Tony Shen

AWS SYSTEM MANAGER

Contents

Revision: v0.2
March 8, 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

Before FBA starts, user needs to tell FBA what to do. In the demo in this article, user instructs FBA to create two VPCs in one AWS account. One VPC is in Oregon region (us-west-2), the other, in N. Virginia region (us-east-1). For each VPC, user instructs FBA to create one subnet in each availability zone. Oregon region has four (4) AZs, but only three (3) AZs are used as the fourth one, Availability Zone D (us-west-2d) does not support VPC endpoints required by System Manager, and hence is excluded by FBA. N. Virginia region has six (6) availability zones (AZs), and FBA will use all of them because there are no such issues in those zones.

User puts FBA instruction in a simple table. See Table 1 below.

Table 1 – User FBA Instructions

VPC Name	CIDR	Number of Instances per subnet	Number of subnets per AZ
ret-vpc-usw2-dev-01	10.10.0.0/16	1	1
ret-vpc-use1-prd-01	10.23.0.0/16	1	1

Per User FBA Instruction, FBA will create one VPC named as ret-vpc-usw2-dev-01 in Oregon region. In that VPC, three subnets will be created, and distributed over three AZs in Oregon region. In each subnet, one Linux EC2 instance will be launched, totaling in three (3) instances running in the VPC in the region.

FBA will create another VPC named as ret-vpc-use1-prd-01 in N. Virginia region. In that VPC, six (6) subnets will be created, and distributed over six (6) AZs in N. Virginia region. In each subnet, one Linux EC2 instance will be launched, totaling in six (6) instances running in the VPC in the region.

To kick off FBA, run the simple command below. FBA goes through each step that follows till FB is fully completed, with nine (9) instances up and running, and managed by System Manager in the end.

[awsdsllcadmin@linux721 ~]$ t3 CreateConstructs

Reading VPCs

ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6

Creating VPCs

ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3

Tagging VPCs

ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3

Creating Subnets

ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958 subnet-071f9679bf6cdd507 us-west-2a 10.10.0.0/24
ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958 subnet-046bcfe63b89ea712 us-west-2b 10.10.1.0/24
ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958 subnet-0728f26ece7f6596e us-west-2c 10.10.2.0/24
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-07ef04c2dfbce7100 us-east-1a 10.23.0.0/24
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-085ae968bf5320a37 us-east-1b 10.23.1.0/24
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-0b799ba7eaa8e3107 us-east-1c 10.23.2.0/24
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-098893cc67f7f650d us-east-1d 10.23.3.0/24
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-0147c0ef1f53a4f7d us-east-1e 10.23.4.0/24
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-002c899ce1244b688 us-east-1f 10.23.5.0/24

Tagging Subnets

ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958 subnet-071f9679bf6cdd507 us-west-2a 10.10.0.0/24 ret-vpc-usw2-dev-01-a-sn000
ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958 subnet-046bcfe63b89ea712 us-west-2b 10.10.1.0/24 ret-vpc-usw2-dev-01-b-sn000
ret-vpc-usw2-dev-01 10.10.0.0/16 us-west-2 4 vpc-04d5afd91e80c3958 subnet-0728f26ece7f6596e us-west-2c 10.10.2.0/24 ret-vpc-usw2-dev-01-c-sn000
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-07ef04c2dfbce7100 us-east-1a 10.23.0.0/24 ret-vpc-use1-prd-01-a-sn000
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-085ae968bf5320a37 us-east-1b 10.23.1.0/24 ret-vpc-use1-prd-01-b-sn000
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-0b799ba7eaa8e3107 us-east-1c 10.23.2.0/24 ret-vpc-use1-prd-01-c-sn000
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-098893cc67f7f650d us-east-1d 10.23.3.0/24 ret-vpc-use1-prd-01-d-sn000
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-0147c0ef1f53a4f7d us-east-1e 10.23.4.0/24 ret-vpc-use1-prd-01-e-sn000
ret-vpc-use1-prd-01 10.23.0.0/16 us-east-1 6 vpc-0cf831745787026c3 subnet-002c899ce1244b688 us-east-1f 10.23.5.0/24 ret-vpc-use1-prd-01-f-sn000

Creating Internet Gateways

aws --region us-west-2 ec2 create-internet-gateway
aws --region us-west-2 ec2 create-tags --resources igw-016037ff6deafe8e9 --tags Key=Name,Value=ret-vpc-usw2-dev-01-igw

```
aws --region us-west-2 ec2 attach-internet-gateway --vpc-id vpc-04d5afd91e80c3958 --internet-gateway igw-016037ff6deafe8e9
aws --region us-east-1 ec2 create-internet-gateway
aws --region us-east-1 ec2 create-tags --resources igw-0015881eccf87df6a --tags Key=Name,Value=ret-vpc-use1-prd-01-igw
aws --region us-east-1 ec2 attach-internet-gateway --vpc-id vpc-0cf831745787026c3 --internet-gateway igw-0015881eccf87df6a
Creating VPC route tables ...
aws --region us-west-2 ec2 create-route-table --vpc-id vpc-04d5afd91e80c3958
aws --region us-west-2 ec2 create-tags --resources rtb-0d38568643b8d277e --tags Key=Name,Value=ret-vpc-usw2-dev-01-rtb
aws --region us-west-2 ec2 create-route --route-table-id rtb-0d38568643b8d277e --destination-cidr-block 0.0.0.0/0 --gateway-id igw-016037ff6deafe8e9
{
   "Return": true
}
aws --region us-east-1 ec2 create-route-table --vpc-id vpc-0cf831745787026c3
aws --region us-east-1 ec2 create-tags --resources rtb-0b99ad4735a9ff057 --tags Key=Name,Value=ret-vpc-use1-prd-01-rtb
aws --region us-east-1 ec2 create-route --route-table-id rtb-0b99ad4735a9ff057 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-0015881eccf87df6a
{
   "Return": true
}
```

Creating VCP Security Groups

```
Creating SG for remote access
aws --region us-west-2 ec2 create-security-group --group-name ret-vpc-usw2-dev-01-sg-remote --description ret-vpc-usw2-dev-01-sg-remote --vpc-id vpc-04d5afd91e80c3958
Creating SG for web access
aws --region us-west-2 ec2 create-security-group --group-name ret-vpc-usw2-dev-01-sg-web --description ret-vpc-usw2-dev-01-sg-web --vpc-id vpc-04d5afd91e80c3958
Creating SG for database access
aws --region us-west-2 ec2 create-security-group --group-name ret-vpc-usw2-dev-01-sg-dbs --description ret-vpc-usw2-dev-01-sg-dbs --vpc-id vpc-04d5afd91e80c3958
Creating SG for System Manager access
aws --region us-west-2 ec2 create-security-group --group-name ret-vpc-usw2-dev-01-sg-sm --description ret-vpc-usw2-dev-01-sg-sm --vpc-id vpc-04d5afd91e80c3958
Creating SG for remote access
aws --region us-east-1 ec2 create-security-group --group-name ret-vpc-use1-prd-01-sg-remote --description ret-vpc-use1-prd-01-sg-remote --vpc-id vpc-0cf831745787026c3
Creating SG for web access
aws --region us-east-1 ec2 create-security-group --group-name ret-vpc-use1-prd-01-sg-web --description ret-vpc-use1-prd-01-sg-web --vpc-id vpc-0cf831745787026c3
Creating SG for database access
```

aws --region us-east-1 ec2 create-security-group --group-name ret-vpc-use1-prd-01-sg-dbs --description ret-vpc-use1-prd-01-sg-dbs --vpc-id vpc-0cf831745787026c3
Creating SG for System Manager access
aws --region us-east-1 ec2 create-security-group --group-name ret-vpc-use1-prd-01-sg-sm --description ret-vpc-use1-prd-01-sg-sm --vpc-id vpc-0cf831745787026c3

Creating Key Pairs

aws --region us-west-2 create-key-pair --key-name ret-vpc-usw2-dev-01-keypair
aws --region us-east-1 create-key-pair --key-name ret-vpc-use1-prd-01-keypair

Creating VPC Endpoints

aws --region us-west-2 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-04d5afd91e80c3958 --service-name com.amazonaws.us-west-2.ssm --subnet-ids subnet-071f9679bf6cdd507 subnet-046bcfe63b89ea712 subnet-0728f26ece7f6596e --security-group-ids sg-076387def23e48443 --private-dns-enabled
aws --region us-west-2 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-04d5afd91e80c3958 --service-name com.amazonaws.us-west-2.ssmmessages --subnet-ids subnet-071f9679bf6cdd507 subnet-046bcfe63b89ea712 subnet-0728f26ece7f6596e --security-group-ids sg-076387def23e48443 --private-dns-enabled
aws --region us-west-2 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-04d5afd91e80c3958 --service-name com.amazonaws.us-west-2.ec2 --subnet-ids subnet-071f9679bf6cdd507 subnet-046bcfe63b89ea712 subnet-0728f26ece7f6596e --security-group-ids sg-076387def23e48443 --private-dns-enabled
aws --region us-west-2 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-04d5afd91e80c3958 --service-name com.amazonaws.us-west-2.ec2messages --subnet-ids subnet-071f9679bf6cdd507 subnet-046bcfe63b89ea712 subnet-0728f26ece7f6596e --security-group-ids sg-076387def23e48443 --private-dns-enabled
aws --region us-east-1 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-0cf831745787026c3 --service-name com.amazonaws.us-east-1.ssm --subnet-ids subnet-07ef04c2dfbce7100 subnet-085ae968bf5320a37 subnet-0b799ba7eaa8e3107 subnet-098893cc67f7f650d subnet-0147c0ef1f53a4f7d subnet-002c899ce1244b688 --security-group-ids sg-07d0932c7ec287429 --private-dns-enabled
aws --region us-east-1 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-0cf831745787026c3 --service-name com.amazonaws.us-east-1.ssmmessages --subnet-ids subnet-07ef04c2dfbce7100 subnet-085ae968bf5320a37 subnet-0b799ba7eaa8e3107 subnet-098893cc67f7f650d subnet-0147c0ef1f53a4f7d subnet-002c899ce1244b688 --security-group-ids sg-07d0932c7ec287429 --private-dns-enabled
aws --region us-east-1 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-0cf831745787026c3 --service-name com.amazonaws.us-east-1.ec2 --subnet-ids subnet-07ef04c2dfbce7100 subnet-085ae968bf5320a37 subnet-0b799ba7eaa8e3107 subnet-098893cc67f7f650d subnet-0147c0ef1f53a4f7d subnet-002c899ce1244b688 --security-group-ids sg-07d0932c7ec287429 --private-dns-enabled
aws --region us-east-1 ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-0cf831745787026c3 --service-name com.amazonaws.us-east-1.ec2messages --subnet-ids subnet-07ef04c2dfbce7100 subnet-085ae968bf5320a37 subnet-0b799ba7eaa8e3107 subnet-

098893cc67f7f650d subnet-0147c0ef1f53a4f7d subnet-002c899ce1244b688 --security-group-ids sg-07d0932c7ec287429 --private-dns-enabled

Making public subnets

aws --region us-west-2 ec2 associate-route-table --subnet-id subnet-0728f26ece7f6596e --route-table-id rtb-0d38568643b8d277e
{
  "AssociationId": "rtbassoc-0adb8872024f98df7"
}
aws --region us-west-2 ec2 modify-subnet-attribute --subnet-id subnet-0728f26ece7f6596e --map-public-ip-on-launch
aws --region us-east-1 ec2 associate-route-table --subnet-id subnet-002c899ce1244b688 --route-table-id rtb-0b99ad4735a9ff057
{
  "AssociationId": "rtbassoc-0e36017ed7f42999d"
}
aws --region us-east-1 ec2 modify-subnet-attribute --subnet-id subnet-002c899ce1244b688 --map-public-ip-on-launch

Running instances in subnets

aws --region us-west-2 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-01e24be29428c15b2 --count 1 --instance-type t2.micro --key-name ret-vpc-usw2-dev-01-keypair --security-group-ids sg-0a3c57bd850192e33 sg-03fad9477259567db sg-049f362af91de0a8e sg-076387def23e48443 --subnet-id subnet-071f9679bf6cdd507
aws --region us-west-2 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-01e24be29428c15b2 --count 1 --instance-type t2.micro --key-name ret-vpc-usw2-dev-01-keypair --security-group-ids sg-0a3c57bd850192e33 sg-03fad9477259567db sg-049f362af91de0a8e sg-076387def23e48443 --subnet-id subnet-046bcfe63b89ea712
aws --region us-west-2 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-01e24be29428c15b2 --count 1 --instance-type t2.micro --key-name ret-vpc-usw2-dev-01-keypair --security-group-ids sg-0a3c57bd850192e33 sg-03fad9477259567db sg-049f362af91de0a8e sg-076387def23e48443 --subnet-id subnet-0728f26ece7f6596e
aws --region us-east-1 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name ret-vpc-use1-prd-01-keypair --security-group-ids sg-0724ec7a66c0d7ba5 sg-04912a4465cd2f544 sg-08d3c576588b094f3 sg-07d0932c7ec287429 --subnet-id subnet-07ef04c2dfbce7100
aws --region us-east-1 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name ret-vpc-use1-prd-01-keypair --security-group-ids sg-0724ec7a66c0d7ba5 sg-04912a4465cd2f544 sg-08d3c576588b094f3 sg-07d0932c7ec287429 --subnet-id subnet-085ae968bf5320a37
aws --region us-east-1 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name ret-vpc-use1-prd-01-keypair --security-group-ids sg-0724ec7a66c0d7ba5 sg-04912a4465cd2f544 sg-08d3c576588b094f3 sg-07d0932c7ec287429 --subnet-id subnet-0b799ba7eaa8e3107

```
aws --region us-east-1 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name ret-vpc-use1-prd-01-keypair --security-group-ids sg-0724ec7a66c0d7ba5 sg-04912a4465cd2f544 sg-08d3c576588b094f3 sg-07d0932c7ec287429 --subnet-id subnet-098893cc67f7f650d
aws --region us-east-1 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name ret-vpc-use1-prd-01-keypair --security-group-ids sg-0724ec7a66c0d7ba5 sg-04912a4465cd2f544 sg-08d3c576588b094f3 sg-07d0932c7ec287429 --subnet-id subnet-0147c0ef1f53a4f7d
aws --region us-east-1 ec2 run-instances --iam-instance-profile Name=ret-vpc-use2-dev-01-ec2-role-ssm --image-id ami-0080e4c5bc078760e --count 1 --instance-type t2.micro --key-name ret-vpc-use1-prd-01-keypair --security-group-ids sg-0724ec7a66c0d7ba5 sg-04912a4465cd2f544 sg-08d3c576588b094f3 sg-07d0932c7ec287429 --subnet-id subnet-002c899ce1244b688
```

Tagging instances

```
aws --region us-west-2 ec2 create-tags --resources i-0e647185f8de6f676 --tags Key=Name,Value=ret-vpc-usw2-dev-01-a-sn000-ip-10-10-0-125
aws --region us-west-2 ec2 create-tags --resources i-08144922ad3947c2b --tags Key=Name,Value=ret-vpc-usw2-dev-01-b-sn000-ip-10-10-1-244
aws --region us-west-2 ec2 create-tags --resources i-0a51737bd6cec5c49 --tags Key=Name,Value=ret-vpc-usw2-dev-01-c-sn000-ip-10-10-2-23
aws --region us-east-1 ec2 create-tags --resources i-02c0060207fa5681c --tags Key=Name,Value=ret-vpc-use1-prd-01-a-sn000-ip-10-23-0-144
aws --region us-east-1 ec2 create-tags --resources i-0d3b8e8b8fd8a1747 --tags Key=Name,Value=ret-vpc-use1-prd-01-b-sn000-ip-10-23-1-103
aws --region us-east-1 ec2 create-tags --resources i-0b68d35fe16d29e88 --tags Key=Name,Value=ret-vpc-use1-prd-01-c-sn000-ip-10-23-2-80
aws --region us-east-1 ec2 create-tags --resources i-0c96189879204c4da --tags Key=Name,Value=ret-vpc-use1-prd-01-d-sn000-ip-10-23-3-169
aws --region us-east-1 ec2 create-tags --resources i-026090a5d2d8a6347 --tags Key=Name,Value=ret-vpc-use1-prd-01-e-sn000-ip-10-23-4-164
aws --region us-east-1 ec2 create-tags --resources i-032d0aaa3031510fd --tags Key=Name,Value=ret-vpc-use1-prd-01-f-sn000-ip-10-23-5-193
```

Displaying ssh instance connection commands

```
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 ~]$
```

At this point, SM has been automatically configured and enabled. SM is ready for use.

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)

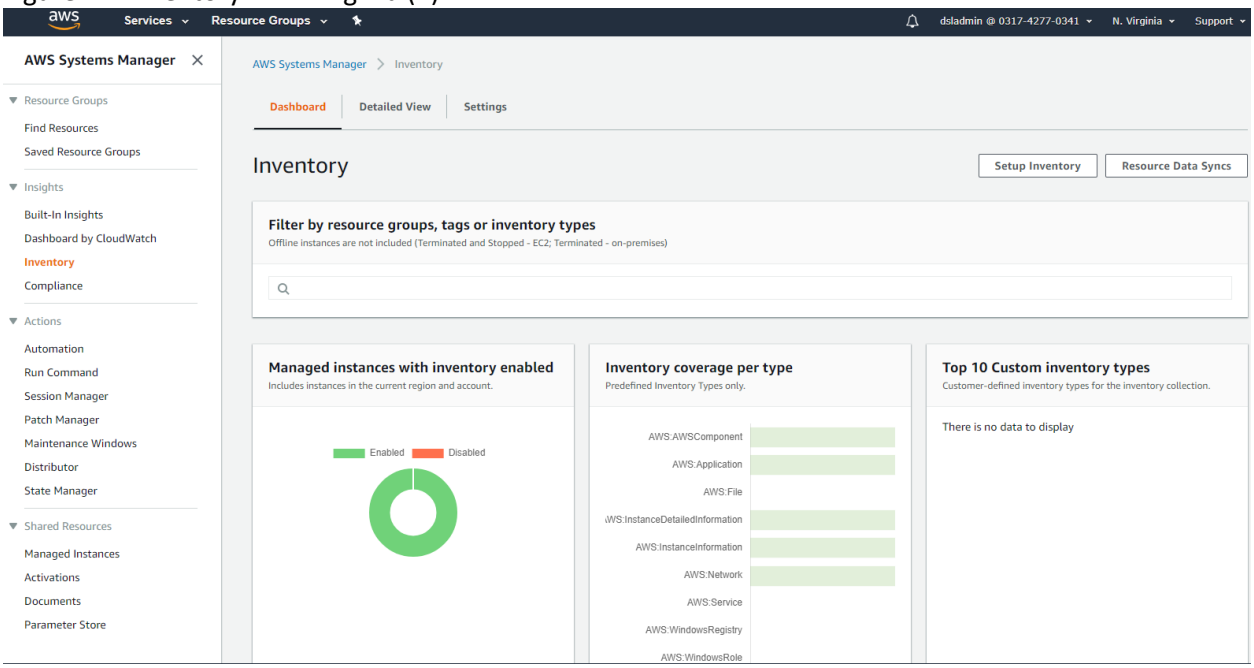


Figure 2 – Inventory in N. Virginia (2)

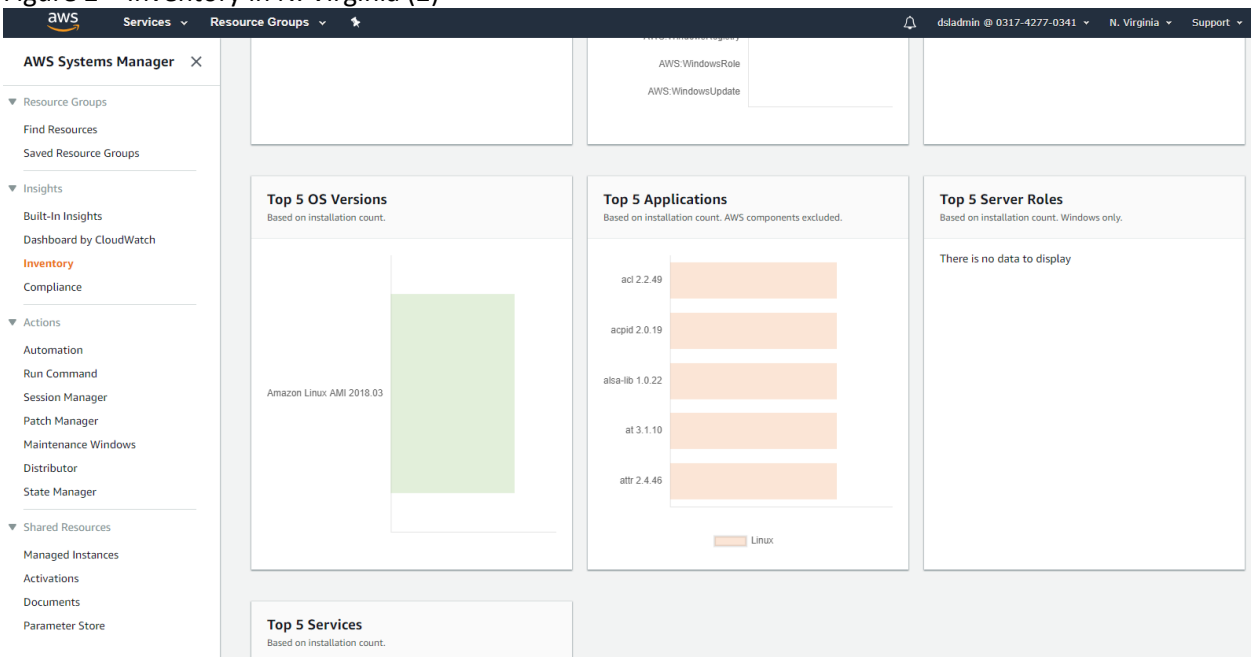


Figure 3 – Inventory in N. Virginia (3)

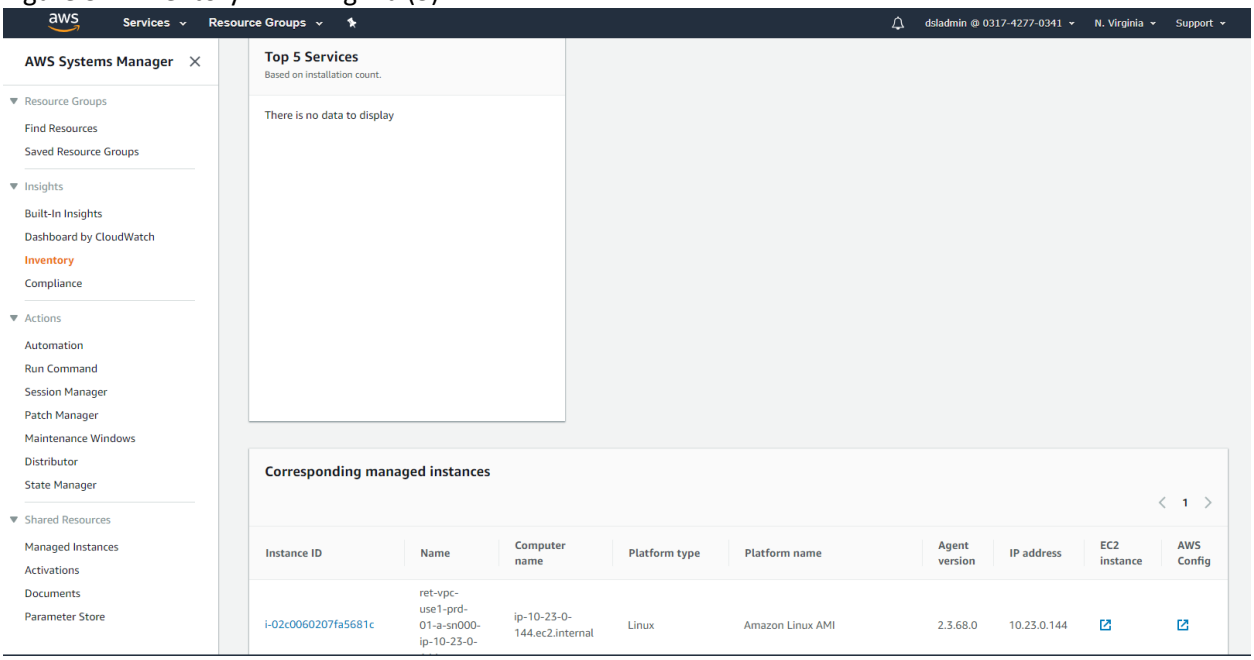


Figure 4 – Inventory in N. Virginia (4)

Figure 5 – Oregon Inventory Dashboard (1)

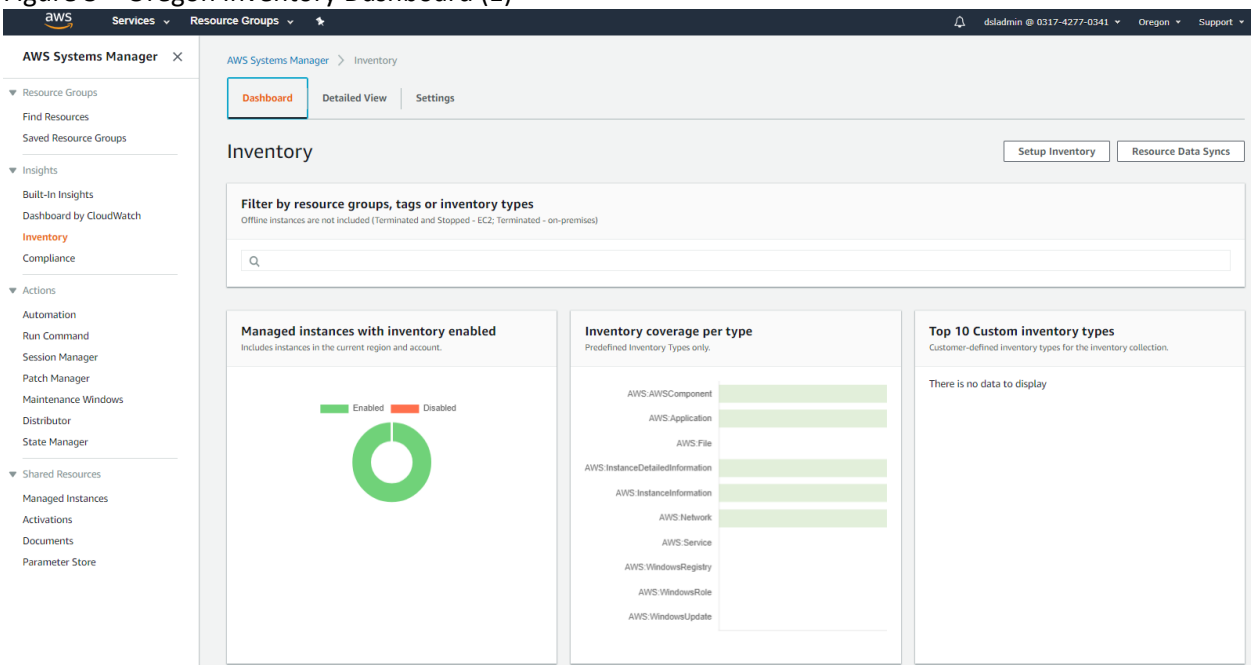


Figure 6 - Oregon Inventory Dashboard (2)

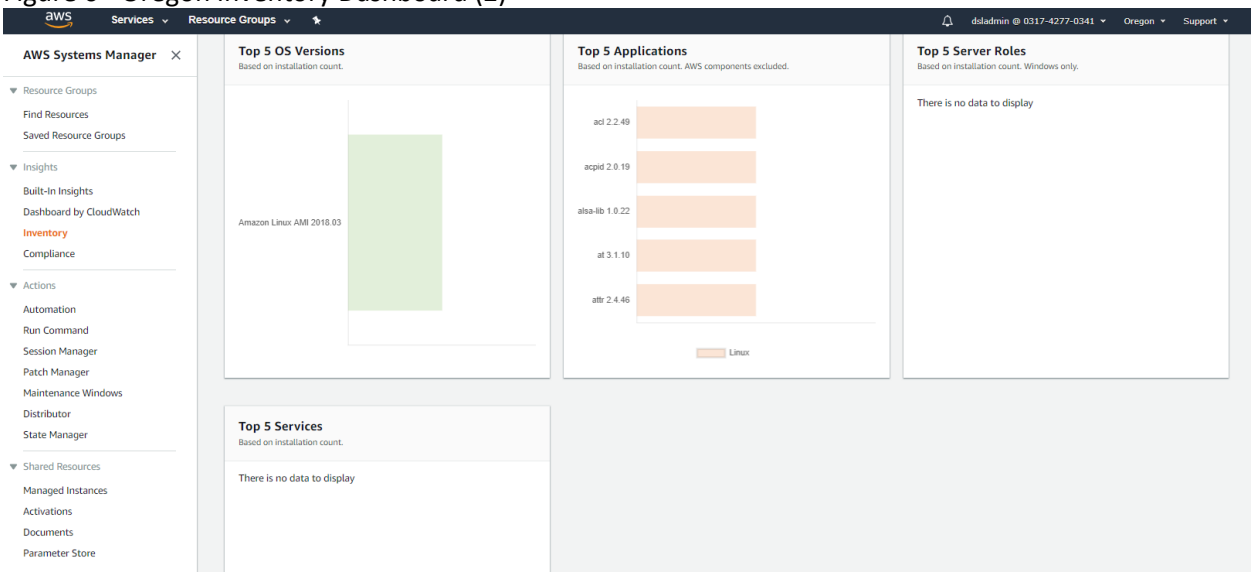
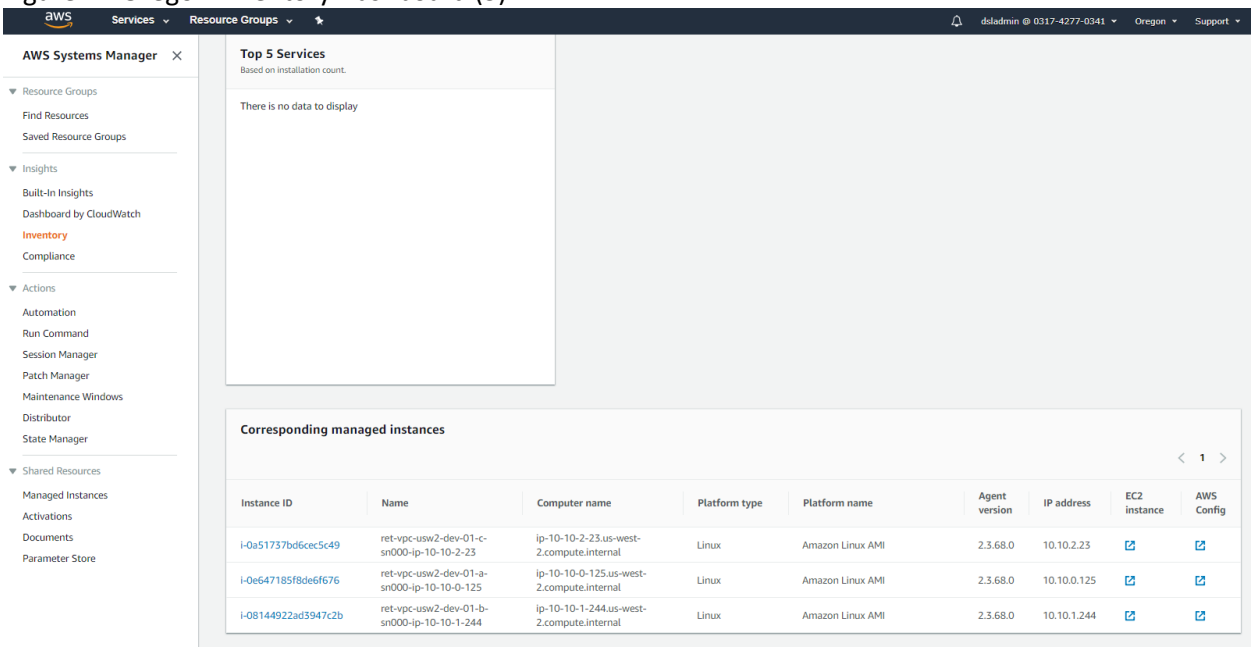


Figure 7 - Oregon Inventory Dashboard (3)



SM Actions - Session Manager

Figure 8 – N. Virginia Session Manager Instance Listing

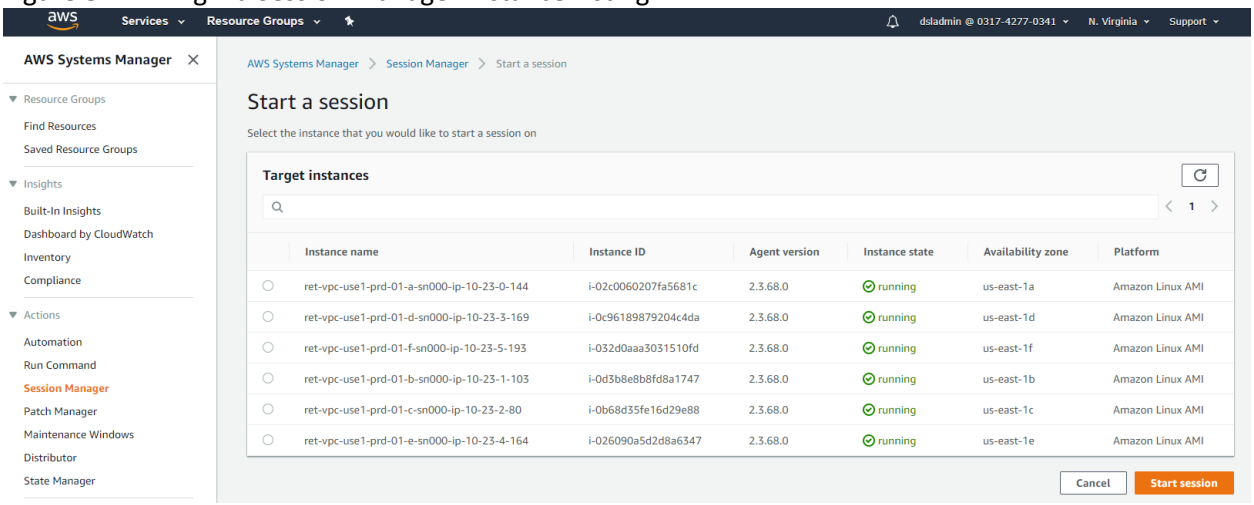
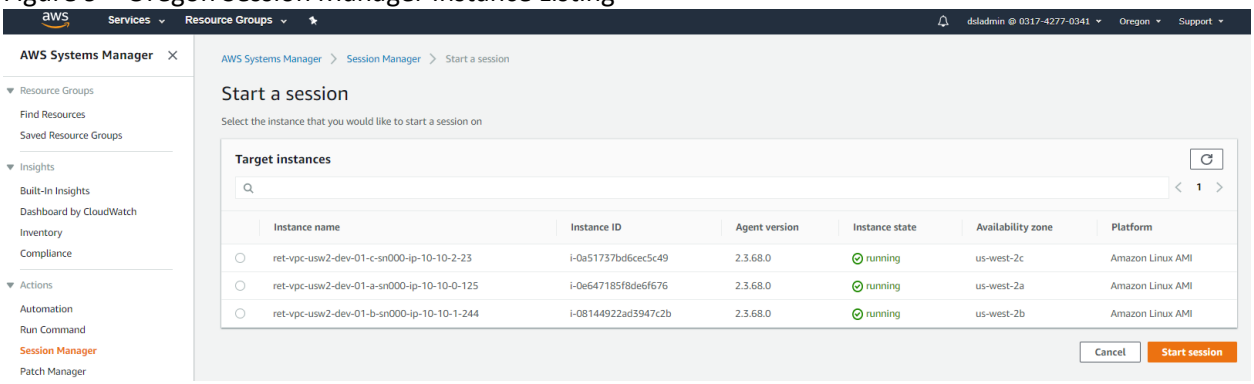


Figure 9 – Oregon Session Manager Instance Listing



SM Shared Resources - Managed Instances

Figure 10 – N. Virginia Managed Instances

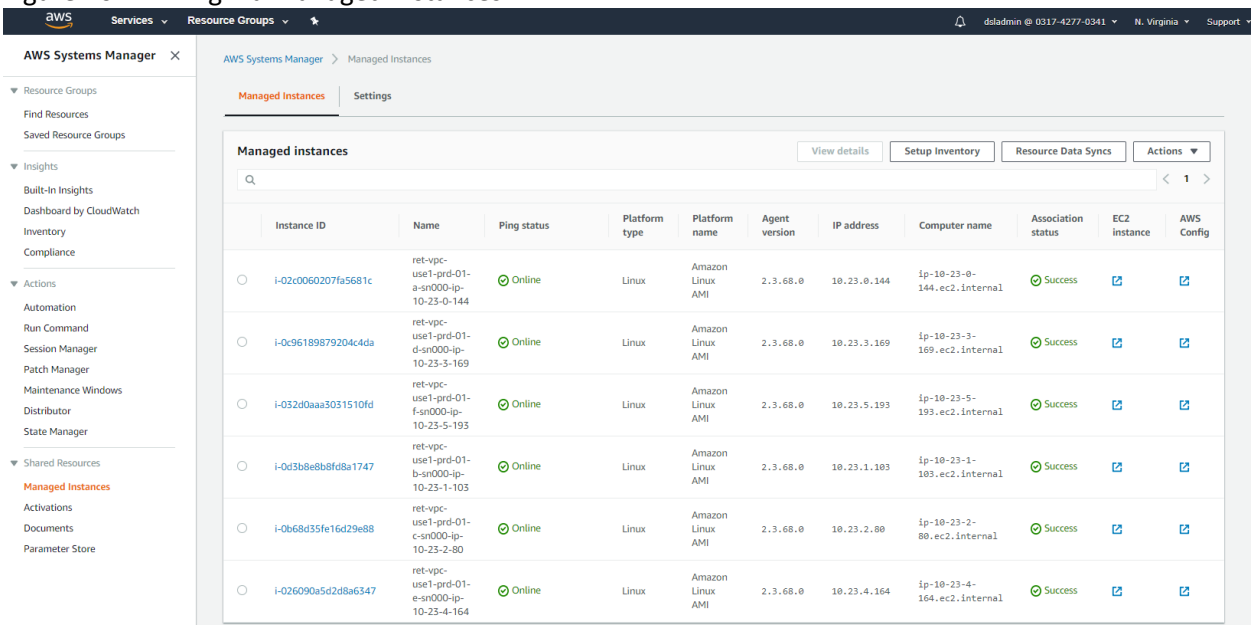
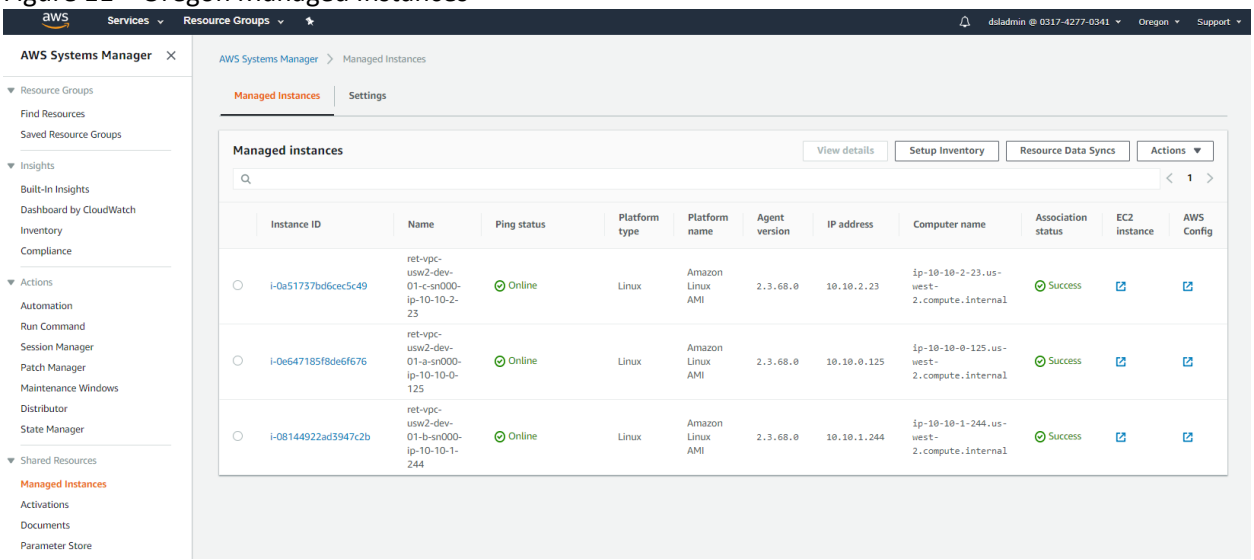


Figure 11 – Oregon Managed Instances



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

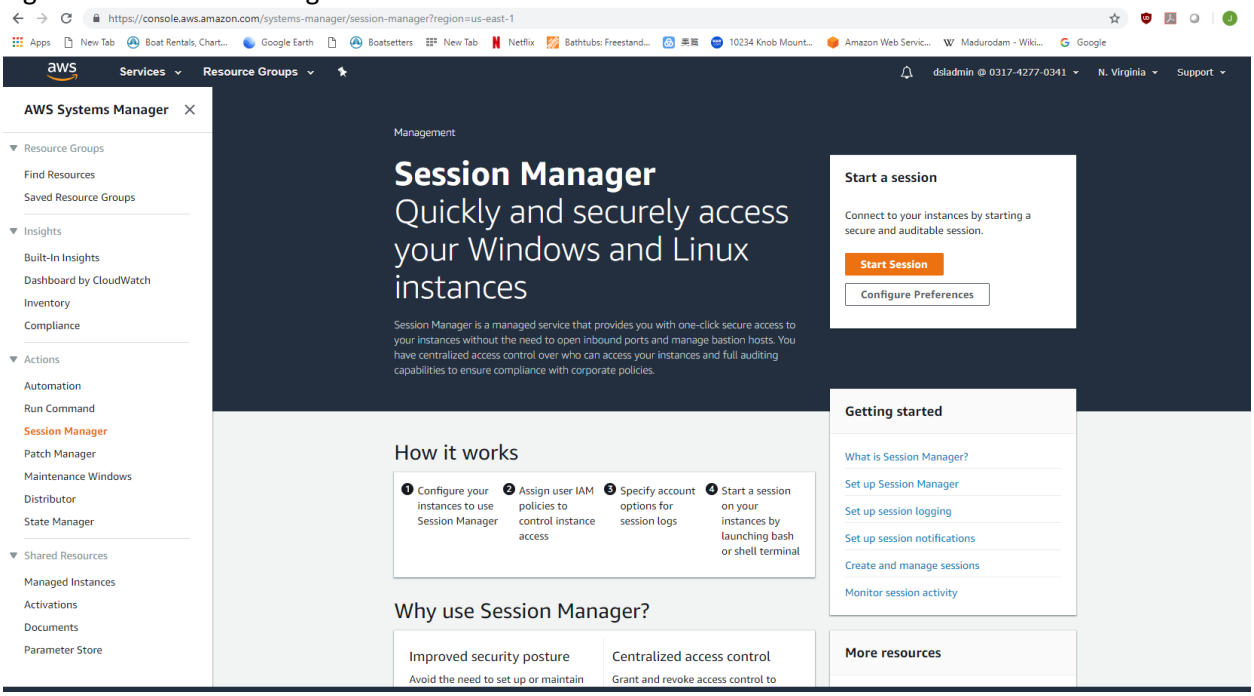
```
[awsdsllcadmin@linux721 ~]$ aws --region us-west-2 ssm get-inventory --output text | grep CONTENT |
grep -v Terminated
CONTENT amazon-ssm-agent	2.3.68.0	ip-10-10-1-244.us-west-2.compute.internal	i-
08144922ad3947c2b	10.10.1.244	Amazon Linux AMI	Linux	2018.03 EC2Instance
CONTENT amazon-ssm-agent	2.3.68.0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



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

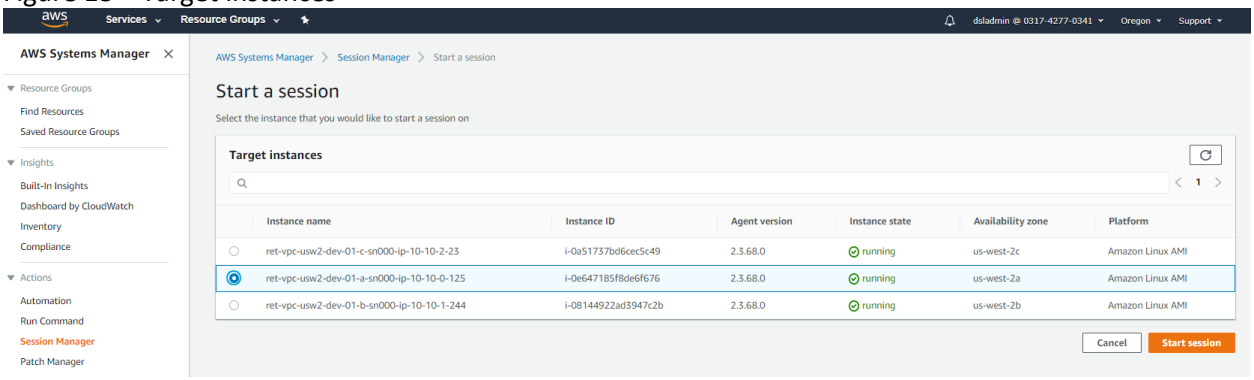
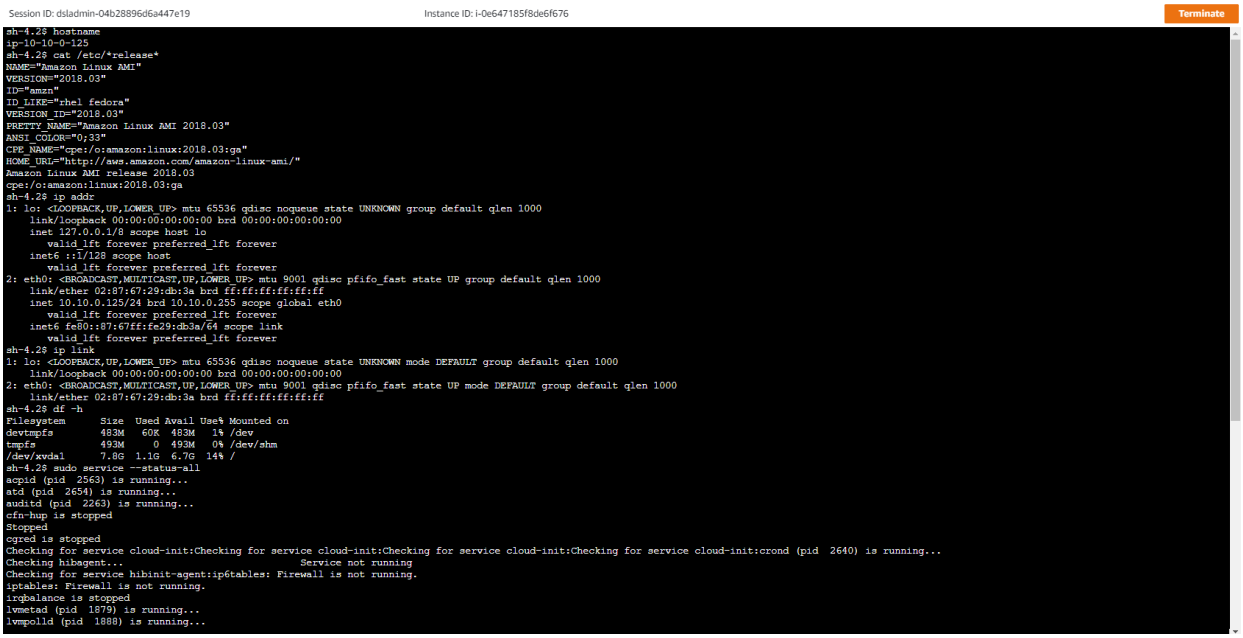


Figure 14 below shows what it looks like when a session is started with an EC2 instance. In this case, the EC2 instance selected is a private instance with IP address of 10.10.0.125, with no Internet access. The session opens up a bash shell in the instance.

Figure 14 - Session shell screen



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. The session creates an isolated shell in the target instance, 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

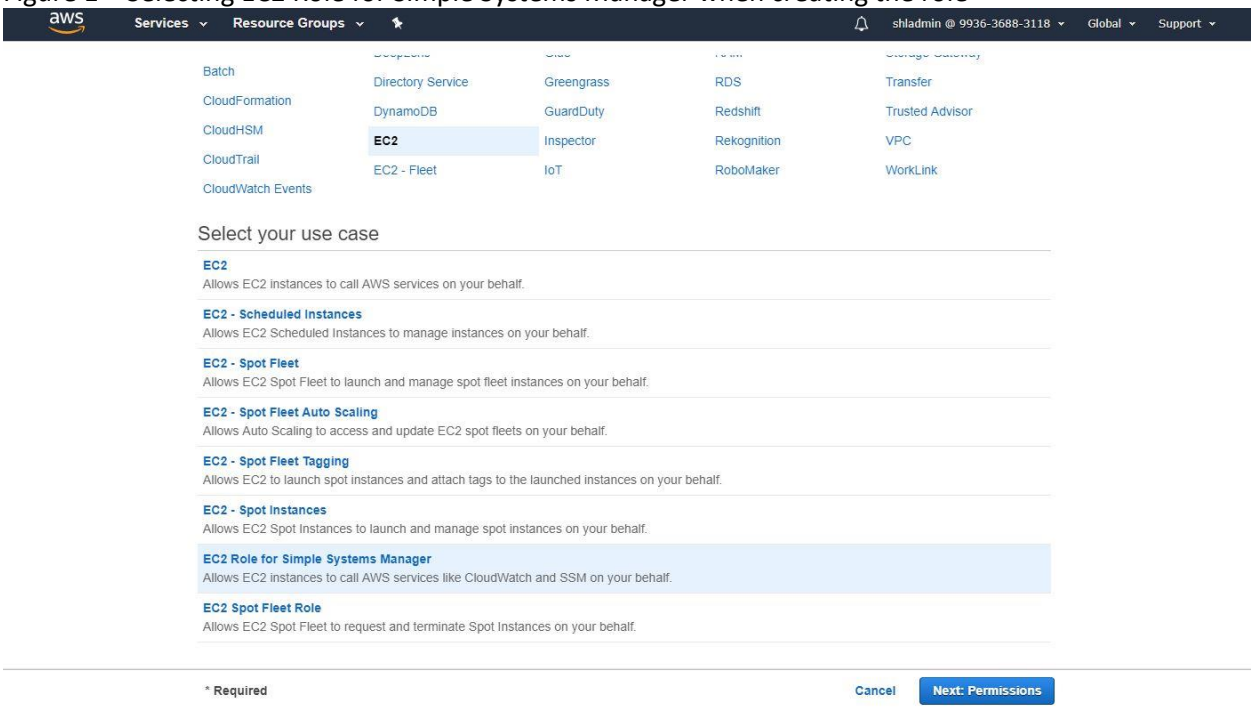


Figure 2 – Attaching AmazonEC2RoleforSSM permissions policy to the role

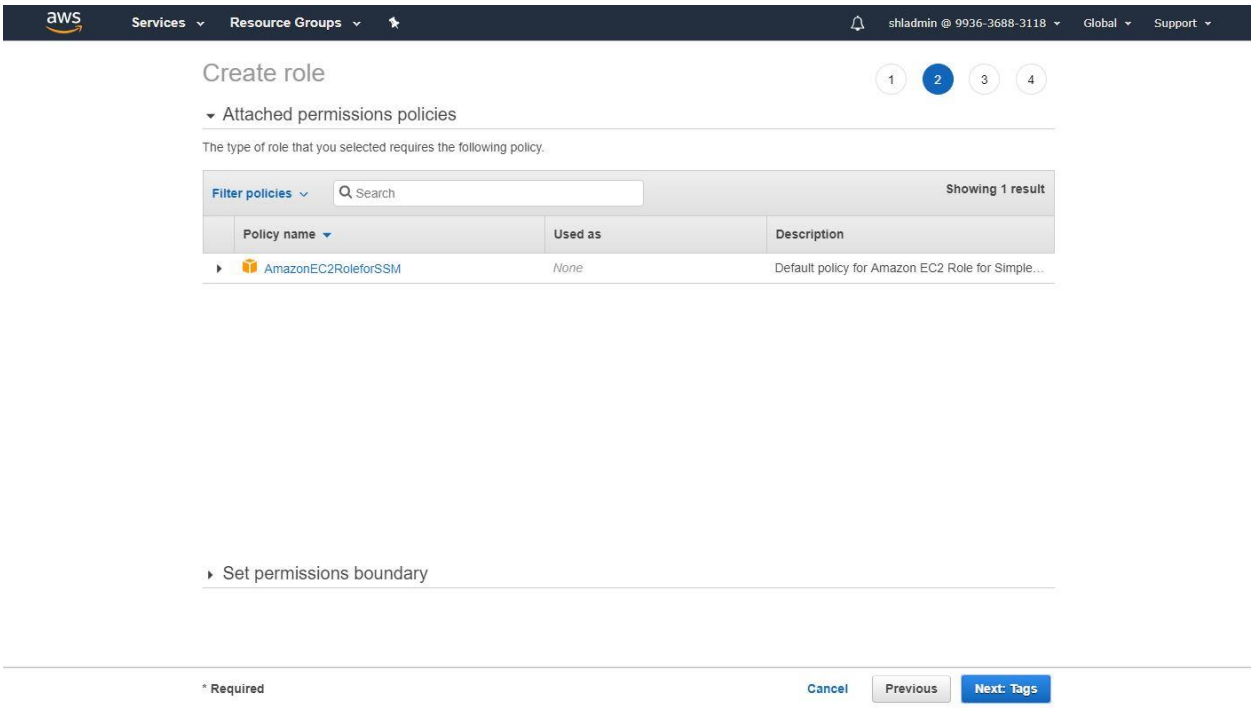


Figure 3 – Click Next: Tags

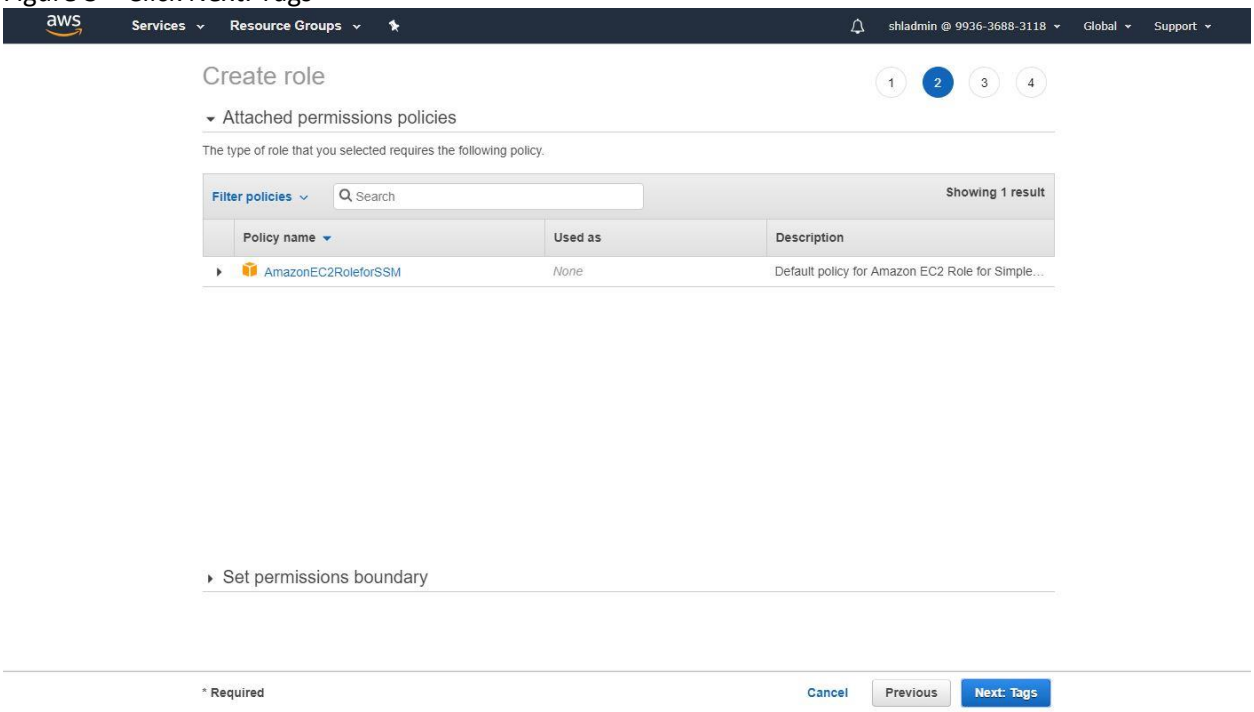


Figure 4 – Naming the role

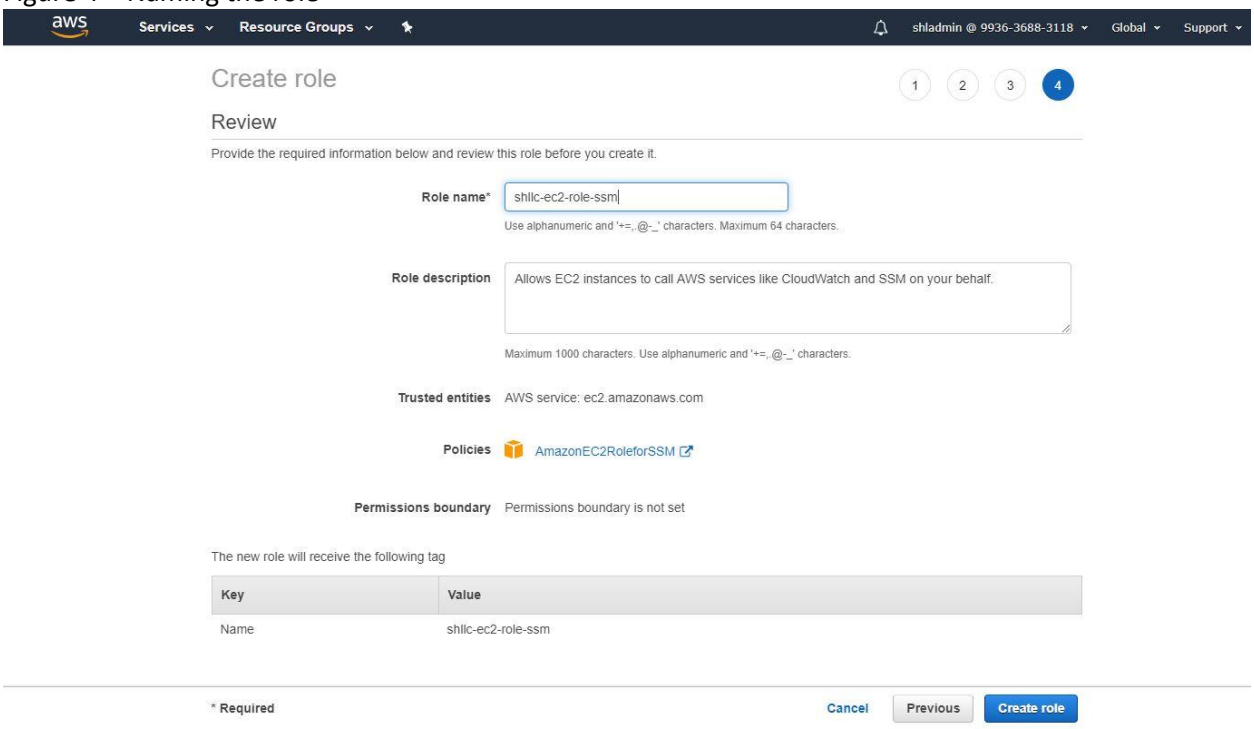


Figure 5 – Role created

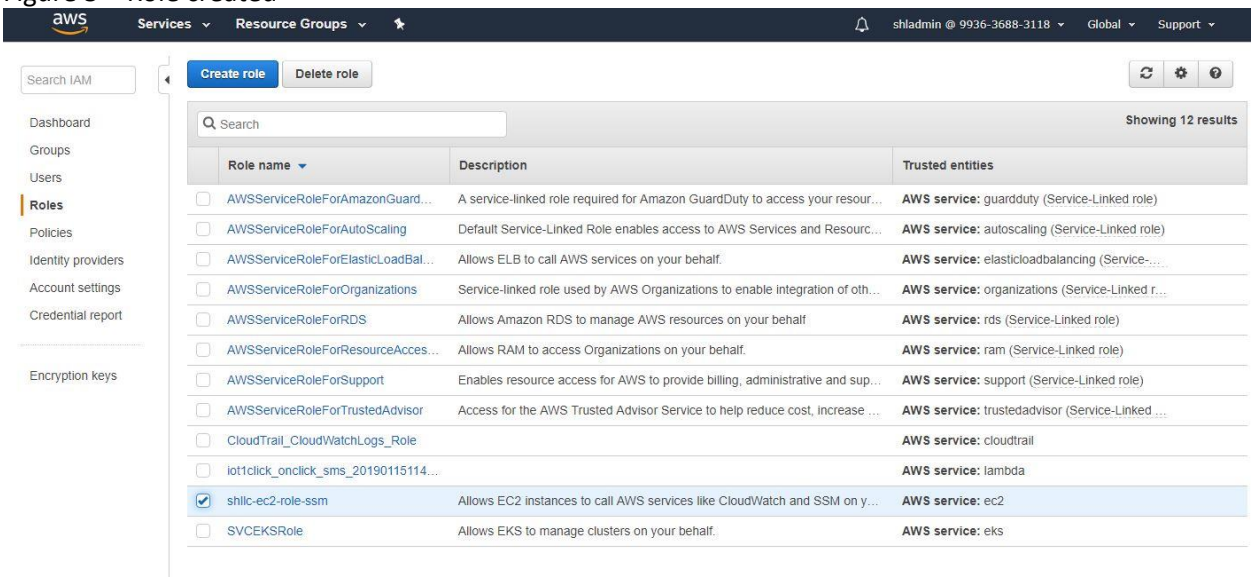
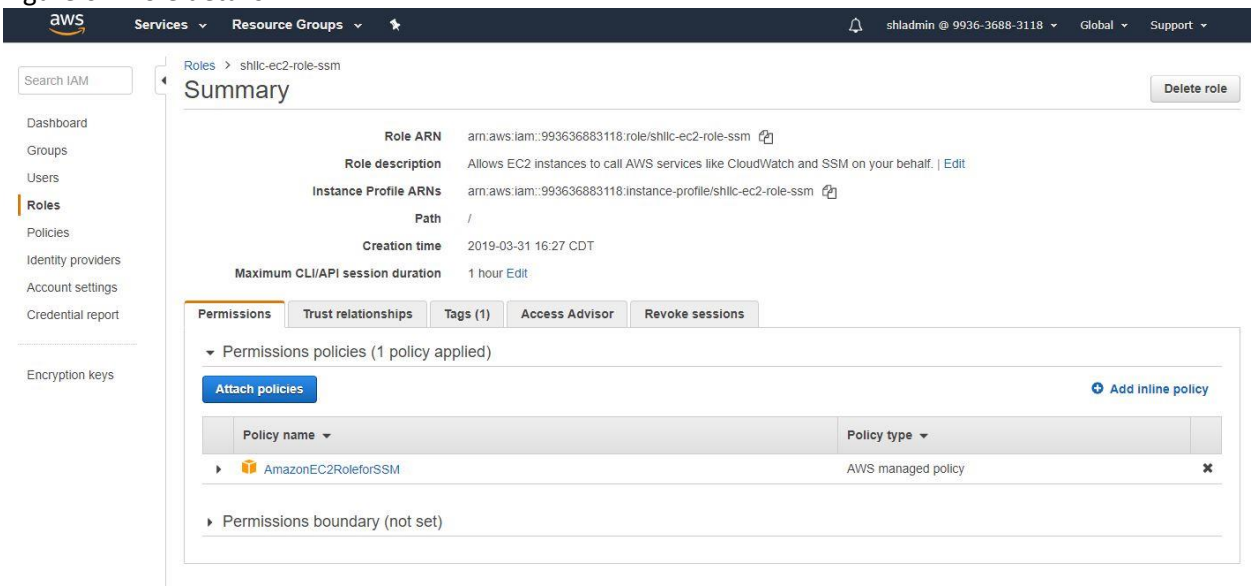


Figure 6 – Role details



References

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