EC2 INSTANCE

What is EC2 instance

An EC2 instance refers to a virtual server in Amazon's Elastic Compute Cloud (EC2) service. It's one of the fundamental building blocks of Amazon Web Services (AWS) cloud computing platform. EC2 instances can be launched with different configurations of CPU, memory, storage, and networking capacity to meet the needs of various applications.

Understanding AMI

AMI stands for Amazon Machine Image. It's essentially a pre-configured template that contains the software configuration (operating system, application server, and applications) required to launch an EC2 instance.

1. **Operating System:** This could be Linux distributions like Ubuntu, CentOS, or Amazon Linux, or Windows Server editions.

2. **Software Configuration**: Any additional software required for your application to run. This might include web servers like Apache or Nginx, databases like MySQL or PostgreSQL, or programming language runtimes like Node.js or Python.

3**. Storage Configuration**: Information about the root volume and any additional volumes attached to the instance, including their size, type, and filesystem.

Launching your first AWS instance

Launching your first AWS instance is a great way to get started with cloud computing. Here's a basic guide to help you through the process:

1. **Sign Up for an AWS Account**: If you don't already have an AWS account, you'll need to sign up for one. Go to the AWS website (aws.amazon.com) and click on the "Create an AWS Account" button. Follow the instructions to create your account.

2. **Access the AWS Management Console**: Once you have an account, sign in to the AWS Management Console using your credentials.

3**. Navigate to the EC2 Dashboard:** In the AWS Management Console, you can find the EC2 service under the "Compute" section. Click on "EC2" to access the EC2 dashboard.

4. **Launch Instance:** In the EC2 dashboard, click on the "Launch Instance" button to start the instance creation process.

5. **Choose an AMI:** AWS provides a variety of Amazon Machine Images (AMIs) to choose from. You can select an AMI based on your requirements, such as operating system and software configuration. For beginners, the "Amazon Linux 2 AMI" is a good option.

6. **Choose an Instance Type:** Select the instance type based on your computing needs. AWS offers a wide range of instance types optimized for different workloads, such as general-purpose computing, memory-intensive applications, or GPU-accelerated tasks.

7. **Configure Instance:** Configure instance details such as the number of instances to launch, network settings, and storage options. You can leave most of the settings as default for your first instance.

8. **Add Storage:** Specify the size and type of storage for your instance. By default, AWS provides a root volume with the selected AMI. You can add additional volumes if needed.

9. **Configure Security Group:** A security group acts as a virtual firewall for your instance, controlling inbound and outbound traffic. You can create a new security group or use an existing one. Make sure to configure it to allow access to the necessary ports for your application.

10. **Review Instance Launch:** Review all the configurations you've made for your instance. Once you're satisfied, click on the "Launch" button.

11. **Create Key Pair**: If you haven't already created an SSH key pair, you'll be prompted to do so. This key pair is used to securely connect to your instance via SSH (for Linux instances) or RDP (for Windows instances). Download the private key file and store it in a safe place.

12. **Launch Instance:** After creating the key pair, click on the "Launch Instances" button to launch your instance.

13. **View Instance:** Once the instance is launched, you'll be redirected to the Instances page in the EC2 dashboard. Here, you can view details about your instance, such as its public IP address, status, and more.

On demand instance pricing

On-demand instance pricing in AWS refers to a pay-as-you-go model where you pay for compute capacity by the hour or by the second, with no long-term commitments or upfront payments.

Here are some key points to understand about on-demand instance pricing:

1. **Hourly Billing:** With on-demand instances, you are billed for the compute capacity used on an hourly basis. The billing starts when you launch the instance and stops when you terminate it.

2. **No Upfront Costs:** There are no upfront costs or long-term commitments required. You only pay for the compute capacity you use, for as long as you use it.

3. **Per-Second Billing:** AWS introduced per-second billing for on-demand instances, with a minimum of one minute. This means that you are charged based on the exact number of seconds your instance is running, rounded up to the nearest minute.

4. **Variety of Instance Types:** AWS offers a wide range of instance types optimized for different workloads, such as general-purpose, compute-optimized, memory-optimized, and storage-optimized instances. Each instance type has its own pricing based on its specifications and capabilities.

5. **Pricing Transparency:** AWS provides pricing information for on-demand instances on their website, including the hourly rates for each instance type in different regions. You can use the AWS Pricing Calculator to estimate the cost of running your desired instances based on your usage requirements.

Reserved instance pricing

Reserved Instances (RIs) in AWS provide significant discounts compared to on-demand pricing in exchange for a commitment to a specific instance configuration, term length, and payment option.

1. **Term Lengths**: RIs are available in one-year or three-year terms. Choosing a longer term typically results in greater discounts.

2. **Payment Options:** You can choose to pay for RIs upfront, partially upfront, or with no upfront payment. Upfront payment options provide the highest discounts, while no upfront options offer lower discounts but require no upfront payment.

3. **Instance Types:** When purchasing an RI, you specify the instance type (e.g., t3.micro, m5.large) and the region in which it will be used. The discounts apply only to the specified instance type and region.

4. **Flexibility:** RIs offer flexibility in terms of instance size within the same instance family, operating system, and tenancy.

5. **Savings:** RIs can result in significant cost savings compared to on-demand pricing, often up to 75% or more depending on the instance type, term length, and payment option chosen.

Spot instance pricing

Spot Instances in AWS allow you to bid on spare EC2 compute capacity at a discounted rate compared to on-demand pricing. Here's how Spot Instance pricing works:

1. **Flexible Pricing:** Spot Instance pricing fluctuates based on supply and demand dynamics within the AWS cloud. AWS sets the Spot price for each instance type and Availability Zone, which can vary over time. The Spot price is typically much lower than the on-demand price.

2. **Spot Price:** The Spot price is the maximum price you are willing to pay per hour for a Spot Instance. You specify your bid price when requesting Spot Instances.

3. **Instance Interruption:** Spot Instances are subject to interruption. If the Spot price exceeds your bid price, AWS may terminate your instances with a two-minute notice. charged for partial hours of usage, and you can specify a maximum price (known as the "bid price") to limit your spending.

4. **Instance Types:** Spot Instances support the same instance types as on-demand instances, providing flexibility to choose the instance type that best suits your needs.

Setting up security

Setting up security in AWS involves implementing measures to protect your cloud resources, data, and applications from unauthorized access, data breaches, and other security threats. Here's a step-by-step guide to setting up security in AWS:

1. Identity and Access Management (IAM):

•Create IAM users: Create individual IAM users for each person or application that needs access to your AWS resources.

•Assign permissions: Use IAM policies to grant permissions to IAM users, groups, or roles based on the principle of least privilege.

•Enable Multi-Factor Authentication (MFA): Require IAM users to authenticate with a second factor, such as a mobile app or hardware token, in addition to their password.

2.Virtual Private Cloud (VPC):

•Create a VPC: Set up a virtual network to isolate your AWS resources from the public internet.

•Subnets and routing: Divide your VPC into subnets and configure routing tables to control traffic flow between them.

•Network Access Control Lists (NACLs): Use NACLs to control inbound and outbound traffic at the subnet level.

3.Data Encryption:

• Encryption at rest: Use AWS Key Management Service (KMS) to encrypt data stored in Amazon S3, EBS volumes, RDS databases, and other AWS services.

• Encryption in transit: Use SSL/TLS to encrypt data transmitted between your users and your AWS resources, such as HTTPS for web traffic and encrypted connections for database connections.

4.Monitoring and Logging:

• AWS CloudTrail: Enable CloudTrail to log API activity and changes to your AWS resources, providing visibility into user activity and resource changes.

• Amazon CloudWatch: Set up CloudWatch alarms to monitor resource utilization, detect anomalies, and trigger automated responses.

Security groups

Security Groups in AWS act as virtual firewalls for your EC2 instances, controlling inbound and outbound traffic. Here's how to set up and manage security groups:

1.Creating Security Groups:

• Navigate to the EC2 dashboard in the AWS Management Console.

• Click on "Security Groups" in the navigation pane.

• Click on the "Create security group" button.

• Provide a name and description for your security group.

• Define inbound and outbound rules to control traffic flow. Each rule specifies a protocol (e.g., TCP, UDP, ICMP), port range, and allowed source or destination (CIDR range, security group, or self-reference).

2.Inbound Rules:

• Inbound rules control incoming traffic to your instances. For example, you might allow SSH (port 22) from your IP address, HTTP (port 80) from anywhere, or RDP (port 3389) from a specific IP range.

3.Outbound Rules:

• Outbound rules control outgoing traffic from your instances. By default, all outbound traffic is allowed. You can restrict outbound traffic as needed based on your application requirements.

4.**Editing Security Groups:**

• To modify an existing security group, select it from the list and click the "Actions" dropdown menu, then choose "Edit inbound rules" or "Edit outbound rules".

5.**Security Group Best Practices:**

• Follow the principle of least privilege: Only open the ports and protocols necessary for your application to function.

Choosing and creating a new AMI

Choosing and creating a new Amazon Machine Image (AMI) involves selecting the appropriate base operating system and software configurations, customizing it to meet your requirements, and then saving it as a new AMI. Here's how to do it:

1.**Selecting a Base AMI:**

•Go to the EC2 dashboard in the AWS Management Console.

•Click on "Launch Instance".

•In the "Choose an Amazon Machine Image (AMI)" step, select the base AMI that closely matches your requirements. AWS provides a variety of pre-configured AMIs, including different Linux distributions, Windows Server editions, and other software configurations.

2. **Customizing the Instance:**

• Configure the instance type, network settings, storage, and other parameters based on your requirements. You can choose the appropriate instance type and size for your workload.

• Optionally, install and configure additional software, applications, or scripts on the instance to customize its functionality.

3.**Preparing the Instance:**

• Connect to the instance using SSH (for Linux) or RDP (for Windows).

• Customize the instance by installing software packages, configuring settings, and performing any necessary setup steps.

• Make sure the instance is in a clean and stable state before proceeding to create the AMI.

4.Creating the AMI:

• Once the instance is customized to your satisfaction, stop the instance to ensure data consistency.

• Select the instance in the EC2 dashboard, click on the "Actions" dropdown menu, and choose "Image and templates" > "Create image (AMI)".

• Provide a name and description for your new AMI.

• Click on "Create image" to initiate the AMI creation process.

• The process may take several minutes to complete, depending on the size of the instance and the amount of data stored on the root volume.

5.Using the New AMI:

• Once the AMI creation process is finished, you can use the new AMI to launch new EC2 instances with the same custom configuration.

6.Testing and Validation:

• Before using the new AMI in production, it's essential to test and validate it thoroughly to ensure that it meets your requirements and functions as expected.

• Launch instances from the new AMI, run automated tests, and perform manual validation to verify its stability and correctness.

Public and private IP's

Public and private IP addresses are fundamental concepts in networking, including within cloud environments like AWS. Here's an overview of each:

1.Public IP Address:

• A public IP address is an address that can be accessed over the internet. It's typically assigned to resources that need to communicate directly with devices outside of their local network.

• Public IP addresses are used for accessing web servers, email servers, VPN gateways, and other services that need to be reachable from the internet.

2.**Private IP Address:**

• A private IP address is an address that is only reachable within a local network. It's typically assigned to resources that communicate within the same network or within a Virtual Private Cloud (VPC).

• Private IP addresses are used for internal communication between servers, database instances, and other resources within the same network.

Deploying a new instance from the created AMI

Deploying a new instance from a created AMI (Amazon Machine Image) in AWS is straightforward. Here's a step-by-step guide:

1.**Navigate to EC2 Dashboard:** Sign in to the AWS Management Console and navigate to the EC2 dashboard.

2.**Launch Instance:** Click on the "Launch Instance" button to start the instance creation process.

3.**Choose AMI:** In the "Choose an Amazon Machine Image (AMI)" step, select the "My AMIs" tab.

4.**Choose Instance Type:** Select the instance type that best suits your requirements. This determines the compute, memory, and networking capacity of your instance.

5. **Configure Instance:** Configure instance details such as the number of instances to launch, network settings, and storage options. You can leave most of the settings as default, or customize them as needed.

6**. Add Storage:** Specify the size and type of storage for your instance. By default, AWS uses the root volume settings from the AMI you selected, but you can customize the storage options if necessary.

7**. Configure Security Group:** Choose an existing security group or create a new one to control inbound and outbound traffic to your instance. Ensure that the security group allows access to the necessary ports for your application.

8. **Review Instance Launch:** Review all the configurations you've made for your instance to ensure they're correct. Click on the "Launch" button to initiate the instance launch process.

9**. Select Key Pair:** Select an existing key pair or create a new one. This key pair is used to securely connect to your instance via SSH (for Linux instances) or RDP (for Windows instances).

10.**Launch Instance:** After selecting the key pair, click on the "Launch Instances" button to launch your instance.

key pairs

Key pairs in AWS are used for secure access to EC2 instances. When you create an EC2 instance, you have the option to associate a key pair with it. Here's what you need to know about key pairs:

1.**SSH Key Pairs (Linux Instances):**

•If you're launching a Linux instance, you'll use an SSH key pair for authentication.

•During the instance launch process, you'll specify the key pair to be associated with the instance. You can either select an existing key pair or create a new one.

•When you connect to the instance using SSH, you'll provide the private key corresponding to the public key of the key pair. AWS stores the public key, and you keep the private key secure on your local machine.

2.RDP Key Pairs (Windows Instances):

•If you're launching a Windows instance, you'll use an RDP (Remote Desktop Protocol) key pair for authentication.

•Similar to SSH key pairs, during the instance launch process, you'll specify the key pair to be associated with the instance. You can either select an existing key pair or create a new one.

•When you connect to the Windows instance using Remote Desktop, you'll use the password associated with the key pair.

3.Creating Key Pairs:

•You can create a new key pair in the EC2 dashboard of the AWS Management Console.

•Navigate to the "Key Pairs" section, click on "Create key pair", provide a name for the key pair, and then download the private key file (*.pem for Linux instances, *.pfx for Windows instances).

•It's crucial to keep the private key secure, as it provides access to your instances.

4.Importing Key Pairs:

•If you already have an existing SSH key pair, you can import it into AWS. Similarly, if you have an existing X.509 certificate, you can import it as an RDP key pair.

•Imported key pairs are useful when you need to use the same key pair across multiple instances or if you want to manage your keys centrally within AWS

Elastic IP's

Elastic IP addresses (EIPs) in AWS are static IPv4 addresses that you can allocate and associate with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers. Here's a closer look at Elastic IP addresses:

1.Static IP Address:

•An Elastic IP address is a static IPv4 address that remains associated with your AWS account until you choose to release it.

•Unlike the dynamic public IP addresses assigned to EC2 instances by default, Elastic IP addresses do not change unless explicitly released by the user.

2.**Use Cases:**

•Elastic IP addresses are commonly used in scenarios where you need a persistent public IP address for your AWS resources.

•They are particularly useful for internet-facing applications, such as web servers, where you want to ensure a consistent IP address for DNS records and external access.

3.**Allocation and Association:**

•To use an Elastic IP address, you must first allocate it from the pool of available addresses in your AWS account.

•Once allocated, you can associate the Elastic IP address with an EC2 instance, NAT gateway, or other supported AWS resources.

•You can associate an Elastic IP address with an instance either during instance launch or after the instance is running.

4.**Cost and Billing:**

•Elastic IP addresses are free to use as long as they are associated with an EC2 instance that is running.

•However, if an Elastic IP address is not associated with any running instance or if it's associated with a stopped or terminated instance

•It's important to release unused Elastic IP addresses to avoid incurring unnecessary charges.

5.**Releasing Elastic IP Addresses:**

•When you no longer need an Elastic IP address, it's essential to release it to avoid ongoing charges.