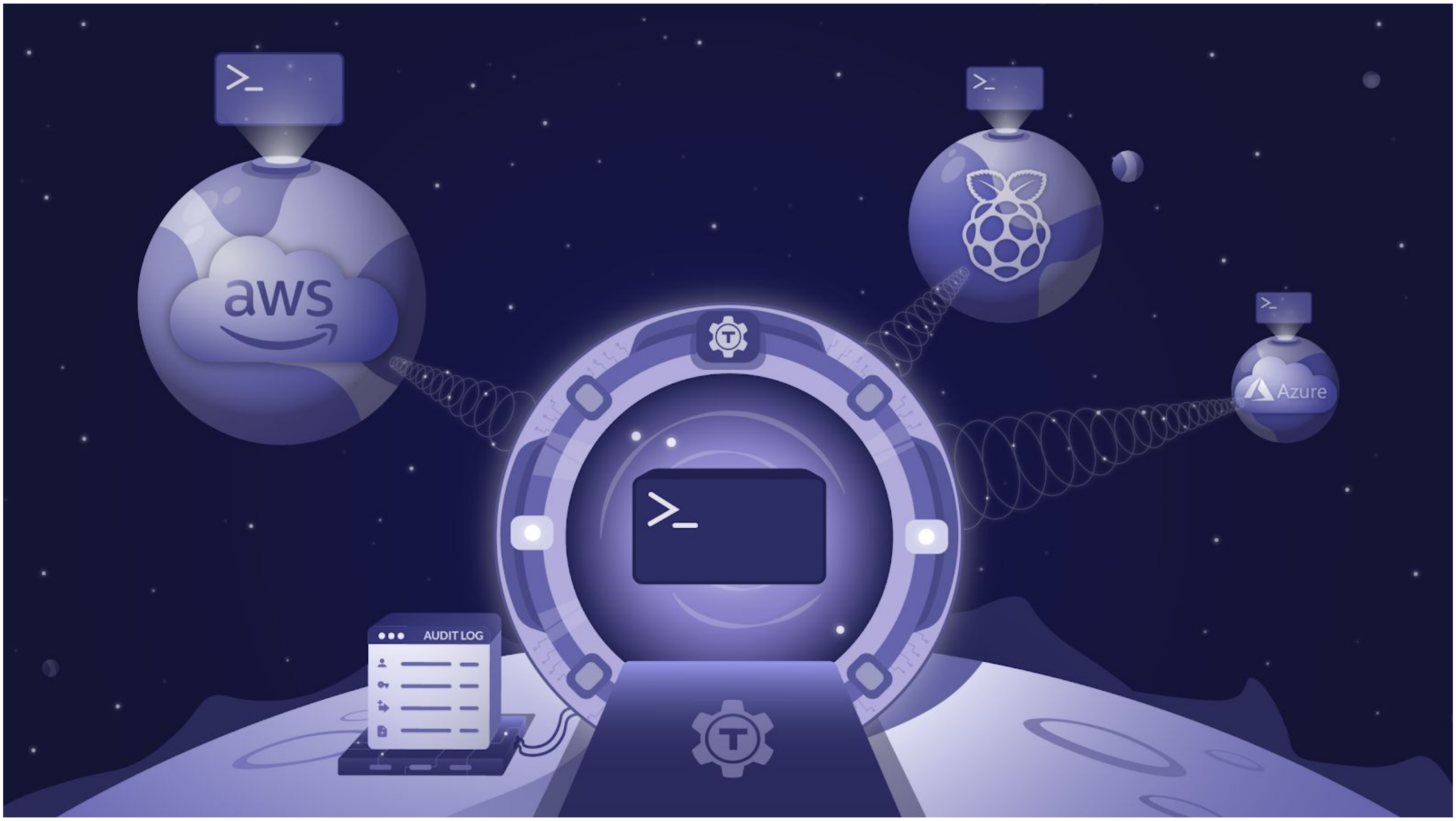
Gravitational Teleport compared to AWS SSM Session Manager

BY GRAVITATIONAL



Abstract

AWS Systems Manager provides “session management” and “session recording” features in their effort to make AWS the best place to run enterprise class workloads. In this paper, we’ll explore these new interactive session capabilities and compare its design and features to Teleport.

Last Updated: July 13th 2020

Table of Contents

Introduction

In this paper, we will provide a brief description of what SSM Session Manager is and how it compares to Gravitational's Teleport secure access for developers. We'll compare the significant design and feature differences and the operational overhead of the solutions. Finally, we have provided a feature matrix of the two solutions.

What is AWS SSM Session Manager?

Amazon's "AWS Systems Manager," better known as "SSM" to long-time AWS users, was announced at the end of 2017, replacing the similarly named "EC2 Systems Manager" that had launched a year prior. Typical of other general AWS services, System Manager provides a broad spectrum of features instead of a focused and opinionated product.

SSM's capabilities revolve around managing bare operating-system level details that would otherwise be inaccessible from Amazon's ever-expanding and all-encompassing `aws` CLI and API control plane layer. It has grown into having its own dedicated interface within the Amazon AWS console and its underlying system-level agent has been released under an open source software license on Github. Amazon has touted SSM as a general-purpose devops tool - a replacement for fleet management stalwarts such as Ansible, Chef, Puppet and similar decade-old cloud agnostic tooling.

How does Session Manager work?

Systems Manager's core is a set of APIs built upon other proprietary AWS services such as IAM, CloudWatch, S3 and KMS. To use SSM, an administrator typically installs the `aws` command line CLI plus additional SSM plugins on their laptop and then also runs an agent, the privileged `amazon-ssm-agent` daemon, on every EC2 instance in their fleet. Access to the SSM service (APIs) is controlled via AWS Identity & Access Management (IAM). For the SSM node-level agent to function, a given EC2 instance must be able to "assume" the requisite roles (aka, node-level permissions).

Once SSM agents are up and running, an administrator submits commands or session requests via the AWS CLI (API) and then the instance-local agents receive messages to take action. In the case of interactive sessions, the admin's AWS CLI plugin opens up a websocket connection to the SSM service in a given region, which in turn is directly connected to the `ssm` agent running on the target instance. Session recording appears to happen locally on each EC2 instance and after a session is closed, the node-local SSM process uploads the logs to long-term (S3) storage.

How does Teleport compare to Session Manager (SM)?

The AWS team is always improving Session Manager and in recent years they have added many of the basic SSH features supported by Teleport and now provide the ability to use Session Manager in different environments. There are a few reasons why you would want to pick Teleport over AWS SM. AWS SM strongly relies on IAM; while a best practice this can also be a con. As organisations move to OIDC for authentication, creating IAM users with the correct permissions can become "politically challenging," as said by AWS EKS Team.

Teleport Trusted Clusters feature is a powerful addition to monitoring and managing access between AWS accounts. This can be extremely helpful for MSP providers, or companies with multiple AWS root accounts. By rolling out Teleport Trusted Clusters, we are able to provide a foundation that'll help streamline multi-cloud access and auditing.

Teleport Bastion setup compared to Session Manager requirements?

Gravitational Teleport enforces an opinionated security design based on Google's BeyondCorp security model. As part of this design, all traffic must be routed through a simple proxy or bastion. While the proxy is designed with the aforementioned security model in mind to be stateless and only pass through encrypted data, it does require provisioning a machine to run the service. This is the greatest appeal of System's Manager and its Session Manager -- instead of having to run a dedicated bastion, an administrator only has to run the ssm daemon and manage appropriate IAM roles for their fleet. On the other hand, some of Teleport's features (which SSM has a hard time replicating) come from running inline as a proxy for all connections.

Session Manager has agent and plugin packages available for all major operating systems. Teleport supports most Unix / Linux based operating systems on the server side. The Teleport client can additionally run on Windows with limited functionality.

For AWS, Teleport comes with a single AMI-based installation option, plus in-depth examples for typical highly-available configurations based on CloudFormation, Terraform, Ansible, etc. that are easily adapted into existing configuration & change management practices.

Teleport end-user usage compared to SSM `aws ssm start-session` ?

Teleport comes with a native client called `tsh` that serves as a drop-in replacement for workflows that currently leverage `ssh` directly. In practice, most Teleport users will alias their ssh to tsh and all of their existing commands/scripts will continue to work just as they did with legacy, static-key or password-authenticated ssh. Teleport also automatically upgrades end users to certificate-based authentication, removing the need to manage and pre-distribute public user and host keys.

Session Manager requires that end-users have the `aws` client and credentials installed. The aws client does not integrate with existing ssh workflows/tools. When ssm starts an end-to-end session, the provided terminal uses a non-SSH HTTPS websockets “upgraded” TCP connection for its transport to materialize a shell on the remote EC2 instance. Users have reported issues with how ssm handles normal and routine shell scenarios such as typing Control-C to cancel a long-running process. While ssm’s handling of `TERM=xterm-256color` appears to work, the resulting sessions recorded in the administrator log review interface show a raw escape-encoded recording rather than a usable plain text recording.

Teleport optionally integrates with normal UNIX (POSIX) login facilities, so active user sessions can have pam.d rules applied to them and sessions may be visible within traditional node-local logging facilities. SSM-connected sessions do not tie back to these long-held POSIX conventions, which can be confusing in multi-user scenarios - “Who is logged into this EC2 instance? I have to go look at SSM’s web UI to see, I can’t simply run `w`?”

Another well-liked aspect of Teleport’s end-user experience which isn’t matched by Session Manager is the ability to interactively share sessions. Users simply run `teleport status` while in a session and can share the unique session URL with coworkers (subject to that coworker having the same Teleport RBAC permissions).

Related to session sharing (or session playback), companies that leverage Teleport will often provide access to session replay (and session reply only) within Teleport directly to SOC-2 CPAs or similar auditors so that they can verify low, shell-level activity was within scope without having to open tons of tickets or ask for screenshots. Within SSM, there is no simple interface for delegation of session replay, and as noted above, ssm’s recordings are often riddled with escape sequences.

Teleport logging and audit compared to SSM + CloudWatch + S3

How does the picture look if we zoom in further on session capture, logging, and audit topics required by information security professionals who depend on accurate data capture? Teleport is a well-known open source project with 8,000+ stars on GitHub with a track record of public audits by security researchers. Security is its primary design goal and accurate logging of all activity is also of paramount concern. Besides logs being captured via an agent, Teleport offers a full “man-in-the-middle” mode that captures sessions even if an instance-level agent is compromised. SSM batches its logs locally on the node to be audited and key aspects of its recordings are trivially easy to “hide” from if you also have “root” on the nodes, which is ssm-agent’s default install. To “hide”, an administrator simply has to kill the ssm recording process at the end of the session instead of logging out cleanly, or they can simply block DNS or network access to S3 & CloudWatch from the target EC2 instance. SSM also only plugs into

CloudWatch, whereas Teleport's events are available as a JSON stream (locally on disk or within a DynamoDB table) and can be fed into standard enterprise log indexing pipelines such as ELK, Splunk, Sumo Logic, LogDNA, etc.

AWS SSM Run vs tsh ssh

AWS provides an option to execute commands on EC2 host using AWS Systems Documents , and provides examples for standard update / run options. This can be helpful when trying to patch and update a fleet of machine, but the syntax is AWS specific.

```
aws ssm send-command \
      --document-name document-name \
      --targets Key=tag:Department,Values=Sales,Finance \
      [...]
```

Teleport's tsh client offers the same functionally but using a familiar SSH syntax. Using TSH Labels provides a powerful abstraction for sysadmins.

```
tsh ssh os=ubuntu apt-get update -y
```

Kubernetes and EKS Support

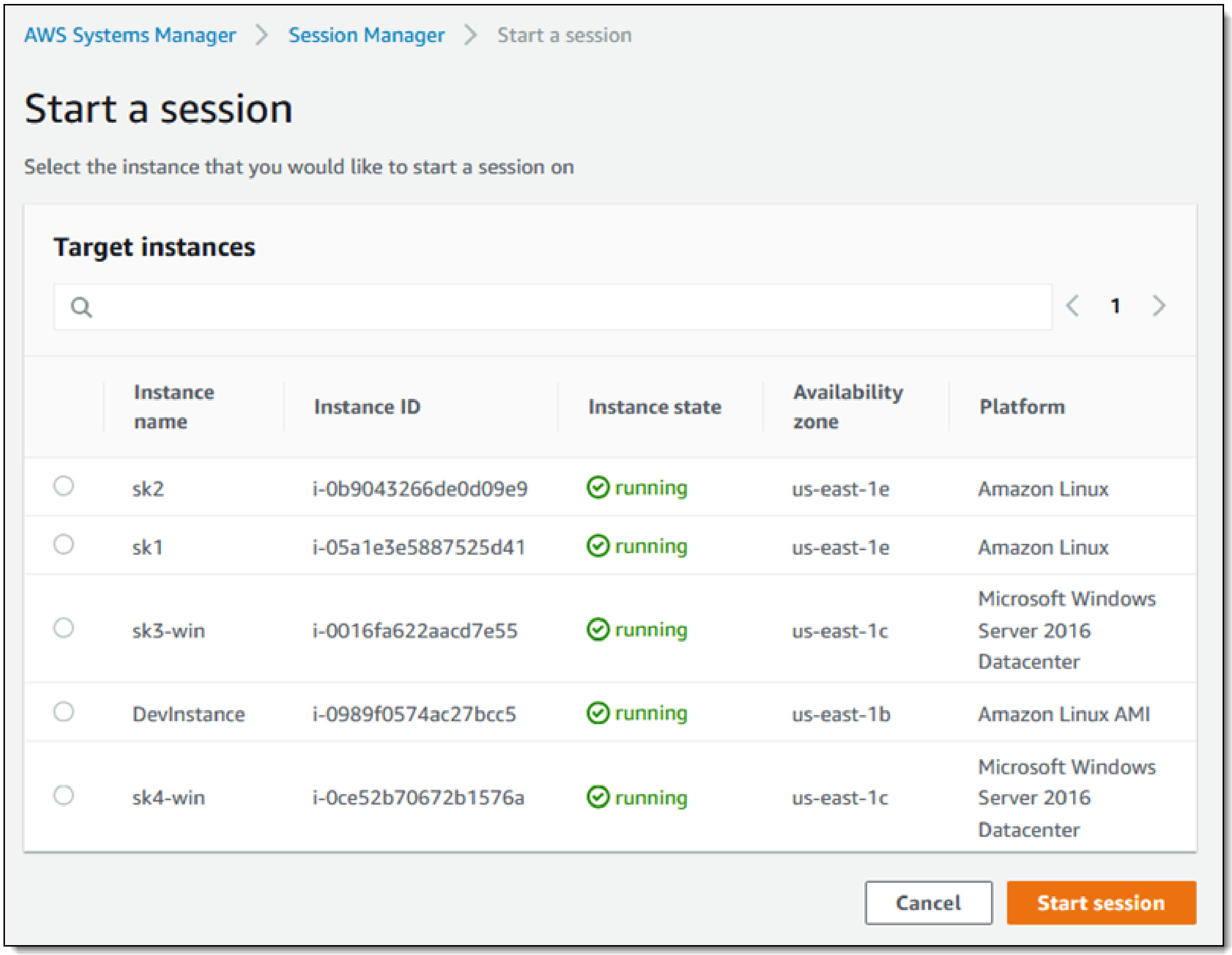
Teleport leverages user and group impersonation to obtain access to Kubernetes. When a cluster is created in AWS. In larger organizations, Teleport offers more dynamic access for team members without the extra overhead of managing individual IAM roles.

Enhanced Linux Session Recording

Gravitational offers best in class session recording, backed by our expertise using BPF. Teleport can transform SSH sessions into structured events. This has a few advantages of SSM auditing capabilities, by providing visibility into Shell Scripts and End-User Objuscation.

User Experience

Session Manager
AWS Session Manager falls under the AWS Systems Manager UI. Systems manager offers a range of services that can provide insight and automation around Operation, Applications and Instances. This has resulted in System Manager being a jack of all trades and master of node. The start a session UI is currently lacking when compared to Teleport's offering.



Start a session

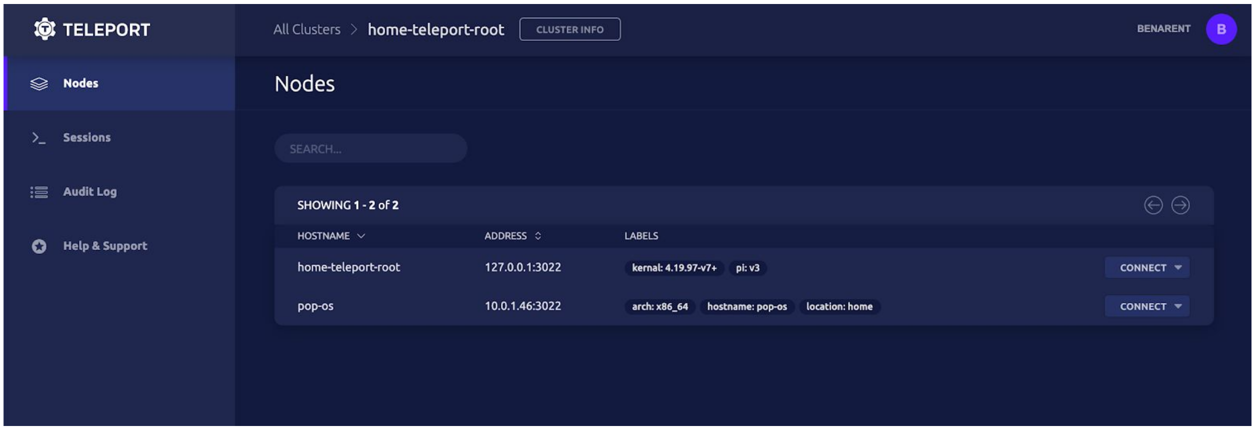
Select the instance that you would like to start a session on

Target instances

	Instance name	Instance ID	Instance state	Availability zone	Platform
○	sk2	i-0b9043266de0d09e9	✓ running	us-east-1e	Amazon Linux
○	sk1	i-05a1e3e5887525d41	✓ running	us-east-1e	Amazon Linux
○	sk3-win	i-0016fa622aacd7e55	✓ running	us-east-1c	Microsoft Windows Server 2016 Datacenter
○	DevInstance	i-0989f0574ac27bcc5	✓ running	us-east-1b	Amazon Linux AMI
○	sk4-win	i-0ce52b70672b1576a	✓ running	us-east-1c	Microsoft Windows Server 2016 Datacenter

Teleport

Gravitational Teleport’s web UI offers a centralized location for starting sessions and auditing events. These options can be limited via RBAC roles, providing customization for teams wanting to provide just enough access to developers.

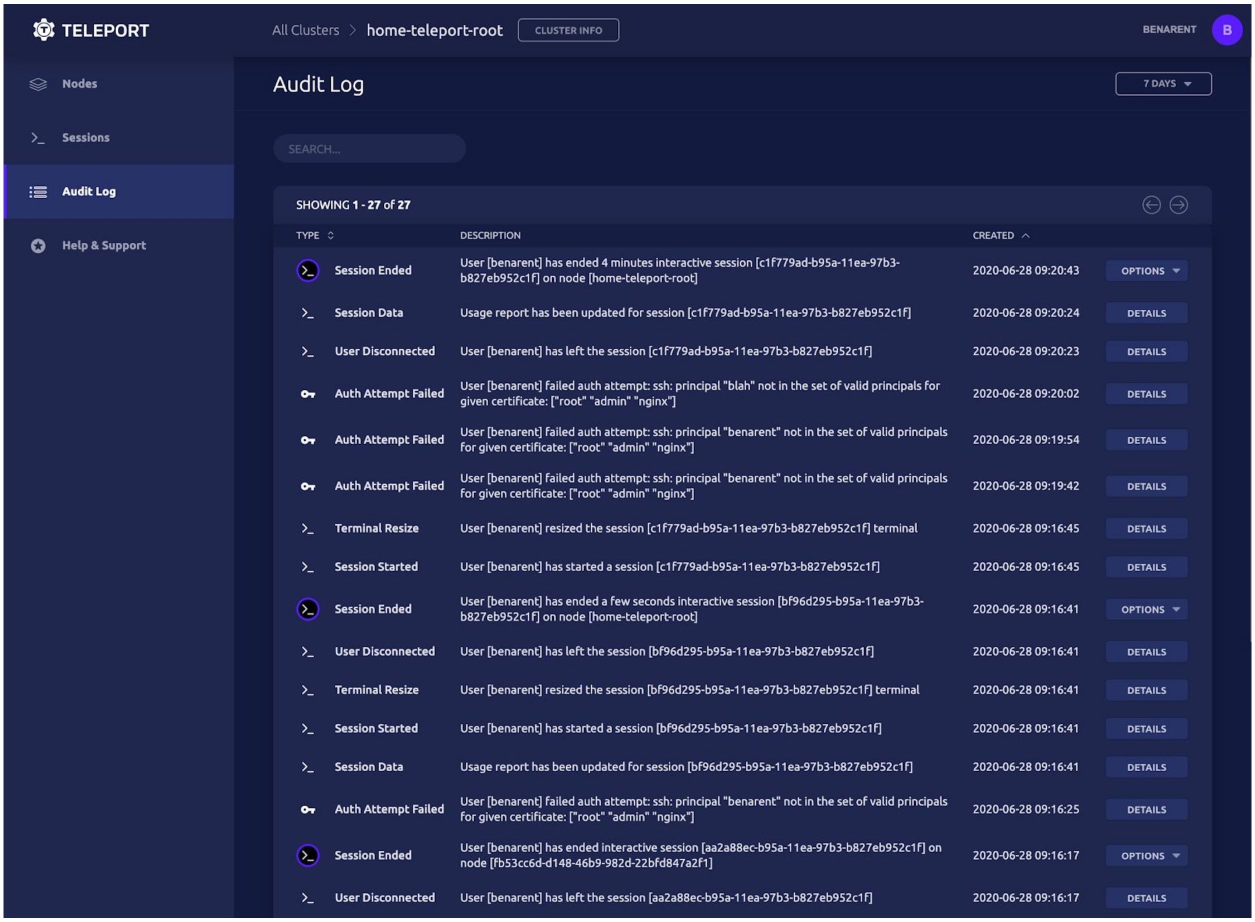


Nodes

SHOWING 1 - 2 of 2

HOSTNAME	ADDRESS	LABELS	
home-teleport-root	127.0.0.1:3022	kernal: 4.19.97-v7+ pi: v3	CONNECT
pop-os	10.0.1.46:3022	arch: x86_64 hostname: pop-os location: home	CONNECT

Teleports Audit log provides at a glance view into session and access activity. The audit log can be consumed by log aggregation systems, but having a UI to review activity can provide a jumping off point for retrospectives.



Audit Log

SHOWING 1 - 27 of 27

TYPE	DESCRIPTION	CREATED	
Session Ended	User [benarent] has ended 4 minutes interactive session [c1f779ad-b95a-11ea-97b3-b827eb952c1f] on node [home-teleport-root]	2020-06-28 09:20:43	OPTIONS
Session Data	Usage report has been updated for session [c1f779ad-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:20:24	DETAILS
User Disconnected	User [benarent] has left the session [c1f779ad-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:20:23	DETAILS
Auth Attempt Failed	User [benarent] failed auth attempt: ssh: principal "blah" not in the set of valid principals for given certificate: ["root" "admin" "nginx"]	2020-06-28 09:20:02	DETAILS
Auth Attempt Failed	User [benarent] failed auth attempt: ssh: principal "benarent" not in the set of valid principals for given certificate: ["root" "admin" "nginx"]	2020-06-28 09:19:54	DETAILS
Auth Attempt Failed	User [benarent] failed auth attempt: ssh: principal "benarent" not in the set of valid principals for given certificate: ["root" "admin" "nginx"]	2020-06-28 09:19:42	DETAILS
Terminal Resize	User [benarent] resized the session [c1f779ad-b95a-11ea-97b3-b827eb952c1f] terminal	2020-06-28 09:16:45	DETAILS
Session Started	User [benarent] has started a session [c1f779ad-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:16:45	DETAILS
Session Ended	User [benarent] has ended a few seconds interactive session [bf96d295-b95a-11ea-97b3-b827eb952c1f] on node [home-teleport-root]	2020-06-28 09:16:41	OPTIONS
User Disconnected	User [benarent] has left the session [bf96d295-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:16:41	DETAILS
Terminal Resize	User [benarent] resized the session [bf96d295-b95a-11ea-97b3-b827eb952c1f] terminal	2020-06-28 09:16:41	DETAILS
Session Started	User [benarent] has started a session [bf96d295-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:16:41	DETAILS
Session Data	Usage report has been updated for session [bf96d295-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:16:41	DETAILS
Auth Attempt Failed	User [benarent] failed auth attempt: ssh: principal "benarent" not in the set of valid principals for given certificate: ["root" "admin" "nginx"]	2020-06-28 09:16:25	DETAILS
Session Ended	User [benarent] has ended interactive session [aa2a88ec-b95a-11ea-97b3-b827eb952c1f] on node [fb53cc6d-d148-46b9-982d-22bfd847a2f1]	2020-06-28 09:16:17	OPTIONS
User Disconnected	User [benarent] has left the session [aa2a88ec-b95a-11ea-97b3-b827eb952c1f]	2020-06-28 09:16:17	DETAILS

Other miscellaneous concerns

Session Manager has native Windows server support for PowerShell. Teleport includes a Windows client with limited functionality and may add Windows server support in the future.

Summary and feature comparison matrix

Both Teleport and Session Manager provide a convenient way to manage and restrict shell access. For AWS-only users, Session Manager offers less operational overhead by integrating with IAM and does not require running any service, other than installing the SSM daemon. However, Teleport provides better interoperability with native SSH based workflows and tools and is open source. Teleport also delivers a more robust security design and has the additional nice-to-have feature of session sharing. You can refer to the matrix below for a more comprehensive feature comparison.

	Gravitational Teleport	AWS Session Manager
Open Source Client & Agent	Yes	Yes
Open Source Server	Yes	No
Legacy Linux Support (RHEL-5.x)	Yes*	No
KMS-backed encryption at rest	Yes	Yes
Native SSH Compatibility	Yes	No****
Works with ssh-agent	Yes	No
Strict session timeouts	Yes	No
Kubernetes support	Yes	No
Works on-premises	Yes	Yes***
Works within air-gapped environments	Yes	No
FedRAMP/FISMA FIPS 140-2 Validated Crypto mode	Yes*	No**
Interactive Session Sharing	Yes	No
Works with any Single Sign-On (SAML or OIDC)	Yes	Yes
Supports tagging and labels	Yes	Yes
Works without bastion server	No	Yes
Works with OpenSSH clients & servers	Yes	No
Logs directly to Cloudwatch	No	Yes
Man-in-the-middle recording mode	Yes	No
Supports SSH tunnels and port forwarding	Yes	Yes

* Only available with the Teleport Enterprise offering.
** Implicit in their GovCloud offering but not available elsewhere.
*** Limit to 1,000 instances (per account, per region)
**** AWS Systems Manager offers a Run Command that could replace parts of SSH workflow.