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

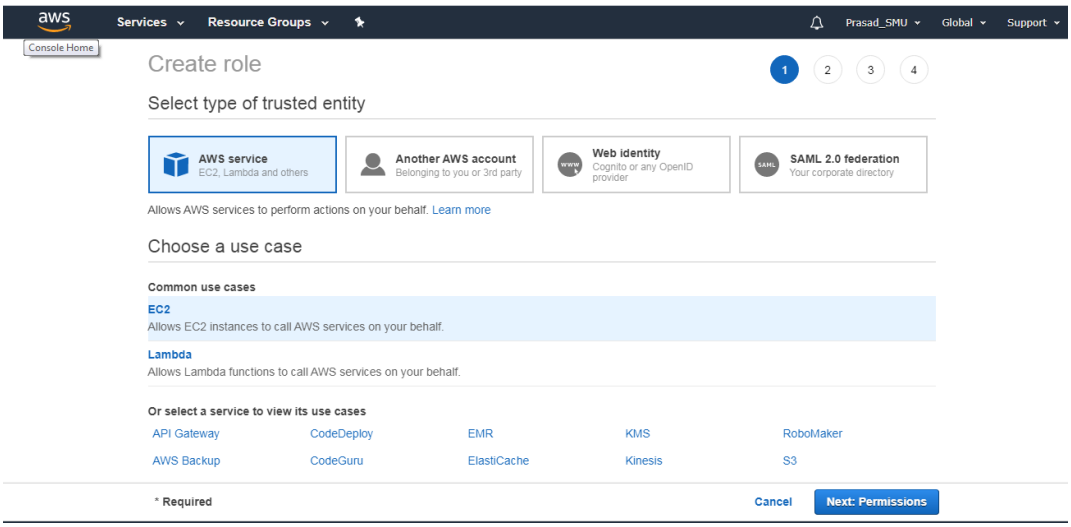
Login to the AWS Management Console.

Navigate to IAM Service and click on Roles.

Click on Create Role.

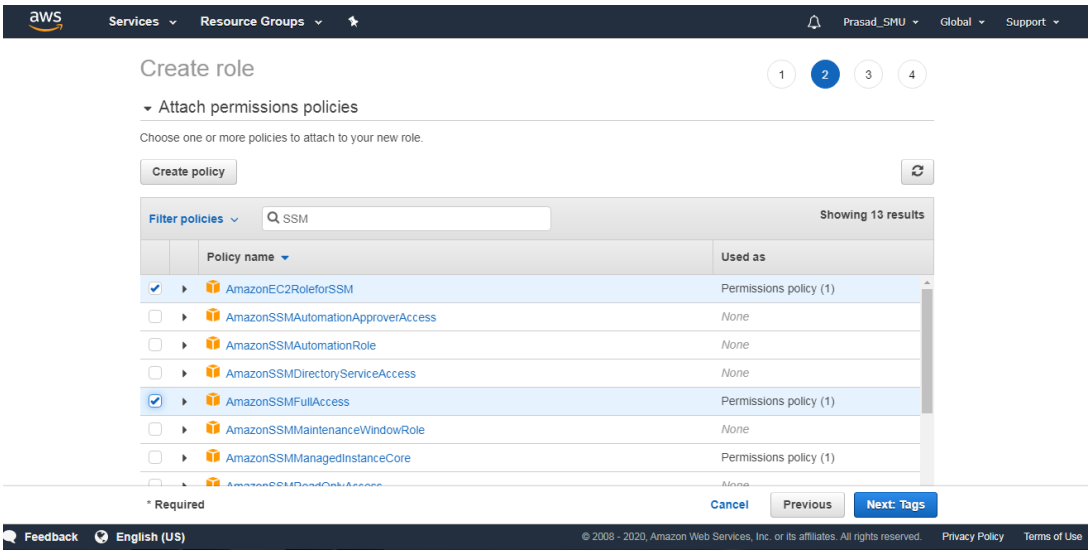
Make sure to select the Use Case as **EC2**.

Click Next: Permissions.

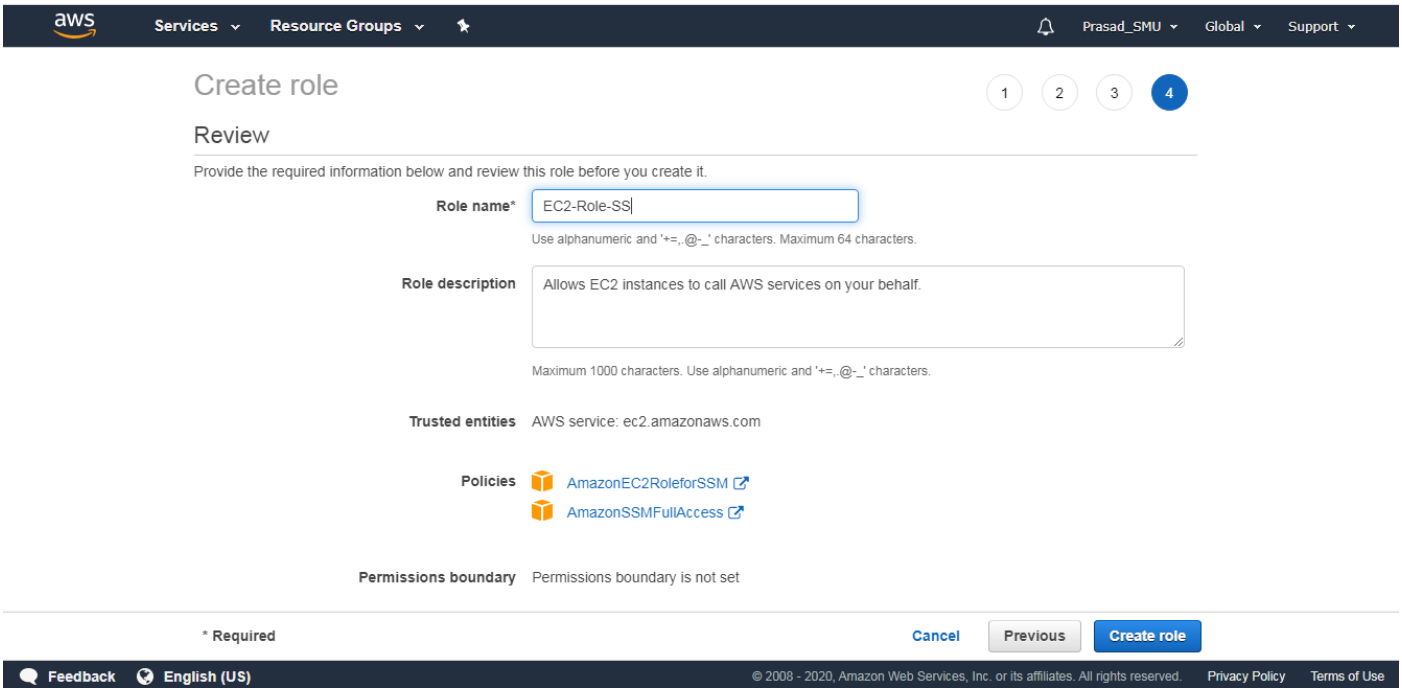


Select the below two Default IAM Policies:

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



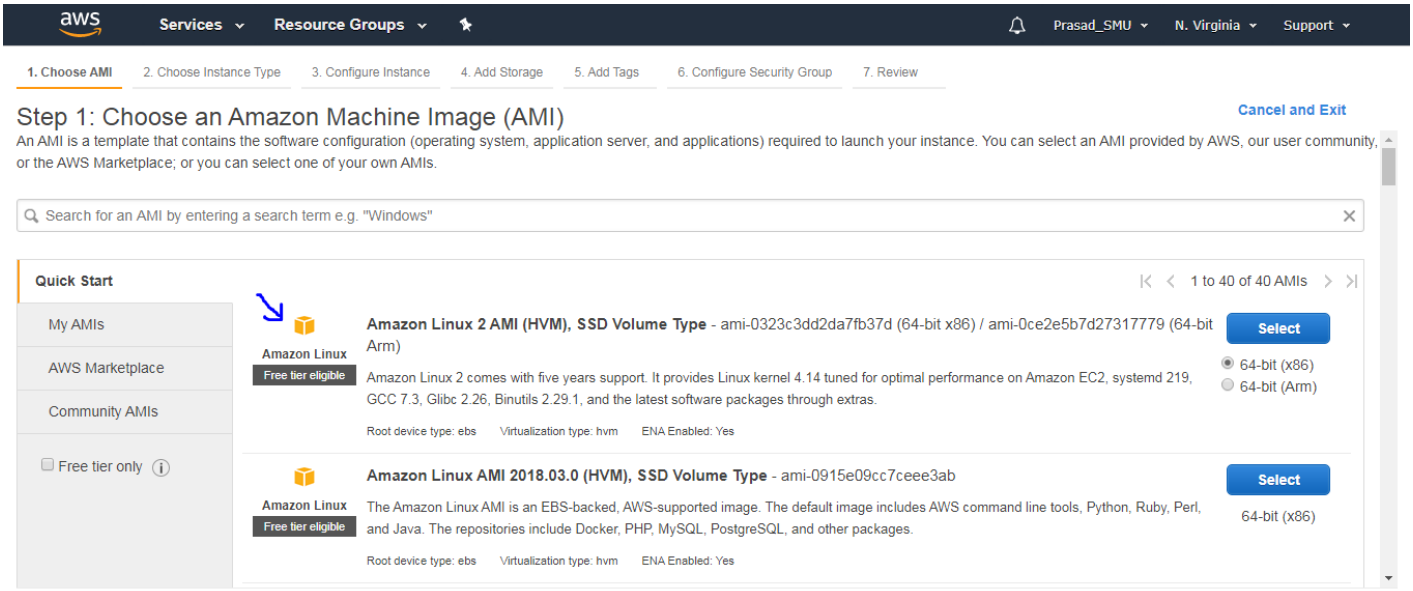
Give the Role Name as per your Choice and click on Create Role.



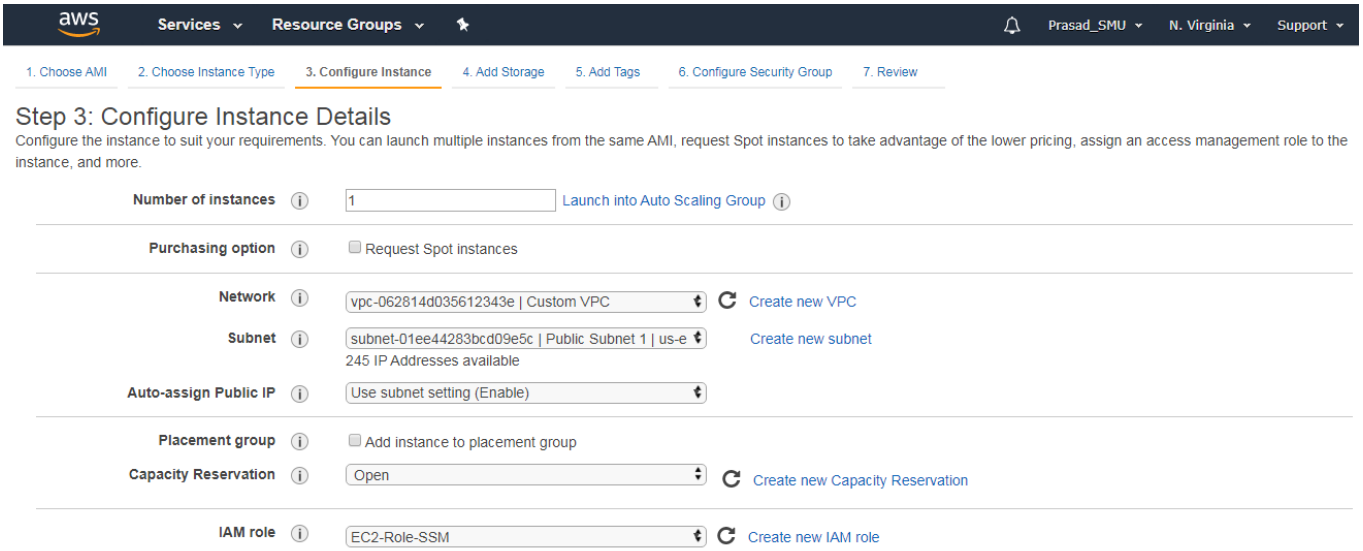
Task 2: Launch & Configure EC2 Instances with SSM Agent

Navigate to EC2 Service and click on Launch Instance.

Select the **Amazon Linux AMI**.



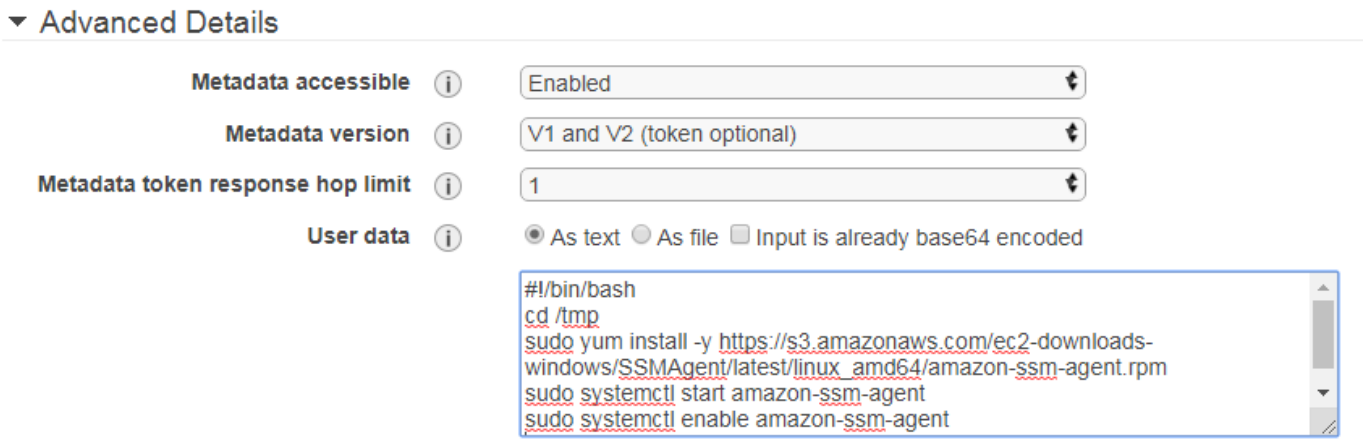
Select the Number of Instances as 1, select the Network as our Custom VPC, Select Subnet as Public Subnet 1 and select the IAM Role which you configured in the Task 1.



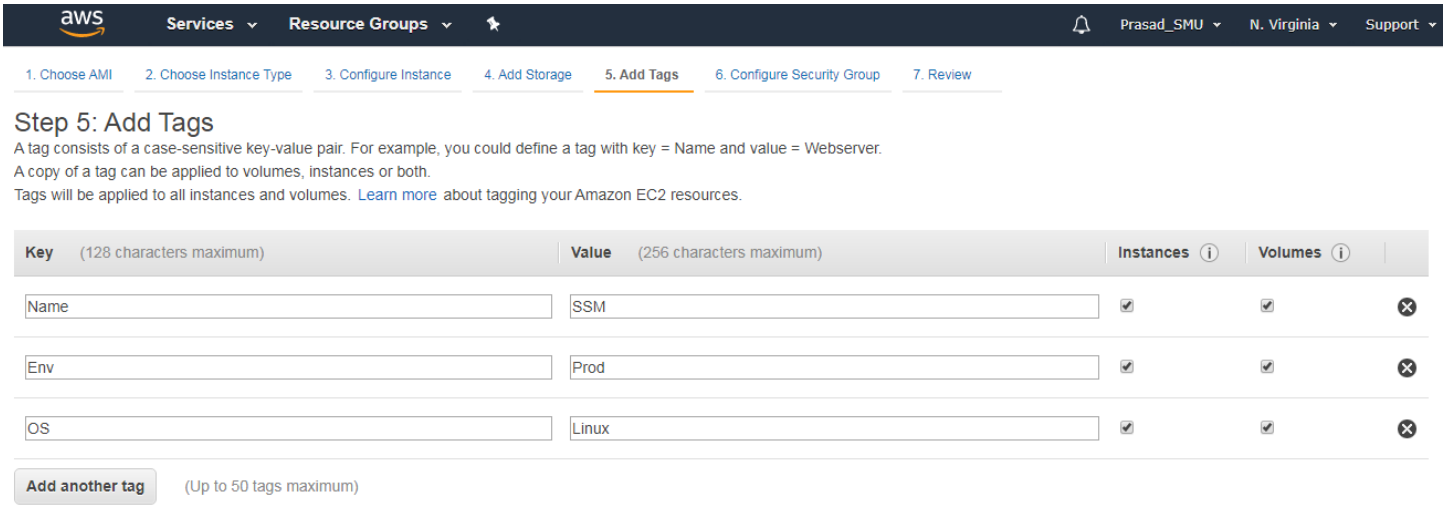
Since AWS Systems Manager is AGENTLESS, we need to install Packages for Systems Manager (SSM) to connect with Target Instances.

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.

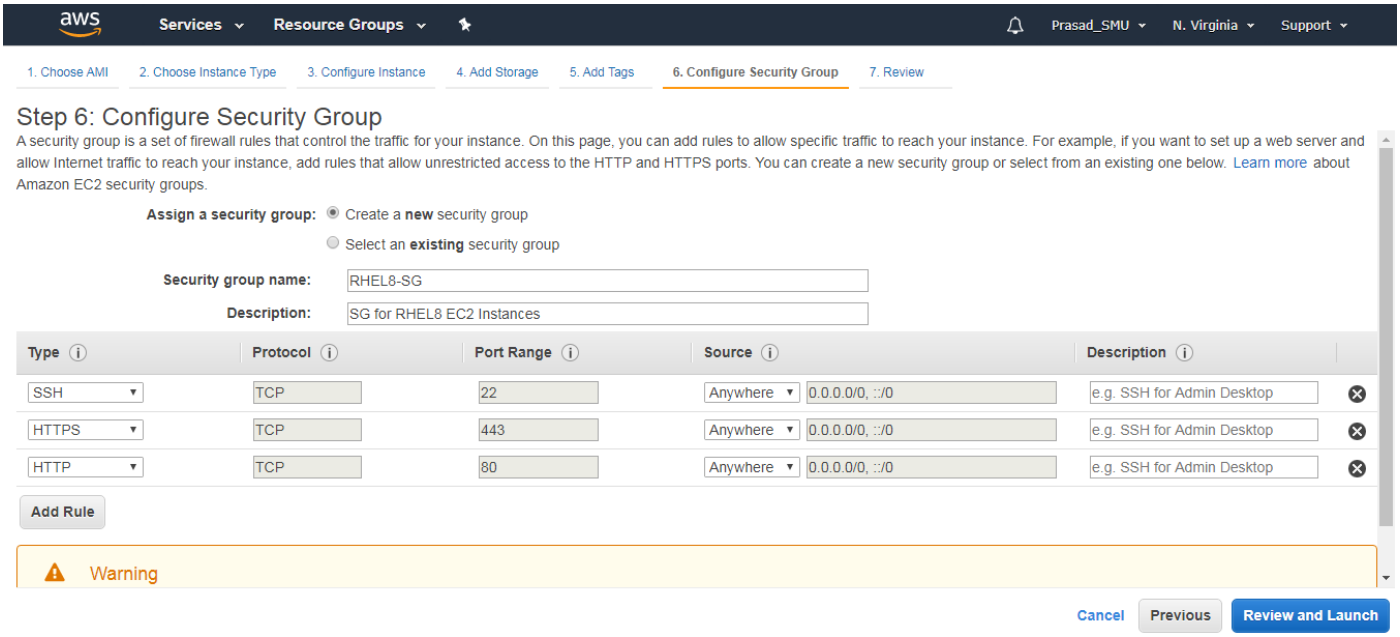


You can mention Tags as per your choice.



Click Next: Security Groups.

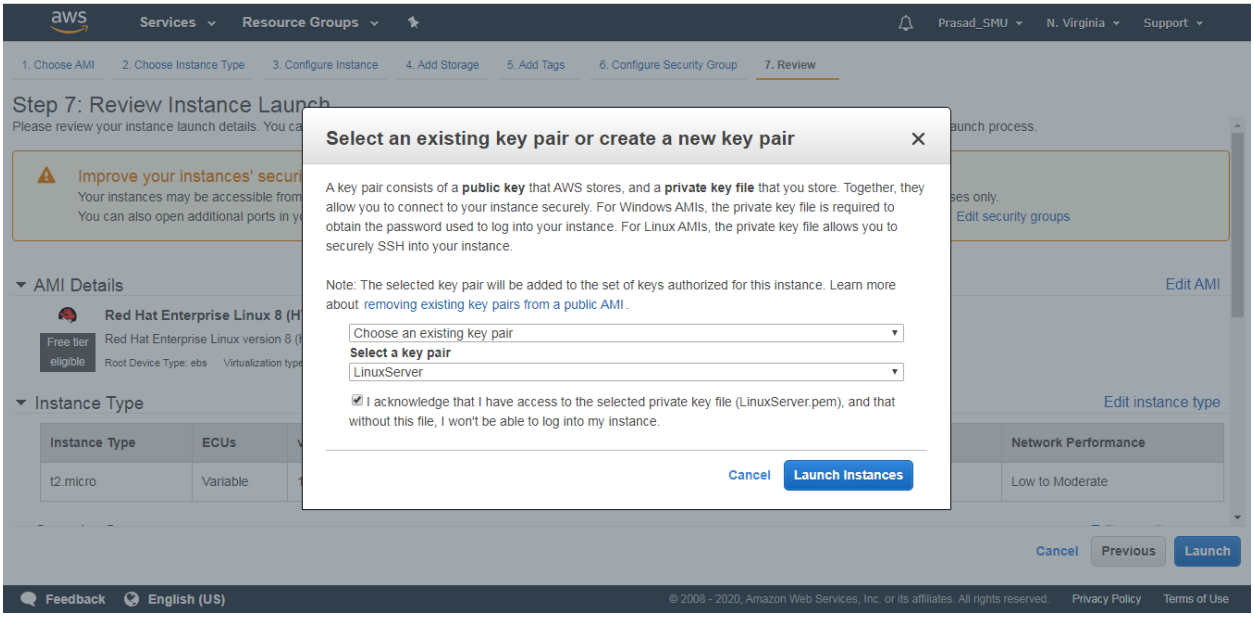
Create a new Security Group. Give the Name & Discription as per your choice. Allow SSH, HTTPS, HTTP Inbound traffic from Anywhere.



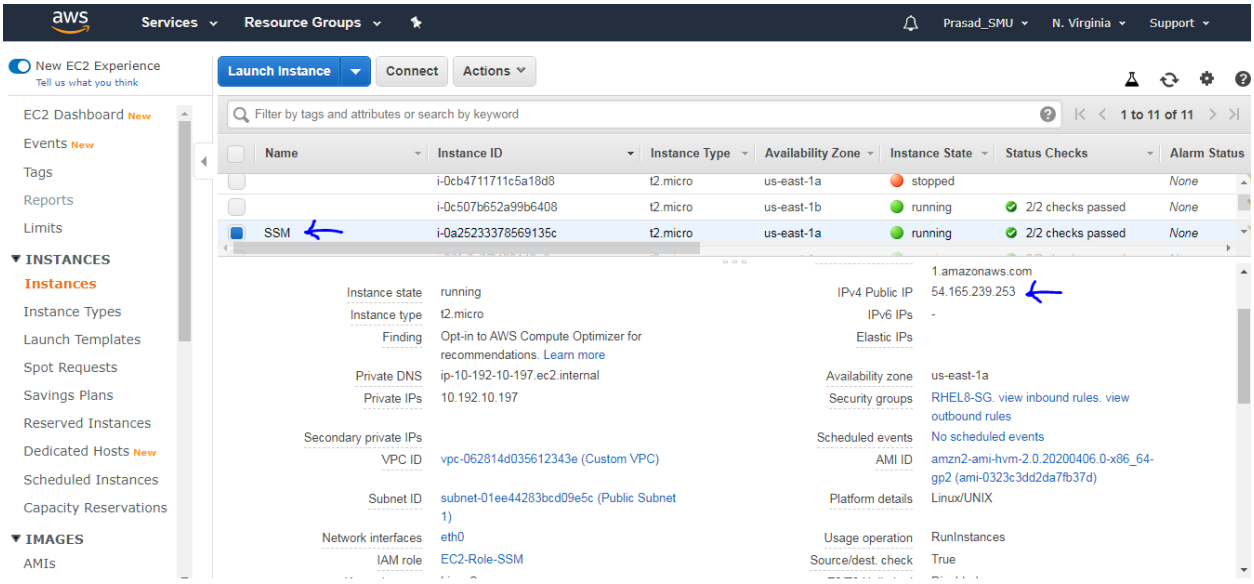
Click on Review and Launch.

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

Click on Launch Instances.



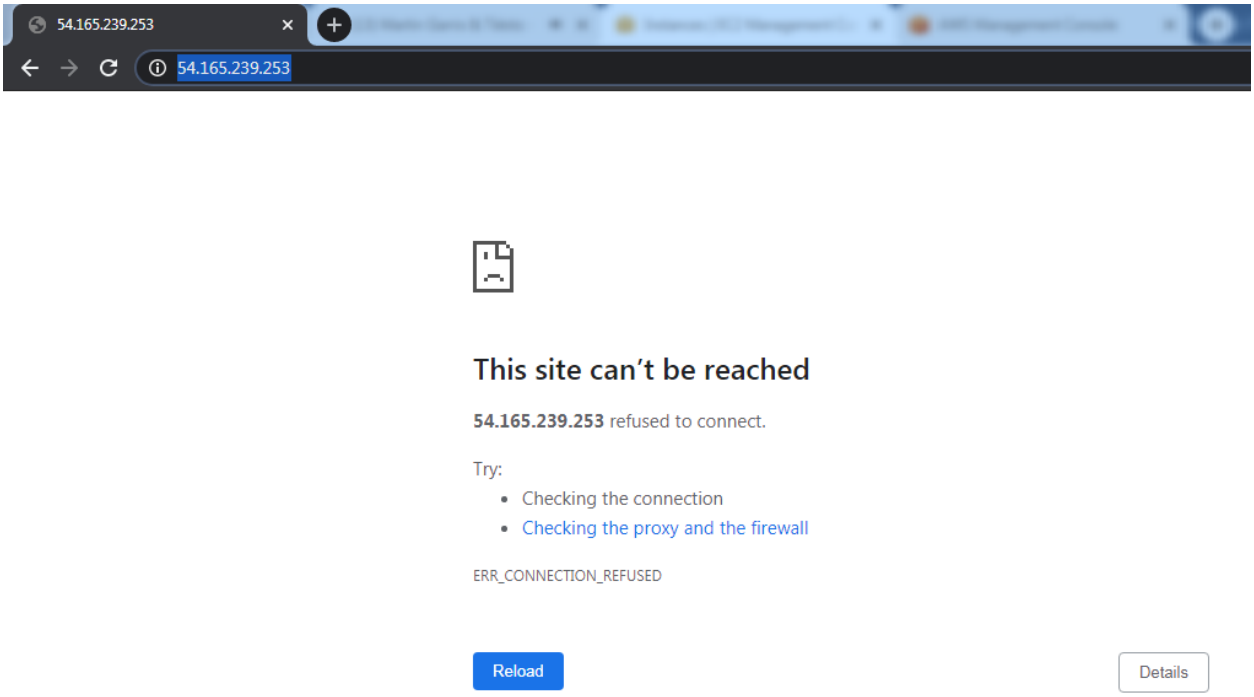
You can see that the Highlighted Instance has been launched Successfully!!!!!



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.

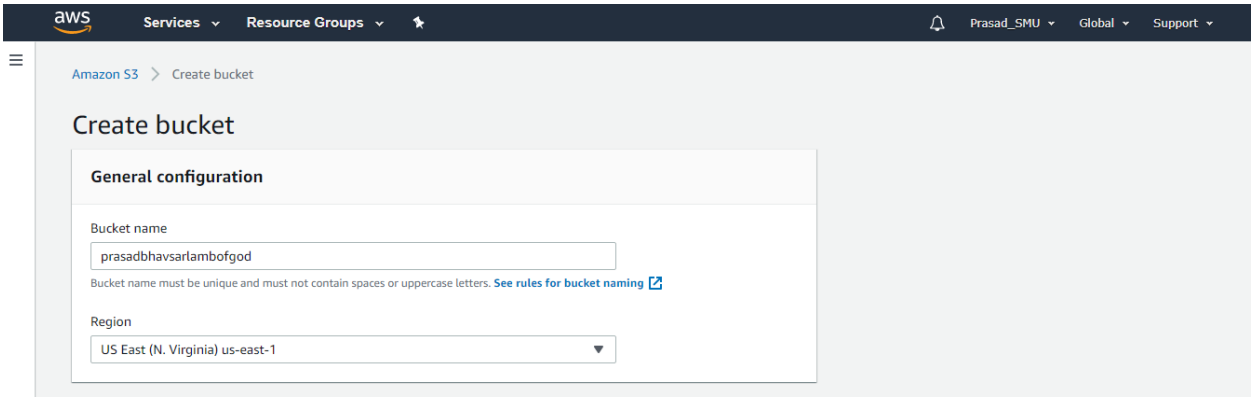


Task 3: Create a S3 Bucket to store SSM Logs

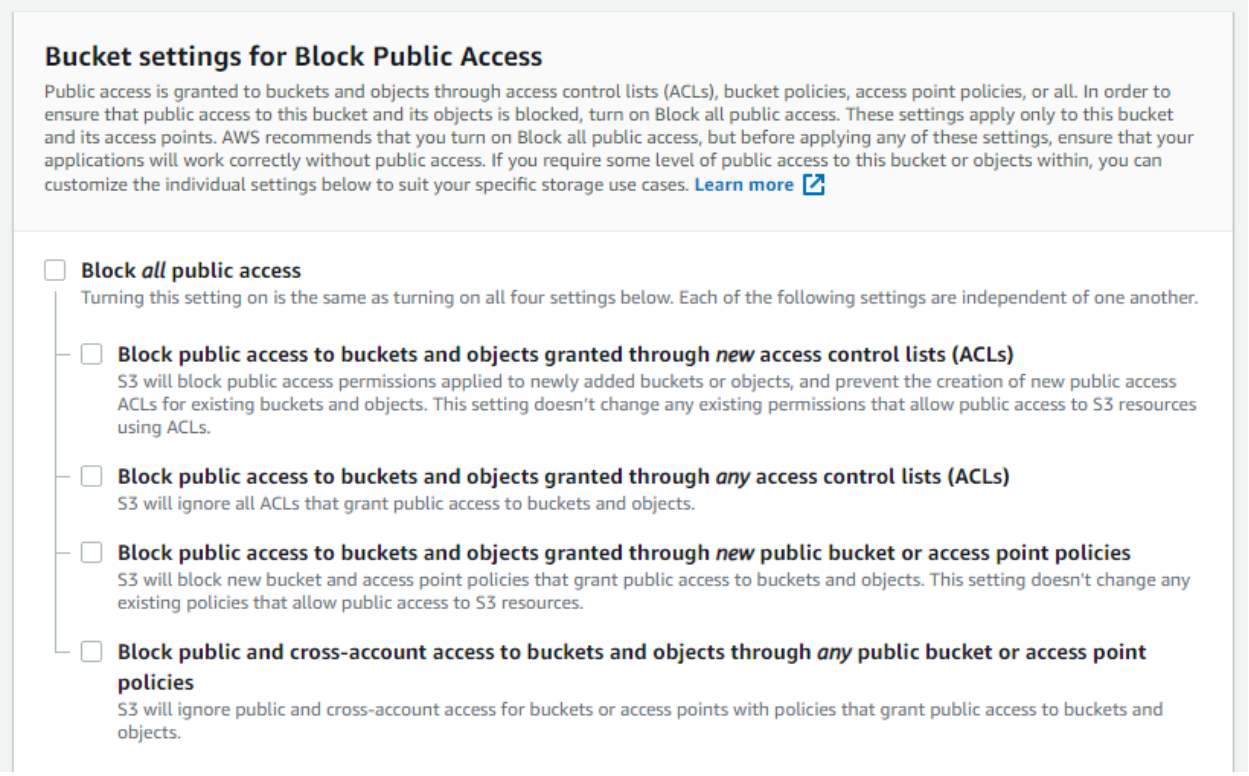
Navigate to S3 Service.

Click on Create Bucket.

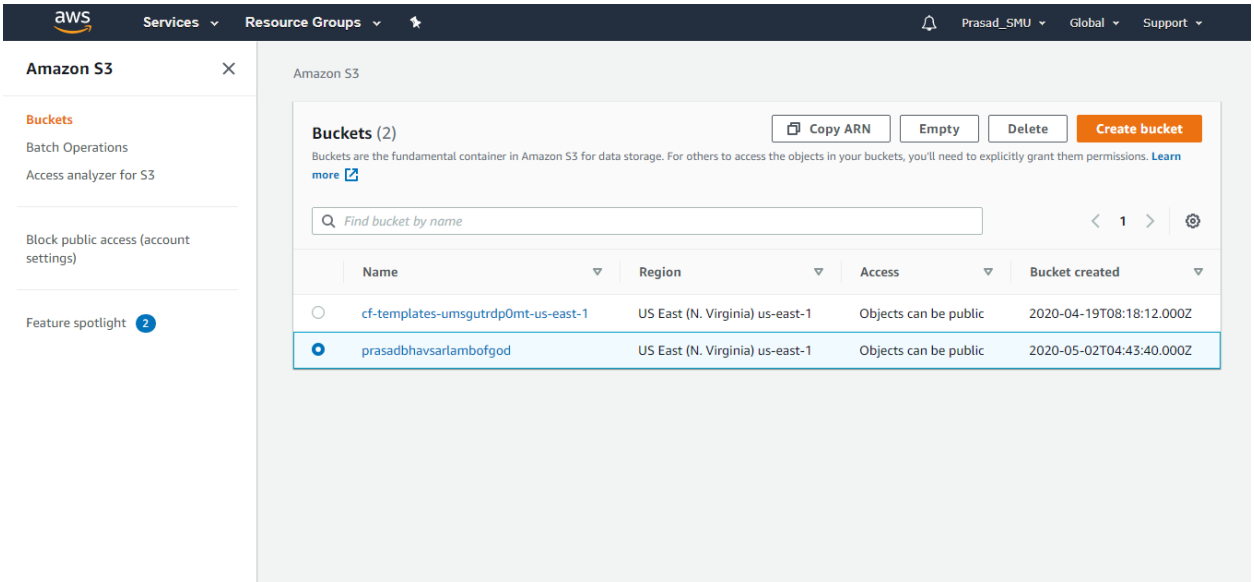
Give a unique Bucket Name as per your choice.



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

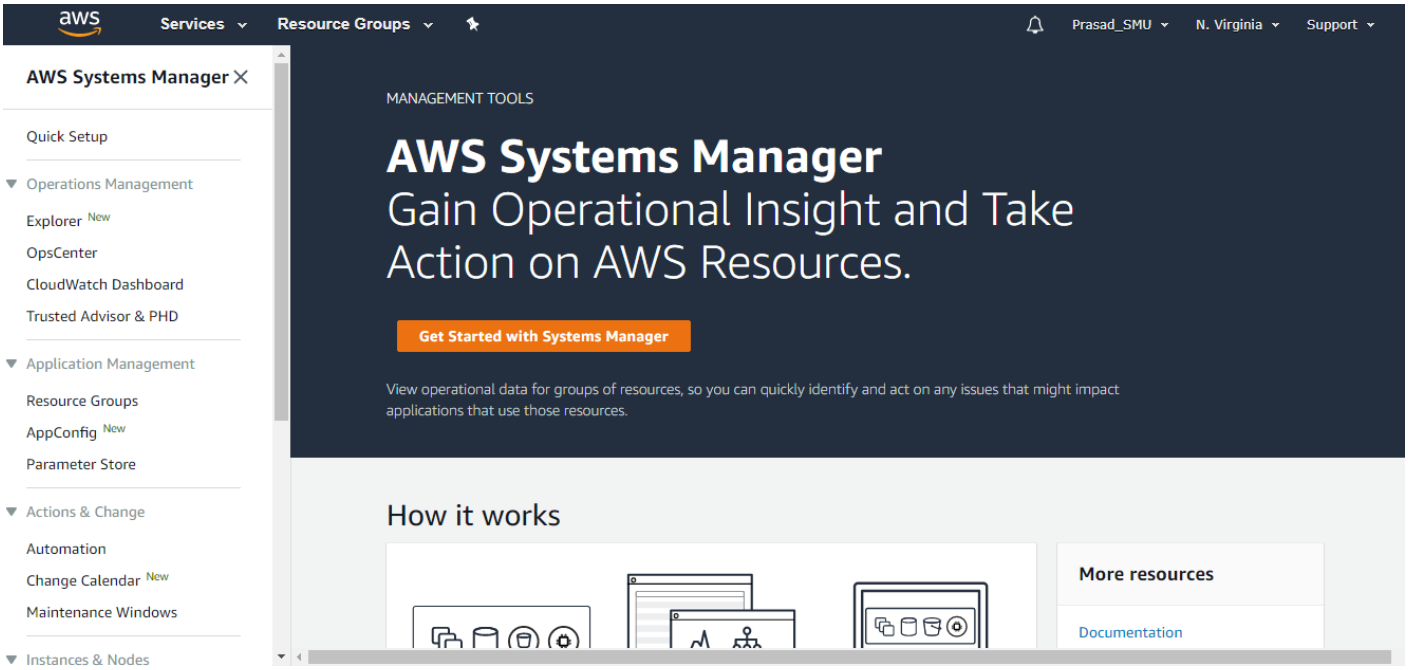


Click on Create. S3 Bucket has been successfully created to store Systems Manager (SSM) logs.

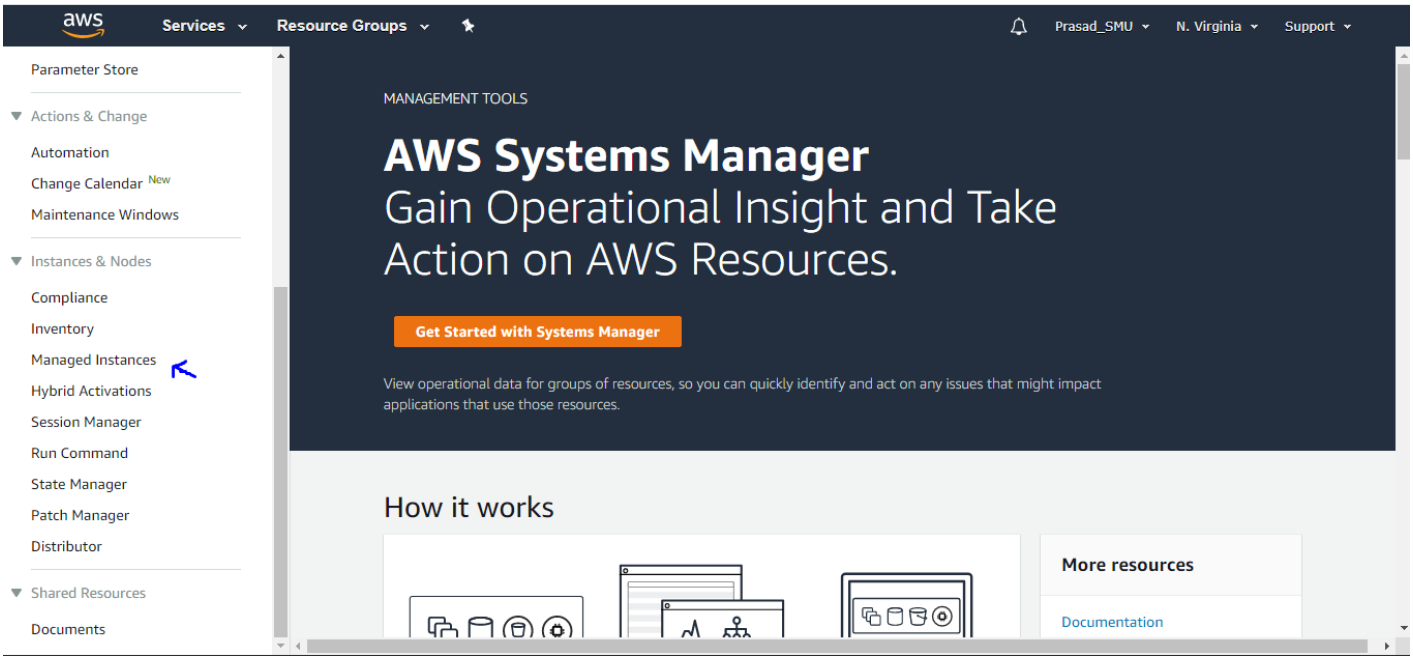


Task 4: AWS Systems Manager: Managed Instances

Navigate to AWS Systems Manager Service.

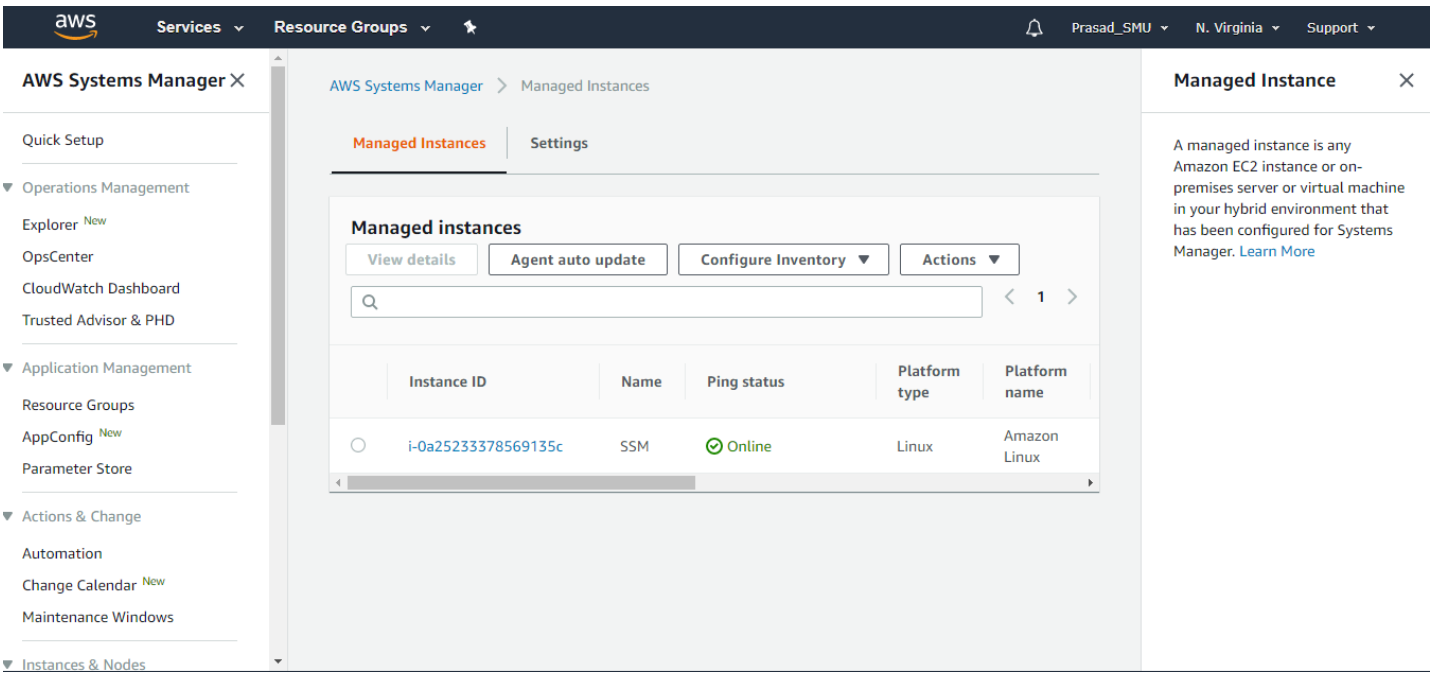


One the left-hand side, click on managed Instances.

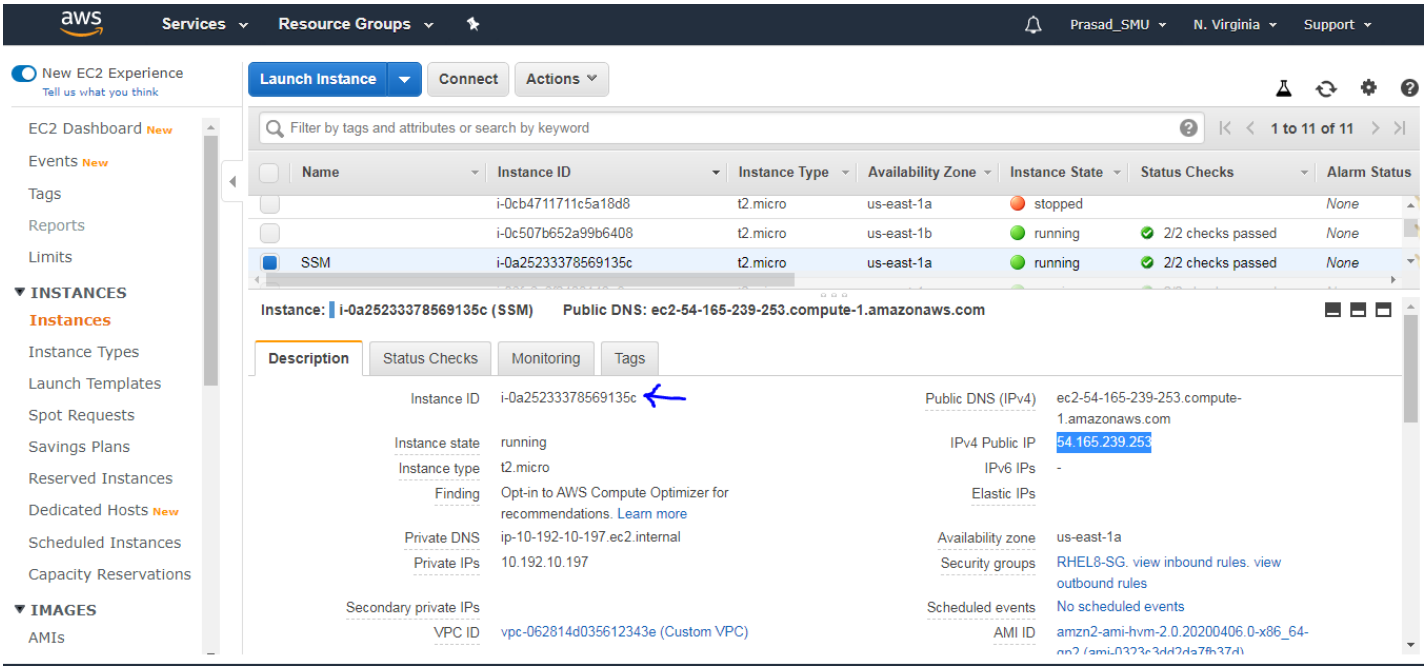


You should see Instance which we launched in Task 2.

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

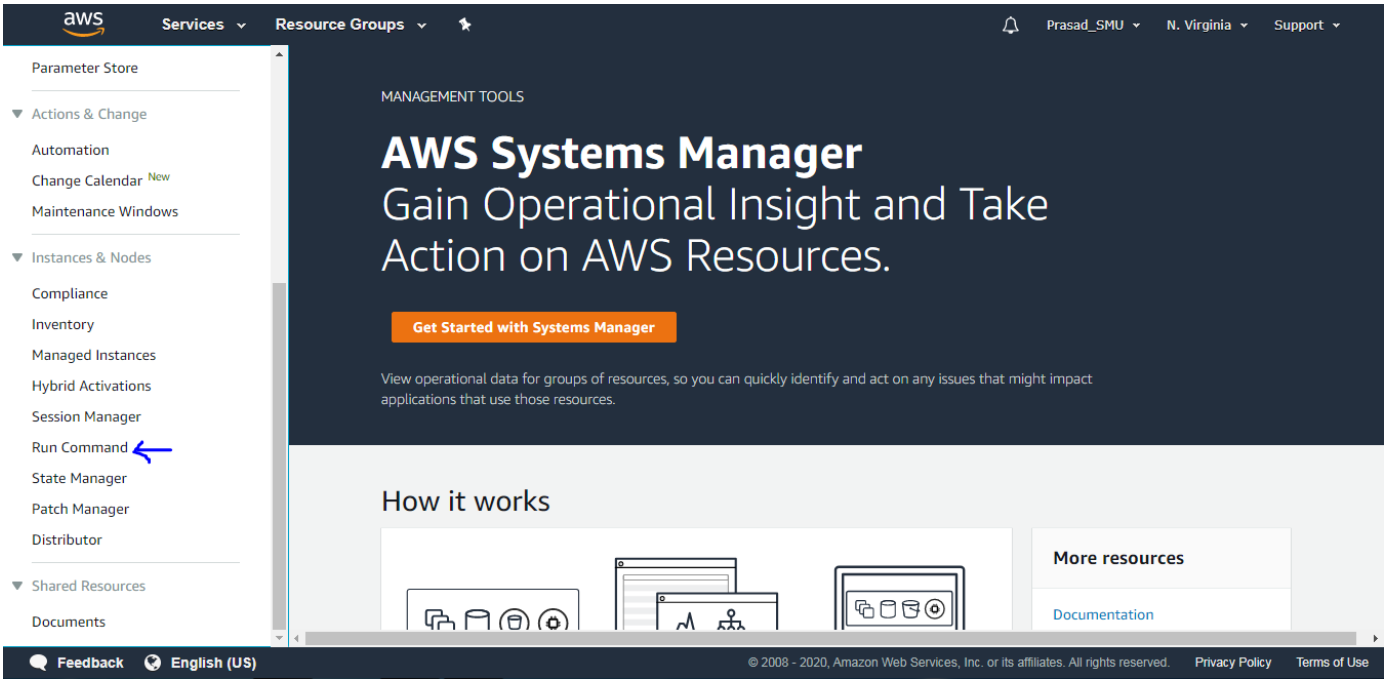


You can also verify the Instance IDs from EC2 Service Dashboard.

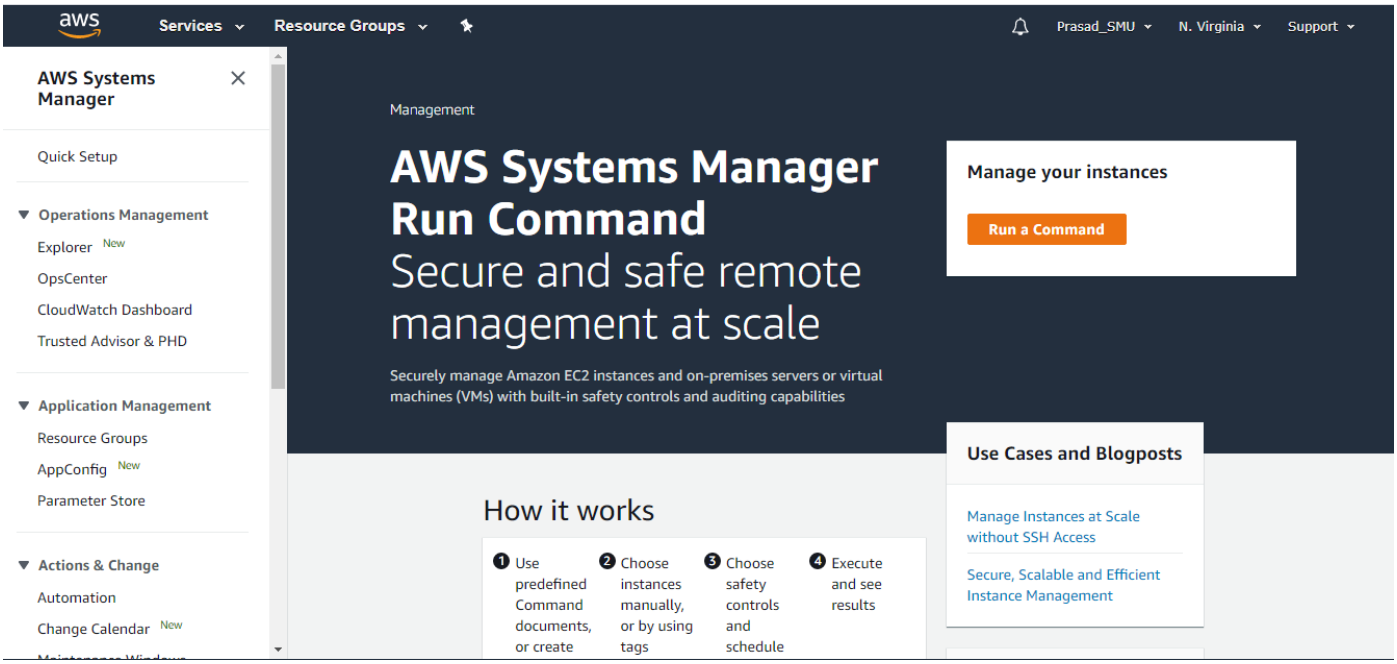


Task 5: AWS-Systems Manager: Run Command (Ansible Installation)

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

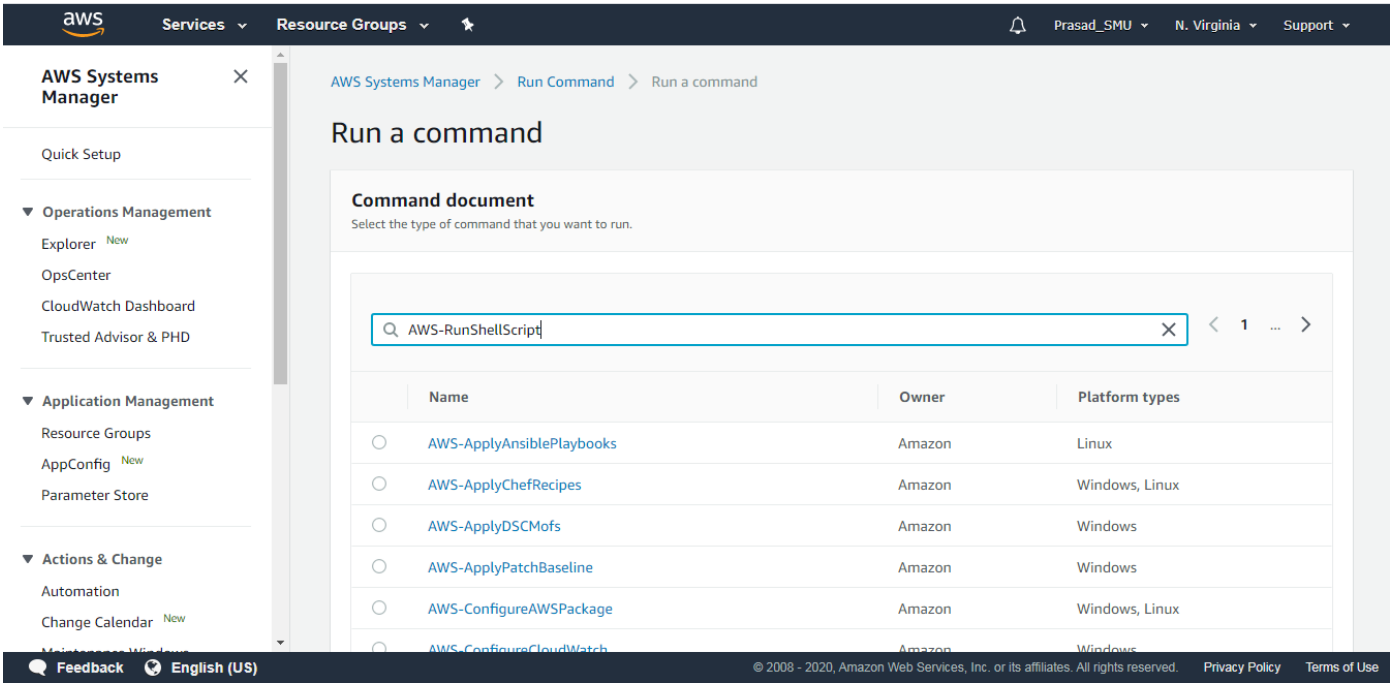


Click on Run a Command.

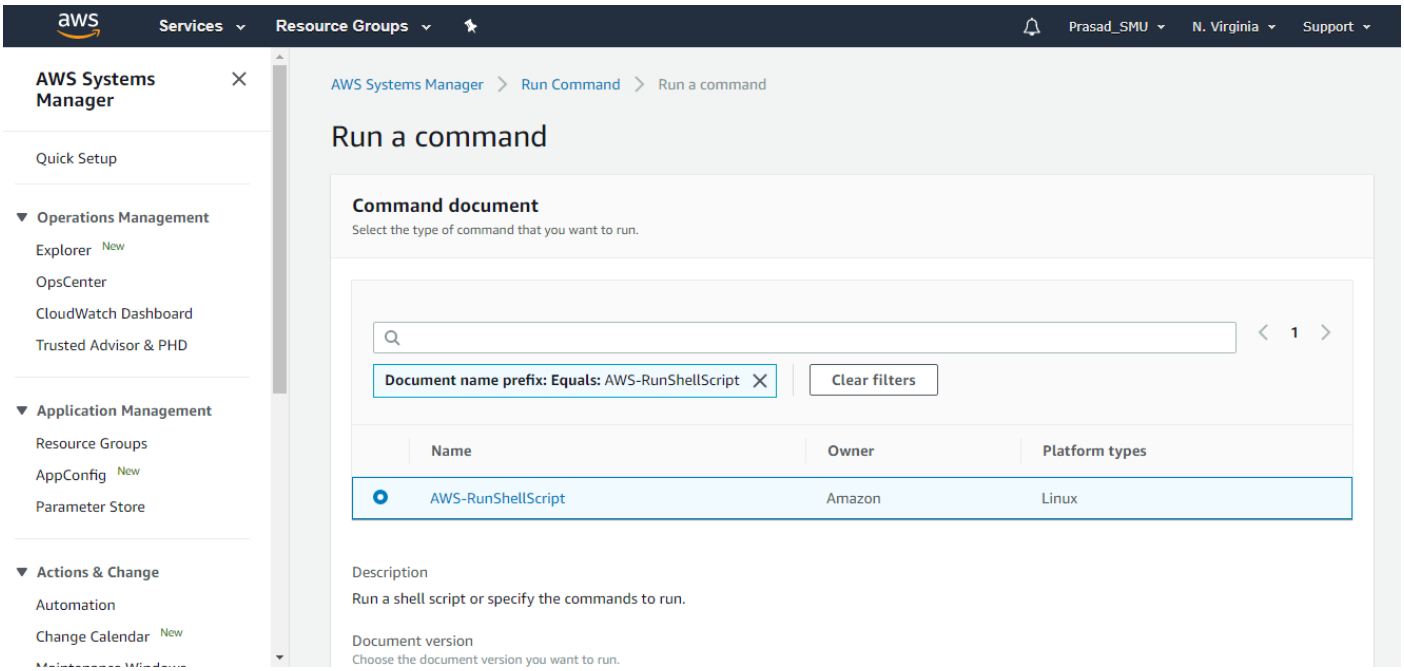


Under Command Document, search for the below AWS Managed Document.

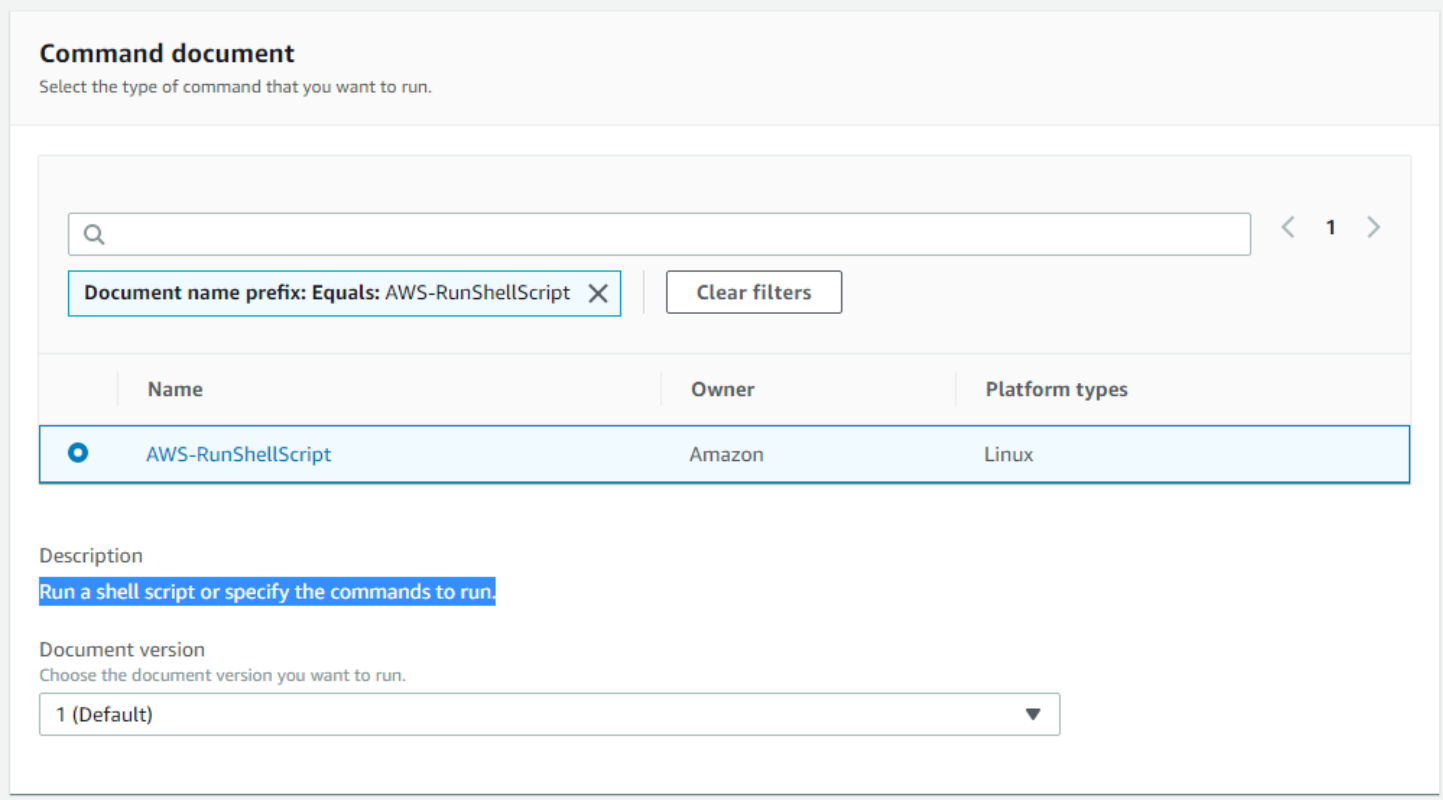
AWS-RunShellScript



Select the Command Document.

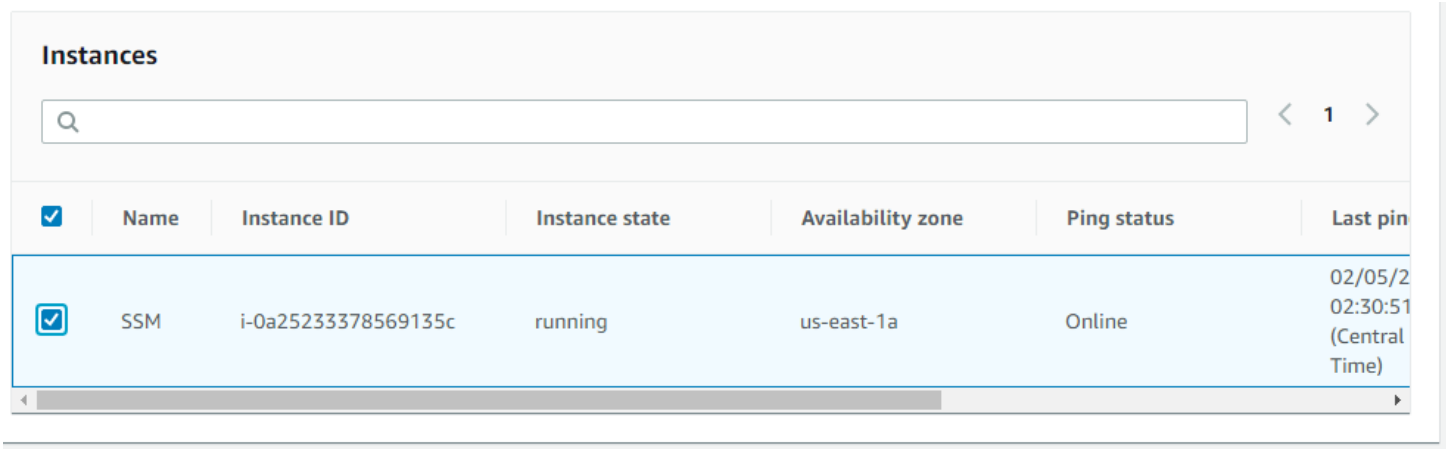


Read the highlighted Description.



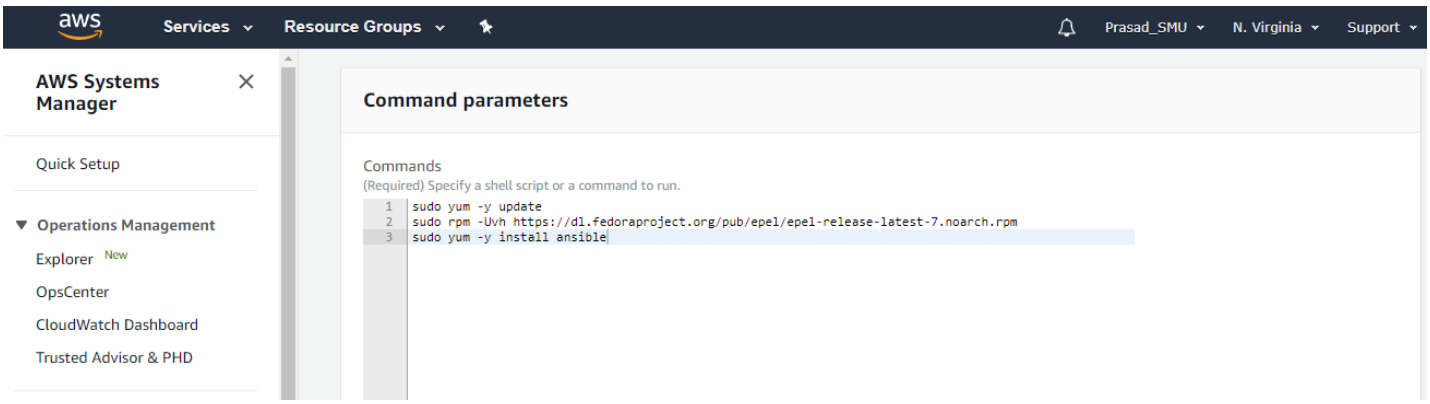
Under Targets, click on Choose Instances Manually and select both the EC2 Instances.

You can also select Instances using Tags.



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

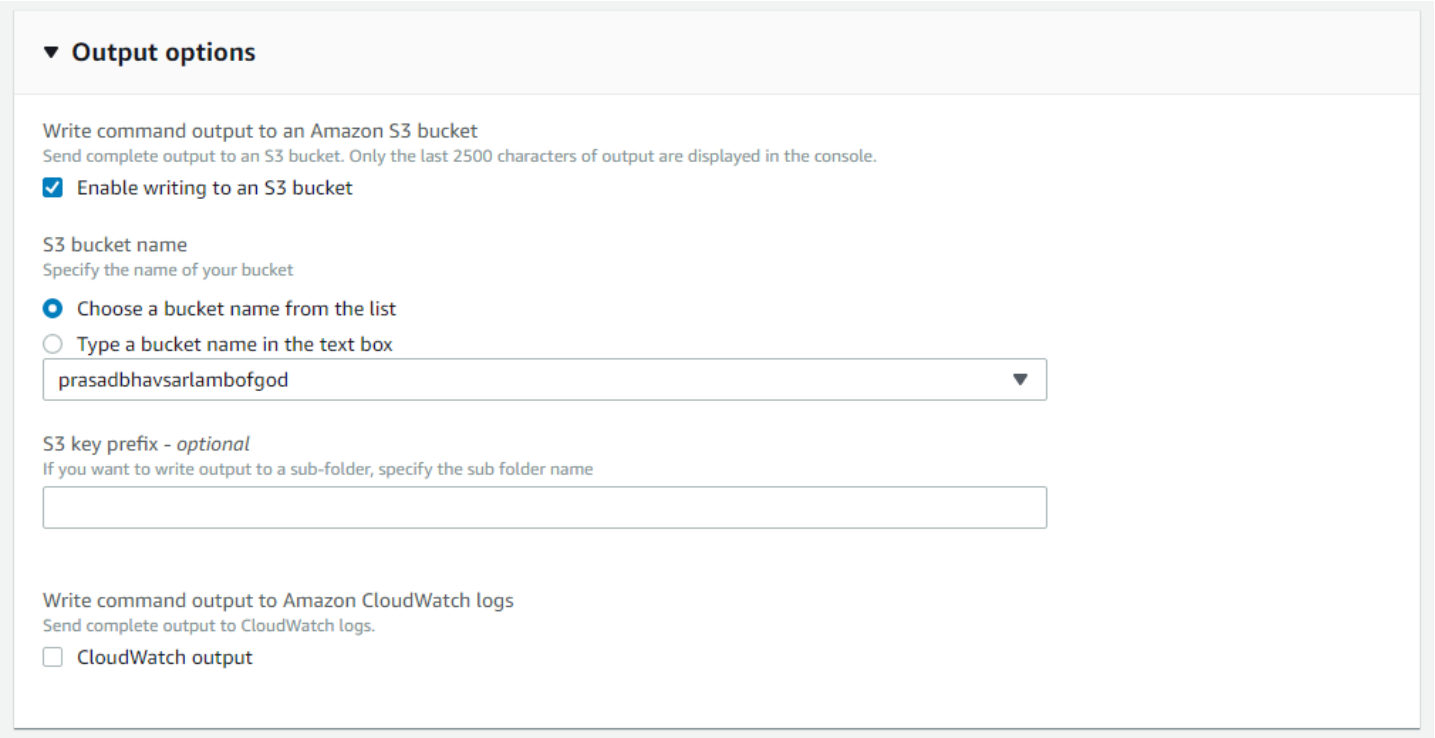
This script does the Ansible Installation on the Target Instance.



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

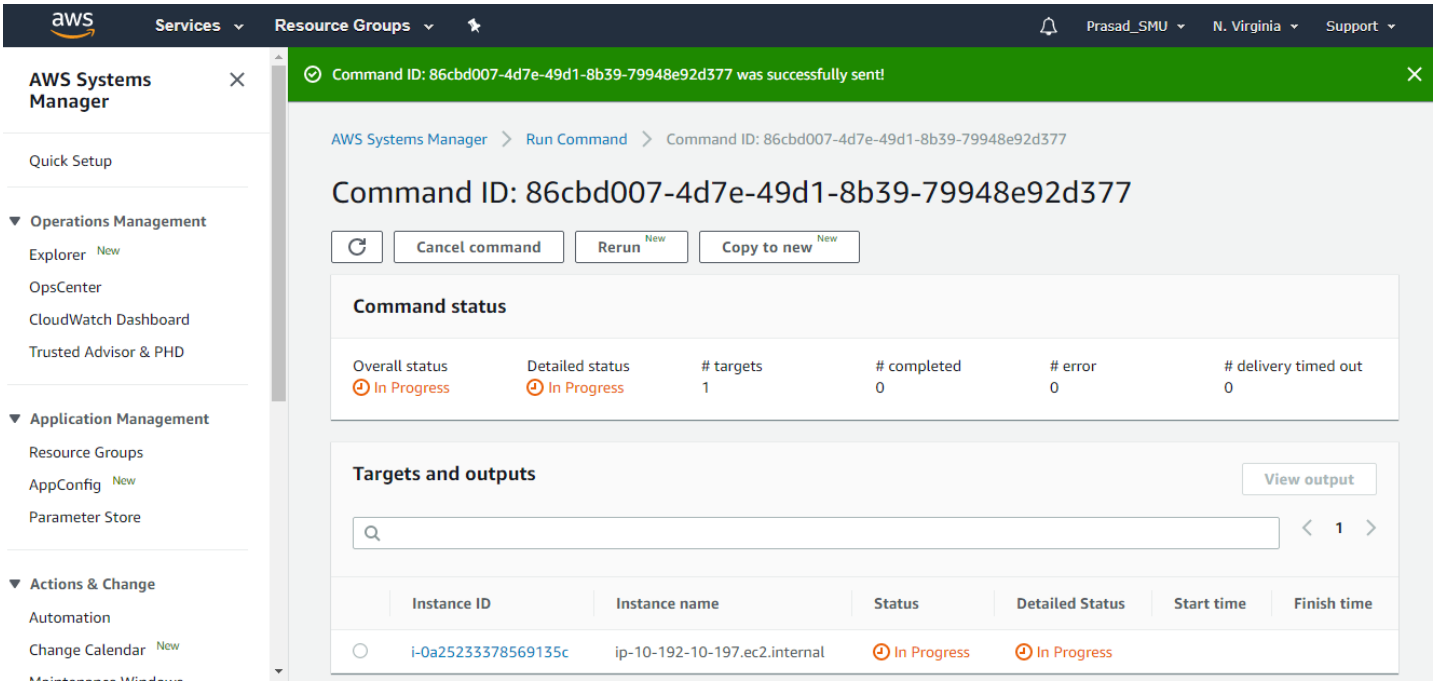
Logs in the S3 Bucket will be saved in stdout.txt and stderr.txt format.

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

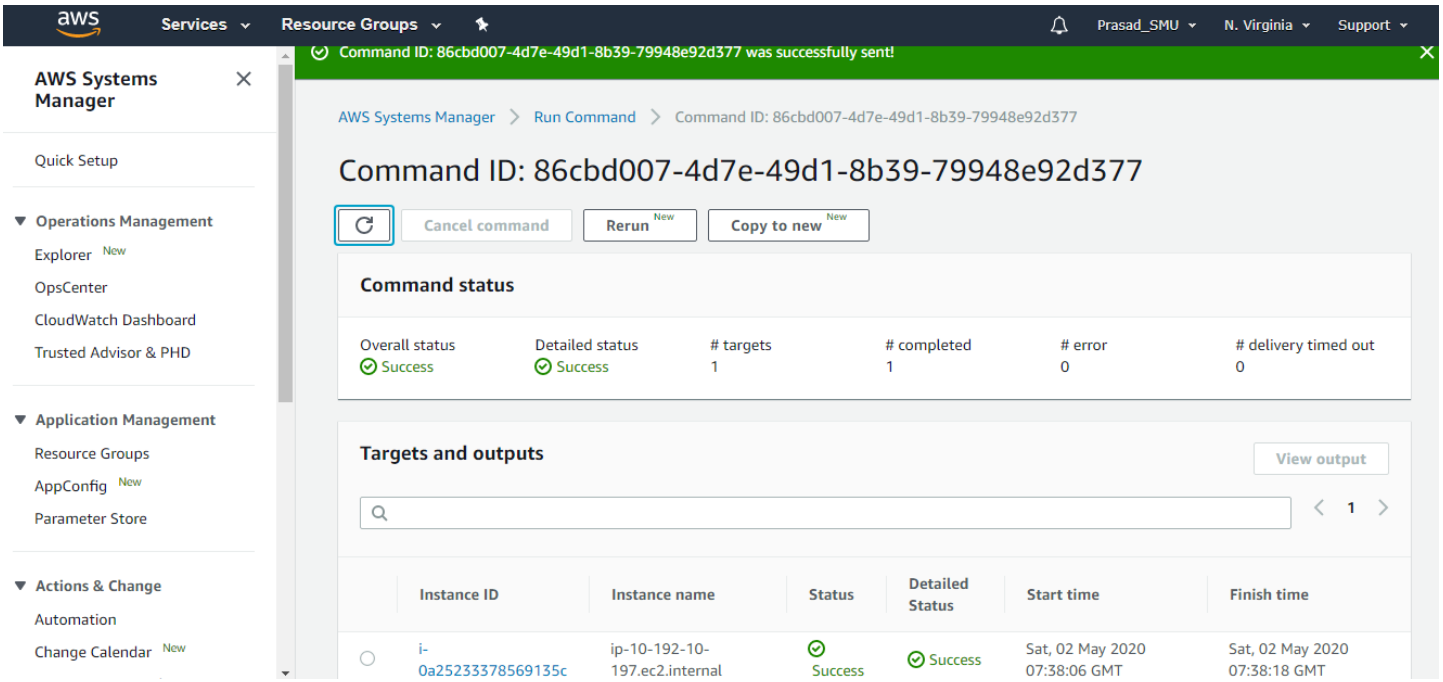


Task 6: Ansible Installation Check

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



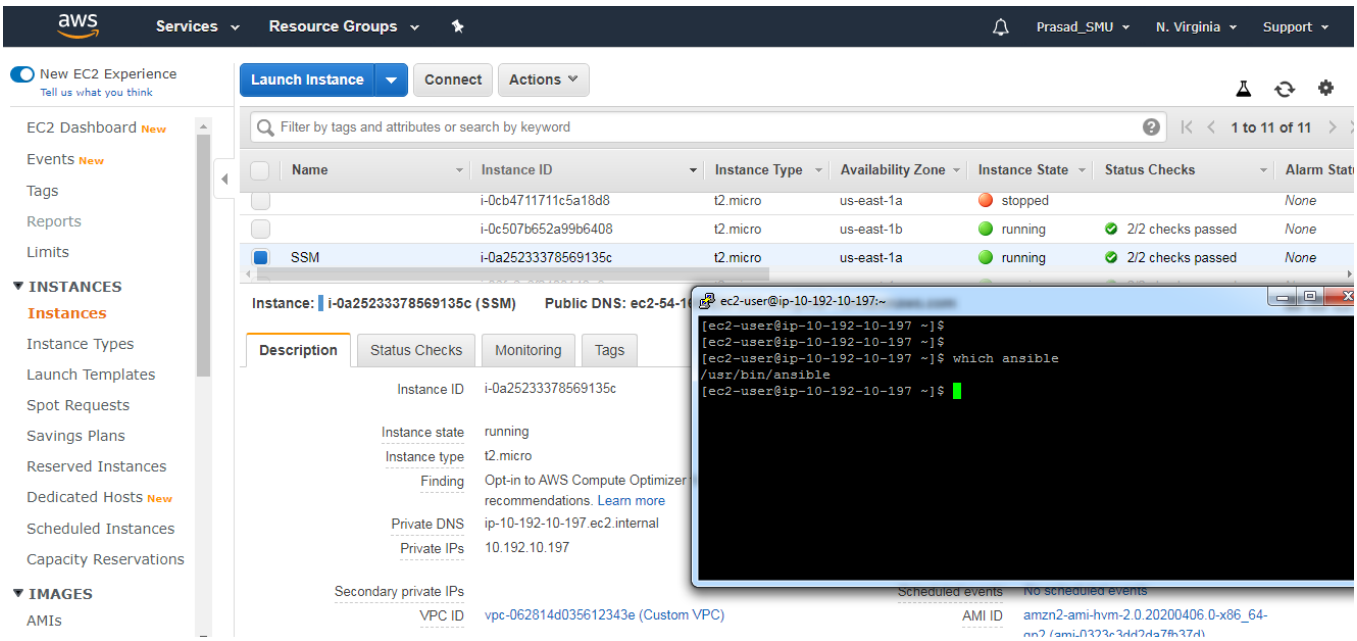
Status changes to **SUCCESS.**



Take SSH session of Target Instance.

Issue the below commands.

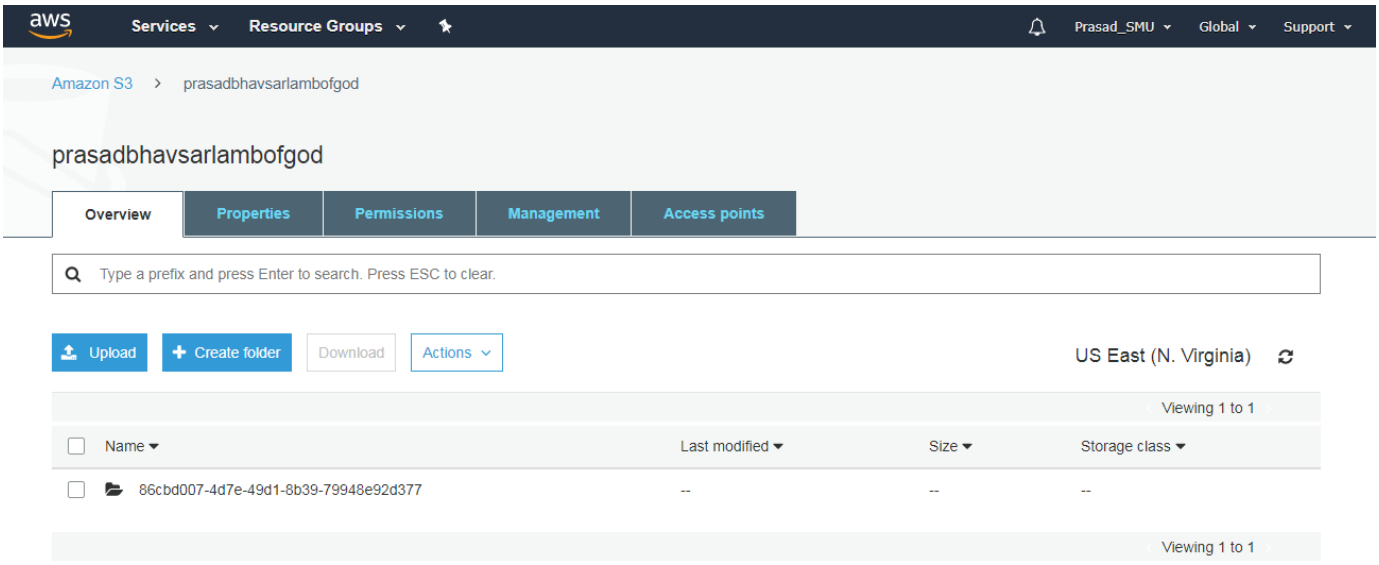
Command: which ansible



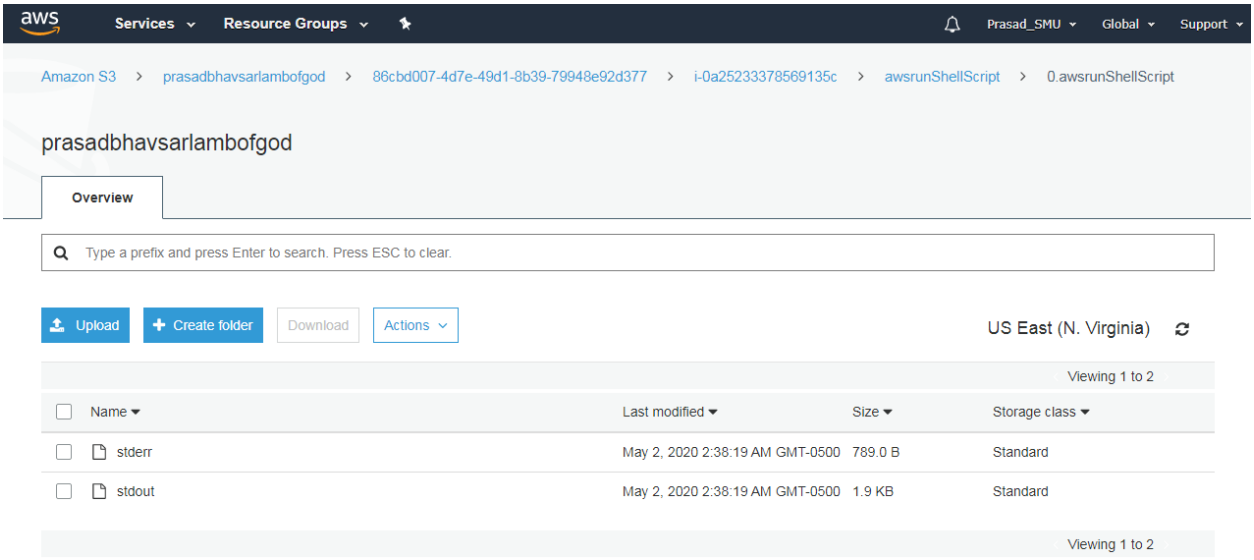
Ansible has been successfully installed on the Target Instance.

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

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

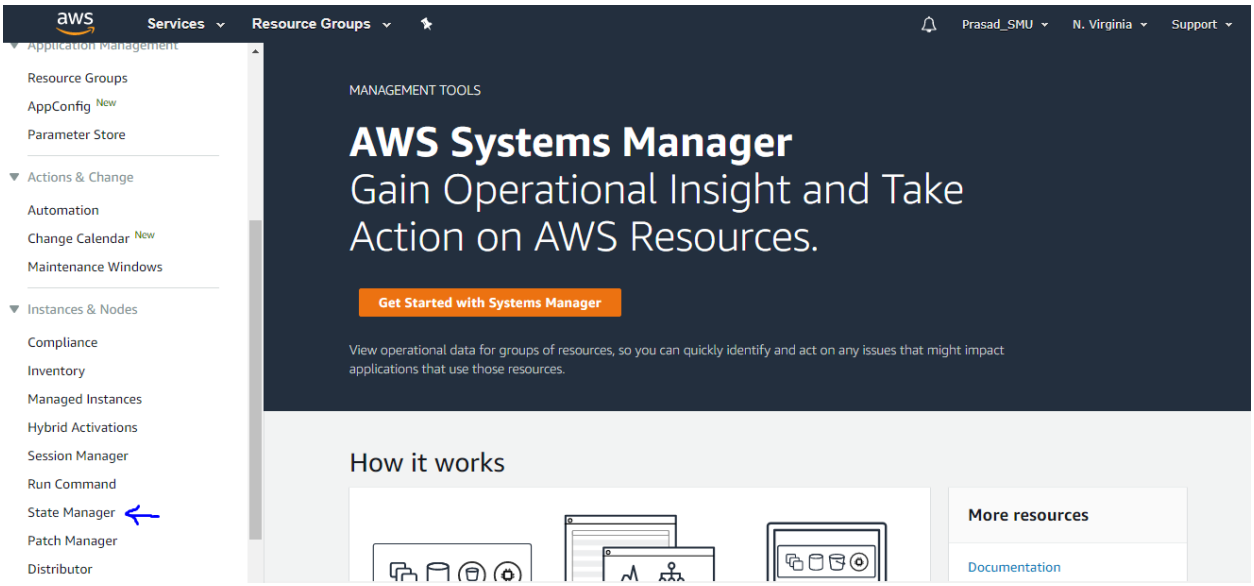


Download the stderr.txt and stdout.txt if you want to check the SSM Logs.

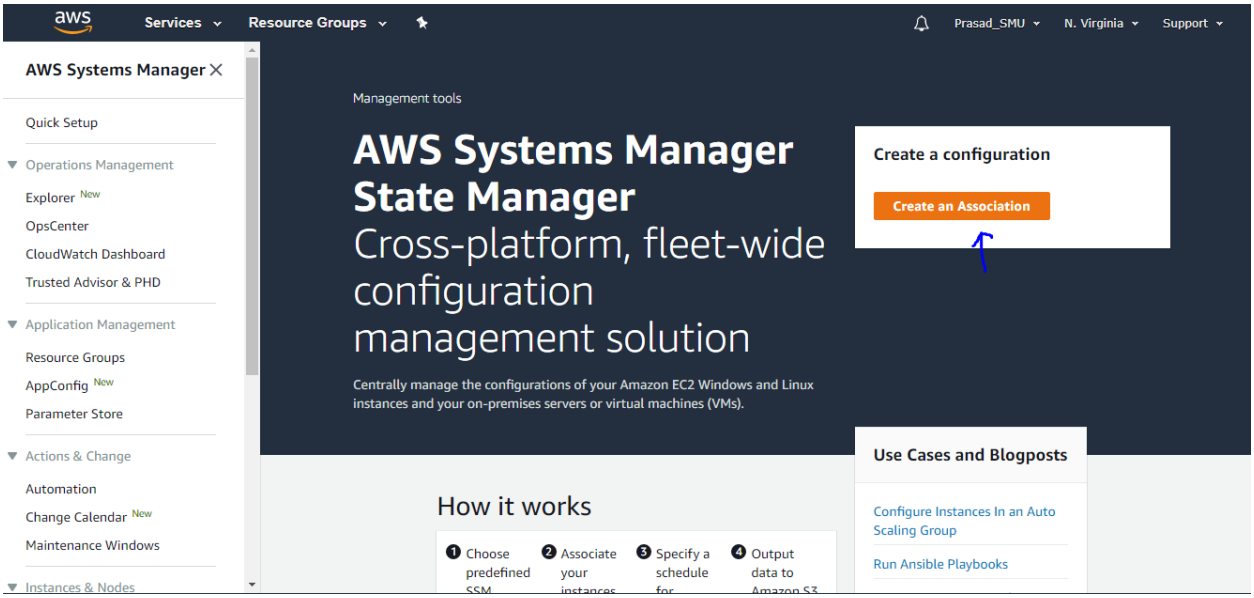


Task 7: Run an Ansible Playbook using AWS Systems Manager

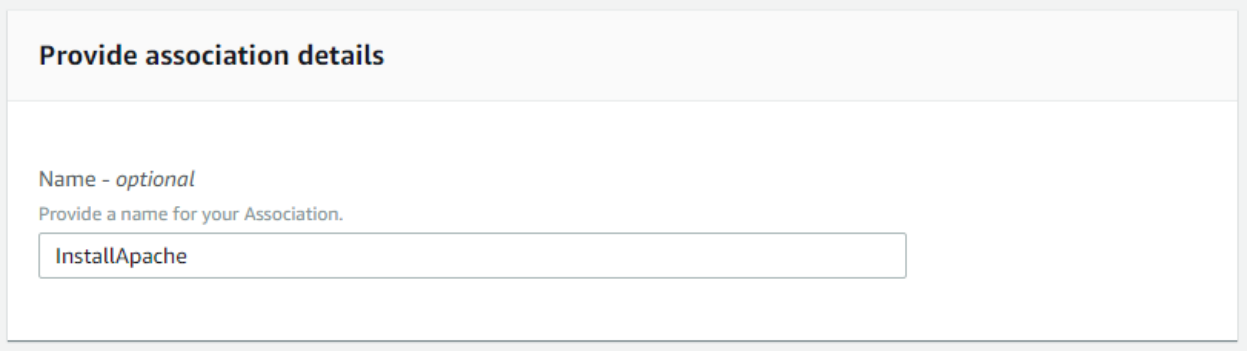
Navigate to AWS Systems Manager Service and on right-hand side click on **State Manager.**



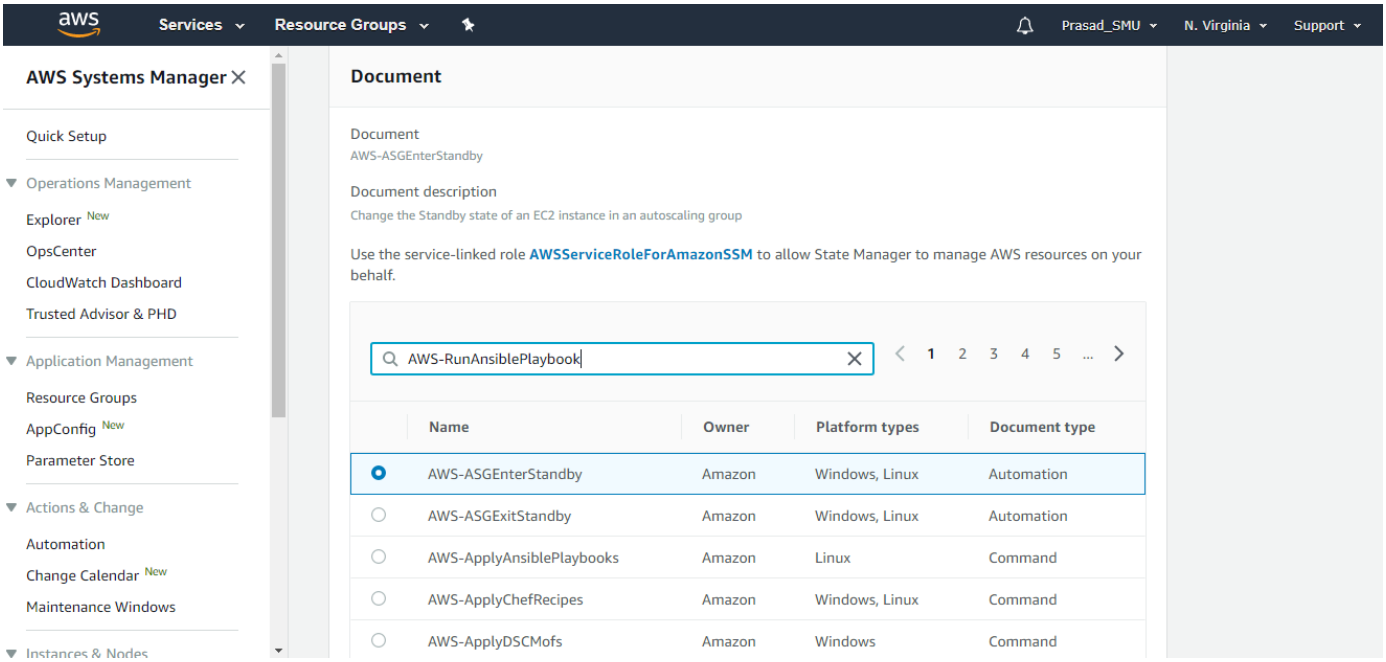
Now click on **Create Association**.



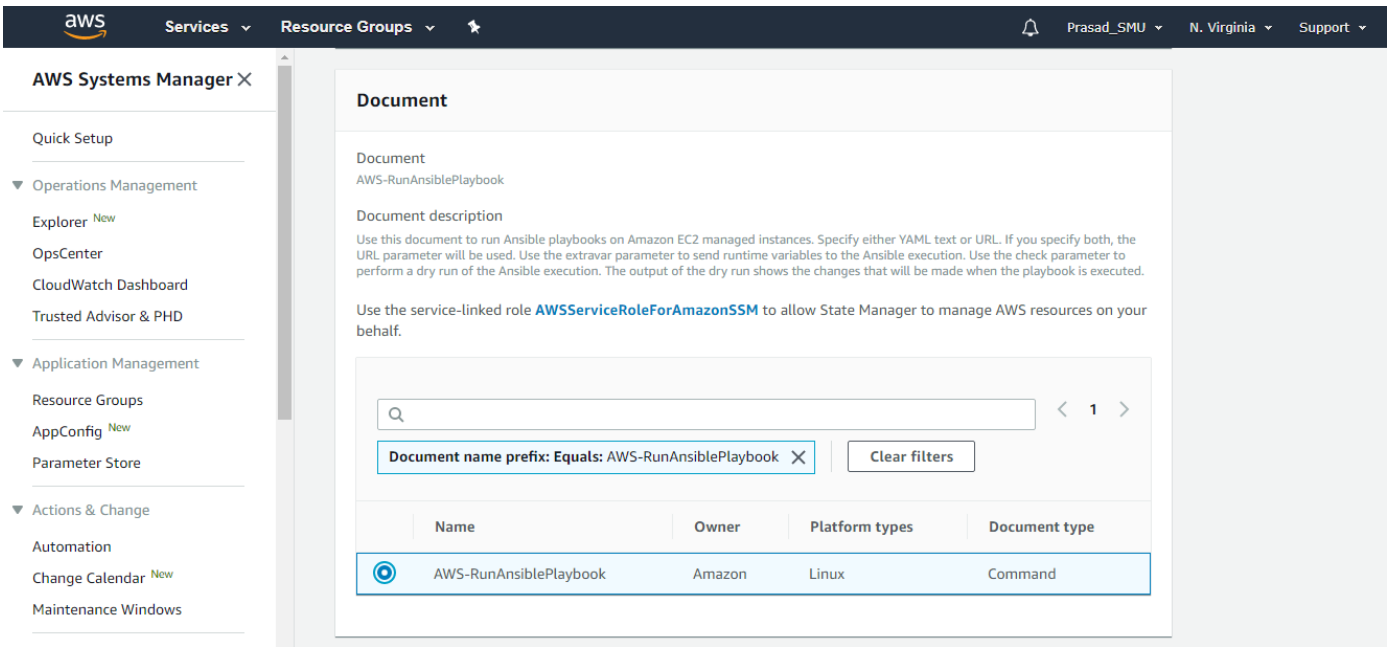
Specify the Association Name as per your choice.



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



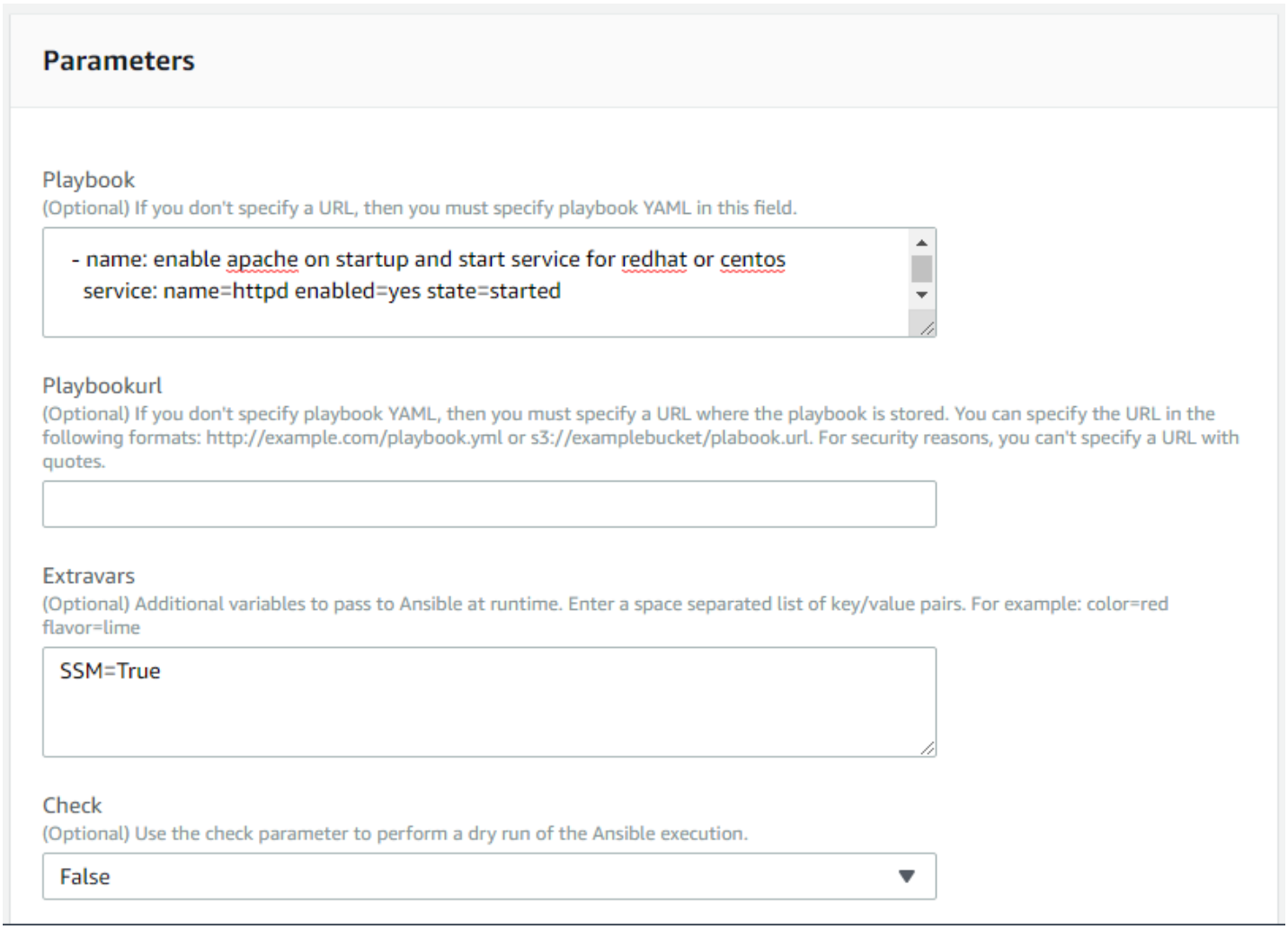
Select the AWS Managed Document **AWS-RunAnsiblePlaybook.**



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

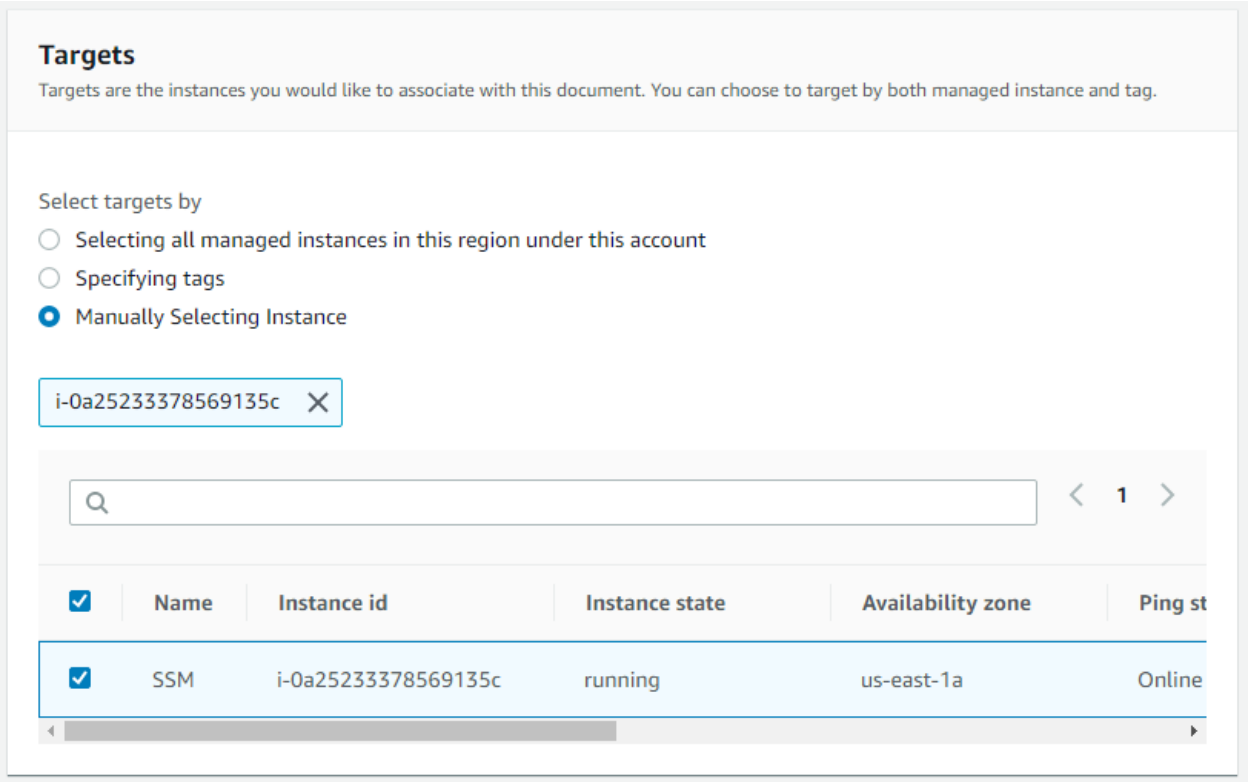
```
---
- name: linux_deploy_httpd
  hosts: all
  tasks:
  - name: Install HTTPD
    yum:
      name: "{{ item }}"
      state: latest
    loop:
     - httpd

  - 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.

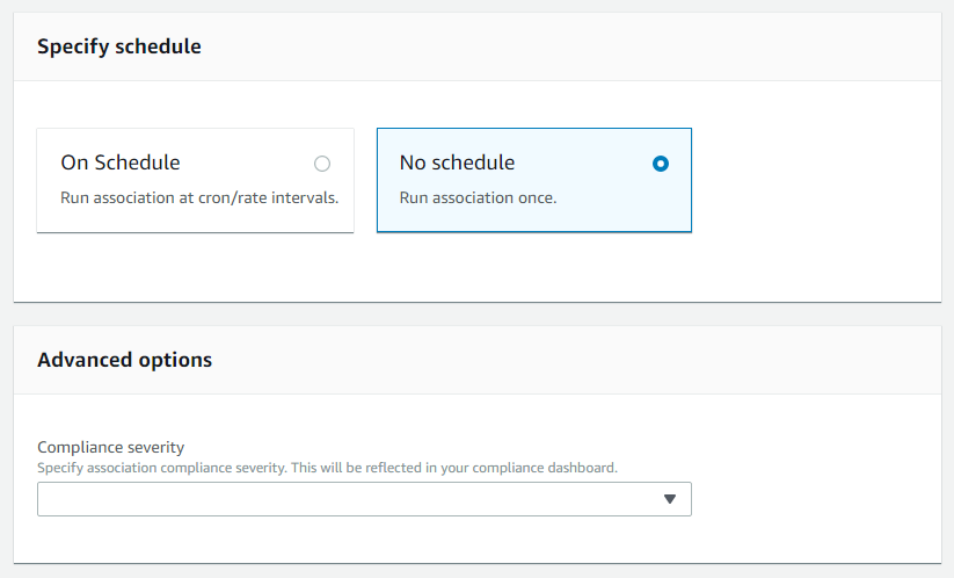


Under Targets, click on Choose Instances Manually and select both the EC2 Instances.

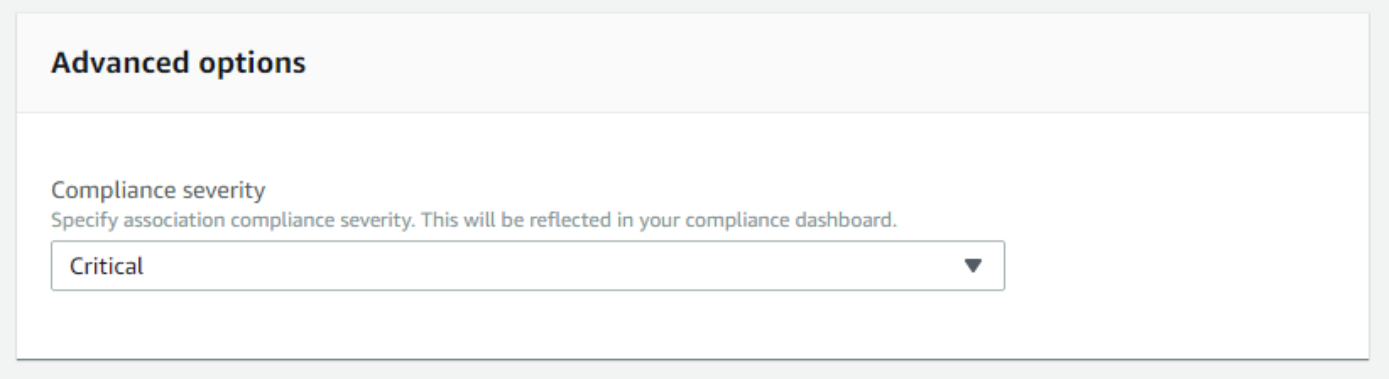
You can also select Instances using Tags.



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



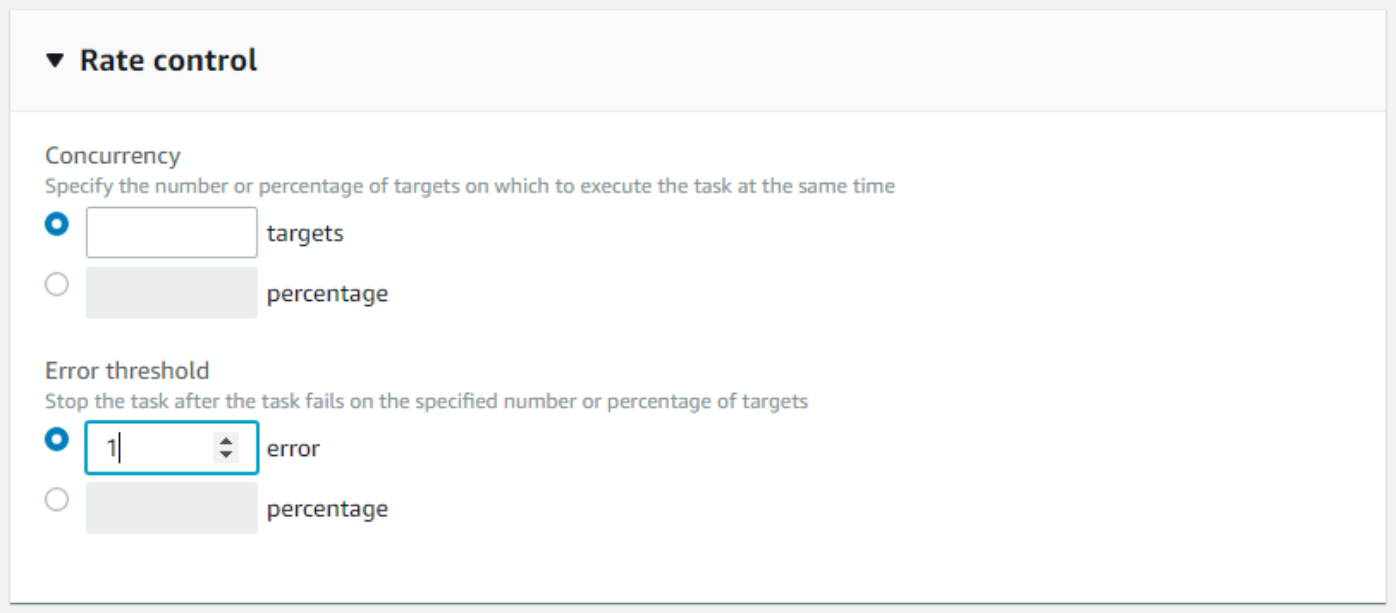
Keep the Compliance Severity CRITICAL.



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.

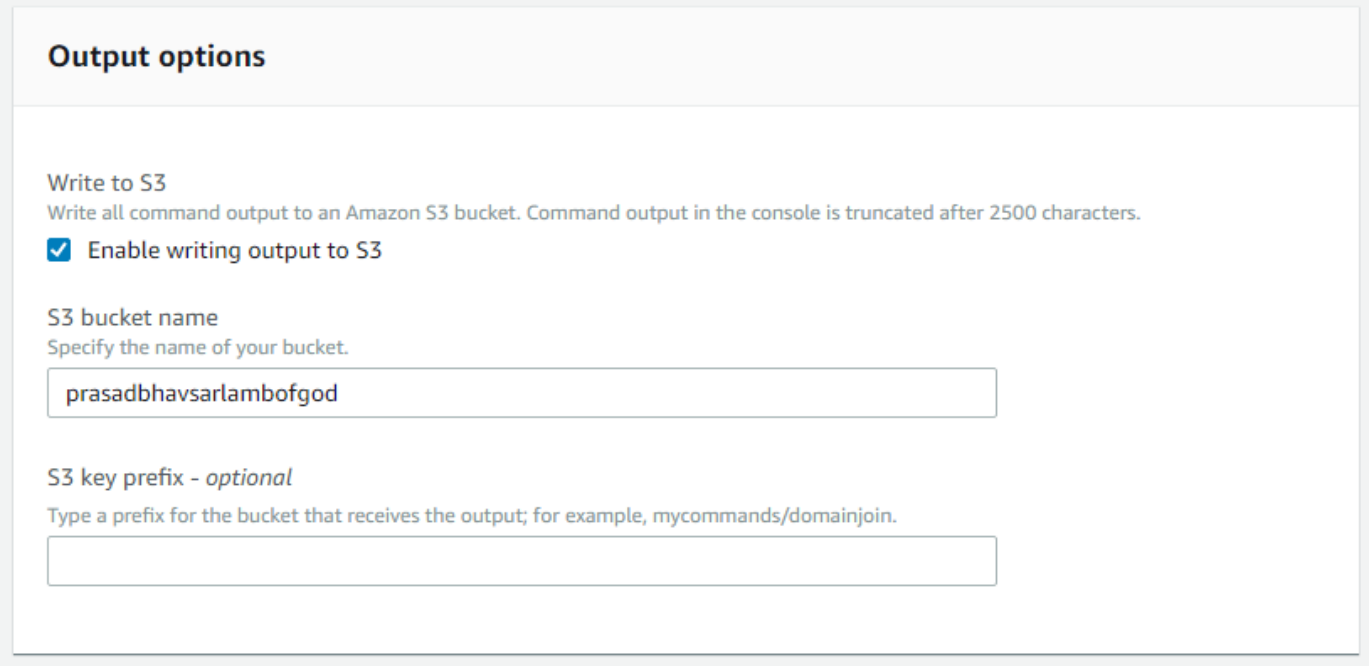
Specify the Error Threshold as One.



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

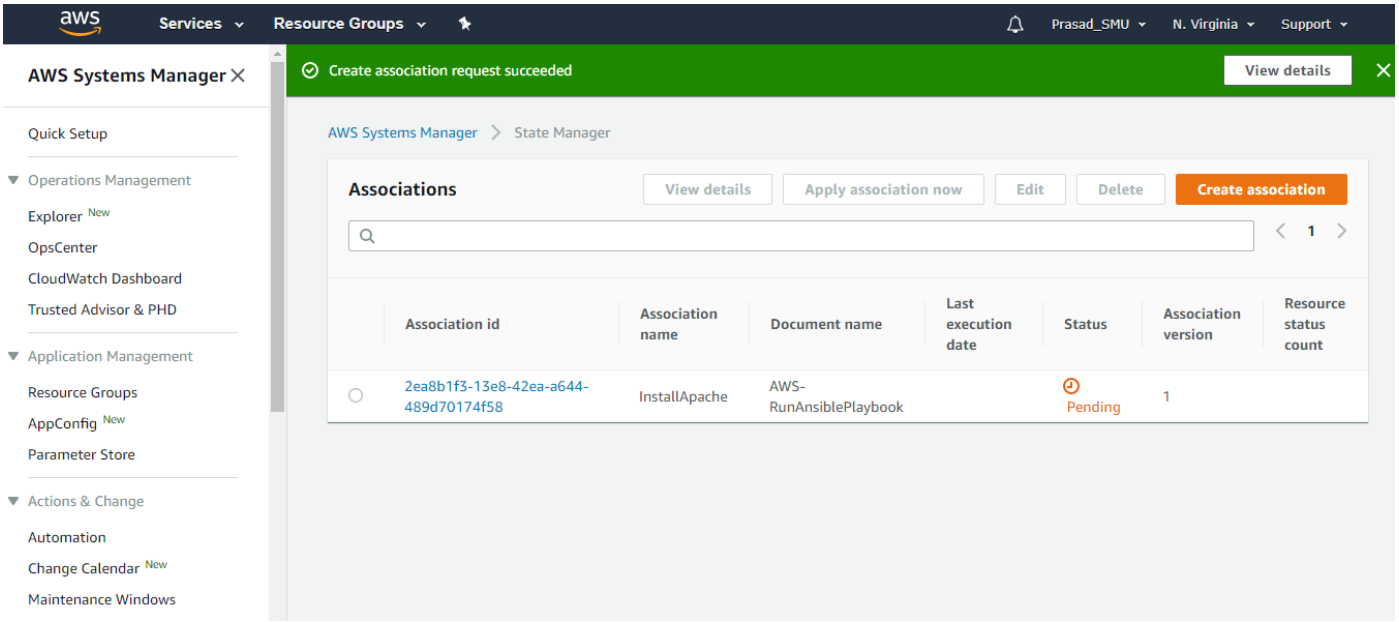
Logs in the S3 Bucket will be saved in stdout.txt and stderr.txt format.

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

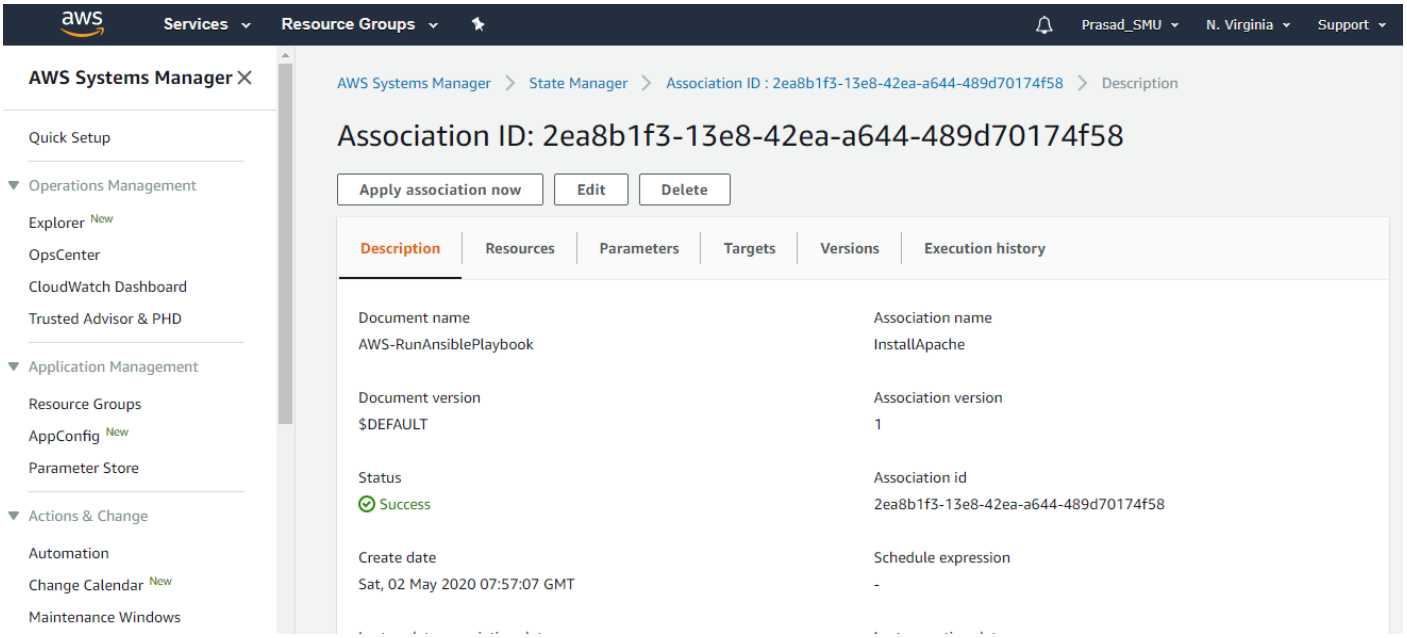


Task 8: Verify Ansible Playbook Execution

The association status is currently Pending.



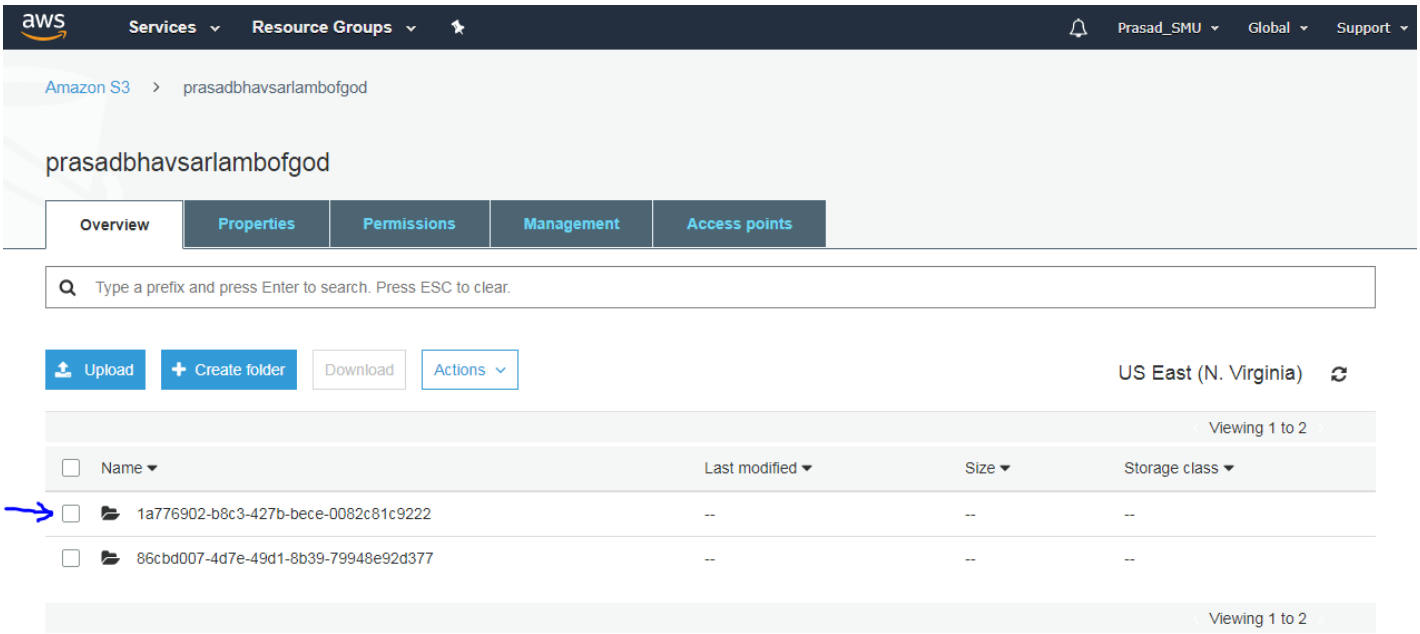
The association status is now Success.



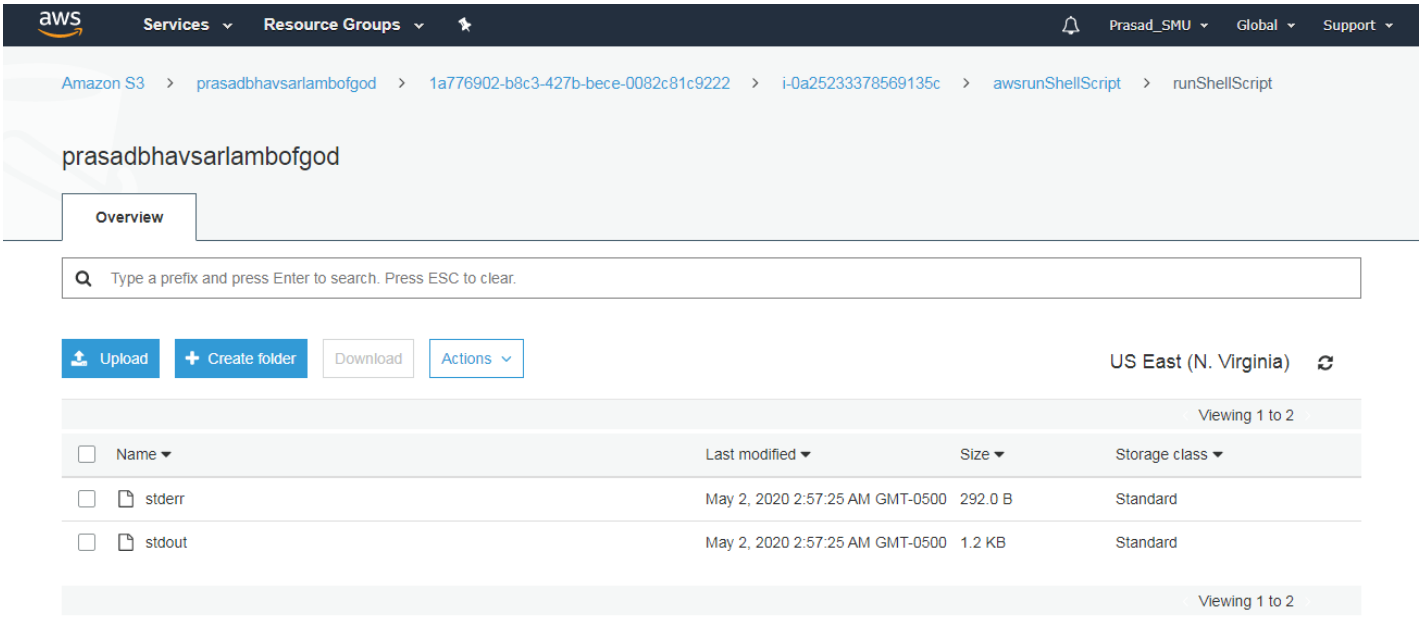
Scroll down on the same page and click on S3 Output.

Click on your S3 Bucket.

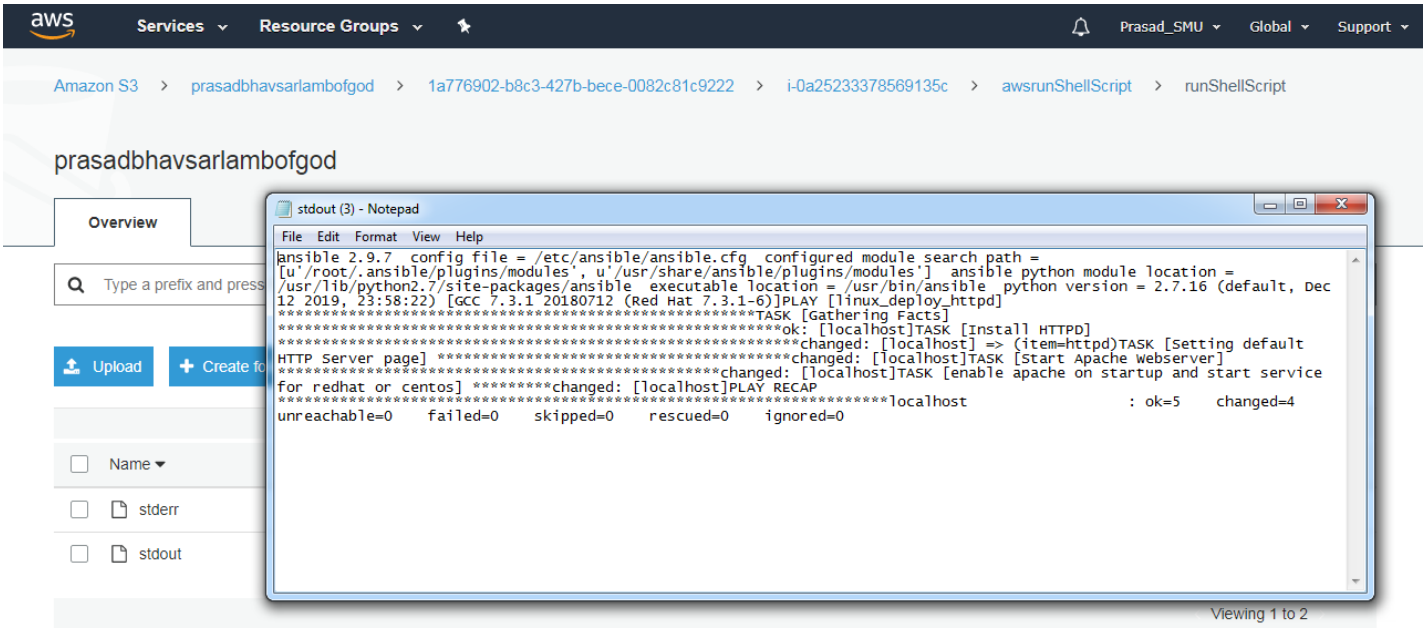
You'll notice that a new Object for SSM logs have been created.



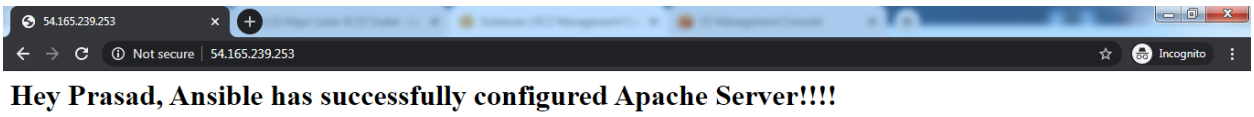
Click on the Objects and Sub-Objects till you see the stderr and stdout files.



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



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



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

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