Module 6: Compute

Thursday, September 11, 2025 7:39 PM

A. Learning Outcomes (LOs):

- Provide an overview of different AWS compute services in the cloud
- Demostrate why to use Amazon Elastic Compute Cloud (Amazon EC2)
- Identify the functionality in the EC2 console
- Perform basic funstions in EC2 to build a virtual computing environment
- Identify Amazon EC2 cost optimization elements
- Demonstrate when to use AWS Elastic Beanstalk
- Demonstrate when to use AWS Lambda
- Identify how to run containerized applications in a cluster of managed servers.

B. Compute Services Overview:

- Amazon Web Services (AWS) offers many compute services:

+ Amazon Elastic Compute Cloud (Amazon EC2) provides resizable virtual machines.
+ Amazon Elastic Container Service (Amazon ECS) is a container orchestration service that supports Docker.
+ Amazon Elastic Container Registry (Amazon ECR) is used to store and retrieve Docker images.
+ AWS Elastic Beanstalk provides a simple way to run and manage web applications.
+ AWS Lambda is a serverless compute solution, you pay only for the compute time that you use.
+ Amazon Elastic Kubemetes Service (Amazon EKS) enables users run managed kubernetes on AWS
+ AWS Fargate provides a way to run containers that reduce the need for users to manage servers or clusters

Services	***Key Concepts***	***Characteristics***	***Ease of use***
Amazon EC2	• IaaS (Means provider repsonsible for physical and user control software • Instance-based • ***Virtual machines***	Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
AWS Lambda	• **Serverless computing** • Function-based • Low-cost	• Write and deploy code that executes on aschedule or that can be triggered by events • Use when possible (architect for the cloud)	A relatively new concept for many IT staff members, but easy to use after you learn how
• Amazon ECS • Amazon EKS • AWS Fargate • Amazon ECR	• Container-based computing • Instance-based	Spin up and execute jobs more quickly	AWS Fargate reduces administrative overhead, but user can use options for controlling.
AWS Elastic Beanstalk	• PaaS • For web applications	• Focus on code (building application) • Can easily tie into other services - database, DNS, etc.	Fast and easy to get started

- Choosing rhe optimal compute service:
  + The optimal compute service or services that user will depend on their use case.
  + Some aspects to consider:
    * What is the application design?
    * What are the usage patterns?
    * Which configuration settings will user want to manage?
  + Selecting the wrong compute solution for an architecture can lead to lower performance efficiency: A good starting place - Understand the available compute options

C. Amazon Elastic Compute Cloud (Amazon EC2)

- Examples uses of Amazon EC2 instances: Application/Web/Database/Game/Mail/Catalog/File/Computing/Proxy server
- Amazon Elastic Compute Cloud (Amazon EC2):
  + Provides virtual machines - referred to as EC2 instances - in the cloud
  + Gives full control over the guest operating system (Windows/Linux) on each instance
- User can launch any instances of any size into an AZ (Availability Zone) anywhere in the world
  + Launch instances from Amazon Machine Images (AMIs)
  + Launch instances wirh a few clicks or a line of code, and they are ready in minutes.
- User can control traffic to and from instances.
- Launching an Amazon EC2 instance: The section of the module walks through nine key decisions to make when creating an EC2 instance by using the AWS Management Console Launch Instance Wizard

1. Select an AMI

- Amazon Machine Image (AMI);
  * Is a template that is used to create an EC2 instance (which is virtual machine, or VM, that runs in the AWS Cloud)
  * Contains a Windows or Linux operating system.
  * Often also has some software pre-installed

- AMI choices:
  * Quick Start - Linux and Windows AMIs that are provided by AWS
  * My AMIs - Any AMIs that you created
  * AWS Marketplace - Pre-configured templates from third parties
  * Community AMIs - AMIs shared by others, use at user's risk

2. Select an instance type

- Consider user's case: How will the EC2 instance create be used?
- The instance type that users choose determines the memory (RAM), processing power (CPU), disk space and disk type (Storage), and Network Performance
- Instance categories:
  * General purpose
  * Compute optimized
  * Memory optimized
  * Storage optimized
  * Accelerated computing
- Instance types offer family, generation, and size
- Instacne types: Networking features:
  + The network bandwidth (Gbps) varies by instance type (See Amazon EC2 Instance types to compare)
  + To maximiza working and bandwidth performance of instance type:
  + Enhanced networking types:
    * Elastic Network Access (ENA): Supports network speeds of up to 100 Gbps
    * Intel 82599 Virtual Function Interface: Supports network speeds up to 10 Gbps

D. Amazon Elastic Compute Cloud (Amazon EC2) - Part 2

3. Specify network settings

- Where should the instance be deployed? => Identify the VPC and optionally the subnet
- Should a public IP address be automatically assigned? => To make internet-accessible

4. Attach IAM Role (Optional)

- Will software on the EC2 instance need to interact with other AWS services? (If yes, attach an appropriate IAM Role)
- An AWS Identify and Access Management (IAM) role that is attached to an EC2 instance is kept in an instance type.
- Users are not restricted to attaching a role only at instace launch => User can also attach a role to an instance that already exits

5. User Data Script (Optional)

- Optionally specify a user data script at instance launch
- Use user data scripts to customize the runtime environment of the user instance
- Can be used strategically

6. Specify Storage

- Configure the root volume (Where the guest operating system is installed?)
- Attach additonal storage volumes (optional) => AMI might alreafy include more than one volume
- For each volume, specify:
  + The size of the disk (in GB)
  + The volume type (Different types of solid state drivers (SSDs) and hard disk drives (HDDs) are available.
  + If the volume will be deleted when the instance is terminated
  + If encryption should be used
- Amazon EC2 Storage Option:
  + Amazon Elastic Block Store (Amazon EBS):
    * Durable, block-level storage volumes
    * Users can stop the instance and start it again, and the data will be there
  + Amazon EC2 Instance Store:
    * Storage is provided on disks that are attached to the host computer where the EC2 instance is running.
    * If the instance stops, data stored here is deleted
  + Other options for storage (not for the root volumes)
    * Mount on Amazon Elastic File System (Amazon EFS) file system
    * Connect to Amazon Simple Storage Service (Amazon S3)

7. Add tags

A tag is a label that user can assign to an AWS resource (Console of a key and an optional value).

- Tagging is how to user can attach metadata to an EC2 instance.
- Potential benefits of tagging - Filtering automation, cost allocation, and access control

8. Add tags

- A security group is a set of firewall rules that control traffic to the instance.
- Create rules that specify the source and which ports that network communication can use:
  + Specify the port number and the protocol, such as Transmission Control Protocol (TCP), User Datagram Protocl (UDP), or Internet Control Message Protocol (ICMP)
  + Specify the source (for example, an IP address or another security group) that is allowed to use the rule.

9. Key Pair: Identify or Create the key pair

- An instance launch, user specify an existing key pair or create a new key pair
- A key pair consists of:
  + A public key that AWS stores
  + A private key file that user stores
- Enables secure the connection to the instance
- For Windows AMIs: Use the private key to obtain the administrator password that usr need to log in to the instance
- For Linux AMIs: Use the private key to use SSH to securely connect to the instance
- Amazon EC2 instance lifecycle picture
- Consider using an Elastic IP address:
  + Rebooting an instance will not change any IP addresses or DNS hostsname
  + When an instance us stopped and then started again:
    * The public IPv4 address and external DNS hostname will change
    * The internal UPv4 address and internal DNS hostname do not change
- If user require a persistent public IP address, associate an Elastic IP address with the instance
- Elastic IP address characteristics:
  + Can be associated with instances in the Region as needed
  + Remains allocated to user account until he/she choose to release it
- EC2 instance metadata: nstance metadat is data about the instance
- Amazon Cloudwatch for monitoring:
  + Use Amazon CloudWatch to monitor EC2 instances:
    * Provides near-real-time metrics
    * Provides charts in the Amazon EC2 console. Monitoring tab that you can view
    * Maintains 15 months of historical data
  + Basic monitoring
    * Default, no additional costs
    * Metric data sent to CloudWatch every 5 minutes
  + Detailed Monitoring:
    * Fixed monthly rate for seven pre-selected metrics
    * Metric data delivered every 1 minute

D. Amazon EC2 Cost Optimization

- On-Demand Instances:
  + Pay by the hour
  + No long-term commitments
  + Eligible for the AWS Free Tier
- Dedicated Hosts: A physical server with EC2 instance capacity fully dedicated to user use
- Dedicated Instances: Instances that run in a VPC on hardware that is dedicated to a single customer.
- Reserved Instances:
  + Full, patrial, or no upfront payment for instance user reserved.
  + Discount on hourly change for that instance.
  + 1-year or 3-year term
- Scheduled Reserved:
  + Purchase a capacity reservation that is always available on a recurring schedule user specify
  + 1-year term
- Spot Instances:
  + Instances run as long as they are available and user bid is above the Spot Instance price
  + They can be interrupted by AWS with a 2-minute notification.
  + Interruption options include terminated, stopped or hibernated
  + Prices can be significantly less expensive compared to On-Demand instances
  + Good choices when user have flexibility in when applications run.
- Amazon EC2 pricing models:

	On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
Benefits	Low costs and felxibility	-Large scale -Dynamic workload	Predictability ensures compute capacity is available when needed	* Save money on licensing costs. * Help meet compliance and regulatory requirements
Use Cases	* Short-term, spiky, or unpredictable workloads * Application development or testing	* Applications with flexible start and end times. * Applications only feasible at very low compute prices. * Users with urgent computing needs for large amount of additional capacity.	* Steady state or predictable usage workloads * Applications that require reserved capacity, including disaster recovery. * Users able to make upfront payments to reduce total computing costs even further.	* Bring your own license (BYOL) * Compliance and regulatory restrictions * Usage and licensing tracking * Control instance placement

- The four pillars of cost optimization:
  + Right-size
  + Increase elasticity
  + Optimal pricing model
  + Optimize storage choices

- Pillar 1: Right size
  + Provision instances to match the need
    * CPU, memory, storage, and network throughput
    * Select appropriate instances types for user use
  + Utilize Amazon CloudWatch metrics:
    * How idle are instances? When?
    * Downsize instances:
  + Best pratice: Right size, then reserve
- Pillar 2: Increase Elasticity:
  + Stop or hibernate Amazon EBS-backed instances that are not actively in use (For example, non-production development or test instances)
  + Use automatic scaling to match needs based on usage (Automatic and time-based elasticity)
- Pillar 3: Optimal pricing model
  + Leverage the right pricing model for user case (Consider usage patterns)
  + Optimize and combine purchase types
  + Examples:
    * Use On-Demand Instance and Spot Instances for variable workloads
    * use Reserved Instances for predictable workloads
  + Consider serverless solutions (AWS Lambda)
- Pillar 4: Optimize Storage choices
  + Reduce costs while maintaining storage performance and availability
  + Resize EBS Volumes
  + Change EBS volume types
  + Delete EBS snapshots that are no longer needed
  + Identity the most appropriate destination for specific types of data
- Measure, Monitor, and Improve:
  + Cost optimization is ongoing process
  + Recommendations:
    * Define and enforce cost allocation tagging
    * Define metrics, set targets, and review regularly
    * Encourage teams to architect for cost
    * Assign the responsibility of optimization to an individual or the team

E. Computer Services

- Container basics:
  + Containers are a method operating system virtualization
  + Benefits:
    * Repeatable
    * Self-contained environments
    * Software runs the same in different environments
    * Faster to launch and stop to terminate than virtual machines
- Docker:
  + Docker is a software platform that enables user to build, test, and deploy applications quickly.
  + Users can run containers on Docker (Containers are created from a template called an image)
  + A container has everything a software application needs to run (libraries, System tools, Code, and Runtime)
- Screebshor container Docker versus virtual machines
- Amazon Elastic Container Service (Amazon ECS0:
  + Amazon Elastic Container Service (Amazon ECS): A highly scalable for container management service
  + Key benefits:
    * Orchestrates the running of Docker containers
    * Maintains and scales the fleet of nodes that run user containers
    * Removes the complexity of standing up the infrastructure
  + Integrated with features that are familiar to Amazon EC2 service users:
    * Elastic Load Balacing
    * Amazon EC2 Security Groups
    * Amazon EBS Volume
    * IAM Roles
- Image of Amazon ECS orchestrates containers
- image of Amazon ECS Cluster options
- Kubernetes:
  + Kubernetes is open source software for container orchestration:
    * Deploy and manage containerized applications at scale
    * The same toolset can be used on premises and on the cloud
  + Complements Docker:
    * Docker enables user to run multiple containers on a single OS host.
    * Kubernets orchestrates multiple Docker hosts (nodes).
  + Automates:
    * Container provisioning
    * Networking
    * Load Distribution
    * Scaling
- Amazon Elastic Kubernetes Service (Amazon EKS)
  + Enables to run Kubernetes on the AWS
  + Certified Kubernetes conformant (supports easy migration)
  + Supports Linux and Windows containers
  + Compatible with Kubernetes community tools and supports popular Kubernetes add-ons
- Use Amazon EKS to:
  + Manage clusters of Amazon EC2 compute instances
  + Run containers that are orchestrated by kubernetes on those instances.
- Amazon Elastic Container Registry (Amazon ECR): Amazon ECR is a fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images:
  + Amazon ECS integration
  + Docker support
  + Team collaboration
  + Access control
  + Third-party integration

F. Introduction to AWS Lambda Service

- Cap image AWS Lambda: Run code without servers (AWS Lambda is a serverless compute service), charge when code is run
- benefits:
  + It supports multiple programming languages
  + Completely automated administration
  + Built-in fault tolerance
  + Supports the orchestration of multiple functions
  + Pay-per-use pricing
- Screenshot AWS Lambda event sources
- Screenshot AWS Lambda function configuration
- Screenshot Schedule-based Lambda function example: Start and stop EC2 instances

- Screenshot Event-based Lambda function example: Create thumbnail images
- AWS Lambda Quotas:
  + Soft limits per Region:
    * Concurrent executions = 1,000
    * Function and layer storage = 75 GB
  + Hard limits for individual functions:
    * Maximum function memory allocation = 10, 240 GB
    * Function timeout = 15 minutes
    * Deployment package size = 250 MG unzipped, including layers
    * Container image code package size = 10 GB

G. Introduction to AWS Elastic Beanstalk

- An easy way to get web applications up and running
- A managed service that automatically handles:
  + infrastructure provisioning and configuration
  + Deployment
  + Load balancing
  + Automatic Scaling
  + Health monitoring
  + Analysis and debugging
  + Logging
- No additional charge for Elastic Beanstalk: Pay only for the underlying resources that are used
- AWS Elastic Beanstalk deployments:
  + Supports web applications written for common platforms including Java, .NET, PHP, Node,js, Ruby, Python, Go, and Docker
- User can upload their code:
  + Elastic Beanstalk automatically handles the deployment
  + Deploys on various servers
- Benefits of Elastic Beanstalk:
  + Fast and simple to start using
  + Developer productivity
  + Difficult to outgrow
  + Complete resource contol