AWS exam:

Section 1: Multiple Choice Questions (MCQs):

1. C
2. A
3. C
4. A
5. B
6. B
7. A
8. B
9. C
10. C
11. A

Research-based AWS Questions - using google only:

12. What are AWS Landing Zones, and how do they help with multi-account governance?

AWS Landing Zones:
a well-architected, multi account AWS environment that is a starting point from which you can deploy workloads and applications. It's a good baseline for multi-account architecture, identity and access management, governance, data security, network design and logging.
how they help with multi account governance:
enables enforcement of controls to ensure compliance with corporate guidelines, across multiple accounts in your environment. LZ is a recommended cloud environment that includes default accounts, account structure, network deployment and security.

13. Explain how AWS WAF protects web applications from common attacks.

WAF - web application firewall- a security tool, protect our web app by filtering, monitoring, and blocking any malicious HTTP/S traffic traveling to the web application, prevents any unauthorized data from leaving the app.
how:
-WAF creates rules to filter web requests based on conditions like IP addresses, http structure, or custom URIs.
-monitor the application's login page for unauthorized access to user accounts using compromised credentials.
-create and maintain rules automatically and incorporate them into the development and design process.

14. What is AWS Snowball, and when should it be used?

AWS Snowball is a service that provides secure devices capable of storing large amounts of data (like 100 TB) with strong end-to-end encryption. It allows clients to bring AWS computing and storage capabilities to edge locations and transfer data securely into and out of AWS. The service accelerates the transfer of large amounts of data to and from the AWS cloud using physical storage devices for transport.

You should use AWS Snowball when you need to Run computing in rugged, austere, mobile, or disconnected environments, or when you transfer large-scale data when bandwidth is insufficient for high-speed online transfer.

What are the key differences between AWS Backup and manual snapshot backups?

AWS snapshot is a point of time copy of an Amazon EBS volume for an EC2 instance with limited storage and recovery options.
AWS ec2 backup is more comprehensive and flexible copy of your cloud workloads, offering reliable protection and ensuring fast and consistent recovery.
key differences:
purpose:
snapshot: quick recovery, testing, virtual environments
backup: focus on data protection and disaster recovery
recovery speed:
snapshot: faster restoration
backup: slower restoration due to larger data volume
storage efficiency:
snapshot: stores changes since last snapshot
backup: Stores complete data regardless of changes
Risk of Data Loss:
snapshot: potential loss of interim data
backup: minimal risk if backups are properly managed

15. How does AWS Shield help mitigate DDoS attacks?

AWS shield managed DDoS protection service by providing dynamic detection and automatic inline mitigation that minimize application downtime and latency.
all AWS customers get am automatic protection with no additional cost.
its always on monitoring, means the AWS shield continuously monitors AWS global network traffic, searching for possible signs of DDoS malicious activity or targeting customer resources DDoS attacks.
when AWS shield detect DDoS attack, it automatically deploy inline mitigations to remove malicious traffic and helps regular traffic to reach intended customer systems.

16. Explain the differences between AWS Transit Gateway and VPC Peering.

VPC peering is network connection between two VPC, enables to route traffic between them privately.

AWS Transit Gateway is service that connects VPCs and on premises networks throw a broker- central hub without relying on few point to point connections or transit VPC.
differences:
connection type:
AWS Transit Gateway: central hub connection
VPC peering: direct connection between VPCs
Scalability:
AWS Transit Gateway: highly scalable, easily connects multiple of VPCs - good at connection many VPCs
VPC peering: complex as more VPCs added - best for connecting small numbers of VPCs
complexity:
AWS Transit Gateway: simpler-one connection to the transit gateway connects to all
VPC peering: more VPCs = more complex
cost:
AWS Transit Gateway: additional cost
VPC peering: no additional charge (beyond data transfer)

17. What is AWS Step Functions, and how does it help with workflow automation?

AWS step function is visual workflow service for distributed applications. Developers are helped with this service to build distributed applications, automate processes, orchestrate microservices, ML pipelines.
the service enable you to coordinate individual tasks into visual workflow, makes you build and update applications quickly.
how it helps with workflow automation:
coordinate the flow between different services automatically, handling transitions between tasks, without manual intervention.

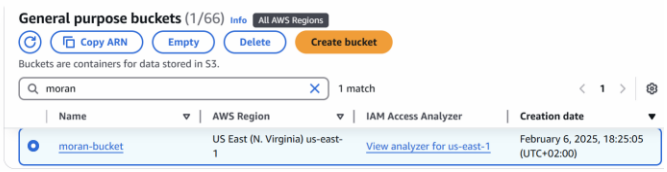
18. How does AWS Control Tower assist organizations in managing multiple AWS accounts?

19. What is the significance of AWS Outposts in hybrid cloud solutions?

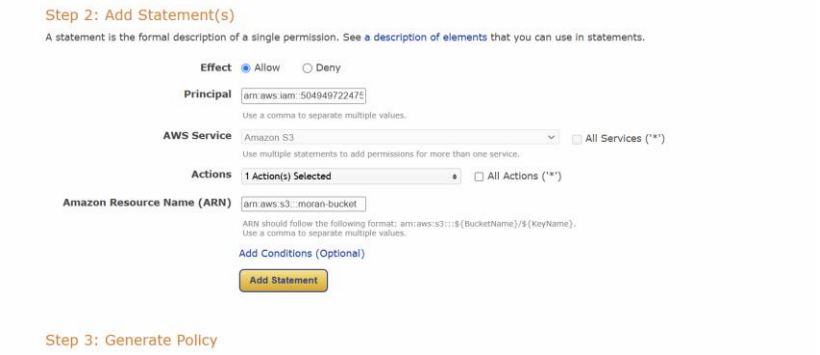
20. Explain the key use cases for AWS Elastic File System (EFS) compared to S3 and EBS.

Section 2: Hands-on UI-Based Questions

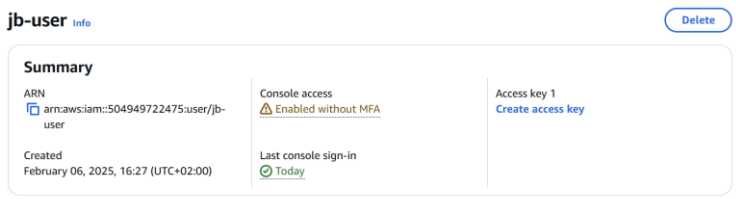
1. **S3 bucket:**
   finding s3 create bucket → gives the bucket name: "moran-bucket" → kept the default settings (bucket type, object ownership, Block Public Access settings for this bucket) → chose enable in backet versioning
   after creating bucket:



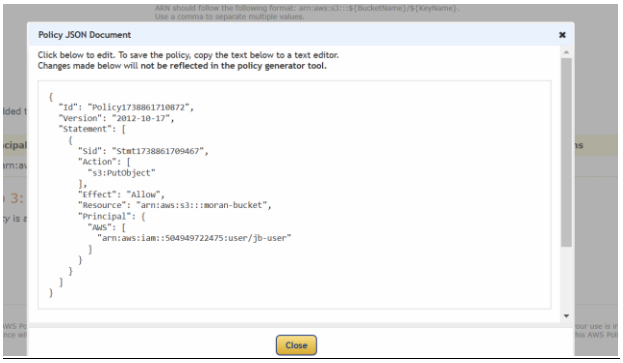
edit permission section and generate policy according to the mission:



Copy our arn user:
arn:aws:iam::504949722475:user/jb-user

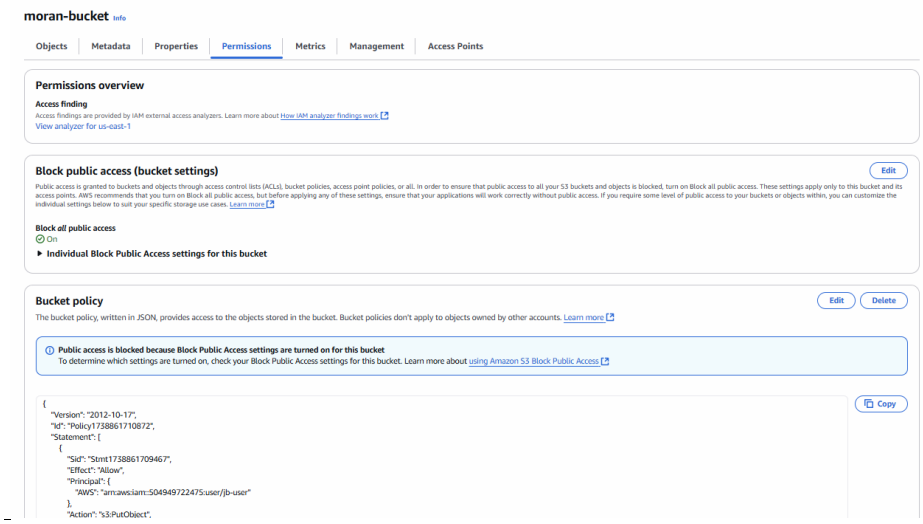


And create Jason file with the generastor:



with action "put object" only for upload objects (only upload objects).

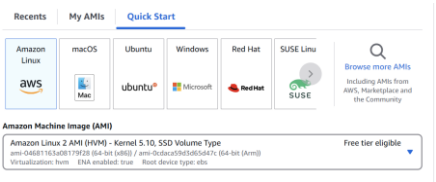
final:



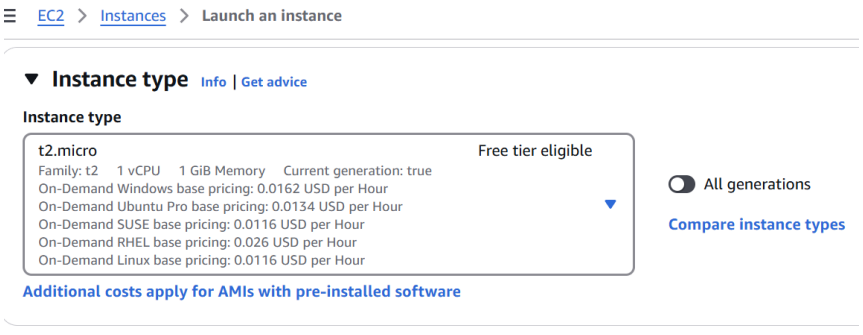
2. Launch ec2 instance:

ec2 name: moran-instance

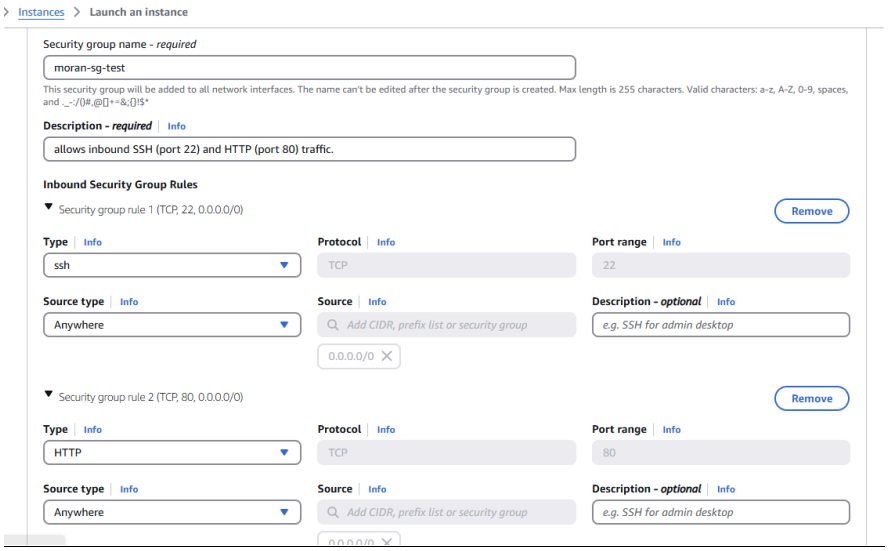
amazon machine image in my decision



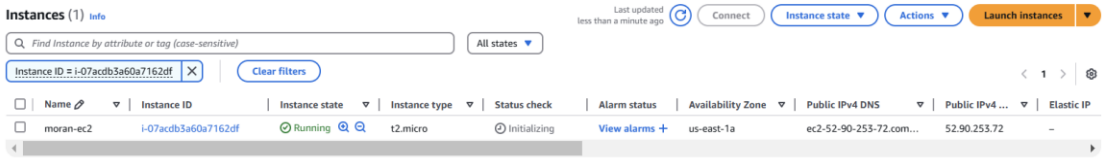
Instance type as we asked for:



Create Security Group that allows inbound SSH (port 22) and HTTP (port 80) traffic:



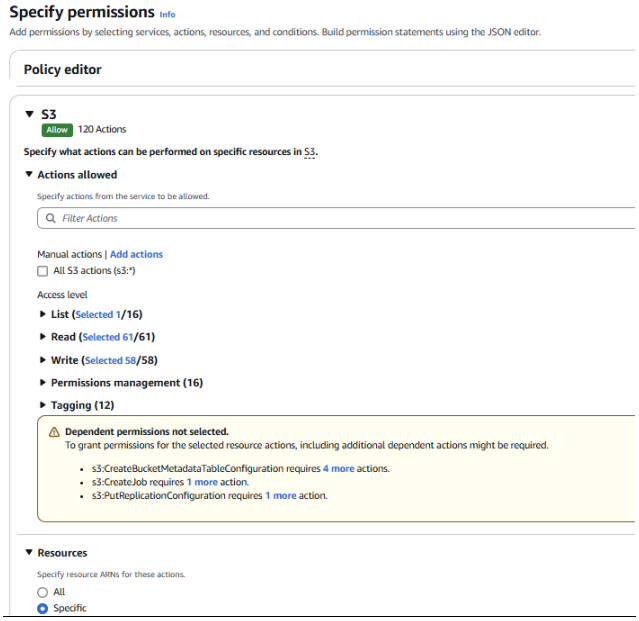
Final:



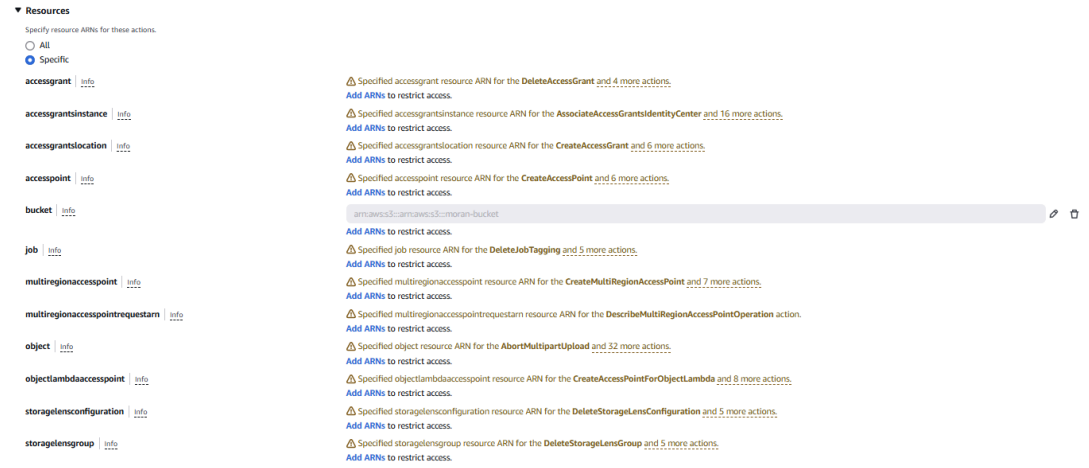
3. **<u>Configure an IAM User with S3 Access:</u>**

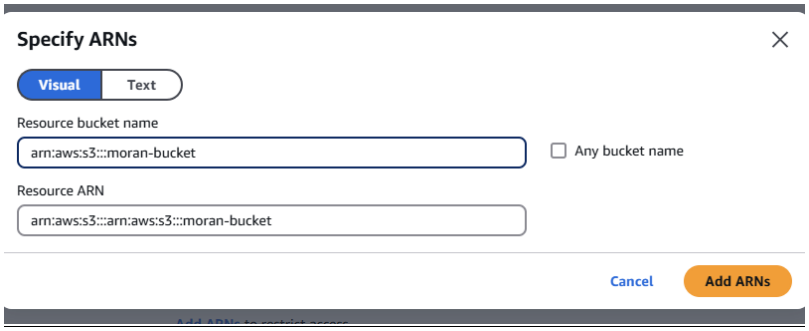
go to IAM and create new user named moran-user:

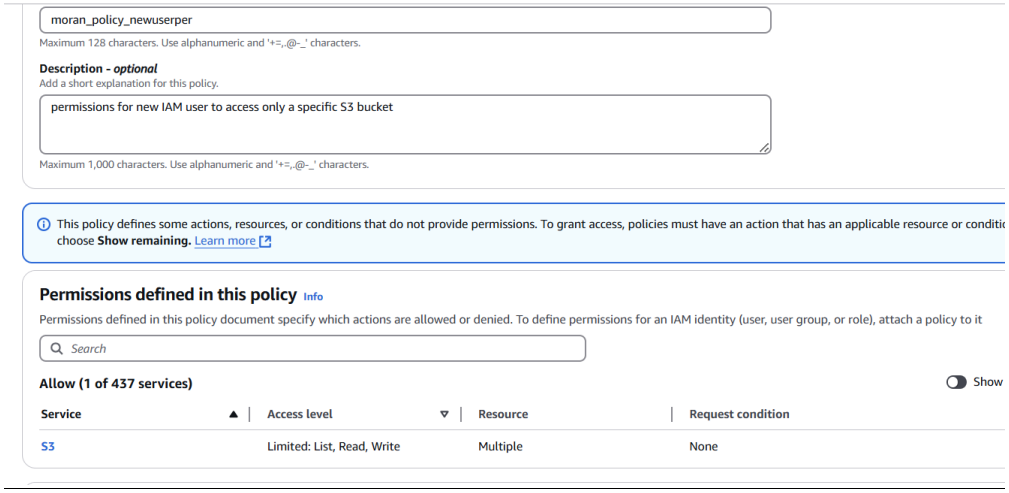
<u>And create password, then create policy with specific perrmisions:</u>



***list- only access to ListBucket, all the options for read and write.**
only one specific bucket:

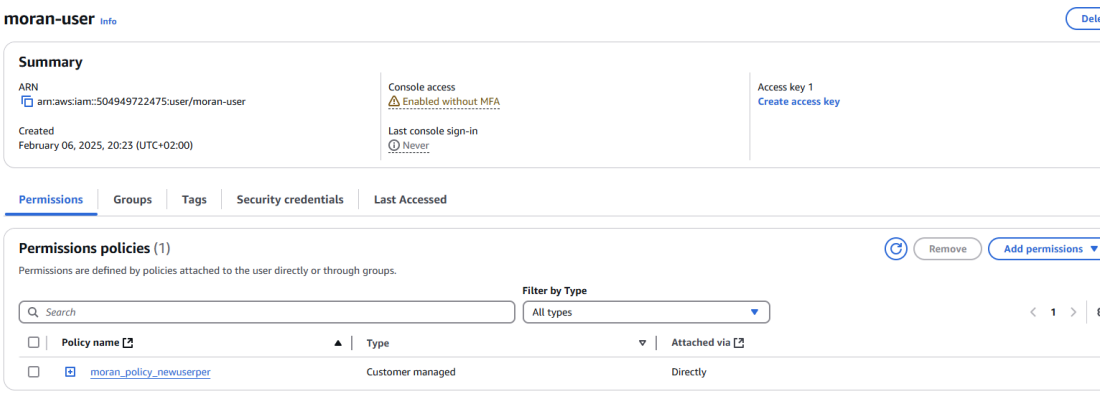






---

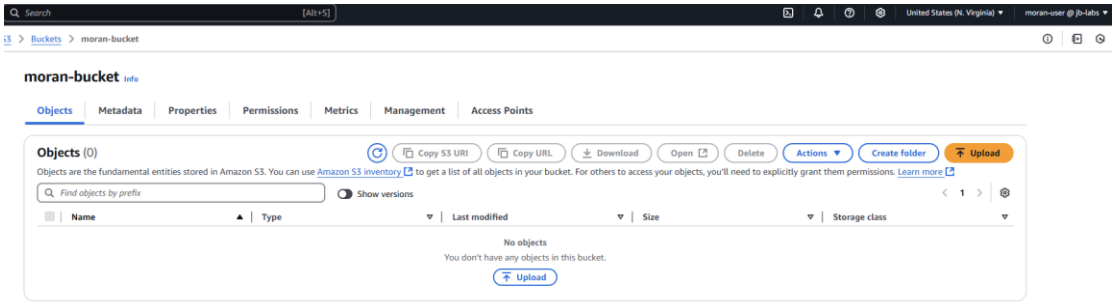
user details:

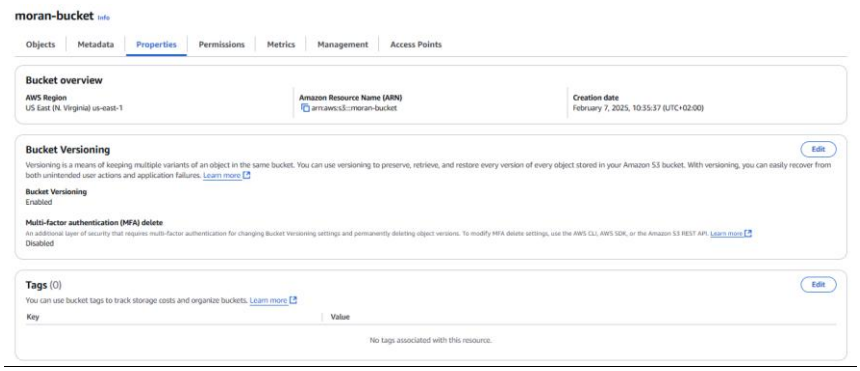


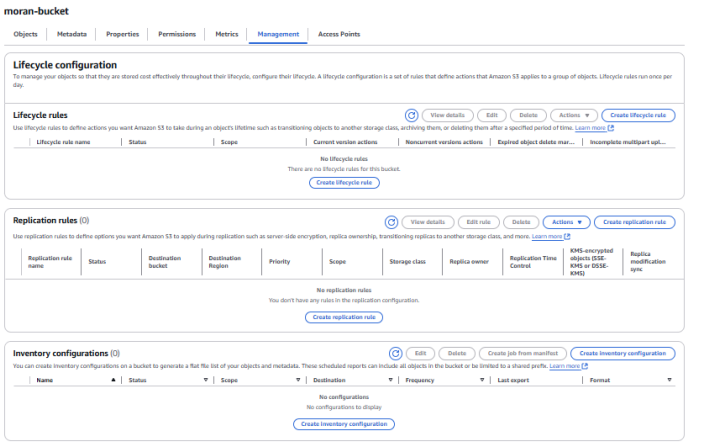
How to verify the user has the correct permissions:

a possible solution can be to login to moran-user and check if the user have access to moran-bucket and do not have permissions to the other buckets.

I login and see this happening:

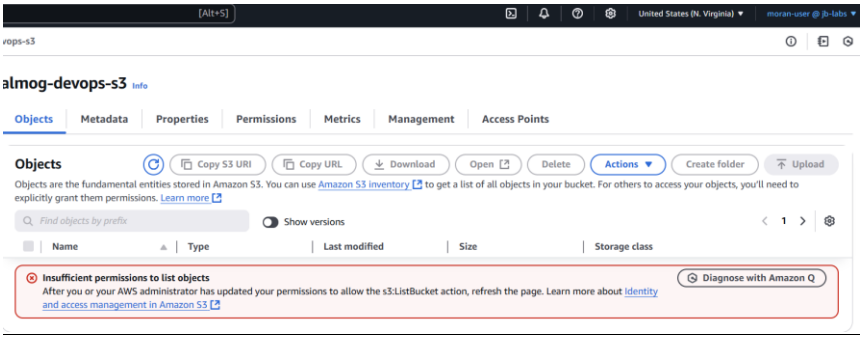


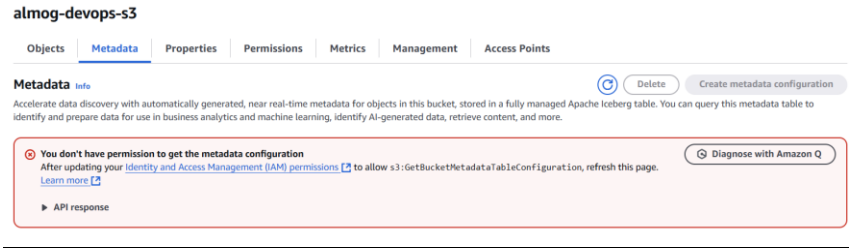


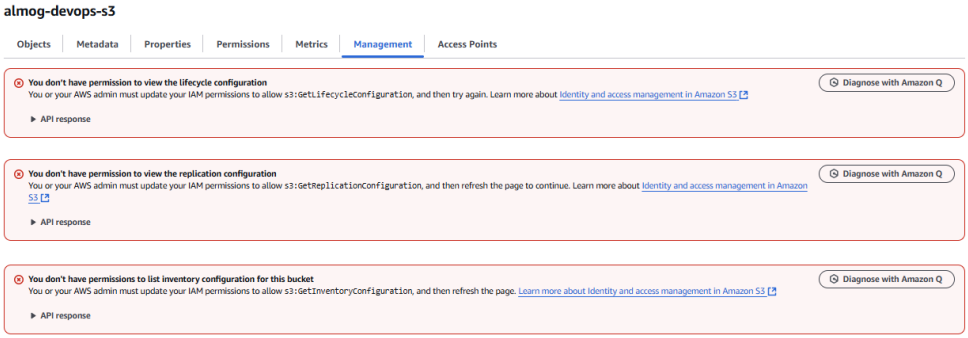


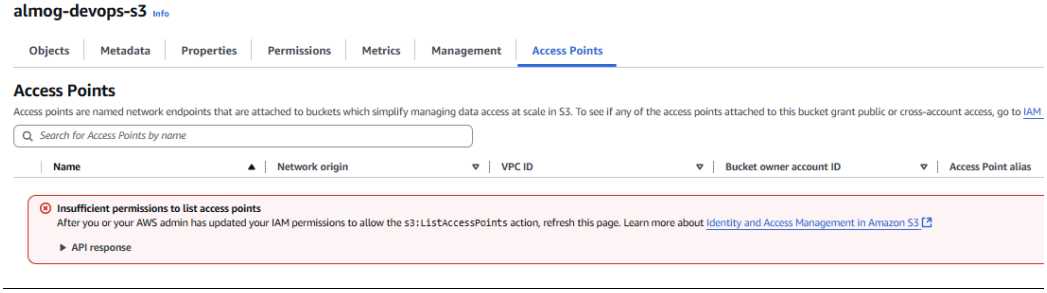
has permission!

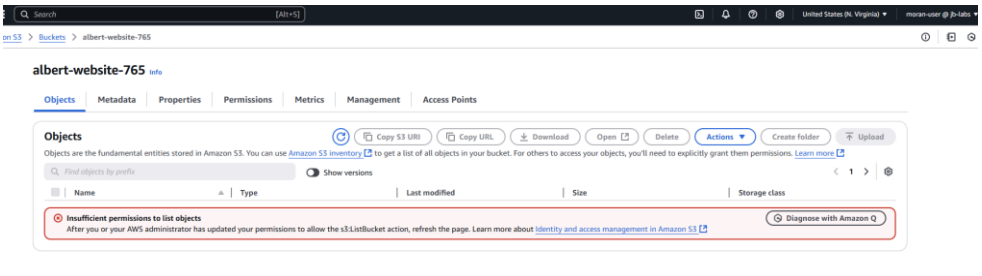
Other buckets:

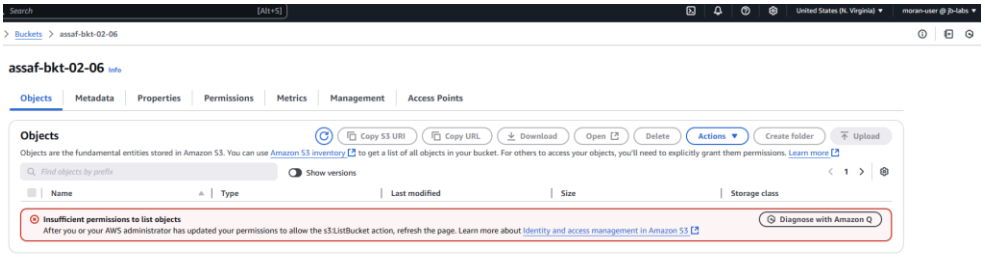






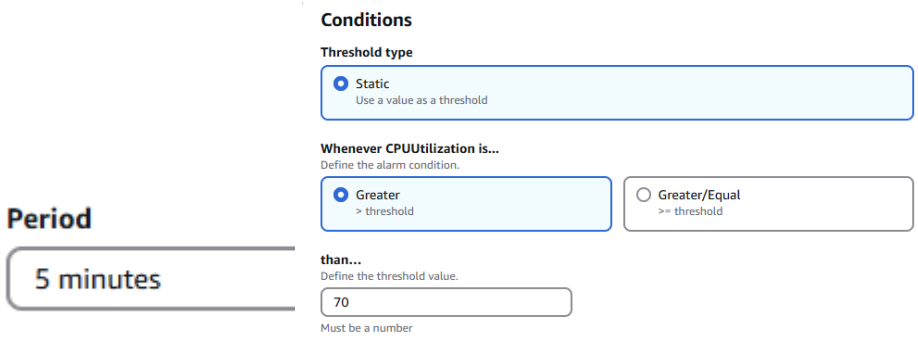




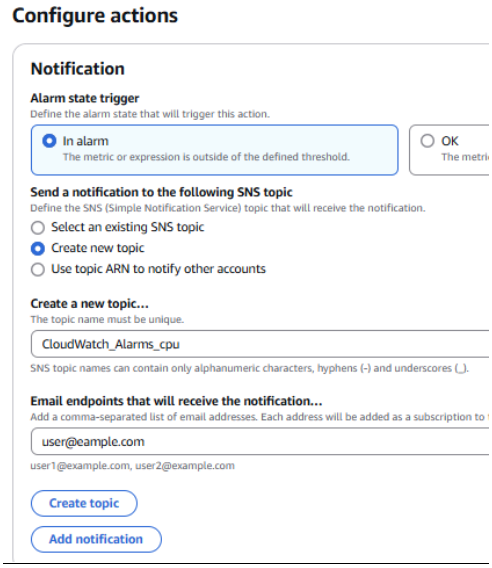


no permission! All the options are block.

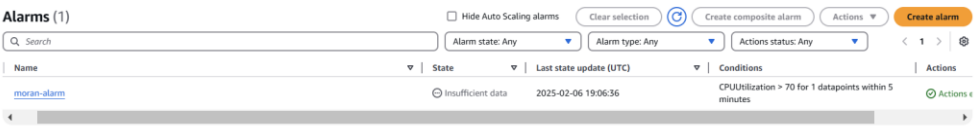
4. **<u>Set Up a CloudWatch Alarm:</u>**
   <u>create -> select metric ec2 -> per instance metric -> CPUUtilization</u>



Configuration of notification:
alarm state trigger- in alarm (when it does outside of the threshold)

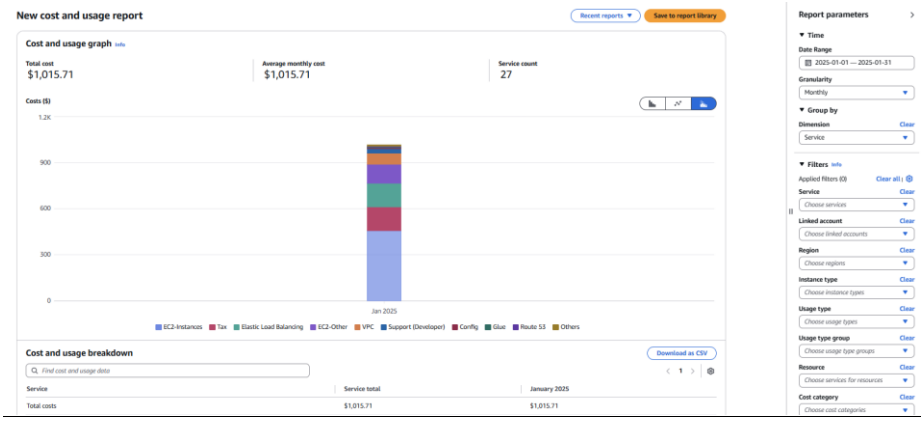


final:

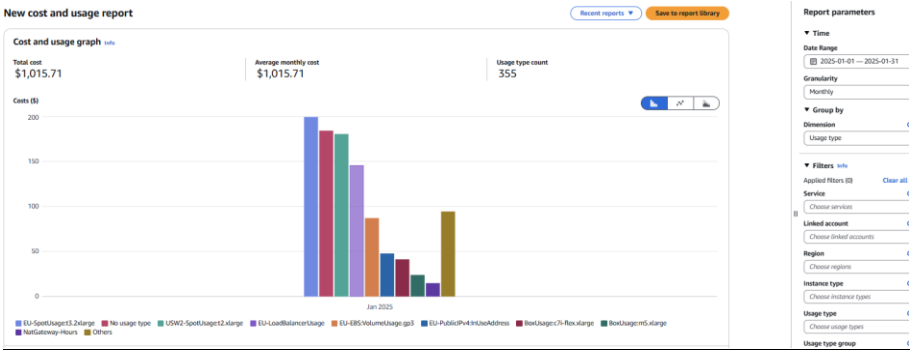


5. **Identify AWS Billing Costs:**

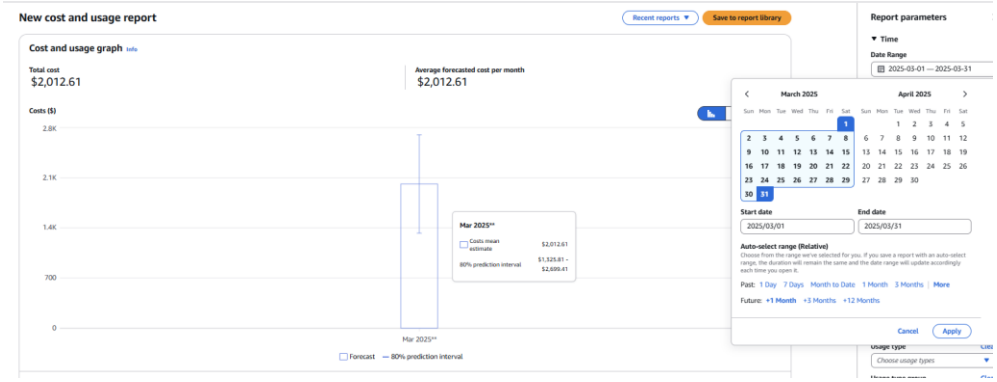
billing details from the last month:



analyze usage:



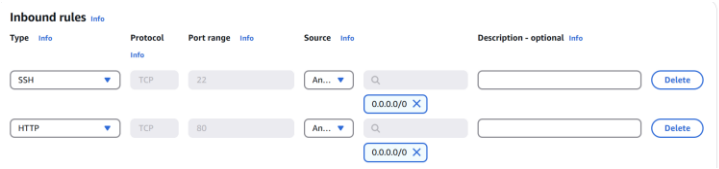
for example, if I filtered with S3 service, they spend 2.98$ on January.
For forecast costs, I chose in data range 1 month in the future – march, the forecast payment will be 2,012.61$ avg.



Section 3: Hands-on advanced:

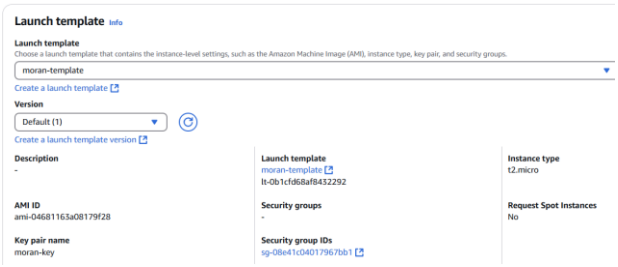
11. Deploy an Auto Scaling Group with a Single EC2 Instance

Creating security group called moran-sg with the inbound rules

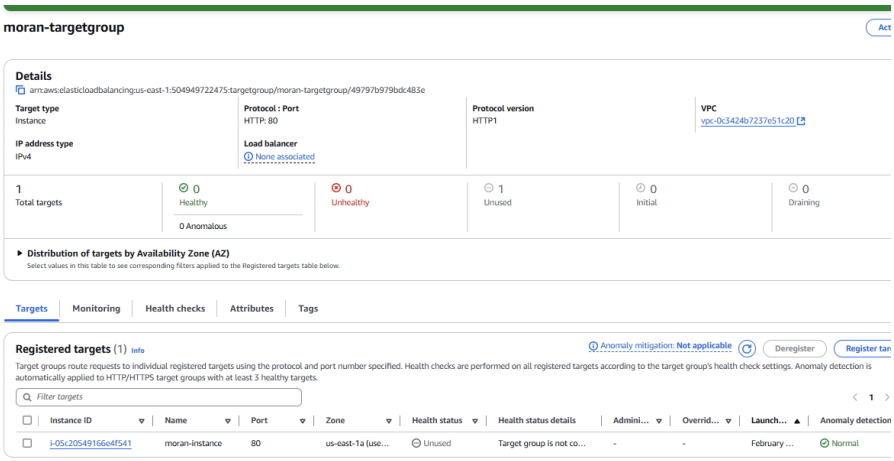


Creating auto scaling:
launch template: moran-template
AMI: Amazon Linux 2 AMI
instance type: t2.micro

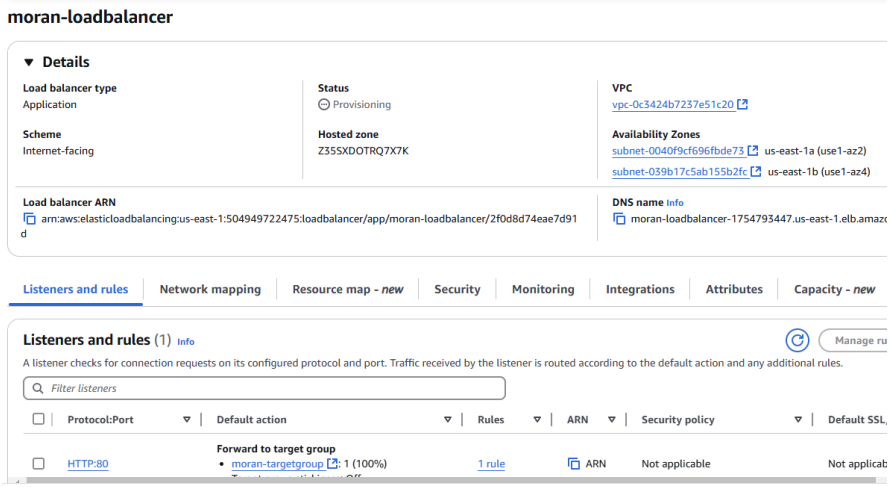
using launch template in the auto scaling



Create target group:

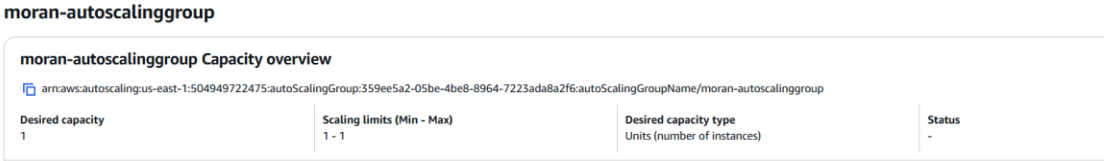


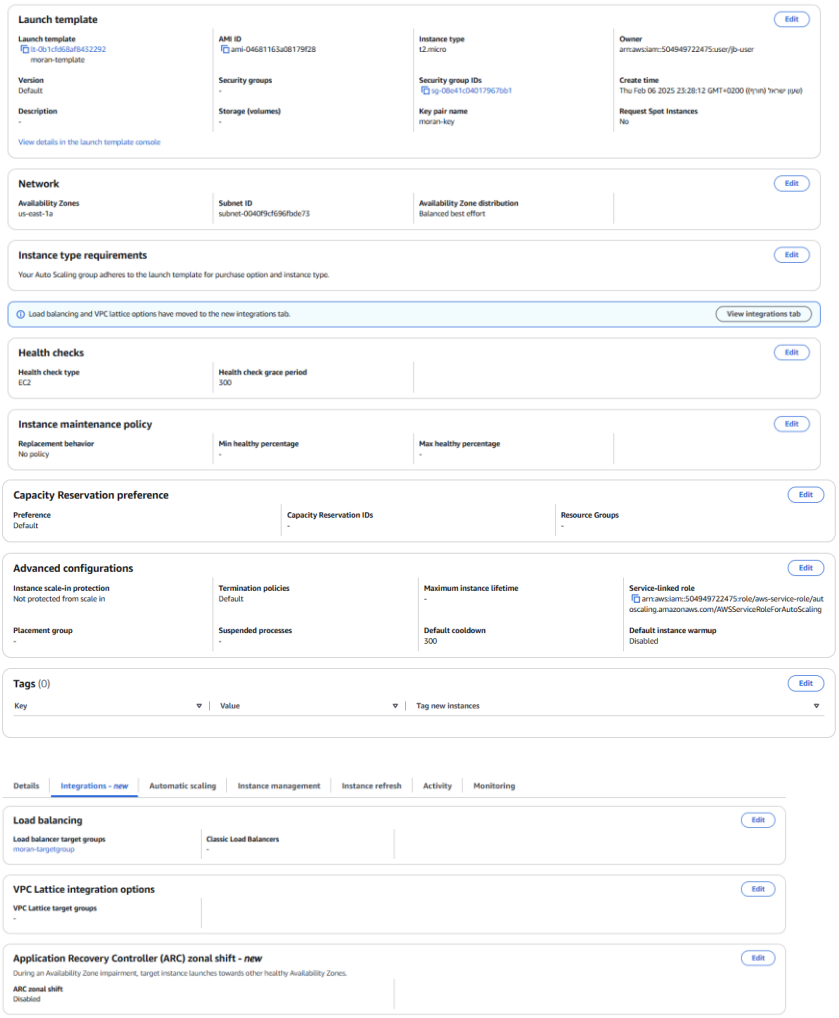
create load balancer with the target group:



Desired capacidy-1

auto scaling group information:



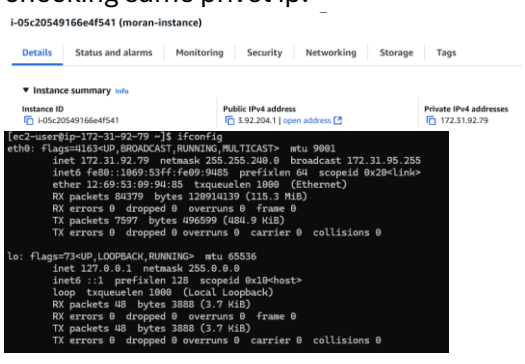


connecting to the machine with ssh:

```
moran@DESKTOP-06F9F43:~$ chmod 400 moran-key.pem
moran@DESKTOP-06F9F43:~$ ls -l | grep -w moran-key.pem
-r-------- 1 moran moran 1678 Feb  6 23:26 moran-key.pem
-rw-r--r-- 1 moran moran   77 Feb  6 23:26 moran-key.pem:Zone.Identifier
moran@DESKTOP-06F9F43:~$ sudo ssh -i "moran-key.pem" ec2-user@ec2-3-92-204-1.compute-1.amazonaws.com
Last login: Thu Feb  6 22:05:02 2025 from bzq-79-177-150-58.red.bezeqint.net
       #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-92-79 ~]$
```

checking same privet ip:

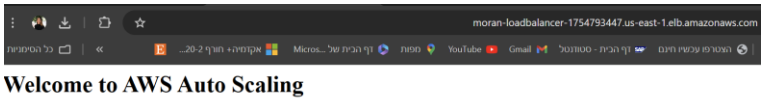


installing and running nginx and check that the curl works:

```
[ec2-user@ip-172-31-92-79 ~]$ systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2025-02-06 22:16:46 UTC; 2min 6s ago
 Main PID: 3616 (nginx)
   CGroup: /system.slice/nginx.service
           ├─3616 nginx: master process /usr/sbin/nginx
           └─3617 nginx: worker process
```

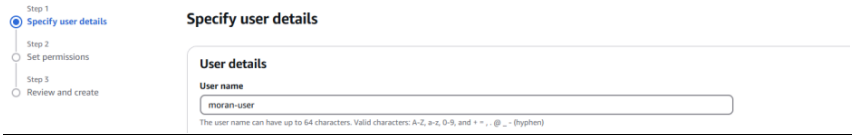
```
[ec2-user@ip-172-31-92-79 ~]$ echo "<h1>Welcome to AWS Auto Scaling</h1>" | sudo tee /usr/share/nginx/html/index.html
<h1>Welcome to AWS Auto Scaling</h1>
[ec2-user@ip-172-31-92-79 ~]$ sudo systemctl start nginx
[ec2-user@ip-172-31-92-79 ~]$ sudo systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-172-31-92-79 ~]$ curl http://localhost:80
<h1>Welcome to AWS Auto Scaling</h1>
[ec2-user@ip-172-31-92-79 ~]$
```

3. Putting DNS name in the browser:
dns name: moran-loadbalancer-1754793447.us-east-1.elb.amazonaws.com

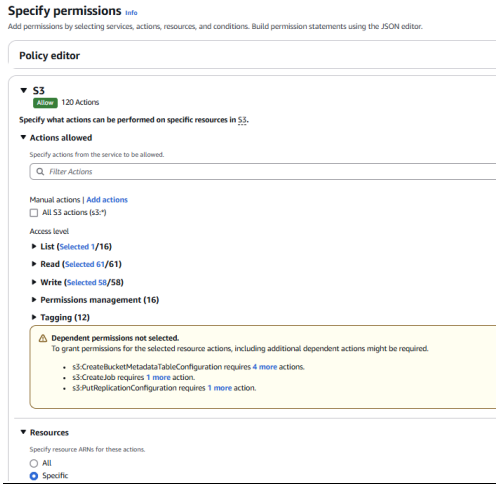


6.

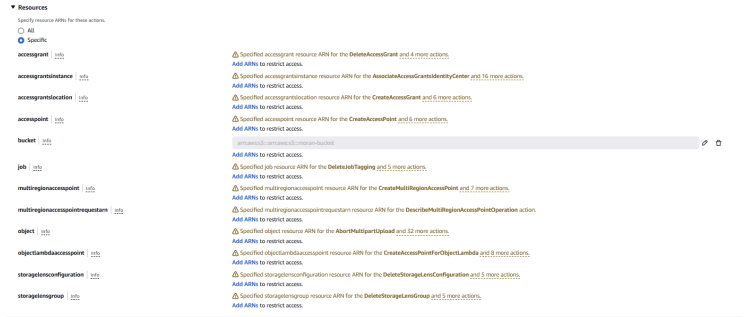
4. **IAM User Setup for S3 Access: - same as section2-(3)**
go to IAM and create new user named moran-user:

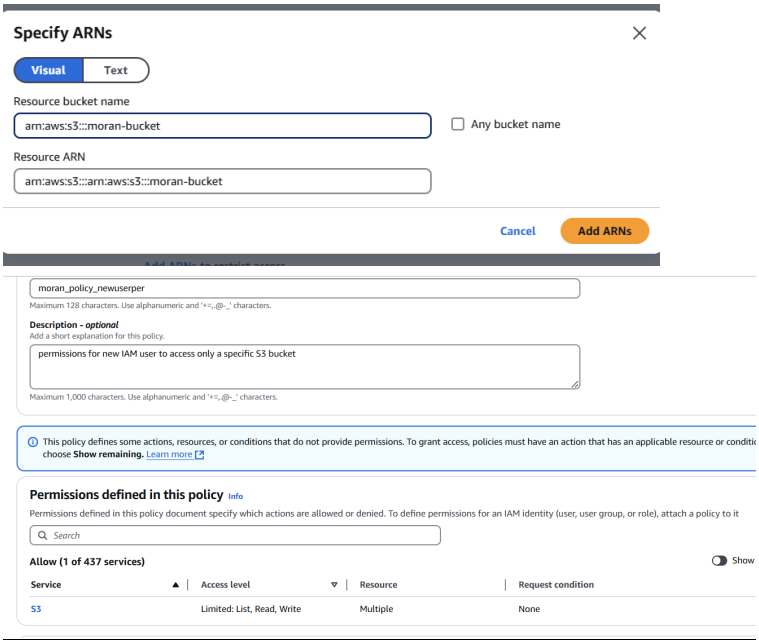


create password, then create policy with specific permissions:
I decided to give read, write and list(only access to ListBucket to see the list buckets) permissions.

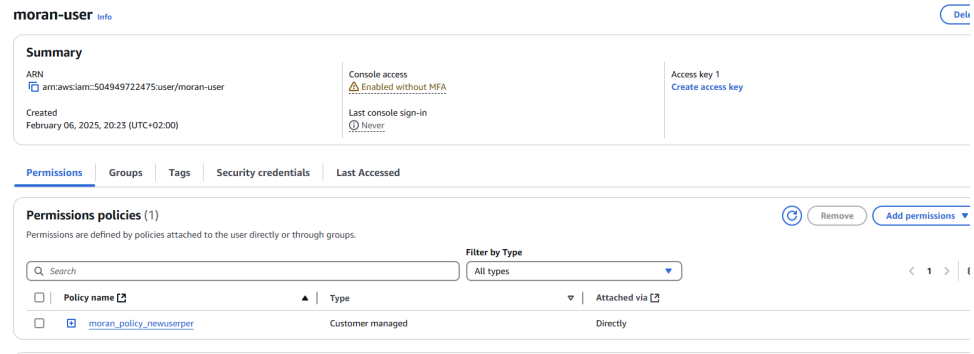


only one specific bucket- moran-bucket:



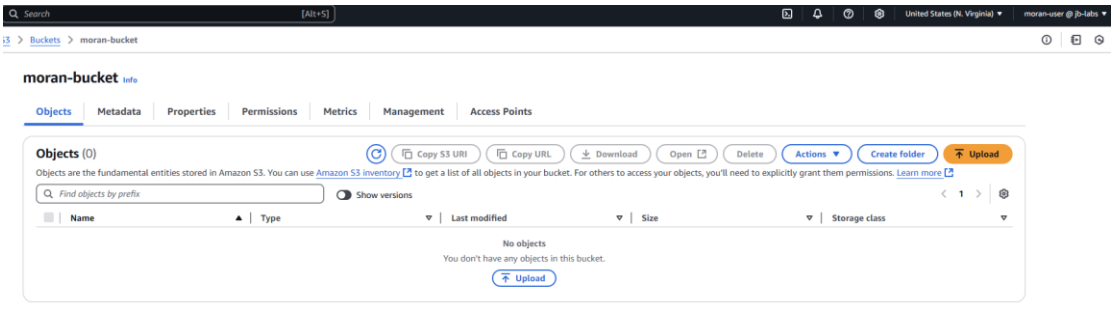


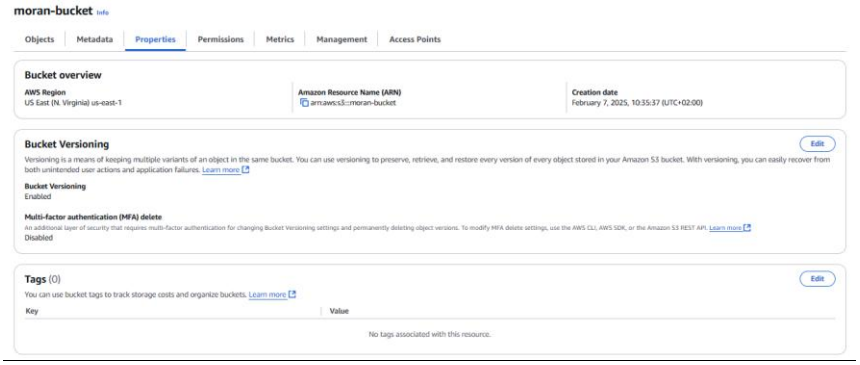
user details + we can see the attached policy:

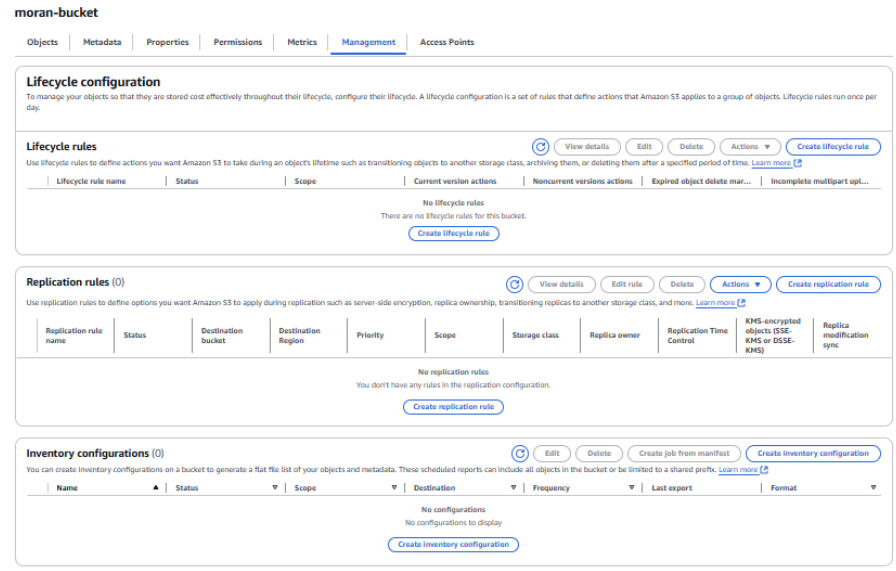


Verify:

login to moran-user and check if I have access only to moran-bucket and not others:

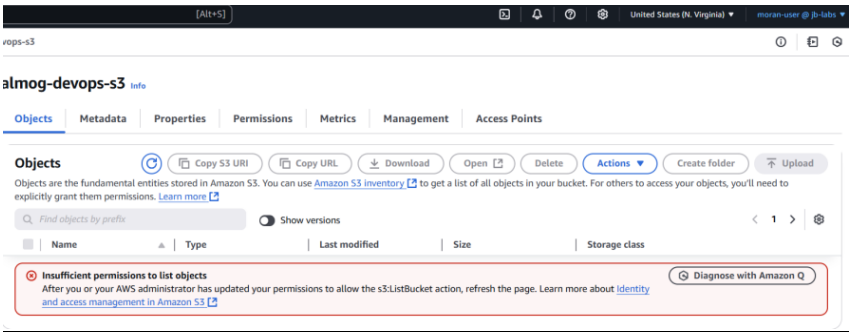


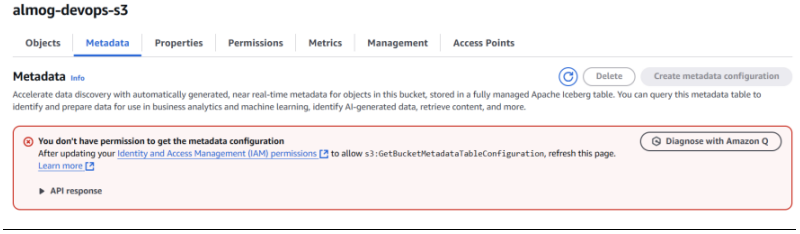


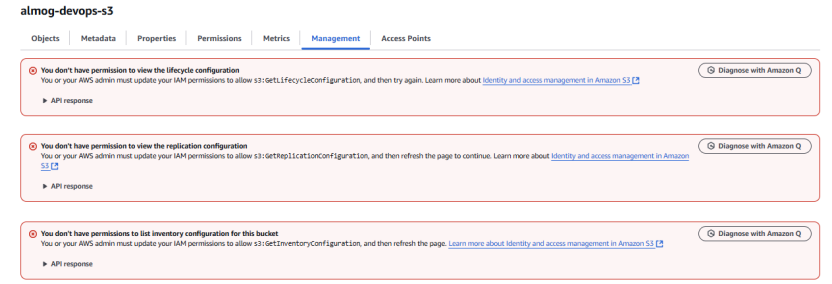


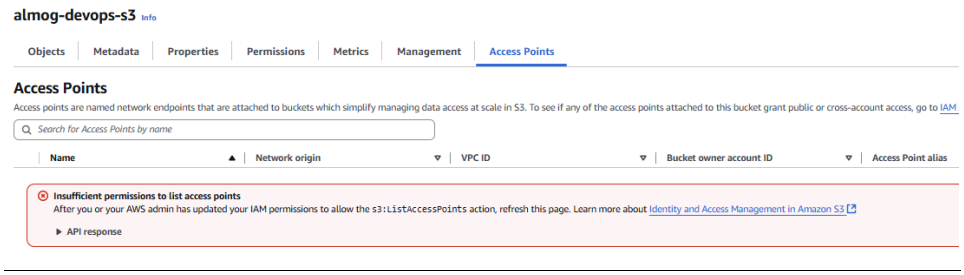
has permission!

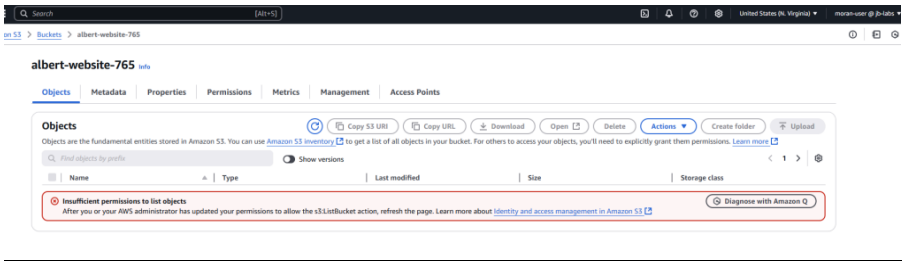
Other buckets:

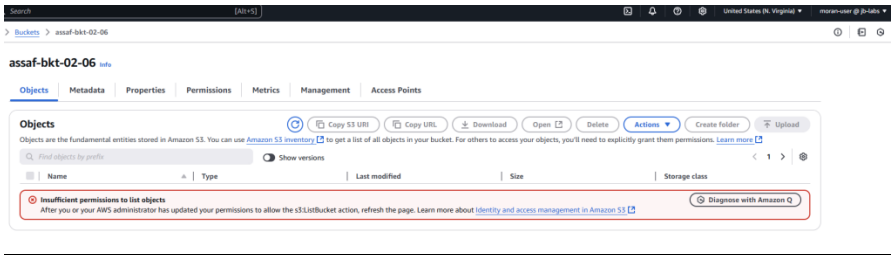






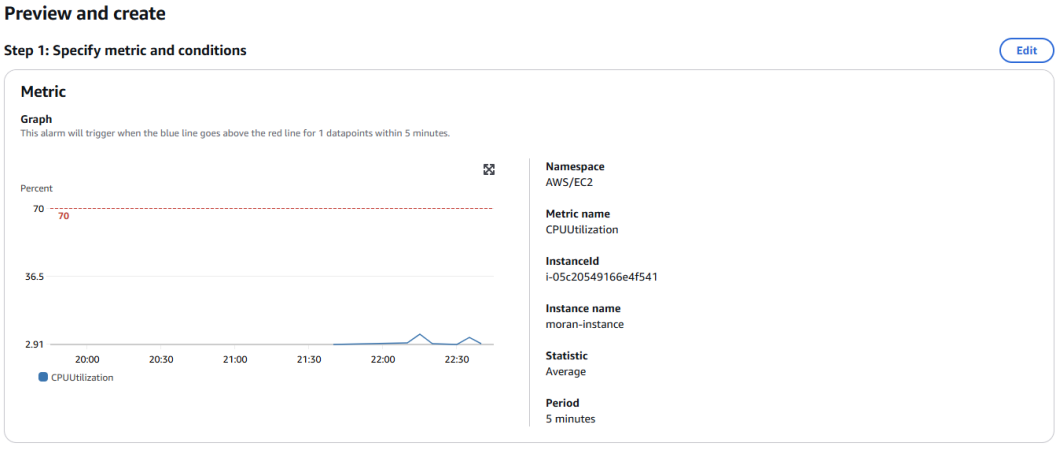






no permission! All the options are block.

5. **Create a CloudWatch Alarm for CPU Usage:**
   CPU utilization exceeds 70% for 5 minutes.



   Configure notifications via email (SNS).

Conditions

Threshold type
Static

Whenever **CPUUtilization** is
Greater (>)

than…
70

▶ **Additional configuration**

Step 2: Configure actions Edit

Actions

Notification
When In alarm, send a notification to "notifications_via_email"

Step 3: Add name and description Edit

Name and description

Name
MoranAlarm

Alarms (1)

Hide Auto Scaling alarms | Clear selection | Create composite alarm | Actions

moran

Alarm state: Any | Alarm type: Any | Actions status: Any

Name	State	Last state update (UTC)	Conditions	Actions
MoranAlarm	Insufficient data	2025-02-06 22:50:48	CPUUtilization > 70 for 1 datapoints within 5 minutes	Actions enabled Warning