What is a Bastion Host?

Bastion Host = Security Check Gate at the Airport
Everyone goes through it before entering private areas. It checks, controls, and records who goes inside.

A **Bastion Host** is a **secure entry point** to reach your **private EC2 servers** that don't have public IPs.

Use Case: Why Do We Need It?
Let's say:

- You have **private EC2 servers** (for backend, databases) that should NOT be exposed to the internet.
- But **developers/admins** need to connect to them for maintenance.

❗ Problem: You don't want to give every private server a public IP. That's risky!

Solution: Use a Bastion Host

- Place 1 server (Bastion Host) in a **public subnet**.
- Only **this host is accessible from your laptop**.
- Once inside the Bastion, you can SSH into the private EC2s **safely using internal IPs**.

How It Improves

Feature	Benefit
Only 1 public server	Reduces attack surface – other servers stay hidden from internet
Controlled Access	You can lock access to **only your IP**
Easier to Monitor	Since all SSH/RDP goes through one point, it's easy to log and track
No Public IPs Needed	Keeps private EC2s truly private
Central Firewall Rules	Configure rules once at Bastion – not on every server

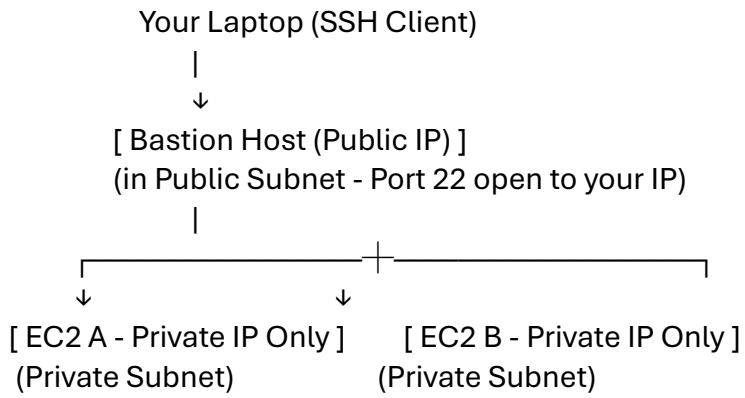
---

AWS Example

1. **Bastion Host**
    - Public subnet
    - Public IP
    - SSH access allowed from your laptop only
2. **Private EC2 Servers**
    - Private subnet
    - No public IP
3. **Access Flow:**
   Your Laptop → SSH → Bastion Host → SSH → Private EC2

💡 **Best Practices**

- Use **SSH key authentication** (no passwords).
- Add **logging or session recording** for auditing.
- Restrict SSH to **your IP only** using Security Group.
- Use **AWS Systems Manager (SSM Session Manager)** instead of Bastion for **passwordless and secure access** (no need for SSH at all).



Without Bastion Host

you can access a private EC2 instance without a Bastion Host using **AWS Systems Manager (SSM) Session Manager** — and it's actually **more secure and simpler** than SSH!

How SSM Session Manager Works
No Public IP? No Problem
Your EC2 instance can be in a **private subnet** with **no public IP** — and you can still connect!

1. **No SSH, No Key Files**
   You **don't need SSH access**, private key files, or open ports like port 22.

2. **Secure, Browser-based or CLI Access**
   You connect to the instance through the **AWS Console** or **AWS CLI**, using IAM permissions and logging everything.

✅ **Requirements to Use SSM Session Manager**

Requirement	Details
✅ IAM Role	The EC2 instance must have an **IAM role** with SSM permissions like AmazonSSMManagedInstanceCore
✅ SSM Agent	EC2 must have the **SSM Agent installed and running** (pre-installed in Amazon Linux 2, Ubuntu 20+)
✅ VPC Endpoints *(for private subnets)*	If the instance is in a private subnet, create **VPC Endpoints** for SSM and EC2 Messages (no internet needed)
✅ No SSH Needed	You do not open port 22, making it more secure

---

🛡️ **Advantages Over Bastion Host**

Feature	Bastion Host	SSM Session Manager
Public IP Needed	Yes (for Bastion)	❌ No
SSH Key Required	Yes	❌ No
Port 22 Open	Yes	❌ No ports needed
Audit & Logging	Manual setup	✅ Built-in with CloudTrail
Cost	Extra EC2 cost	✅ No extra instance needed

Real-Life Use Case
You want to access a **private EC2 instance in a secure VPC** with **no open ports and no public IP**.
Instead of setting up and securing a Bastion Host, you just:

- Attach an IAM role to your EC2
- Make sure SSM agent is running
- Start session from AWS Console or CLI

1: Which of the following is TRUE about using a Bastion Host to connect to EC2 instances in a private subnet?
A. Bastion Host resides in a private subnet with no public IP.
B. Bastion Host provides a direct internet connection to all private instances.
C. Bastion Host requires SSH key and security group rules to allow inbound traffic.
D. Bastion Host does not need any monitoring or access logs.

✅ **Correct Answer:** C
Explanation: A Bastion Host needs to be in a **public subnet** with a public IP and requires **SSH key-based login** and properly configured **security groups** to allow SSH access.

2: What is a key advantage of using AWS Systems Manager Session Manager over a Bastion Host for EC2 access?

A. Session Manager requires you to open port 22 in the security group.
B. Session Manager does not support EC2 instances without internet access.
C. Session Manager allows passwordless access without exposing instances to the internet.
D. Session Manager requires a separate EC2 instance to work.

✅ **Correct Answer:** C
Explanation: AWS Session Manager provides **secure, passwordless access** to EC2 instances **without needing a Bastion Host or SSH** and **does not expose** the instances to the public internet.