

Running Ansible Playbook using AWS Systems Manager



In this Lab, we are going to use AWS Systems Manager Service to install Ansible on the Target Instances. Once the Ansible is successfully installed on the Target Instance then we are going to run a Ansible Playbook on the Target Instance using AWS Systems Manager Service to Install & Configure Apache Server.

Below is the List of Tasks:

Task 1: Create IAM Role

Task 2: Launch & Configure EC2 Instances with SSM Agent

Task 3: Create a S3 Bucket to store SSM Logs

Task 4: AWS Systems Manager: Managed Instances

Task 5: AWS-Systems Manager: Run Command (Ansible Installation)

Task 6: Ansible Installation Check

Task 7: Run an Ansible Playbook using AWS Systems Manager

Task 8: Verify Ansible Playbook Execution

Task 1: Create IAM Role

Login to the AWS Management Console.

Navigate to IAM Service and click on Roles.

Click on Create Role.

Make sure to select the Use Case as **EC2**.

Click Next: Permissions.

The screenshot shows the 'Create role' page in the AWS IAM console. The 'Select type of trusted entity' step is active, showing four options: 'AWS service' (selected), 'Another AWS account', 'Web Identity', and 'SAML 2.0 federation'. Below this, the 'Choose a use case' section lists 'EC2' as the selected use case, with a description: 'Allows EC2 instances to call AWS services on your behalf.' Other use cases like 'Lambda' are also visible. At the bottom, there are buttons for 'Cancel' and 'Next: Permissions'.

Select the below two Default IAM Policies:

1. **AmazonEC2RoleforSSM**
2. **AmazonSSMFullAccess**

The screenshot shows the 'Create role' page in the AWS IAM console, specifically the 'Attach permissions policies' step. A search bar with 'SSM' is active, showing 13 results. The results table lists several policies, with 'AmazonEC2RoleforSSM' and 'AmazonSSMFullAccess' selected. The 'Used as' column shows 'Permissions policy (1)' for the selected policies. At the bottom, there are buttons for 'Cancel', 'Previous', and 'Next: Tags'.

Policy name	Used as
<input checked="" type="checkbox"/> AmazonEC2RoleforSSM	Permissions policy (1)
<input type="checkbox"/> AmazonSSMAutomationApproverAccess	None
<input type="checkbox"/> AmazonSSMAutomationRole	None
<input type="checkbox"/> AmazonSSMDirectoryServiceAccess	None
<input checked="" type="checkbox"/> AmazonSSMFullAccess	Permissions policy (1)
<input type="checkbox"/> AmazonSSMMaintenanceWindowRole	None
<input type="checkbox"/> AmazonSSMManagedInstanceCore	Permissions policy (1)
<input type="checkbox"/> AmazonSSMDirectConnectAccess	None

Give the Role Name as per your Choice and click on Create Role.

The screenshot shows the 'Create role' wizard in the AWS IAM console, specifically the 'Review' step. The role name is 'EC2-Role-SSM'. The role description is 'Allows EC2 instances to call AWS services on your behalf.' The trusted entities are 'AWS service: ec2.amazonaws.com'. The policies selected are 'AmazonEC2RoleforSSM' and 'AmazonSSMFullAccess'. The permissions boundary is 'Permissions boundary is not set'. At the bottom, there are buttons for 'Cancel', 'Previous', and 'Create role'.

Create role

Review

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

Role name* EC2-Role-SSM
Use alphanumeric and '+=, @, _' characters. Maximum 64 characters.

Role description Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=, @, _' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies
AmazonEC2RoleforSSM
AmazonSSMFullAccess

Permissions boundary Permissions boundary is not set

* Required

Cancel Previous **Create role**

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Task 2: Launch & Configure EC2 Instances with SSM Agent

Navigate to EC2 Service and click on Launch Instance.

Select the **Amazon Linux AMI**.

The screenshot shows the 'Launch Instance' wizard in the AWS Management Console, Step 1: Choose an Amazon Machine Image (AMI). The search bar contains 'Search for an AMI by entering a search term e.g. "Windows"'. The 'Quick Start' sidebar is on the left. The main area shows two AMIs: 'Amazon Linux 2 AMI (HVM), SSD Volume Type' and 'Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type'. Both are marked as 'Free tier eligible'. The first AMI has a 'Select' button and radio buttons for '64-bit (x86)' and '64-bit (Arm)'. The second AMI has a 'Select' button and a radio button for '64-bit (x86)'.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs
- ☐ Free tier only ⓘ

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d (64-bit x86) / ami-0ce2e5b7d27317779 (64-bit Arm)
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0915e09cc7ceee3ab
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

1 to 40 of 40 AMIs

Select the Number of Instances as 1, select the Network as our Custom VPC, Select Subnet as Public Subnet 1 and select the IAM Role which you configured in the Task 1.

The screenshot shows the 'Configure Instance Details' step in the AWS Management Console. The navigation bar at the top includes the AWS logo, 'Services', 'Resource Groups', and a star icon. On the right, there's a notification bell, the user 'Prasad_SMU', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, a progress bar shows seven steps: '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance' (which is the active step), '4. Add Storage', '5. Add Tags', '6. Configure Security Group', and '7. Review'. The main content area is titled 'Step 3: Configure Instance Details' with a sub-header 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.' The configuration options are as follows: 'Number of instances' is set to 1 with a 'Launch into Auto Scaling Group' link; 'Purchasing option' has a checkbox for 'Request Spot instances' which is unchecked; 'Network' is set to 'vpc-062814d035612343e | Custom VPC' with a 'Create new VPC' link; 'Subnet' is set to 'subnet-01ee44283bcd09e5c | Public Subnet 1 | us-e' with a 'Create new subnet' link and a note '245 IP Addresses available'; 'Auto-assign Public IP' is set to 'Use subnet setting (Enable)'; 'Placement group' has a checkbox for 'Add instance to placement group' which is unchecked; 'Capacity Reservation' is set to 'Open' with a 'Create new Capacity Reservation' link; and 'IAM role' is set to 'EC2-Role-SSM' with a 'Create new IAM role' link.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)
245 IP Addresses available

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation [Create new Capacity Reservation](#)

IAM role [Create new IAM role](#)

Since AWS Systems Manager is AGENTLESS, we need to install Packages for Systems Manager (SSM) to connect with Target Instances.

Scroll down on the same page, click on Advanced Details and in User Data field bootstrap the below commands.

I've provided the Commands in text file.

Advanced Details

The screenshot shows the 'Advanced Details' section of the AWS Management Console. It contains four configuration options: 'Metadata accessible' is set to 'Enabled'; 'Metadata version' is set to 'V1 and V2 (token optional)'; 'Metadata token response hop limit' is set to '1'; and 'User data' has three radio buttons: 'As text' (which is selected), 'As file', and 'Input is already base64 encoded'. Below these options is a text area containing a series of commands for installing and configuring the AWS Systems Manager agent.

Metadata accessible

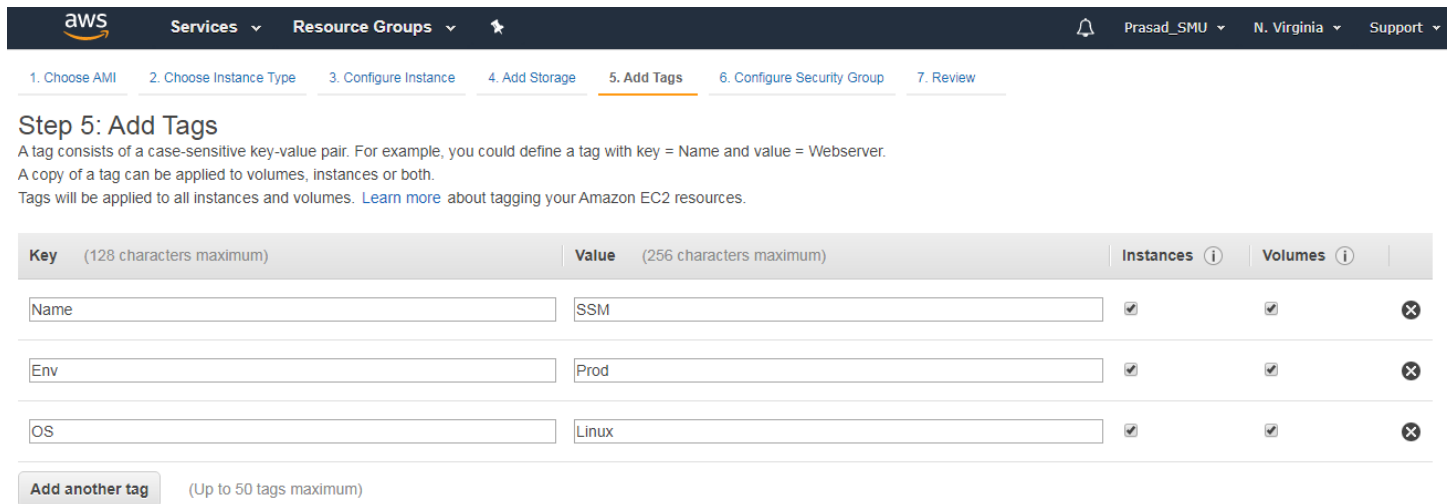
Metadata version

Metadata token response hop limit

User data ☒ As text ☐ As file ☐ Input is already base64 encoded

```
#!/bin/bash
cd /tmp
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-
windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm
sudo systemctl start amazon-ssm-agent
sudo systemctl enable amazon-ssm-agent
```

You can mention Tags as per your choice.



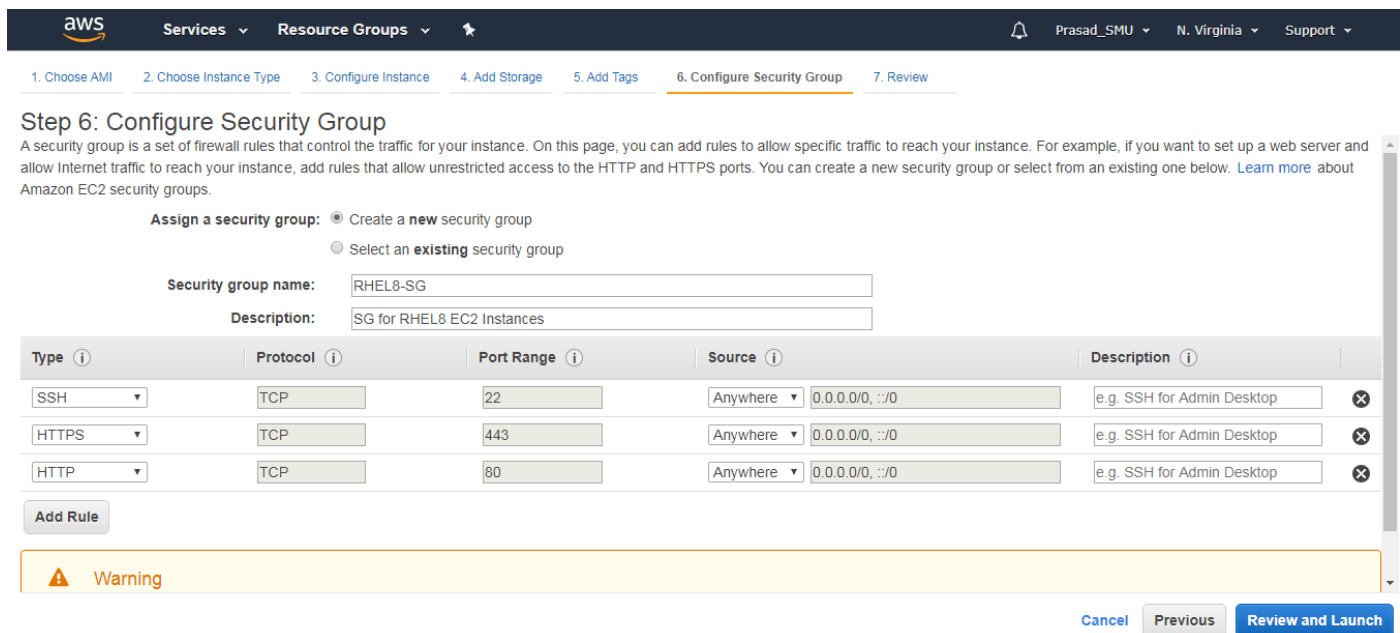
The screenshot shows the 'Add Tags' step in the AWS Management Console. The breadcrumb trail includes: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (active), 6. Configure Security Group, and 7. Review. The page title is 'Step 5: Add Tags'. A note explains that a tag is a case-sensitive key-value pair and can be applied to instances and volumes. Below this, there is a table with columns for 'Key' (128 characters maximum), 'Value' (256 characters maximum), 'Instances', and 'Volumes'. Three tags are added: 'Name' with value 'SSM', 'Env' with value 'Prod', and 'OS' with value 'Linux'. Each tag has checkboxes for 'Instances' and 'Volumes', both of which are checked. At the bottom, there is a button 'Add another tag' and a note '(Up to 50 tags maximum)'.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	SSM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Env	Prod	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OS	Linux	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

Click Next: Security Groups.

Create a new Security Group. Give the Name & Discription as per your choice. Allow SSH, HTTPS, HTTP Inbound traffic from Anywhere.



The screenshot shows the 'Configure Security Group' step in the AWS Management Console. The breadcrumb trail includes: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group (active), and 7. Review. The page title is 'Step 6: Configure Security Group'. A note explains that a security group is a set of firewall rules that control traffic. Below this, there is a section 'Assign a security group:' with two radio buttons: 'Create a new security group' (selected) and 'Select an existing security group'. Below this, there are input fields for 'Security group name:' (RHEL8-SG) and 'Description:' (SG for RHEL8 EC2 Instances). Below these fields is a table with columns for 'Type', 'Protocol', 'Port Range', 'Source', and 'Description'. Three rules are added: SSH (TCP, 22, Anywhere, 0.0.0.0/0, ::/0), HTTPS (TCP, 443, Anywhere, 0.0.0.0/0, ::/0), and HTTP (TCP, 80, Anywhere, 0.0.0.0/0, ::/0). At the bottom, there is a button 'Add Rule'. Below the table, there is a yellow warning box with a warning icon and the text 'Warning'. At the bottom right, there are buttons 'Cancel', 'Previous', and 'Review and Launch'.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

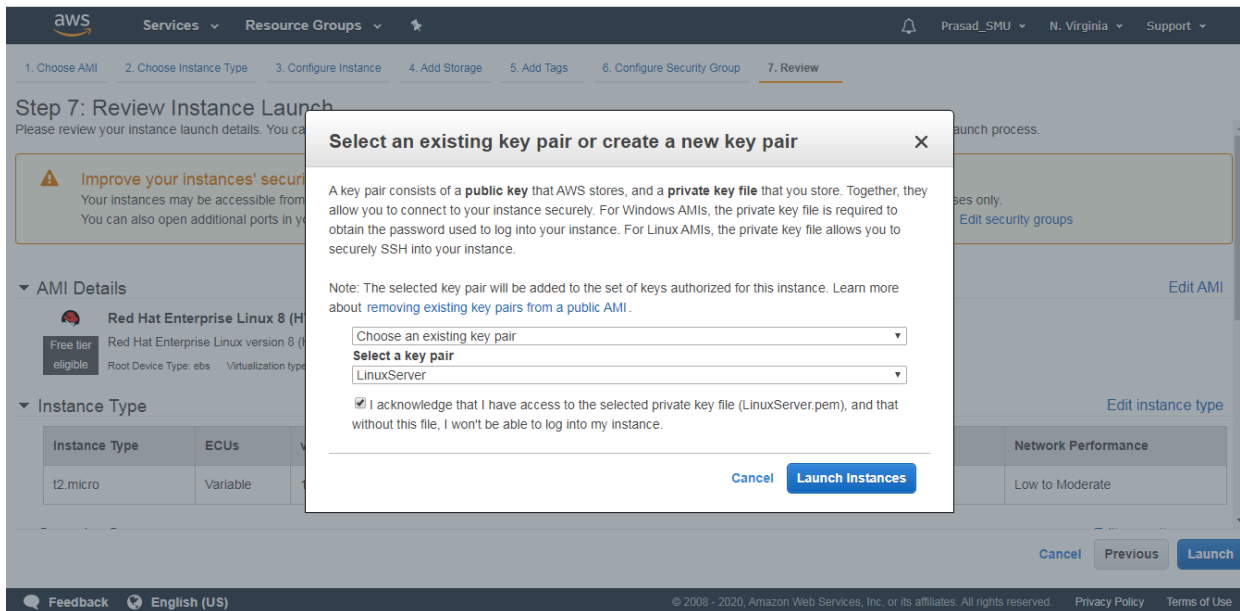
Warning

[Cancel](#) [Previous](#) [Review and Launch](#)

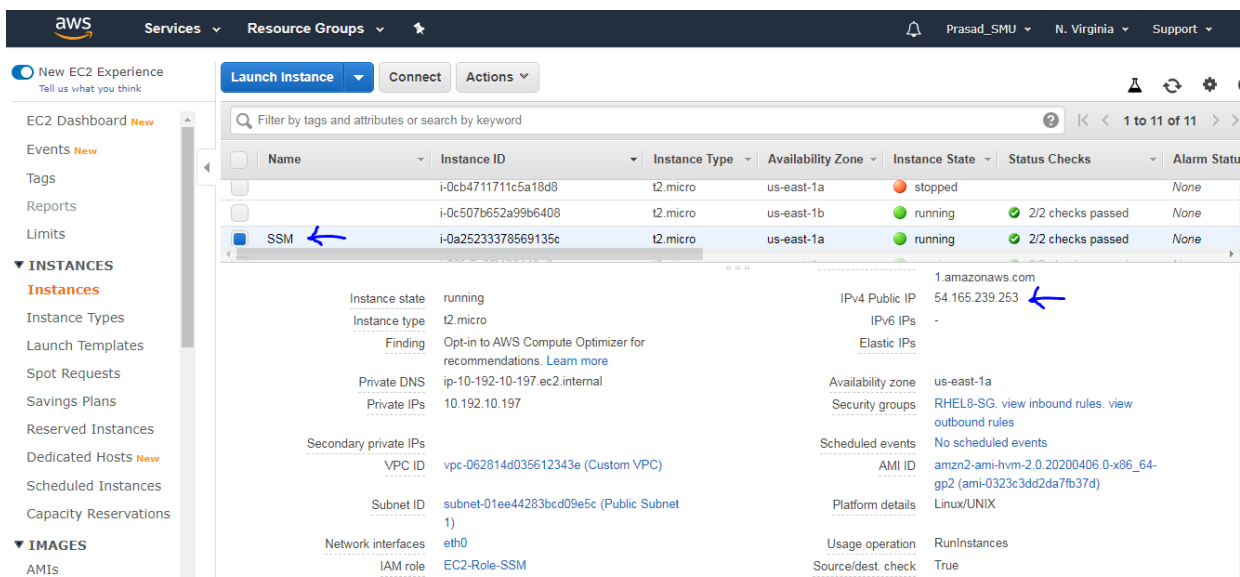
Click on Review and Launch.

Select the existing Key Pair which you've using for previous labs.

Click on Launch Instances.



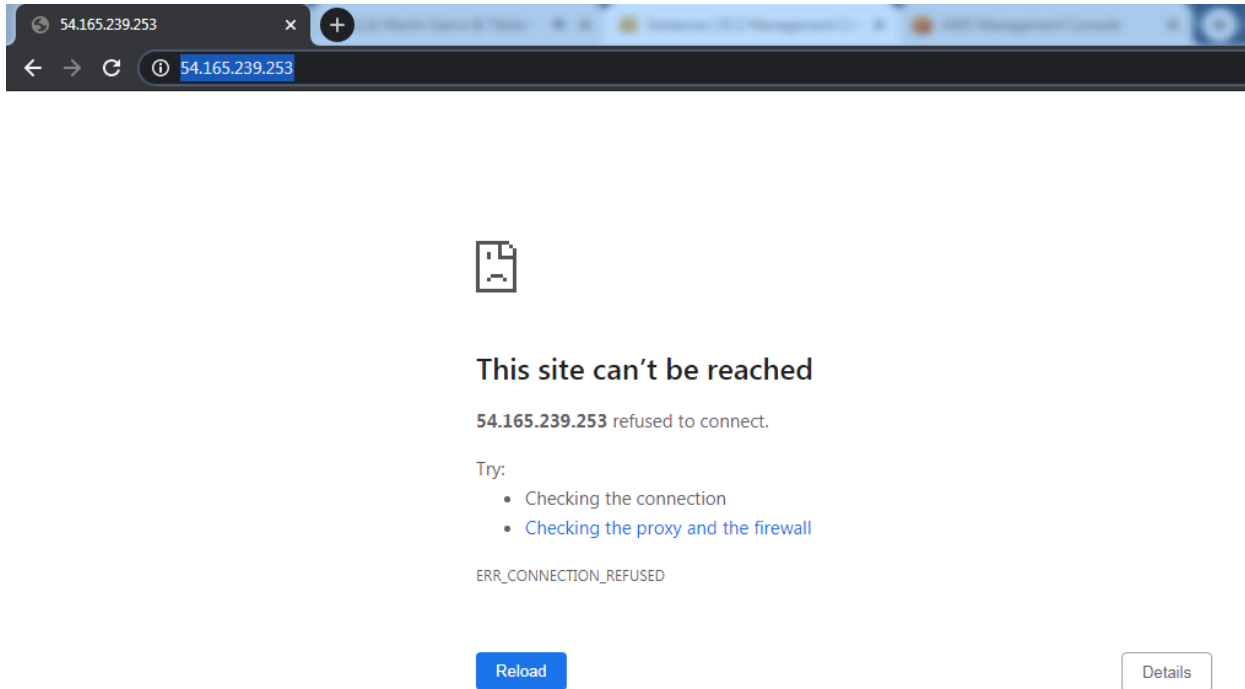
You can see that the Highlighted Instance has been launched Successfully!!!!



Copy the Public IP Address and paste it in your browser.

You'll observe that the Instance hasn't been configure to launch any Web Application.

Keep the browser open, we'll come back to it later.

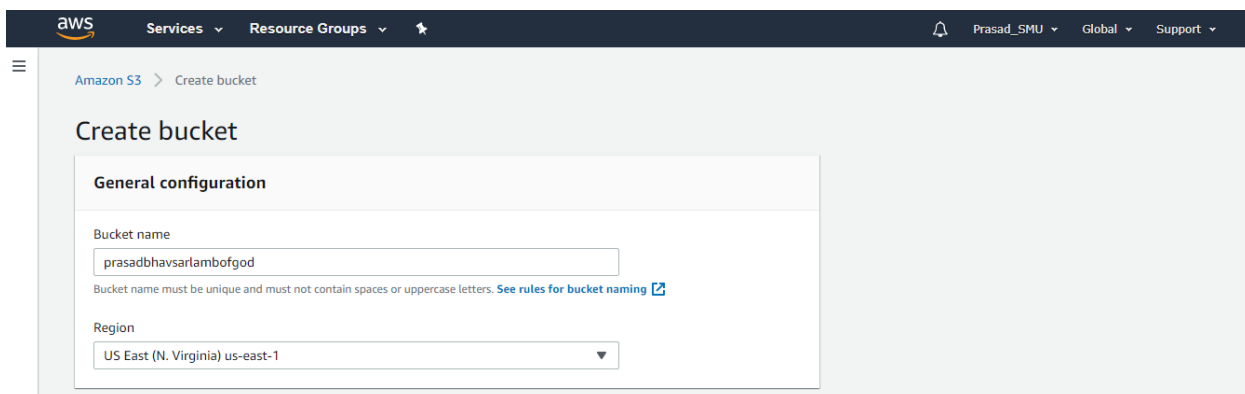


Task 3: Create a S3 Bucket to store SSM Logs

Navigate to S3 Service.

Click on Create Bucket.

Give a unique Bucket Name as per your choice.



Make Bucket publicly Available by unchecking the Block all public access.

Bucket settings for Block Public Access

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block *all* public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

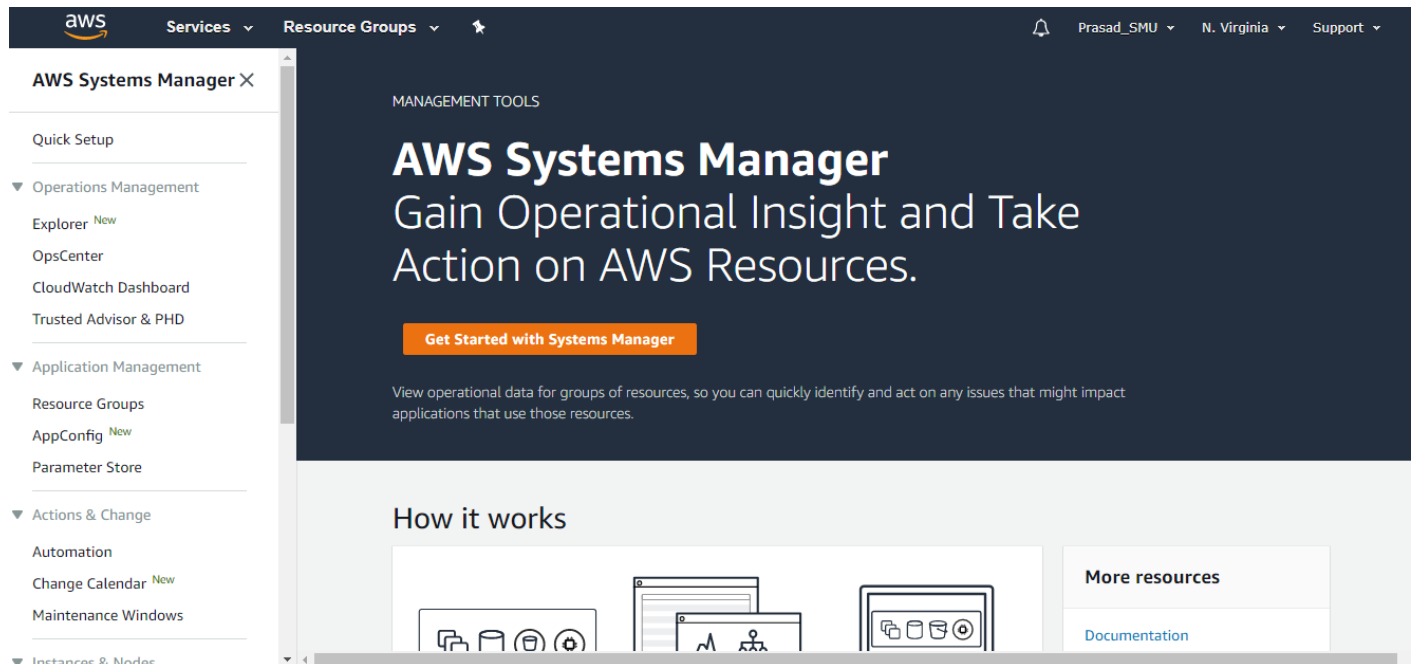
Click on Create. S3 Bucket has been successfully created to store Systems Manager (SSM) logs.

The screenshot shows the AWS Management Console interface for Amazon S3. The left sidebar contains navigation links for Buckets, Batch Operations, Access analyzer for S3, Block public access (account settings), and Feature spotlight. The main content area shows the 'Buckets (2)' page. At the top, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. Below this is a search bar 'Find bucket by name' and a table of buckets. The table has columns: Name, Region, Access, and Bucket created. Two buckets are listed: 'cf-templates-umsgutrdp0mt-us-east-1' and 'prasadbhavsarlambogod'. The 'prasadbhavsarlambogod' bucket is selected, and its 'Access' is 'Objects can be public'. The 'Create bucket' button is visible in the top right corner.

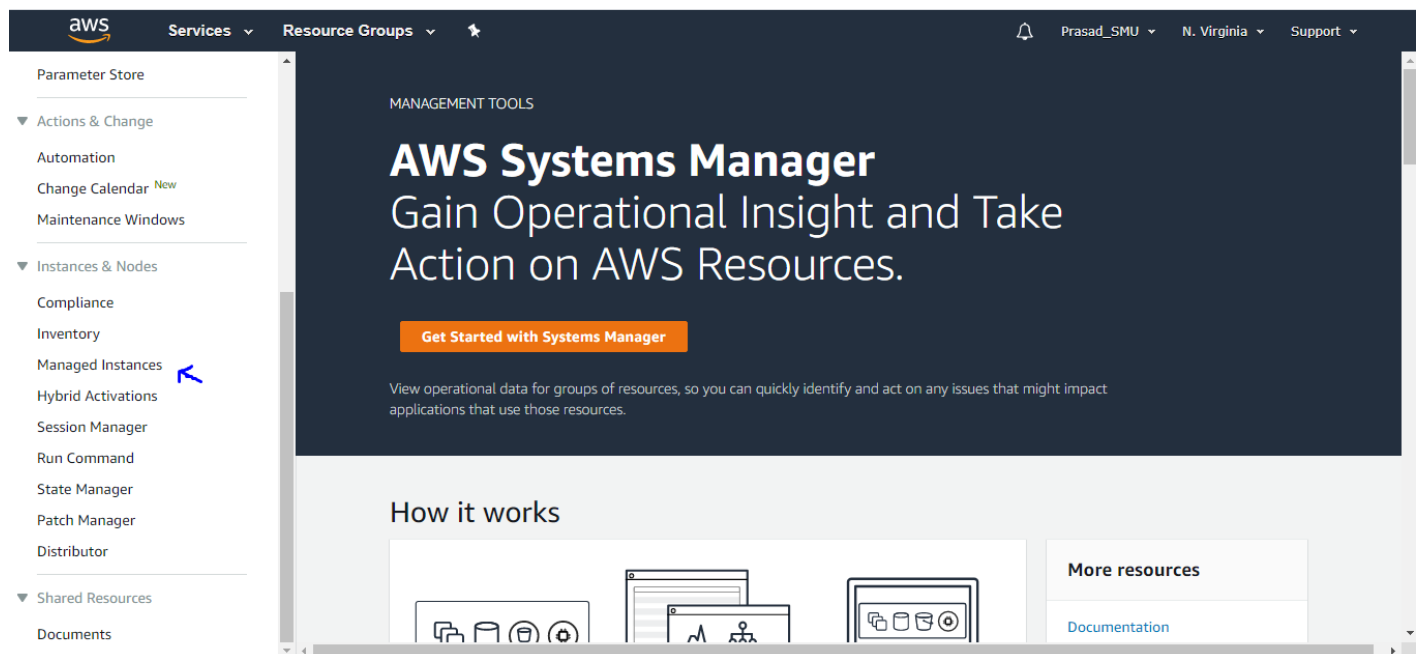
Name	Region	Access	Bucket created
cf-templates-umsgutrdp0mt-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	2020-04-19T08:18:12.000Z
prasadbhavsarlambogod	US East (N. Virginia) us-east-1	Objects can be public	2020-05-02T04:43:40.000Z

Task 4: AWS Systems Manager: Managed Instances

Navigate to AWS Systems Manager Service.



On the left-hand side, click on managed Instances.



You should see Instance which we launched in Task 2.

If you do not see any Instance in Managed Instances tab, it means Systems Manager Agent is not Installed on the EC2 Instance.

AWS Systems Manager

Managed Instances

Managed instances

Instance ID	Name	Ping status	Platform type	Platform name
i-0a25233378569135c	SSM	Online	Linux	Amazon Linux

You can also verify the Instance IDs from EC2 Service Dashboard.

EC2 Dashboard

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-0cb4711711c5a18d8	t2.micro	us-east-1a	stopped		None
	i-0c507b652a99b6408	t2.micro	us-east-1b	running	2/2 checks passed	None
SSM	i-0a25233378569135c	t2.micro	us-east-1a	running	2/2 checks passed	None

Instance: i-0a25233378569135c (SSM) Public DNS: ec2-54-165-239-253.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID i-0a25233378569135c

Instance state running

Instance type t2.micro

Finding Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)

Private DNS ip-10-192-10-197.ec2.internal

Private IPs 10.192.10.197

Secondary private IPs

VPC ID vpc-062814d035612343e (Custom VPC)

Public DNS (IPv4) ec2-54-165-239-253.compute-1.amazonaws.com

IPv4 Public IP 54.165.239.253

IPv6 IPs -

Elastic IPs

Availability zone us-east-1a

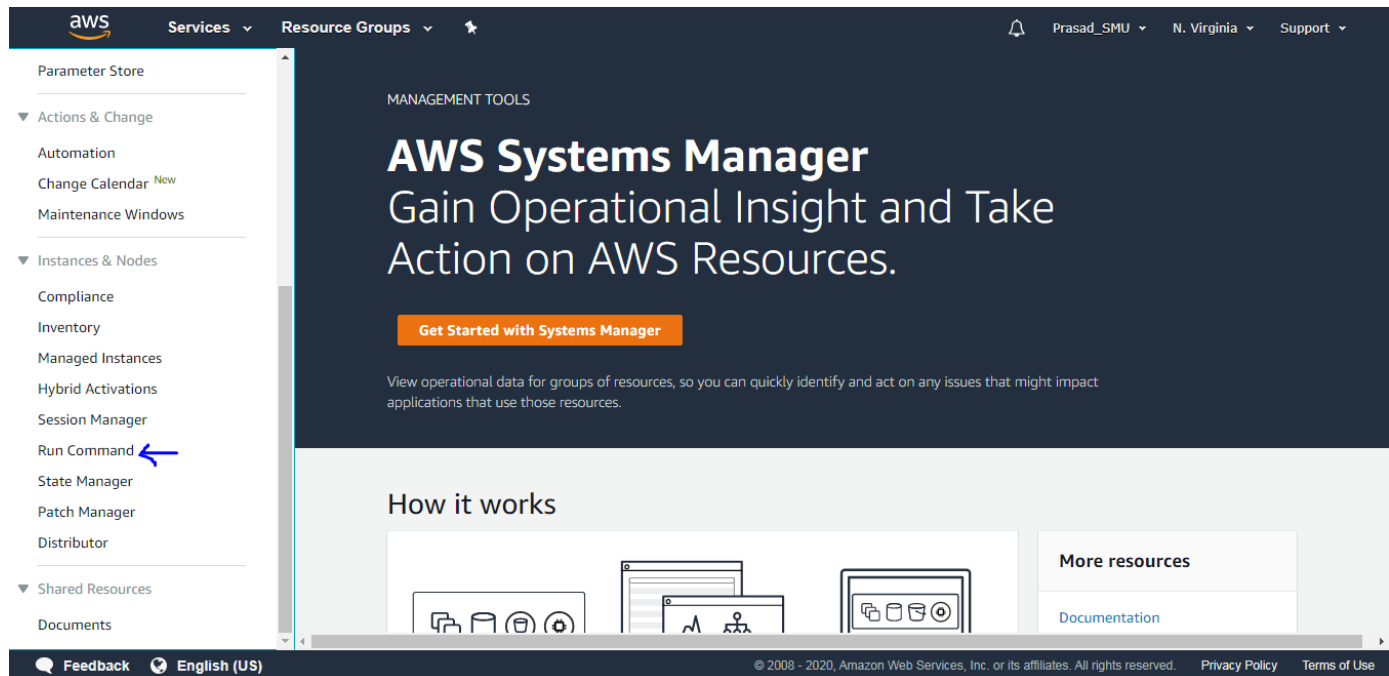
Security groups RHEL8-SG, view inbound rules, view outbound rules

Scheduled events No scheduled events

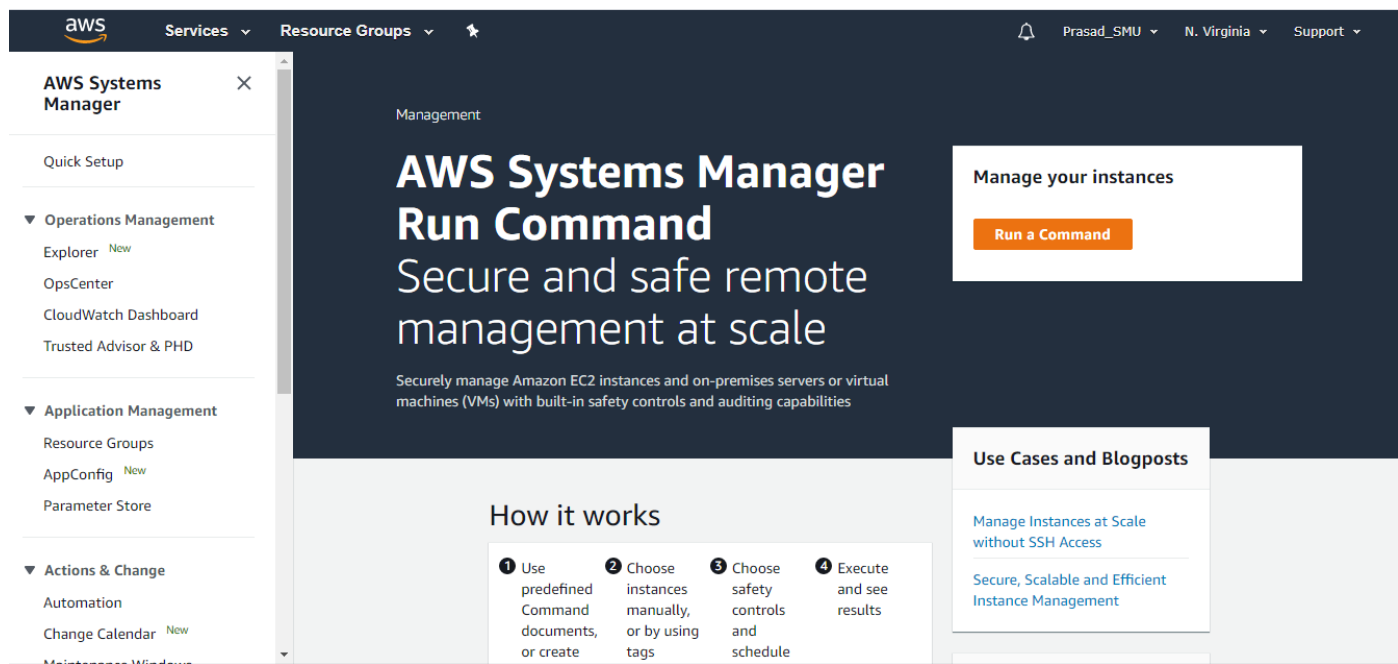
AMI ID amzn2-ami-hvm-2.0.20200406.0-x86_64-
nn7 / ami.0373e3d47d7a7637d1

Task 5: AWS-Systems Manager: Run Command (Ansible Installation)

Now under the same Service, on the left-hand side, click on Run Command.

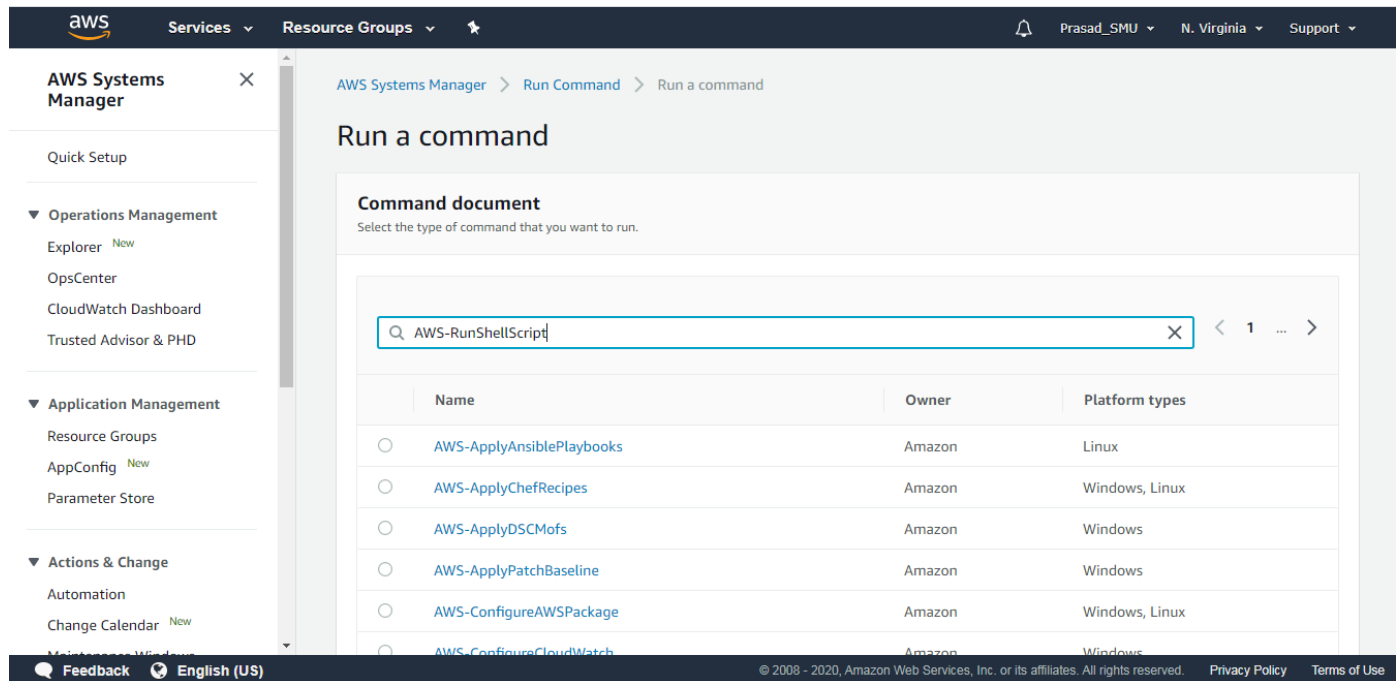


Click on Run a Command.



Under Command Document, search for the below AWS Managed Document.

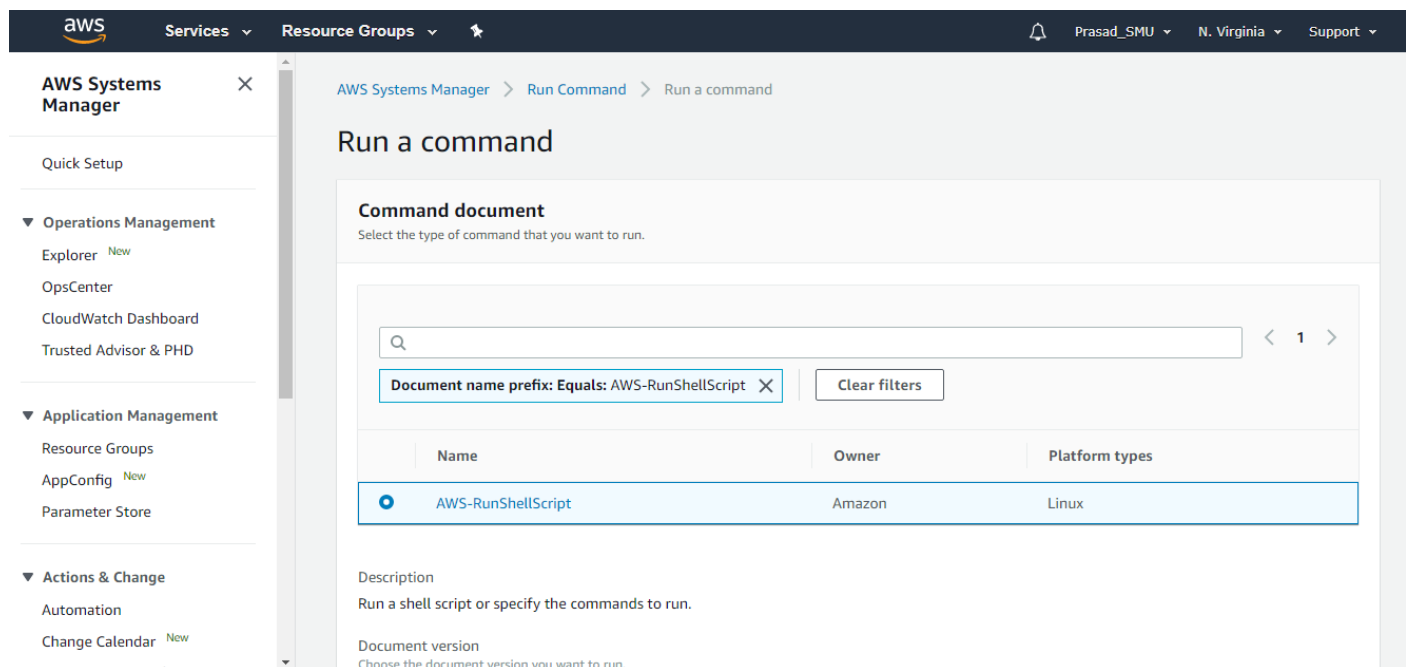
AWS-RunShellScript



The screenshot shows the AWS Systems Manager console. The left sidebar contains navigation options like Quick Setup, Operations Management, Application Management, and Actions & Change. The main content area is titled 'Run a command' and 'Command document'. A search bar at the top of the document list contains the text 'AWS-RunShellScript'. Below the search bar, a table lists several command documents:

	Name	Owner	Platform types
<input type="radio"/>	AWS-ApplyAnsiblePlaybooks	Amazon	Linux
<input type="radio"/>	AWS-ApplyChefRecipes	Amazon	Windows, Linux
<input type="radio"/>	AWS-ApplyDSCMofs	Amazon	Windows
<input type="radio"/>	AWS-ApplyPatchBaseline	Amazon	Windows
<input type="radio"/>	AWS-ConfigureAWSPackage	Amazon	Windows, Linux
<input type="radio"/>	AWS-ConfigureCloudWatch	Amazon	Windows

Select the Command Document.



The screenshot shows the same AWS Systems Manager console. The search bar is now empty. A filter is applied: 'Document name prefix: Equals: AWS-RunShellScript'. The table below shows only one result, which is selected with a radio button:

	Name	Owner	Platform types
<input checked="" type="radio"/>	AWS-RunShellScript	Amazon	Linux

Below the table, the 'Description' and 'Document version' sections are visible.

Read the highlighted Description.

Command document
Select the type of command that you want to run.

< 1 >

Document name prefix: Equals: AWS-RunShellScript X Clear filters

Name	Owner	Platform types
<input checked="" type="radio"/> AWS-RunShellScript	Amazon	Linux

Description
Run a shell script or specify the commands to run.

Document version
Choose the document version you want to run.

1 (Default) ▼

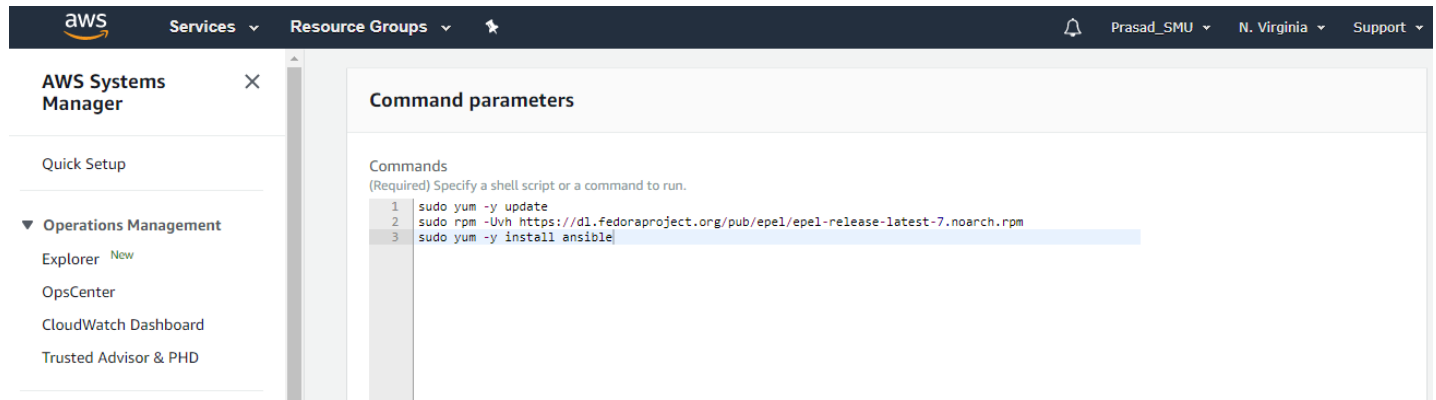
Under Targets, click on Choose Instances Manually and select both the EC2 Instances.

You can also select Instances using Tags.

Instances						
<input type="text"/>						
< 1 >						
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Availability zone	Ping status	Last pin
<input checked="" type="checkbox"/>	SSM	i-0a25233378569135c	running	us-east-1a	Online	02/05/2023:02:30:51 (Central Time)

Type the below Script/Commands under Command Parameters. I've provided the Script in Text File.

This script does the Ansible Installation on the Target Instance.



aws Services Resource Groups

Prasad_SMU N. Virginia Support

AWS Systems Manager

Quick Setup

▼ Operations Management

Explorer New

OpsCenter

CloudWatch Dashboard

Trusted Advisor & PHD

Command parameters

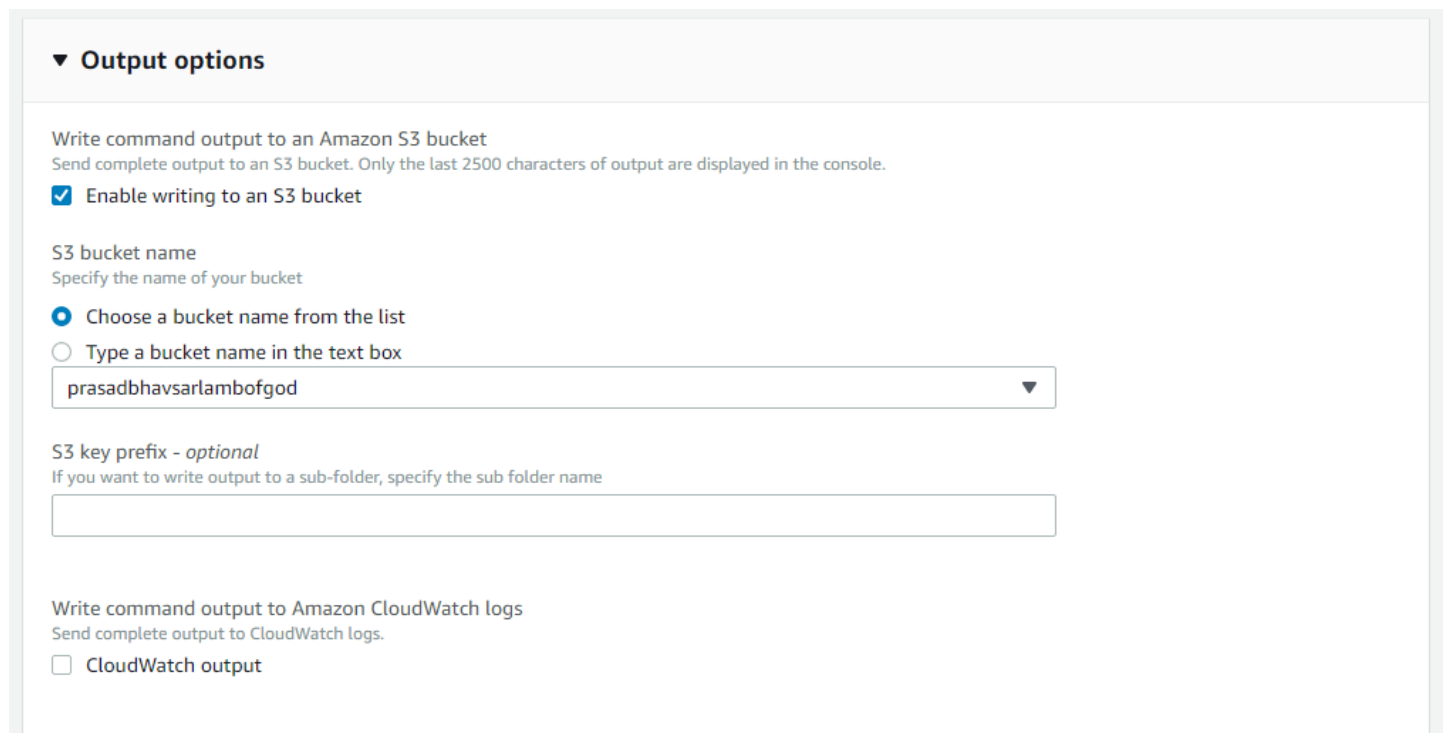
Commands
(Required) Specify a shell script or a command to run.

```
1 sudo yum -y update
2 sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
3 sudo yum -y install ansible
```

You can now specify the S3 Bucket Name wherein Systems Manager logs will be saved.

Logs in the S3 Bucket will be saved in stdout.txt and stderr.txt format.

Stderr.txt file is quite useful if the Ansible Installation fails.



▼ **Output options**

Write command output to an Amazon S3 bucket
Send complete output to an S3 bucket. Only the last 2500 characters of output are displayed in the console.

☒ Enable writing to an S3 bucket

S3 bucket name
Specify the name of your bucket

☒ Choose a bucket name from the list

☐ Type a bucket name in the text box

prasadbhavsarlambogod ▼

S3 key prefix - optional
If you want to write output to a sub-folder, specify the sub folder name

Write command output to Amazon CloudWatch logs
Send complete output to CloudWatch logs.

☐ CloudWatch output

Task 6: Ansible Installation Check

Make a note of Command ID and keep observing Overall Status.

The screenshot shows the AWS Systems Manager console. A green banner at the top indicates the command was successfully sent. The command ID is 86cbd007-4d7e-49d1-8b39-79948e92d377. The command status is 'In Progress'. The 'Targets and outputs' section shows one target with ID i-0a25233378569135c, named ip-10-192-10-197.ec2.internal, with a status of 'In Progress'.

Overall status	Detailed status	# targets	# completed	# error	# delivery timed out
In Progress	In Progress	1	0	0	0

Instance ID	Instance name	Status	Detailed Status	Start time	Finish time
i-0a25233378569135c	ip-10-192-10-197.ec2.internal	In Progress	In Progress		

Status changes to **SUCCESS**.

The screenshot shows the AWS Systems Manager console. A green banner at the top indicates the command was successfully sent. The command ID is 86cbd007-4d7e-49d1-8b39-79948e92d377. The command status is 'Success'. The 'Targets and outputs' section shows one target with ID i-0a25233378569135c, named ip-10-192-10-197.ec2.internal, with a status of 'Success'.

Overall status	Detailed status	# targets	# completed	# error	# delivery timed out
Success	Success	1	1	0	0

Instance ID	Instance name	Status	Detailed Status	Start time	Finish time
i-0a25233378569135c	ip-10-192-10-197.ec2.internal	Success	Success	Sat, 02 May 2020 07:38:06 GMT	Sat, 02 May 2020 07:38:18 GMT

Take SSH session of Target Instance.

Issue the below commands.

Command: which ansible

The screenshot shows the AWS Management Console. On the left, the 'INSTANCES' section is expanded, showing a list of instances. The instance 'i-0a25233378569135c' (SSM) is selected. The 'Description' tab is active, showing details like Instance ID, Instance state (running), Instance type (t2.micro), Private DNS (ip-10-192-10-197.ec2.internal), and Private IPs (10.192.10.197). An overlay terminal window shows the command 'which ansible' being executed, returning '/usr/bin/ansible', indicating Ansible is installed.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm State
	i-0cb4711711c5a18d8	t2.micro	us-east-1a	stopped		None
	i-0c507b652a99b6408	t2.micro	us-east-1b	running	2/2 checks passed	None
SSM	i-0a25233378569135c	t2.micro	us-east-1a	running	2/2 checks passed	None

Instance: i-0a25233378569135c (SSM) Public DNS: ec2-54-10-192-10-197.us-east-1.compute.amazonaws.com

Instance ID: i-0a25233378569135c

Instance state: running

Instance type: t2.micro

Finding: Opt-in to AWS Compute Optimizer recommendations. [Learn more](#)

Private DNS: ip-10-192-10-197.ec2.internal

Private IPs: 10.192.10.197

Secondary private IPs: None

VPC ID: vpc-062814d035612343e (Custom VPC)

AMI ID: amzn2-ami-hvm-2.0.20200406.0-x86_64-ann7 / ami-0373e3dd7da7db37d1

```
ec2-user@ip-10-192-10-197:~$ which ansible
/usr/bin/ansible
ec2-user@ip-10-192-10-197:~$
```

Ansible has been successfully installed on the Target Instance.

Now navigate to S3 Service and click on your S3 Bucket.

You'll notice that a new object for SSM logs have been created.

The screenshot shows the AWS S3 console. The bucket 'prasadbhavsarlambogod' is selected. The 'Overview' tab is active, showing the bucket's details. The 'Properties' tab is also visible. The bucket is located in the 'US East (N. Virginia)' region. The 'Actions' dropdown menu is open, showing options like 'Upload', 'Create folder', 'Download', and 'Actions'. The bucket contains one object: '86cbd007-4d7e-49d1-8b39-79948e92d377'.

Amazon S3 > prasadbhavsarlambogod

prasadbhavsarlambogod

Overview Properties Permissions Management Access points

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

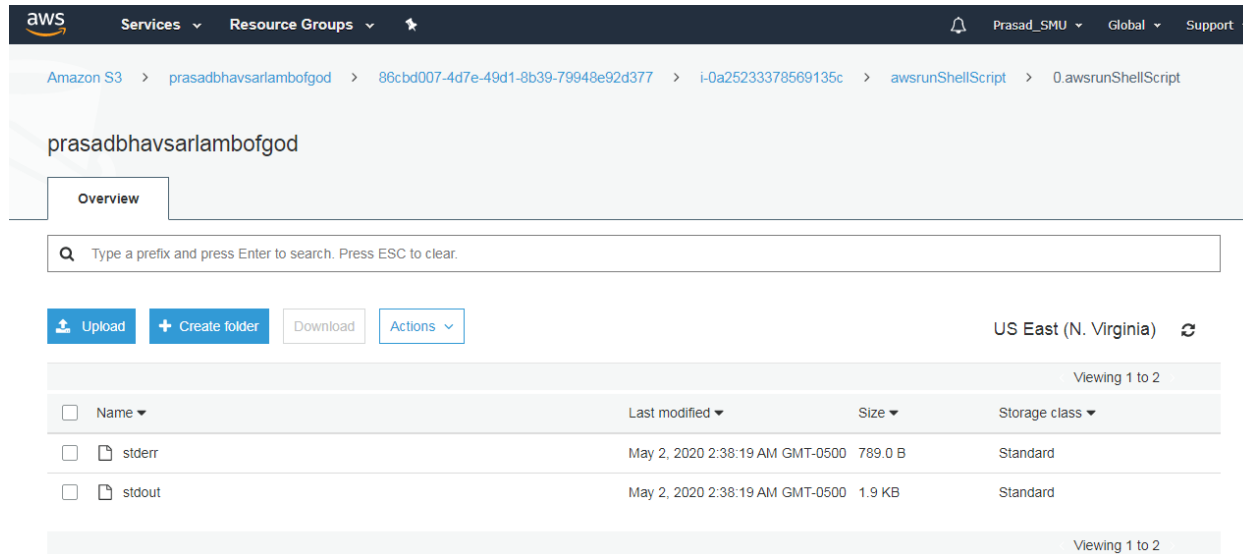
US East (N. Virginia)

Viewing 1 to 1

Name	Last modified	Size	Storage class
86cbd007-4d7e-49d1-8b39-79948e92d377	--	--	--

Viewing 1 to 1

Download the stderr.txt and stdout.txt if you want to check the SSM Logs.



aws Services Resource Groups

Amazon S3 > prasadbhavsarlambofgod > 86cbd007-4d7e-49d1-8b39-79948e92d377 > i-0a25233378569135c > awsrnShellScript > 0.awsrnShellScript

prasadbhavsarlambofgod

Overview

Search: Type a prefix and press Enter to search. Press ESC to clear.


Upload Create folder Download Actions

US East (N. Virginia)

Name	Last modified	Size	Storage class
stderr	May 2, 2020 2:38:19 AM GMT-0500	789.0 B	Standard
stdout	May 2, 2020 2:38:19 AM GMT-0500	1.9 KB	Standard

Task 7: Run an Ansible Playbook using AWS Systems Manager

Navigate to AWS Systems Manager Service and on right-hand side click on **State Manager**.



aws Services Resource Groups

Application Management

- Resource Groups
- AppConfig New
- Parameter Store
- Actions & Change
 - Automation
 - Change Calendar New
 - Maintenance Windows
- Instances & Nodes
 - Compliance
 - Inventory
 - Managed Instances
 - Hybrid Activations
 - Session Manager
 - Run Command
 - State Manager ←
 - Patch Manager
 - Distributor

MANAGEMENT TOOLS

AWS Systems Manager

Gain Operational Insight and Take Action on AWS Resources.

Get Started with Systems Manager

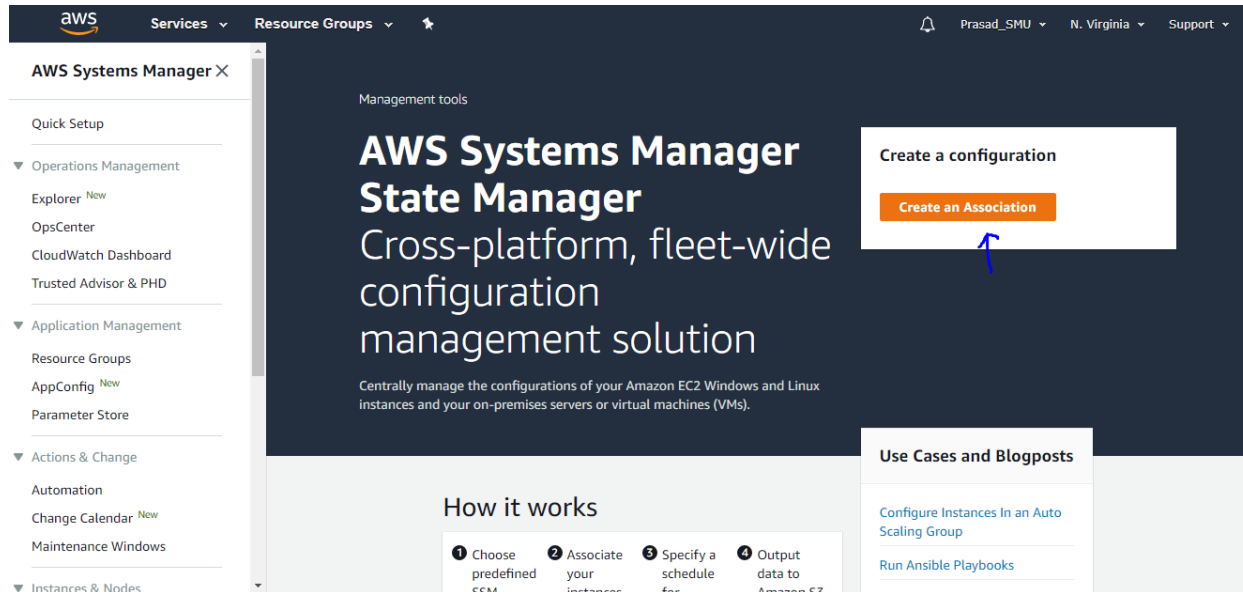
View operational data for groups of resources, so you can quickly identify and act on any issues that might impact applications that use those resources.

How it works

More resources

Documentation

Now click on **Create Association**.



Specify the Association Name as per your choice.

Provide association details

Name - *optional*

Provide a name for your Association.

Search for the AWS Managed Document **AWS-RunAnsiblePlaybook**.

The screenshot shows the AWS Systems Manager console. The left sidebar contains navigation options: Quick Setup, Operations Management (Explorer, OpsCenter, CloudWatch Dashboard, Trusted Advisor & PHD), Application Management (Resource Groups, AppConfig, Parameter Store), Actions & Change (Automation, Change Calendar, Maintenance Windows), and Instances & Nodes. The main content area is titled 'Document' and shows details for 'AWS-ASGEnterStandby'. Below this, a search bar contains the text 'AWS-RunAnsiblePlaybook'. A table lists search results:

	Name	Owner	Platform types	Document type
<input checked="" type="radio"/>	AWS-ASGEnterStandby	Amazon	Windows, Linux	Automation
<input type="radio"/>	AWS-ASGExitStandby	Amazon	Windows, Linux	Automation
<input type="radio"/>	AWS-ApplyAnsiblePlaybooks	Amazon	Linux	Command
<input type="radio"/>	AWS-ApplyChefRecipes	Amazon	Windows, Linux	Command
<input type="radio"/>	AWS-ApplyDSCMofs	Amazon	Windows	Command

Select the AWS Managed Document **AWS-RunAnsiblePlaybook**.

The screenshot shows the AWS Systems Manager console with the search results from the previous step. The 'Document' section now displays details for 'AWS-RunAnsiblePlaybook'. The search bar is empty, and a filter is applied: 'Document name prefix: Equals: AWS-RunAnsiblePlaybook'. The table below shows the selected document:

	Name	Owner	Platform types	Document type
<input checked="" type="radio"/>	AWS-RunAnsiblePlaybook	Amazon	Linux	Command

Write the below Playbook in the Playbook section. I'll provide the Playbook in the .txt format.

```
---
- name: linux_deploy_httpd
  hosts: all
  tasks:
    - name: Install HTTPD
      yum:
        name: "{{ item }}"
        state: latest
      loop:
        - httpd
    - name: Setting default HTTP Server page
      shell: echo "<h1>Hey Prasad, Ansible has successfully configured Apache Server!!!</h1>" >> /var/www/html/index.html
    - name: Start Apache Webserver
      service:
        name: httpd
        state: restarted
    - name: enable apache on startup and start service for redhat or centos
      service: name=httpd enabled=yes state=started
```

This playbook installs the httpd package, starts the httpd service and enable httpd service to start while bootup on the target Instance.

Parameters

Playbook

(Optional) If you don't specify a URL, then you must specify playbook YAML in this field.

```
- name: enable apache on startup and start service for redhat or centos
  service: name=httpd enabled=yes state=started
```

Playbookurl

(Optional) If you don't specify playbook YAML, then you must specify a URL where the playbook is stored. You can specify the URL in the following formats: http://example.com/playbook.yml or s3://examplebucket/plabook.url. For security reasons, you can't specify a URL with quotes.

Extravars

(Optional) Additional variables to pass to Ansible at runtime. Enter a space separated list of key/value pairs. For example: color=red flavor=lime

SSM=True

Check

(Optional) Use the check parameter to perform a dry run of the Ansible execution.

False

Under Targets, click on Choose Instances Manually and select both the EC2 Instances.

You can also select Instances using Tags.

Targets

Targets are the instances you would like to associate with this document. You can choose to target by both managed instance and tag.

Select targets by

☐ Selecting all managed instances in this region under this account

☐ Specifying tags

☒ Manually Selecting Instance

i-0a25233378569135c

X

< 1 >

<input checked="" type="checkbox"/>	Name	Instance id	Instance state	Availability zone	Ping st
<input checked="" type="checkbox"/>	SSM	i-0a25233378569135c	running	us-east-1a	Online

Since we are going to run the Playbook only once, under Specify Schedule, select **No Schedule**.

Specify schedule

On Schedule

☐

Run association at cron/rate intervals.

No schedule

☒

Run association once.

Advanced options

Compliance severity

Specify association compliance severity. This will be reflected in your compliance dashboard.

Keep the Compliance Severity CRITICAL.

Advanced options

Compliance severity

Specify association compliance severity. This will be reflected in your compliance dashboard.

Critical

If you have 100s of Servers and you want to run the Playbook on all the Servers but not at a onetime then you can specify the number of Targets under Concurrency.

Do not specify any targets for since we only have only one Instance.

Specify the Error Threshold as One.

▼ Rate control

Concurrency

Specify the number or percentage of targets on which to execute the task at the same time



targets



percentage

Error threshold

Stop the task after the task fails on the specified number or percentage of targets



error



percentage

You can now specify the S3 Bucket Name wherein Systems Manager logs will be saved.

Logs in the S3 Bucket will be saved in stdout.txt and stderr.txt format.

Stderr.txt file is quite useful if the Playbook fails. Click on Create.

Output options

Write to S3
Write all command output to an Amazon S3 bucket. Command output in the console is truncated after 2500 characters.

☒ Enable writing output to S3

S3 bucket name
Specify the name of your bucket.

S3 key prefix - optional
Type a prefix for the bucket that receives the output; for example, mycommands/domainjoin.

Task 8: Verify Ansible Playbook Execution

The association status is currently Pending.

Services

Resource Groups

Prasad_SMU

N. Virginia

Support

AWS Systems Manager

Quick Setup

Operations Management

Explorer

OpsCenter

CloudWatch Dashboard

Trusted Advisor & PHD

Application Management

Resource Groups

AppConfig

Parameter Store

Actions & Change

Automation

Change Calendar

Maintenance Windows

Create association request succeeded

View details

AWS Systems Manager > State Manager

Associations

View details

Apply association now

Edit

Delete

Create association

Association id

Association name

Document name

Last execution date

Status

Association version

Resource status count

2ea8b1f3-13e8-42ea-a644-489d70174f58

InstallApache

AWS-RunAnsiblePlaybook

Pending

1

The association status is now Success.

The screenshot shows the AWS Systems Manager console. The breadcrumb trail is: AWS Systems Manager > State Manager > Association ID : 2ea8b1f3-13e8-42ea-a644-489d70174f58 > Description. The main heading is 'Association ID: 2ea8b1f3-13e8-42ea-a644-489d70174f58'. Below it are buttons for 'Apply association now', 'Edit', and 'Delete'. A tabbed interface shows 'Description' as the active tab. The details are as follows:

Description	Resources	Parameters	Targets	Versions	Execution history
Document name				Association name	
AWS-RunAnsiblePlaybook				InstallApache	
Document version				Association version	
\$DEFAULT				1	
Status				Association id	
Success				2ea8b1f3-13e8-42ea-a644-489d70174f58	
Create date				Schedule expression	
Sat, 02 May 2020 07:57:07 GMT				-	

Scroll down on the same page and click on S3 Output.

Click on your S3 Bucket.

You'll notice that a new Object for SSM logs have been created.

The screenshot shows the Amazon S3 console for the bucket 'prasadbhavsarlambofgod'. The breadcrumb trail is: Amazon S3 > prasadbhavsarlambofgod. The bucket name 'prasadbhavsarlambofgod' is displayed. Below it are tabs for 'Overview', 'Properties', 'Permissions', 'Management', and 'Access points'. A search bar is present with the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', 'Create folder', 'Download', and 'Actions'. The region is 'US East (N. Virginia)'. A table shows the contents of the bucket:

Name	Last modified	Size	Storage class
1a776902-b8c3-427b-bece-0082c81c9222	--	--	--
86cbd007-4d7e-49d1-8b39-79948e92d377	--	--	--

A blue arrow points to the first object, '1a776902-b8c3-427b-bece-0082c81c9222'. The footer of the table indicates 'Viewing 1 to 2'.

Click on the Objects and Sub-Objects till you see the stderr and stdout files.

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

US East (N. Virginia)

Name	Last modified	Size	Storage class
stderr	May 2, 2020 2:57:25 AM GMT-0500	292.0 B	Standard
stdout	May 2, 2020 2:57:25 AM GMT-0500	1.2 KB	Standard

Download and open the stdout file, you'll notice that the Playbook has been executed successfully.

Overview

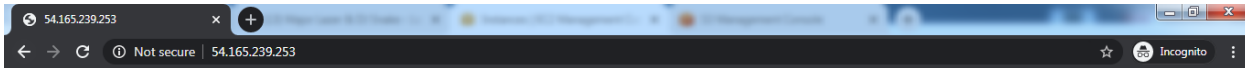
Type a prefix and press

Upload Create folder

Name
stderr
stdout

```
ansible 2.9.7 config file = /etc/ansible/ansible.cfg configured module search path =
[u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules'] ansible python module location =
/usr/lib/python2.7/site-packages/ansible executable location = /usr/bin/ansible python version = 2.7.16 (default, Dec
12 2019, 23:58:22) [GCC 7.3.1 20180712 (Red Hat 7.3.1-6)]PLAY [linux_deploy_httpd]
*****TASK [Gathering Facts]*****
ok: [localhost]TASK [Install HTTPD]
changed: [localhost] => (item=httpd)TASK [Setting default
HTTP Server page]
changed: [localhost]TASK [Start Apache webserver]
*****changed: [localhost]TASK [enable apache on startup and start service
for redhat or centos] *****changed: [localhost]PLAY RECAP
*****localhost
unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
ok=5 changed=4
```

Now again put the Public IP Address of the EC2 Instance in the browser, you'll notice that the Website Opened Successfully!!!!!!



Hey Prasad, Ansible has successfully configured Apache Server!!!!

This completes the lab on Running Ansible Playbook using AWS System Manager.

For questions, contact me on pbhavsar@smu.edu .