AWS Handson Session

Create IAM User

1) Create a User in IAM
2) Click Create User
3) Mention any username you prefer
4) Click on Provide user access to the AWS Management Console – *optional as we are creating an admin user*
5) Select I want to create a IAM user since it is more simple
6) You can click on Autogenerated Password if you are creating for someone else.
7) Since you are creating for your self you can enter your password there.
8) Click on Next

Set Permission to this User-Permission can be set for user or a group and add user to the group.

1. Click on Create group
2. Provide a meaningful group name eg: admin
3. Click Administrator Access policy from Permission Policies
4. Click Create User Group
5. Now add the user to admin group-Click on Next
6. Review the options selected
7. Then click on Create User
8. You can email sign-in instructions or download .csv file

Revisit the configuration after creating the user and group.

Attach Policy to your user

1. Go to IAM > User >Select the user created
2. In Permission Click Add Permission
3. Select Attach policies directly
4. In Permission Policies search for IAMReadOnlyAccess
5. Click on AddPermission
6. Task: Try created a developer group using the newly created user

IAM Roles

1. Go to IAM
2. Click on Roles
3. Create a custom Role-Click Create Role
4. Role is a way to give aws entities do stuff on aws
5. Select AWS Service
6. Select which Service for which we need this role to apply to.
7. Select EC2 and Select Use Case as EC2 and Click on Next
8. Next we need to attach a Policy
9. Attach IAMReadOnlyAccess and click Next
10. Provide a suitable Role Name
11. Then select the trusted entities
12. Click on Create Role

IAM Best Practices

1. Don’t use root account except for AWS Account Setup
2. One Physical User=One Aws User
3. Assign users to group and assign permission to groups
4. Create a strong password policy
5. Use and enforce use of MFA
6. Create and use Roles for giving permissions to AWS services
7. Use Access keys for Programmatic Access
8. Audit permissions of a your account using IAM Credentials Report and IAM Access Advisor
9. Never ever share IAM users and Access Keys

Launching a EC2 instance running on Linux

1. Search for EC2 service
2. Click on Instances and Launch Instances
3. Add Name and tags-DemoInstance
4. Select Base Image for EC2 Instance-Select Amazon Linux AWS
5. In it Select Amazon Linux 2 AMI-which free tier eligible
6. Select Instance Type-t2.micro
7. Key pair to Login to instance-Create a new key Pair
8. Give EC2 Learning
9. Select RSA in Key pair type
10. Private key file format-pem
11. Click Create Key Pair
12. Go to Network Settings

13. Rest options as it is and Allow HTTP traffic from the internet
14. Configure storage- 8 Gib gp 2 as Root Volume
15. Fill User Data

```
#!/bin/bash
# install httpd (Linus 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo “<h1>Ratheesh</h1>” > /var/www/html/index.html
```

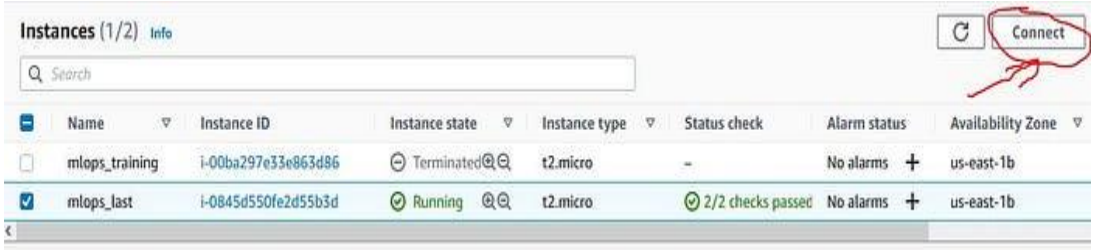
16. Summary: Just 1 instance and Review everything
17. Launch Instance

Security Group SSH into EC2

1. Click on Security Groups in Network and Security
2. Identify the security group ID:_____
3. Click on Inbound Rules
4. Find the Inbound Rule Types: HTTP and SSH
5. Click on Edit Inbound Rules
6. Mention the Port Range available: _____ and _____
7. Delete the HTTP Rule Type by clicking on Delete
8. Click on Save the rules
9. Mention whether the Page is loading? Yes/No
10. Click on Outbound Rules
11. Identify the security group ID created for data going out of EC2 instance

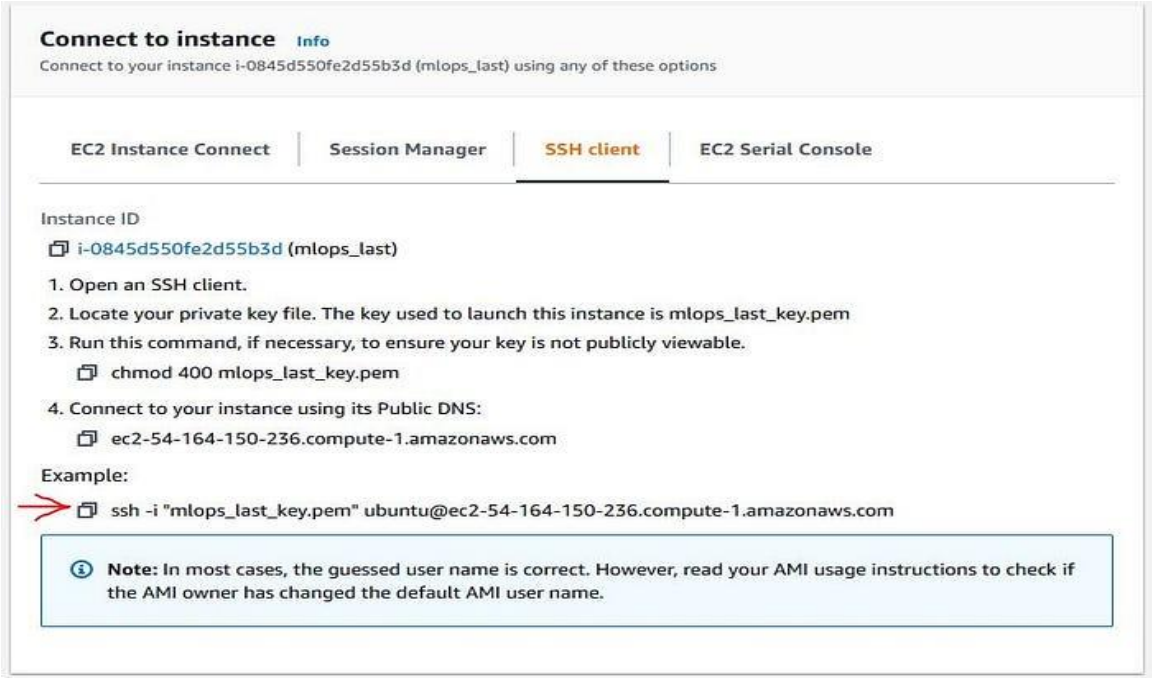
Private/Public/Elastic IP

1. Click on Instance
2. Click on Connect

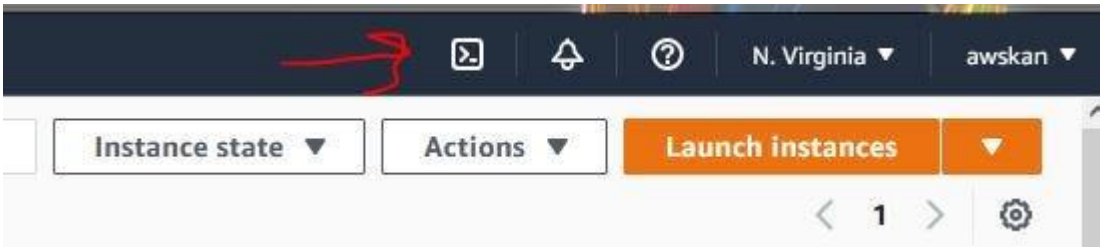


3. Click on SSH Client

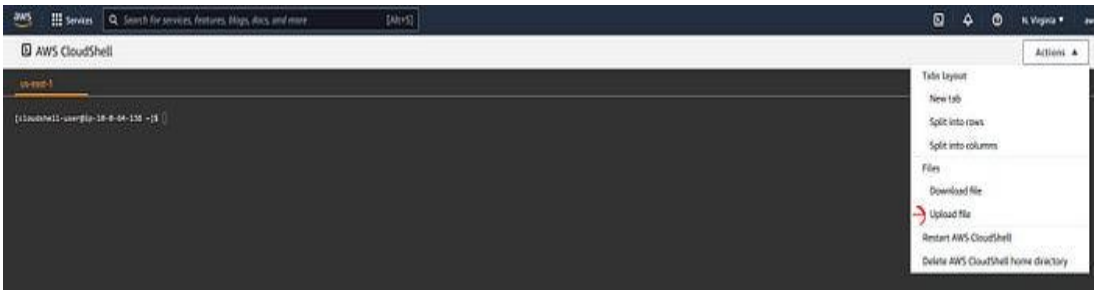
4. Copy the command



5. Click on Cloudshell icon



6. Upload the Key Pair download using the Upload file option



7. Provide permission to .pem file- chmod 400 EC2KeyPair.pem

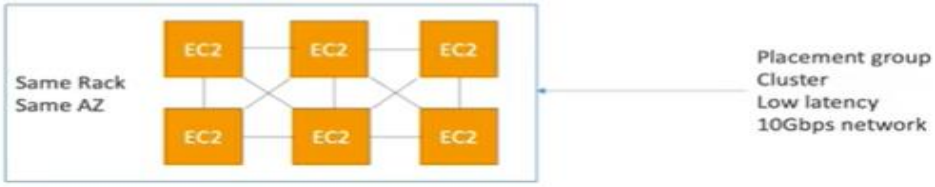
8. Paste the command copied from SSH Client
9. You have logged into EC2 instance from AWS CloudShell.
10. Try connecting the private ip for your instance and see what message you get
11. Stop the Instance and try connect using the same Public IP and see what happens
12. Every time the instance is stopped and restarted the Public IP changes and to avoid this we can use the Elastic IP
13. To create a Elastic IP you can click on the Elastic IP and Allocate Elastic IP Address
14. Ensure the Resource Type Selected is Instance
15. Select your Instance
16. Select your private IP address
17. Click on Associate
18. Go to Instances and verify if elastic ip is reflecting in details tab.

Placement Groups

Sometimes you want control over the EC2 instance placement strategy. For this we can use Placement Group.

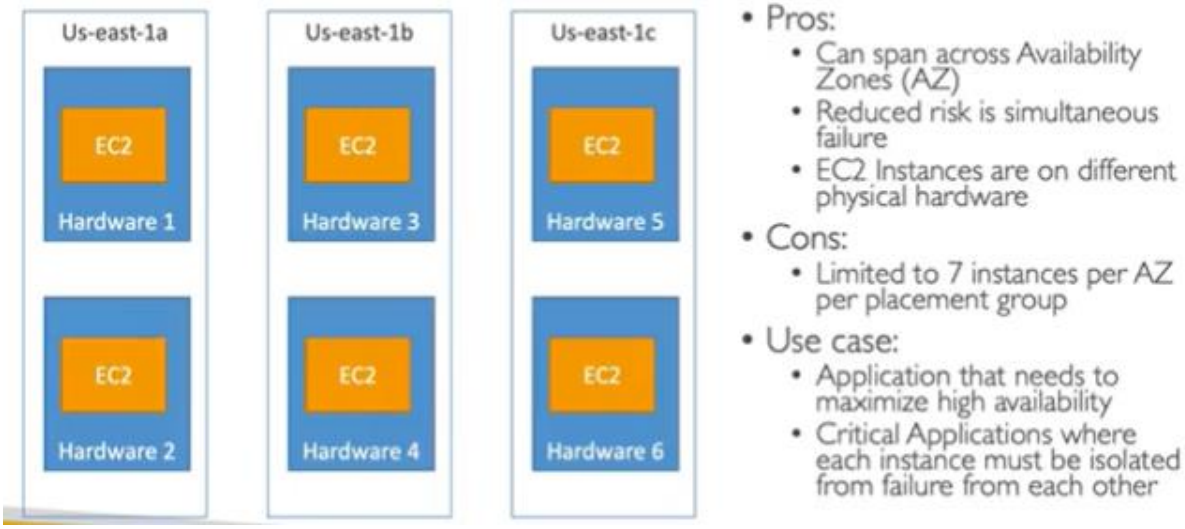
We can have different strategies for Placement Groups:

1. Cluster: cluster instances into a low latency group in a single availability zone



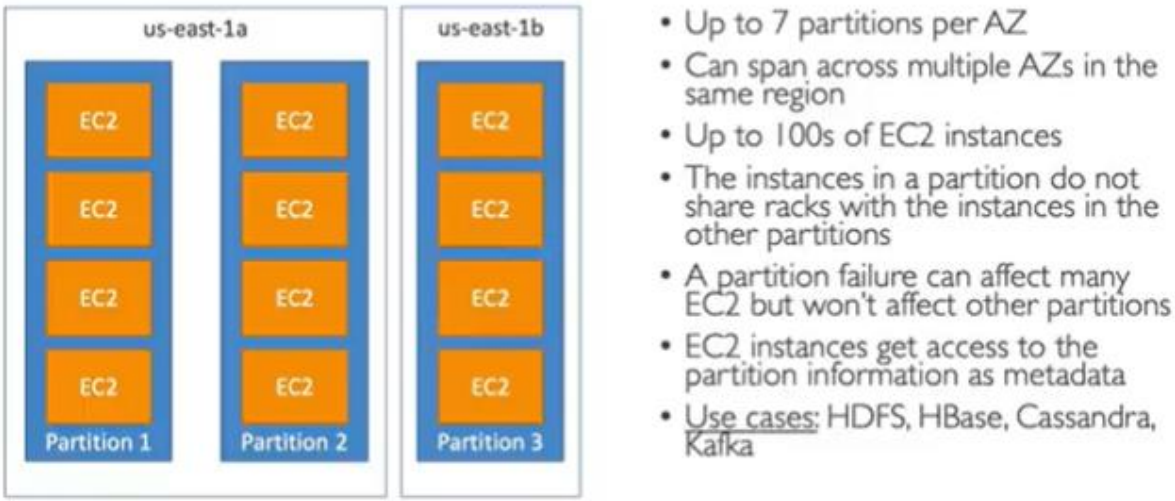
- Pros: Great network (10 Gbps bandwidth between instances)
- Cons: If the rack fails, all instances fails at the same time
- Use case:
  - Big Data job that needs to complete fast
  - Application that needs extremely low latency and high network throughput

2. Spread: spread across underlying hardware(max 7 instances per group per



AZ)-critical applications

3. Partition: spreads instances across many different partitions(which rely on different sets of racks) within an AZ. Scales to 100s of EC2 instances per group (Hadoop, Cassandra, Kafka)



EC2 placement Groups Handson

1. To create Placement Group Click on Placement Group
2. Click on Create Placement Group
3. Mention name as my-performance-group
4. Select Placement Startegy as Cluster
5. Click on Create Group

6. Create another group as my-critical-group
7. Select Placement Strategy as Spread
8. Select Spread Level as Rack (No Restriction)
9. Click on Create Group
10. Create another group as my-distributed-group
11. Select Placement Strategy as Partition
12. Select Number of Partitions to 4
13. Click on Instances and Go to Advanced Details
14. You will find the Placement Group Name

Elastic Block Store

- An EBS (Elastic Block Store) Volume is a network drive you can attach to your instances while they run
- It allows your instances to persist data, even after their termination
- They can only be mounted to one instance at a time (at the CCP level)
- They are bound to a specific availability zone

- Analogy: Think of them as a "network USB stick"
- Free tier: 30 GB of free EBS storage of type General Purpose (SSD) or Magnetic per month

EBS Volume

- It's a network drive (i.e. not a physical drive)
  - It uses the network to communicate the instance, which means there might be a bit of latency
  - It can be detached from an EC2 instance and attached to another one quickly

- It's locked to an Availability Zone (AZ)
  - An EBS Volume in us-east-1a cannot be attached to us-east-1b
  - To move a volume across, you first need to snapshot it

- Have a provisioned capacity (size in GBs, and IOPS)

EBS Handson

1. Click on Instances and Check the Storage Tab

2. Under Block Devices you will Volume ID with Device Name and Volume Size
3. Click on Volume ID
4. Check the Volume State should be In-use.
5. Check Attached Instances

Attached Instances
i-08f20d251679e55eb (My First Instance): /dev/xvda (attached)

6. Click on Volumes under Elastic Block Store
7. Click on Create Volume to create one more volume
8. Change Size GiB to 2
9. Select the same Availability Zone (You can find it from Instances and Network Tab)
10. Click on Create Volume
11. Click on Volume and Check Volume State it should be available state.

Volume state
Available

12. Click on Attach a Volume

Elastic File System

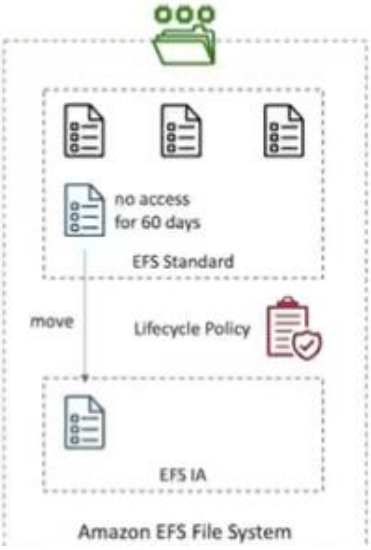
- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use

- Use cases: content management, web serving, data sharing, Wordpress
- Uses NFSv4.1 protocol
- Uses security group to control access to EFS
- Compatible with Linux based AMI (not Windows)
- Encryption at rest using KMS

- POSIX file system (~Linux) that has a standard file API
- File system scales automatically, pay-per-use, no capacity planning!
- EFS Scale
  - 1000s of concurrent NFS clients, 10 GB+ /s throughput
  - Grow to Petabyte-scale network file system, automatically
- Performance Mode (set at EFS creation time)
  - General Purpose (default) – latency-sensitive use cases (web server, CMS, etc…)
  - Max I/O – higher latency, throughput, highly parallel (big data, media processing)
- Throughput Mode
  - Bursting – 1 TB = 50MiB/s + burst of up to 100MiB/s
  - Provisioned – set your throughput regardless of storage size, ex: 1 GiB/s for 1 TB storage
  - Elastic – automatically scales throughput up or down based on your workloads
    - Up to 3GiB/s for reads and 1GiB/s for writes
    - Used for unpredictable workloads

EFS – Storage Classes

- Storage Tiers (lifecycle management feature – move file after N days)
  - Standard: for frequently accessed files
  - Infrequent access (EFS-IA): cost to retrieve files, lower price to store. Enable EFS-IA with a Lifecycle Policy
- Availability and durability
  - Standard: Multi-AZ, great for prod
  - One Zone: One AZ, great for dev, backup enabled by default, compatible with IA (EFS One Zone-IA)
- Over 90% in cost savings



EFS Handson

1. Go to EFS Console by typing EFS in Service Search
2. Click on Create File System
3. Click on Customize
4. You can keep Name of File system empty
5. You can select Standard
6. Enable Automated Backup
7. In Lifecycle management you automated movement of data less costly storage options for data which persisted for longer period
8. In Performance Mode Select Enhanced and Elastic and click on Next
9. In Network keep Default VPC selected
10. Under Mounts create a security group for EFS
11. Remove the default security group and select the security created for EFS
12. File System Policy can be ignored for now
13. Go ahead and click on Create
14. Create a new instance follow the steps provided for EC2 instance creation don't create key pair this time
15. Click on Edit

16. Go to Network Settings and Edit a select one subnet from there
17. In Filesystem you will get EFS option enabled
18. Clicked on Shared File System
19. Create just 1 instance and click on Launch
20. Create Another Instance and use a different subnet this time
21. Select the Security Group for Second Instance-launch-wizard 2
22. Click on launch the instance
23. In Instances you should get two instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Instance A	i-027141f7b4fc8e42e	Running	t2.micro	2/2 checks passed	No alarms
Instance B	i-04ab8dcc678a1e7c5	Running	t2.micro	Initializing	No alarms

24. Click on Connect and Select EC2 Instance Connect
25. Repeat the same for Instance B also.
26. You type ls /mnt/efs/fs1/ and type sudo su and echo “Hello World” > /mnt/efs/fs1/hello.txt
27. In Second instance also you will same file there also

Clean Up Session

1. Go to File System > Actions > Delete
2. Go to EC2 instance Terminate any newly created Instances
3. Any Volumes you had created you can terminate it.