VPC:

Amazon Virtual Private Cloud is a commercial cloud computing service that provides users a virtual private cloud, by "provisioning a logically isolated section of Amazon Web Services Cloud". VPC gives you full control over your virtual networking environment, including resource placement, connectivity, and security. VPC is a collection of the region, IG, route table, ACL, security group, subnet, instances.

Subnet:

Subnet is a key component in VPC. A VPC can contain all public subnets (or) public/private subnet combination. Subnet is to split a large network into a grouping of smaller, interconnected networks to help minimize traffic. This way, traffic doesn't have to flow through unnecessary routs, increasing network speeds.

Private Subnet: Private Subnet is a subnet which doesn't have a route to the internet gateway. A subnet can be configured as a VPN-only subnet by routing traffic via virtual private gateway. There is no capability for resources in a private subnet to communicate directly with the Internet, and vice versa.

Public Subnet: Public Subnet is one that has a Route Table entry that directs traffic destined for 0.0.0.0/0 to an Internet Gateway. This gives the resources on that VPC the ability to communicate with the Internet. The resources will need to be assigned a Public IP address to be able to communicate with the Internet.

Security Group:

A security group is like a virtual firewall. It consists of a set of rules that can be used to monitor and filter an instance's incoming and outgoing traffic in a Virtual Private Cloud (VPC) instance. Filtering is done on the basis of protocols and ports.

For example: after you associate a security group with an EC2 instance, it controls the inbound and outbound traffic for the instance.

Route Table:

A route table contains a set of rules called routes, that are used to determine where network traffic from subnet or gateway is directed. A route table tells network packets which way they need to go to get to their destination. Each packet contains information about its origin and destination.

NAT: Network Address Translation Gateway is used to enable instances present in a private subnet to help connect to the internet or AWS services. In addition to this, the gateway makes sure that the internet doesn't initiate a connection with the instances.

Difference between IAM Role and Policy:

A policy is an object in AWS that defines permissions. Policies are JSON documents in AWS that let you specify who has access to AWS resources, and what actions they can perform on those resources. You can attach a policy to an identity or resource to define their permissions. AWS evaluates these policies when the IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied.

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations.

IAM Roles manage who has access to your AWS resources, whereas IAM policies control their permissions.

Ansible:

Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, intraservice orchestration, and provisioning.

Ansible works by **connecting to your server with SSH and thereby pushing out small programs, known as 'Ansible modules' to it**. Ansible's most powerful feature of creating playbooks, a small piece of YAML code, is used to automate almost anything related to IT requirements.

Ansible is agentless, which means the nodes it manages do not require any software to be installed on them.

Hosts:

In ansible, host files are **those files that are used for storing information about remote nodes information**, which we need to manage. This file can be placed anywhere but its location needs to be set up either in a configuration file or give on the command line.

Tasks:

A Task is th**e smallest unit of action you can automate using an Ansible playbook**. Playbooks typically contain a series of tasks that serve a goal, such as to set up a web server, or to deploy an application to remote environments. Ansible executes tasks in the same order they are defined inside a playbook

Vars:

Ansible uses variables **to manage differences between systems**. With Ansible, you can execute tasks and playbooks on multiple different systems with a single command. To represent the variations among those different systems, you can create variables with standard YAML syntax, including lists and dictionaries.

To define a variable in a playbook, simply **use the keyword vars before writing your variables with indentation**. To access the value of the variable, place it between the double curly braces enclosed with quotation marks.

Module:

A Module is a reusable, standalone script that Ansible runs on your behalf, either locally or remotely. Modules interact with your local machine, an API, or a remote system to perform specific tasks like changing a database password or spinning up a cloud instance.

Ansible modules are standalone scripts that can be used inside an Ansible playbook.

How does Ansible works?

Ansible works by connecting to your nodes and pushing out small programs, called modules to them. Modules are used to accomplish automation tasks in Ansible. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules and removes them when finished.