available) version of Frame Terminal.
- **Previous**: The previous GA version.
- **Next Gen**: Our next-generation session experience which features a sleek, modern interface with enhanced menus and controls. Now available in **Early Access**.

Advanced Arguments

The following flags can be used in either the Advanced Terminal Arguments (T) or Advanced Server Arguments (S) fields to modify the behavior of Frame Terminal or session behavior, associated with the Frame Account or Launchpad. These arguments are supported in both Frame Remoting Protocol (FRP) 7 and 8 unless otherwise noted.

Argument	Type	Syntax	Details
Enable Sound on First Interaction	Terminal	`enableSoundOnFirstInteraction`	Automatically enables sound in the session upon first user interaction (mouse click or move) for embedded scenarios.
Disable Onboarding Dialog	Terminal	`disableOnboardDialog`	Disables the initial welcome message for new users.
Enable Mouse Modes	Terminal	`*features*mouseModes*isEnabled*=true`	Enables Frame Terminal to display to the user the Mouse Mode selection icon for standard, relative, and touchpad mouse modes.
Frame App Update Indicator	Terminal	`appAutoUpdateNotification`	Use this argument to show or not show the update indicator in Frame App. Value of 'always' will display indicator whenever user is using a version that is not the newest. Value of 'old' will display the indicator if user's Frame App is a version more than 1 year old.

Argument	Type	Syntax	Details
Adjust Upload Chunk Size	Terminal	`channelFileChunkSizeBytes`	Use this argument to adjust the FRP8 upload chunk size when uploading files from Frame Terminal to the workload VM. Default value is 4096 bytes.
Disable Bandwidth Indicator	Terminal	`disableBandwidth`	Use this argument to disable the Network Bandwidth Indicator on the Frame Terminal Status Bar.
Disable Local Timezone	Terminal	`disableClockSync`	Use this argument to prevent the timezone in the user's endpoint from being used within the workload VM at the startup of a Frame session.
Disable Network Latency Indicator	Terminal	`disableLatency`	Use this argument to hide the Network Latency Indicator on the Frame Terminal Status Bar.
Enable Autofocus	Terminal	`enableAutoFocus`	Use this argument to automatically set keyboard focus to the Frame session without requiring initial mouse input within the Frame session window.
Force Lossless Video Quality	Terminal	`forceLosslessVideoQuality`	Use this argument to enable Lossless encoding (requires YUV444 to be enabled).
Force Display Resolution	Terminal	`forceResolution=<width>x<height>`	Use this argument to require Frame Terminal to display in a preset resolution (width: 1024/1280/1920, height: 768/1024/1080).

Argument	Type	Syntax	Details
Hide Status Bar on Full Screen	Terminal	`hideStatusBarOnFullscreen`	Use this argument to hide the Frame Terminal Status Bar when Frame Terminal is in full-screen mode.
Set Max Session Timeout Warning	Terminal	`lastMinutesWarningTimeout=<minutes>`	Use this argument to specify the number of minutes left in a session before the max session timeout warning is displayed.
Use TCP by Default for FRP8	Terminal	`preferredIceCandidateProtocol=tcp`	Force FRP8 to always use TCP instead of the default UDP protocol.
Refresh Stream on Packet Loss	Terminal	`refreshStreamOnPacketsLoss=<n>/<t>/<b>`	Automatically refresh video stream when packet loss threshold is exceeded. n=packets lost, t=timeframe(ms), b=backoff factor.
Enable Automatic Audio Playback	Terminal	`unmuteAudio`	Enable automatic audio playback in supported browsers without requiring users to unmute audio in the Frame Terminal Bar.
Scan Input Media Device Once	Terminal	`scanInputMediaDeviceOnce`	Limits camera and microphone scanning to once per session, improving compatibility with features like Apple's Continuity Camera.
Set Disk Attach Timeout	Server	`-diskattachtimeoutms <timeout in ms>`	Sets profile disk/personal drive attach timeout value. Default is 90000 ms (90 sec).
Force CPU Encoding	Server	`-encode <encoder>`	Force CPU encoding if GPU driver isn't working correctly. Use '`-encode ffcpu`' or '`-encode dxgigpu`'.

Argument	Type	Syntax	Details
Set Frame Session Label	Server	`-frame-session-label=<string>`	Set custom value accessible from `FRAME_SESSION_LABEL` environment or registry.
Set Key Frame Frequency	Server	`-gopsize <N>`	Set key frame frequency for video refresh. Default: 240 for FRP7, infinite for FRP8.
Set Link Handler	Server	`-linkhandler <browser>`	Designates which browser is used to launch URLs within the Frame environment.
Set Logoff Timeout	Server	`-logofftimeoutms <timeout in ms>`	Set Windows user logoff operation timeout. Default is 10000 ms.
Refresh Stream on Packet Loss	Server	`-refreshStreamOnPacketLossPercent <loss %>`	Send periodic key frames when packet loss exceeds specified percentage.
Set Trailing Display Frames	Server	`-trailingFramesNumber <#>`	Override default trailing display frames (2 for CPU, 8 for GPU encoding).
Set Upload Folder Path	Server	`-uploadto <upload_path>`	Designate custom upload folder path. Default: `C:\\Users\\<username>\\Uploads`.

> Users may have to close their existing session and start a new session in order to see any newly applied session settings.

Frequently Asked Questions

How does YUV444 improve the end user experience? What are the limitations?

If the user is working with 2D or 3D design, YUV444 may provide sharper text and shapes. If the text or shapes appear blurry when using the standard YUV420, enable YUV444 and see if

there are any visual improvements. This feature requires up to 50% more bandwidth for video streaming. More detailed information, as well as some test images that illustrate the differences, can be found online in articles like this one.

My end user's bandwidth is limited. How can I have the best possible end user experience with limited bandwidth?

Frame dynamically adjusts bandwidth based on streamed content and network conditions to consume the lowest bandwidth while maintaining the best possible user experience. If you need to set a bandwidth limit for a user connection, administrators can adjust the "Max Bandwidth" slider in session settings.

What settings should be adjusted to best support conferencing applications like Zoom, Microsoft Teams, and Skype?

Administrators wishing to support video conferencing applications in Frame must first have FRP8 enabled for their Frame account. Once FRP8 has been configured, admins can simply enable the “Camera” and “Microphone” options in Session Settings. Accounts using FRP7 can still enable the microphone feature for voice calls. These applications perform best with low-latency network connections. The end user will need to allow certain permissions in the browser, which is discussed in further detail in our Session Features End User Guide.

Under what circumstances can I enable lossless video?

Lossless quality encoding requires up to 40x more bandwidth than “normal” video streaming, as well as the Chrome browser (version 100+) or Frame App on the endpoint device. Administrators will need to add the ``forceLosslessVideoQuality` Advanced Terminal Argument in session settings in order to use this feature.

Application Mode

Application Mode is a feature that transforms the way users interact with remote Windows applications. This feature provides a native web application-like experience while streamlining and enhancing the user interface.

Application Mode has been carefully designed to create a user experience akin to that of a native web application, deviating from the conventional Windows (or Linux) environment. When Application Mode is enabled, the session excludes common Windows/Linux UI elements, such as the taskbar and Start Menu, focusing the user's attention purely on the application. This maximizes the use of the screen's real estate, as the remote application will be started in a fullscreen/maximized mode.

> It is still possible to use multiple applications within Application Mode. The behavior is still more refined and less flexible than multitasking in Desktop Mode.

To be able to use Application mode, you must first create and configure an Application Launchpad. As Frame offers two different Launchpad types, so do we also offer two different types of session modes (Application and Desktop mode). Desktop mode is, as suggested, a mode where a full OS Desktop experience is desired (Along with all of the multitasking features and interfaces).

Launchpad and Session modes might seem similar at first, but there are important distinctions between the two.

Application Mode vs. Launchpads

While both Application Mode and Frame's Launchpads focus on delivering a curated and enjoyable experience, they are different in terms of their operational scope.

Launchpads are client-side interfaces that have no impact on the remote environment. They focus on providing a tailored user experience at the browser or FrameApp level. Launchpads help you navigate the Frame feature set as an end-user with a user experience based on the desired workflow and Session Mode (Application or Desktop).

In contrast, Application Mode alters the remote Windows and Linux environments to mimic the experience of using a native web application. The objective of Application Mode is to blur the distinction between a remote legacy application and a native web application, resulting in a

seamless, immersive experience for the user.

Through Frame's Application Mode, we aim to make remote application use simpler, more intuitive, and more efficient. As we continue to enhance this mode, our goal is to make your workflow smoother and more productive.

If you wish to dig further into what Launchpads can do (and more about what they are), refer to the Launchpad overview page in our docs.

Enhanced Security

In addition to its core features, Application Mode bolsters the security of your remote environment. This includes disabling many default Windows file explorer behaviors, such as right-click context menus and non-client area behaviors, ensuring that user interaction is confined to the application's parameters. By doing so, Frame deters activities outside the intended application experience. These behaviors are not present by default in Desktop Mode (Exclusive to Application Mode).

Custom Taskbar and Start Menu

As mentioned earlier, Application Mode removes the traditional Windows taskbar and Start Menu. The latest version of Application Mode (App Mode 2.0) adds a few features which also include a custom, Frame-branded taskbar and Start Menu. These are designed to simplify task management and provide an uncluttered, user-friendly interface. Their configuration is also flexible, meaning you can adjust and personalize your workspace.

Taskbar Override Configuration

To take things further with the new taskbar and start menu, you can customize the taskbar programmatically through a JSON configuration file. This JSON file provides the settings for overriding the default behavior of the custom taskbar and start menu. The flexibility of the customization is currently set on specifying which apps should appear in the menu, the order of those applications, and also a folder/subfolder hierarchy for all applications.

This override process outlines and controls the structure and content of the taskbar. With this file, you can organize your taskbar to align with your work style, improving your overall efficiency.

You can find this JSON file in the `C:\ProgramData\Nutanix\Frame\Config` directory. The filename is `server_launchpad_override.json`.

> The Config directory does not exist by default, so make sure you create the directory before attempting to create the server_launchpad_override.json if you are attempting this override with a script.

An example of what this file could look like is the following:

```
{
  "name": "root",
  "order": 0,
  "applications": [
    {
      "name": "Frame Explorer",
      "path": "C:\\Program Files\\Frame\\FrameExplorer\\FrameExplorer.exe",
      "icon_url": "https://next-cpanel-
dev.s3.amazonaws.com/images/icons/frame_explorer.png",
      "order": 0
    },
    {
      "name": "Notepad",
      "path": "C:\\Windows\\system32\\notepad.exe",
      "icon_url": "https://next-cpanel-dev.s3.amazonaws.com/images/icons/notepad.png",
      "order": 2
    }
  ],
  "folders": [
    {
      "name": "Folder between two applications",
      "order": 1,
      "applications": [
        {
          "name": "Command prompt",
          "order": 0,
          "path": "C:\\Windows\\System32\\cmd.exe",
          "working_directory": "C:\\"
        },
        {
          "name": "Browser",
          "order": 2,
          "path": "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe"
        },
        {
          "name": "Dizzion Website",
          "order": 3,
          "path": "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe",
          "arguments": "https://www.dizzion.com/",
          "icon_url": "https://staging-cpa-6ae2ad7f7ca734e5abe8.s3.us-east-
1.amazonaws.com/images/icons/ce69f714bad94c83ac40f5fb9648308e.png"
        }
      ]
    }
```

```
  ]
}
```

> The "name": "root property, and "order": 0, property at the root level of the JSON file are both required. You must have these in the file for this to work properly. This is basically specifying the “root folder” of the entire hierarchy (Top-most level of the hierarchy).

> Every Application and Folder item that is specified in the corresponding arrays absolutely require both the `order` and name fields.

As you can see in the JSON file above, there are two properties in the JSON file that are relevant to the proper customization of the taskbar and start menu: “applications” and “folders”. Both of these properties are arrays that allow you to add as many items as you would like (Folders allows for as many sub-folders and sub-applications as you would like as well).

For an application object, here is a list of fields that can (or must) be specified:

Required fields

path (string) - This is an absolute path to the application's executable file (.exe, .sh, etc).

Optional fields

arguments (string) - If provided, these arguments will be used as the application’s starting arguments. This can be left empty..

working_directory (string) - If provided it will be used as working directory when starting the application.

icon_url (string) - This is a URL from which to load the application’s icon. If not provided, Frame's service will attempt to use the corresponding Operating System API to fetch the icon associated with the provided executable's path.

> Unlike applications, all folders use the default OS folder icon.

Frame Remoting Protocol

Frame Remoting Protocol (FRP) defines the communication between an end user's device with a browser or Frame App and the remote virtual machine running the Frame Guest Agent (FGA).

Frame Remoting Protocol (FRP) 8

Frame Remoting Protocol 8 (FRP8) is the latest remoting protocol for, based on WebRTC. FRP8, by default, uses UDP as its transport layer. FRP8 provides customers with an extensive list of new features that were previously unavailable with FRP7.

Once FRP8 is enabled for a Frame Account, the following features/enhancements become available:

- Improved audio/video synchronization which better adapts to poor network conditions (packet loss, network jitter) by utilizing UDP instead of TCP for transport
- Webcam support
- High-quality microphone redirection from the endpoint device to the session.
- Generic USB Redirection support in Frame App for Windows (requires Frame Virtual USB Driver to be installed on Frame Sandbox/workload VMs) allows local USB devices to be passed into the Frame session using an USB Filter Driver included with Frame App for Windows.
- App Mode 2.0 for Application Launchpads, which delivers a more intuitive user experience by providing a Windows desktop with default lockdowns to the Desktop, Start Menu, Taskbar, Notification Area/System Tray, etc.

Features

Each new feature of FRP8 is documented below. Administrators may wish to provide their end users with the Session Features End User Guide to understand how they can use these features in a session.

Webcam Support and Microphone Enhancement

FRP8 sessions utilize WebRTC technology, which enables Frame to efficiently capture audio/video data from the endpoint device and stream it to the VM in real time. WebRTC provides a seamless webcam and microphone experience to those using applications such as Zoom, Microsoft Teams, Slack, Google Meet, Cisco Webex, or any other audio/video telephony

solutions running natively within a Frame session.

FrameWebCam-f843202b3128892cd4e6ecee357f0291 copy.jpeg

Frame Session with Zoom

Webcam and microphone support can be enabled by administrators from the Session Settings page in the account's Dashboard. Multiple webcams are supported in the session, end users need simply to select the desired device. End user instructions for these features can be found here.

> **Attention**
> User experience with audio and video-conferencing solutions within a Frame session depends significantly on the networking conditions (bandwidth, latency, packet loss, network jitter, packet shaping, QoS, etc.) between the user and the Frame VM and from the Frame VM to the Internet. POC testing at scale is strongly recommended to ensure satisfactory user experience.

Prerequisites

Use of FRP8 requires the following prerequisites:

- Networking configuration, as described in our Network Requirements documentation for FRP8 and your specific deployment model.
- Use of Frame Guest Agent 8.1 or greater, as discussed in our Frame Guest Agent 8 documentation in all Frame-managed workload VMs (Sandbox, Utility Server, test/production instances).
- OS firewall (e.g., Windows Firewall or third-party firewall) configuration changes to support the required TCP/UDP ports.
- Streaming Gateway Appliance (SGA) 4 or greater (version 3 is being deprecated soon), if required, as discussed in our Streaming Gateway Appliance documentation.

Configuration

Once the prerequisites have been met for FRP8, Administrators can enable FRP8 in Session Settings (Dashboard > Settings > Session).

> If the Frame Account was created in **public cloud using Frame-managed networking** and the FRP8-required ports are not enabled, you will need to first enable the ports and then enable use of FRP8. FRP8 cannot be enabled until the required ports have been

> opened, as shown in the image above.

Enabling FRP8 on a Frame account using Frame-managed networking results in the following:

- For Frame accounts using public cloud infrastructure:
  - Inbound protocols/ports udp/4503-4509 and tcp/4503-4509 on the workload VPC/VNET and udp/3478 on the SGA VPC/VNET (if SGA is auto-deployed during account creation) are enabled through security group updates.
- FRP 8.0 protocol is enabled for Frame sessions.

For Frame accounts using customer-managed networking (AHV or public cloud), the Frame administrator is responsible for ensuring the required FRP8 network requirements are satisfied before FRP8 is enabled in Session Settings.