Amazon Capstone Project 1

Process that needs to be setup to remote manage EC2 instances from an operations perspective.

This will be executed in 4 different phases:

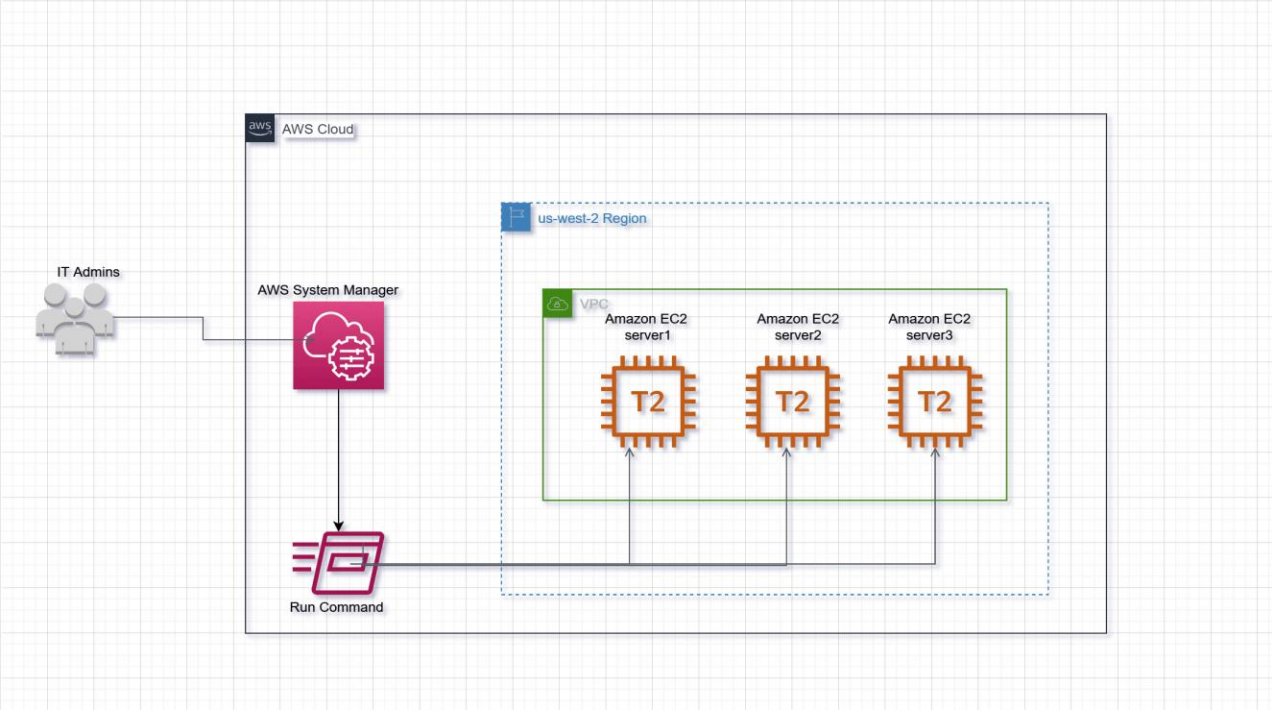
Phase 1: Identity and Access Management (IAM) - Creating role that will be used to give Systems Manager permission to perform actions on your EC2 instances.

Phase 2: Compute - Creating an EC2 instance & enabling the agent on this instance that communicates with System manager

Phase 3: Amazon system Manager: Update the AWS-UpdateSSMAgent

Phase 4: **Amazon system Manager**: Run a Remote Shell Script

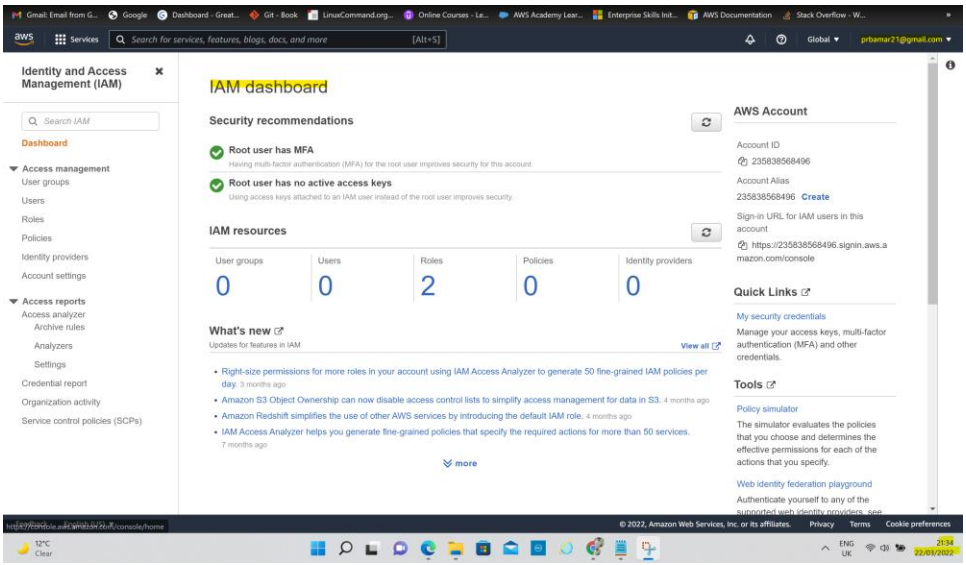
Architectural Diagram



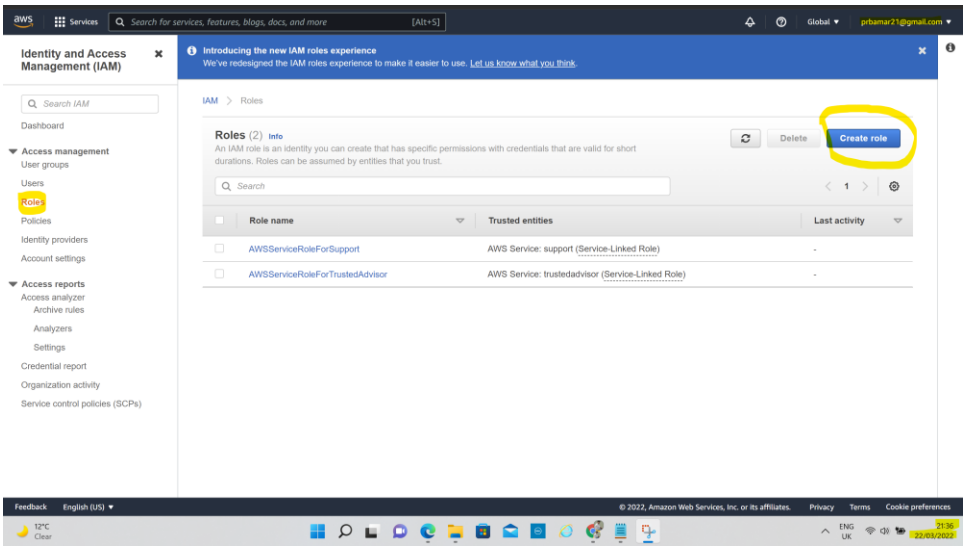
Phase 1. Create an Identity and Access Management (IAM) role

In this step, an IAM role that will be used to give Systems Manager permission to perform actions on your instances.

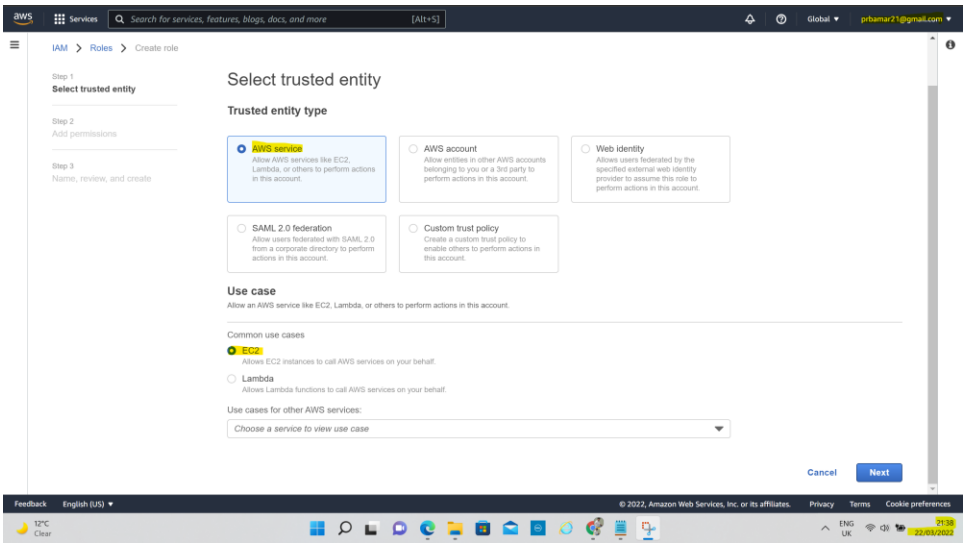
1a. Open the IAM console:



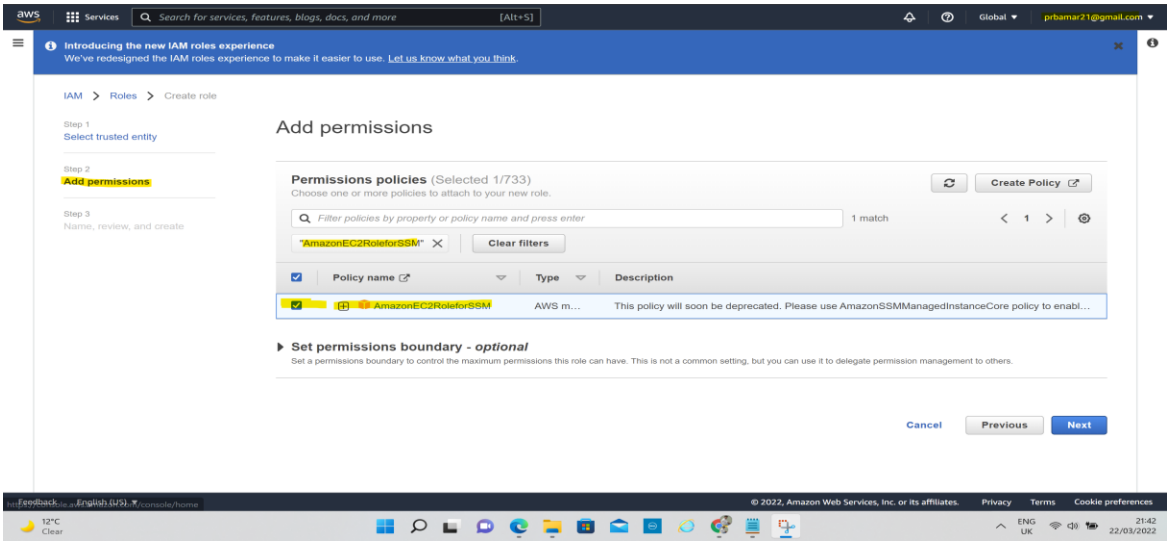
1b. Choose Roles, and then choose Create role.



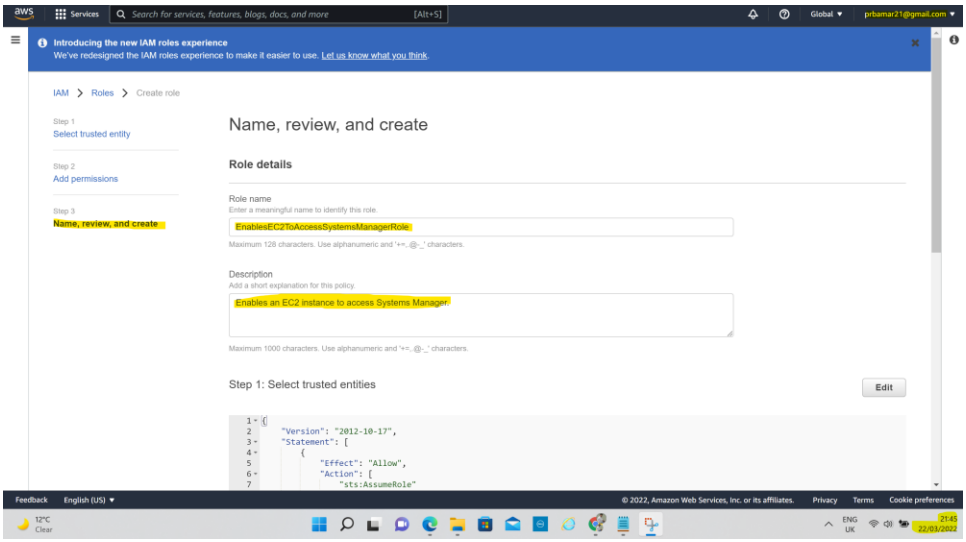
1c. On the Select type of trusted entity page, under AWS Service, choose EC2, and then choose Next: Permissions.



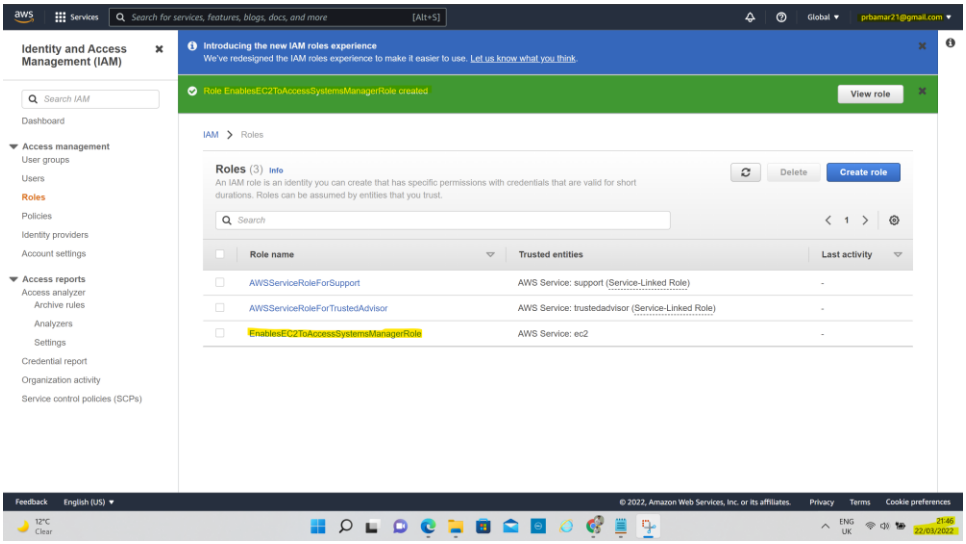
1d. On the Attached permissions policy page, in the search bar type AmazonEC2RoleforSSM then from the policy list select AmazonEC2RoleforSSM, and then choose Next: Review.



1e. On the Review page, in the Role name box type in EnablesEC2ToAccessSystemsManagerRole. In the Role description box type in Enables an EC2 instance to access Systems Manager. Choose Create role.



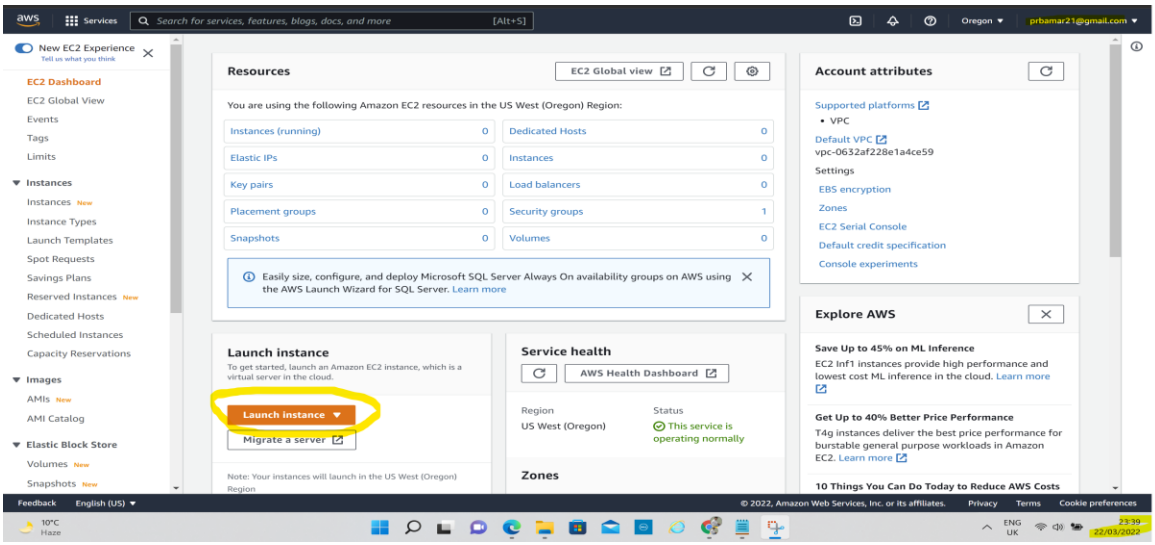
after Creating role



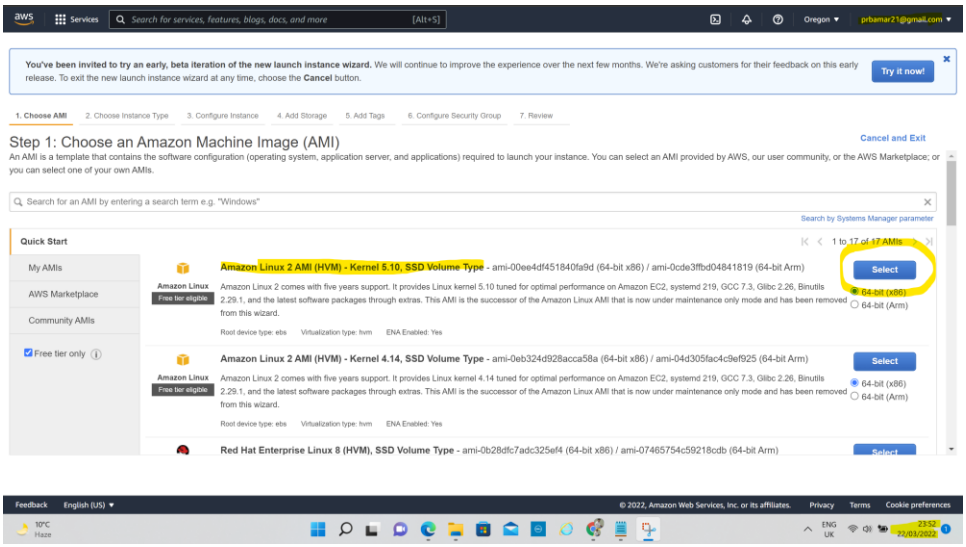
Phase2. Create an EC2 instance: Selected region: US West (Oregon) us-west-2

In this step will create an EC2 instance using the EnablesEC2ToAccessSystemsManagerRole role. This will allow the EC2 instance to be managed by Systems Manager.

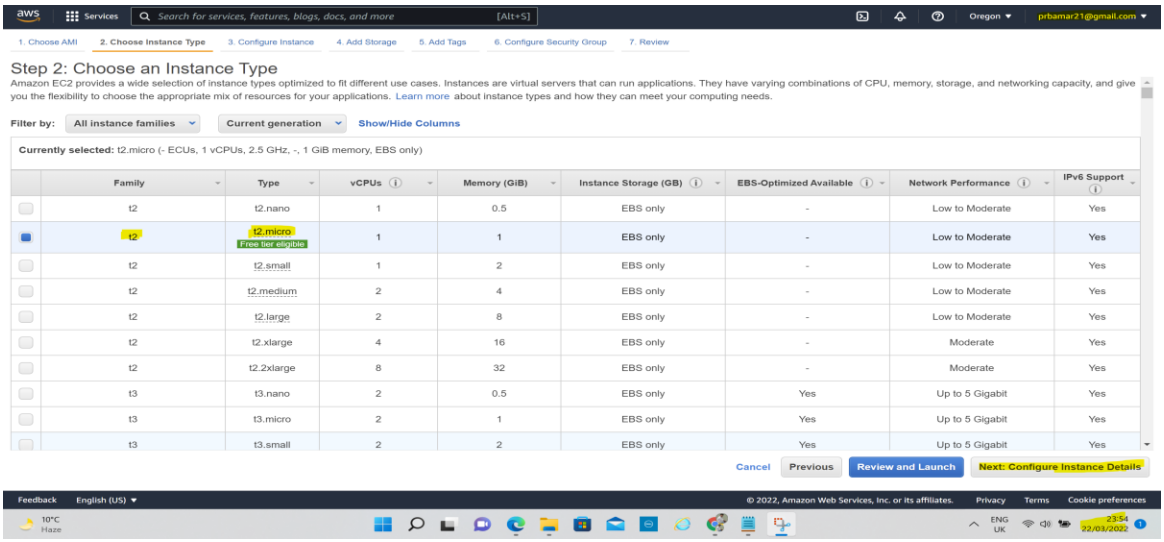
2a. Open the Amazon EC2 console. From the EC2 console select your preferred region. Systems Manager is supported in all AWS Regions. Now choose Launch Instance.



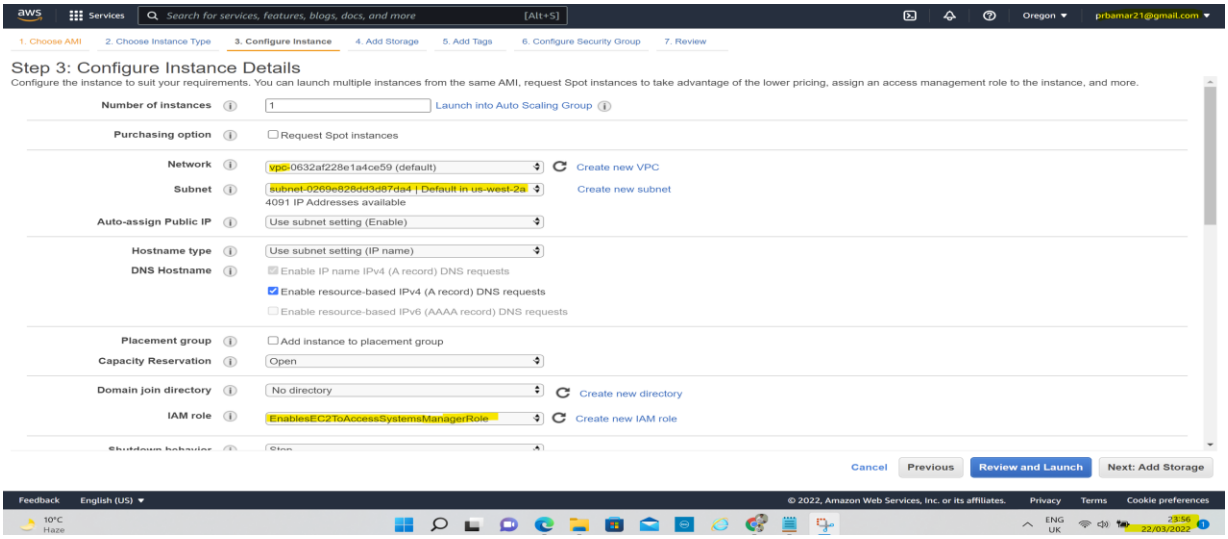
2b. Select the Amazon Linux AMI. Make sure you select Amazon Linux base AMI dated 2017.09 or later which includes the Systems Manager Agent by default. You can also install the Systems Manger Agent on your own Windows or Linux system.



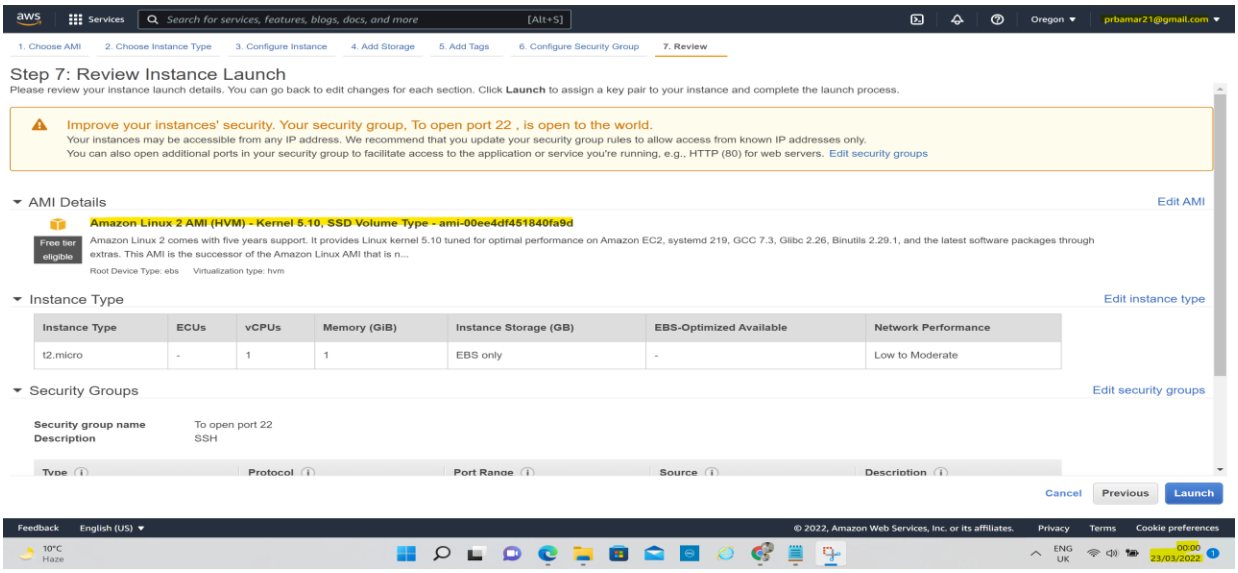
2c. On the Step 2: Choose an Instance Type page, choose the t2. micro instance type and then click Next: Configure Instance Details.



2d. On the Step 3: Configure Instance Details page, In the IAM role dropdown choose the EnablesEC2ToAccessSystemsManagerRole role you created earlier. Leave everything else as default. Choose Review and Launch.

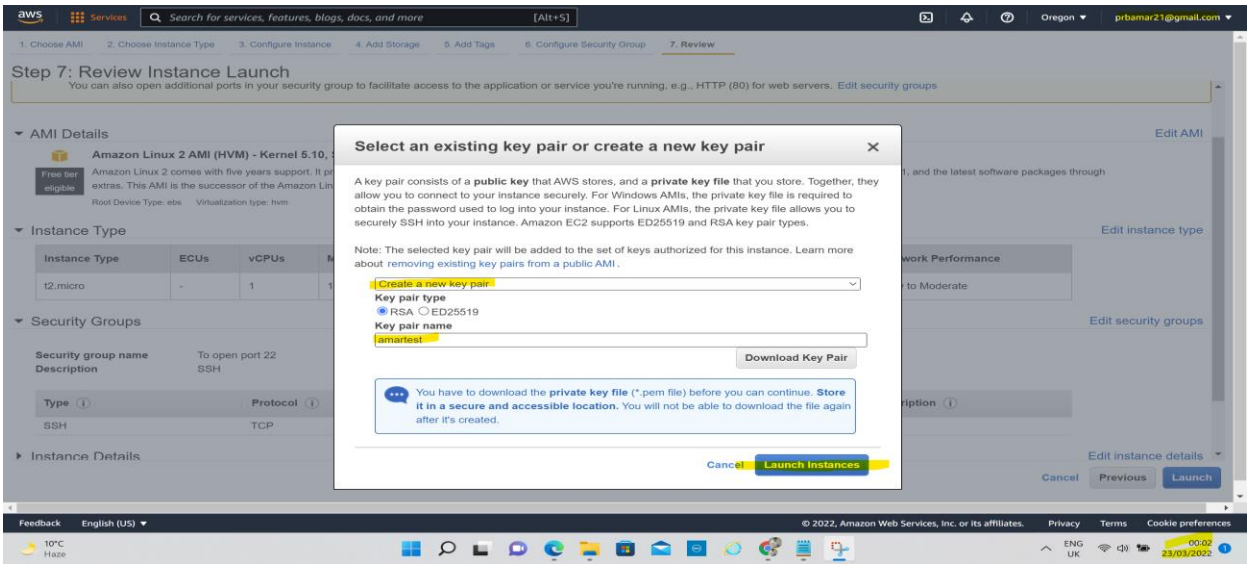


2e. On the Step 7: Review Instance Launch page, choose Launch to launch your instance.

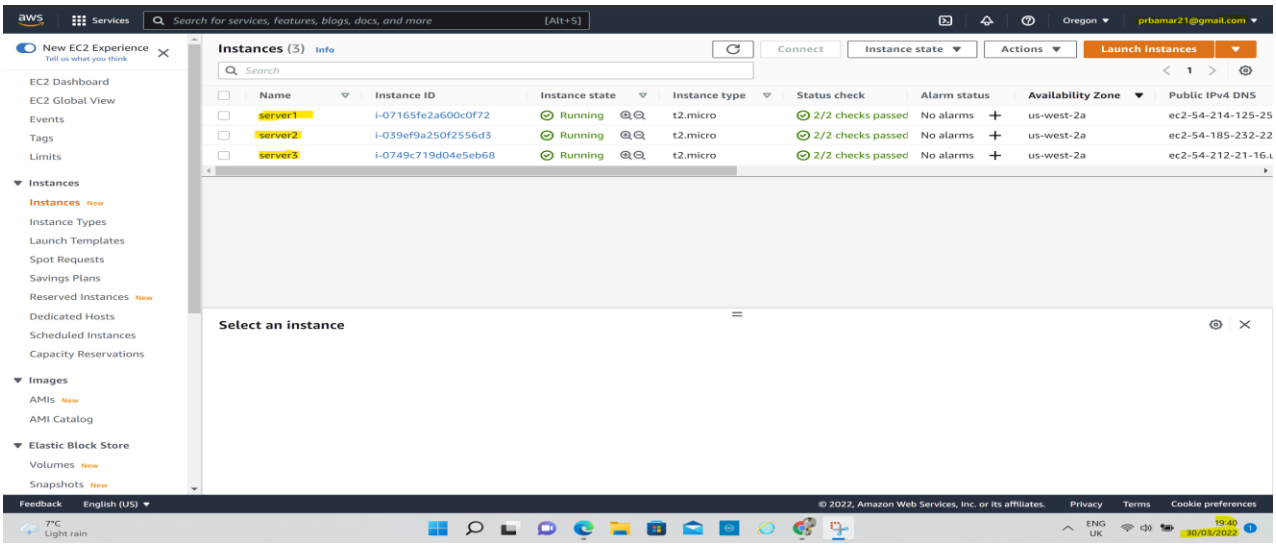


2f. Next the Select an existing keypair or create a new key pair dialog will appear. You will not need a keypair to use Systems Manager to remotely run commands.

Next select Launch Instance.



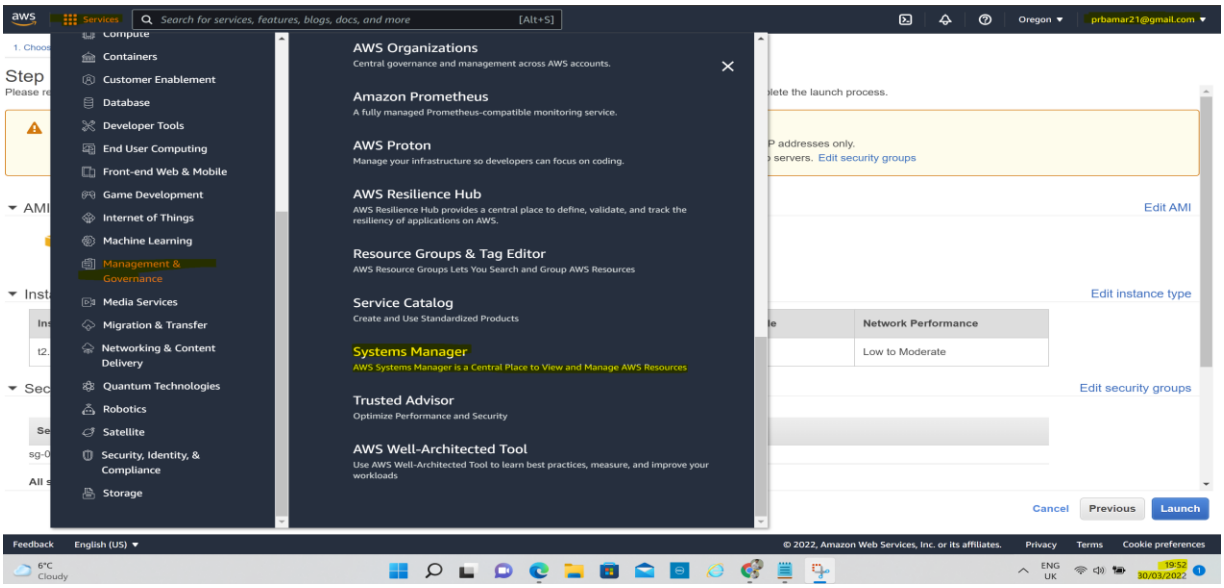
Created 3 EC2 instances server1, server2 & server3 with IAM role EnablesEC2ToAccessSystemsManagerRole attached to each of these instances



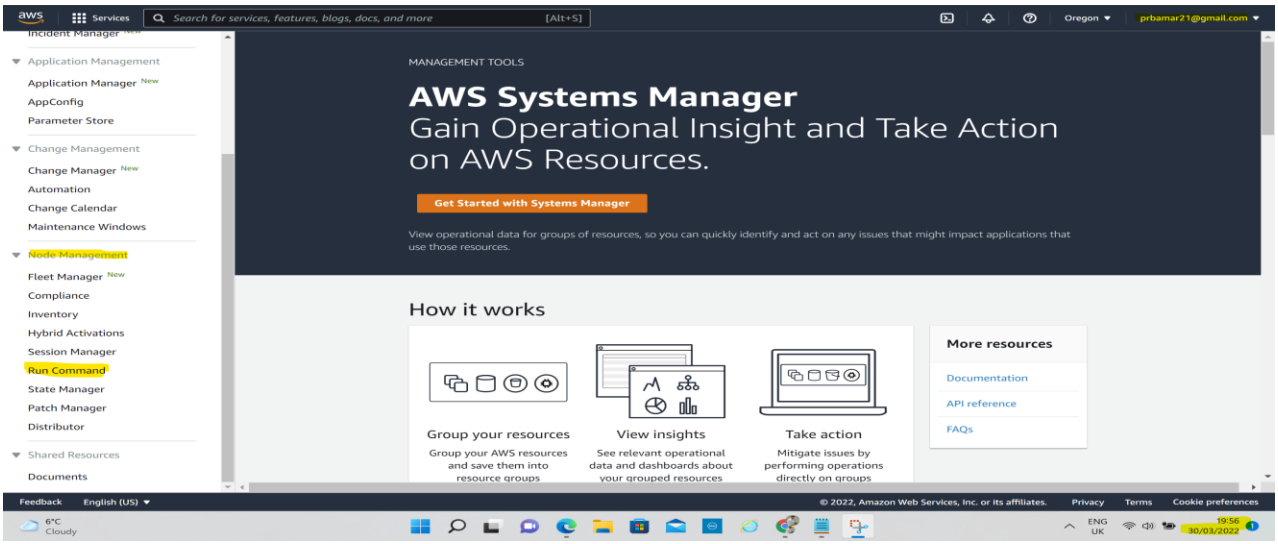
Phase 3. Update the Systems Manager Agent

3a. Now we have 3 EC2 instances running with the Systems Manager agent, in this step, we will you run a pre-packaged command, called a document, that will upgrade the agent. It is best practice to update the System Manager Agent when you create a new instance.

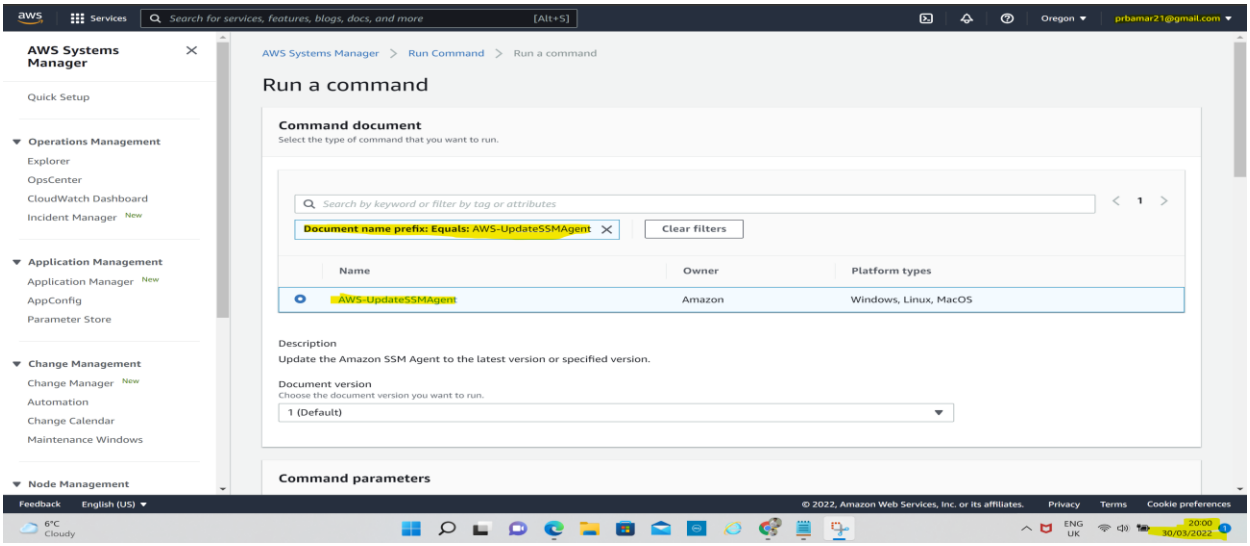
-> Navigate to Service under Management tools select Systems Manger



3b. Select run command in the Node management

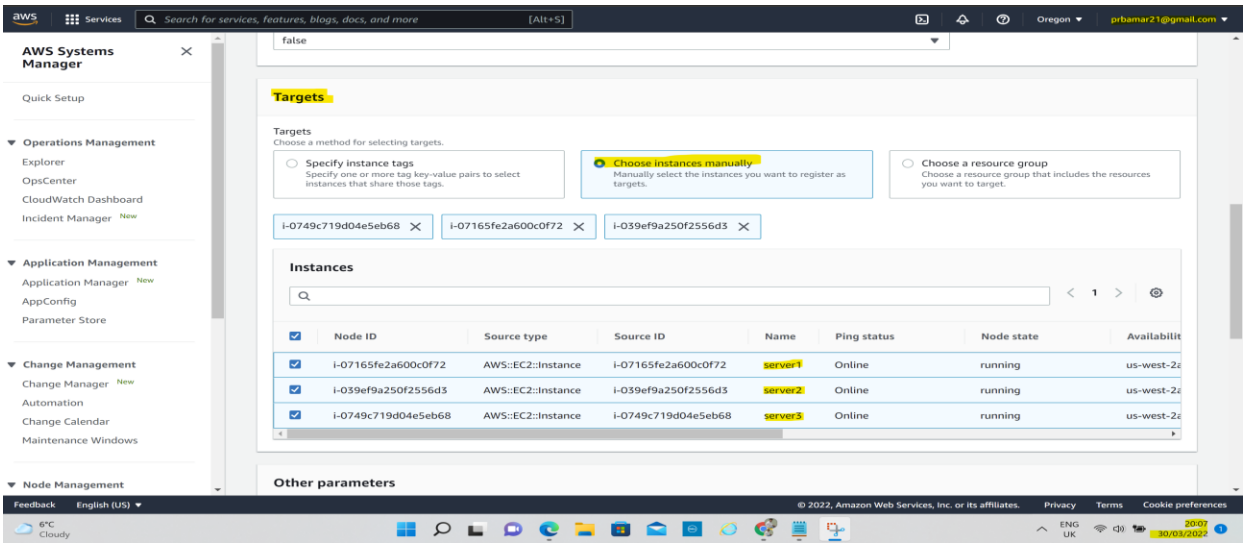


3c. On the Run a command page click in the search bar and select, Document name prefix, then click on Equal, then type in AWS-UpdateSSMAgent.

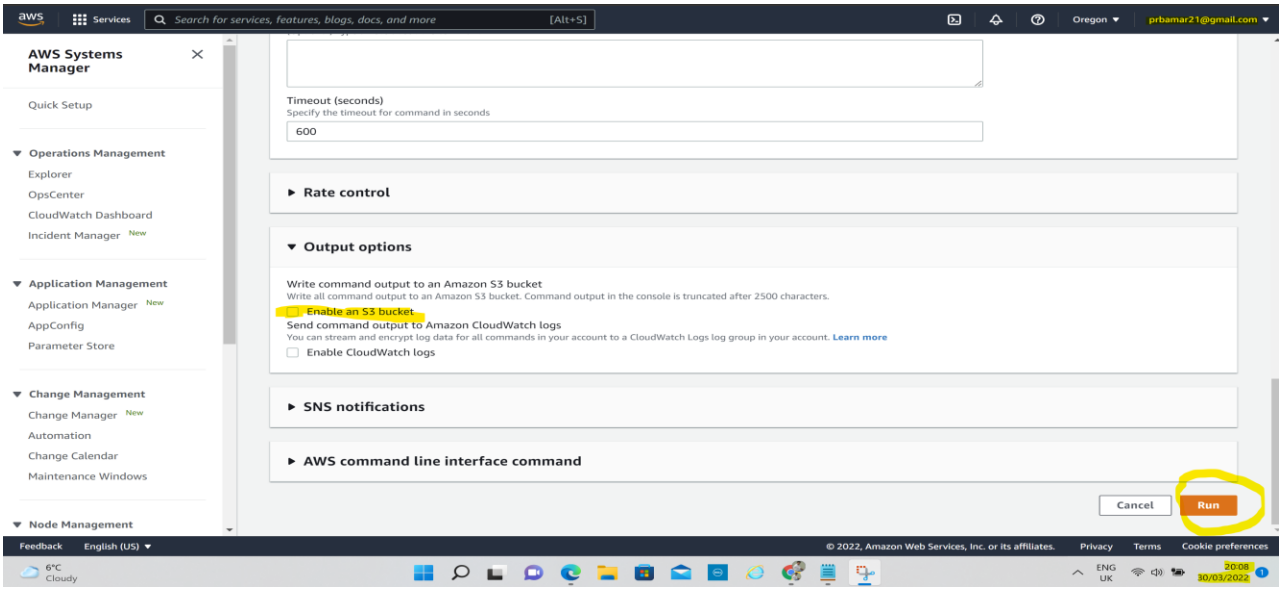


3d. Scroll down to the Targets panel and select Choose instances manually, select required EC2 instances for example here server1, server2 & server3 and Uncheck S3 bucket (It is not needed for this project).

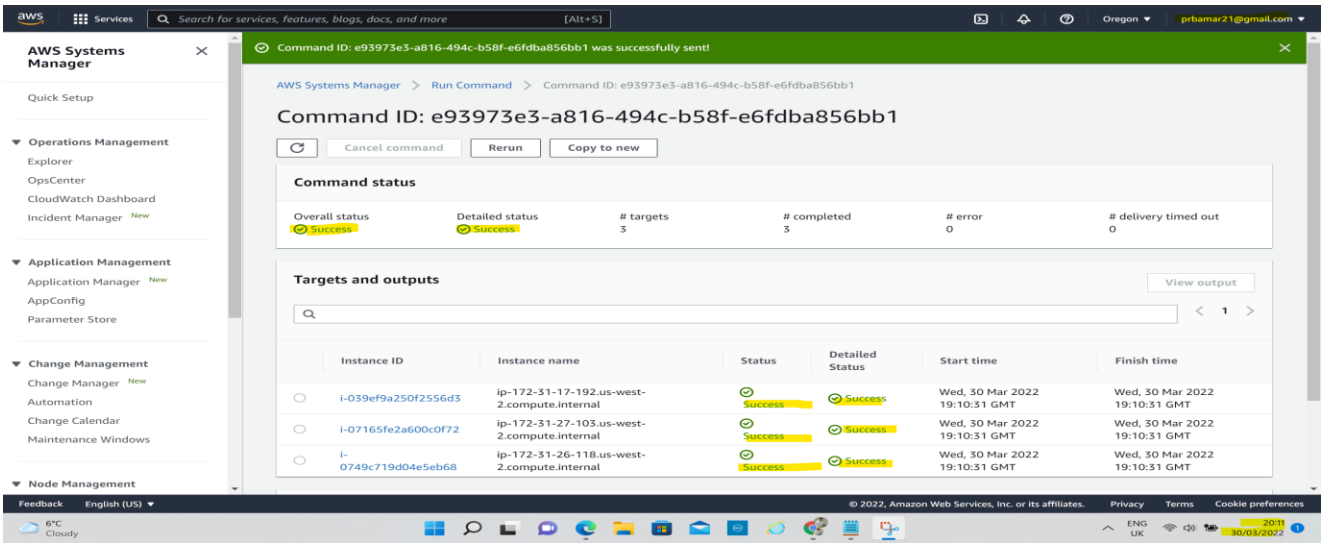
If you Enable S3 bucket then output will be written to S3 bucket.



Scroll down & Click on Run

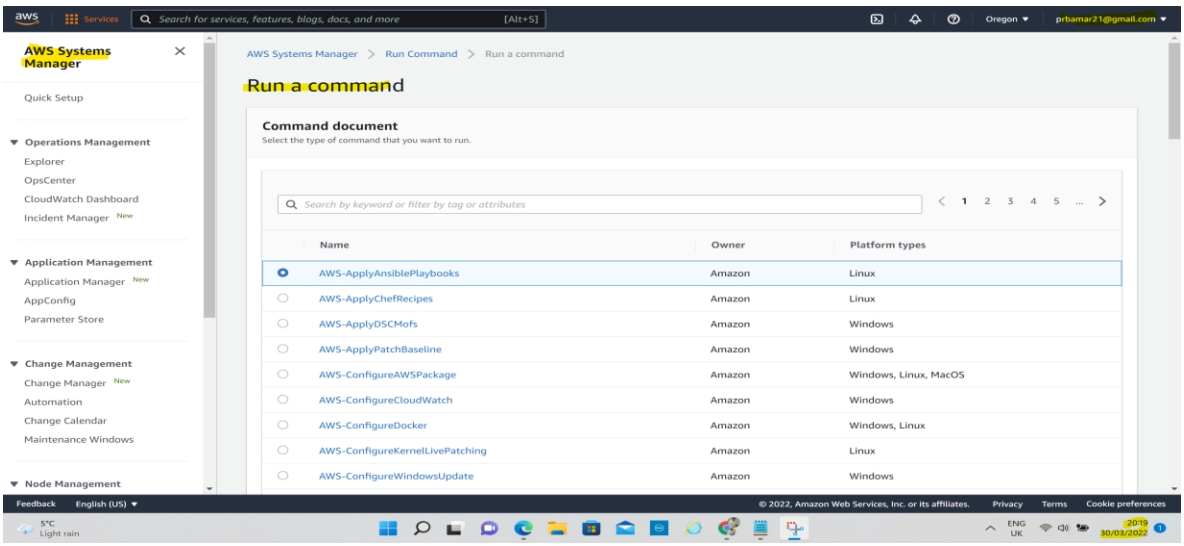


3e. Overall success can be seen in green

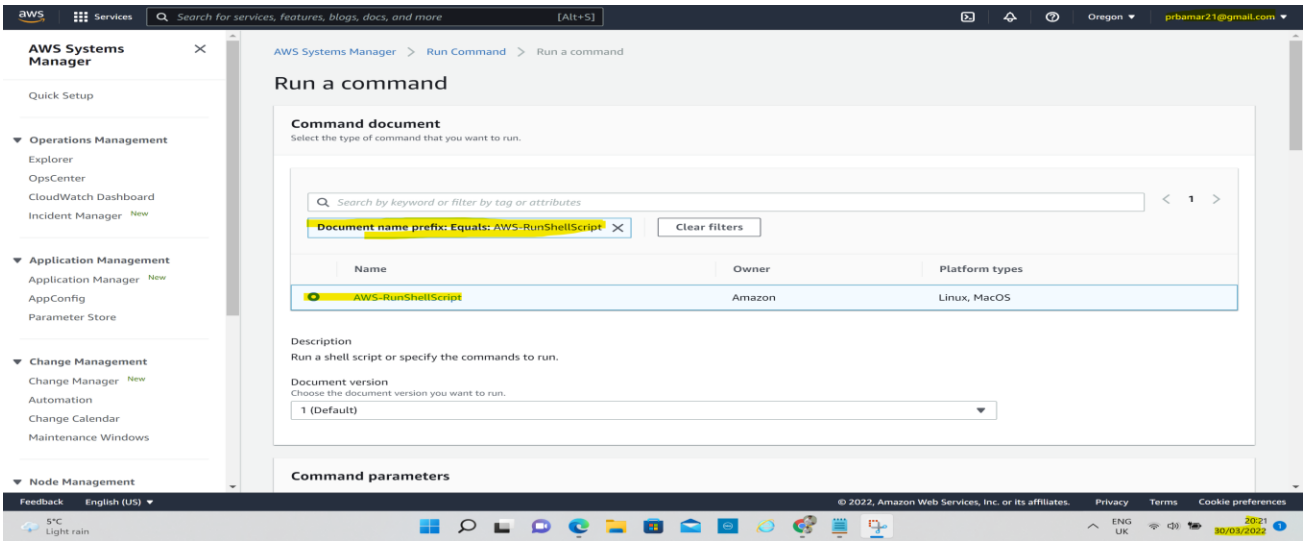


Phase 4. Run a Remote Shell Script

4a. In this step, you will run a shell script through Run Command. Navigate to Service under Management tools select System Manager -> Select run command ->

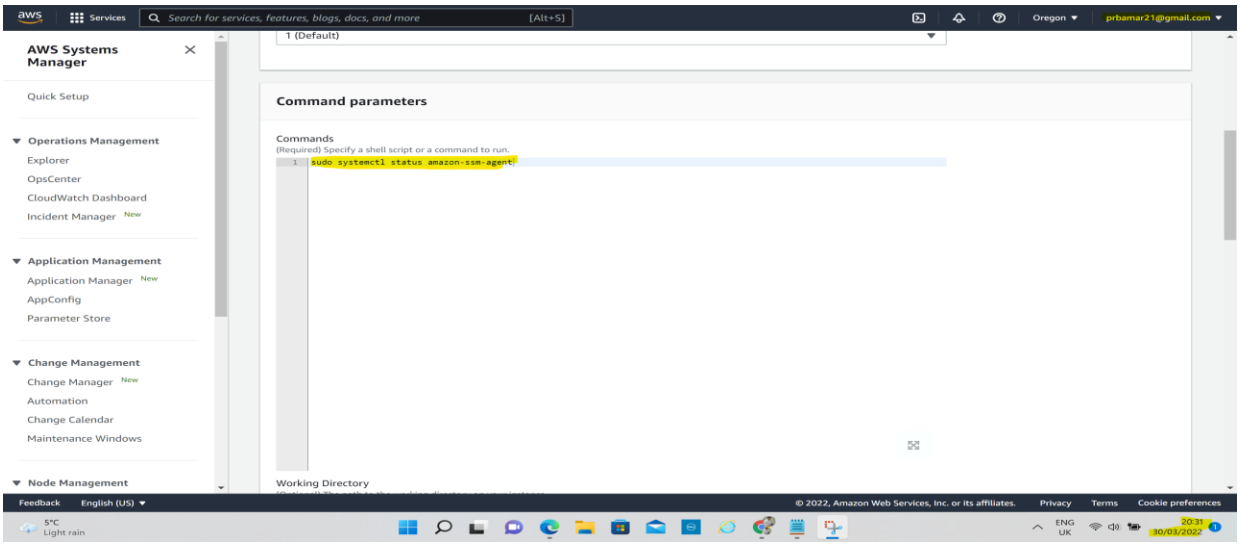


4b. On the Run a command page, click in the search bar and select, Document name prefix, then click on Equal, then type in AWS-RunShellScript.



4d. Scroll down to the Command Parameters panel and insert the following command in the Commands text box:

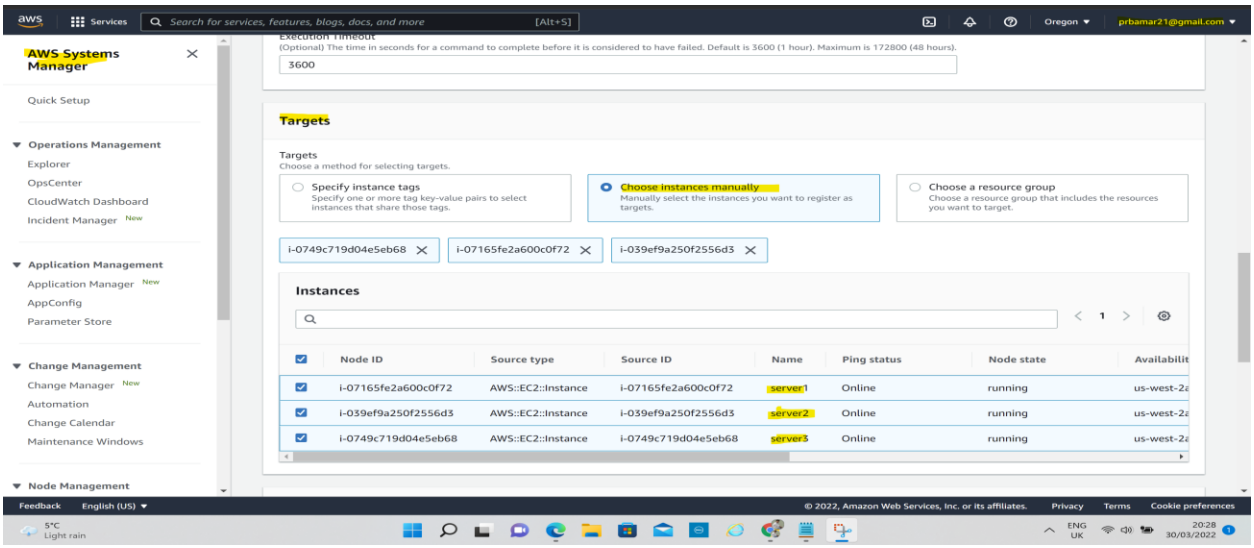
sudo systemctl status amazon-ssm-agent



4d. Scroll down to the Targets panel and select Choose instances manually, select required EC2 instances for example here server1, server2 & server3 and Uncheck S3 bucket (It is not needed for this project).

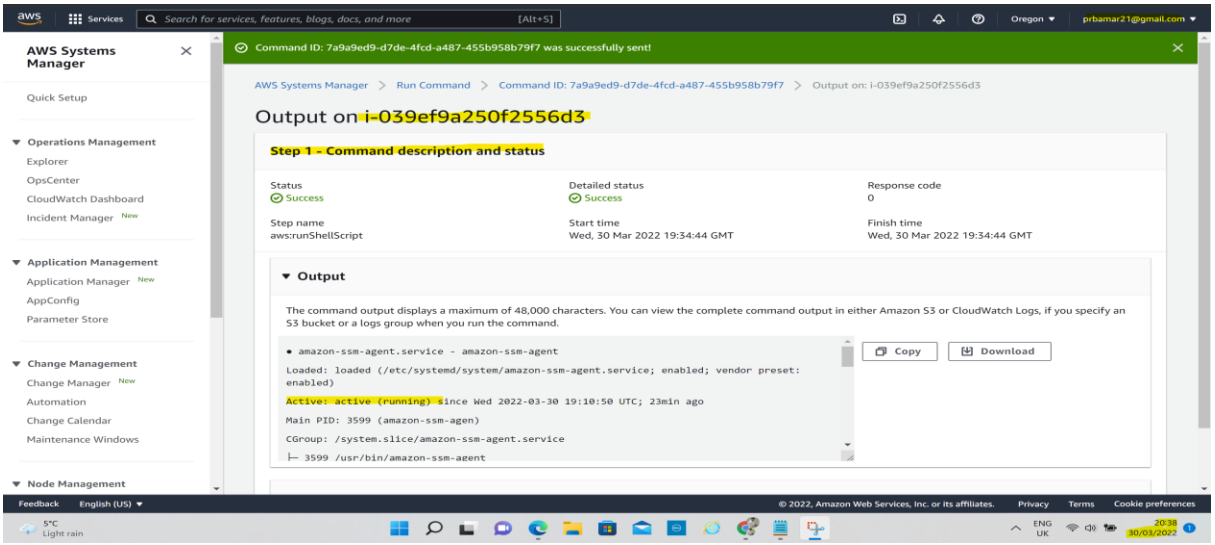
If you Enable S3 bucket then output will be written to S3 bucket.

Finally, scroll down and select Run

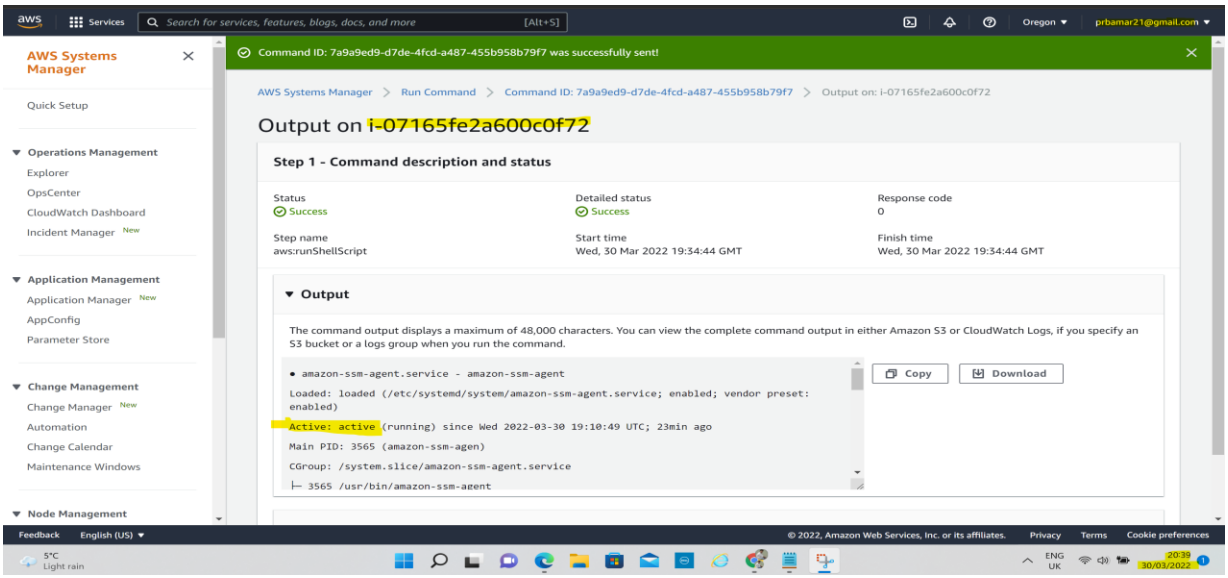


4e. From the Output on: i-XX page click on the header of the Step 1 - Output panel to view the output of the update command from the instance.

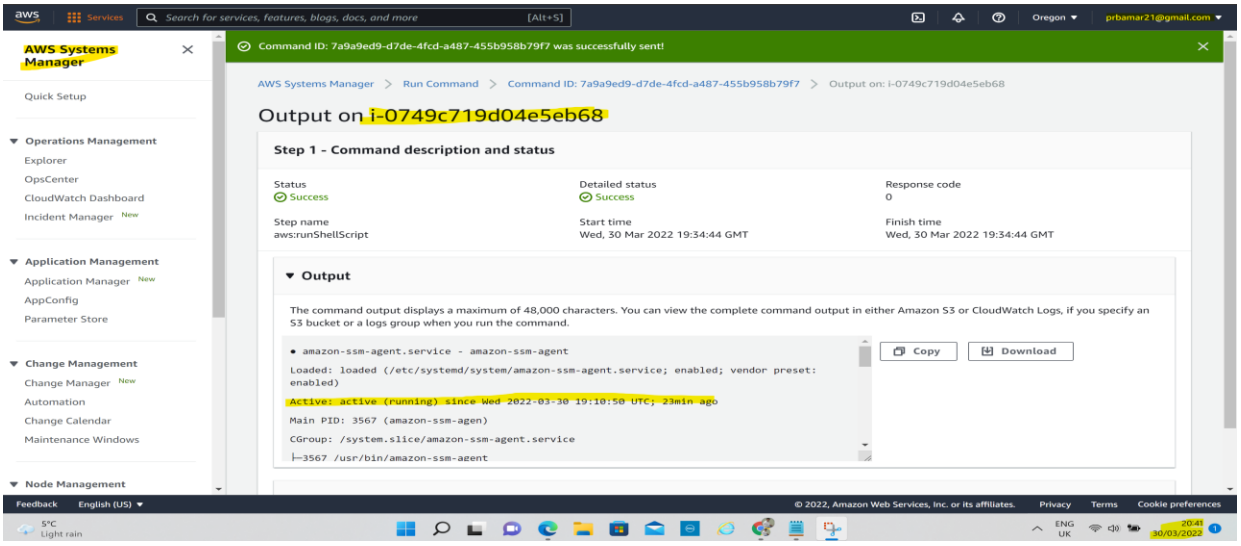
Output on i-039ef9a250f2556d3 server2



Output on i-07165fe2a600c0f72 server1



Output on i-0749c719d04e5eb68 server3



Cost Analysis:

- Here Free tier account is used, for 12 months EC2 instance (only free tier eligible) will be free, we get 750 hours per month and 30 GB Elastic block storage with any combinations of General Purpose SSD or Magnetic.
- AWS Systems Manager: As a part of the AWS Free Tier, we get started with the 13 AWS Systems Manager features for free.
- We can reduce human efforts by automating task using system manager this will increase productivity with minimum manpower.

Lessons & Observations

- Learn Amazon system manager service and steps for managing EC2 instances remotely without logging to individual instances via SSH or Bastian host.
- Operations task can be automated using system manager like patch scans, inventory collections.
- With the latest AWS console options have changed on AWS system manager console.
- We can now store output in S3 buckets, there is option to enable it.