**How to connect a private server without using bastion host?**

Connecting an ec2 server via VPC endpoints and session manger are possible.

First method is by using **VPC endpoints** and the second method is by using **session manager**.

Let’s see in detail …

1. **Connecting an EC2 server via VPC endpoints**

Steps to create a private server using endpoint are as follows …

Go to VPC

VPC > endpoints > create endpoints

Give

**Name:** myec2endpt

**Type:** Select - > EC2 Instance Connect Endpoint

**Network Settings**

**VPC:** Select a default VPC

**Security Group:** Select default sg group and make sure SSH port is enabled

**Subnet**: Select your subnet

**Create Endpoint**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Go to vpc

A screenshot of a computer

Description automatically generated

Create a server and during connect option window

Select the option > **connect using ec2 instance connect endpoint**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

 --------

1. Connect an EC2 private server via Session manager

What is VPC session manager, and how to connect ec2 instance with using session manager?

My session manger connect was disable, Here’s the step to enable it.

A screenshot of a computer

Description automatically generated

For Session manager we should attach IAM role "AmazonSSMManagedInstanceCore"

Create a IAM role

A screenshot of a computer

Description automatically generated

Now attach this to your ec2

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

 --------

Some notes on VPC endpoints and Session Manger are as follows for better understanding …

What is VPC Endpoints?

A **VPC endpoint** is a private connection within Amazon Web Services (AWS) that allows resources in your **Amazon Virtual Private Cloud (VPC)** to securely communicate with AWS services or third-party services without requiring an internet gateway, NAT device, VPN, or Direct Connect. It ensures that traffic between your VPC and the service remains entirely within the AWS network.

**Key Features of VPC Endpoints**

1. **Private Connectivity**:
   * Traffic does not traverse the public internet, improving security and reducing latency.
2. **Cost Efficiency**:
   * Avoid data transfer costs associated with NAT gateways or internet gateways for accessing AWS services.
3. **Enhanced Security**:
   * Use AWS Identity and Access Management (IAM) and VPC endpoint policies to control access.
4. **Highly Available**:
   * VPC endpoints are fully redundant and scalable, ensuring high availability.

**Types of VPC Endpoints**

AWS offers two main types of VPC endpoints:

**1. Interface Endpoint**

* Uses **Elastic Network Interfaces (ENI)** in your VPC to connect to supported AWS services or third-party services.
* Traffic between your VPC and the service stays within AWS's private network.
* Examples:
  + Connect to **Amazon S3**, **DynamoDB**, **Secrets Manager**, or other AWS services.
  + Supported for most AWS services.

**Use Case**: When connecting to services that require API calls over HTTPS.

**2. Gateway Endpoint**

* Creates a **gateway** in your VPC route table to redirect traffic to specific AWS services.
* Available only for **Amazon S3** and **DynamoDB**.
* No need for ENIs; works directly through route tables.

**Use Case**: When accessing S3 or DynamoDB frequently and want high-performance, low-cost connectivity.

**How VPC Endpoints Work**

1. **Service Configuration**:
   * Identify the service you want to connect to (e.g., S3, DynamoDB, or other AWS services).
   * Choose between an **interface endpoint** or **gateway endpoint**.
2. **Endpoint Creation**:
   * Create the endpoint using the AWS Management Console, CLI, or SDK.
   * For **interface endpoints**, AWS creates an ENI in the specified subnet.
   * For **gateway endpoints**, update the route tables in your VPC to route traffic through the endpoint.
3. **Access Control**:
   * Use VPC endpoint policies to define which services, actions, and resources are accessible.
   * Combine with IAM policies for granular control.
4. **Testing and Validation**:
   * Verify connectivity to the service without requiring an internet connection.

**Benefits of VPC Endpoints**

1. **Improved Security**: Keeps traffic within AWS's private network, reducing exposure to the internet.
2. **Better Performance**: Eliminates latency introduced by public internet routing.
3. **Cost Savings**: Reduces reliance on NAT gateways, internet gateways, and associated data transfer costs.
4. **Simplified Architecture**: Removes the need for complex setups involving VPNs or bastion hosts for accessing AWS services.

**Use Cases**

1. **Private Access to AWS Services**:
   * Access S3 buckets without opening the VPC to the internet using a **gateway endpoint**.
2. **Third-Party Integrations**:
   * Use **interface endpoints** to connect with third-party SaaS providers listed in AWS Marketplace.
3. **Hybrid Cloud Scenarios**:
   * Securely integrate on-premises environments with AWS services using VPC endpoints.
4. **Regulatory Compliance**:
   * Ensure sensitive data remains within the AWS private network.

What is session manager?

The **VPC Session Manager** typically refers to using **AWS Systems Manager Session Manager** within a Virtual Private Cloud (VPC). It is a tool that provides a secure, auditable way to manage and interact with Amazon EC2 instances or other AWS resources within your VPC, without needing to open inbound ports like SSH or RDP.

**Key Features of VPC Session Manager**

1. **Portless Access**:
   * You don’t need to open SSH or RDP ports in your VPC's security groups. Session Manager connects via the **AWS Systems Manager Agent (SSM Agent)** installed on your EC2 instances.
2. **Secure Communication**:
   * Communication happens over HTTPS, and authentication is managed through AWS Identity and Access Management (IAM), making it highly secure.
3. **No Need for Bastion Hosts**:
   * Traditionally, you’d set up a bastion host to access instances in a private subnet. With Session Manager, this is no longer required, simplifying the architecture and reducing costs.
4. **Logging and Auditing**:
   * Sessions can be logged to **Amazon S3** or **Amazon CloudWatch Logs** for audit and compliance purposes.
5. **Cross-Platform Support**:
   * Supports instances running Linux, Windows, and MacOS.

**Benefits of Using VPC Session Manager**

* **Enhanced Security**: Eliminates the need for SSH keys, passwords, and open ports.
* **Cost-Effective**: Reduces the need for additional resources like bastion hosts or NAT gateways.
* **Centralized Access Management**: IAM policies control who can start sessions and what they can do within the session.
* **Ease of Use**: Access instances directly from the AWS Management Console, AWS CLI, or SDKs.

**How VPC Session Manager Works**

1. **Instance Setup**:
   * Ensure the **SSM Agent** is installed and running on your EC2 instances (many Amazon-provided AMIs include it by default).
   * Attach an **IAM role** to your instance with appropriate permissions for Systems Manager.
2. **Session Creation**:
   * Start a session from the AWS Management Console, AWS CLI (aws ssm start-session), or AWS SDK.
   * The session connects securely to the instance without requiring open ports.
3. **Optional Logging**:
   * Configure **CloudWatch Logs** or **S3 buckets** to store session logs for later review.

**Use Cases**

1. **Managing Instances in Private Subnets**:
   * Use Session Manager to manage EC2 instances in isolated VPCs without exposing them to the internet.
2. **Compliance and Auditing**:
   * Track and log all session activities to meet compliance requirements.
3. **Simplified Remote Access**:
   * Access EC2 instances from anywhere without needing a VPN or bastion host.
4. **Automation and Troubleshooting**:
   * Integrate with AWS Systems Manager for advanced automation and to execute scripts on multiple instances.