AWS Systems Manager Session Manager

PDF (systems-manager-ug.pdf#session-manager) | RSS (aws-systems-manager-user-guide-updates.rss)

Session Manager is a fully managed AWS Systems Manager capability. With Session Manager, you can manage your Amazon Elastic Compute Cloud (Amazon EC2) instances, edge devices, and on-premises servers and virtual machines (VMs). You can use either an interactive one-click browser-based shell or the AWS Command Line Interface (AWS CLI). Session Manager provides secure and auditable node management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. Session Manager also allows you to comply with corporate policies that require controlled access to managed nodes, strict security practices, and fully auditable logs with node access details, while providing end users with simple one-click cross-platform access to your managed nodes. To get started with Session Manager, open the Systems Manager console (https://console.aws.amazon.com/systems-manager/session-manager) . In the navigation pane, choose **Session Manager**.

How can Session Manager benefit my organization?

Session Manager offers these benefits:

- **Centralized access control to managed nodes using IAM policies**

  Administrators have a single place to grant and revoke access to managed nodes. Using only AWS Identity and Access Management (IAM) policies, you can control which individual users or groups in your organization can use Session Manager and which managed nodes they can access.

- **No open inbound ports and no need to manage bastion hosts or SSH keys**

  Leaving inbound SSH ports and remote PowerShell ports open on your managed nodes greatly increases the risk of entities running unauthorized or malicious commands on the managed nodes. Session Manager helps you improve your security posture by letting you close these inbound ports, freeing you from managing SSH keys and certificates, bastion hosts, and jump boxes.

- **One-click access to managed nodes from the console and CLI**

  Using the AWS Systems Manager console or Amazon EC2 console, you can start a session with a single click. Using the AWS CLI, you can also start a session that runs a single command or a sequence of commands. Because permissions to managed nodes are provided through IAM policies instead of SSH keys or other mechanisms, the connection time is greatly reduced.

- **Port forwarding**

  Redirect any port inside your managed node to a local port on a client. After that, connect to the local port and access the server application that is running inside the node.

- **Cross-platform support for Windows, Linux, and macOS**

  Session Manager provides support for Windows, Linux, and macOS from a single tool. For example, you don't need to use an SSH client for Linux and macOS managed nodes or an RDP connection for Windows Server managed nodes.

- **Logging and auditing session activity**

  To meet operational or security requirements in your organization, you might need to provide a record of the connections made to your managed nodes and the commands that were run on them. You can also receive notifications when a user in your organization starts or ends session activity.

  Logging and auditing capabilities are provided through integration with the following AWS services:

  - **AWS CloudTrail** – AWS CloudTrail captures information about Session Manager API calls made in your AWS account and writes it to log files that are stored in an Amazon Simple Storage Service (Amazon S3) bucket you specify. One bucket is used for all CloudTrail logs for your account. For more information, see Logging AWS Systems Manager API calls with AWS CloudTrail (./monitoring-cloudtrail-logs.html) .
  - **Amazon Simple Storage Service** – You can choose to store session log data in an Amazon S3 bucket of your choice for debugging and troubleshooting purposes. Log data can be sent to your Amazon S3 bucket with or without encryption using your AWS KMS key. For more information, see Logging session data using Amazon S3 (console) (./session-manager-logging.html#session-manager-logging-s3) .
  - **Amazon CloudWatch Logs** – CloudWatch Logs allows you to monitor, store, and access log files from various AWS services. You can send session log data to a CloudWatch Logs log group for debugging and troubleshooting purposes. Log data can be sent to your log group with or without AWS KMS encryption using your KMS key. For more information, see Logging session data using Amazon CloudWatch Logs (console) (./session-manager-logging.html#session-manager-logging-cloudwatch-logs) .
  - **Amazon EventBridge** and **Amazon Simple Notification Service** – EventBridge allows you to set up rules to detect when changes happen to AWS resources that you specify. You can create a rule to detect when a user in your organization starts or stops a session, and then receive a notification through Amazon SNS (for example, a text or email message) about the event. You can also configure a CloudWatch event to initiate other responses. For more information, see Monitoring session activity using Amazon EventBridge (console) (./session-manager-auditing.html#session-manager-auditing-eventbridge-events) .

> **Note**

Logging isn't available for Session Manager sessions that connect through port forwarding or SSH. This is because SSH encrypts all session data, and Session Manager only serves as a tunnel for SSH connections.

Who should use Session Manager?

- Any AWS customer who wants to improve their security and audit posture, reduce operational overhead by centralizing access control on managed nodes, and reduce inbound node access.
- Information Security experts who want to monitor and track managed node access and activity, close down inbound ports on managed nodes, or allow connections to managed nodes that don't have a public IP address.
- Administrators who want to grant and revoke access from a single location, and who want to provide one solution to users for Linux, macOS, and Windows Server managed nodes.
- Users who want to connect to a managed node with just one click from the browser or AWS CLI without having to provide SSH keys.

What are the main features of Session Manager?

- **Support for Windows Server, Linux and macOS managed nodes**

  Session Manager enables you to establish secure connections to your Amazon Elastic Compute Cloud (EC2) instances, edge devices, and on-premises servers and virtual machines (VMs). For a list of supported operating system types, see Setting up Session Manager (./session-manager-getting-started.html) .

  > **Note**
  >
  > Session Manager support for on-premises machines is provided for the advanced-instances tier only. For information, see Turning on the advanced-instances tier (./systems-manager-managedinstances-advanced.html) .

- **Console, CLI, and SDK access to Session Manager capabilities**

  You can work with Session Manager in the following ways:

  The **AWS Systems Manager console** includes access to all the Session Manager capabilities for both administrators and end users. You can perform any task that is related to your sessions by using the Systems Manager console.

  The Amazon EC2 console provides the ability for end users to connect to the EC2 instances for which they have been granted session permissions.

The **AWS CLI** includes access to Session Manager capabilities for end users. You can start a session, view a list of sessions, and permanently end a session by using the AWS CLI.

> **Note**
>
> To use the AWS CLI to run session commands, you must be using version 1.16.12 of the CLI (or later), and you must have installed the Session Manager plugin on your local machine. For information, see (Optional) Install the Session Manager plugin for the AWS CLI (./session-manager-working-with-install-plugin.html) .

The **Session Manager SDK** consists of libraries and sample code that allows application developers to build front-end applications, such as custom shells or self-service portals for internal users that natively use Session Manager to connect to managed nodes. Developers and partners can integrate Session Manager into their client-side tooling or Automation workflows using the Session Manager APIs. You can even build custom solutions.

- **IAM access control**

  Through the use of IAM policies, you can control which members of your organization can initiate sessions to managed nodes and which nodes they can access. You can also provide temporary access to your managed nodes. For example, you might want to give an on-call engineer (or a group of on-call engineers) access to production servers only for the duration of their rotation.

- **Logging and auditing capability support**

  Session Manager provide you with options for auditing and logging session histories in your AWS account through integration with a number of other AWS services. For more information, see Auditing session activity (./session-manager-auditing.html) and Logging session activity (./session-manager-logging.html) .

- **Configurable shell profiles**

  Session Manager provides you with options to configure preferences within sessions. These customizable profiles allow you to define preferences such as shell preferences, environment variables, working directories, and running multiple commands when a session is started.

- **Customer key data encryption support**

  You can configure Session Manager to encrypt the session data logs that you send to an Amazon Simple Storage Service (Amazon S3) bucket or stream to a CloudWatch Logs log group. You can also configure Session Manager to further encrypt the data transmitted between client machines and your managed nodes during your sessions. For information, see Logging session activity (./session-manager-logging.html) and Configure session preferences (./session-manager-getting-started-configure-preferences.html) .

- **AWS PrivateLink support for managed nodes without public IP addresses**

  You can also set up VPC Endpoints for Systems Manager using AWS PrivateLink to further secure your sessions. AWS PrivateLink limits all network traffic between your managed

nodes, Systems Manager, and Amazon EC2 to the Amazon network. For more information, see (Optional) Create a VPC endpoint (./setup-create-vpc.html) .

- **Tunneling**

  In a session, use a Session-type AWS Systems Manager (SSM) document to tunnel traffic, such as http or a custom protocol, between a local port on a client machine and a remote port on a managed node.

- **Interactive commands**

  Create a Session-type SSM document that uses a session to interactively run a single command, giving you a way to manage what users can do on a managed node.

What is a session?

A session is a connection made to a managed node using Session Manager. Sessions are based on a secure bi-directional communication channel between the client (you) and the remote managed node that streams inputs and outputs for commands. Traffic between a client and a managed node is encrypted using TLS 1.2, and requests to create the connection are signed using Sigv4. This two-way communication allows interactive bash and PowerShell access to managed nodes. You can also use an AWS Key Management Service (AWS KMS) key to further encrypt data beyond the default TLS encryption.

For example, say that John is an on-call engineer in your IT department. He receives notification of an issue that requires him to remotely connect to a managed node, such as a failure that requires troubleshooting or a directive to change a simple configuration option on a node. Using the AWS Systems Manager console, the Amazon EC2 console, or the AWS CLI, John starts a session connecting him to the managed node, runs commands on the node needed to complete the task, and then ends the session.

When John sends that first command to start the session, the Session Manager service authenticates his ID, verifies the permissions granted to him by an IAM policy, checks configuration settings (such as verifying allowed limits for the sessions), and sends a message to SSM Agent to open the two-way connection. After the connection is established and John types the next command, the command output from SSM Agent is uploaded to this communication channel and sent back to his local machine.

Topics

- Troubleshooting Session Manager (./session-manager-troubleshooting.html)