Tony Shen

AWS SYSTEM MANAGER’S SESSION MANAGER

AWS System Manager’s Session Manager

Contents

Revision: v0.3
October 24, 2019

Glossary

Term	Description
AWS	Amazon Web Services
SM	System Manager
AMI	Amazon Machine Image
FB	Foundation Build
FBA	Foundation Build Automation
VPC	Virtual Private Cloud
CIDR	Classless Inter-Domain Routing
SG	Security Group
AZ	Availability Zone
KP	Key Pair
IaaS	Infrastructure as a Service
Local machine	An off-cloud computer that connects to cloud
Ingress traffic	Inbound traffic
AA	User Authentication and Authorization

Introduction

AWS System Manager (SM) is a management tool for cloud operators and administrators to use in managing EC2 instances. In AWS cloud, EC2 instances are virtual machines, constituting the core of AWS cloud Infrastructure as a Service (IaaS).

SM provides EC2 management capabilities in three areas:

1. Insights
2. Actions
3. Shared Resources

In Insights, SM builds and maintains an inventory of EC2 instances. In Actions, SM offers Session Manager for user to connect to an EC2 instance. In Shared Resources, SM provides a single point of entry to all Managed Instances. Managed Instances are EC2 instances that communicate with SM via SM agents. SM agent is included in an AMI (AWS Machine Image) released in 2017 or later. An EC2 instance launched using such an AMI has SM agent installed. When the instance is up and running, SM agent is running by default. An EC2 instance launched using an AMI older than 2017, however, SM agent was not included, and therefore, a manual agent installation is required in the instance in order for it to be able to communicate with SM.

In this article, we will show you how SM can be configured and enabled automatically using Foundation Build Automation (FBA). We will also explain how this SM enablement is achieved behind the scene. An example of manually configuring and enabling SM using AWS Console follows so as to provide reader with visual details using the console’s GUI interface. In the demo, all EC2 instances are launched from AMIs released in 2018, with SM agent installed.

System Manger Configuration and Enablement Procedure

The procedure is outlined at a high level below. Most of the steps are included and performed by FBA automatically. We will address and discuss the remaining steps in the rest of the article where and when needed.

1. Enabling VPC for both DNS and Hostname support
2. Verifying Availability Zone’s support to VPC endpoint for System Manager
3. Creating VPC Security Group that allows Ingress traffic on port 443 from VPC CIDR
4. Creating System Manager required IAM role
5. Creating System Manager required permission policy
6. Attaching the policy to the role
7. Creating S3 buckets for System Manager log store
8. Creating VPC endpoints
9. Creating instances that are associated with the role and SG
10. Verifying that System Manager and Session Manager are functioning with the instances

The next section details FBA process.

FBA Process

(Omitted)

Using System Manager

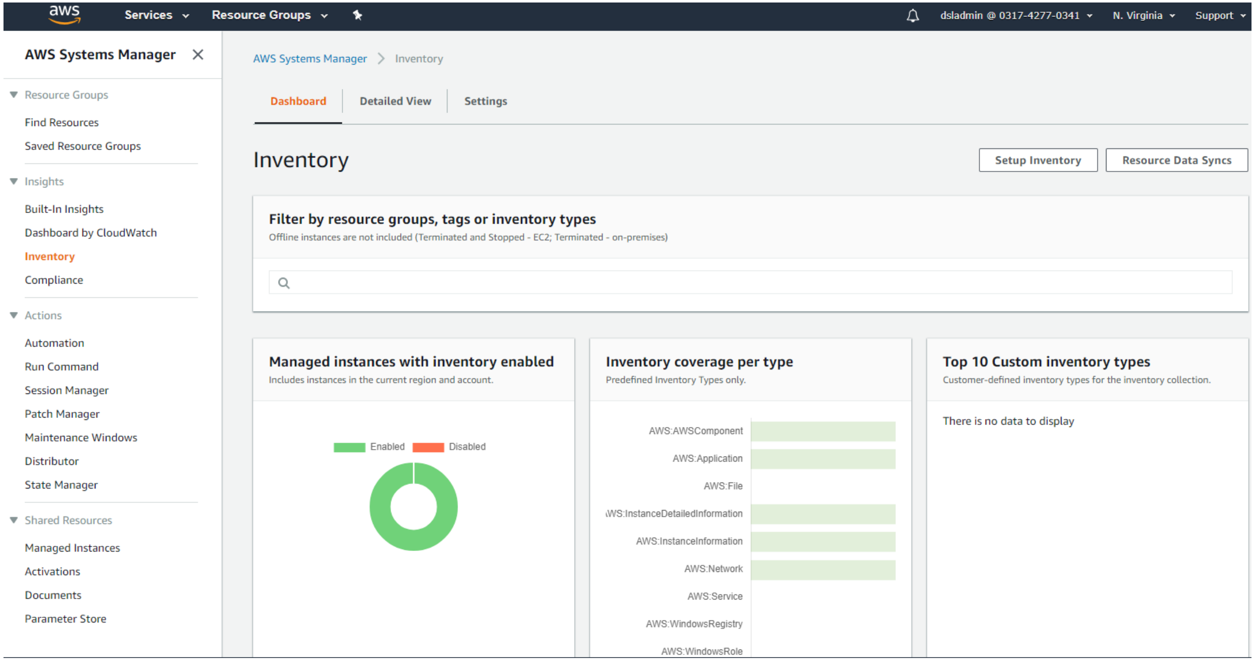
SM has two user interfaces, one is GUI by AWS Console; the other is command line interface, by AWS CLI.

System Manager (SM) GUI Interface by AWS Console

In AWS Console, SM is listed under Management & Governance. Selecting SM by clicking on it, SM page comes up as shown in Figure 1 below. From the left navigation menu on this page, you can access Inventory under Insights, Session Manager under Actions, and Managed Instances under Shared Resources

SM Insights - Inventory

Figure 1 – Inventory in N. Virginia (1)



Inventory

Filter by resource groups, tags or inventory types

Offline instances are not included (Terminated and Stopped - EC2; Terminated - on-premises)

Managed instances with inventory enabled

Includes instances in the current region and account.

Inventory coverage per type

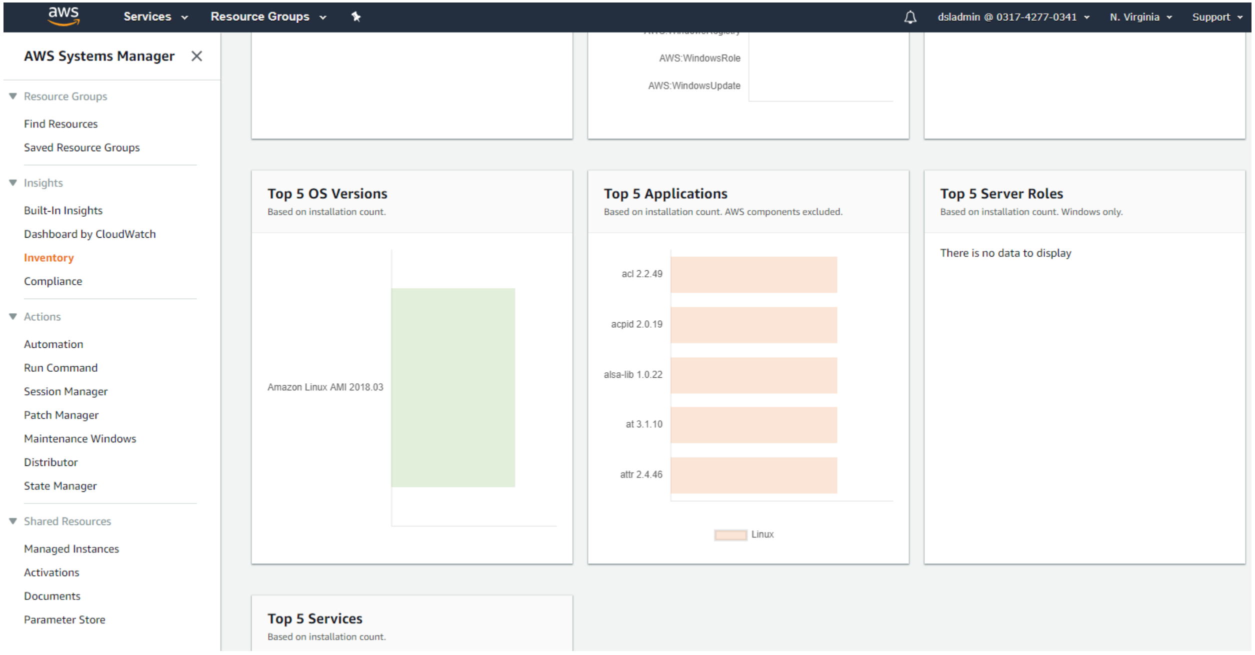
Predefined Inventory Types only.

Top 10 Custom inventory types

Customer-defined inventory types for the inventory collection.

There is no data to display

Figure 2 – Inventory in N. Virginia (2)



Top 5 OS Versions
Based on installation count.

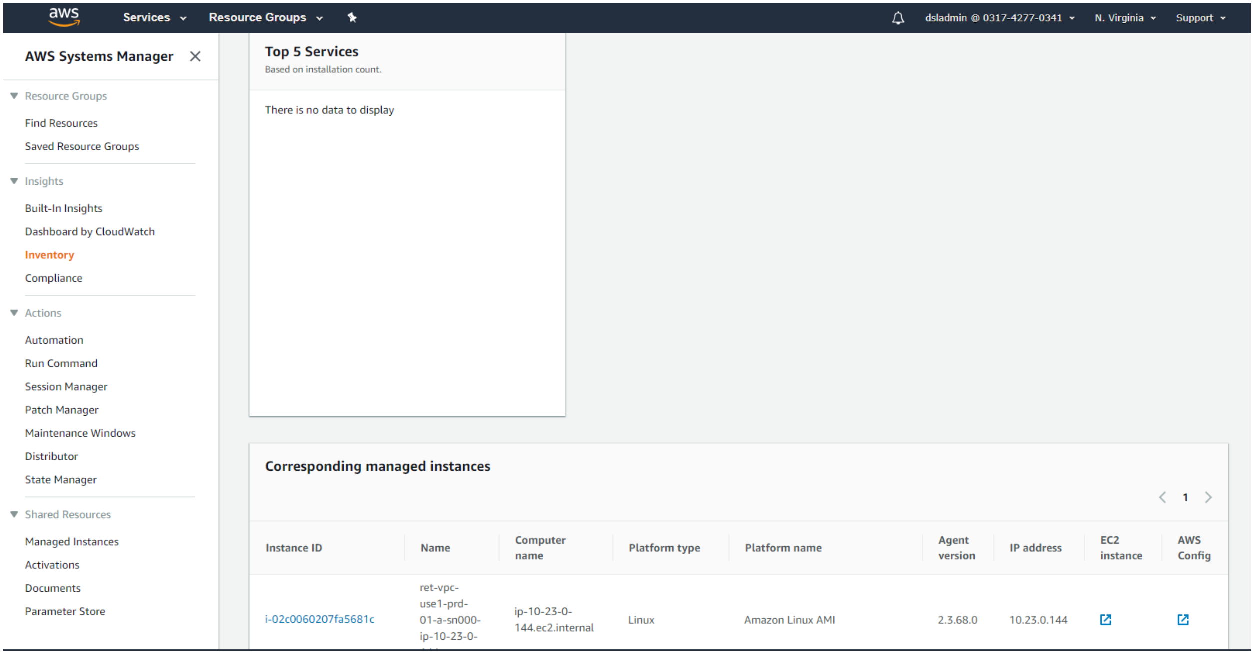
Top 5 Applications
Based on installation count. AWS components excluded.

Top 5 Server Roles
Based on installation count. Windows only.

There is no data to display

Top 5 Services
Based on installation count.

Figure 3 – Inventory in N. Virginia (3)



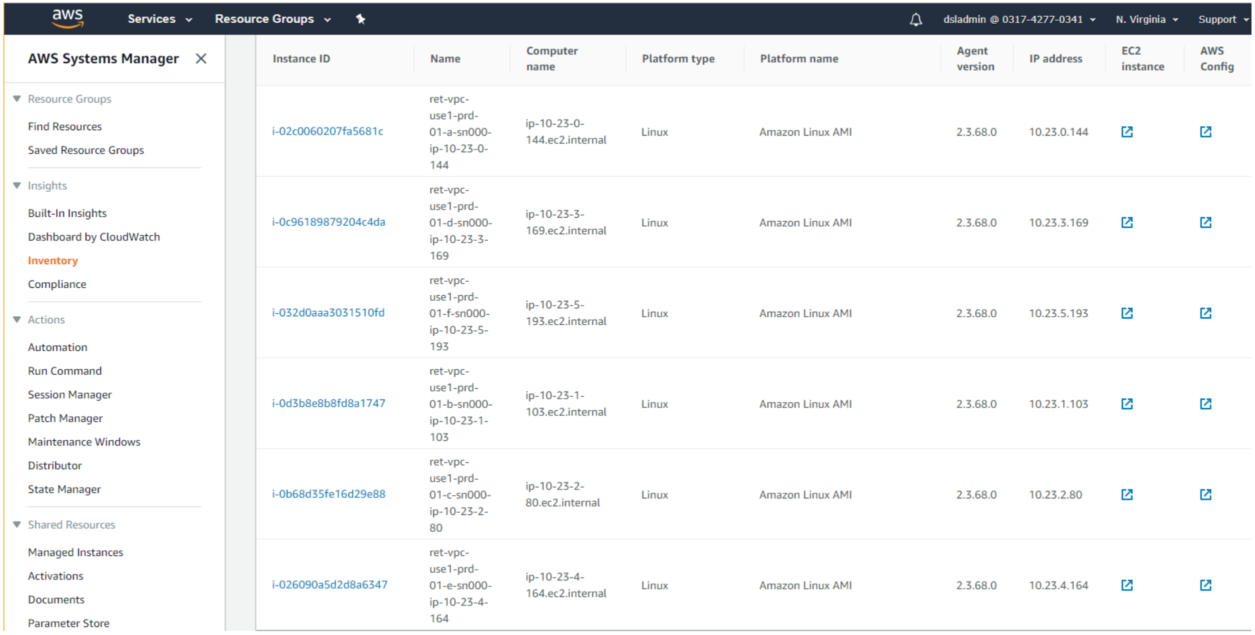
Top 5 Services
Based on installation count.

There is no data to display

Corresponding managed instances

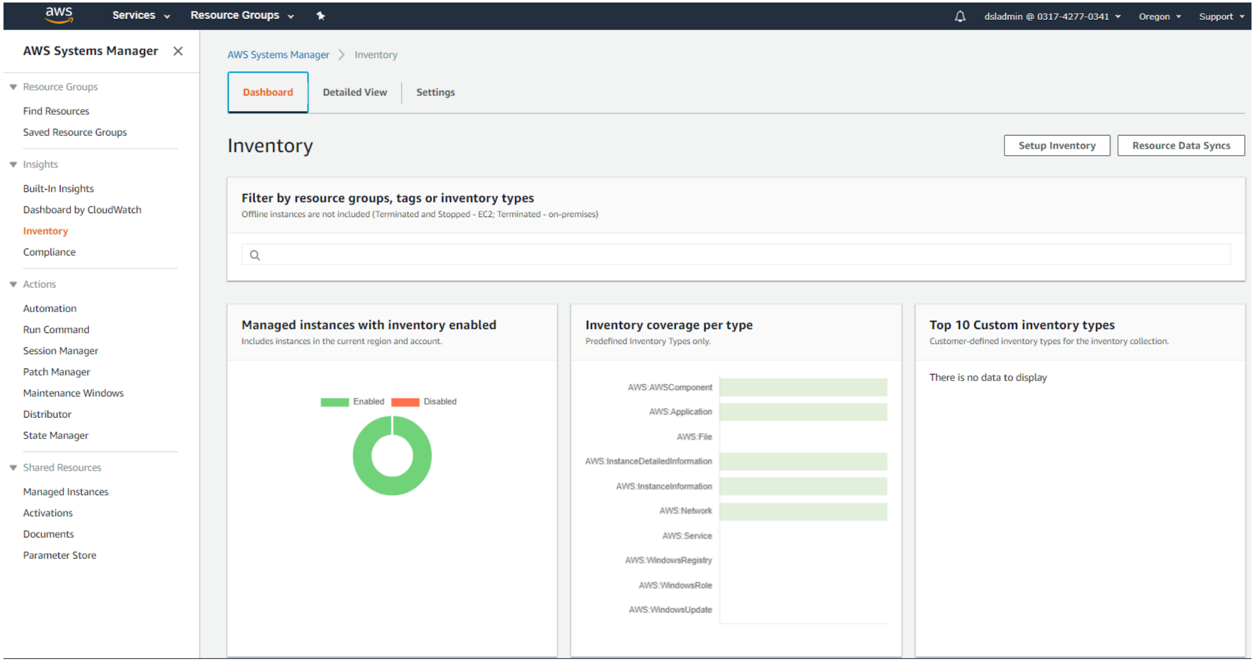
Instance ID	Name	Computer name	Platform type	Platform name	Agent version	IP address	EC2 instance	AWS Config
i-02c0060207fa5681c	ret-vpc-use1-prd-01-a-sn000-ip-10-23-0-	ip-10-23-0-144.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.0.144	↗	↗

Figure 4 – Inventory in N. Virginia (4)



Instance ID	Name	Computer name	Platform type	Platform name	Agent version	IP address	EC2 instance	AWS Config
i-02c0060207fa5681c	ret-vpc-use1-prd-01-a-sn000-ip-10-23-0-144	ip-10-23-0-144.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.0.144	↗	↗
i-0c96189879204c4da	ret-vpc-use1-prd-01-d-sn000-ip-10-23-3-169	ip-10-23-3-169.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.3.169	↗	↗
i-032d0aaa3031510fd	ret-vpc-use1-prd-01-f-sn000-ip-10-23-5-193	ip-10-23-5-193.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.5.193	↗	↗
i-0d3b8e8b8fd8a1747	ret-vpc-use1-prd-01-b-sn000-ip-10-23-1-103	ip-10-23-1-103.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.1.103	↗	↗
i-0b68d35fe16d29e88	ret-vpc-use1-prd-01-c-sn000-ip-10-23-2-80	ip-10-23-2-80.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.2.80	↗	↗
i-026090a5d2d8a6347	ret-vpc-use1-prd-01-e-sn000-ip-10-23-4-164	ip-10-23-4-164.ec2.internal	Linux	Amazon Linux AMI	2.3.68.0	10.23.4.164	↗	↗

Figure 5 – Oregon Inventory Dashboard (1)



Inventory

Filter by resource groups, tags or inventory types

Offline instances are not included (Terminated and Stopped - EC2; Terminated - on-premises)

Managed instances with inventory enabled

Includes instances in the current region and account.

Inventory coverage per type

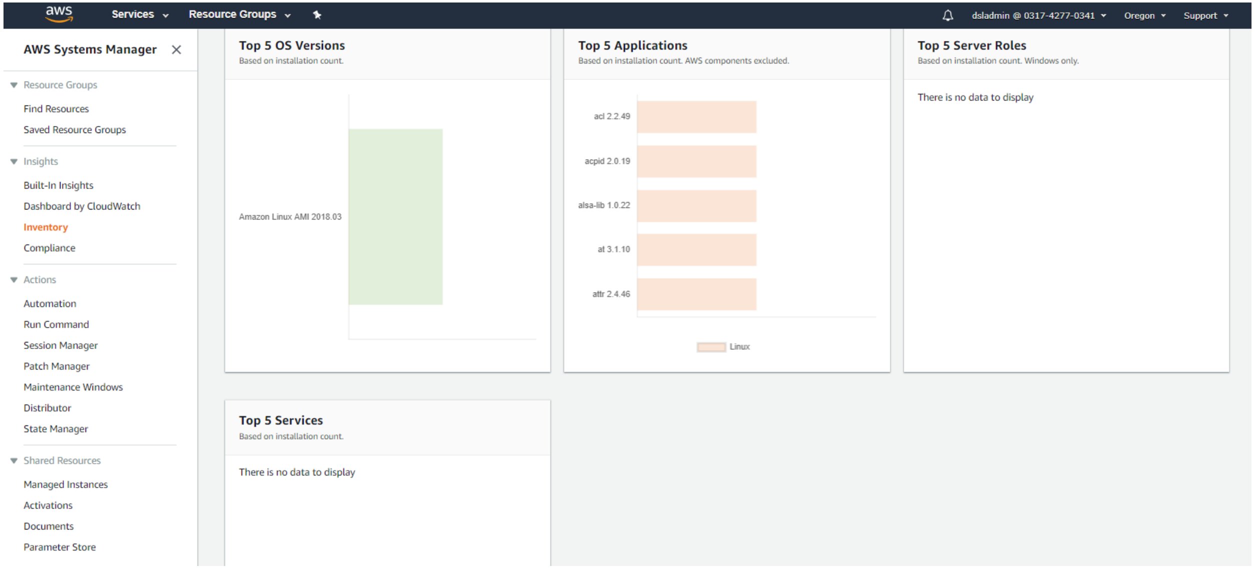
Predefined Inventory Types only.

Top 10 Custom inventory types

Customer-defined inventory types for the inventory collection.

There is no data to display

Figure 6 - Oregon Inventory Dashboard (2)



Top 5 OS Versions
Based on installation count.

Top 5 Applications
Based on installation count. AWS components excluded.

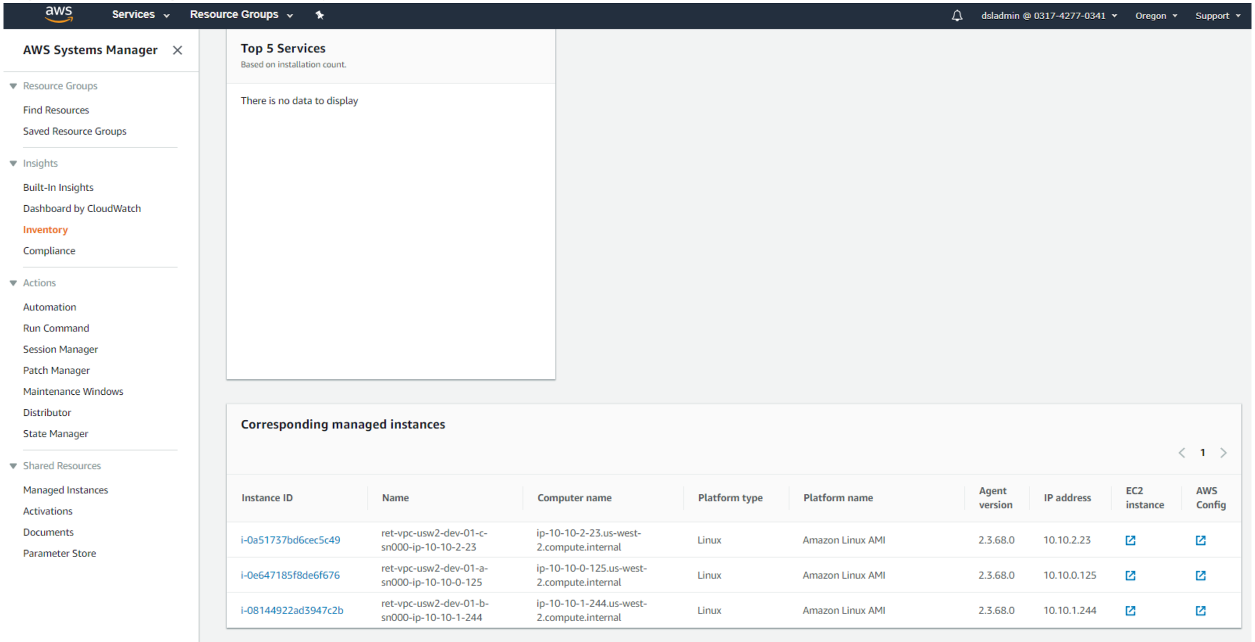
Top 5 Server Roles
Based on installation count. Windows only.

There is no data to display

Top 5 Services
Based on installation count.

There is no data to display

Figure 7 - Oregon Inventory Dashboard (3)



Top 5 Services
Based on installation count.

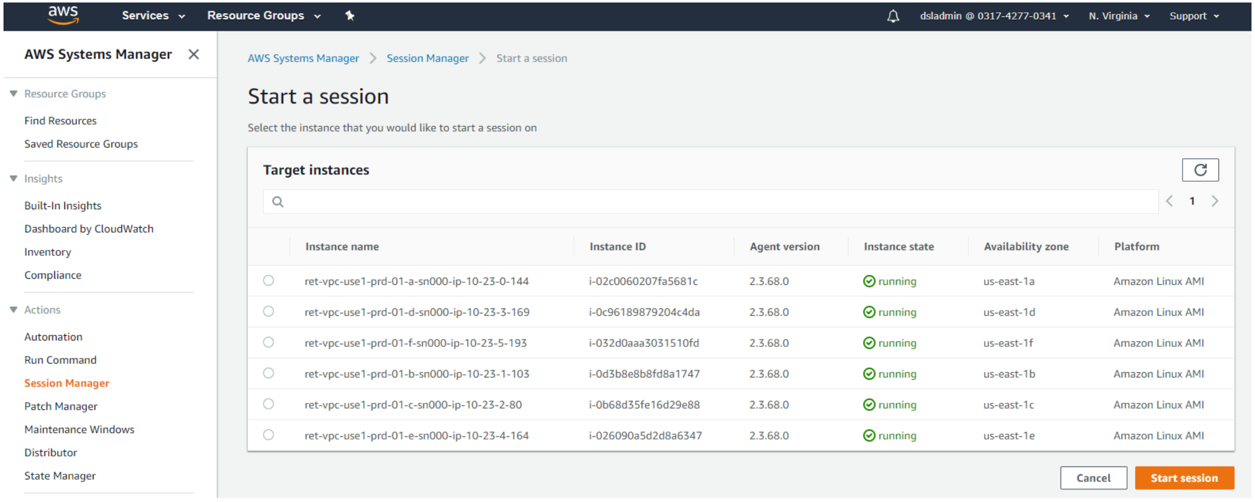
There is no data to display

Corresponding managed instances

Instance ID	Name	Computer name	Platform type	Platform name	Agent version	IP address	EC2 instance	AWS Config
i-0a51737bd6cec5c49	ret-vpc-usw2-dev-01-c-sn000-ip-10-10-2-23	ip-10-10-2-23.us-west-2.compute.internal	Linux	Amazon Linux AMI	2.3.68.0	10.10.2.23	↗	↗
i-0e647185f8de6f676	ret-vpc-usw2-dev-01-a-sn000-ip-10-10-0-125	ip-10-10-0-125.us-west-2.compute.internal	Linux	Amazon Linux AMI	2.3.68.0	10.10.0.125	↗	↗
i-08144922ad3947c2b	ret-vpc-usw2-dev-01-b-sn000-ip-10-10-1-244	ip-10-10-1-244.us-west-2.compute.internal	Linux	Amazon Linux AMI	2.3.68.0	10.10.1.244	↗	↗

SM Actions - Session Manager

Figure 8 – N. Virginia Session Manager Instance Listing



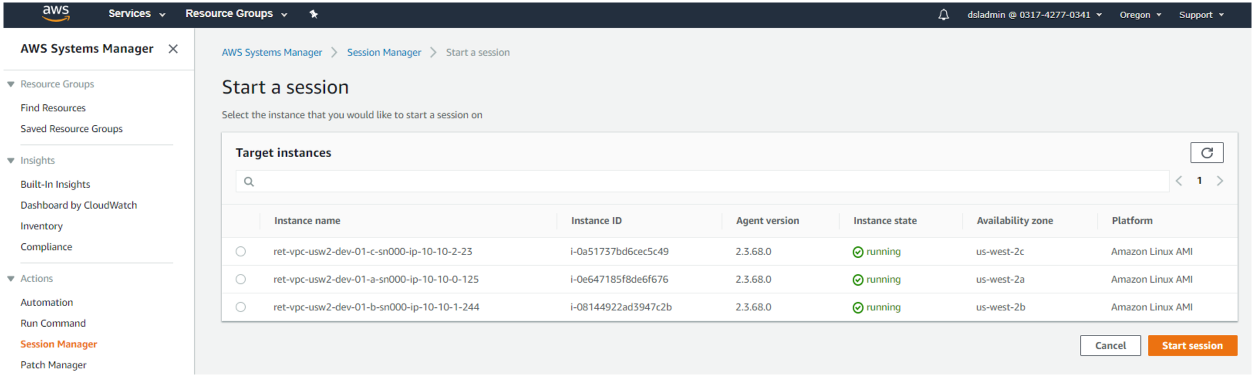
Start a session

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

Target instances

	Instance name	Instance ID	Agent version	Instance state	Availability zone	Platform
○	ret-vpc-use1-prd-01-a-sn000-ip-10-23-0-144	i-02c0060207fa5681c	2.3.68.0	✓ running	us-east-1a	Amazon Linux AMI
○	ret-vpc-use1-prd-01-d-sn000-ip-10-23-3-169	i-0c96189879204c4da	2.3.68.0	✓ running	us-east-1d	Amazon Linux AMI
○	ret-vpc-use1-prd-01-f-sn000-ip-10-23-5-193	i-032d0aaa3031510fd	2.3.68.0	✓ running	us-east-1f	Amazon Linux AMI
○	ret-vpc-use1-prd-01-b-sn000-ip-10-23-1-103	i-0d3b8e8b8fd8a1747	2.3.68.0	✓ running	us-east-1b	Amazon Linux AMI
○	ret-vpc-use1-prd-01-c-sn000-ip-10-23-2-80	i-0b68d35fe16d29e88	2.3.68.0	✓ running	us-east-1c	Amazon Linux AMI
○	ret-vpc-use1-prd-01-e-sn000-ip-10-23-4-164	i-026090a5d2d8a6347	2.3.68.0	✓ running	us-east-1e	Amazon Linux AMI

Figure 9 – Oregon Session Manager Instance Listing



Start a session

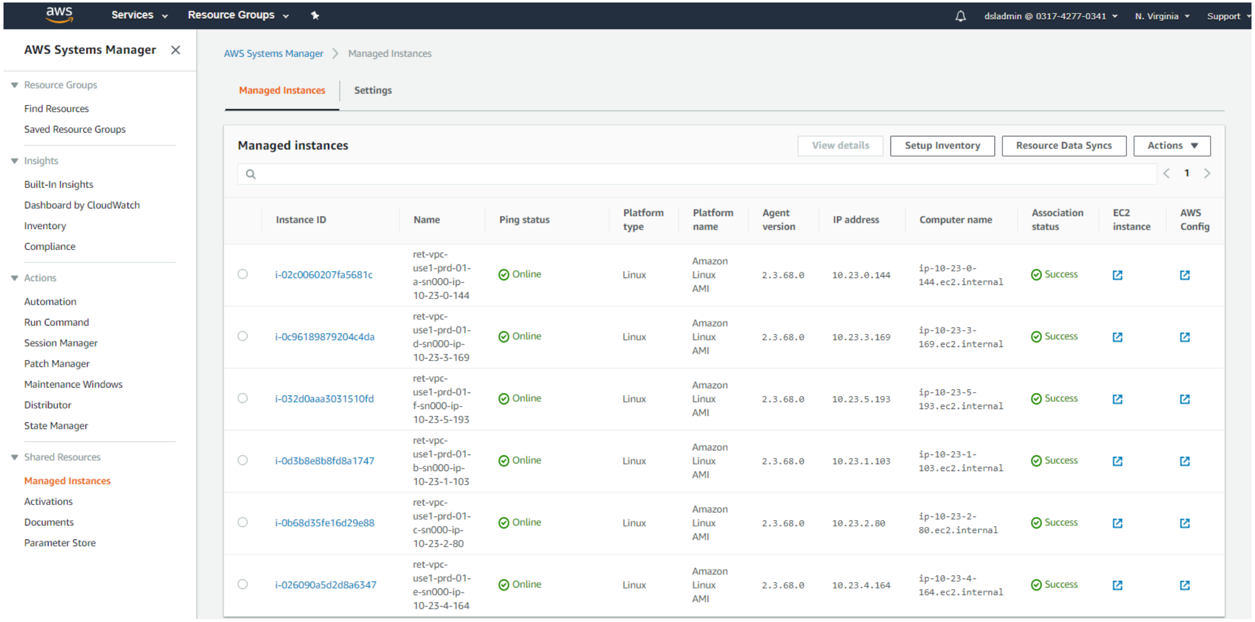
Select the instance that you would like to start a session on

Target instances

	Instance name	Instance ID	Agent version	Instance state	Availability zone	Platform
○	ret-vpc-usw2-dev-01-c-sn000-ip-10-10-2-23	i-0a51737bd6cec5c49	2.3.68.0	✓ running	us-west-2c	Amazon Linux AMI
○	ret-vpc-usw2-dev-01-a-sn000-ip-10-10-0-125	i-0e647185f8de6f676	2.3.68.0	✓ running	us-west-2a	Amazon Linux AMI
○	ret-vpc-usw2-dev-01-b-sn000-ip-10-10-1-244	i-08144922ad3947c2b	2.3.68.0	✓ running	us-west-2b	Amazon Linux AMI

SM Shared Resources - Managed Instances

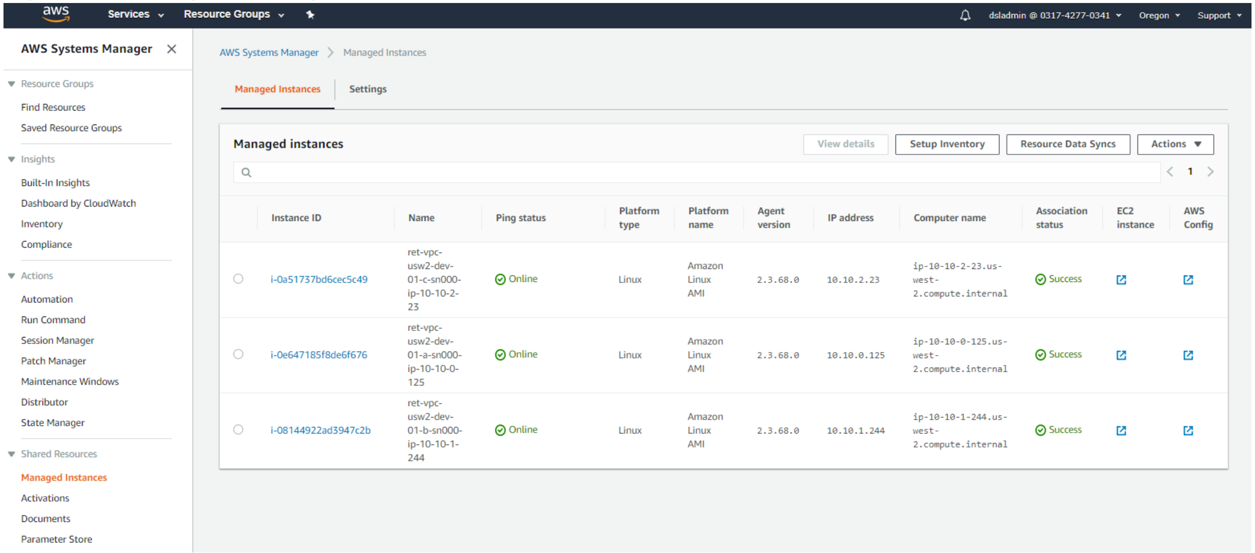
Figure 10 – N. Virginia Managed Instances



Managed instances

	Instance ID	Name	Ping status	Platform type	Platform name	Agent version	IP address	Computer name	Association status	EC2 instance	AWS Config
○	i-02c0060207fa5681c	ret-vpc-use1-prd-01-a-sn000-ip-10-23-0-144	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.23.0.144	ip-10-23-0-144.ec2.internal	✓ Success	↗	↗
○	i-0c96189879204c4da	ret-vpc-use1-prd-01-d-sn000-ip-10-23-3-169	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.23.3.169	ip-10-23-3-169.ec2.internal	✓ Success	↗	↗
○	i-032d0aaa3031510fd	ret-vpc-use1-prd-01-f-sn000-ip-10-23-5-193	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.23.5.193	ip-10-23-5-193.ec2.internal	✓ Success	↗	↗
○	i-0d3b8e8b8fd8a1747	ret-vpc-use1-prd-01-b-sn000-ip-10-23-1-103	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.23.1.103	ip-10-23-1-103.ec2.internal	✓ Success	↗	↗
○	i-0b68d35fe16d29e88	ret-vpc-use1-prd-01-c-sn000-ip-10-23-2-80	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.23.2.80	ip-10-23-2-80.ec2.internal	✓ Success	↗	↗
○	i-026090a5d2d8a6347	ret-vpc-use1-prd-01-e-sn000-ip-10-23-4-164	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.23.4.164	ip-10-23-4-164.ec2.internal	✓ Success	↗	↗

Figure 11 – Oregon Managed Instances



Managed instances

	Instance ID	Name	Ping status	Platform type	Platform name	Agent version	IP address	Computer name	Association status	EC2 instance	AWS Config
○	i-0a51737bd6cec5c49	ret-vpc-usw2-dev-01-c-sn000-ip-10-10-2-23	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.10.2.23	ip-10-10-2-23.us-west-2.compute.internal	✓ Success	↗	↗
○	i-0e647185f8de6f676	ret-vpc-usw2-dev-01-a-sn000-ip-10-10-0-125	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.10.0.125	ip-10-10-0-125.us-west-2.compute.internal	✓ Success	↗	↗
○	i-08144922ad3947c2b	ret-vpc-usw2-dev-01-b-sn000-ip-10-10-1-244	✓ Online	Linux	Amazon Linux AMI	2.3.68.0	10.10.1.244	ip-10-10-1-244.us-west-2.compute.internal	✓ Success	↗	↗

System Manager (SM) Command Line Interface by AWS CLI

Getting N. Virginia inventory excluding terminated instances

```
[awsdsllcadmin@linux721 ~]$ aws --region us-east-1 ssm get-inventory --output text | grep CONTENT |
grep -v Terminated
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-23-4-164.ec2.internal     i-026090a5d2d8a6347
10.23.4.164     Amazon Linux AMI       Linux   2018.03     EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-23-0-144.ec2.internal     i-02c0060207fa5681c
10.23.0.144     Amazon Linux AMI       Linux   2018.03     EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-23-5-193.ec2.internal     i-032d0aaa3031510fd
10.23.5.193     Amazon Linux AMI       Linux   2018.03     EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-23-2-80.ec2.internal     i-0b68d35fe16d29e88
10.23.2.80     Amazon Linux AMI       Linux   2018.03     EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-23-3-169.ec2.internal     i-0c96189879204c4da
10.23.3.169     Amazon Linux AMI       Linux   2018.03     EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-23-1-103.ec2.internal     i-0d3b8e8b8fd8a1747
10.23.1.103     Amazon Linux AMI       Linux   2018.03     EC2Instance
[awsdsllcadmin@linux721 ~]$

[awsdsllcadmin@linux721 ~]$ aws --region us-east-1 ec2 describe-instances | grep InstanceId
                "InstanceId": "i-026090a5d2d8a6347",
                "InstanceId": "i-0d3b8e8b8fd8a1747",
                "InstanceId": "i-02c0060207fa5681c",
                "InstanceId": "i-0c96189879204c4da",
                "InstanceId": "i-032d0aaa3031510fd",
                "InstanceId": "i-0b68d35fe16d29e88",

[awsdsllcadmin@linux721 ~]$ aws --region us-east-1 ssm describe-instance-associations-status --output
text --instance-id i-026090a5d2d8a6347
INSTANCEASSOCIATIONSTATUSINFOS  eac1f353-b4c1-4a7e-b1bd-cd54a2cd3cc1    Inventory-Association
1       1       1551988912.0    1 out of 1 plugin processed, 1 success, 0 failed, 0 timedout, 0 skipped.    i-
026090a5d2d8a6347     AWS-GatherSoftwareInventory     Success
S3OUTPUTURL     ret-s3-use1-sm-log-sync-bucket/i-026090a5d2d8a6347/eac1f353-b4c1-4a7e-b1bd-
cd54a2cd3cc1/2019-03-07T19-57-12.045Z
[awsdsllcadmin@linux721 ~]$
```

Getting Oregon inventory excluding terminated instances

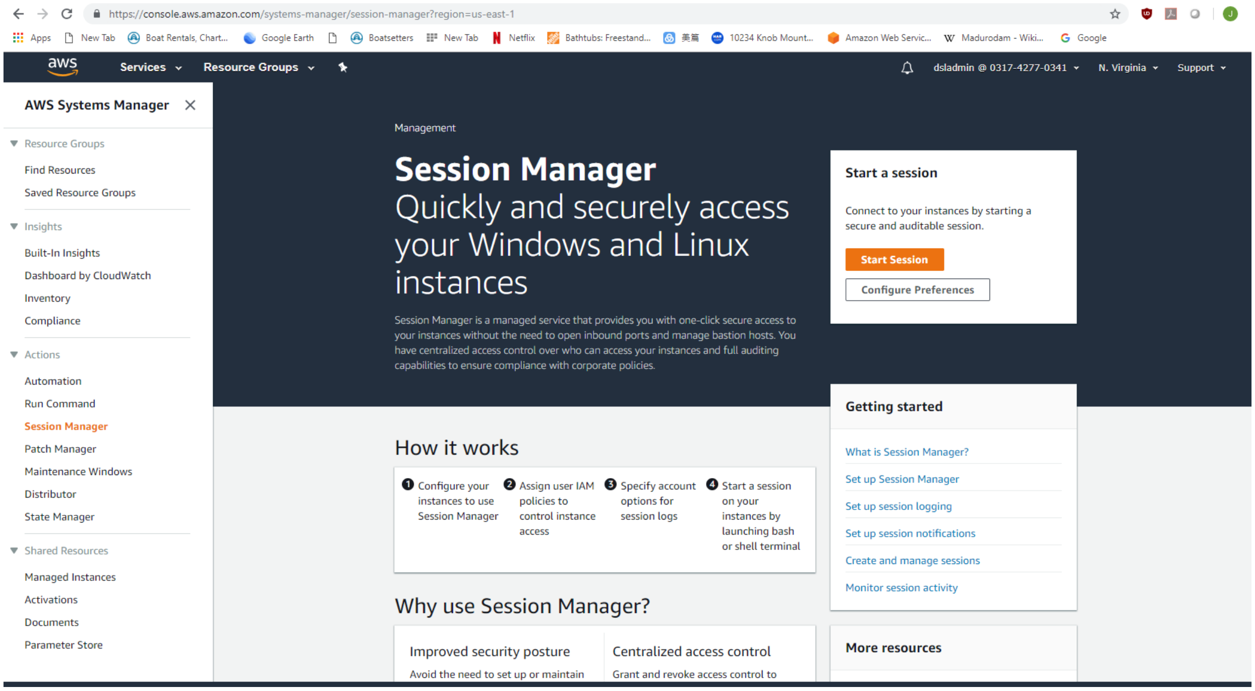
```
[awsdsllcadmin@linux721 ~]$ aws --region us-west-2 ssm get-inventory --output text | grep CONTENT |
grep -v Terminated
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-10-1-244.us-west-2.compute.internal      i-
08144922ad3947c2b     10.10.1.244     Amazon Linux AMI   Linux    2018.03 EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-10-2-23.us-west-2.compute.internal      i-
0a51737bd6cec5c49     10.10.2.23     Amazon Linux AMI   Linux    2018.03 EC2Instance
CONTENT amazon-ssm-agent      2.3.68.0      ip-10-10-0-125.us-west-2.compute.internal      i-
0e647185f8de6f676     10.10.0.125     Amazon Linux AMI   Linux    2018.03 EC2Instance
```

```
[awsdsllcadmin@linux721 ~]$

[awsdsllcadmin@linux721 ~]$ aws --region us-west-2 ec2 describe-instances | grep InstanceId
            "InstanceId": "i-08144922ad3947c2b",
            "InstanceId": "i-0e647185f8de6f676",
            "InstanceId": "i-0a51737bd6cec5c49",
[awsdsllcadmin@linux721 ~]$

[awsdsllcadmin@linux721 ~]$ aws --region us-west-2 ssm describe-instance-associations-status --output
text --instance-id i-08144922ad3947c2b
INSTANCEASSOCIATIONSTATUSINFOS  df9f2d2b-1fd7-42d1-9b0b-a64c4fbd5372   Inventory-Association
1      1      1551988145.0    1 out of 1 plugin processed, 1 success, 0 failed, 0 timedout, 0 skipped.    i-
08144922ad3947c2b    AWS-GatherSoftwareInventory    Success
S3OUTPUTURL    ret-s3-usw2-sm-log-sync-bucket/i-08144922ad3947c2b/df9f2d2b-1fd7-42d1-9b0b-
a64c4fbd5372/2019-03-07T19-41-56.027Z
[awsdsllcadmin@linux721 ~]$
```

Using Session Manager

Click on Session Manager in AWS System Manager, Session Manager's page opens up. See Figure 12 below.

Figure 12 – Session Manager



Session Manager
Quickly and securely access your Windows and Linux instances

Session Manager is a managed service that provides you with one-click secure access to your instances without the need to open inbound ports and manage bastion hosts. You have centralized access control over who can access your instances and full auditing capabilities to ensure compliance with corporate policies.

Start a session

Connect to your instances by starting a secure and auditable session.

How it works

1. Configure your instances to use Session Manager
2. Assign user IAM policies to control instance access
3. Specify account options for session logs
4. Start a session on your instances by launching bash or shell terminal

Getting started

- What is Session Manager?
- Set up Session Manager
- Set up session logging
- Set up session notifications
- Create and manage sessions
- Monitor session activity

Why use Session Manager?

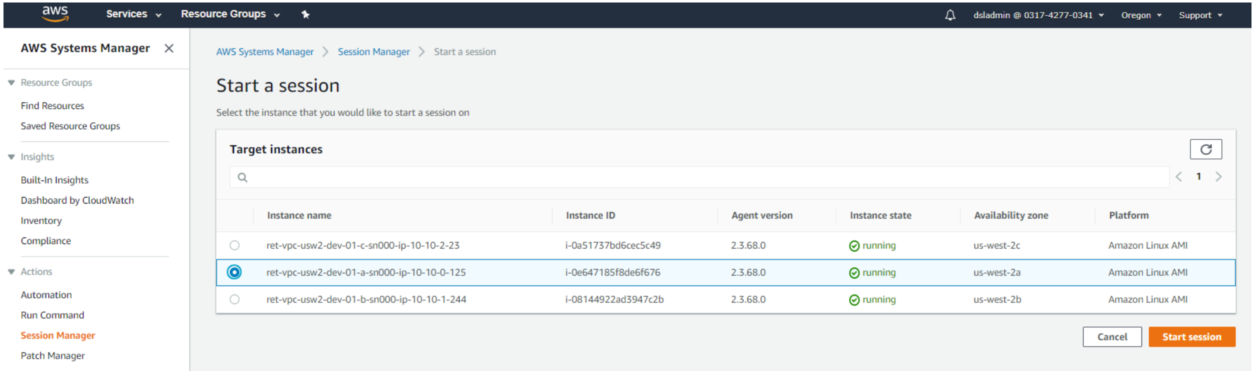
Improved security posture
Avoid the need to set up or maintain

Centralized access control
Grant and revoke access control to

More resources

Click on Start Session, a list of target instances appears. Select an instance, and click on Start session. See Figure 13 below.

Figure 13 – Target instances



Start a session

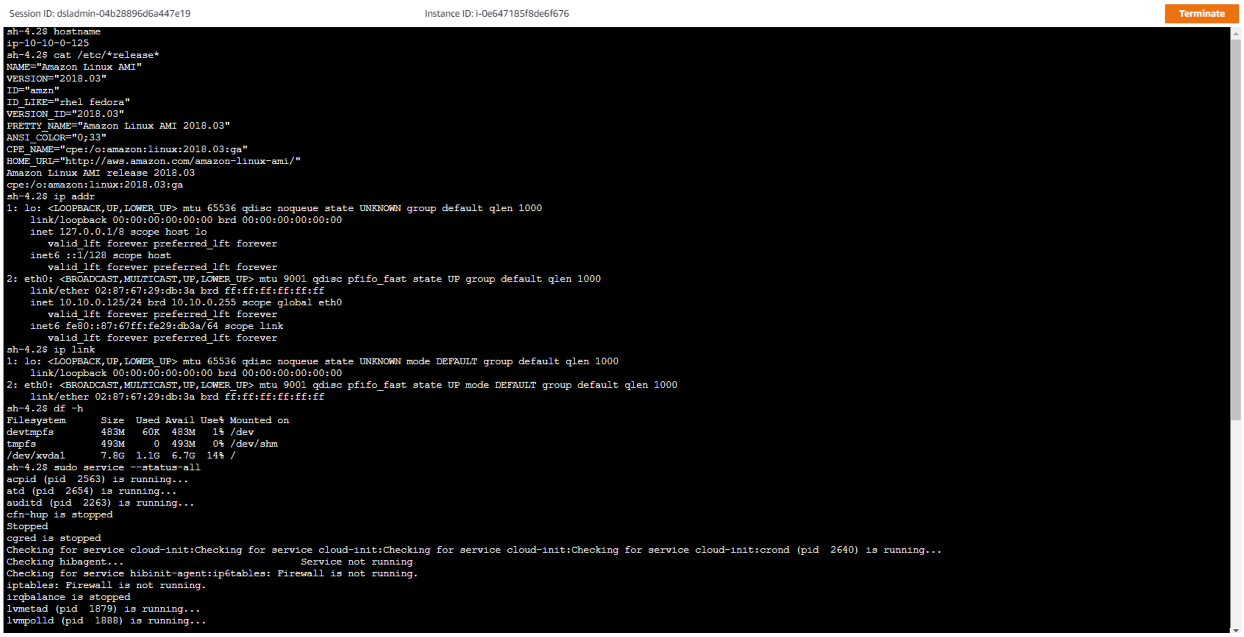
Select the instance that you would like to start a session on

Target instances

	Instance name	Instance ID	Agent version	Instance state	Availability zone	Platform
○	ret-vpc-usw2-dev-01-c-sn000-ip-10-10-2-23	i-0a51737bd6cec5c49	2.3.68.0	✓ running	us-west-2c	Amazon Linux AMI
●	ret-vpc-usw2-dev-01-a-sn000-ip-10-10-0-125	i-0e647185f8de6f676	2.3.68.0	✓ running	us-west-2a	Amazon Linux AMI
○	ret-vpc-usw2-dev-01-b-sn000-ip-10-10-1-244	i-08144922ad3947c2b	2.3.68.0	✓ running	us-west-2b	Amazon Linux AMI

Figure 14 below shows what it looks like when a session is started with an EC2 instance. In this case, the EC2 instance selected is a private instance with IP address of 10.10.0.125, with no Internet access. The session opens up a bash shell in the instance.

Figure 14 - Session shell screen


Session ID: dsladmin-04b28896d6a447e19

Instance ID: i-0e647185f8de6f676

```
sh-4.2$ hostname
ip-10-10-0-125
sh-4.2$ cat /etc/*release*
NAME="Amazon Linux AMI"
VERSION="2018.03"
ID="amzn"
ID_LIKE="rhel fedora"
VERSION_ID="2018.03"
PRETTY_NAME="Amazon Linux AMI 2018.03"
ANSI_COLOR="0;33"
CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"
HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
Amazon Linux AMI release 2018.03
cpe:/o:amazon:linux:2018.03:ga
sh-4.2$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:87:67:29:db:3a brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.125/24 brd 10.10.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::87:67ff:fe29:db3a/64 scope link
       valid_lft forever preferred_lft forever
sh-4.2$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 02:87:67:29:db:3a brd ff:ff:ff:ff:ff:ff
sh-4.2$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        483M   60K  483M   1% /dev
tmpfs           493M     0  493M   0% /dev/shm
/dev/xvda1      7.8G  1.1G  6.7G  14% /
sh-4.2$ sudo service --status-all
acpid (pid  2563) is running...
atd (pid  2654) is running...
auditd (pid  2263) is running...
cfn-hup is stopped
Stopped
cgred is stopped
Checking for service cloud-init:Checking for service cloud-init:Checking for service cloud-init:Checking for service cloud-init:crond (pid  2640) is running...
Checking hibagent...                                   Service not running
Checking for service hibinit-agent:ip6tables: Firewall is not running.
iptables: Firewall is not running.
irqbalance is stopped
lvmetad (pid  1879) is running...
lvmpolld (pid  1888) is running...
```

Output 1 below shows the screen output yielded from a few query commands typed in the session.

Output 1 – Session screen output

```
sh-4.2$ hostname
ip-10-10-0-125

sh-4.2$ cat /etc/*release*
NAME="Amazon Linux AMI"
VERSION="2018.03"
ID="amzn"
ID_LIKE="rhel fedora"
```

```
VERSION_ID="2018.03"
PRETTY_NAME="Amazon Linux AMI 2018.03"
ANSI_COLOR="0;33"
CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"
HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
Amazon Linux AMI release 2018.03
cpe:/o:amazon:linux:2018.03:ga

sh-4.2$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 02:87:67:29:db:3a brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.125/24 brd 10.10.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::87:67ff:fe29:db3a/64 scope link
       valid_lft forever preferred_lft forever

sh-4.2$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode DEFAULT
group default qlen 1000
    link/ether 02:87:67:29:db:3a brd ff:ff:ff:ff:ff:ff

sh-4.2$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        483M   60K  483M   1% /dev
tmpfs           493M     0  493M   0% /dev/shm
/dev/xvda1      7.8G  1.1G  6.7G  14% /

sh-4.2$ sudo service --status-all
acpid (pid  2563) is running...
atd (pid  2654) is running...
auditd (pid  2263) is running...
cfn-hup is stopped
Stopped
cgred is stopped
Checking for service cloud-init:Checking for service cloud-init:Checking for service cloud-init:Checking for
service cloud-init:crond (pid  2640) is running...
Checking hibagent...                                 Service not running
Checking for service hibinit-agent:ip6tables: Firewall is not running.
iptables: Firewall is not running.
irqbalance is stopped
lvmetad (pid  1879) is running...
lvmpolld (pid  1888) is running...
dmeventd is stopped
mdmonitor is stopped
messagebus (pid  2383) is running...
netconsole module not loaded
Configured devices:
lo eth0
Currently active devices:
lo eth0
rpc.svcgssd is stopped
rpc.mountd is stopped
nfsd is stopped
rpc.rquotad is stopped
rpc.statd (pid  2349) is running...
ntpd (pid  2598) is running...
Process accounting is disabled.
quota_nld is stopped
rdisc is stopped
rngd (pid  2310) is running...
rpcbind (pid  2328) is running...
rpc.gssd is stopped
rpc.idmapd is stopped
rpc.svcgssd is stopped
rsyslogd (pid  2284) is running...
saslauthd is stopped
sendmail (pid  2619) is running...
sm-client (pid  2628) is running...
openssh-daemon (pid  2587) is running...
sh-4.2$
```

In the meantime, attempt to connect to the same instance by SSH from a local machine timed out because the target instance is a private instance. Connecting to a private instance by SSH requires a VPN connection open from the local machine to the VPC where the instance is running in. In this case, VPN is not there yet. See Output 2 below

Output 2 – Attempt to connect to a private instance by SSH from a local machine timed out

```
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-usw2-dev-01-keypair ec2-user@10.10.0.125
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-usw2-dev-01-keypair ec2-user@10.10.1.244
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-usw2-dev-01-keypair ec2-user@34.212.169.107
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-use1-prd-01-keypair ec2-user@10.23.0.144
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-use1-prd-01-keypair ec2-user@10.23.1.103
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-use1-prd-01-keypair ec2-user@10.23.2.80
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-use1-prd-01-keypair ec2-user@10.23.3.169
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-use1-prd-01-keypair ec2-user@10.23.4.164
ssh -i /home/awsdsllcadmin/.aws/ret-vpc-use1-prd-01-keypair ec2-user@34.204.205.152
```

```
[awsdsllcadmin@linux721 ~]$ ssh -i /home/awsdsllcadmin/.aws/ret-vpc-usw2-dev-01-keypair ec2-user@10.10.0.125
ssh: connect to host 10.10.0.125 port 22: Connection refused
[awsdsllcadmin@linux721 ~]$
```

Using AWS CLI Session Manager Plugin

Without VPN, installing AWS CLI and Session Manager Plugin on a local machine allows user to connect to a private instance like the one shown before. See Output 3 below.

Output 3 – Connecting to a private instance using AWS CLI Session Manager Plugin

Verify AWS CLI installed and version is higher than 1.16.12

```
[awsdsllcadmin@linux721 ~]$ aws --version
aws-cli/1.16.106 Python/3.7.2 Linux/3.10.0-693.5.2.el7.x86_64 botocore/1.12.96
[awsdsllcadmin@linux721 ~]$
```

Download session manager plugin

```
[awsdsllcadmin@linux721 ~]$ curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.rpm" -o "session-manager-plugin.rpm"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2368k  100 2368k    0     0  1428k      0  0:00:01  0:00:01 --:--:-- 1427k

[awsdsllcadmin@linux721 ~]$ ls -lt
total 2400
-rw-rw-r--. 1 awsdsllcadmin awsdsllcadmin 2425759 Mar  7 16:19 session-manager-plugin.rpm
…
```

Installing session manager plugin

```
[awsdsllcadmin@linux721 ~]$ sudo rpm -ivh session-manager-plugin.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:session-manager-plugin-1.0.37.0-1################################# [100%]
Created symlink from /etc/systemd/system/multi-user.target.wants/session-manager-plugin.service to /etc/systemd/system/session-manager-plugin.service.
```

Verifying that Session Manager Plugin has been installed successfully

```
[awsdsllcadmin@linux721 ~]$ session-manager-plugin

Session-Manager-Plugin is installed successfully. Use AWSCLI to start a session.

[awsdsllcadmin@linux721 ~]$
```

Listing target instances available to connect to

```
[awsdsllcadmin@linux721 ~]$ aws ec2 describe-instances | egrep "InstanceId|Value"
                "InstanceId": "i-08144922ad3947c2b",
                        "Value": "ret-vpc-usw2-dev-01-b-sn000-ip-10-10-1-244"
                "InstanceId": "i-0e647185f8de6f676",
                        "Value": "ret-vpc-usw2-dev-01-a-sn000-ip-10-10-0-125"
                "InstanceId": "i-0a51737bd6cec5c49",
                        "Value": "ret-vpc-usw2-dev-01-c-sn000-ip-10-10-2-23"
```

The instance in blue above was the instance that Session Manager connected to previously.

Starting a session

See Output 4 below

Output 4 – Staring a session using AWS CLI Session Manager Plugin

```
[awsdsllcadmin@linux721 ~]$ aws ssm start-session --target i-0e647185f8de6f676
Starting session with SessionId: dsladmin-09a442f9a93b47ed7
sh-4.2$ hostname
ip-10-10-0-125

sh-4.2$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 02:87:67:29:db:3a brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.125/24 brd 10.10.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::87:67ff:fe29:db3a/64 scope link
       valid_lft forever preferred_lft forever

sh-4.2$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode DEFAULT
group default qlen 1000
    link/ether 02:87:67:29:db:3a brd ff:ff:ff:ff:ff:ff

sh-4.2$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        483M   60K  483M   1% /dev
```

```
tmpfs         493M     0  493M   0% /dev/shm
/dev/xvda1     7.8G  1.1G  6.7G  14% /
```

```
sh-4.2$ cat /etc/*release*
NAME="Amazon Linux AMI"
VERSION="2018.03"
ID="amzn"
ID_LIKE="rhel fedora"
VERSION_ID="2018.03"
PRETTY_NAME="Amazon Linux AMI 2018.03"
ANSI_COLOR="0;33"
CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"
HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
Amazon Linux AMI release 2018.03
cpe:/o:amazon:linux:2018.03:ga
sh-4.2$
```

```
sh-4.2$ sudo service --status-all
acpid (pid  2563) is running...
atd (pid  2654) is running...
auditd (pid  2263) is running...
cfn-hup is stopped
Stopped
cgred is stopped
Checking for service cloud-init:Checking for service cloud-init:Checking for service cloud-init:Checking for service cloud-init:crond (pid  2640) is running...
Checking hibagent...                         Service not running
Checking for service hibinit-agent:ip6tables: Firewall is not running.
iptables: Firewall is not running.
irqbalance is stopped
lvmetad (pid  1879) is running...
lvmpolld (pid  1888) is running...
dmeventd is stopped
mdmonitor is stopped
messagebus (pid  2383) is running...
netconsole module not loaded
Configured devices:
lo eth0
Currently active devices:
lo eth0
rpc.svcgssd is stopped
rpc.mountd is stopped
nfsd is stopped
rpc.rquotad is stopped
rpc.statd (pid  2349) is running...
ntpd (pid  2598) is running...
```

```
Process accounting is disabled.
quota_nld is stopped
rdisc is stopped
rngd (pid  2310) is running...
rpcbind (pid  2328) is running...
rpc.gssd is stopped
rpc.idmapd is stopped
rpc.svcgssd is stopped
rsyslogd (pid  2284) is running...
saslauthd is stopped
sendmail (pid  2619) is running...
sm-client (pid  2628) is running...
openssh-daemon (pid  2587) is running...
sh-4.2$
```

As you may have noticed at the beginning of Output 4 above, a session id was created and shown upon the session start. The session id began with “dsladmin”, which was the IAM user name who started the session. The IAM user needs to have proper roles and permissions granted to him/her for the session to succeed or it will fail. In other words, user authentication and authorization (AA) is handled by AWS IAM when AWS CLI Session Manager Plugin is used to connect to a target instance, just like AA is managed by AWS IAM when System Manager/Session Manager in AWS Console GUI interface is used to connect to an instance.

What can Session Manager do and not do?

1. Session Manager allows a user to connect to a target instance even it is a private one, without VPN
2. Session Manager allows a user to start a session without additional authentication
3. Cloud administrator can control user access to Session Manager by granting or revoking proper roles and permissions for a user. Cloud admin exercises this control via AWS IAM. This way System Manager/Session Manager helps centralize managing user access to EC2 instances in the cloud.
4. Authorized users can connect to EC2 instances via Session Manager in AWS Console if the user is permitted console access.
5. Authorized users can also connect to EC2 instances using AWS CLI Session Manager Plugin, in the command line interface.
6. When using Session Manager to connect to an instance it opens up a session as an isolated shell in the target instance. In the session, no file transfers nor input/output redirect between the instance and the local machine are possible. From security perspective, this restriction is desirable as it limits what regular users can do. For legitimate power users, developers, operators, administrators, and the like, however, this restriction may be too severe so as to hinder or even prevent them from getting their work done.

Conclusion

This article introduced you to the basics of System Manager, with Session Manager included. In addition to Session Manager, Automation, Patch Manager, Distributor, and Maintenance Windows are in SM and designed for facilitating and automating various routine operation tasks. System Manager can interface with CloudWatch for monitoring, with Athena for in-depth log data analytics, with Glue for ETL for log data filtering. System Manager's viewing scope is region-based, for instance, by default its inventory dashboard displays managed instances and related information one region at a time. A consolidated view can be built by setting up Resource Data Syncs to bring log data not only across regions but also across multiple AWS accounts, and even data from on-premise systems and display them all in one pane of glass. AWS is constantly working on improving SM. If you are using AWS cloud providing IaaS to your internal users and external customers, or IaaS constitutes a bulk of your operations in AWS cloud, System Manager can definitely help.

Appendix – Creating an EC2 Instance Profile for Systems Manager

For Systems Manager to access managed instances, an EC2 Role for Simple Systems Manager is required.

To create this role, go to IAM Console, select Roles, select Create Role, Select type of trusted entity, select EC2, then select EC2 Role for Simple Systems Manager.

Or follow this navigation path:
IAM -> Roles -> Create Role -> Select type of trusted entity -> EC2 -> EC2 Role for Simple Systems Manager

When Tag page comes up, add "Name" for Key, <Account>-ec2-role-ssm for Value, for example,

shllc-ec2-role-ssm

where "shllc" is the account name

When prompted for entering a name for the role, repeat <Account>-ec2-rle-ssm

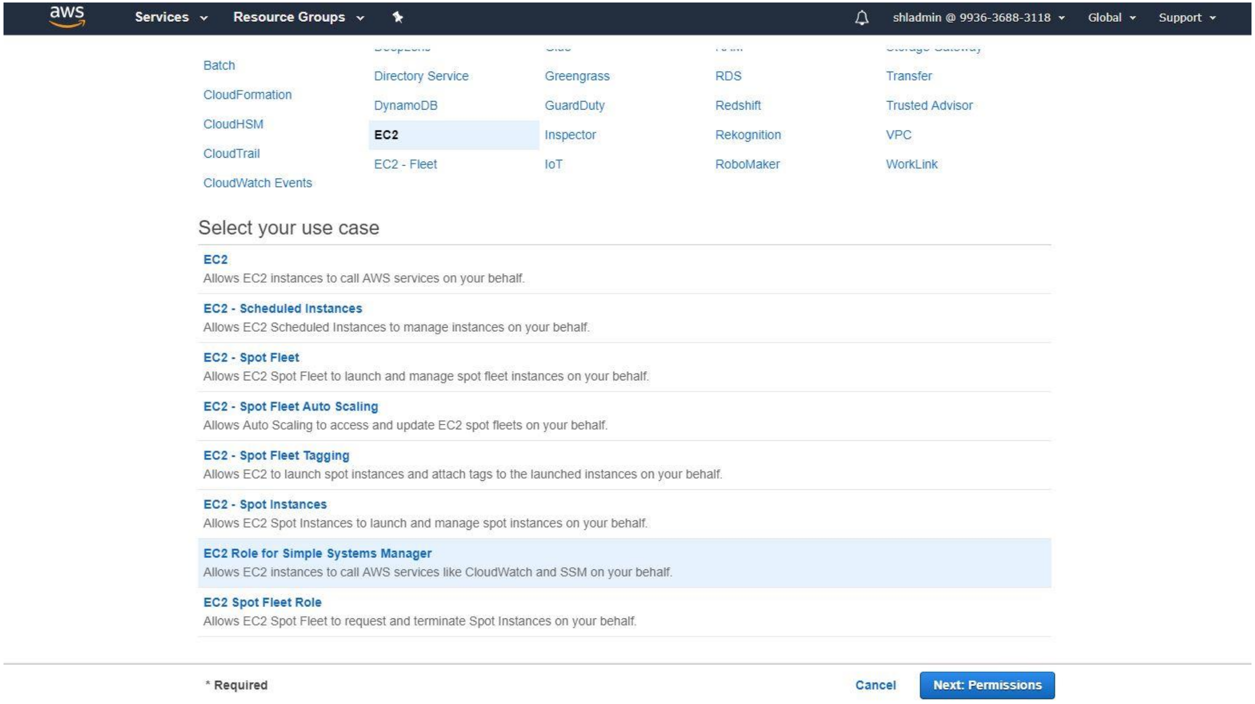
Click next to review, and create the role if no changes are required to redefine the role

Attach this role to existing instances
Specify this role when launching new instances

IAM Role is global in scope. This newly created role applies to instances in all VPCs in all regions in the account

Since this role becomes part of instance profile, the role is also referred to as Instance Profile

Figure 1 – Selecting EC2 Role for Simple Systems Manager when creating the role



Select your use case

EC2
Allows EC2 instances to call AWS services on your behalf.

EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.

EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

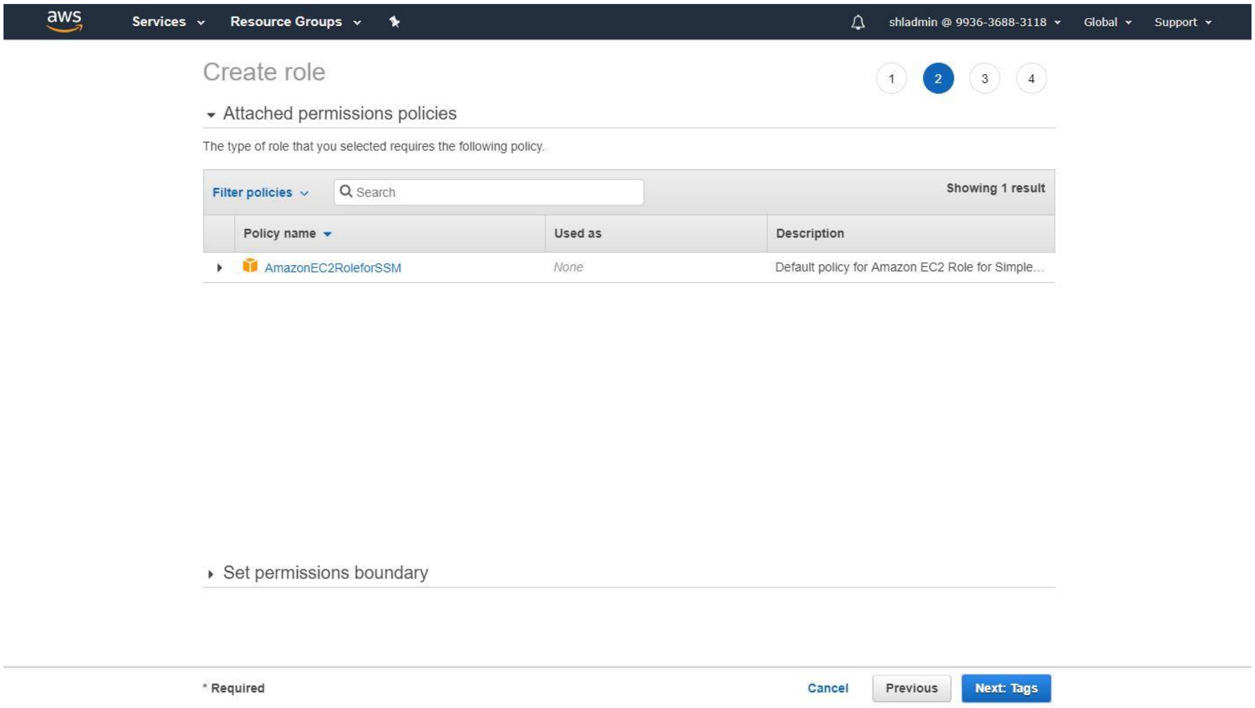
EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

EC2 Role for Simple Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and SSM on your behalf.

EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

Figure 2 – Attaching AmazonEC2RoleforSSM permissions policy to the role



Create role

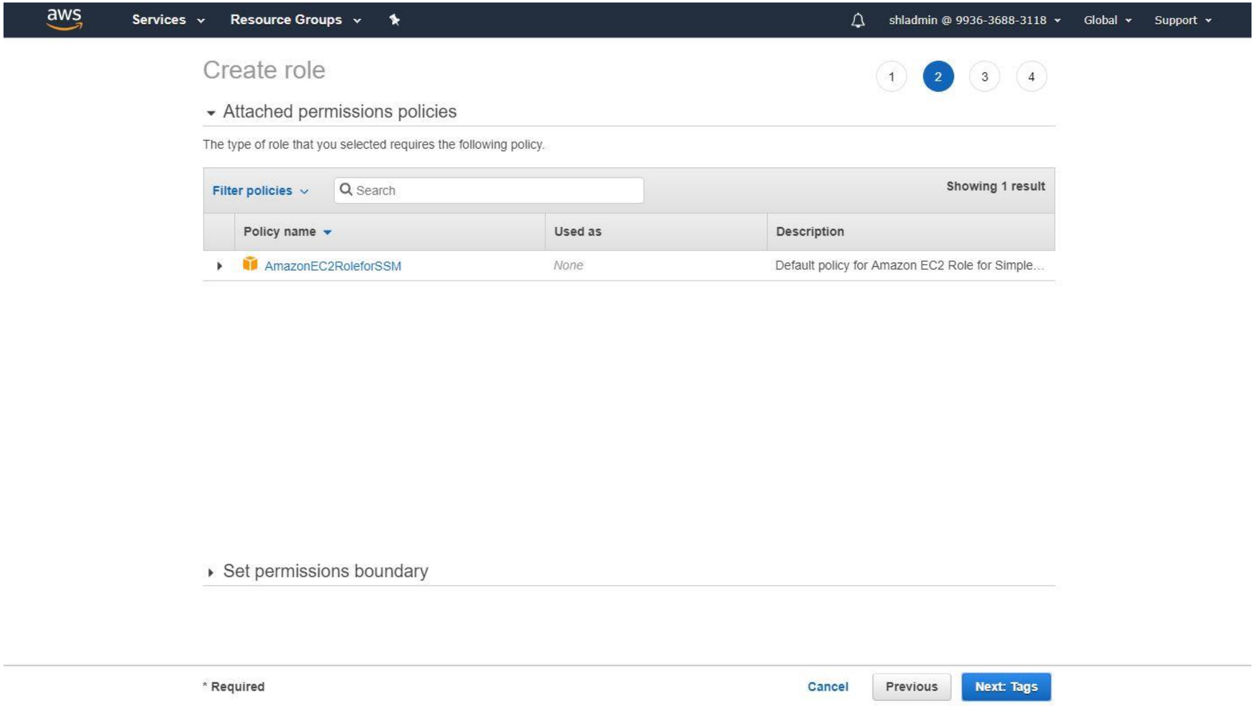
Attached permissions policies

The type of role that you selected requires the following policy.

Policy name	Used as	Description
AmazonEC2RoleforSSM	None	Default policy for Amazon EC2 Role for Simple…

Set permissions boundary

Figure 3 – Click Next: Tags

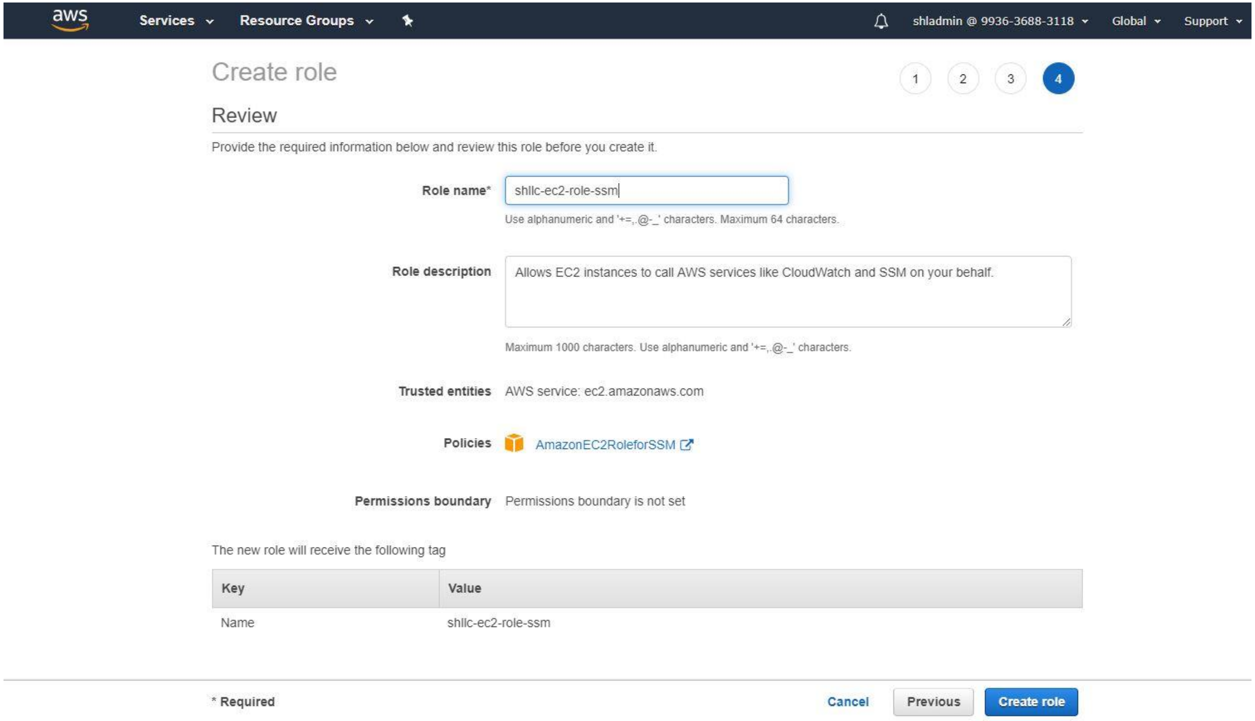


Attached permissions policies

The type of role that you selected requires the following policy.

Policy name	Used as	Description
AmazonEC2RoleforSSM	None	Default policy for Amazon EC2 Role for Simple...

Figure 4 – Naming the role



Review

Provide the required information below and review this role before you create it.

Role name* shllc-ec2-role-ssm

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows EC2 instances to call AWS services like CloudWatch and SSM on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: ec2.amazonaws.com

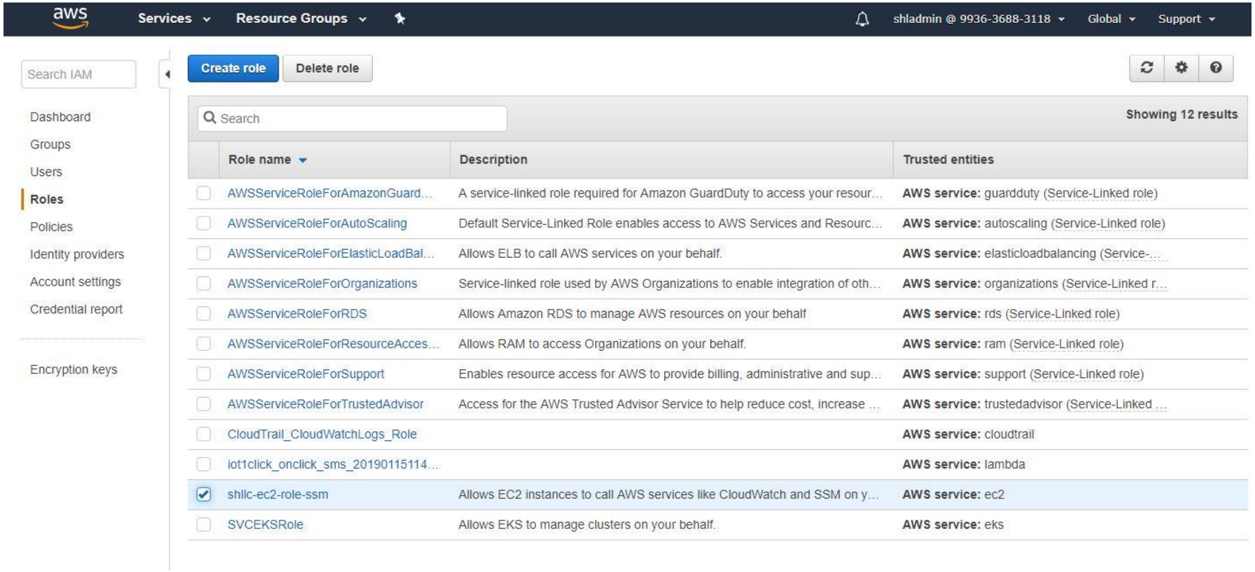
Policies AmazonEC2RoleforSSM

Permissions boundary Permissions boundary is not set

The new role will receive the following tag

Key	Value
Name	shllc-ec2-role-ssm

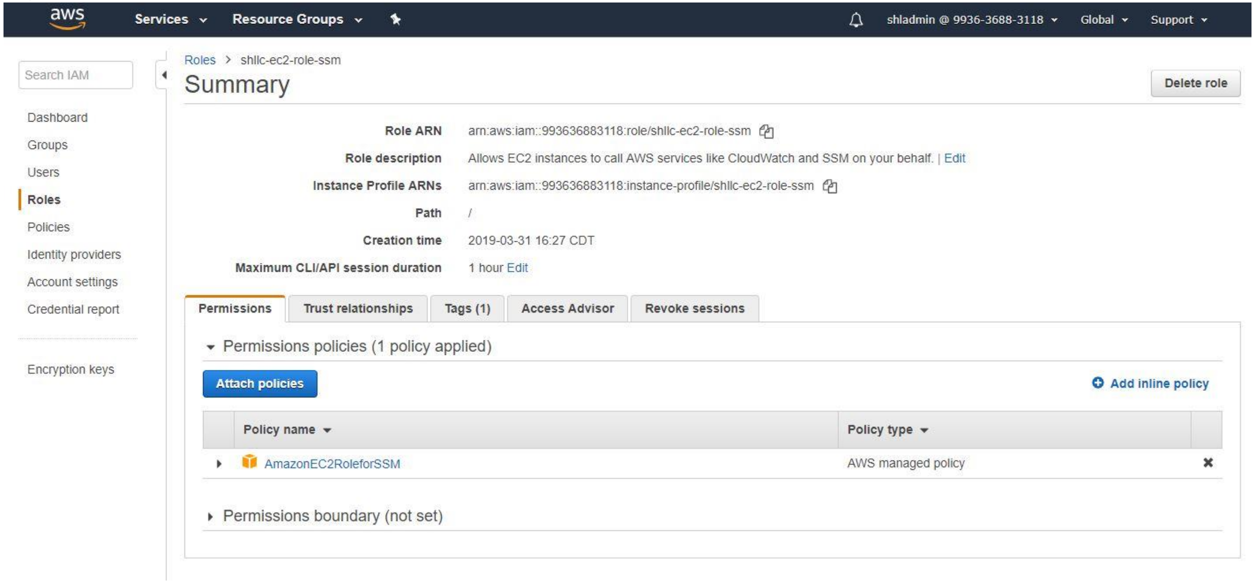
Figure 5 – Role created



Showing 12 results

	Role name	Description	Trusted entities
□	AWSServiceRoleForAmazonGuard...	A service-linked role required for Amazon GuardDuty to access your resour...	**AWS service:** guardduty (Service-Linked role)
□	AWSServiceRoleForAutoScaling	Default Service-Linked Role enables access to AWS Services and Resourc...	**AWS service:** autoscaling (Service-Linked role)
□	AWSServiceRoleForElasticLoadBal...	Allows ELB to call AWS services on your behalf.	**AWS service:** elasticloadbalancing (Service-...
□	AWSServiceRoleForOrganizations	Service-linked role used by AWS Organizations to enable integration of oth...	**AWS service:** organizations (Service-Linked r...
□	AWSServiceRoleForRDS	Allows Amazon RDS to manage AWS resources on your behalf	**AWS service:** rds (Service-Linked role)
□	AWSServiceRoleForResourceAcces...	Allows RAM to access Organizations on your behalf.	**AWS service:** ram (Service-Linked role)
□	AWSServiceRoleForSupport	Enables resource access for AWS to provide billing, administrative and sup...	**AWS service:** support (Service-Linked role)
□	AWSServiceRoleForTrustedAdvisor	Access for the AWS Trusted Advisor Service to help reduce cost, increase ...	**AWS service:** trustedadvisor (Service-Linked ...
□	CloudTrail_CloudWatchLogs_Role		**AWS service:** cloudtrail
□	iot1click_onclick_sms_20190115114...		**AWS service:** lambda
☑	shllc-ec2-role-ssm	Allows EC2 instances to call AWS services like CloudWatch and SSM on y...	**AWS service:** ec2
□	SVCEKSRole	Allows EKS to manage clusters on your behalf.	**AWS service:** eks

Figure 6 – Role details



Summary

Role ARN	arn:aws:iam::993636883118:role/shllc-ec2-role-ssm
Role description	Allows EC2 instances to call AWS services like CloudWatch and SSM on your behalf. | Edit
Instance Profile ARNs	arn:aws:iam::993636883118:instance-profile/shllc-ec2-role-ssm
Path	/
Creation time	2019-03-31 16:27 CDT
Maximum CLI/API session duration	1 hour Edit

▾ Permissions policies (1 policy applied)

Policy name	Policy type
▸ AmazonEC2RoleforSSM	AWS managed policy

▸ Permissions boundary (not set)

References

AWS System Manager User Guide
AWS Cloud Foundation Build Automation v0.1