Computing in AWS

Analytics Tensor

Mahesh KC

mahesh.kc@analyticstensor.com

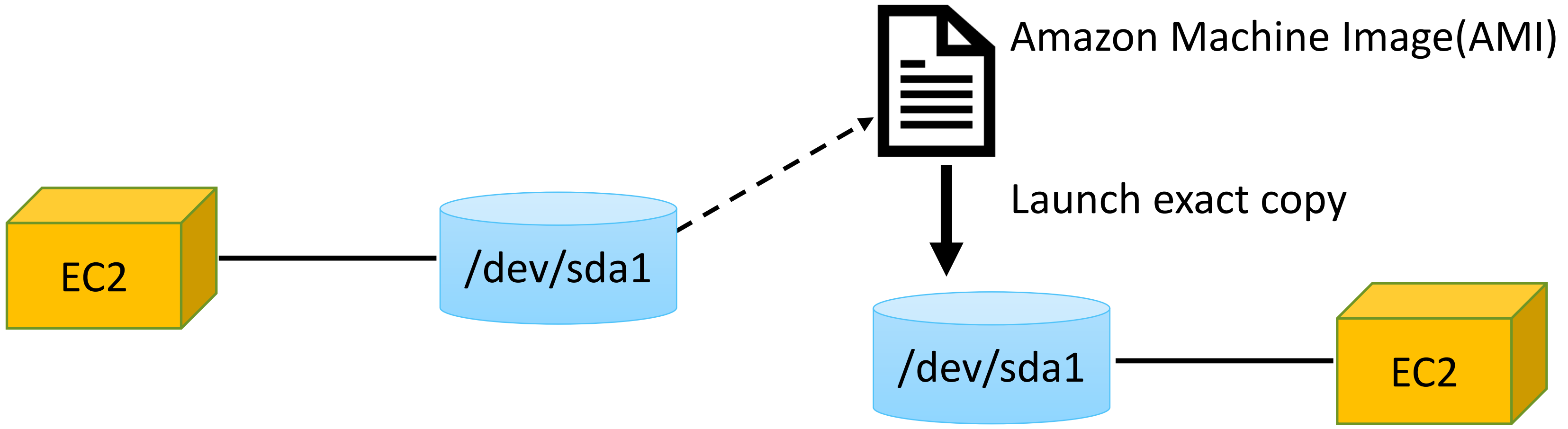
https://analyticstensor.com

Amazon Elastic Compute Cloud (EC2)

- EC2 provides Virtual Machine which is also known as Instances.
- One virtual machine (VM) an instance of EC2.
- It is based on Xen Hypervisor.
- It comes with price based on combinations of CPU, memory, disk, IO.
- We can launch from 1 to 1000's of instances. But there are service limit based on the account. This is known as soft limit. We can submit a ticket to AWS support to increase the limit of resources.
- EC2 instances are disposable. When we have an error on some instances we can dispose those and re-instances new.
- Best practice is to replace the machine instead of patching.
- It comes with various billing models:
  - On-Demand
  - Reserved
  - Spot

Amazon Machine Image (AMI)

- Amazon Machine Image (AMI) is a bit-for-bit copy of root volume. Whatever we have on the root device volume for e.g. Operating System(OS), applications, configuration, users etc. or anything as the part of root volume when the machine was started will become part of Amazon Machine Image (AMI).
- We can choose that AMI and launch clones of original. We'll exact copy of that instances.
- The machine images are stored in S3 but we don't direct access to that bucket or objects. It is automatically handled by AWS.



Demo: Launching Linux Instances.

- Choose EC2.
- Choose Instances. We can choose AMI provided by different vendors. OR Under Images on left side, we can see the AMIs.
- Choose the AMI and choose Next.
- For this tutorial, we chose t2.micro, since it is free for 1 years when we sign-up AWS account.
- We can choose number of instances.
- We can also choose network (Haven't described yet!)
- Choose IAM role.
- Click Next.
- We can add storage. The root volume is 8GB. Choose Magnetic.
- Tag helps to identify the machine when there are lot of instance so that we can filter based on application. Tag helps to organize the infrastructure.
- Choose Add Tag. Type "Linux Machine" in the tag.
- Choose the security Group. Choose SSH with source my ip. Allow port 22, 80, 443 to open.
- Click review and launch.
- Create Key/Pair and download. It will download .pem file.
- The instance should be running.
- Copy the public IP address of the instance. chmod the the pem file with 400.
- Type: ssh –i pemfile ec2-user@ip_address. (The super user is ec2-user. Find on documentation for other instances.)
- You should be able to login to t2.micro running Amazon Linux.

EC2 Types: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html

Key Pairs

- Public and Private keys
- 2048-bit SSH-2 RSA
- AWS keeps public keys
- AWS doesn’t keep private key

Key/Pairs are a way to authenticate into Operating System of AWS EC2 instances.

Linux

- When we launch Linux EC2 instance AWS will take that public key and add to .ssh/authorized_keys file. In ubuntu, it will be added to ubuntu user, in Amazon Linux it will add to EC2 user.
- When we try to login to remote machine or ssh then we’ll use our private key to authenticate over that OS.

Windows

- In windows, the administrator password will be encrypted using the public key. And we’ll our private key to decrypt the window administrator password.

Instance Metadata Service

- It is a services from which we can retrieve information about EC2 instances.
- The instance metadata is available in http://169.254.169.254/latest/meta-data/
- It is only available with in the instances. So, it is not available outside of EC2. In Linux, we can use curl, powershell in window.
- We can retrieve information such as: AMI id, hostname, instance-id, instance-type, private IP address
- We can use these services with script so that we can bootstrap the instances for configuring automatically. When the instance boot then it call the metadata services and get information about instance and use those information to automatically configure and bootstrap these instances to install software, configure software etc. automatically.

Demo: Instance Metadata Service

- ssh to the instances. i.e. ssh –i pemfile ec2-user@ip_address
- Type: curl http://169.254.169.254/latest/meta-data/. Press Enter. It will show list e.g. ami-id, mac, network etc.
- Type: curl http://169.254.169.254/latest/meta-data/ami-id. It will print ami-id.
- Type: curl http://169.254.169.254/latest/meta-data/instance-id. It will print instance-id.
- We can create shell script to get information. For e.g. type: export instance_id=$(curl curl http://169.254.169.254/latest/meta-data/instance-id) . Type: echo$ instance_id. It will retrieve the instance id.

This how we can automate and get information using instance metadata services.

Demo: Stopping and Terminating Instances

Stop: Stopping the machine to eliminate charges.

Terminate: Terminate forever. It will gone. The EBS volume will be deleted.

Choose instances-> Click Stop.

Choose instances-> Click Terminate.

Once the machine is terminated, it will be delete from EC2 instances.

EC2 Billing Option

https://aws.amazon.com/ec2/pricing/

Reserved Instances

- Up-front payment
- 1 or 3 year term
- Pay hourly fee regardless of use
- Guaranteed availability
- Significant savings over on-demand
- Best for:
  - Constant applications
  - Databases

Spot Instances

- Fee based on supply/demand. For e.g. when we the EC2 is ideal then we can leverage to spot market. We'll specify the price.
- Hourly fee fractions of on-demand
- Specify maximum spend
- Instance can be terminated by AWS
- Best for:
  - Temporary works
  - Batch processing
  - Testing/experimentation
  - Create some year. For e.g. End-of-year billing reports, Quarterly sales reports etc.
  - Apps must tolerate interruption.

Web Application Hosting in AWS Cloud

Traditional Web Server Architecture

- https://www.slideshare.net/dsbw2012/unit05architecture-13708969

AWS Web Server Architecture

- https://d1.awsstatic.com/whitepapers/aws-web-hosting-best-practices.pdf
- https://aws.amazon.com/web-applications/gsg-webapps-linux/

Web server Architecture

https://docs.mattermost.com/overview/architecture.html

Install Apache HTTP Server in RedHat

- https://access.redhat.com/documentation/en-us/jboss_enterprise_application_platform/6.3/html/administration_and_configuration_guide/install_the_apache_httpd_in_red_hat_enterprise_linux_with_jboss_eap_6_rpm
- https://linuxconfig.org/installing-apache-on-linux-redhat-8

Install Anaconda

- https://docs.anaconda.com/anaconda/install/linux/

Create a Classic Load Balancer

- https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-getting-started.html
- https://www.youtube.com/watch?v=VIgAT7vjol8