Introduction to Amazon Web Services (AWS)

AWS (Amazon Web Services) is a comprehensive and widely adopted cloud platform, offering over 200 fully featured services from data centers globally. It provides various services including computing power, storage options, and networking capabilities.

How to Create an AWS Account

1. **Visit AWS Website**: Go to the AWS homepage at aws.amazon.com.
2. **Sign Up**: Click on "Create an AWS Account".
3. **Provide Email and Password**: Enter your email address and create a password.
4. **Enter Contact Information**: Provide your contact details, including phone number and address.
5. **Payment Information**: Enter your credit/debit card details. AWS offers a free tier for new users, but a card is required for identity verification.
6. **Verify Identity**: Complete the phone verification process.
7. **Select Support Plan**: Choose a support plan (you can start with the basic free plan).
8. **Complete Signup**: Once the verification is done, your AWS account is ready.

How to Create a Server (Instance) in AWS

Using Amazon EC2 (Elastic Compute Cloud)

1. **Log in to AWS Management Console**: Go to the AWS Management Console and sign in.
2. **Navigate to EC2**: In the AWS Management Console, type "EC2" in the search bar and select EC2 from the results.
3. **Launch Instance**:
    - Click on "Launch Instance".
    - Choose an Amazon Machine Image (AMI). The AMI is a template that contains the software configuration required to launch your instance (e.g., an operating system, an application server, and applications).
4. **Choose Instance Type**: Select the type of instance you want (e.g., t2.micro, which is free tier eligible).

5. **Configure Instance Details**: Customize the instance settings as needed. You can leave the default settings for basic configurations.
6. **Add Storage**: Specify the storage size and type for your instance.
7. **Add Tags**: (Optional) Add tags to your instance for better management and organization.
8. **Configure Security Group**: Set up the firewall rules that control the traffic to your instance. For a web server, you would typically allow HTTP (port 80) and HTTPS (port 443) traffic.
9. **Review and Launch**: Review your settings and click "Launch".
10. **Select or Create a Key Pair**: Choose an existing key pair or create a new one to securely connect to your instance. Download the key pair and keep it safe.
11. **Launch Instance**: Click on "Launch Instances". Your instance will start within a few minutes.

Types of Cloud Services: SAAS, PAAS, IAAS

1. **SAAS (Software as a Service)**:
   - **Definition**: Software that is delivered over the internet and managed by a third-party provider.
   - **Examples**: Google Workspace, Salesforce, Microsoft Office 365.
   - **Use Case**: Ideal for applications that need to be accessed from anywhere, anytime.
2. **PAAS (Platform as a Service)**:
   - **Definition**: Platform services that provide a framework for developers to build and deploy applications.
   - **Examples**: AWS Elastic Beanstalk, Google App Engine, Microsoft Azure App Services.
   - **Use Case**: Ideal for developers who want to build applications without worrying about underlying infrastructure.
3. **IAAS (Infrastructure as a Service)**:
   - **Definition**: Basic computing resources such as virtual machines, storage, and networks provided over the internet.

  - **Examples**: Amazon EC2, Google Compute Engine, Microsoft Azure VMs.
  - **Use Case**: Ideal for IT administrators who need control over the infrastructure without managing physical hardware.

How to Push Code from GitHub to the Cloud

1. **Set Up AWS CLI**: Install and configure the AWS Command Line Interface (CLI) on your local machine.
2. **Authenticate AWS CLI**: Use aws configure to set up your credentials.
3. **Clone Your GitHub Repository**: Clone the repository to your local machine using git clone <repository-url>.
4. **Push Code to AWS Service**:
   - For deploying web applications, you can use AWS services like Elastic Beanstalk, Lambda, or CodeDeploy.
   - Example using Elastic Beanstalk:
     - Initialize Elastic Beanstalk: eb init.
     - Create environment: eb create <environment-name>.
     - Deploy: eb deploy.

How to Purchase a Domain in AWS

1. **Navigate to Route 53**: In the AWS Management Console, type "Route 53" in the search bar and select it.
2. **Register a Domain**: Click on "Domains" in the left sidebar, then click on "Register Domain".
3. **Search for Domain**: Enter the domain name you want to purchase and check its availability.
4. **Add to Cart and Checkout**: If the domain is available, add it to your cart and proceed with the checkout process.
5. **Configure Domain**: Once purchased, configure DNS settings in Route 53 to point to your resources (e.g., EC2 instances, load balancers).

How to Secure the Domain and Add Restrictions

1. **Enable HTTPS**:
   - Use AWS Certificate Manager (ACM) to request an SSL/TLS certificate.
   - Attach the certificate to your CloudFront distribution or load balancer to enable HTTPS.
2. **Configure Security Groups**:
   - Set up security groups to restrict access to your instances by IP address or range.
3. **Use WAF (Web Application Firewall)**:
   - AWS WAF helps protect your web applications from common web exploits.
   - Create and attach a WAF rule to your CloudFront distribution or Application Load Balancer.
4. **Enable Shield**:
   - AWS Shield provides DDoS protection for your applications.
   - Enable AWS Shield Standard for basic protection, or AWS Shield Advanced for advanced protection.

Brief Overview of Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 provides scalable computing capacity in the AWS cloud. With EC2, you can launch virtual servers, called instances, to run applications. Key features include:

1. **Scalability**: Easily scale up or down to handle changes in requirements or traffic.
2. **Choice of Instances**: A wide variety of instance types optimized for different use cases (e.g., compute-optimized, memory-optimized).
3. **Flexible Pricing**: Pay-as-you-go pricing, reserved instances for long-term savings, and spot instances for cost efficiency.
4. **Security**: Secure your instances with security groups, network ACLs, and IAM roles.
5. **Integration with Other AWS Services**: Seamlessly integrate with services like S3, RDS, and VPC.