**AWS NAT Gateway & Elastic IP - Complete Guide**

**Table of Contents**

1. Introduction to Elastic IP
2. Why Use Elastic IP?
3. NAT Gateway Explained
4. Lab Architecture
5. Step-by-Step Implementation
   * Step 1: Create VPC with Public/Private Subnets
   * Step 2: Configure Elastic IP
   * Step 3: Set Up NAT Gateway
   * Step 4: Launch EC2 Instances
6. Testing Connectivity
7. Cleanup
8. Real-World Use Cases
9. Key Takeaways

**1. Introduction to Elastic IP**

* **What is Elastic IP?**  
  A static IPv4 address that persists even if you stop/restart your instance.
* **Key Feature**:
  + Remains constant (unlike auto-assigned public IPs that change on instance restart).
  + Can be reassigned to other instances.

**Example Scenario**

* Without Elastic IP:  
  Instance Public IP = 54.123.141.50 → Restart → Changes to 52.223.231.74
* With Elastic IP:  
  Elastic IP = 143.189.52.224 → Always stays the same.

**2. Why Use Elastic IP?**

| **Use Case** | **Benefit** |
| --- | --- |
| **Web Servers** | DNS records (Route 53) won't break on instance restart |
| **Hybrid Cloud (AWS + Azure/On-prem)** | Fixed IP for VPN/Transit Gateway connections |
| **Cost Optimization** | Free if attached to running instance |

⚠️ **Warning**: Always release unused Elastic IPs to avoid charges ($0.005/hr per IP).

**3. NAT Gateway Explained**

**What is NAT Gateway?**

* Allows **private subnets** to access the internet (e.g., for updates) while **blocking inbound traffic**.
* **Key Features**:
  + Uses Elastic IP.
  + Highly available (unlike NAT instances).
  + Scales automatically.

**NAT vs Internet Gateway**

| **Feature** | **NAT Gateway** | **Internet Gateway** |
| --- | --- | --- |
| **Direction** | Outbound-only | Bidirectional |
| **Use Case** | Private subnets | Public subnets |
| **Attached To** | Public subnet | VPC |

**4. Lab Architecture**

1. **VPC**: 10.0.0.0/16
2. **Public Subnets**:
   * 10.0.1.0/24 (Web Server)
   * 10.0.2.0/24 (NAT Gateway)
3. **Private Subnets**:
   * 10.0.3.0/24 (App Server)
   * 10.0.4.0/24 (Database)

**5. Step-by-Step Implementation**

**Step 1: Create VPC with Public/Private Subnets**

1. Go to **VPC Dashboard → Create VPC**.
2. Configure:
   * **IPv4 CIDR**: 10.0.0.0/16
   * **Public Subnets**: 2 (across 2 AZs)
   * **Private Subnets**: 2
   * **NAT Gateway**: 1 (in a public subnet)
   * **Internet Gateway**: Auto-created.

**Step 2: Configure Elastic IP**

1. Go to **EC2 → Elastic IPs → Allocate Elastic IP**.
2. **Allocate** and note the IP (e.g., 143.189.52.224).

**Step 3: Set Up NAT Gateway**

1. Go to **VPC → NAT Gateways → Create NAT Gateway**.
   * **Subnet**: Select a **public subnet**.
   * **Elastic IP**: Attach the allocated IP.
2. **Edit Route Tables** for private subnets:
   * Add route 0.0.0.0/0 → Target: nat-gateway-id.

**Step 4: Launch EC2 Instances**

1. **Public Instance** (Web Server):
   * Subnet: 10.0.1.0/24 (public)
   * Auto-assign Public IP: Enabled
2. **Private Instance** (App Server):
   * Subnet: 10.0.3.0/24 (private)
   * Auto-assign Public IP: Disabled

**6. Testing Connectivity**

**Public Instance**

bash

ping google.com *# Should work (uses Internet Gateway)*

**Private Instance**

bash

ping google.com *# Should work (uses NAT Gateway)*

wget https://example.com/file.zip *# Test outbound traffic*

**7. Cleanup**

1. **Terminate EC2 Instances**.
2. **Delete NAT Gateway**.
3. **Release Elastic IP**.
4. **Delete VPC**.

**8. Real-World Use Cases**

| **Scenario** | **Solution** |
| --- | --- |
| **Private DB needs updates** | NAT Gateway allows outbound traffic |
| **Static IP for VPN** | Elastic IP + NAT Gateway |
| **Cost-effective HA** | NAT Gateway scales automatically |

**9. Key Takeaways**

1. **Elastic IP** = Static IP for instances.
2. **NAT Gateway** = Outbound internet for private subnets.
3. **Always attach NAT Gateway to public subnet**.
4. **Release unused Elastic IPs** to avoid costs.