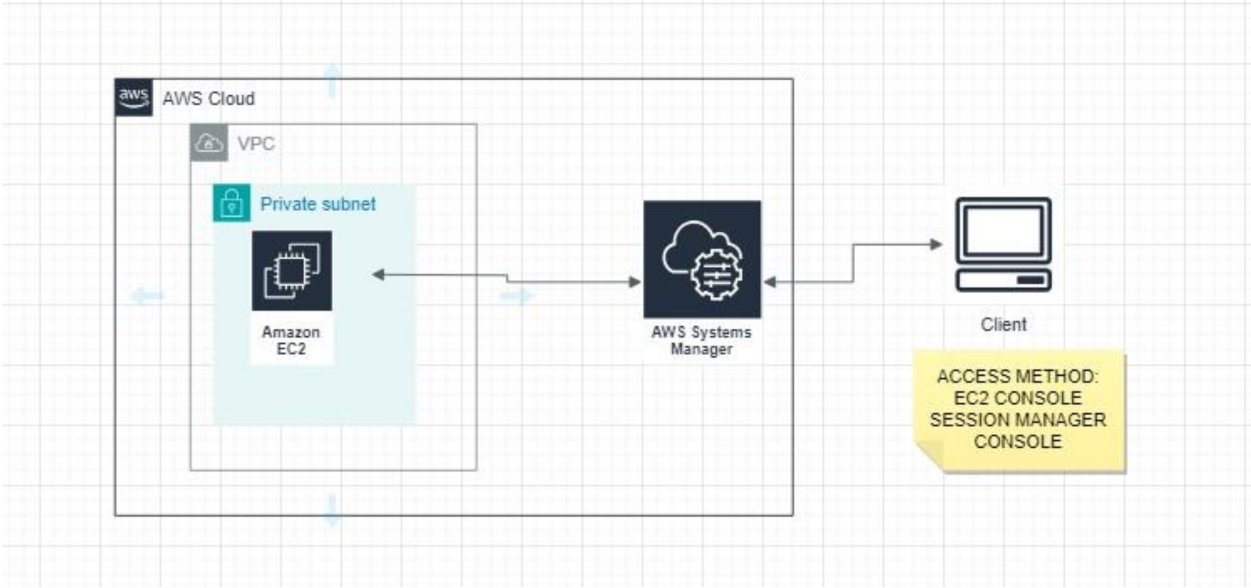
PROJECT 1

Remotely Run Commands on an EC2 Instance with AWS Systems Manager

Architecture diagram



In this project we will learn how to use AWS Systems Manager to remotely run commands on our Amazon EC2 instances. Systems Manager is a management tool that enables us to gain operational insights and take action on AWS resources safely and at scale. Using the run command, one of the automation features of Systems Manager, we can simplify management tasks by eliminating the need to use bastion hosts, SSH, or remote PowerShell.

as a System Administrator, we need to update the packages on our EC2 instances. To complicate this normally simple admin task, our security team does not allow you to direct access production servers via SSH or allow you to use bastion hosts. Fortunately, you can use Systems Manager to remotely run commands, like update packages, on your EC2 instances.

To solve this challenging scenario, you will create an Identity and Access Management (IAM) role, enable an agent on your instance that communicates with Systems Manager, then follow best practices by running the AWS-UpdateSSMAgent document to upgrade your Systems Manager Agent, and finally use Systems Manager to run a command on your instance.

AWS Systems Manager

AWS Systems Manager is a management service that helps you automatically collect software inventory, apply patches, create system images, and configure Windows and Linux operating systems. It provides a unified user interface so you can view operational data from multiple AWS services and automate operational tasks across your AWS resources. Systems Manager simplifies resource and application management, shortens the time to detect and resolve operational problems, and makes it easy to operate and manage your infrastructure securely at scale.

IAM roles

IAM roles in AWS are used to delegate access to AWS resources securely. Here are some common use cases for IAM roles.

EC2 intance

When you launch an EC2 instance, you can specify an IAM role to be associated with the instance. The IAM role determines the permissions that the EC2 instance has. You can change the IAM role associated with an instance while the instance is running, but you can't remove the IAM role from an instance once it has been assigned.

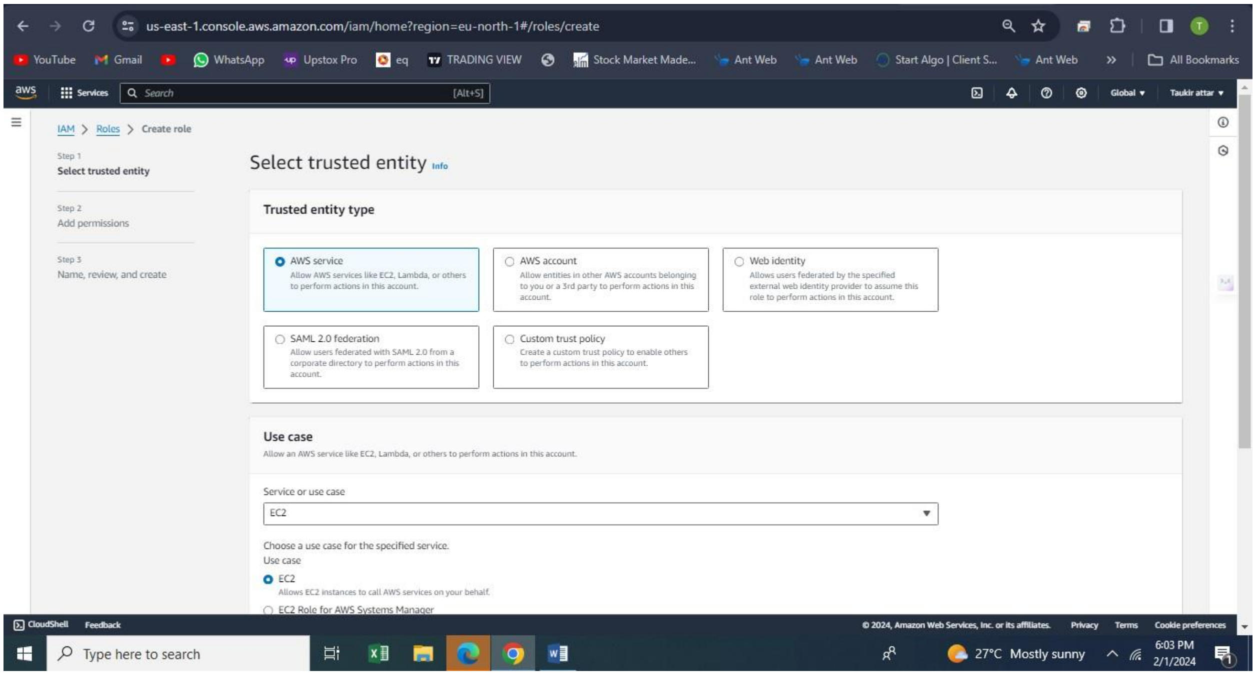
Let's Start Creating Project

- Create an Identity and Access Management (IAM) role
- In this step, we will create an IAM role that will be used to give Systems Manager permission to perform actions on our instances.

Open the IAM console

In the left navigation pane, choose Roles, and then choose Create role

On the Select trusted entity page, under AWS Service, choose EC2, and then choose Next.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

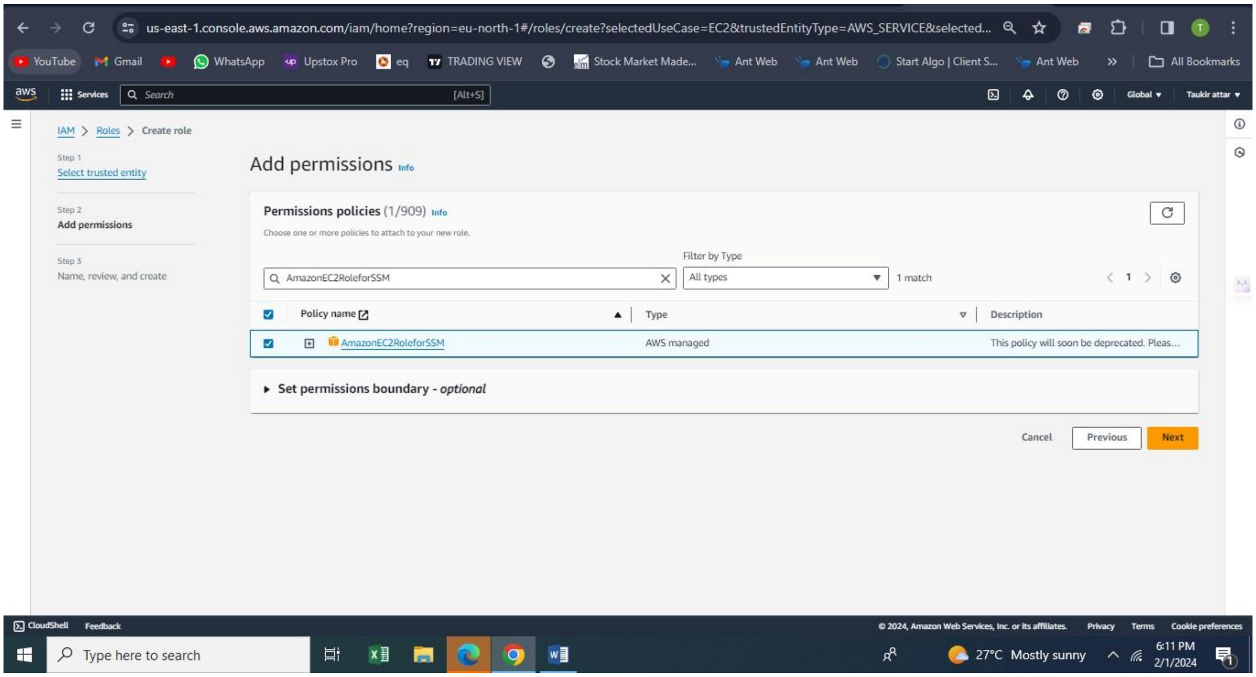
Service or use case
EC2

Choose a use case for the specified service.
Use case

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

○ EC2 Role for AWS Systems Manager

- On the Add permissions page, in the search bar type AmazonEC2RoleforSSM. From the policy list select AmazonEC2RoleforSSM and then choose Next.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/909) Info
Choose one or more policies to attach to your new role.

AmazonEC2RoleforSSM

Filter by Type
All types

1 match

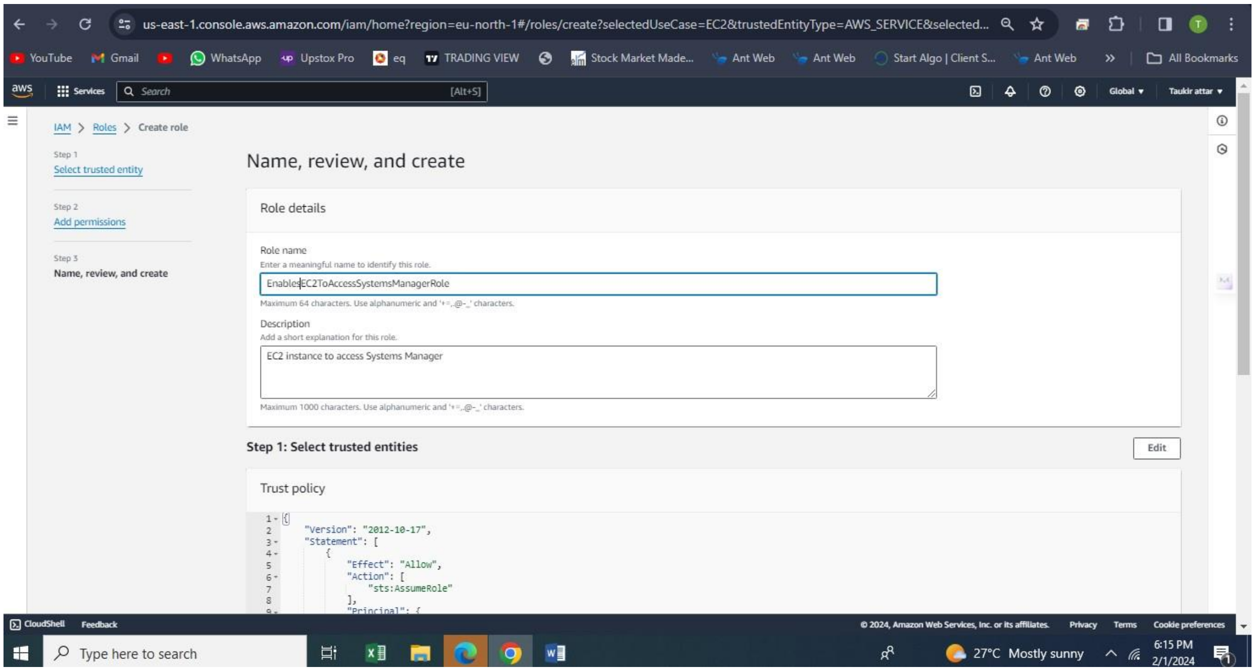
☑	Policy name	Type	Description
☑	AmazonEC2RoleforSSM	AWS managed	This policy will soon be deprecated. Pleas...

▸ Set permissions boundary - *optional*

Cancel Previous Next

- On the Name, review, and create page, in the Role name box, type in

EnablesEC2ToAccessSystemsManagerRole. In the Description box, type in Enables an EC2 instance to access Systems Manager. Choose Create role.



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

EnablesEC2ToAccessSystemsManagerRole

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

Description

Add a short explanation for this role.

EC2 instance to access Systems Manager

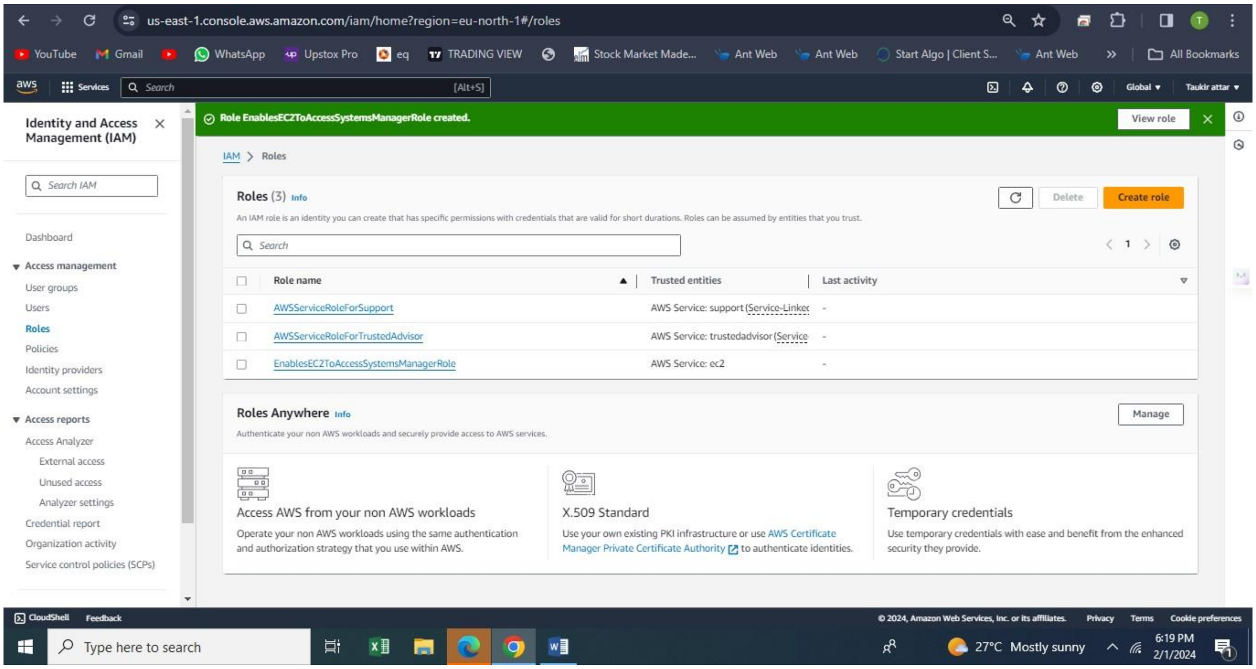
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
```

- Role is created in the below screenshot



Role EnablesEC2ToAccessSystemsManagerRole created.

Roles (3)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linkec	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	EnablesEC2ToAccessSystemsManagerRole	AWS Service: ec2	-

Roles Anywhere

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

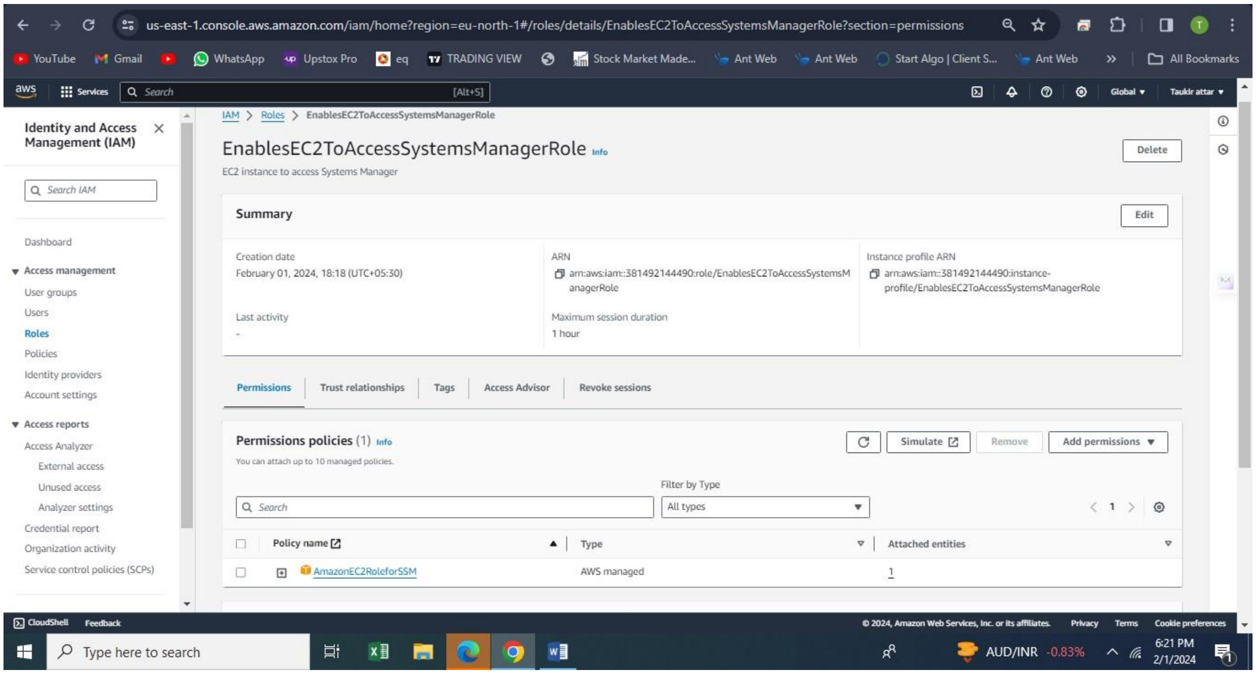
X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

In details I had explained role in below screenshot



us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/roles/details/EnablesEC2ToAccessSystemsManagerRole?section=permissions

IAM > Roles > EnablesEC2ToAccessSystemsManagerRole

EnablesEC2ToAccessSystemsManagerRole Info

EC2 instance to access Systems Manager

Summary

Creation date: February 01, 2024, 18:18 (UTC+05:30)

ARN: arn:aws:iam::381492144490:role/EnablesEC2ToAccessSystemsManagerRole

Instance profile ARN: arn:aws:iam::381492144490:instance-profile/EnablesEC2ToAccessSystemsManagerRole

Last activity: -

Maximum session duration: 1 hour

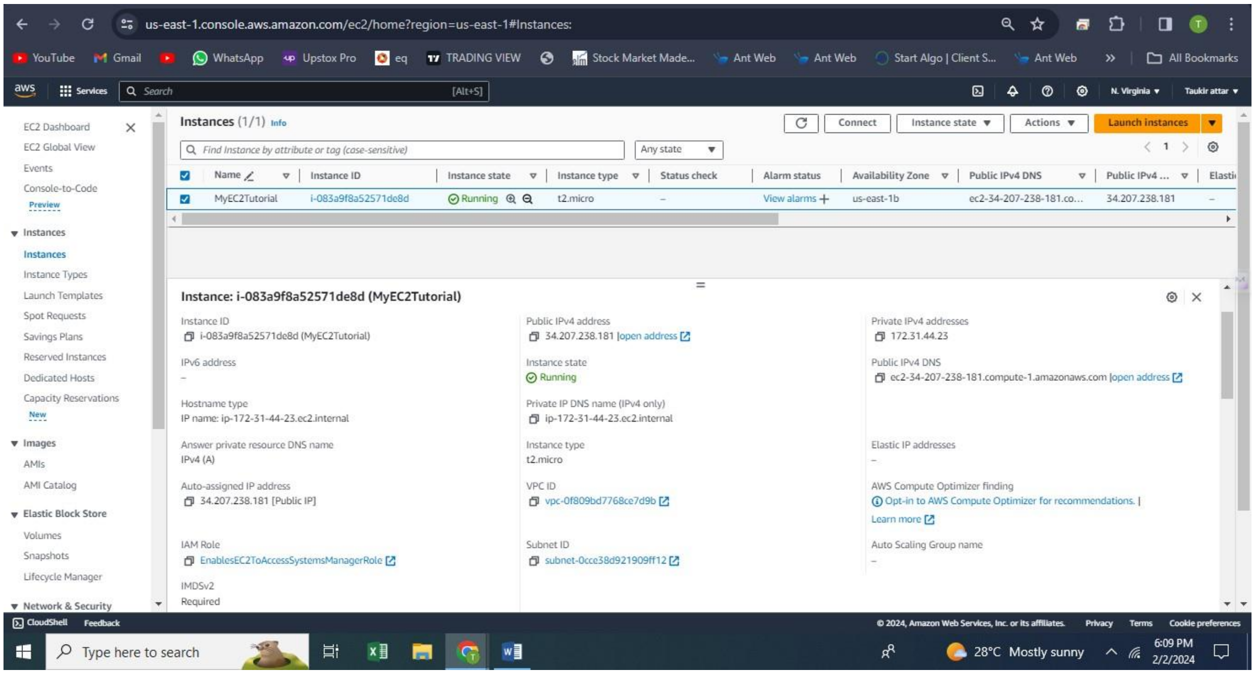
Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type: All types

	Policy name	Type	Attached entities
□	AmazonEC2RoleforSSM	AWS managed	1

- Now we are creating ec2 instance
- From the EC2 console, select our preferred Region.
- In the Name field, I have entered MyEC2Tutorial. Selected the Amazon Linux AMI. Retain the default selection that appears in the dropdown. We can also install the Systems Manager Agent on our own Windows or Linux system.
- Choose the t2.micro instance type.
- We will not need a keypair to use Systems Manager to remotely run commands. Scroll down to Key pair and under the Key pair name dropdown, choosed Proceed without a key pair.
- Under Advanced details, in the IAM instance profile dropdown choose the EnablesEC2ToAccessSystemsManagerRole role you created earlier. Leave everything else as default. Choose Launch instance.
- Ec2 is created in the below screen shot



Instances (1/1) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elasti
☑	MyEC2Tutorial	i-083a9f8a52571de8d	Running	t2.micro	–	View alarms +	us-east-1b	ec2-34-207-238-181.co...	34.207.238.181	–

Instance: i-083a9f8a52571de8d (MyEC2Tutorial)

Instance ID
i-083a9f8a52571de8d (MyEC2Tutorial)

IPv6 address
–

Hostname type
IP name: ip-172-31-44-23.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
34.207.238.181 [Public IP]

IAM Role
EnablesEC2ToAccessSystemsManagerRole

IMDSv2
Required

Public IPv4 address
34.207.238.181 |open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-44-23.ec2.internal

Instance type
t2.micro

VPC ID
vpc-0f809bd7768ce7d9b

Subnet ID
subnet-0cce38d921909ff12

Private IPv4 addresses
172.31.44.23

Public IPv4 DNS
ec2-34-207-238-181.compute-1.amazonaws.com |open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

- Update the Systems Manager Agent

- Now that you have an EC2 instance running the Systems Manager agent, you can automate administration tasks and manage the instance. In this step, you run a prepackaged command, called a document, that will upgrade the agent. It is best practice to update the Systems Manager Agent when you create a new instance.

- In the top navigation bar, search for Systems Manager and open the Systems Manager console.

- Under the Node Management section on the left navigation bar, choose Fleet Manager.

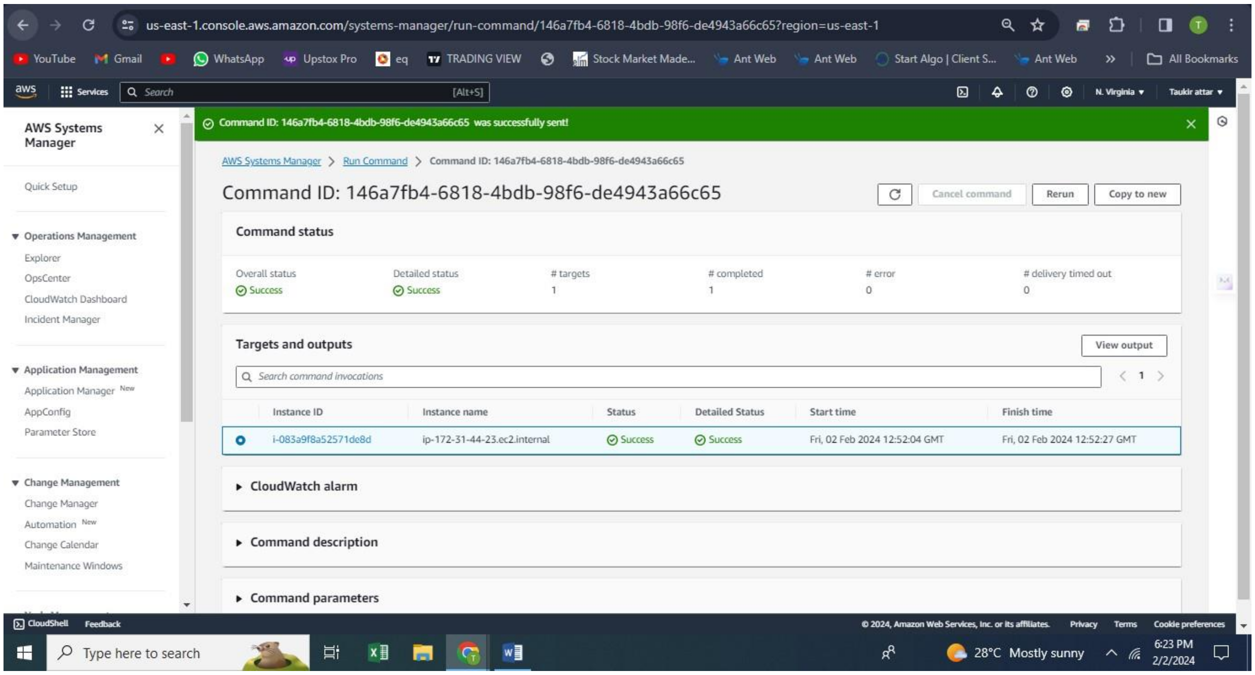
- Select the node ID created in step 2, MyEC2Tutorial, to open the node detail page.

- On the node detail page, in the Node actions dropdown, select Execute run command.

- On the Run a command page, click in the search bar and select, Document name prefix, then click on Equals, then type in AWS-UpdateSSMAgent.

- Scroll down to the Targets panel and select the check box next to your managed EC2 instance.

- Next you will see a page documenting your running command, and then overall success in green. Congrats, you have just run your first remote command using Systems Manager.



Command ID: 146a7fb4-6818-4bdb-98f6-de4943a66c65 was successfully sent!

Command ID: 146a7fb4-6818-4bdb-98f6-de4943a66c65

Command status

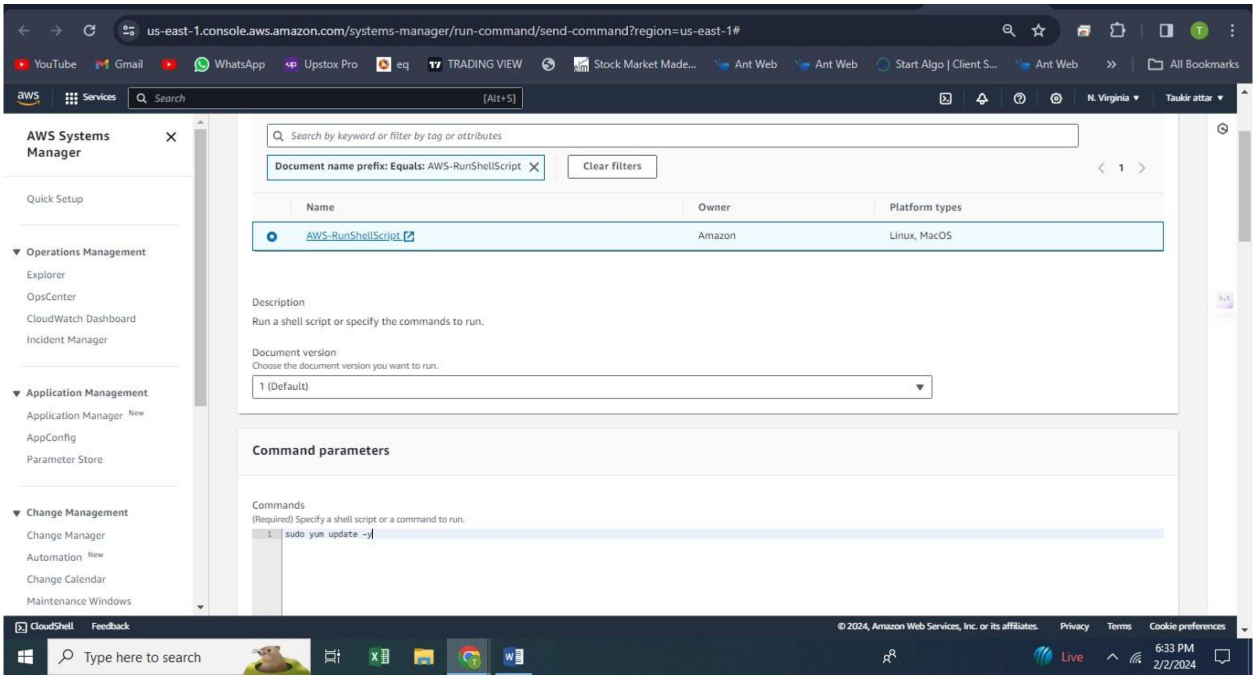
Overall status	Detailed status	# targets	# completed	# error	# delivery timed out
Success	Success	1	1	0	0

Targets and outputs

	Instance ID	Instance name	Status	Detailed Status	Start time	Finish time
●	i-083a9f8a52571de8d	ip-172-31-44-23.ec2.internal	Success	Success	Fri, 02 Feb 2024 12:52:04 GMT	Fri, 02 Feb 2024 12:52:27 GMT

Run remote shell script

- Now that your EC2 instance has the latest Systems Manager Agent, you can upgrade the packages on the EC2 instance. In this step, you will run a shell script through Run Command.
- Under the Node Management section on the left navigation bar, choose Fleet Manager.
- Select the node ID created in step 2, MyEC2Tutorial, to open the node detail page.
- On the node detail page, in the Node actions dropdown, select Execute run command.
- On the Run a command page, click in the search bar and select, Document name prefix, then click on Equals, then type in *AWS-RunShellScript.*
- Now select the radio button on the left of AWS-RunShellScript.



Document name prefix: Equals: AWS-RunShellScript

	Name	Owner	Platform types
●	AWS-RunShellScript	Amazon	Linux, MacOS

Description

Run a shell script or specify the commands to run.

Document version

Choose the document version you want to run.

1 (Default)

Command parameters

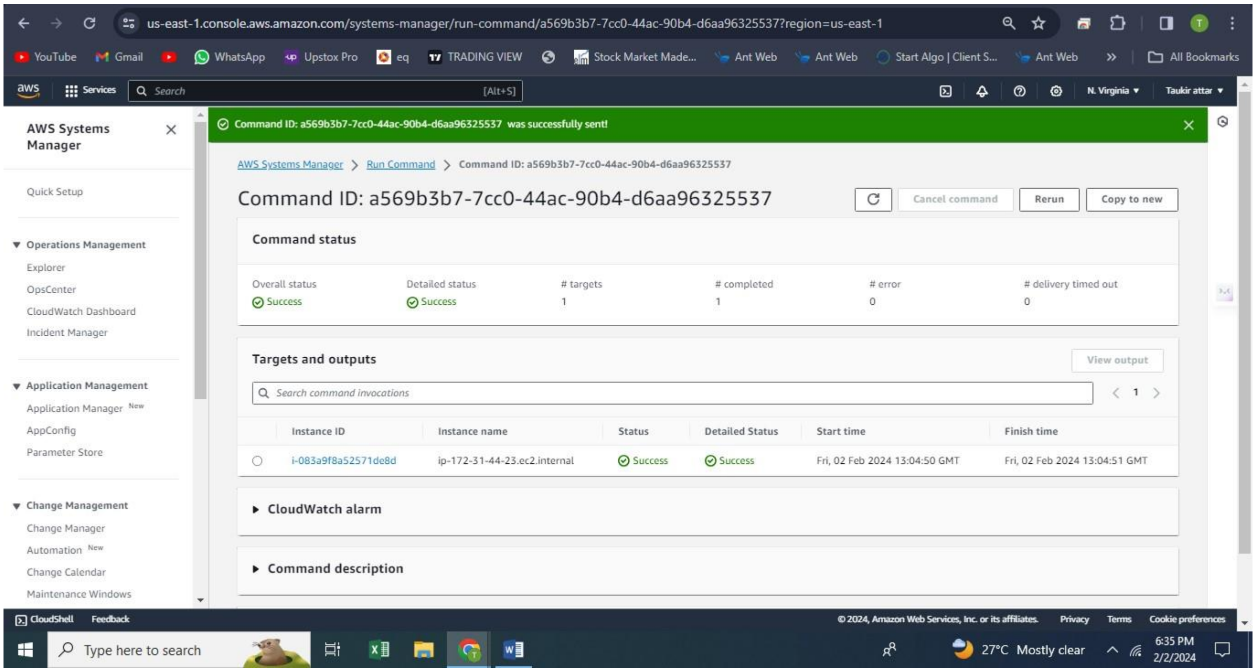
Commands

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

```
sudo yum update -y
```

- Scroll down to the Command Parameters panel and insert the following command in the Commands text box:
- sudo yum update –y

- Scroll down to the Targets panel and select the check box next to your managed EC2 instance.
- Finally, scroll down and select Run.
- While your script is running remotely on the managed EC2 instance, the Overall status will be In Progress. Soon the Overall status will turn to Success. When it does, scroll down to the Targets and outputs panel and select the Instance ID of your instance. Your Instance ID will be different than the one pictured.



Command ID: a569b3b7-7cc0-44ac-90b4-d6aa96325537 was successfully sent!

AWS Systems Manager > Run Command > Command ID: a569b3b7-7cc0-44ac-90b4-d6aa96325537

Command ID: a569b3b7-7cc0-44ac-90b4-d6aa96325537

Cancel command | Rerun | Copy to new

Command status

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

Targets and outputs

View output

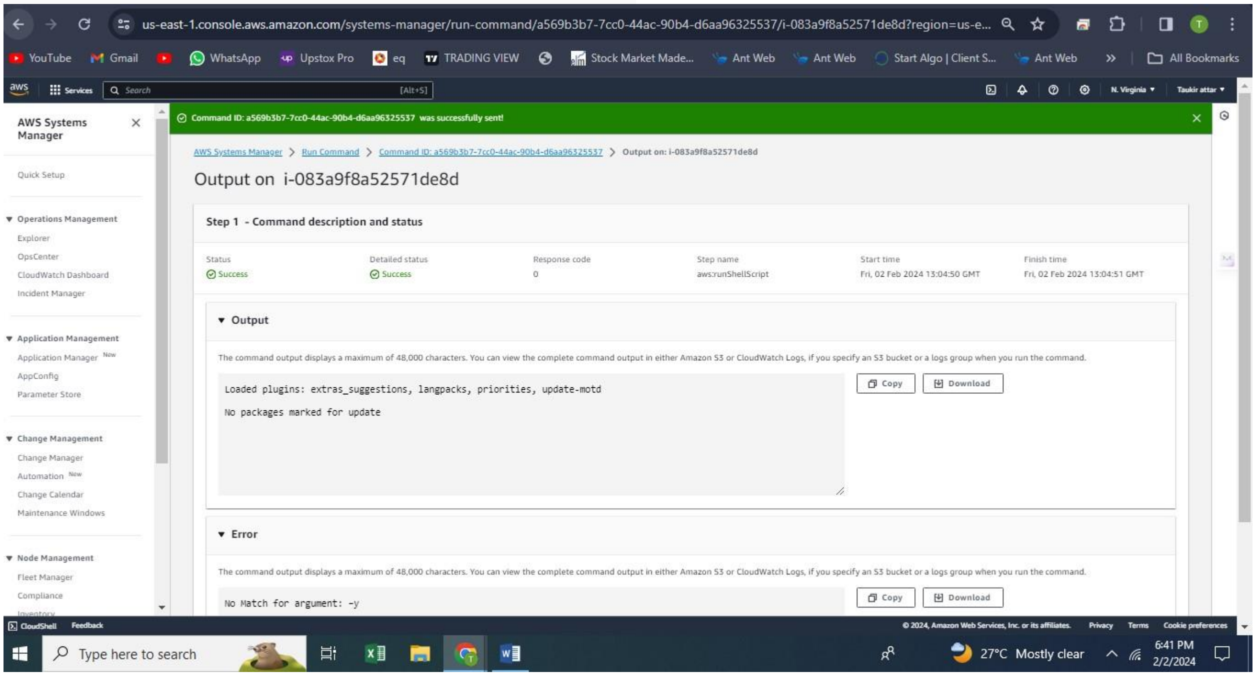
Search command invocations

	Instance ID	Instance name	Status	Detailed Status	Start time	Finish time
○	i-083a9f8a52571de8d	ip-172-31-44-23.ec2.internal	Success	Success	Fri, 02 Feb 2024 13:04:50 GMT	Fri, 02 Feb 2024 13:04:51 GMT

▸ CloudWatch alarm

▸ Command description

From the Output on: i-083a9f8a52571d8d, select the header of the Output panel to view the output of the update command from the instance.



Command ID: a569b3b7-7cc0-44ac-90b4-d6aa96325537 was successfully sent!

AWS Systems Manager > Run Command > Command ID: a569b3b7-7cc0-44ac-90b4-d6aa96325537 > Output on: i-083a9f8a52571de8d

Output on i-083a9f8a52571de8d

Step 1 - Command description and status

Status	Detailed status	Response code	Step name	Start time	Finish time
Success	Success	0	aws:runShellScript	Fri, 02 Feb 2024 13:04:50 GMT	Fri, 02 Feb 2024 13:04:51 GMT

▾ Output

The command output displays a maximum of 48,000 characters. You can view the complete command output in either Amazon S3 or CloudWatch Logs, if you specify an S3 bucket or a logs group when you run the command.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd

No packages marked for update
```

Copy | Download

▾ Error

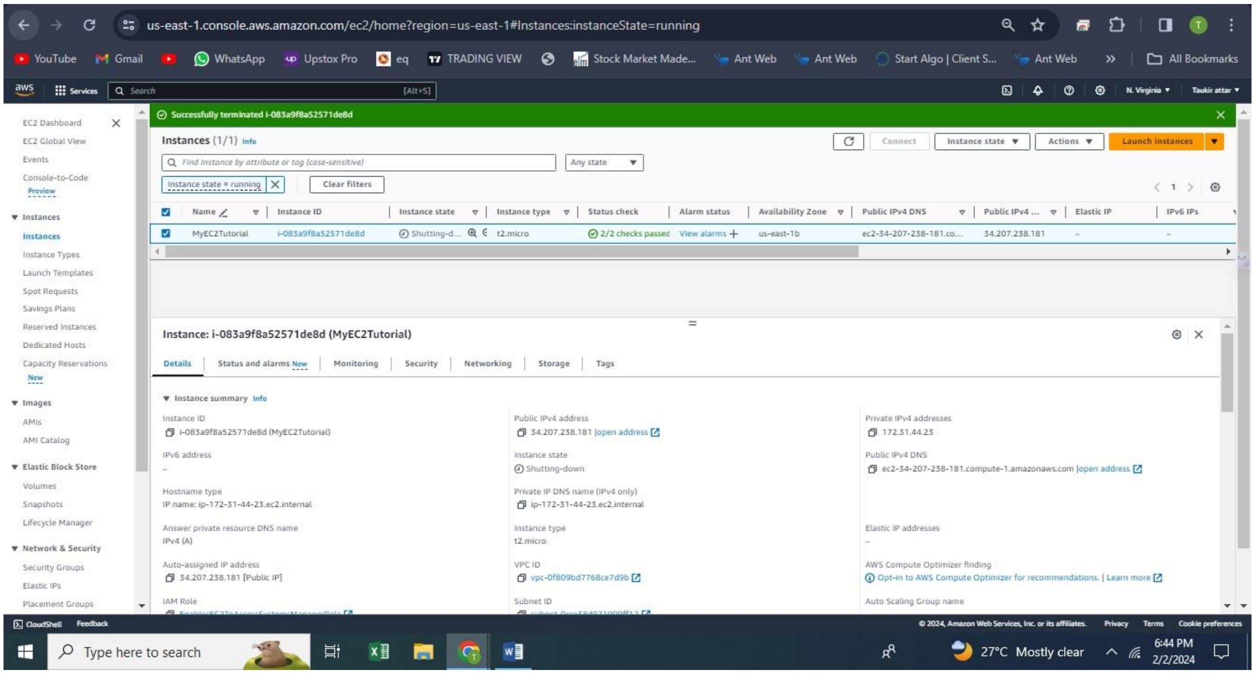
The command output displays a maximum of 48,000 characters. You can view the complete command output in either Amazon S3 or CloudWatch Logs, if you specify an S3 bucket or a logs group when you run the command.

```
No Match for argument: -y
```

Copy | Download

- In this step, you will terminate your Systems Manager and EC2 related resources. Important: Terminating resources that are not actively being used reduces costs and is a best practice. Not terminating your resources can result in a charge.

□



Successfully terminated i-083a9f8a52571de8d

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
MyEC2Tutorial	i-083a9f8a52571de8d	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-34-207-238-181.co...	34.207.238.181	–	–

Instance: i-083a9f8a52571de8d (MyEC2Tutorial)

Instance summary Info

Instance ID: i-083a9f8a52571de8d (MyEC2Tutorial)

IPv6 address: –

Hostname type: IP name: ip-172-31-44-23.ec2.internal

Answer private resource DNS name: IPv4 (A)

Auto-assigned IP address: 34.207.238.181 [Public IP]

IAM Role

Public IPv4 address: 34.207.238.181 |open address

Instance state: Shutting-down

Private IP DNS name (IPv4 only): ip-172-31-44-23.ec2.internal

Instance type: t2.micro

VPC ID: vpc-0f809bd7768ce7d9b

Subnet ID

Private IPv4 addresses: 172.31.44.23

Public IPv4 DNS: ec2-34-207-238-181.compute-1.amazonaws.com |open address

Elastic IP addresses: –

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name

Cost Analysis

AWS Systems Manager: The pricing for

Systems Manager Run Command is $0.05 . Assuming each user triggers one command per day (30 commands per month per user):

Cost per user = $0.05 * 30 = $1.5

Total cost for 1000 users = $1.5 * 1000 = $1500

Amazon EC2: The pricing for a t2.micro instance in the US East (N. Virginia) region is $0.0116 per hour. Assuming the instance runs continuously for a month:

Cost per instance = $0.0052 * 24 * 30 = $3.744

Total cost for 1000 users = $3.744 * 1000 = $3744

Total Monthly Billing Estimate:

Systems Manager:	$1500
EC2 instances:	$3744
Total Estimate:	$5244

lessons and observations

AWS Systems Manager to remotely run commands on Amazon EC2 instances, several key lessons and observations can be drawn:

- Cost Considerations:

Pay-as-You-Go Pricing: AWS Systems Manager follows a pay-as-you-go pricing model, where you pay only for the resources you use. This can be cost-effective, especially for small to medium-sized businesses or projects.

Optimization: To optimize costs, it's important to monitor and manage the usage of Systems Manager and EC2 instances. This includes reviewing usage patterns, selecting the right instance types, and leveraging cost-saving options like reserved instances or spot instances where applicable.

- Security and Compliance:

Secure Communication: Systems Manager uses secure communication channels to execute commands on EC2 instances, ensuring that sensitive information is protected.

Compliance Standards: Systems Manager helps in maintaining compliance with standards such as PCI DSS, HIPAA, and GDPR by providing audit logs, compliance reports, and encryption features.

- Operational Efficiency:

Automation: Systems Manager offers automation features that allow you to automate common administrative tasks, reducing manual intervention and improving efficiency.

Centralized Management: With Systems Manager, you can centrally manage EC2 instances across multiple AWS accounts and regions, streamlining operations and reducing complexity.

- Agent Management:

Agent Installation: Installing the SSM Agent on EC2 instances is essential for Systems Manager to communicate with the instances. You can automate this process using AWS Systems Manager State Manager or AWS CloudFormation.

Agent Updates: Regularly updating the SSM Agent ensures that you have access to the latest features, improvements, and security patches.

- Scalability:

Scaling Resources: Systems Manager is designed to scale with your infrastructure, allowing you to manage a large number of instances efficiently.

Resource Groups: Using resource groups in Systems Manager, you can organize and manage instances based on tags, simplifying management in large-scale environments.

By considering these aspects, you can effectively leverage AWS Systems Manager to remotely run commands on Amazon EC2 instances, ensuring cost-effectiveness, security, compliance, operational efficiency, and scalability in your AWS environment.