Lunch EC2 Instance across Region by Running Ansible Playbook on AWS Systems Manager

In the previous lab, we've learnt how to deploy an EC2 Instance by running Ansible Playbook on Ansible Controller EC2 Instance. In this Lab, we are going to learn how to launch EC2 Instance in our environment by running Ansible Playbook on AWS Systems Manager. In this we are going to run Ansible Playbook using AWS Systems Manager in **N. Virginia (US-EAST-1)** region which will deploy an EC2 Instance in **Ireland (EU-WEST-1)** region.

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: Create an IAM User

Task 8: PIP, BOTO Configurations

Task 9: Run an Ansible Playbook using AWS Systems Manager

Task 10: Verify Ansible Playbook Execution

Task 1: Create IAM Role

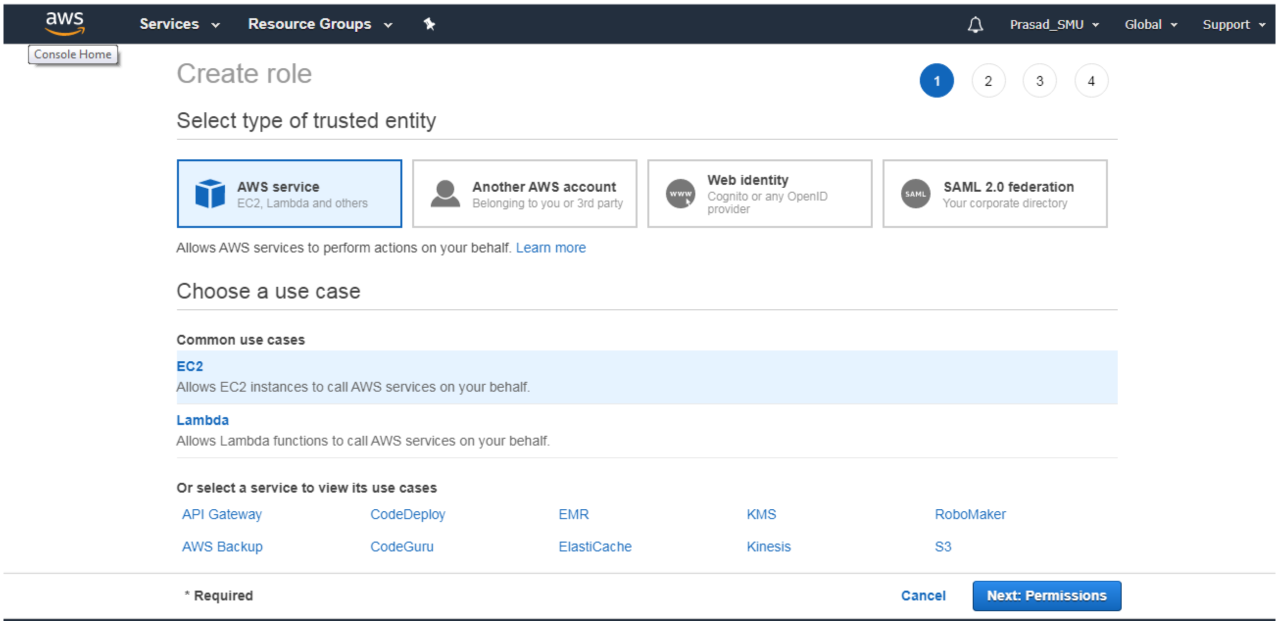
Login to the AWS Management Console.

Navigate to IAM Service and click on Roles.

Click on Create Role.

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

Click Next: Permissions.



Create role

Select type of trusted entity

Allows AWS services to perform actions on your behalf. Learn more

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

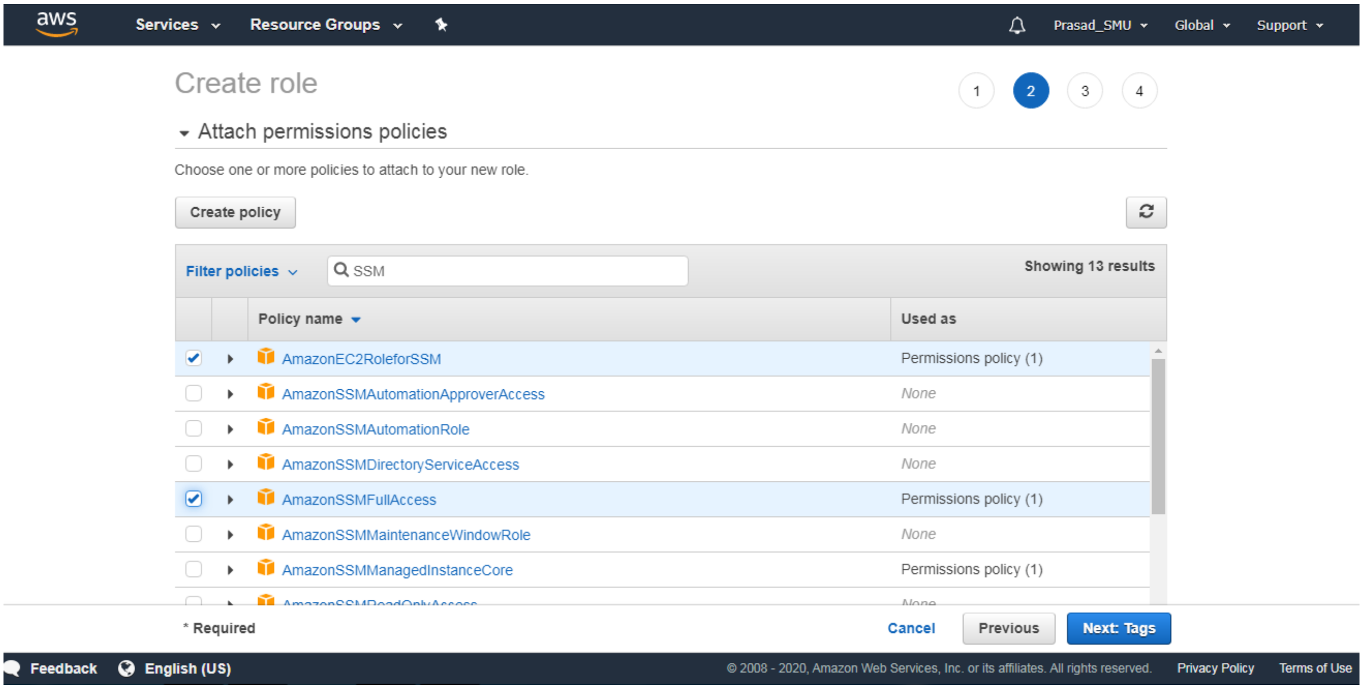
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

* Required

Select the below three Default IAM Policies:

1. **AmazonEC2RoleforSSM**
2. **AmazonSSMFullAccess**
3. **AmazonEC2FullAccess**



Create role

Attach permissions policies

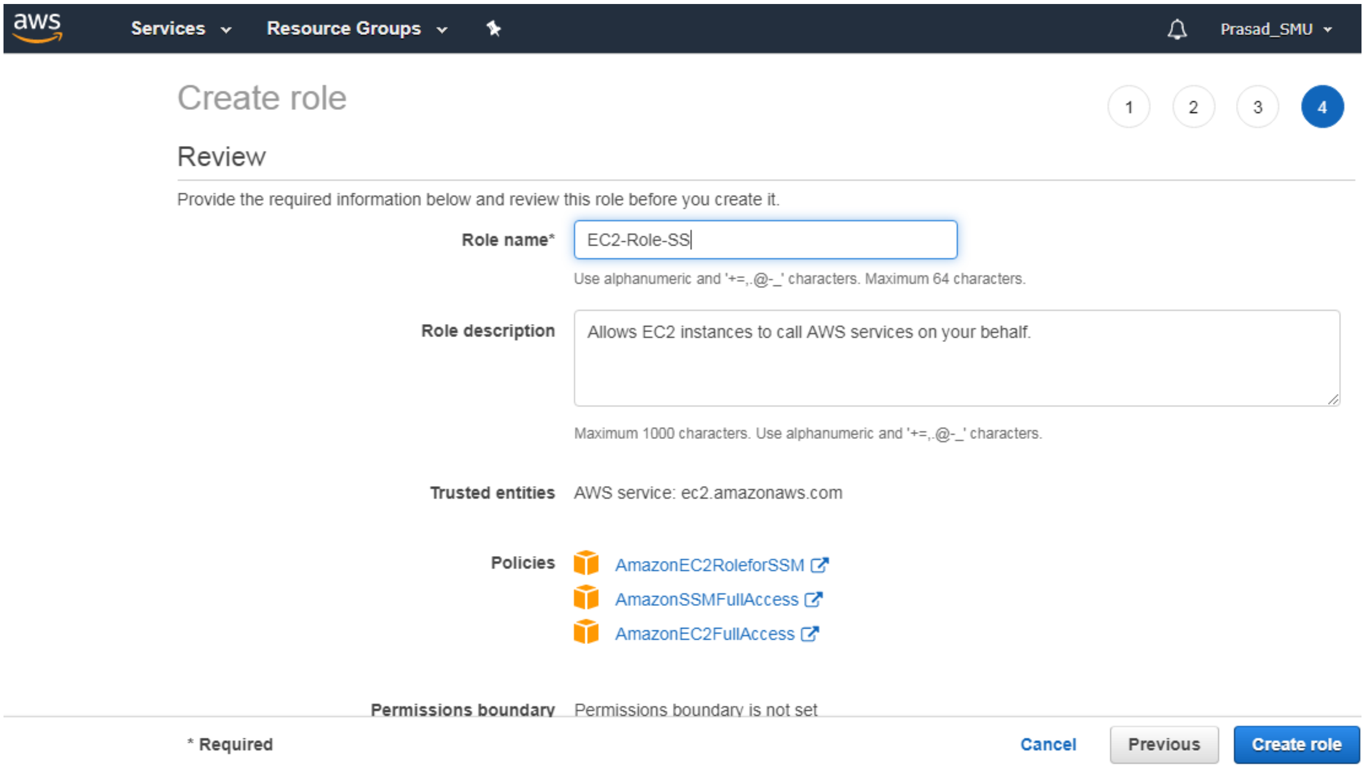
Choose one or more policies to attach to your new role.

Search: SSM — Showing 13 results

	Policy name	Used as
☑	AmazonEC2RoleforSSM	Permissions policy (1)
□	AmazonSSMAutomationApproverAccess	None
□	AmazonSSMAutomationRole	None
□	AmazonSSMDirectoryServiceAccess	None
☑	AmazonSSMFullAccess	Permissions policy (1)
□	AmazonSSMMaintenanceWindowRole	None
□	AmazonSSMManagedInstanceCore	Permissions policy (1)
□	AmazonSSMReadOnlyAccess	None

* Required

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



Create role

Review

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

Role name*: EC2-Role-SS

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
- AmazonEC2FullAccess

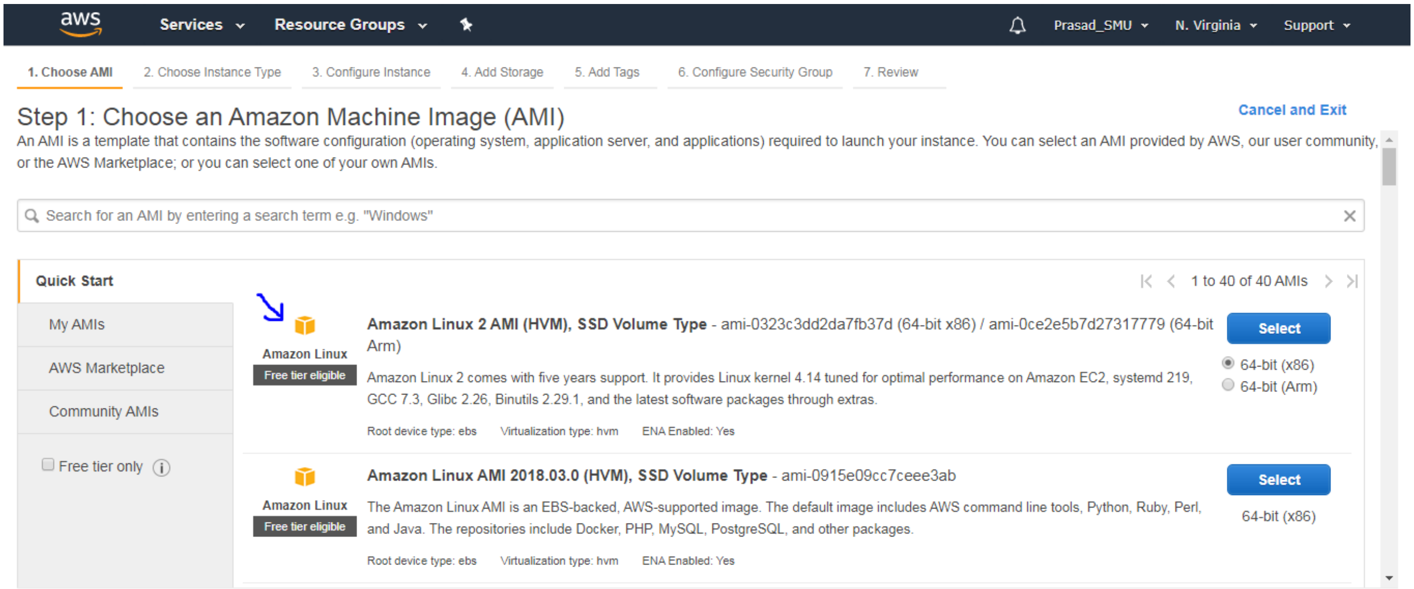
Permissions boundary: Permissions boundary is not set

* Required

Task 2: Launch & Configure EC2 Instances with SSM Agent

Navigate to EC2 Service and click on Launch Instance.

Select the **Amazon Linux AMI**.



Step 1: Choose an Amazon Machine Image (AMI)

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"

□ Free tier only

Amazon Linux — Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d (64-bit x86) / ami-0ce2e5b7d27317779 (64-bit Arm)

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

● 64-bit (x86)
○ 64-bit (Arm)

Amazon Linux — Free tier eligible

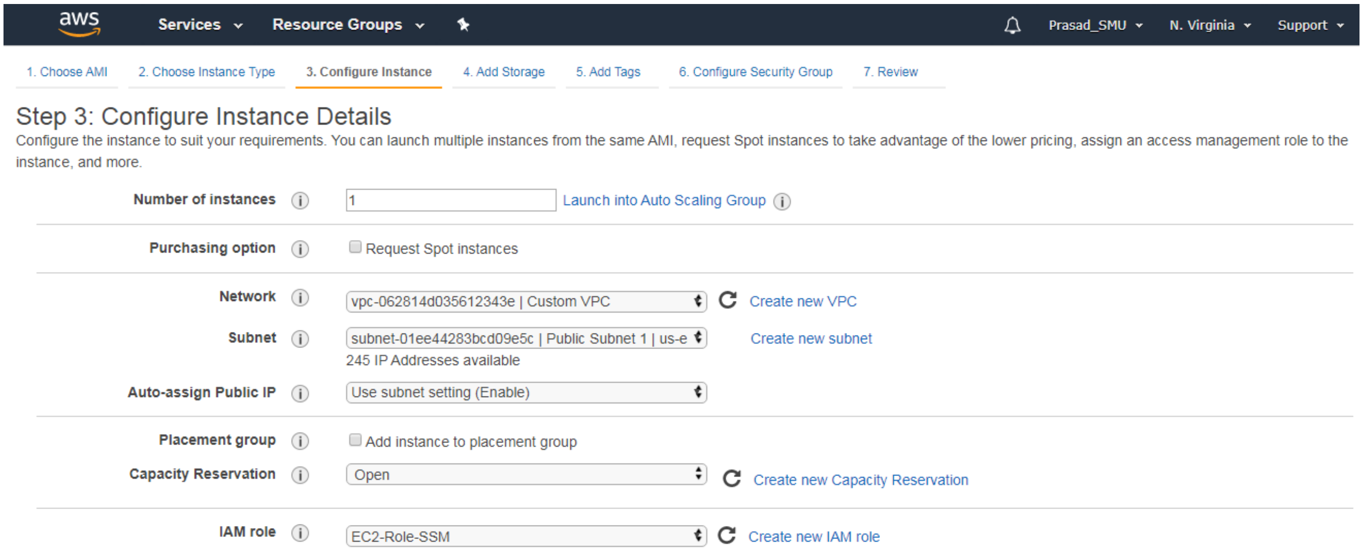
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0915e09cc7ceee3ab

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

64-bit (x86)

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.



Step 3: Configure Instance Details

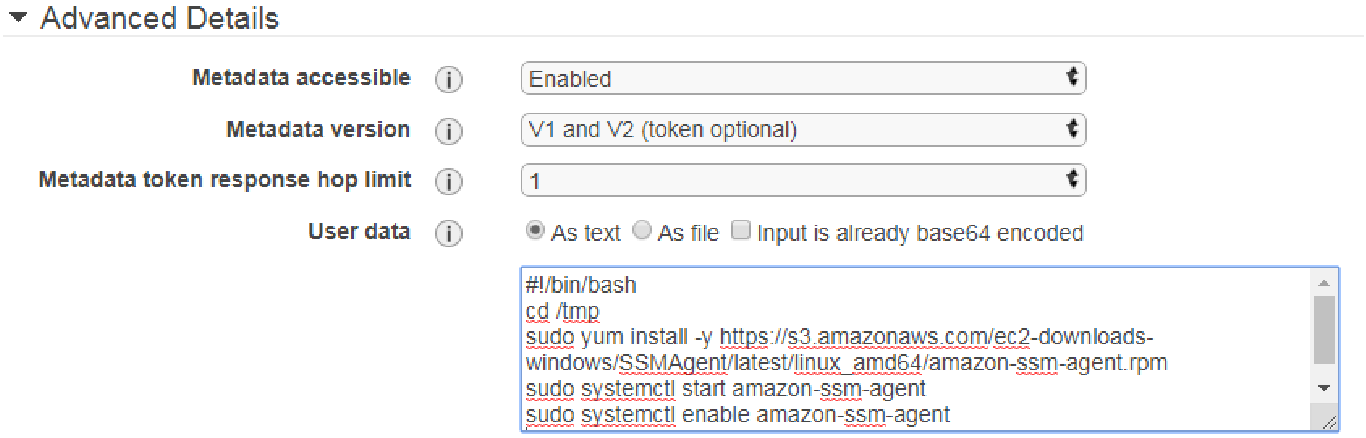
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	□ Request Spot instances
Network	vpc-062814d035612343e | Custom VPC
Subnet	subnet-01ee44283bcd09e5c | Public Subnet 1 | us-e 245 IP Addresses available
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	□ Add instance to placement group
Capacity Reservation	Open
IAM role	EC2-Role-SSM

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.

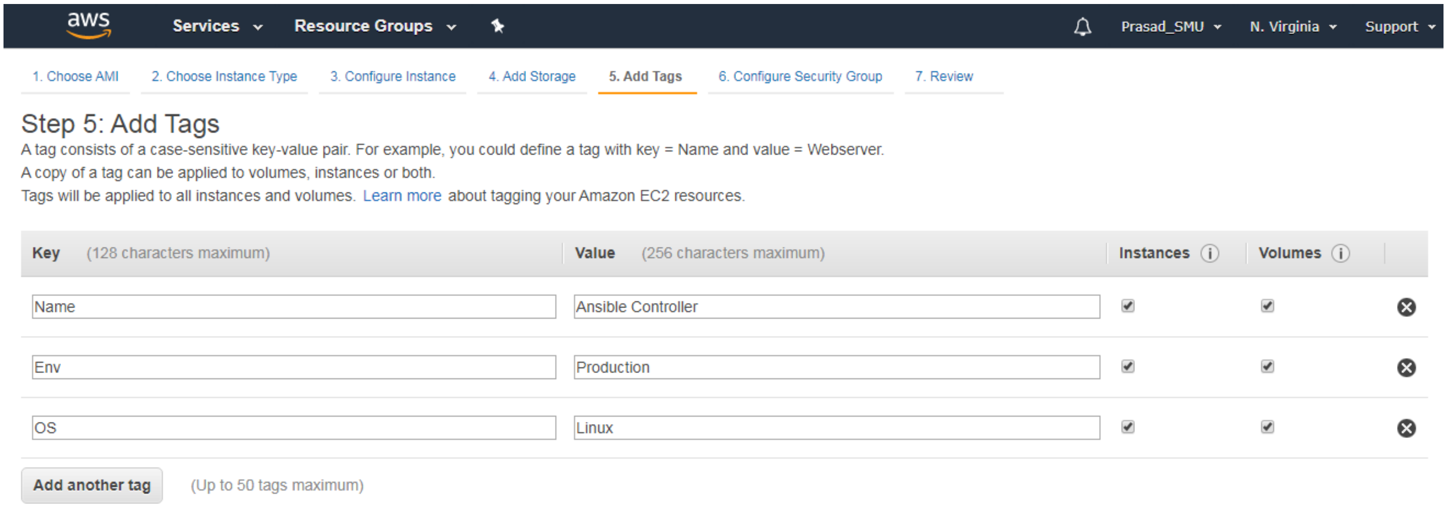


▾ Advanced Details

Metadata accessible	Enabled
Metadata version	V1 and V2 (token optional)
Metadata token response hop limit	1
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.



Step 5: Add Tags

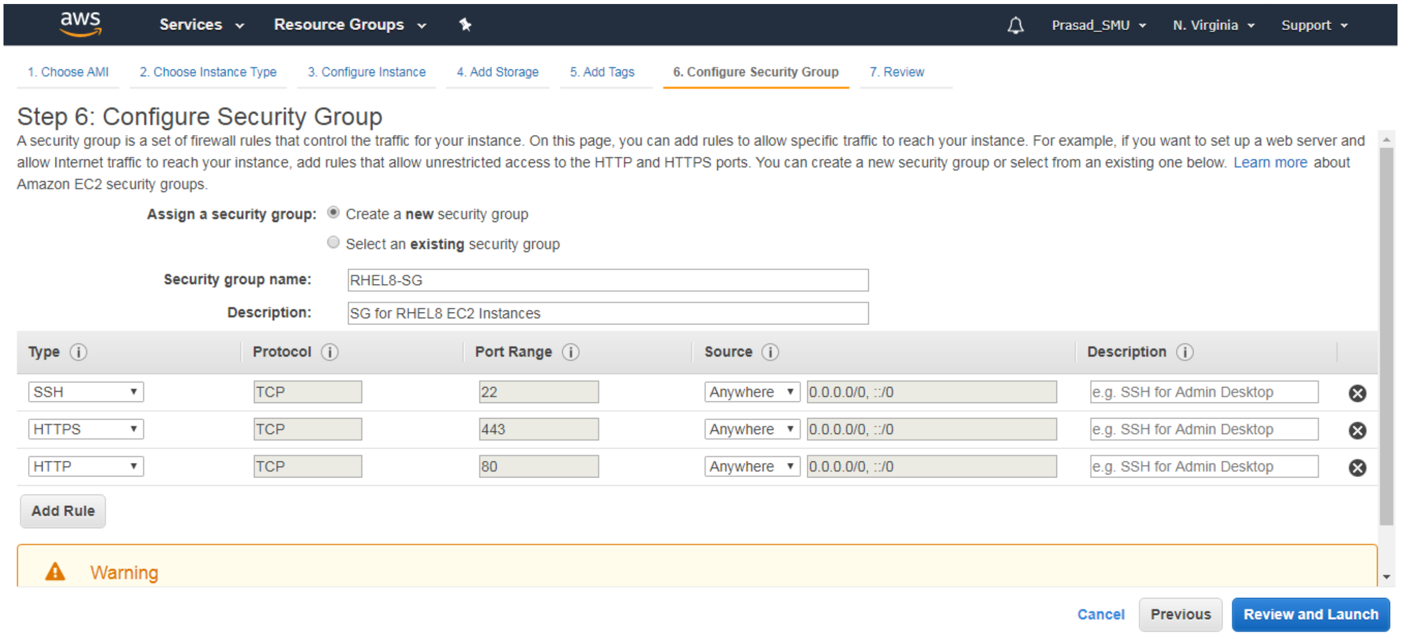
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	
Name	Ansible Controller	☑	☑	⊗
Env	Production	☑	☑	⊗
OS	Linux	☑	☑	⊗

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.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your server, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: RHEL8-SG

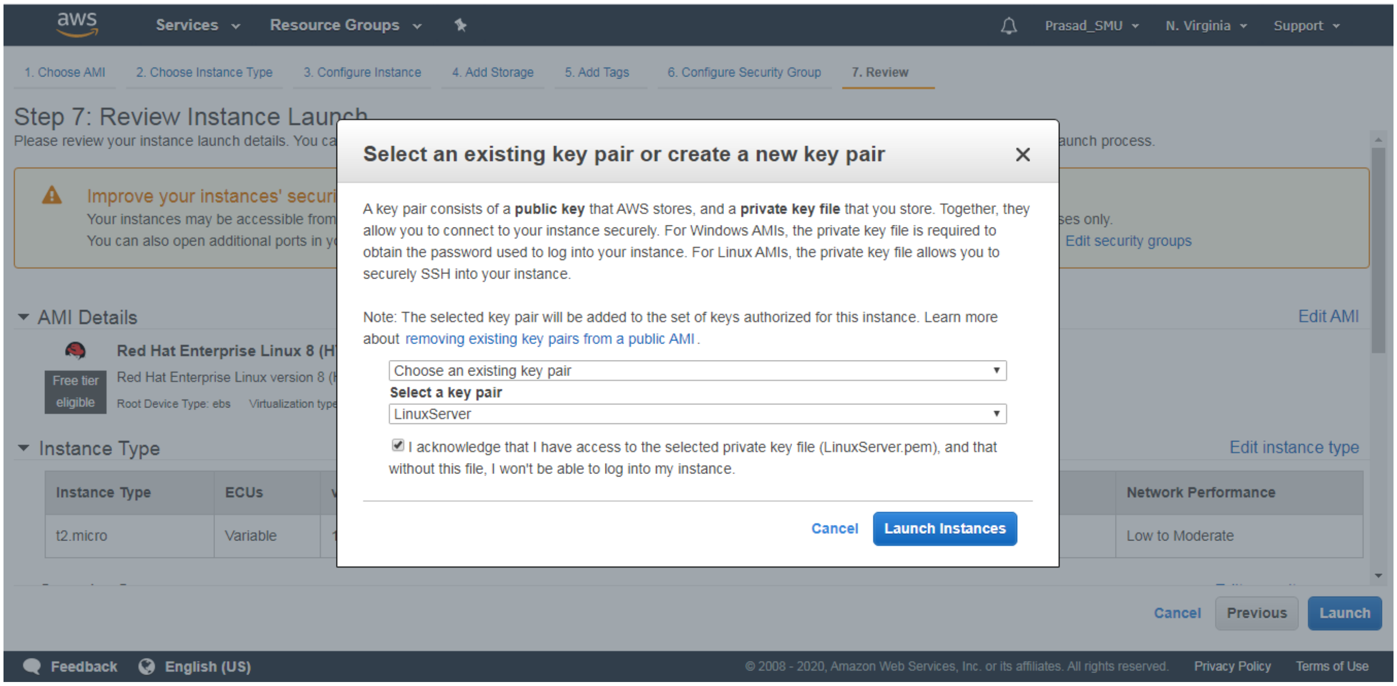
Description: SG for RHEL8 EC2 Instances

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

Click on Review and Launch.

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

Click on Launch Instances.


aws
Services
Resource Groups
Prasad_SMU
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Please review your instance launch details. You ca
aunch process.
Improve your instances' securi
Your instances may be accessible from
You can also open additional ports in y
es only.
Edit security groups
AMI Details
Edit AMI
Red Hat Enterprise Linux 8 (H
Free tier eligible
Red Hat Enterprise Linux version 8 (
Root Device Type: ebs Virtualization type
Instance Type
Edit instance type
Instance Type
ECUs
Network Performance
t2.micro
Variable
Low to Moderate
Cancel
Previous
Launch
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

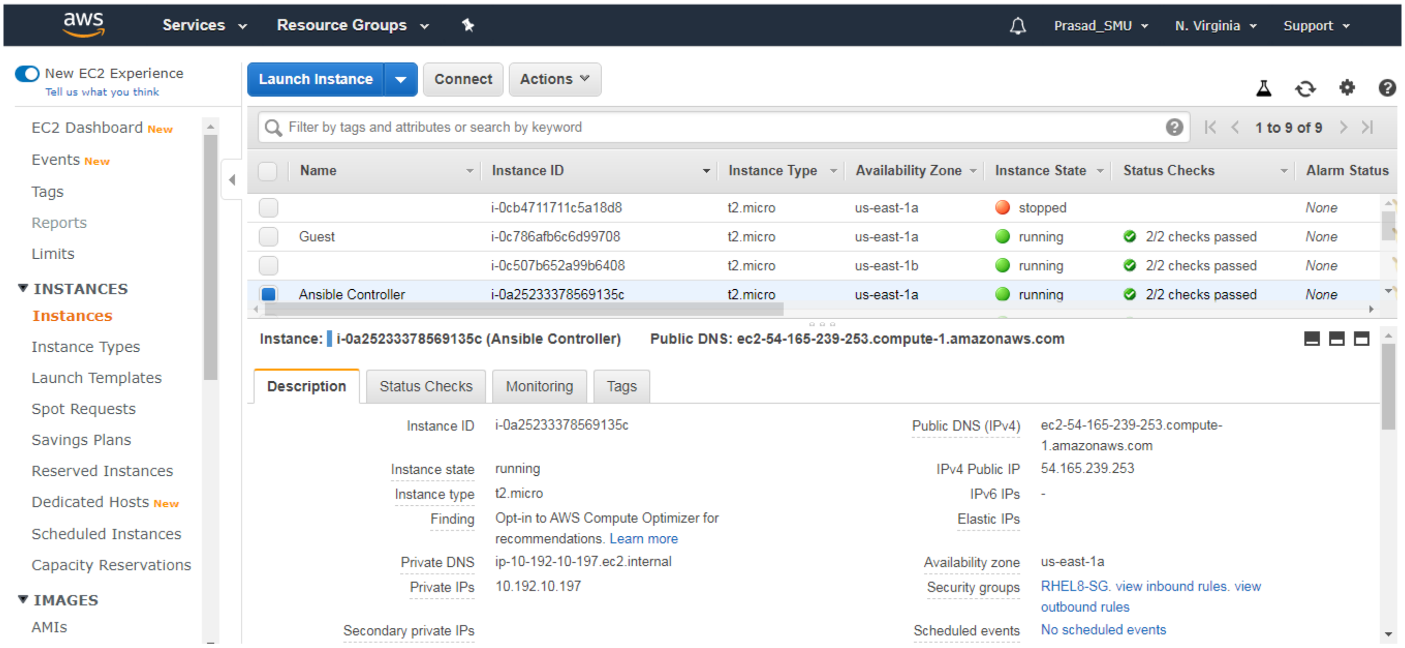
Choose an existing key pair

Select a key pair

LinuxServer

☑ I acknowledge that I have access to the selected private key file (LinuxServer.pem), and that without this file, I won't be able to log into my instance.

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



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
□		i-0cb4711711c5a18d8	t2.micro	us-east-1a	● stopped		None
□	Guest	i-0c786afb6c6d99708	t2.micro	us-east-1a	● running	✓ 2/2 checks passed	None
□		i-0c507b652a99b6408	t2.micro	us-east-1b	● running	✓ 2/2 checks passed	None
■	Ansible Controller	i-0a25233378569135c	t2.micro	us-east-1a	● running	✓ 2/2 checks passed	None

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

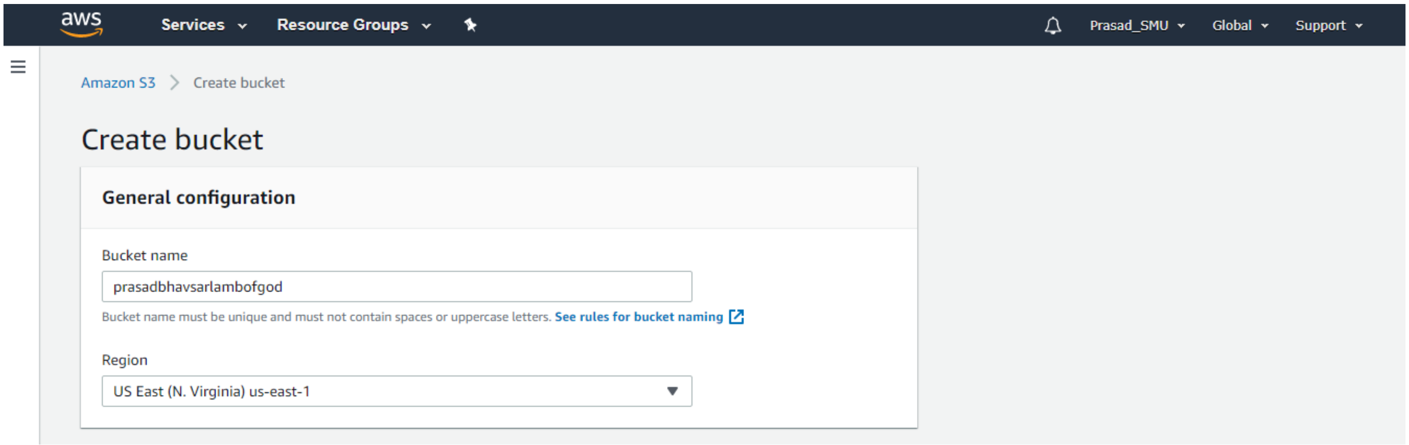
Instance ID	i-0a25233378569135c	Public DNS (IPv4)	ec2-54-165-239-253.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.165.239.253
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-10-192-10-197.ec2.internal	Availability zone	us-east-1a
Private IPs	10.192.10.197	Security groups	RHEL8-SG. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events

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.



Create bucket

General configuration

Bucket name

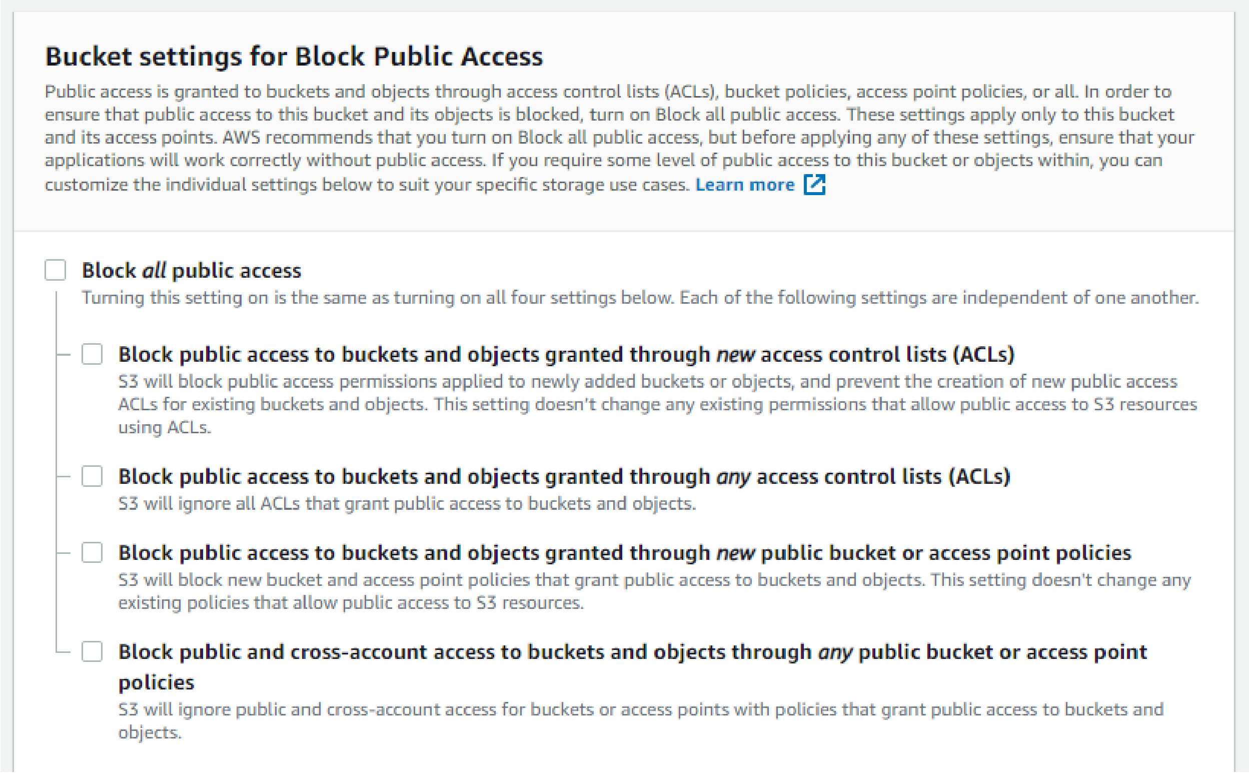
prasadbhavsarlambofgod

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

Region

US East (N. Virginia) us-east-1

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



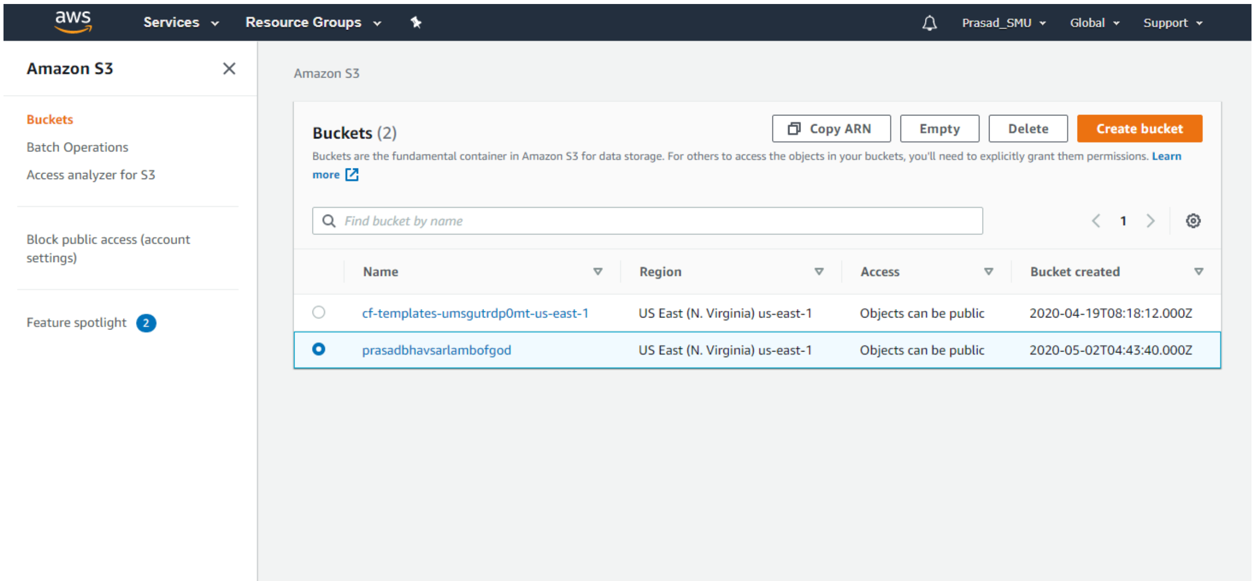
Bucket settings for Block Public Access

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.



Amazon S3

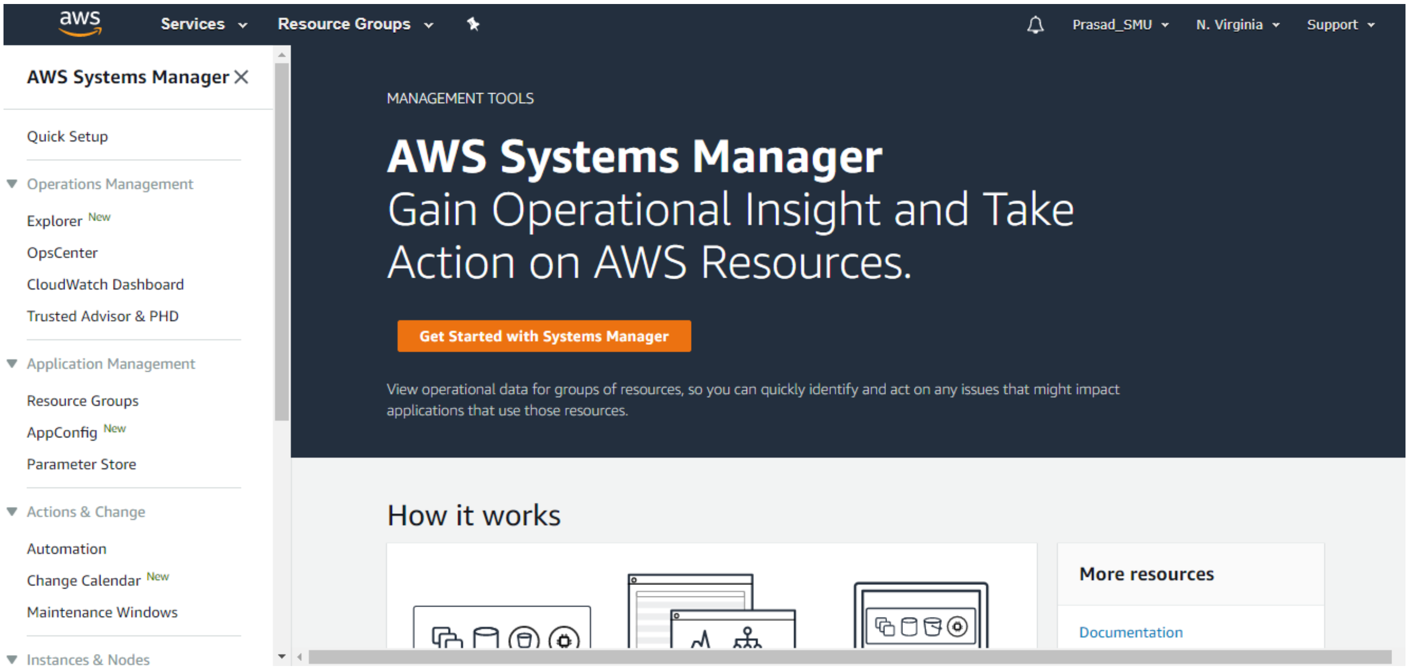
Buckets (2)

Buckets are the fundamental container in Amazon S3 for data storage. For others to access the objects in your buckets, you'll need to explicitly grant them permissions. Learn more

	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
●	prasadbhavsarlambofgod	US East (N. Virginia) us-east-1	Objects can be public	2020-05-02T04:43:40.000Z

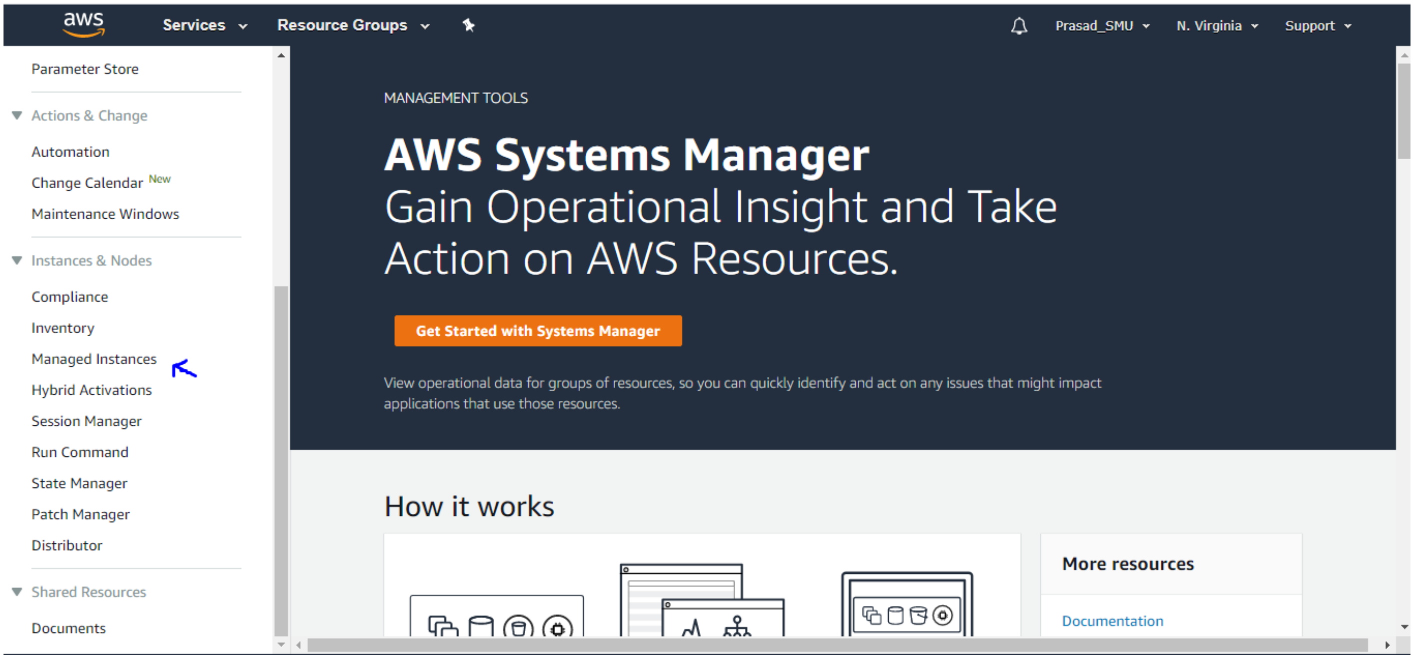
Task 4: AWS Systems Manager: Managed Instances

Navigate to AWS Systems Manager Service.



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.

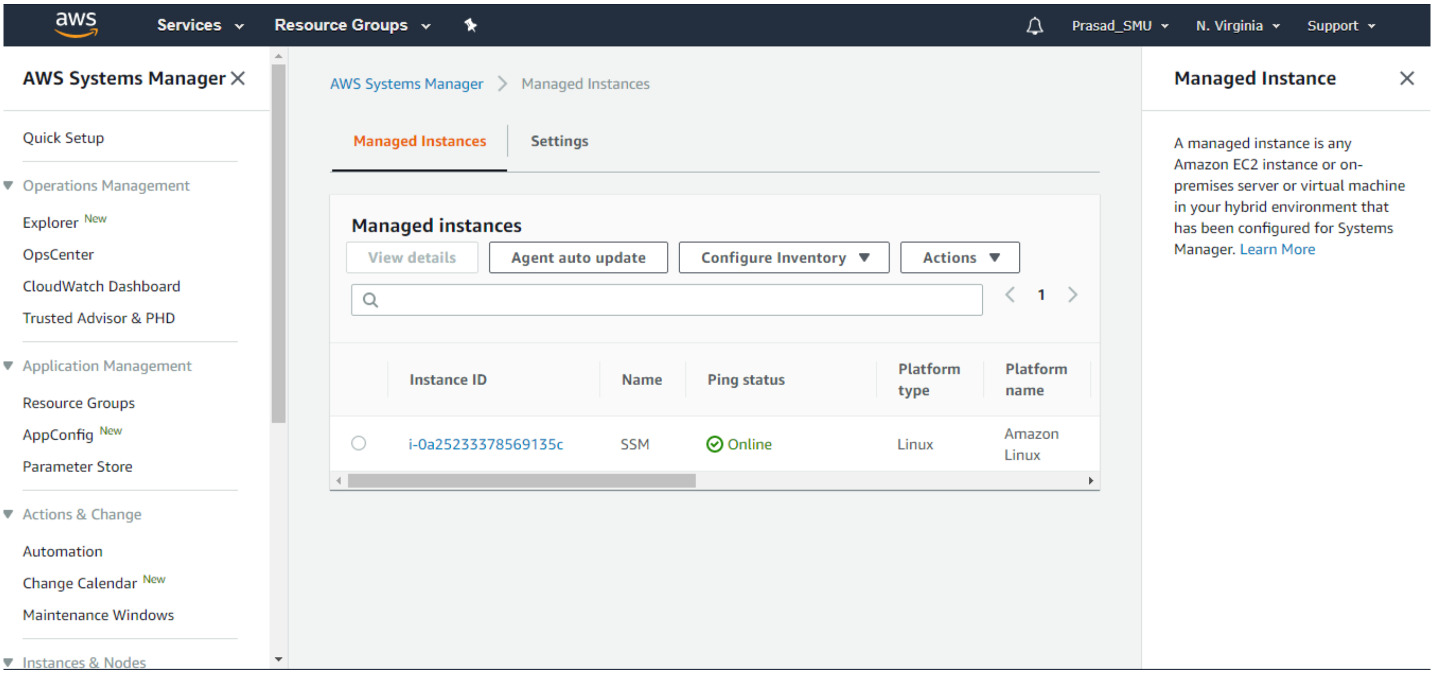
One the left-hand side, click on Managed Instances.



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.

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.



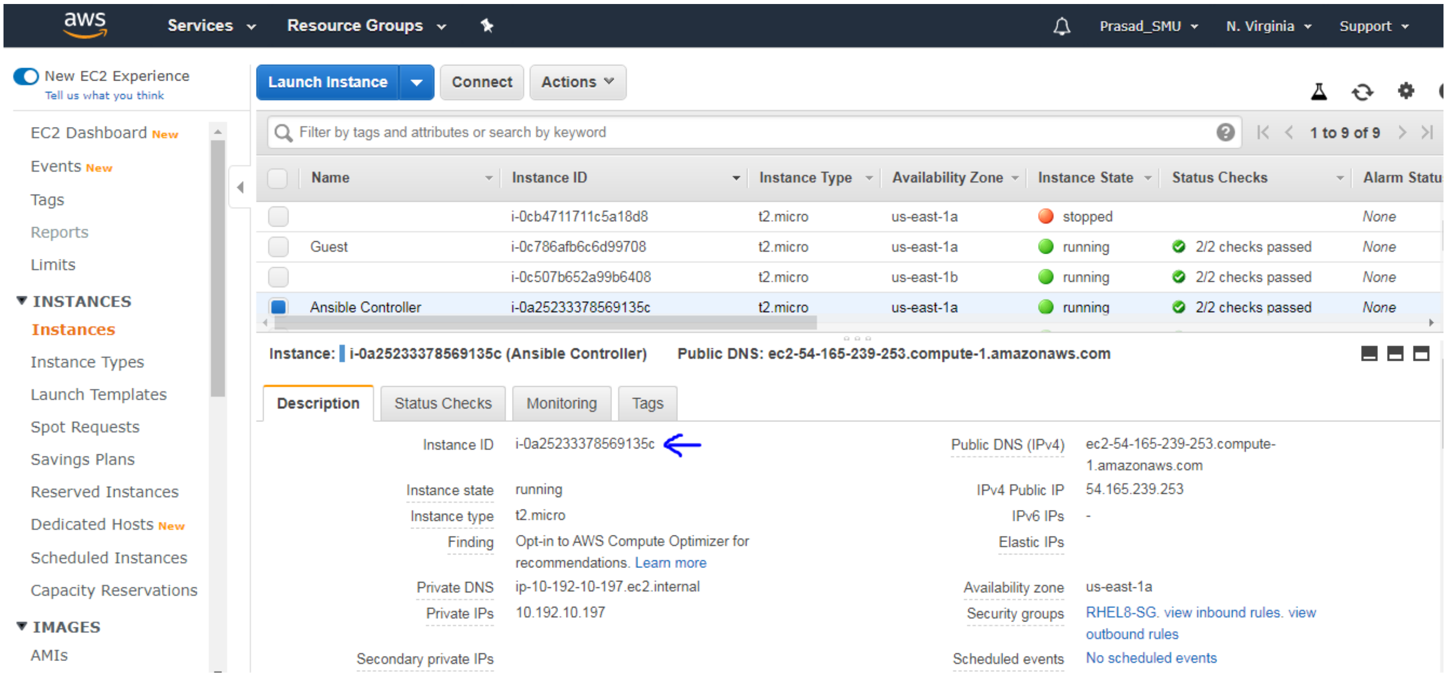
Managed instances

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

Managed Instance

A managed instance is any Amazon EC2 instance or on-premises server or virtual machine in your hybrid environment that has been configured for Systems Manager. Learn More

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



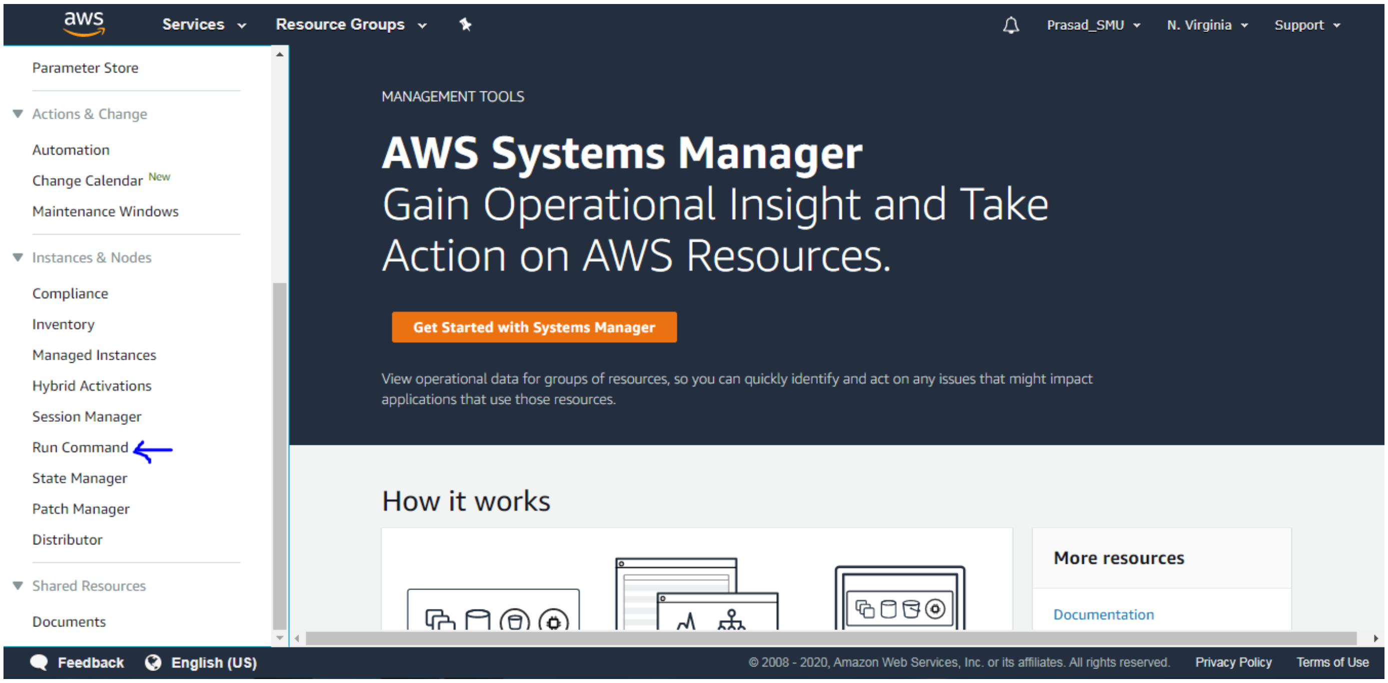
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
□		i-0cb4711711c5a18d8	t2.micro	us-east-1a	stopped		None
□	Guest	i-0c786afb6c6d99708	t2.micro	us-east-1a	running	2/2 checks passed	None
□		i-0c507b652a99b6408	t2.micro	us-east-1b	running	2/2 checks passed	None
■	Ansible Controller	i-0a25233378569135c	t2.micro	us-east-1a	running	2/2 checks passed	None

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

Instance ID	i-0a25233378569135c ←	Public DNS (IPv4)	ec2-54-165-239-253.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.165.239.253
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-10-192-10-197.ec2.internal	Availability zone	us-east-1a
Private IPs	10.192.10.197	Security groups	RHEL8-SG. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events

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

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

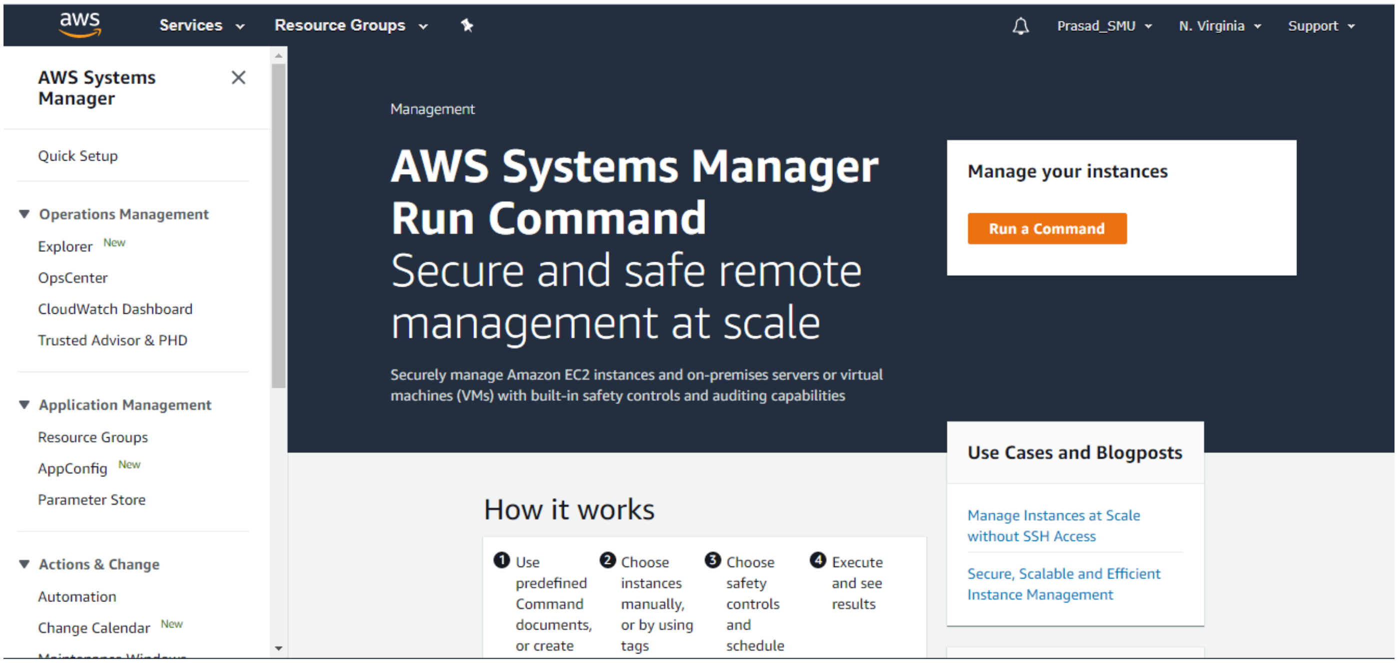


AWS Systems Manager
Gain Operational Insight and Take Action on AWS Resources.

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.

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click on Run a Command.



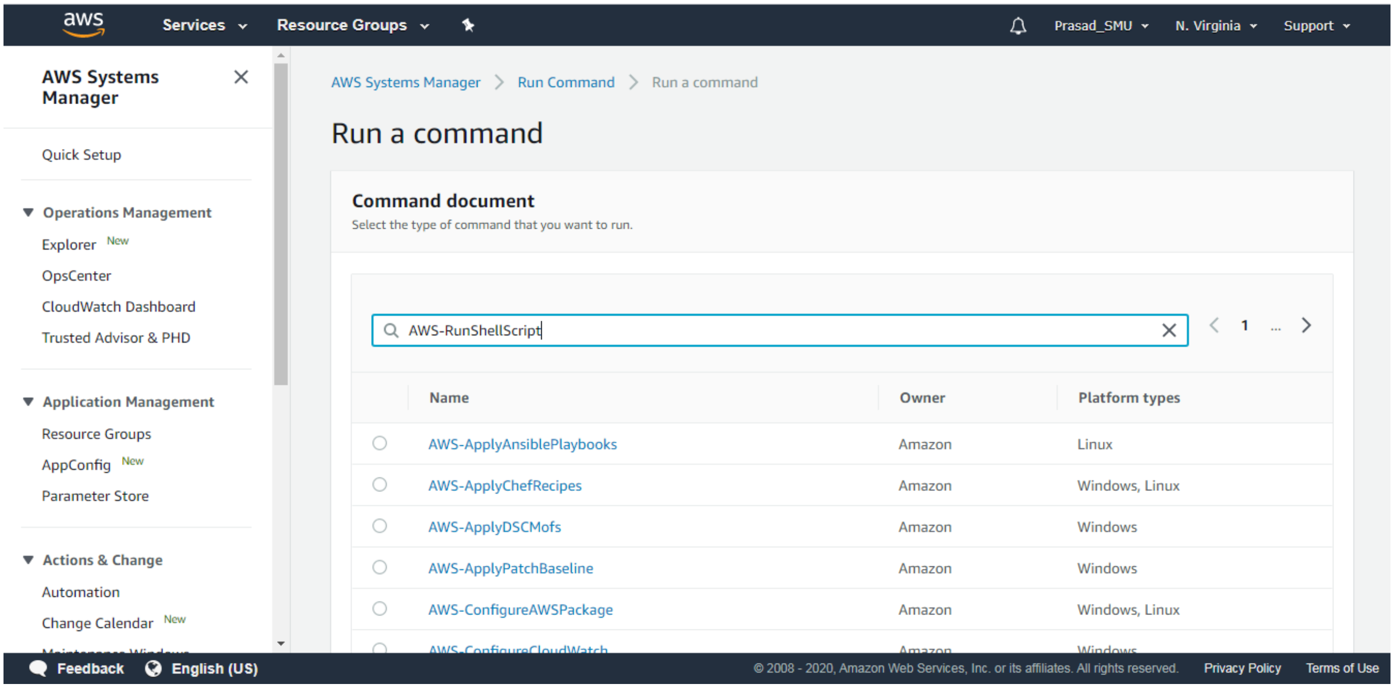
AWS Systems Manager Run Command
Secure and safe remote management at scale

Securely manage Amazon EC2 instances and on-premises servers or virtual machines (VMs) with built-in safety controls and auditing capabilities

1. Use predefined Command documents, or create
2. Choose instances manually, or by using tags
3. Choose safety controls and schedule
4. Execute and see results

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

AWS-RunShellScript



AWS Systems Manager > Run Command > Run a command

Run a command

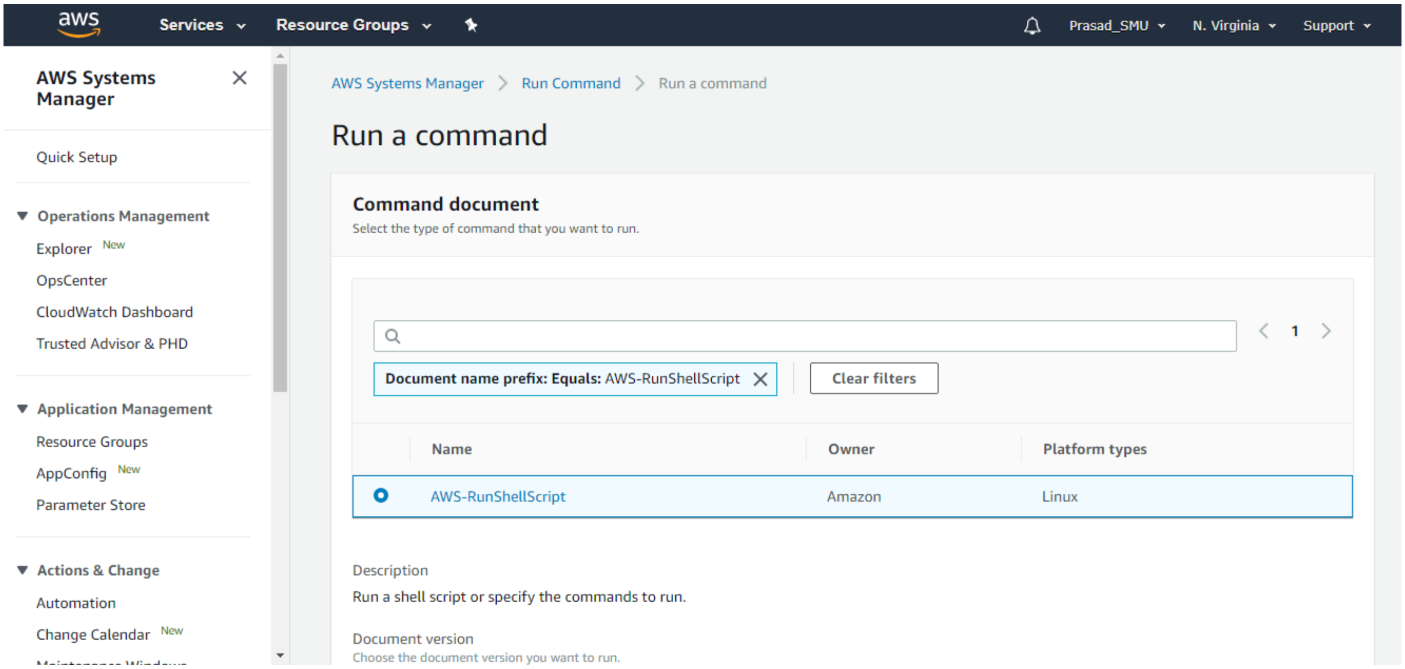
Command document

Select the type of command that you want to run.

AWS-RunShellScript

	Name	Owner	Platform types
○	AWS-ApplyAnsiblePlaybooks	Amazon	Linux
○	AWS-ApplyChefRecipes	Amazon	Windows, Linux
○	AWS-ApplyDSCMofs	Amazon	Windows
○	AWS-ApplyPatchBaseline	Amazon	Windows
○	AWS-ConfigureAWSPackage	Amazon	Windows, Linux
○	AWS-ConfigureCloudWatch	Amazon	Windows

Select the Command Document.



AWS Systems Manager > Run Command > Run a command

Run a command

Command document

Select the type of command that you want to run.

Document name prefix: Equals: AWS-RunShellScript

	Name	Owner	Platform types
●	AWS-RunShellScript	Amazon	Linux

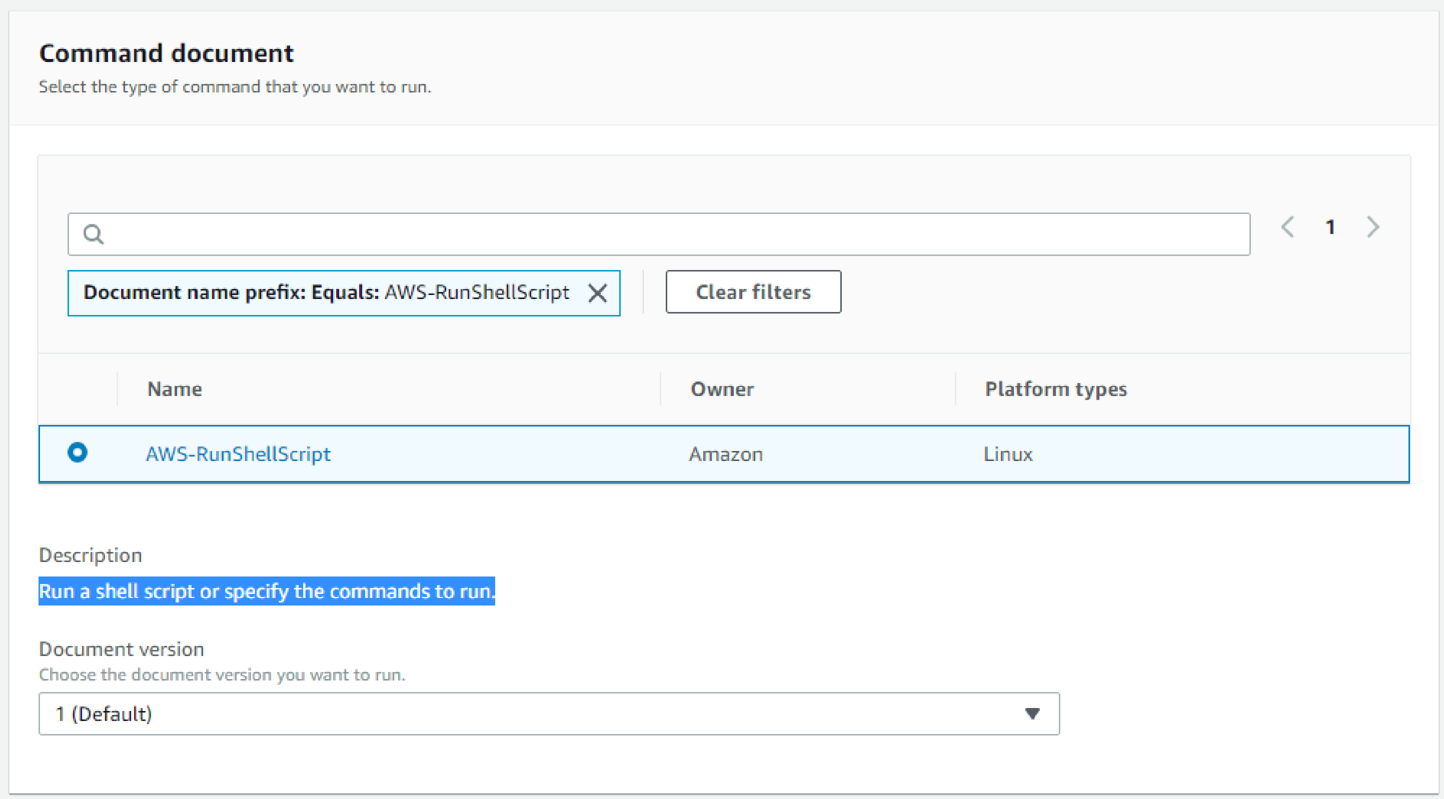
Description

Run a shell script or specify the commands to run.

Document version

Choose the document version you want to run.

Read the highlighted Description.



Command document

Select the type of command that you want to run.

Document name prefix: Equals: AWS-RunShellScript ✕

	Name	Owner	Platform types
●	AWS-RunShellScript	Amazon	Linux

Description

Run a shell script or specify the commands to run.

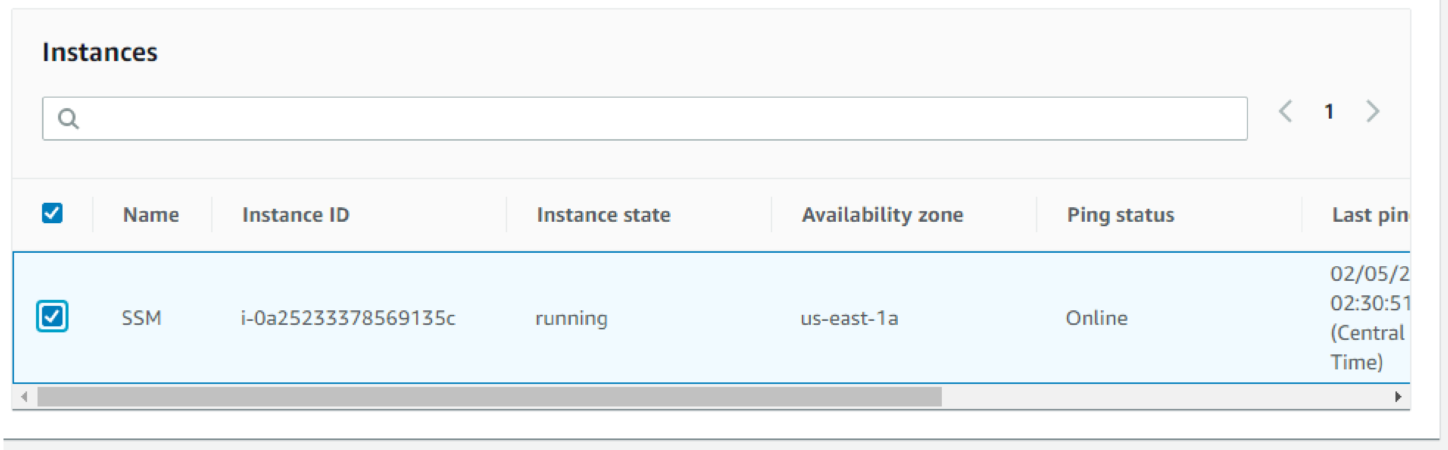
Document version

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.

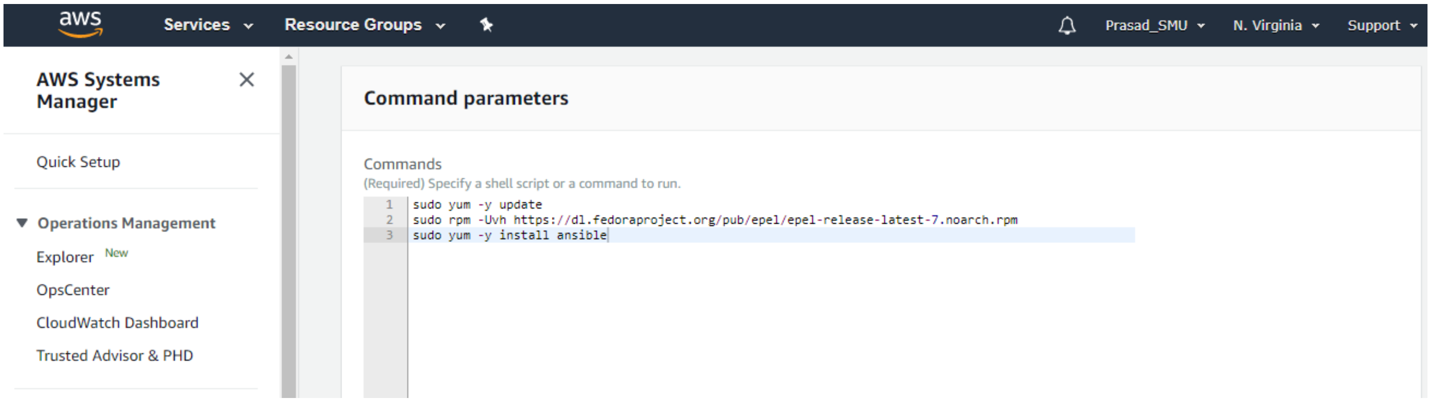


Instances

☑	Name	Instance ID	Instance state	Availability zone	Ping status	Last pin
☑	SSM	i-0a25233378569135c	running	us-east-1a	Online	02/05/2 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.



Command parameters

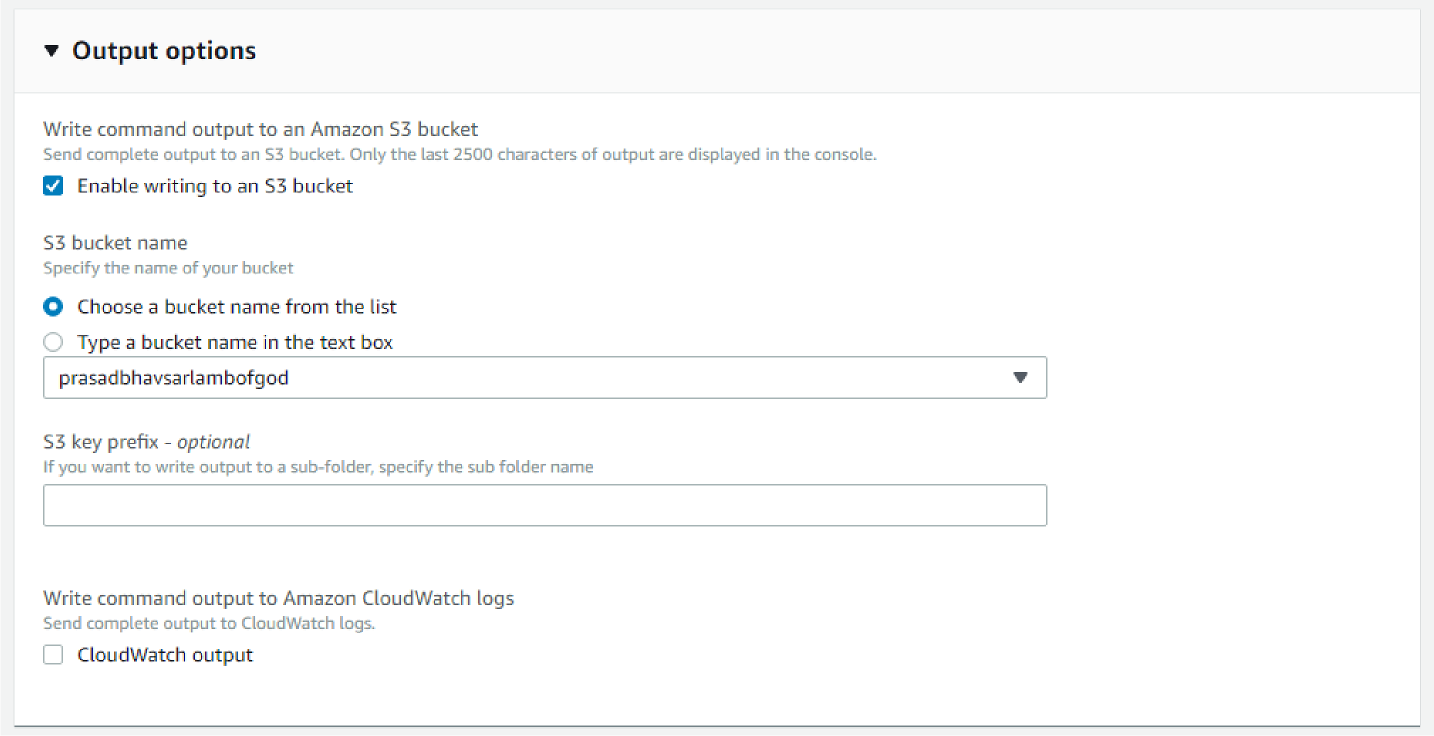
Commands
(Required) Specify a shell script or a command to run.

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

prasadbhavsarlambofgod ▼

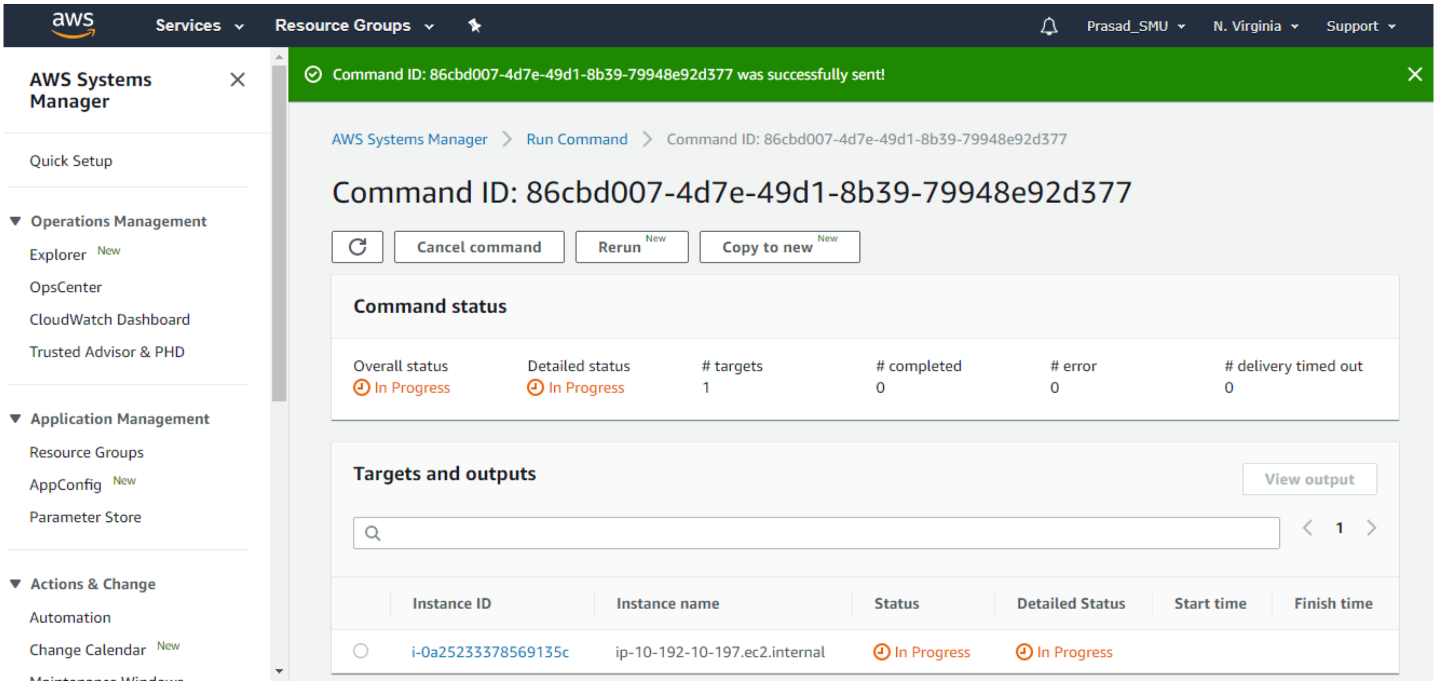
S3 key prefix - *optional*
If you want to write output to a sub-folder, specify the sub folder name

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.



Command ID: 86cbd007-4d7e-49d1-8b39-79948e92d377 was successfully sent!

AWS Systems Manager > Run Command > Command ID: 86cbd007-4d7e-49d1-8b39-79948e92d377

Command ID: 86cbd007-4d7e-49d1-8b39-79948e92d377

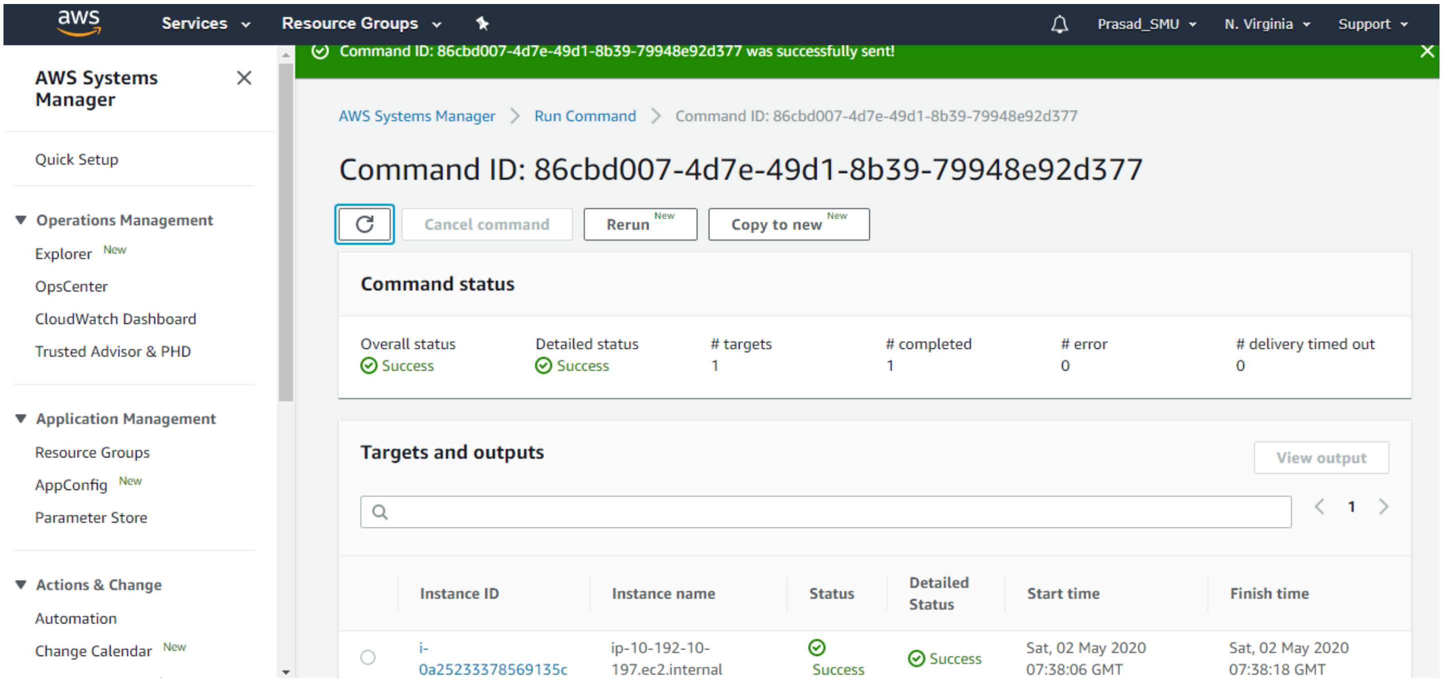
Command status

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

Targets and outputs

	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.**



Command ID: 86cbd007-4d7e-49d1-8b39-79948e92d377 was successfully sent!

AWS Systems Manager > Run Command > Command ID: 86cbd007-4d7e-49d1-8b39-79948e92d377

Command ID: 86cbd007-4d7e-49d1-8b39-79948e92d377

Command status

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

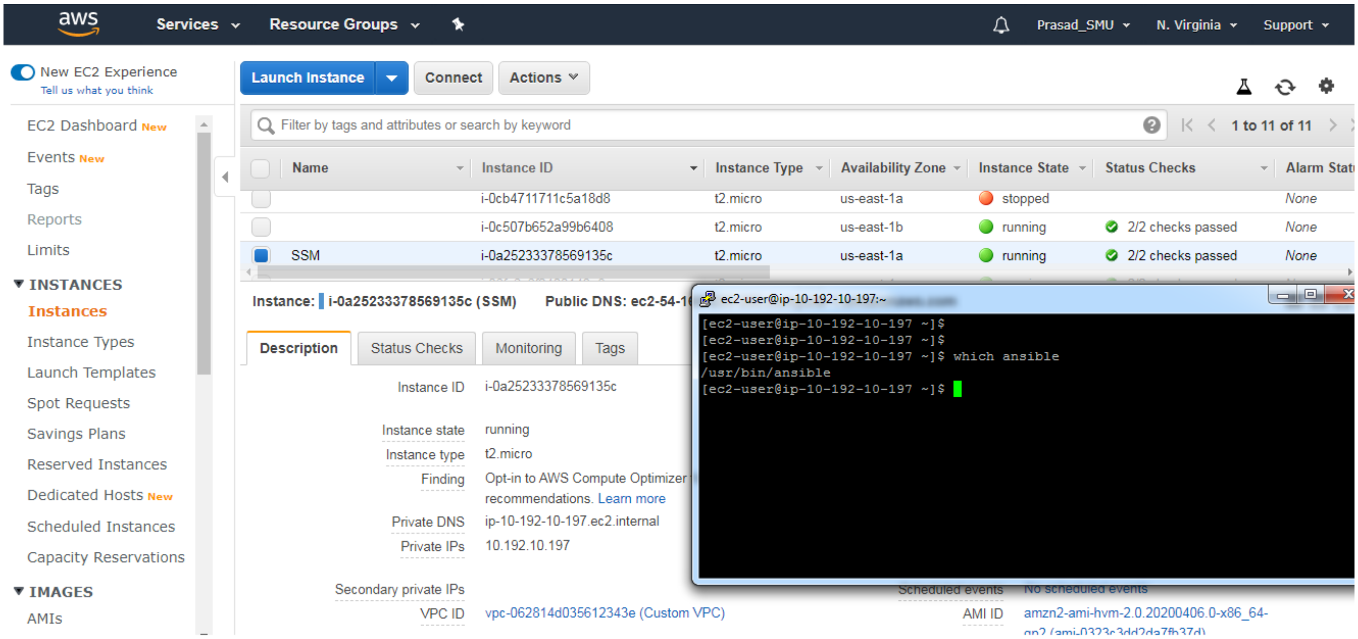
Targets and outputs

	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



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Stat
□		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: ec2-54-1

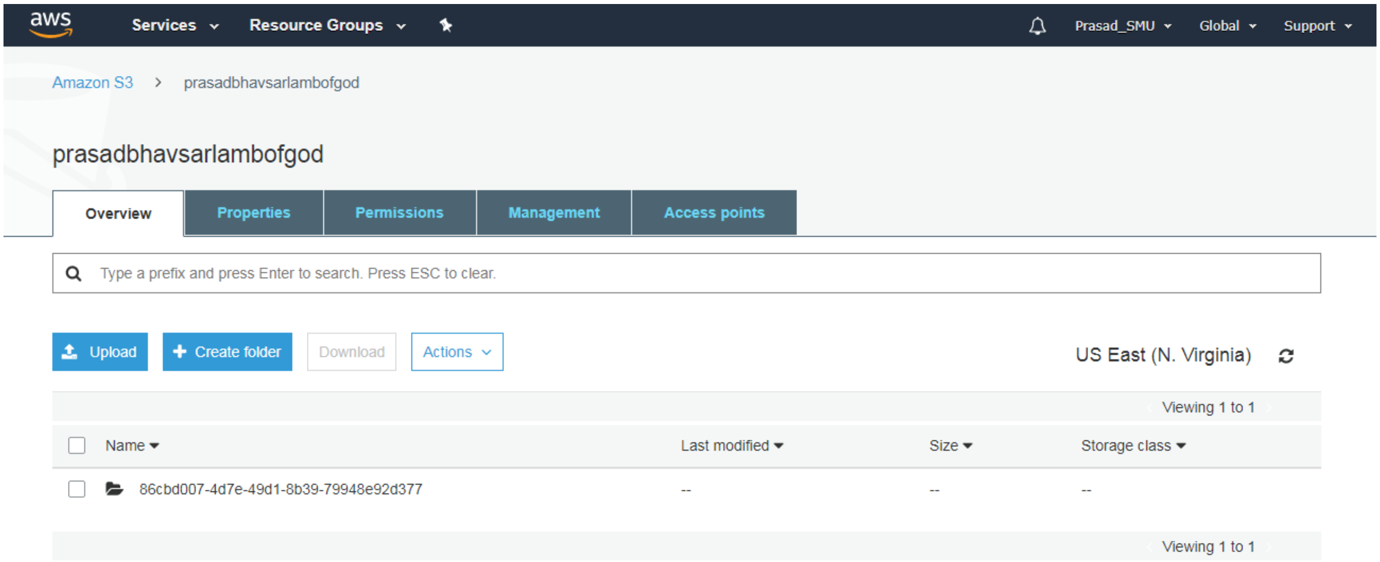
Instance ID: i-0a25233378569135c
Instance state: running
Instance type: t2.micro
Finding: Opt-in to AWS Compute Optimizer recommendations. Learn more
Private DNS: ip-10-192-10-197.ec2.internal
Private IPs: 10.192.10.197
Secondary private IPs:
VPC ID: vpc-062814d035612343e (Custom VPC)
Scheduled events: No scheduled events
AMI ID: amzn2-ami-hvm-2.0.20200406.0-x86_64-gp2 (ami-0323c3dd2da7fb37d)

```
[ec2-user@ip-10-192-10-197 ~]$
[ec2-user@ip-10-192-10-197 ~]$
[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.



Amazon S3 > prasadbhavsarlambofgod

prasadbhavsarlambofgod

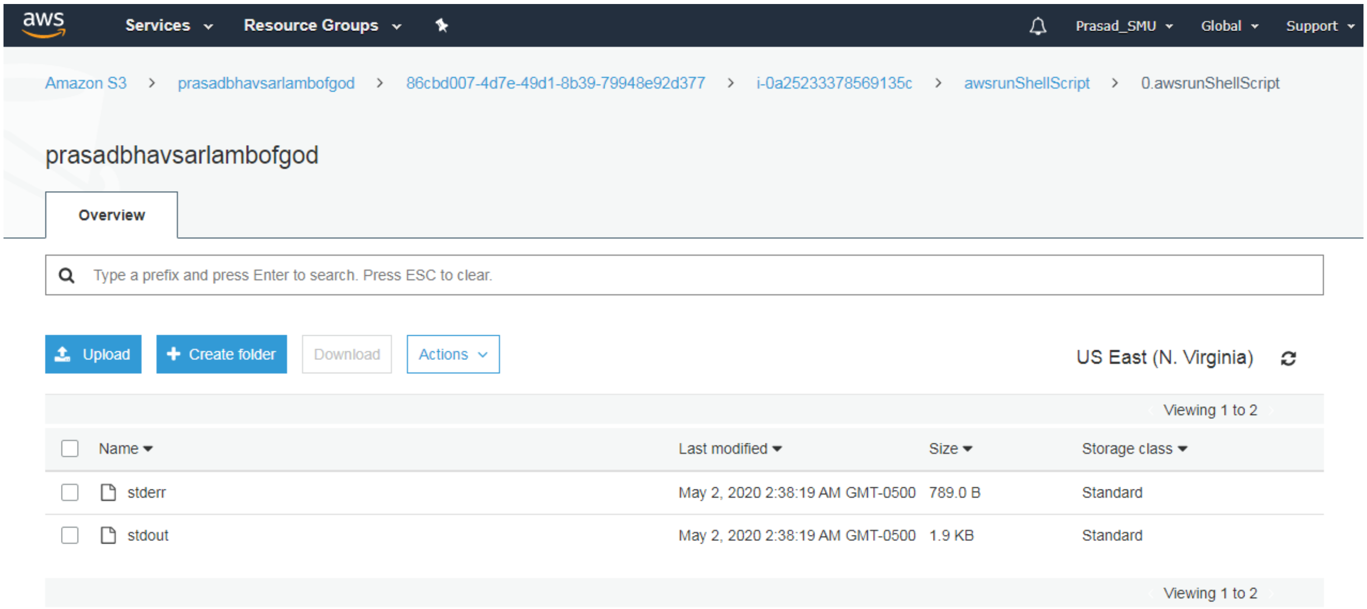
US East (N. Virginia)

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.



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

prasadbhavsarlambofgod

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

Viewing 1 to 2

	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

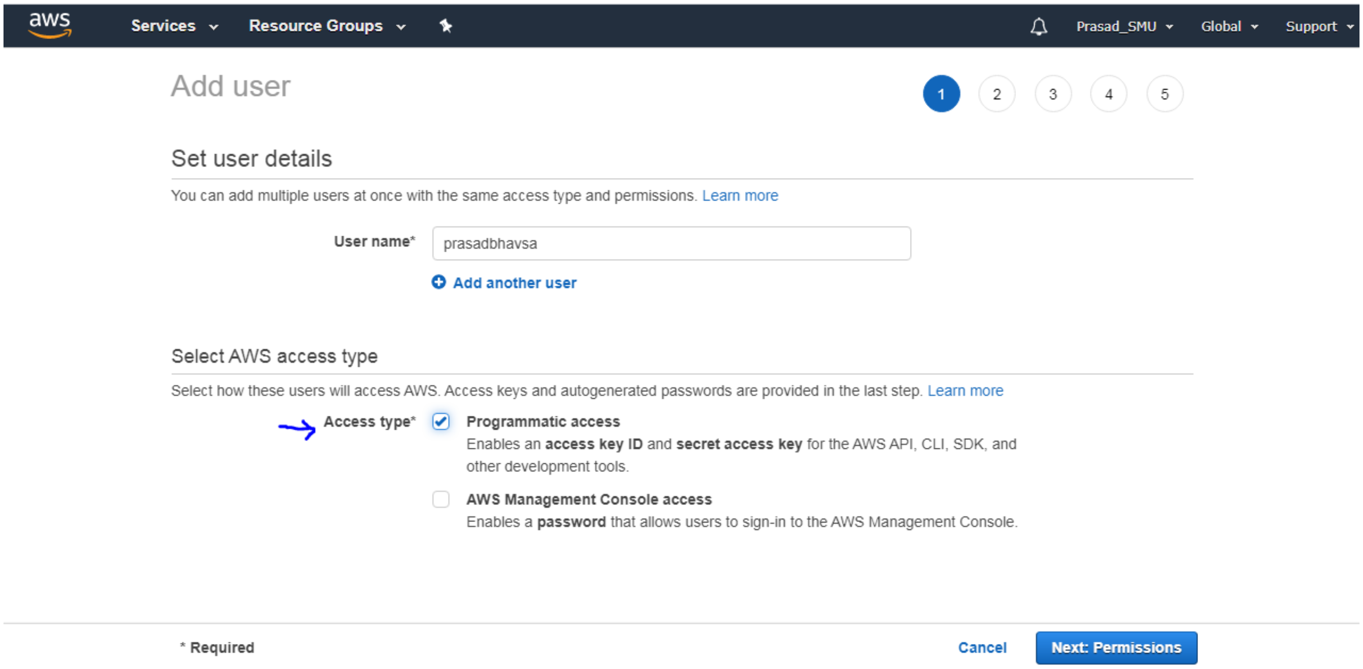
Viewing 1 to 2

Task 7: Create an IAM User

Navigate to IAM Service.

On left-hand side, click on Users and click on Add User.

Give the User name as per your choice and select the Access Type as **Programmatic Access**.



Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* prasadbhavsa

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

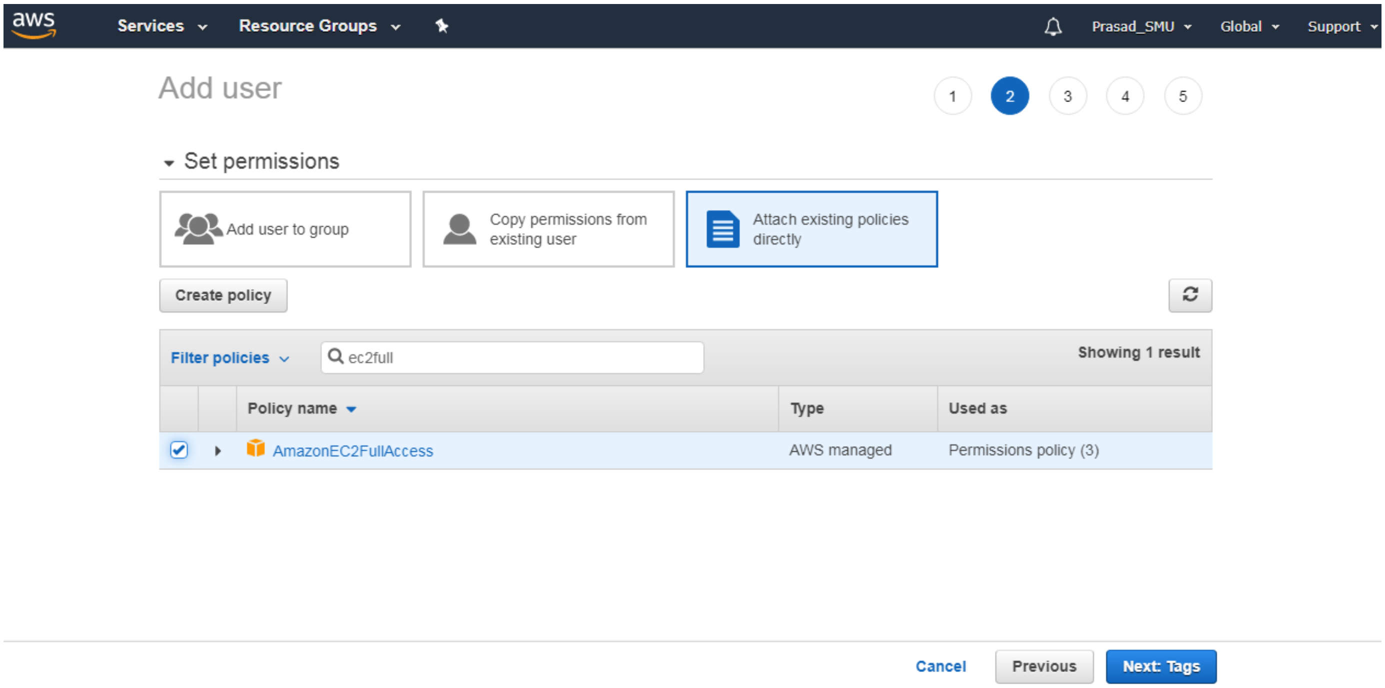
□ AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

Click on Next: Permissions.

Click on **Attach existing policies directly**.

Search and select **AmazonEC2FullAccess** Policy.



ec2full

Showing 1 result

		Policy name	Type	Used as
☑	▸	AmazonEC2FullAccess	AWS managed	Permissions policy (3)

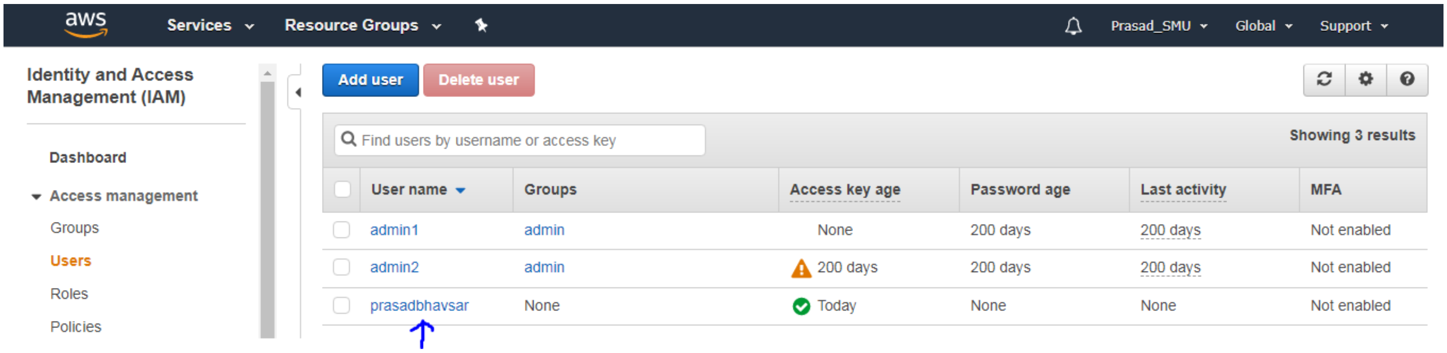
Click on Next: Tags.

You can add Tags if you wish else click Next: Review.

Review the configurations and click on Create User.

Make sure to note down the Access Key ID and Secret Access Key. You can also download the .csv file for a safe side. We will need these keys while doing boto configurations.

Click on Create User. User has been created successfully.



Showing 3 results

□	User name	Groups	Access key age	Password age	Last activity	MFA
□	admin1	admin	None	200 days	200 days	Not enabled
□	admin2	admin	⚠ 200 days	200 days	200 days	Not enabled
□	prasadbhavsar	None	✓ Today	None	None	Not enabled

Task 8: PIP, BOTO Configurations

Take SSH Session of the Ansible Controller EC2 Instance.

Python will be pre-installed on the Linux EC2 Instances.

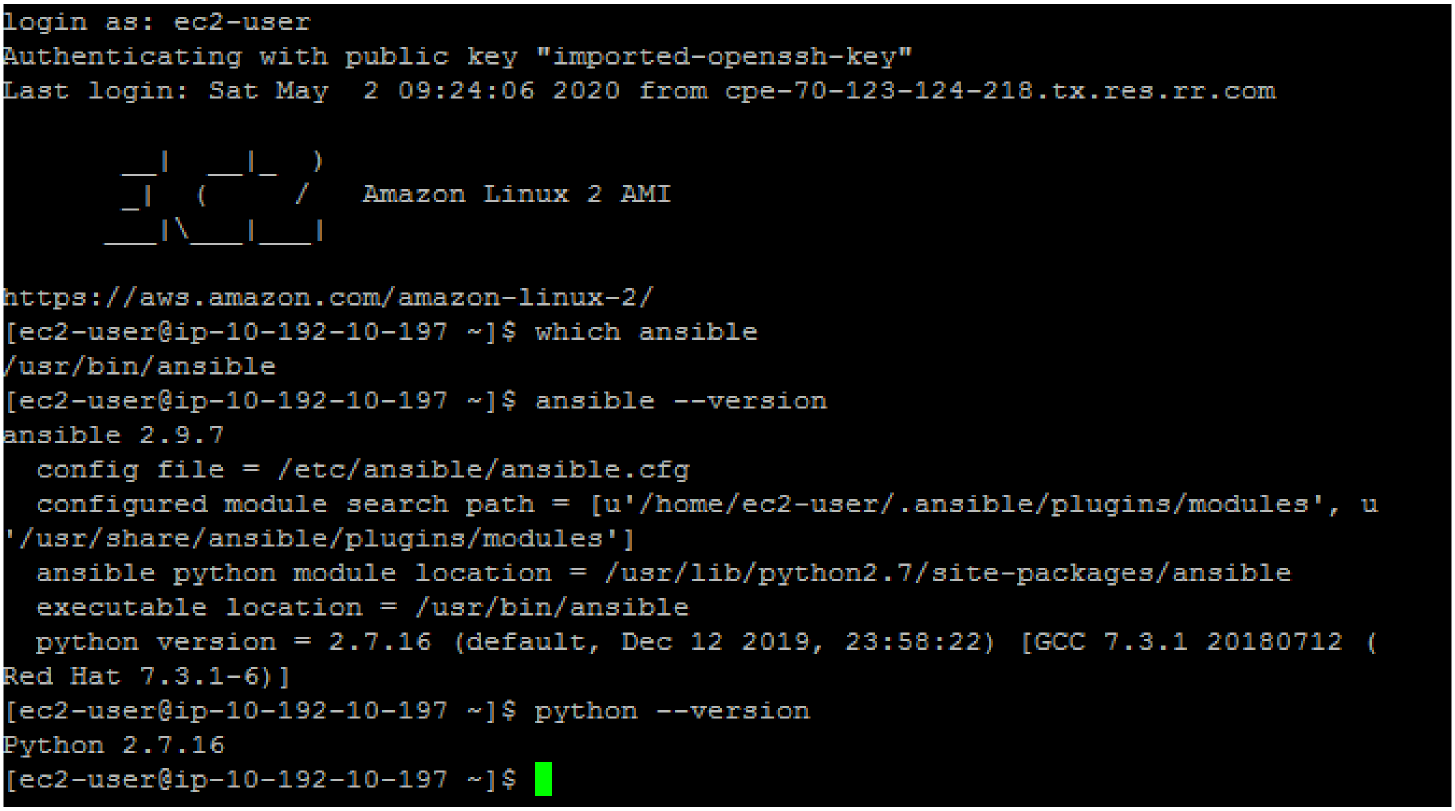
Make sure the Ansible and Python is Installed on the EC2 Instance. Run the below commands.

Commands:

which ansible

ansible –version

python --version

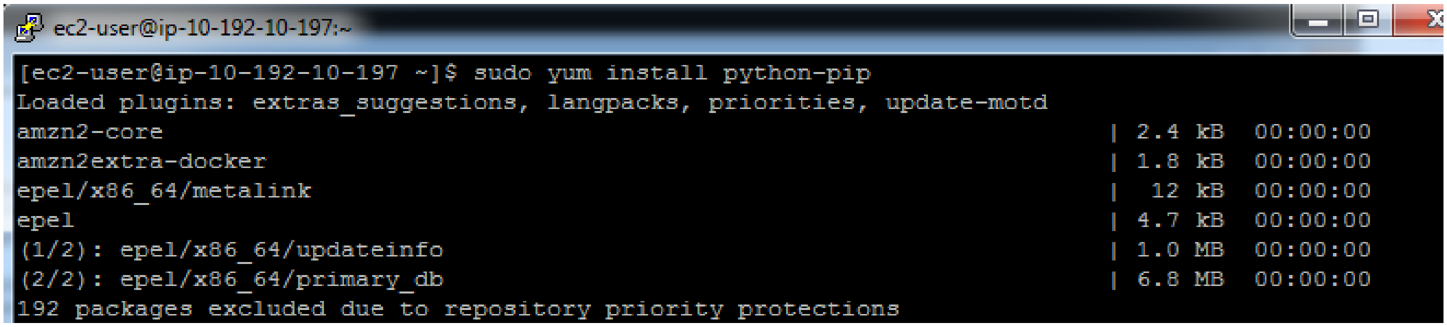


```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sat May  2 09:24:06 2020 from cpe-70-123-124-218.tx.res.rr.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-192-10-197 ~]$ which ansible
/usr/bin/ansible
[ec2-user@ip-10-192-10-197 ~]$ ansible --version
ansible 2.9.7
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/ec2-user/.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)]
[ec2-user@ip-10-192-10-197 ~]$ python --version
Python 2.7.16
[ec2-user@ip-10-192-10-197 ~]$
```

To install BOTO, we would need a Python Module "PIP".

To install Python Module "PIP", run the below command.

Command: sudo yum install python-pip

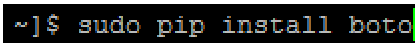


```
[ec2-user@ip-10-192-10-197 ~]$ sudo yum install python-pip
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                           | 2.4 kB  00:00:00
amzn2extra-docker                                                    | 1.8 kB  00:00:00
epel/x86_64/metalink                                                 |  12 kB  00:00:00
epel                                                                 | 4.7 kB  00:00:00
(1/2): epel/x86_64/updateinfo                                        | 1.0 MB  00:00:00
(2/2): epel/x86_64/primary_db                                        | 6.8 MB  00:00:00
192 packages excluded due to repository priority protections
```

Once PIP is installed, we will now install BOTO.

To install BOTO, run the below command.

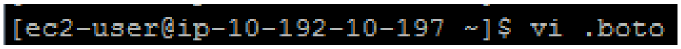
Command: sudo pip install boto



```
~]$ sudo pip install boto
```

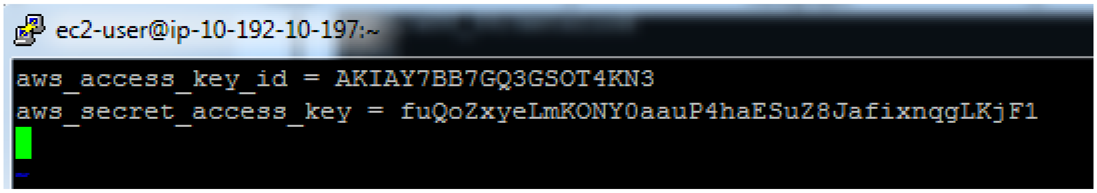
Now create a ".boto" file in your Home Directory.

Command: vi .boto



```
[ec2-user@ip-10-192-10-197 ~]$ vi .boto
```

Put the **aws_access_key_id** and **aws_secret_access_key** which you copied in Task 7 as follows.



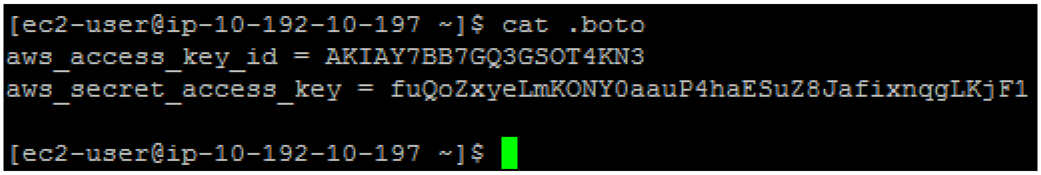
```
aws_access_key_id = AKIAY7BB7GQ3GSOT4KN3
aws_secret_access_key = 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

~
```

Save the vi editor by issuing below command.

Command: :wq!

Review the BOTO file again, issue the following command.

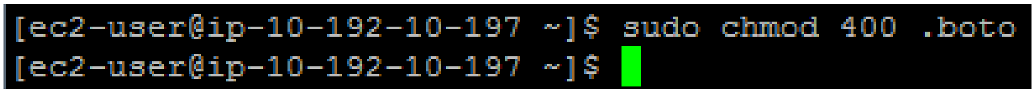
Command: cat .boto



```
[ec2-user@ip-10-192-10-197 ~]$ cat .boto
aws_access_key_id = AKIAY7BB7GQ3GSOT4KN3
aws_secret_access_key = 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

[ec2-user@ip-10-192-10-197 ~]$
```

Now save the .boto file with the permission 400.

Command: sudo chmod 400 .boto

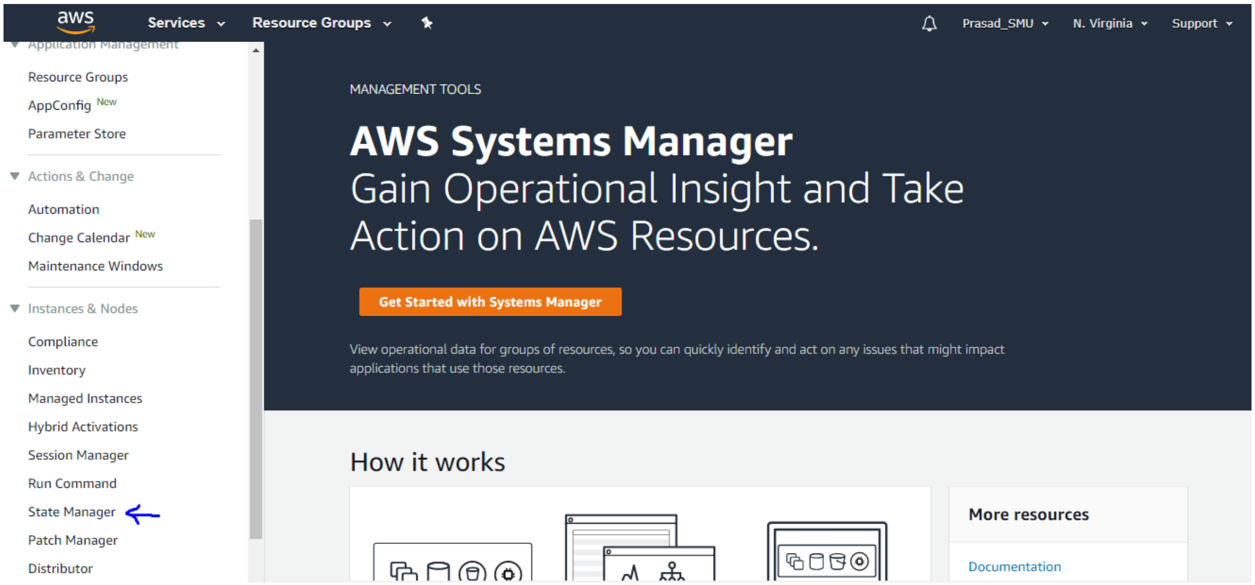


```
[ec2-user@ip-10-192-10-197 ~]$ sudo chmod 400 .boto
[ec2-user@ip-10-192-10-197 ~]$
```

PIP and BOTO configurations are now completed.

Task 9: Run an Ansible Playbook using AWS Systems Manager

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



MANAGEMENT TOOLS

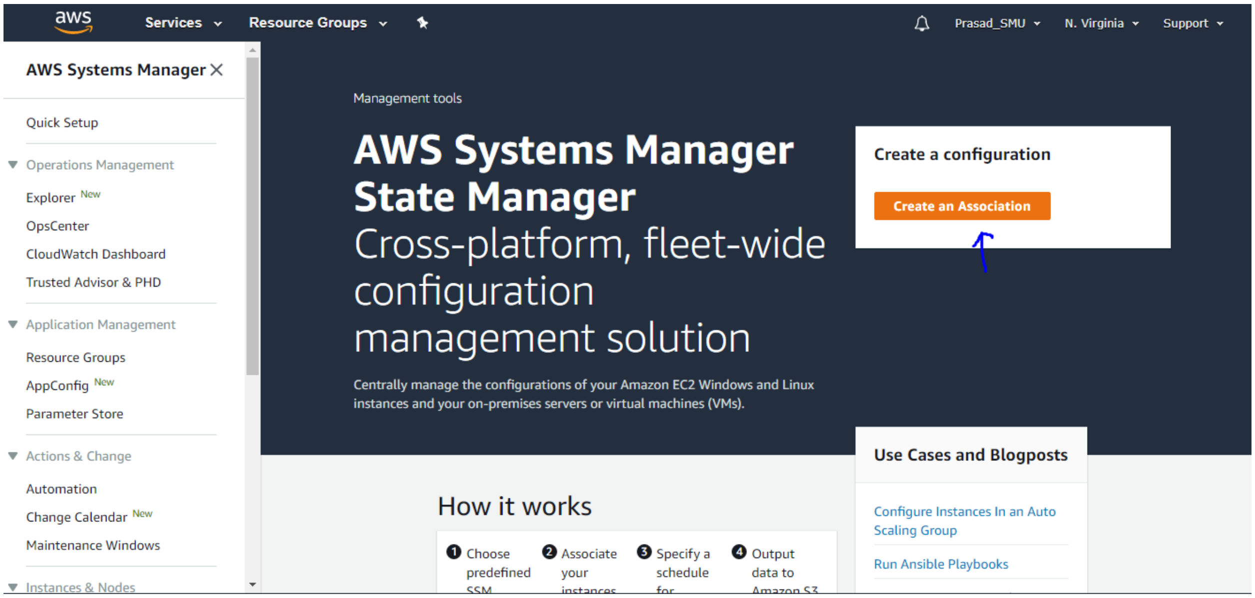
AWS Systems Manager

Gain Operational Insight and Take Action on AWS Resources.

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

Now click on **Create Association**.



Management tools

AWS Systems Manager State Manager

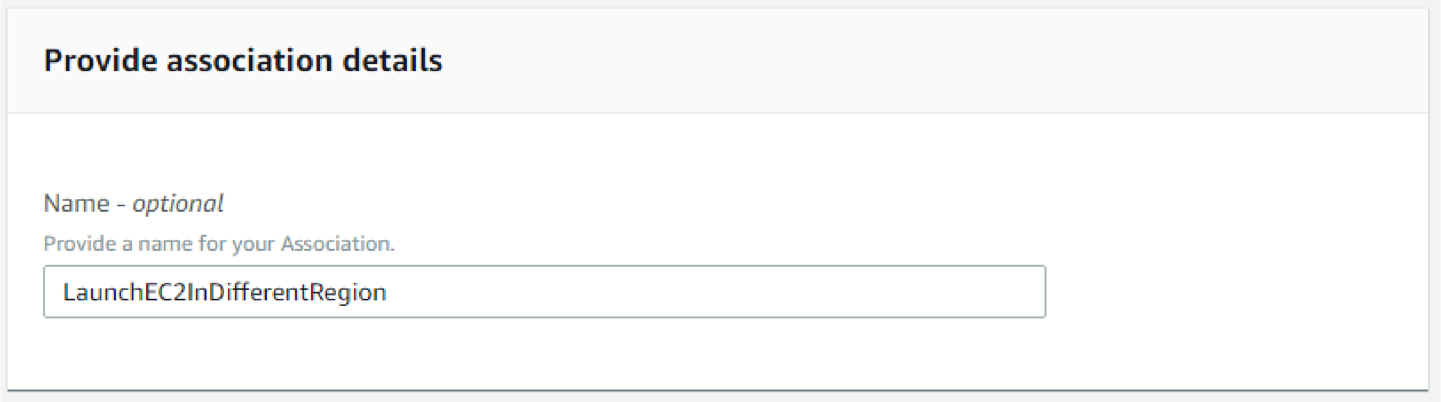
Cross-platform, fleet-wide configuration management solution

Centrally manage the configurations of your Amazon EC2 Windows and Linux instances and your on-premises servers or virtual machines (VMs).

How it works

1 Choose predefined SSM
2 Associate your instances
3 Specify a schedule for
4 Output data to Amazon S3

Specify the Association Name as per your choice.



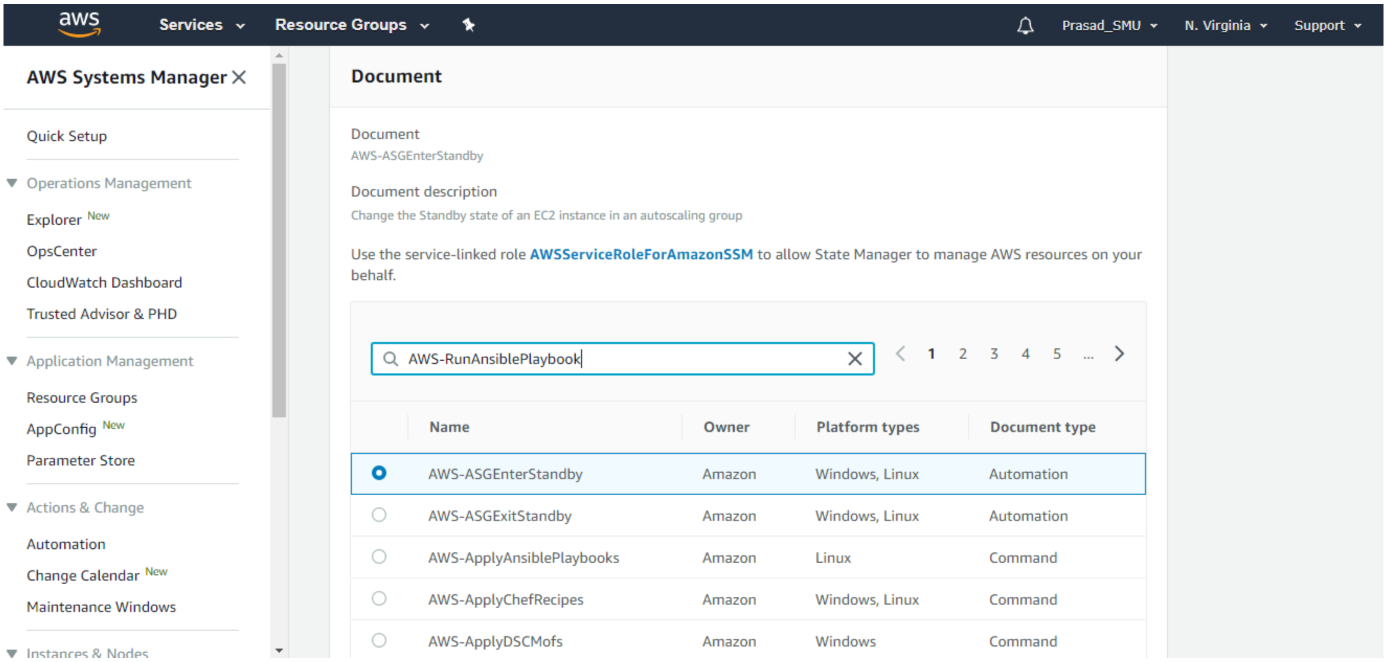
Provide association details

Name - *optional*

Provide a name for your Association.

LaunchEC2InDifferentRegion

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



Document

Document
AWS-ASGEnterStandby

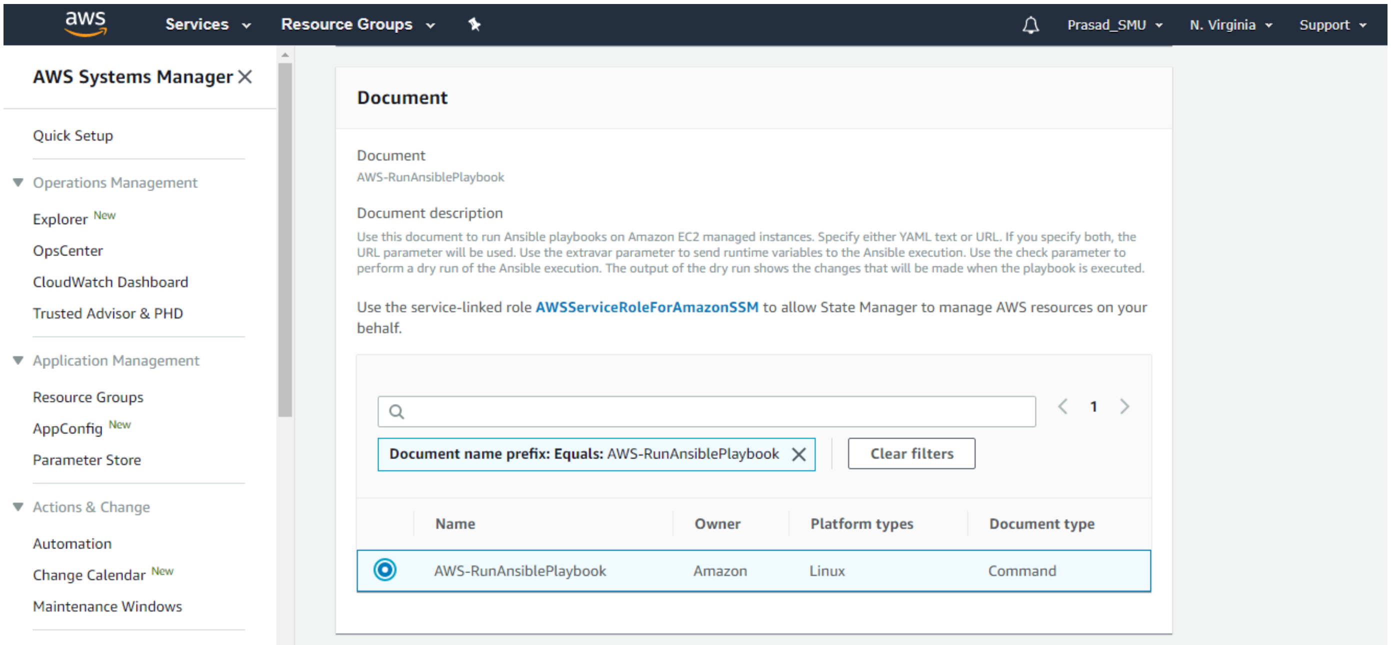
Document description
Change the Standby state of an EC2 instance in an autoscaling group

Use the service-linked role **AWSServiceRoleForAmazonSSM** to allow State Manager to manage AWS resources on your behalf.

AWS-RunAnsiblePlaybook

	Name	Owner	Platform types	Document type
●	AWS-ASGEnterStandby	Amazon	Windows, Linux	Automation
○	AWS-ASGExitStandby	Amazon	Windows, Linux	Automation
○	AWS-ApplyAnsiblePlaybooks	Amazon	Linux	Command
○	AWS-ApplyChefRecipes	Amazon	Windows, Linux	Command
○	AWS-ApplyDSCMofs	Amazon	Windows	Command

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



Document

Document
AWS-RunAnsiblePlaybook

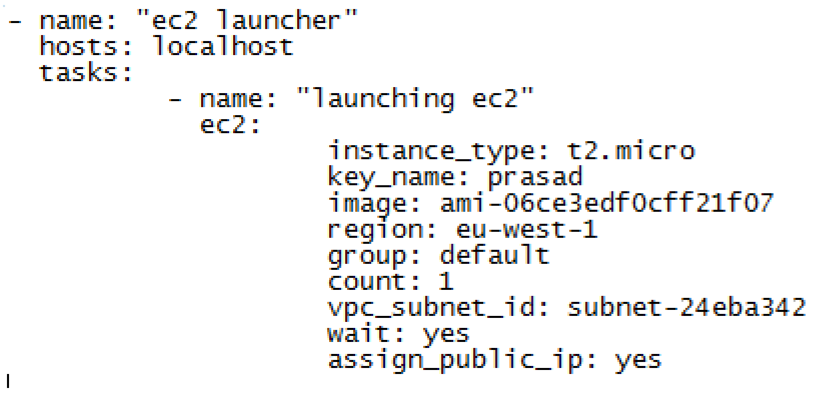
Document description
Use this document to run Ansible playbooks on Amazon EC2 managed instances. Specify either YAML text or URL. If you specify both, the URL parameter will be used. Use the extravar parameter to send runtime variables to the Ansible execution. Use the check parameter to perform a dry run of the Ansible execution. The output of the dry run shows the changes that will be made when the playbook is executed.

Use the service-linked role **AWSServiceRoleForAmazonSSM** to allow State Manager to manage AWS resources on your behalf.

Document name prefix: Equals: AWS-RunAnsiblePlaybook

	Name	Owner	Platform types	Document type
●	AWS-RunAnsiblePlaybook	Amazon	Linux	Command

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



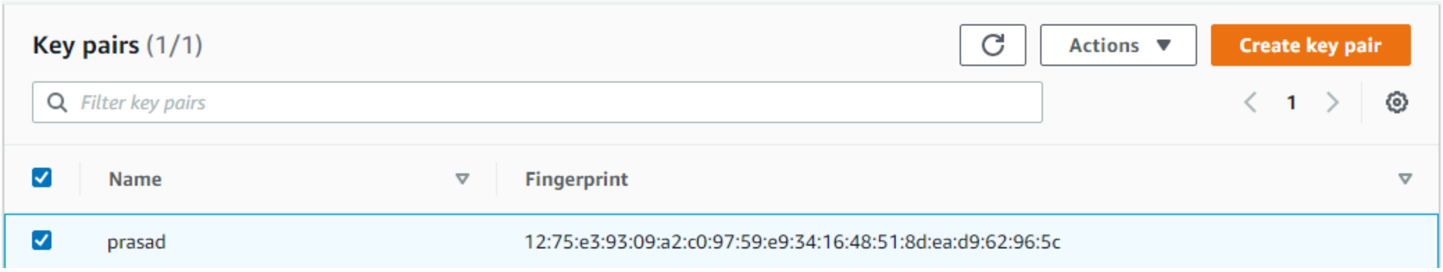
```
- name: "ec2 launcher"
  hosts: localhost
  tasks:
          - name: "launching ec2"
            ec2:
                    instance_type: t2.micro
                    key_name: prasad
                    image: ami-06ce3edf0cff21f07
                    region: eu-west-1
                    group: default
                    count: 1
                    vpc_subnet_id: subnet-24eba342
                    wait: yes
                    assign_public_ip: yes
```

Question: How to get key_name, image, group, vpc_subnet_id parameters?

If you notice, we are currently running our Playbook in N. Virginia (US-EAST-1) region and our Ansible Playbook is going to deploy EC2 Instance in Ireland (EU-WEST-1) region. You need to specify parameters such as key_name, image, group, vpc_subnet_id of the Ireland (EU-WEST-1) region.

key_name:

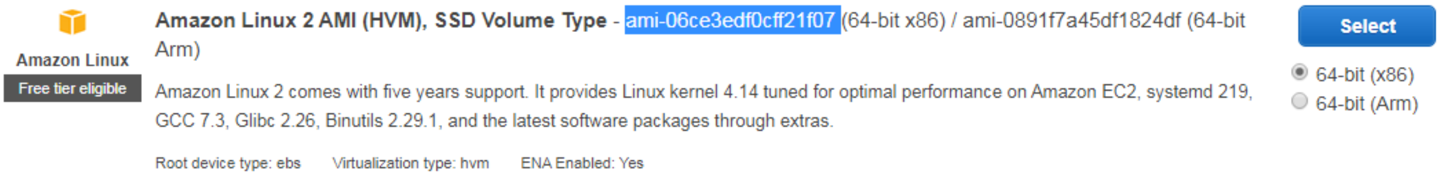
I've created few Key Pairs in Ireland (EU-WEST-1) region. You can find the Key Pair information on the EC2 Service Dashboard. You can also create Key Pair if you wish. I've selected LinuxServer Key Pair. You can create new Key Pair In-case if you don't have it.



☑	Name	Fingerprint
☑	prasad	12:75:e3:93:09:a2:c0:97:59:e9:34:16:48:51:8d:ea:d9:62:96:5c

Image:

AMI Image information can be found while launching an EC2 Instance.



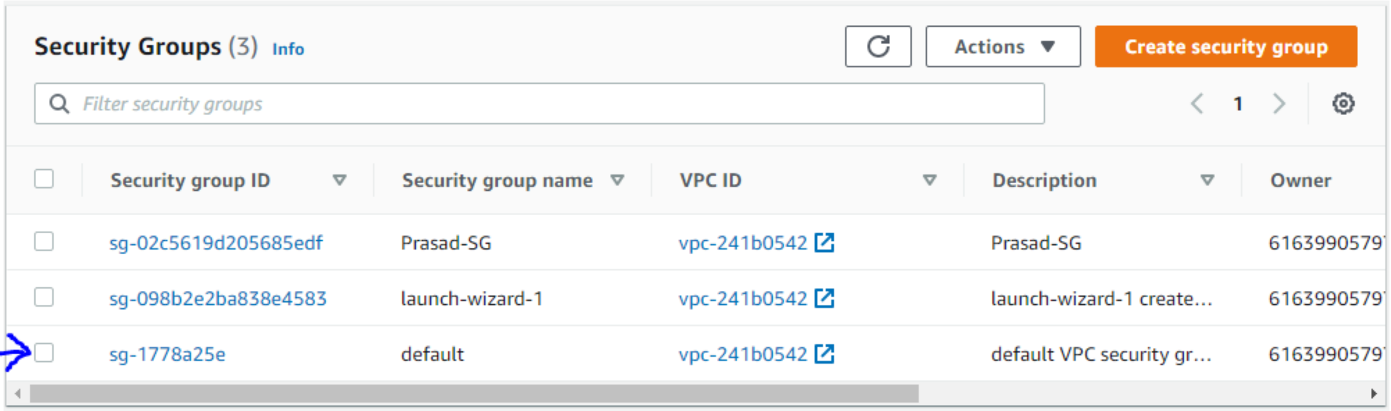
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-06ce3edf0cff21f07 (64-bit x86) / ami-0891f7a45df1824df (64-bit Arm)

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

Group:

Security Group information can be found on the EC2 Service Dashboard. Select the Security Group as per your choice. I've selected the Default Security Group.

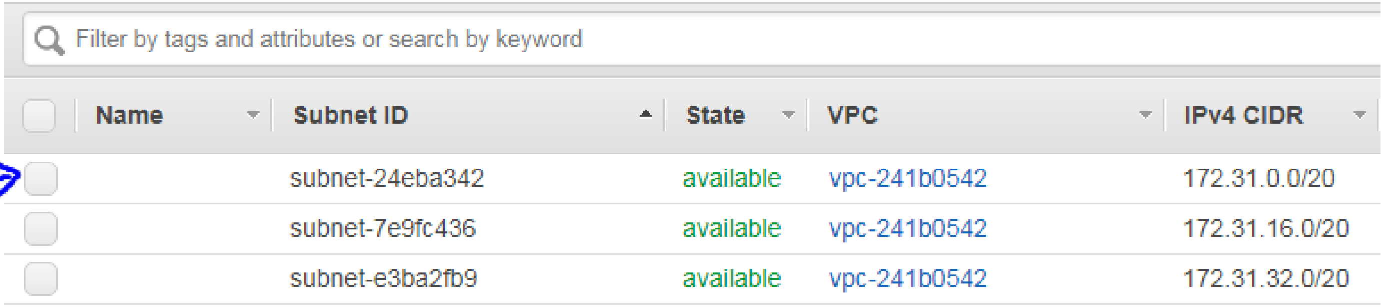


□	Security group ID	Security group name	VPC ID	Description	Owner
□	sg-02c5619d205685edf	Prasad-SG	vpc-241b0542	Prasad-SG	6163990579
□	sg-098b2e2ba838e4583	launch-wizard-1	vpc-241b0542	launch-wizard-1 create...	6163990579
□	sg-1778a25e	default	vpc-241b0542	default VPC security gr...	6163990579

vpc_subnet_id:

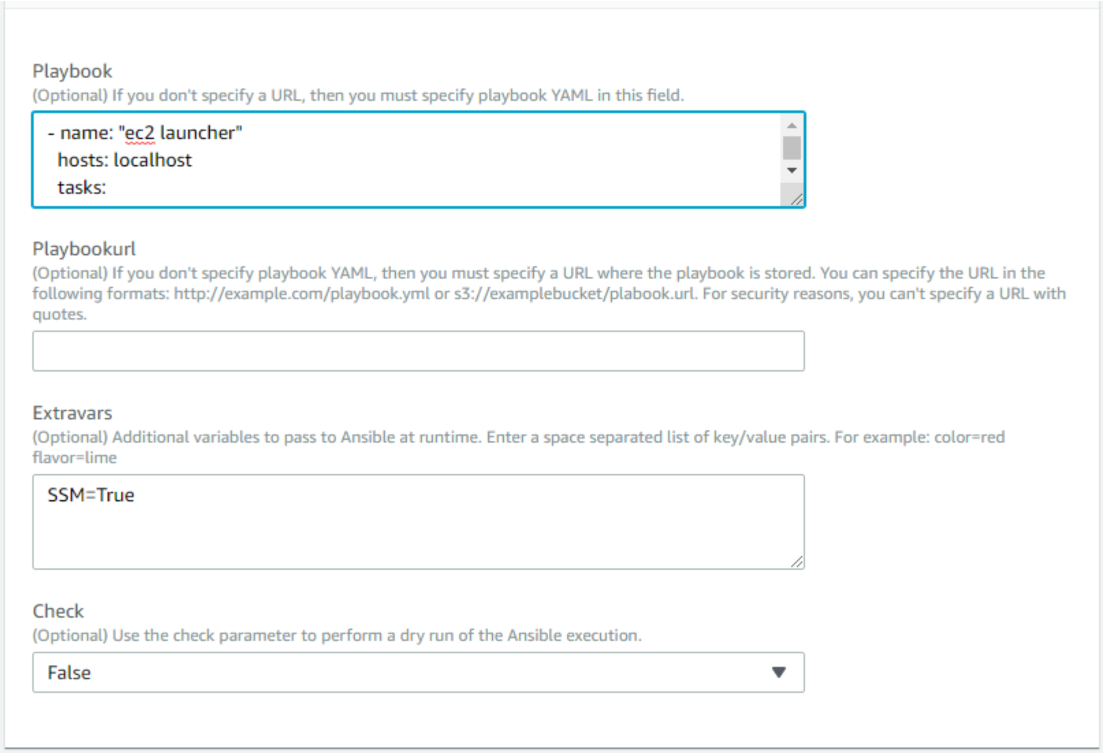
Navigate to VPC Service and click on Subnet.

Copy the Subnet ID in which you want to deploy an EC2 Instance. I've selected the Default Subnet of Default VPC.



□	Name	Subnet ID	State	VPC	IPv4 CIDR
□		subnet-24eba342	available	vpc-241b0542	172.31.0.0/20
□		subnet-7e9fc436	available	vpc-241b0542	172.31.16.0/20
□		subnet-e3ba2fb9	available	vpc-241b0542	172.31.32.0/20

Keep in mind that, these parameters key_name, image, group, vpc_subnet_id changes per Region. If you want to deploy EC2 Instance in another Region, make sure to associate parameters of that particular Region.



Playbook

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

```
- name: "ec2 launcher"
  hosts: localhost
  tasks:
```

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

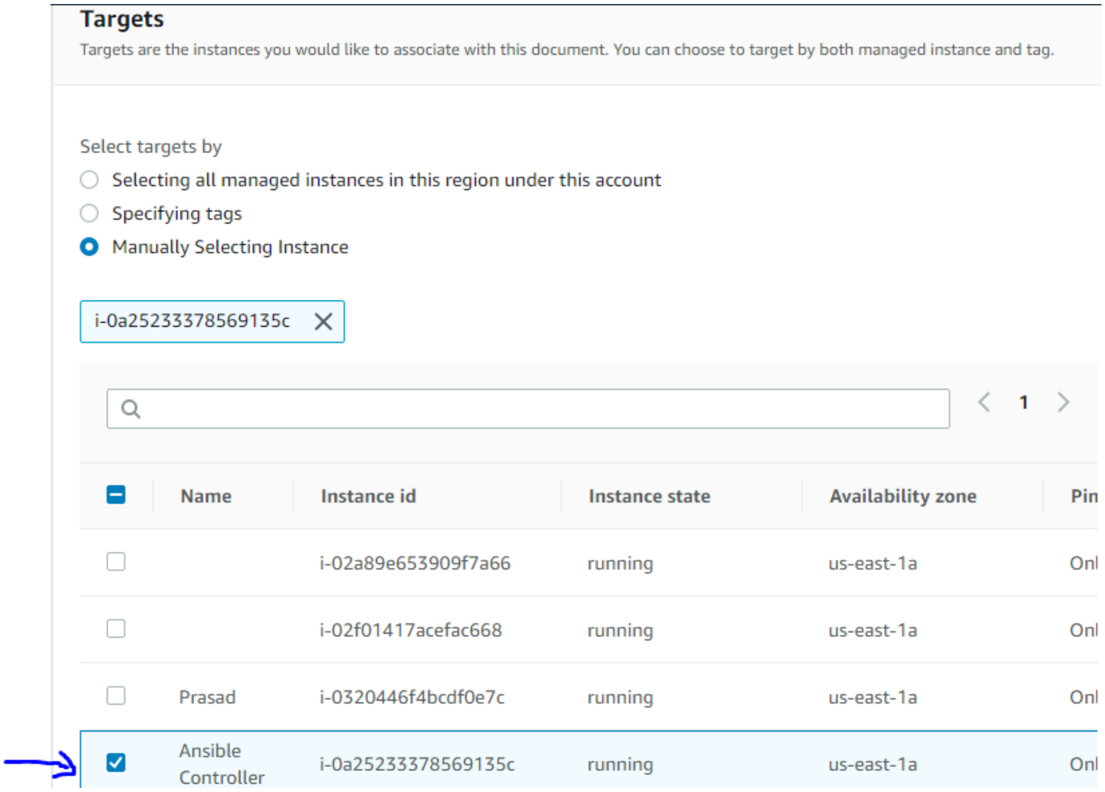
Check

(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

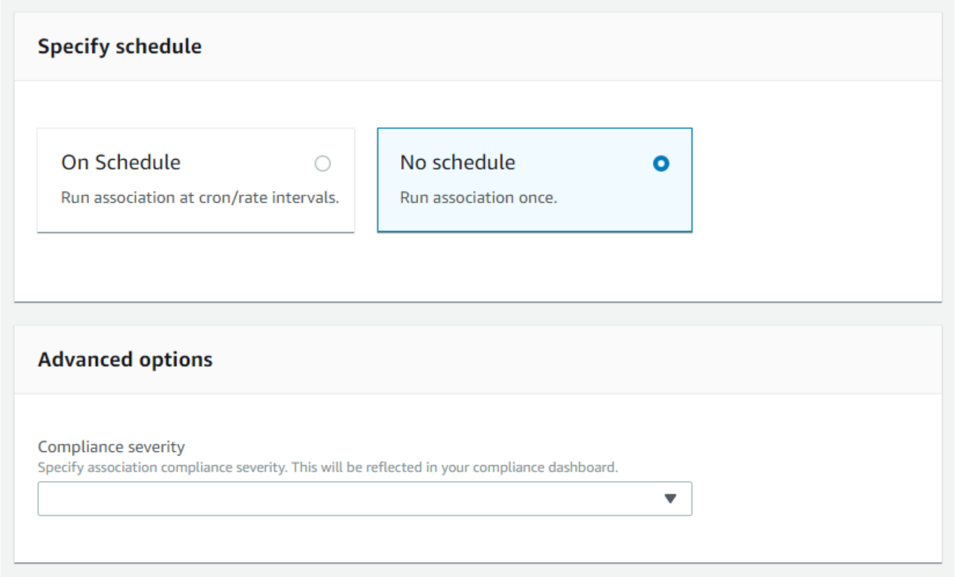
○ Specifying tags

● Manually Selecting Instance

i-0a25233378569135c ✕

⊟	Name	Instance id	Instance state	Availability zone	Pin
☐		i-02a89e653909f7a66	running	us-east-1a	Onl
☐		i-02f01417acefac668	running	us-east-1a	Onl
☐	Prasad	i-0320446f4bcdf0e7c	running	us-east-1a	Onl
☑	Ansible Controller	i-0a25233378569135c	running	us-east-1a	Onl

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



Specify schedule

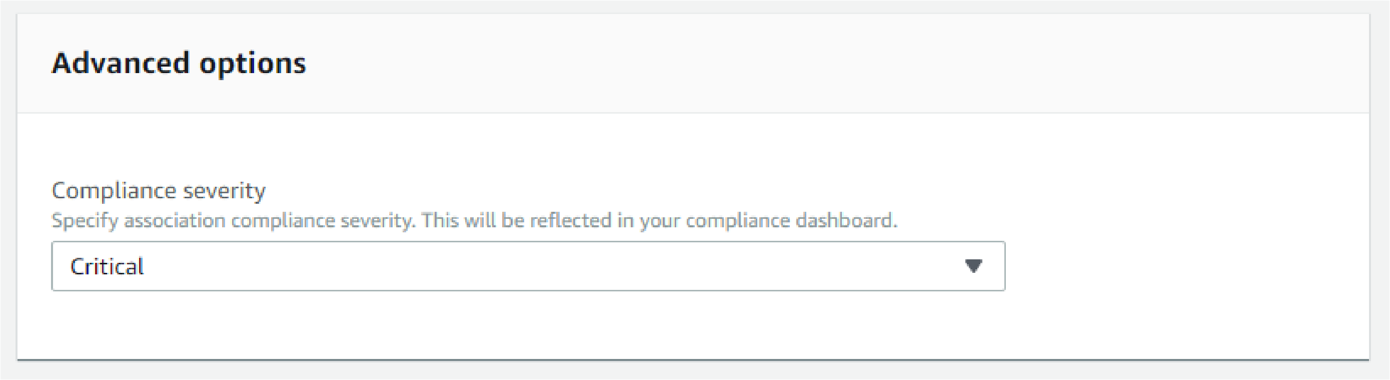
○ On Schedule
Run association at cron/rate intervals.

● 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.



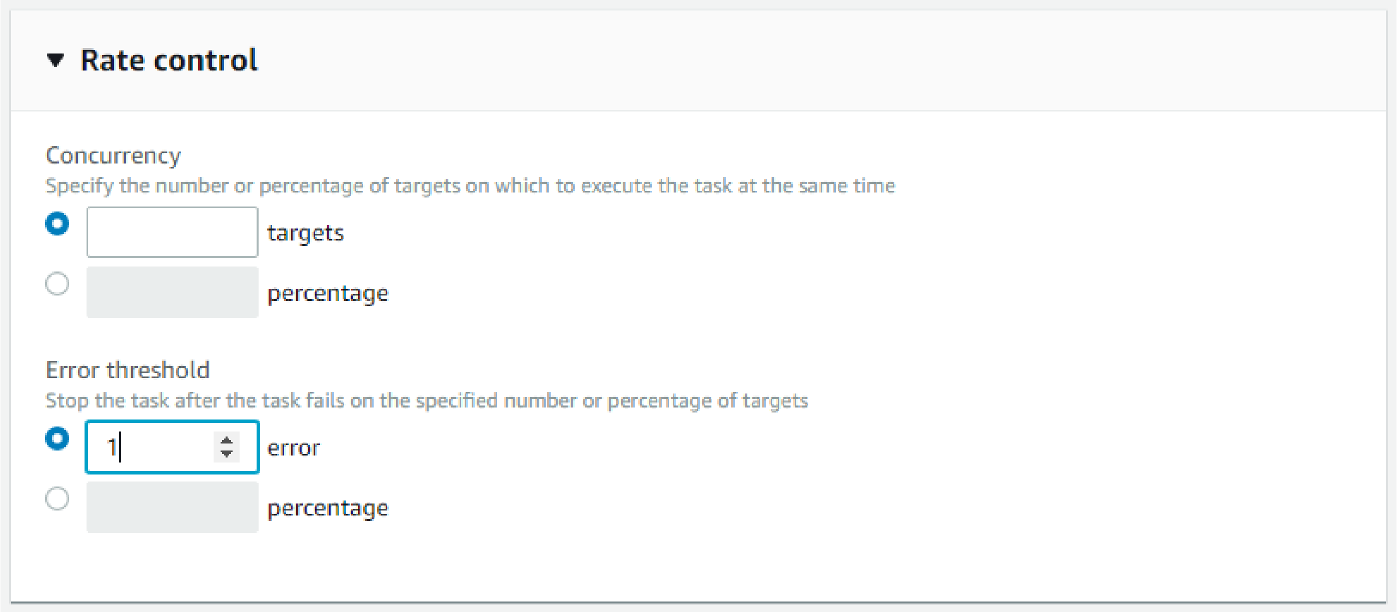
Advanced options

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

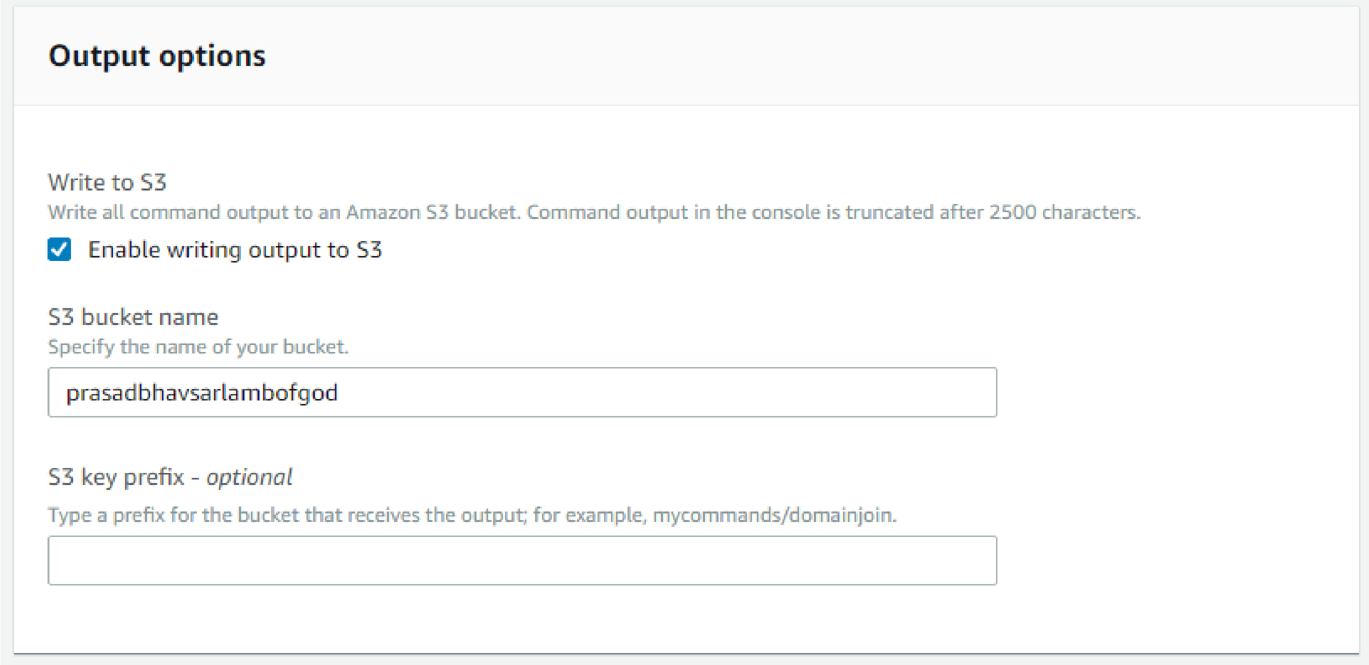
● 1 error

○ 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.

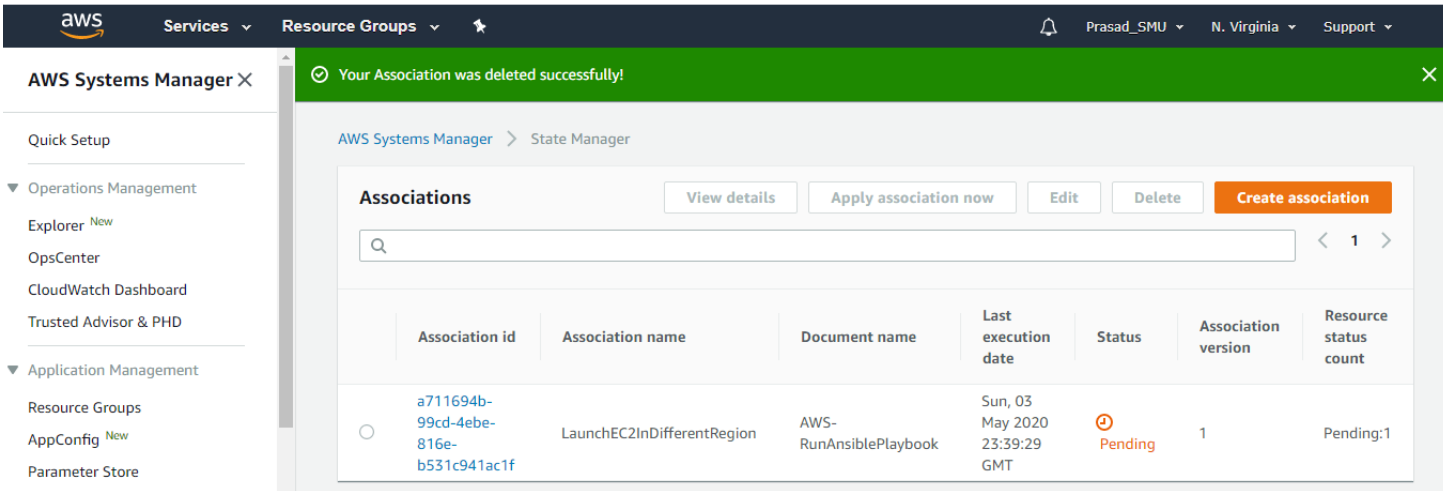
prasadbhavsarlambofgod

S3 key prefix - *optional*

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

Task 10: Verify Ansible Playbook Execution

The association status is currently Pending.

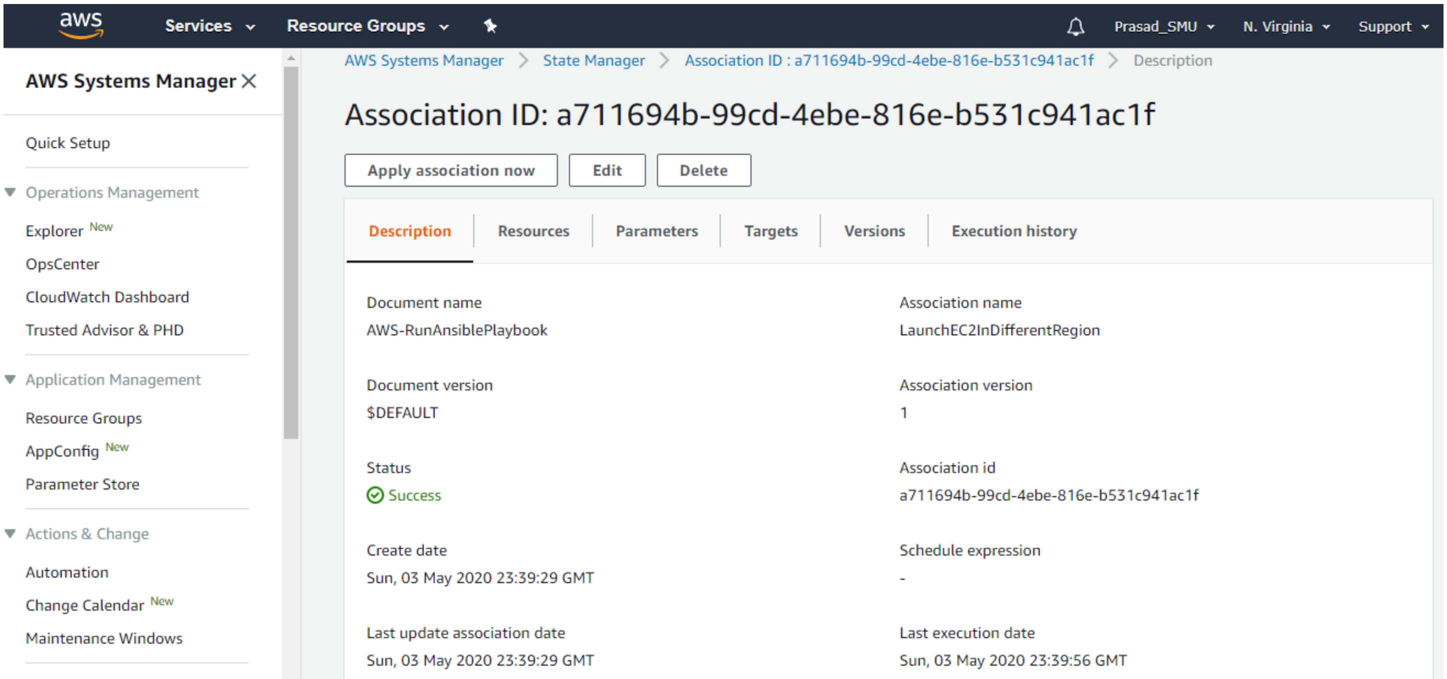


Your Association was deleted successfully!

Associations

	Association id	Association name	Document name	Last execution date	Status	Association version	Resource status count
○	a711694b-99cd-4ebe-816e-b531c941ac1f	LaunchEC2InDifferentRegion	AWS-RunAnsiblePlaybook	Sun, 03 May 2020 23:39:29 GMT	Pending	1	Pending:1

The association status is now Success.



Association ID: a711694b-99cd-4ebe-816e-b531c941ac1f

Document name
AWS-RunAnsiblePlaybook

Association name
LaunchEC2InDifferentRegion

Document version
$DEFAULT

Association version
1

Status
Success

Association id
a711694b-99cd-4ebe-816e-b531c941ac1f

Create date
Sun, 03 May 2020 23:39:29 GMT

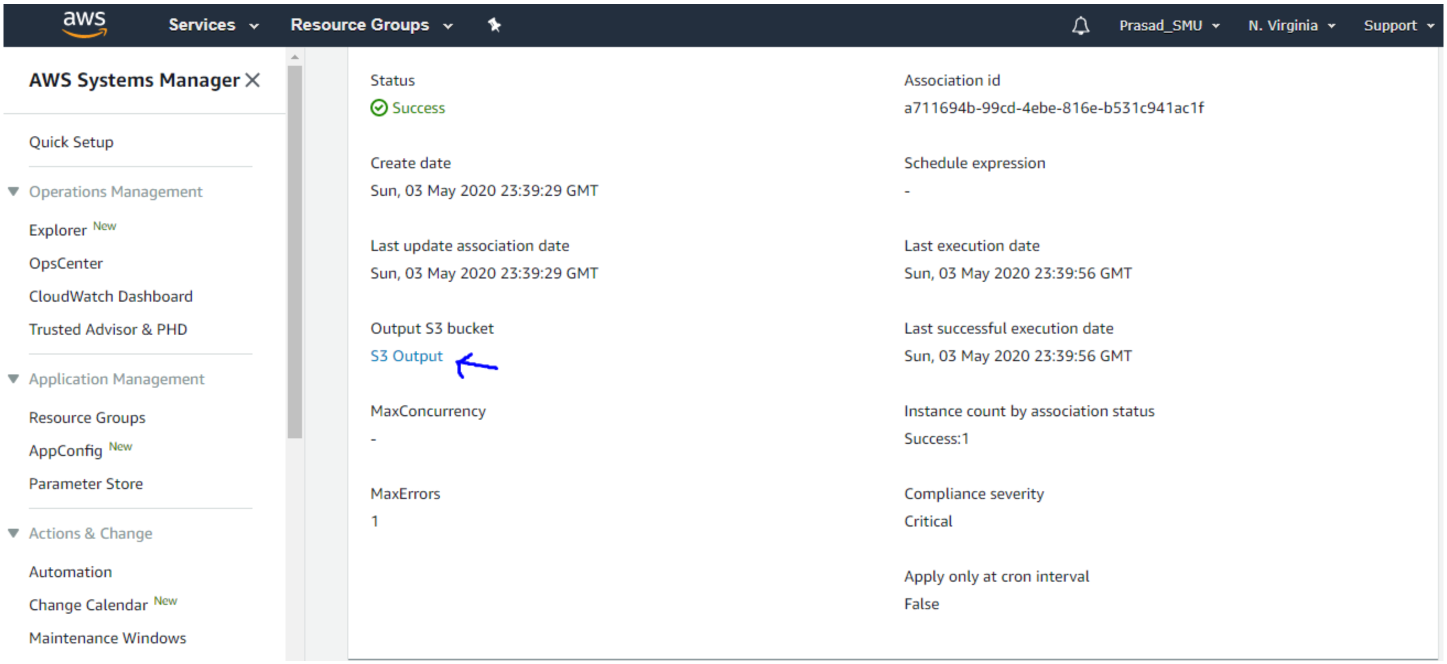
Schedule expression
-

Last update association date
Sun, 03 May 2020 23:39:29 GMT

Last execution date
Sun, 03 May 2020 23:39:56 GMT

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

Click on your S3 Bucket.



Status
Success

Association id
a711694b-99cd-4ebe-816e-b531c941ac1f

Create date
Sun, 03 May 2020 23:39:29 GMT

Schedule expression
-

Last update association date
Sun, 03 May 2020 23:39:29 GMT

Last execution date
Sun, 03 May 2020 23:39:56 GMT

Output S3 bucket
S3 Output

Last successful execution date
Sun, 03 May 2020 23:39:56 GMT

MaxConcurrency
-

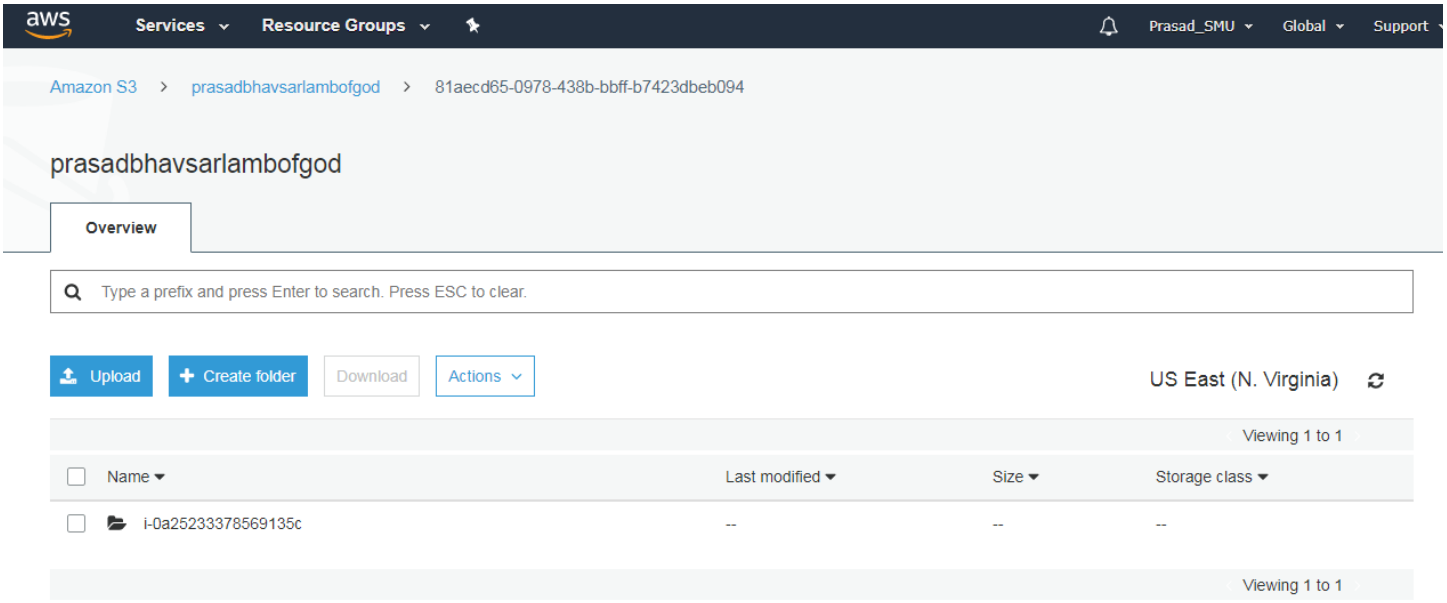
Instance count by association status
Success:1

MaxErrors
1

Compliance severity
Critical

Apply only at cron interval
False

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



Amazon S3 > prasadbhavsarlambofgod > 81aecd65-0978-438b-bbff-b7423dbeb094

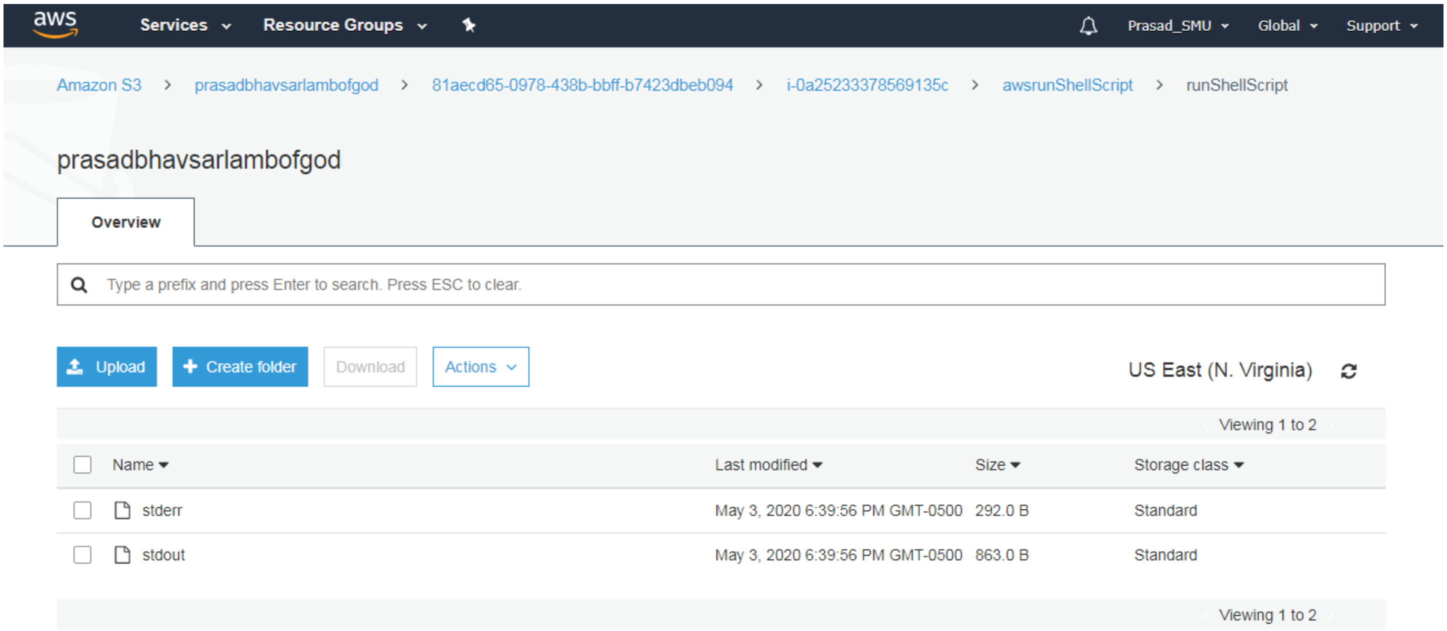
prasadbhavsarlambofgod

Viewing 1 to 1

	Name	Last modified	Size	Storage class
□	i-0a25233378569135c	--	--	--

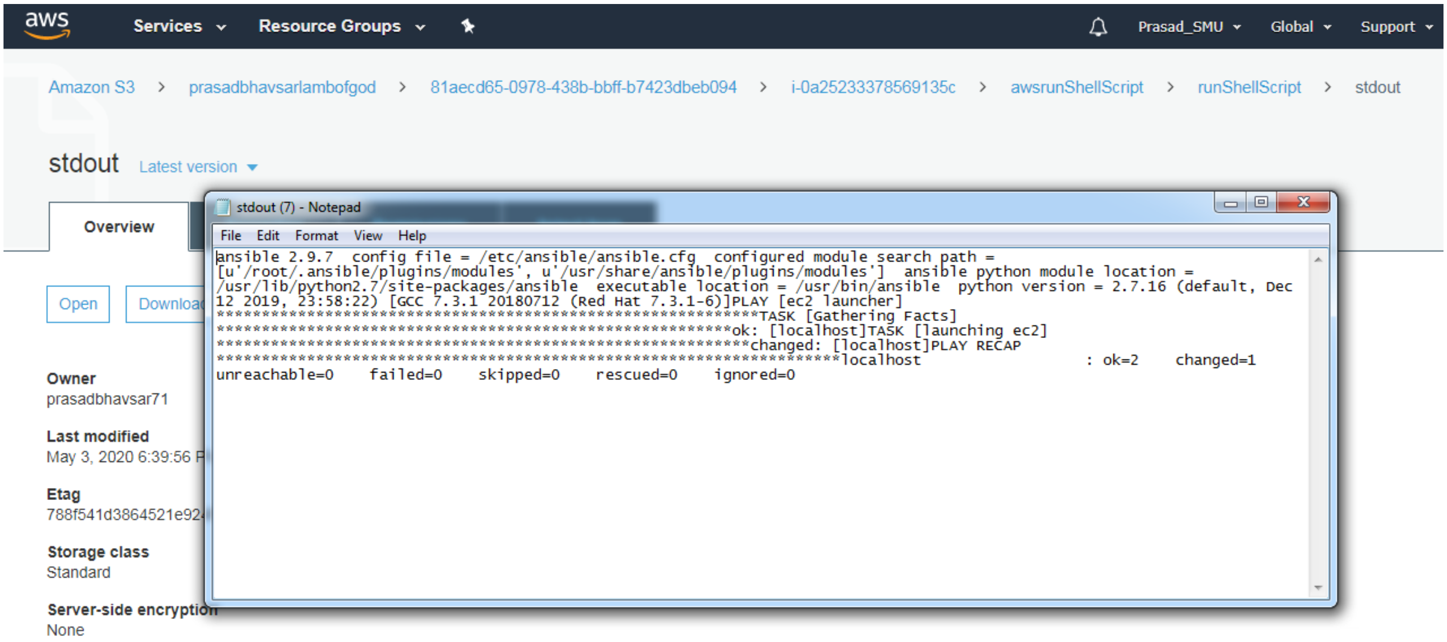
Viewing 1 to 1

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


aws
Services
Resource Groups
Prasad_SMU
Global
Support
Amazon S3 > prasadbhavsarlambofgod > 81aecd65-0978-438b-bbff-b7423dbeb094 > i-0a25233378569135c > awsrunShellScript > runShellScript
prasadbhavsarlambofgod
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US East (N. Virginia)
Viewing 1 to 2

	Name	Last modified	Size	Storage class
□	stderr	May 3, 2020 6:39:56 PM GMT-0500	292.0 B	Standard
□	stdout	May 3, 2020 6:39:56 PM GMT-0500	863.0 B	Standard

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


aws
Services
Resource Groups
Prasad_SMU
Global
Support
Amazon S3 > prasadbhavsarlambofgod > 81aecd65-0978-438b-bbff-b7423dbeb094 > i-0a25233378569135c > awsrunShellScript > runShellScript > stdout
stdout
Latest version
Overview
Open
Downloa
stdout (7) - Notepad
File
Edit
Format
View
Help

Owner
prasadbhavsar71

Last modified
May 3, 2020 6:39:56 P

Etag
788f541d3864521e92

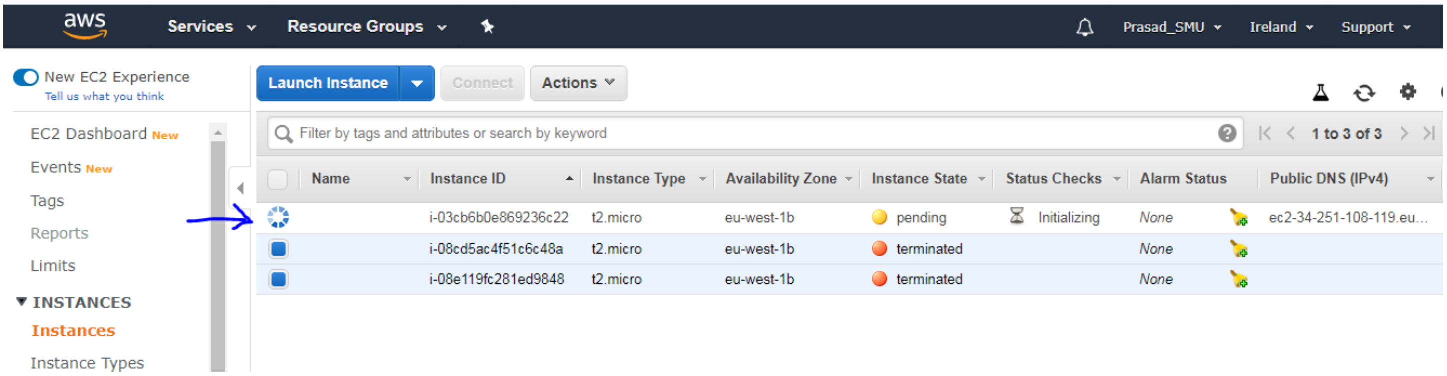
Storage class
Standard

Server-side encryption
None

```
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 [ec2 launcher]
************************************************************TASK [Gathering Facts]
*********************************************************ok: [localhost]TASK [launching ec2]
***********************************************************changed: [localhost]PLAY RECAP
*********************************************************************localhost                  : ok=2    changed=1
unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

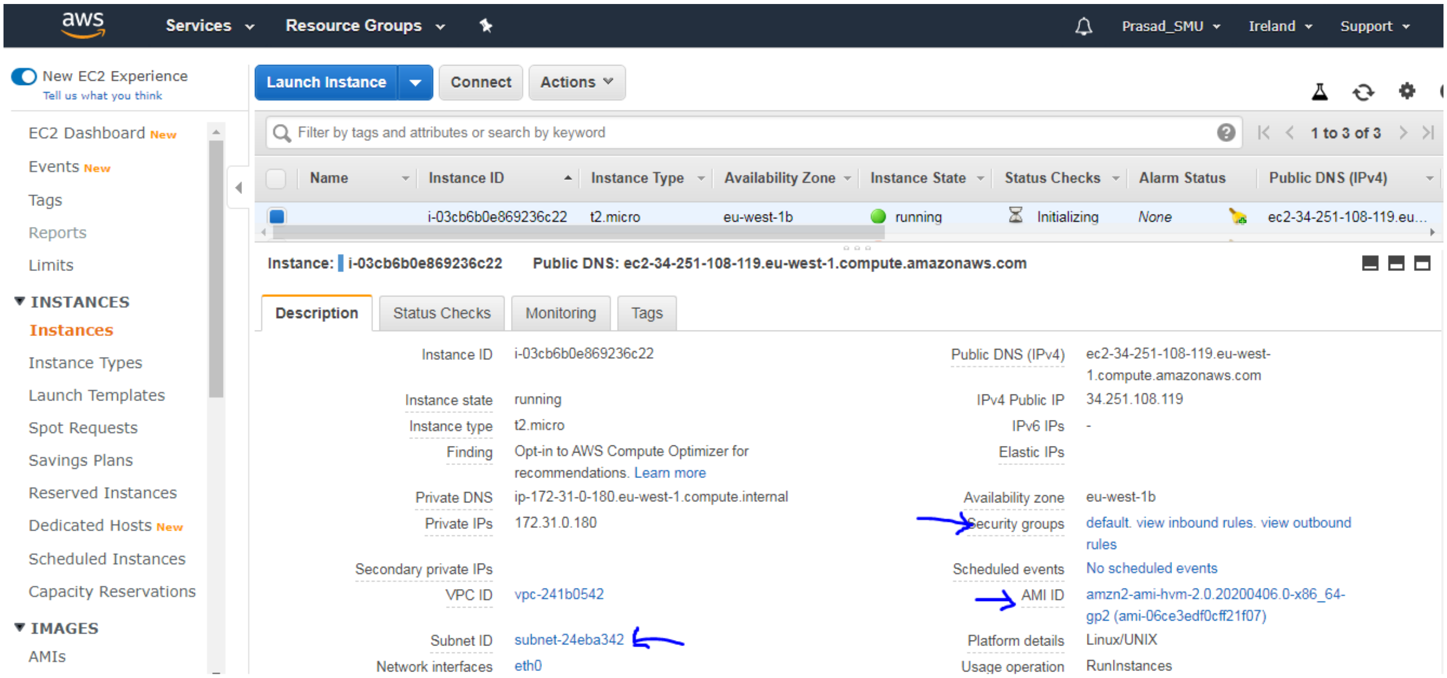
Navigate to Ireland (eu-west-1) region and go to EC2 Service.

You'll notice a new EC2 Instance has been launched.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-03cb6b0e869236c22	t2.micro	eu-west-1b	pending	Initializing	None	ec2-34-251-108-119.eu...
	i-08cd5ac4f51c6c48a	t2.micro	eu-west-1b	terminated		None	
	i-08e119fc281ed9848	t2.micro	eu-west-1b	terminated		None	

You can now verify its parameters such as key_name, image, group, vpc_subnet_id which we defined in our Playbook.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-03cb6b0e869236c22	t2.micro	eu-west-1b	running	Initializing	None	ec2-34-251-108-119.eu...

Instance: i-03cb6b0e869236c22 Public DNS: ec2-34-251-108-119.eu-west-1.compute.amazonaws.com

Instance ID	i-03cb6b0e869236c22	Public DNS (IPv4)	ec2-34-251-108-119.eu-west-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	34.251.108.119
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-172-31-0-180.eu-west-1.compute.internal	Availability zone	eu-west-1b
Private IPs	172.31.0.180	Security groups	default. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-241b0542	AMI ID	amzn2-ami-hvm-2.0.20200406.0-x86_64-gp2 (ami-06ce3edf0cff21f07)
Subnet ID	subnet-24eba342	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances

This completes the Lab-Lunch EC2 Instance across Region by Running Ansible Playbook on AWS Systems Manager

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