Cloud-Based Offensive Security Infrastructure: The Definitive Guide to Building a Sustainable Ethical Hacking Lab on AWS (2025 Edition)

By Muhammad Yousaf Raza
MS/MPhil Information Security | Certified Ethical Hacker (CEH)

Abstract

The paradigm of information security training and operations has fundamentally shifted from localized, hardware-constrained environments to scalable, cloud-native infrastructures. For the modern cybersecurity scholar and practitioner, the ability to rapidly provision, secure, and weaponize cloud resources is not merely a logistical convenience but a core competency. This comprehensive technical report provides an exhaustive methodology for engineering a persistent, cost-neutral penetration testing laboratory using Amazon Web Services (AWS). We will meticulously dissect the post-July 2025 AWS Free Tier architecture, navigate the complexities of cross-platform cryptographic access via PuTTY and WinSCP, and implement a robust, maintainable offensive toolset on Ubuntu Linux using the Penetration Testers Framework (PTF). This document serves as a reference architecture for students, researchers, and security professionals seeking to emulate real-world engagement environments while adhering to strict fiscal constraints.

1. The Strategic Imperative of Cloud-Based Security Labs

The traditional approach to building an ethical hacking lab involved virtualization software (like VMware or VirtualBox) running locally on a laptop. While functional, this model suffers from resource contention—running a Kali Linux attacker machine alongside multiple victim targets (Metasploitable, Windows Servers) can quickly exhaust the RAM and CPU of a standard workstation. Furthermore, local labs do not simulate the remote engagement model that characterizes modern red teaming and penetration testing.

By migrating this infrastructure to the cloud, specifically AWS, we achieve three critical objectives:

1. **Scalability:** We can provision compute resources (EC2) that exceed the capabilities of local hardware, leveraging AWS's vast infrastructure.
2. **Accessibility:** The lab becomes accessible from any location with an internet connection, decoupling the toolkit from a specific physical device.
3. **Realism:** Conducting assessments against remote cloud servers mimics the latency, firewall configurations, and network segmentation hurdles found in actual corporate environments.

However, the cloud is a metered utility. Without a deep understanding of the billing logic, "free" trials can quickly transform into significant financial liabilities. This report begins by decoding the financial architecture of AWS as it stands in late 2025.

---

2. Decoding the AWS Free Tier Architecture: The 2025 Paradigm Shift

To effectively utilize AWS for a 12-month period without cost, one must navigate the significant structural changes introduced to the billing model on July 15, 2025. The strategies that worked for the past decade are no longer universally applicable.

2.1 The Bifurcation of Account Benefits

Historically, the AWS Free Tier was a monolithic offering: new accounts received 12 months of free usage for specific resource types, most notably 750 hours per month of t2.micro or t3.micro EC2 compute capability. This allowed for a "always-on" micro-instance for a full year.[1]

As of the July 2025 update, AWS has bifurcated its Free Tier benefits based on the account creation date, creating a legacy tier and a new credit-based tier. Understanding which category you fall into is the single most important factor in cost avoidance.

Feature	**Legacy Accounts (Pre-July 15, 2025)**	**New Accounts (Post-July 15, 2025)**
Duration of Benefit	12 Months from Sign-Up	**6 Months** Free Plan (or until credits exhaust) [2]
Compute Limit	750 Hours/Month (Continuous usage allowed)	Metered against **$200 USD Credit Balance** [3]

Eligible Instances	Restricted to t2.micro or t3.micro [4]	Expanded to include **t3.small**, **t4g.small**, c7i-flex.large [2]
Cost Logic	Usage cap (time-based)	Budget cap (credit-based)
Overage Penalty	Immediate billing upon exceeding 750 hours	Deductions from credit pool; billing starts after pool exhaustion

Implications for the "12-Month" Requirement:
The user requirement specifies a "12-month free trial." For a newly created account in late 2025, the nominal "Free Plan" duration is strictly 6 months.5 However, the introduction of the $200 USD credit mechanism ($100 at sign-up + $100 for engagement tasks) allows for a simulated 12-month experience if managed aggressively.3
By utilizing the credits judiciously—running the lab machine only when actively studying (e.g., 20 hours/week)—the cost per month remains negligible, allowing the credits to potentially stretch effectively over a longer horizon than the 6-month base period, although the official "Free Plan" status expires. Alternatively, students should leverage **AWS Educate**, which provides renewable credits and distinct Starter Accounts that do not require a credit card, offering a parallel pathway to sustain the lab beyond the initial commercial free tier limits.[5]

2.2 Financial Prerequisites and Identity Verification

Creating an AWS account is a binding contract. AWS employs rigorous identity verification to prevent the proliferation of botnets and crypto-mining schemes.

1. **Payment Method:** A valid credit or debit card is non-negotiable. Upon entry, AWS initiates a **$1.00 USD (or local equivalent, e.g., 2 INR)** pre-authorization charge to validate the card's liquidity.[8] This is a temporary hold, not a fee, and is typically refunded within 3-5 business days. *Note:* Prepaid cards are frequently rejected due to high fraud rates.
2. **Telephonic Verification:** An automated system validates the user's phone number. In 2025, this system supports a broader array of languages (Mandarin, Spanish, German, etc.).[8] If the SMS OTP (One-Time Password) fails to arrive due to carrier filtering—a common issue in certain geopolitical regions—users must immediately toggle to the "Voice Call" option, which is generally more reliable.
3. **Root User Integrity:** The email address used for sign-up becomes the "Root User." This identity has unlimited power, including the ability to close the account or provision $10,000/month servers. It must be secured with a complex, unique password and Multi-Factor Authentication (MFA) immediately upon creation.

2.3 The Cost-Avoidance Safety Net: Budgets and Alerts

Before a single server is launched, we must construct a financial firewall. The new credit model makes this even more critical, as burning through credits unnoticed leads to direct billing.

Configuration Protocol:

1. **Access the AWS Billing Console.**
2. **Enable Cost Explorer:** This requires 24 hours to generate initial data but must be activated on Day 1.[9]
3. **Create a Zero-Spend Budget:**
   - Navigate to **Budgets** > **Create Budget**.
   - Select **Cost Budget**.
   - Set the **Budgeted Amount** to **$0.01**.
   - **Thresholds:** Configure an alert to email you at **50% ($0.005)** and **100% ($0.01)** of the budget.
   - *Strategic Insight:* This acts as a "canary in the coal mine." If you accidentally launch a resource that isn't covered by the Free Tier (e.g., a NAT Gateway or a large EBS volume), you will be notified immediately, likely before the charge exceeds a few cents.[6]
4. **Credit Monitoring:** For post-July 2025 accounts, regularly monitor the "Credits" section in the Billing Dashboard to track the depletion rate of your $200 promotional balance.

---

3. Infrastructure Engineering: Provisioning the Compute Layer

With the account secured and billing alerts active, we proceed to the core engineering task: provisioning the Virtual Private Cloud (VPC) resources and the Elastic Compute Cloud (EC2) instance that will serve as our hacking station.

3.1 Region Strategy: Latency vs. Features

AWS operates in distinct geographic "Regions" (e.g., us-east-1 in N. Virginia, eu-central-1 in Frankfurt).

- **Selection Criteria:** Choose the region physically closest to your location to minimize keystroke latency in the SSH terminal. High latency (>200ms) makes interactive command-line work frustrating.
- **Verification:** Ensure your chosen region supports the **t3** or **t4g** instance families, which are the mainstays of the 2025 Free Tier.[2] us-east-1 (N. Virginia) is the default and most feature-rich, often receiving new instance types first.

3.2 The Operating System: Ubuntu Server 24.04 LTS

While the AWS Marketplace offers pre-packaged Kali Linux AMIs (Amazon Machine Images), professional best practice dictates building your toolkit on top of a standard, supported distribution. Pre-built AMIs often lag in updates, contain unnecessary bloatware, or use non-standard kernels that complicate module compilation.

Recommendation: Ubuntu Server 24.04 LTS (Noble Numbat).

- **Reasoning:** Ubuntu 24.04 is the latest Long Term Support release as of this writing. It offers native compatibility with the apt package management ecosystem used by Kali tools. It is lightweight, stable, and widely supported by the security community.[10]
- **Architecture Selection (x86 vs. ARM):**
  - **x86_64 (Intel/AMD):** The safe, standard choice. Virtually all legacy security binaries, exploit code, and pre-compiled scripts are designed for x86.
  - **ARM64 (Graviton):** The modern, efficient choice. AWS offers the t4g.small instance in the free tier, which runs on AWS Graviton processors.[2] While faster and more energy-efficient, using ARM can present challenges when trying to run older, closed-source security tools or binary exploits that lack ARM compilation. **For this guide, we select x86_64 to ensure maximum compatibility with the broadest range of ethical hacking tools.**

3.3 Instance Sizing and Family Selection

This is the most critical decision for cost management.

- **For Legacy Accounts:** You are restricted to **t2.micro** (1 vCPU, 1 GB RAM) or **t3.micro** if t2 is unavailable. These are modest resources. You must run a swap file to prevent out-of-memory crashes during heavy tasks (like compiling tools).
- **For New Accounts (Post-July 2025):** The credit model allows access to larger instances. The **t3.small** (2 vCPUs, 2 GB RAM) is highly recommended. It consumes credits faster than a micro instance but offers double the RAM, which is transformative for running memory-intensive tools like Metasploit, Burp Suite, or vulnerability scanners.[2]
  - *Warning:* Ensure **T2/T3 Unlimited Mode** is **Disabled**. Unlimited mode allows the instance to burst beyond its baseline CPU credits, charging extra fees once credits are exhausted. We want the instance to throttle rather than charge us.[12]

3.4 Key Pair Cryptography: The Access Token

AWS EC2 instances do not allow password-based login by default. They utilize public-key cryptography.

1. **Creation:** During the launch wizard, create a new key pair.
2. **Naming:** Kali_Lab_Key_2025.
3. **Type: RSA**. While ED25519 is newer and faster, RSA remains the most universally compatible with legacy versions of access tools.
4. **Format: .pem** (Privacy Enhanced Mail). This is the OpenSSH standard. We will convert

this for Windows use later.

5. **Custody:** Download this file immediately. AWS does not store the private key. If lost, the instance becomes a digital brick—inaccessible and unrecoverable.

3.5 Network Security Groups: The First Line of Defense

An AWS Security Group acts as a stateful virtual firewall controlling traffic at the instance level. The default setting often allows SSH (Port 22) from 0.0.0.0/0 (The entire internet). **This is a critical vulnerability.** Botnets scan AWS IP ranges continuously, launching thousands of brute-force attacks per hour against open SSH ports.

Hardening Configuration:

- **Name:** Ethical-Hacking-Lab-SG.
- **Inbound Rule 1 (SSH):**
  - **Type:** SSH (TCP/22).
  - **Source: My IP** (Select this from the dropdown). This restricts access solely to your current physical location. If your home IP changes (dynamic IP), you must update this rule in the AWS Console before you can connect again. This slight inconvenience provides a massive upgrade in operational security (OpSec).
- **Inbound Rule 2 (Reverse Shells/Listeners - Optional):**
  - **Type:** Custom TCP.
  - **Port Range:** 4444 (Default Metasploit) or 8000-8100.
  - **Source: My IP**. Open these only when testing reverse connections back to your lab machine.

3.6 Storage Provisioning (EBS)

The Free Tier includes **30 GB** of General Purpose SSD (gp2 or gp3) storage.[2]

- **Default:** Ubuntu AMIs default to 8 GB. This is insufficient for a full pentest suite.
- **Optimization:** Increase the root volume size to **25 GB**. This leaves a 5 GB buffer (unallocated) to stay safely within the 30 GB limit, avoiding accidental overage charges if a snapshot creates a backup volume.
- **Type:** Select **gp3**. It creates a decoupling between IOPS (Input/Output Operations Per Second) and storage size, offering a baseline of 3000 IOPS regardless of volume size, which is superior to the older gp2 standard for small disks.[2]

---

4. Cryptographic Access and Connectivity: The PuTTY & WinSCP Bridge

For users operating on Windows, connecting to a Linux cloud instance requires bridging the gap between Windows' internal logic and the Linux SSH protocol. We rely on the PuTTY suite and WinSCP, industry-standard tools for remote administration.

4.1 The Protocol Gap: PEM vs. PPK

AWS provides a .pem file (standard OpenSSH format). However, the PuTTY client (and by extension, WinSCP's integration with it) historically relies on its proprietary .ppk (PuTTY Private Key) format. While modern versions of PuTTY *can* accept .pem files, converting to .ppk is the most robust method to ensure compatibility across the entire PuTTY/WinSCP ecosystem.[13]

The Conversion Protocol:

1. **Download:** Ensure **PuTTY** and **PuTTYgen** (Key Generator) are installed.
2. **Launch PuTTYgen.**
3. **Load Key:** Click **Load**. In the file dialog, change the filter from "PuTTY Private Key Files (_.ppk)" to **"All Files (_.*)"** to make your Kali_Lab_Key_2025.pem visible. Select it.
4. **Import:** You will receive a notice: "Successfully imported foreign key."
5. **Passphrase Protection (Crucial Step):** Enter a strong passphrase in the "Key passphrase" and "Confirm passphrase" fields.
   - *Why?* If your .ppk file is stolen from your laptop by malware, the attacker cannot use it without this passphrase. This is "Data at Rest" encryption for your credentials.
6. **Save:** Click **Save private key**. Name it Kali_Lab_Key_2025.ppk.

4.2 Establishing the SSH Command Channel (PuTTY)

PuTTY provides the terminal interface (Command Line Interface - CLI) to control the server.

1. **Host Name:** Locate the **Public IPv4 address** of your instance in the AWS Console (e.g., 54.123.45.67). Enter ubuntu@54.123.45.67 in the Host Name field.
   - *Note on Usernames:* The default user is strictly defined by the AMI. For Ubuntu, it is ubuntu. For Amazon Linux, ec2-user. For Debian, admin. Using the wrong user will result in immediate rejection.[15]
2. **Auth Configuration:**
   - Navigate to **Connection** > **SSH** > **Auth** > **Credentials**.
   - In the "Private key file for authentication" field, browse to your newly created .ppk file.
3. **Session Persistence:**
   - Return to the **Session** category at the top.
   - In "Saved Sessions," type AWS-Kali-Lab.
   - Click **Save**.
4. **Connection:** Click **Open**.
5. **Trust Verification:** On the first connection, a security alert will display the server's **host key fingerprint** (e.g., ssh-ed25519 256 SHA256:...). Compare this with the fingerprint shown in the "System Log" of the AWS Console to guarantee you are not being subjected to a Man-In-The-Middle (MITM) attack. If they match, click **Accept**.

4.3 Establishing the File Transfer Channel (WinSCP)

WinSCP allows for the graphical transfer of files (exploits, wordlists, exfiltrated data) between your local machine and the cloud. However, managing system files on the remote server requires elevated privileges (root), which introduces a specific technical challenge in Ubuntu 24.04.

The "Sudo" Privilege Escalation Challenge:
By default, you log in as the user ubuntu, which has standard privileges. You cannot upload a file directly to /usr/bin/ or edit /etc/proxychains.conf via the WinSCP GUI because the SFTP subsystem runs as ubuntu, not root.
The Solution: SFTP Sudo Masquerading
We must instruct WinSCP to launch the SFTP server process on the remote machine using sudo immediately upon connection.16

1. **Session Setup:**
    - **File Protocol:** SFTP.
    - **Host Name:** Your instance Public IP.
    - **User Name:** ubuntu.
    - **Advanced > SSH > Authentication:** Select your .ppk key.
2. **Advanced Environment Configuration:**
    - In the **Advanced Site Settings** dialog, navigate to **Environment** > **SFTP**.
    - Locate the **"SFTP Server"** field. It defaults to "Default".
    - We must replace this with a command that invokes sudo.
    - *Critical Path Variance:* In **Ubuntu 24.04**, the path to the SFTP binary may have shifted or requires explicit invocation. The robust command to use is:

      ```bash
      sudo /usr/lib/openssh/sftp-server
      ```

    - *Verification:* If connection fails, try sudo /usr/libexec/openssh/sftp-server (common in newer Debian variants) or simply sudo su -c /usr/lib/sftp-server.[19] You can verify the exact path by running find / -name sftp-server in your PuTTY terminal first.
3. **Protocol Versioning (Ubuntu 24.04 Specifics):**
    - Ubuntu 24.04 utilizes OpenSSH 9.x, which has deprecated the legacy SCP protocol in favor of SFTP-under-the-hood.[11] Ensure WinSCP is updated to the latest version to handle the handshake correctly.
4. **Save and Connect:** Save these settings. When you login, WinSCP will ask for your key passphrase (if set). Once connected, you will have full **root** write access to the file system via the GUI, essential for editing configuration files rapidly.

---

5. The Arsenal: Weaponizing the Instance with

Maintained Repositories

The core requirement of this project is to install "necessary Kali Linux tools" using a "regularly maintained GitHub repo." This is a contentious area in the Linux community. Many scripts (like the once-popular Katoolin) are now deprecated, unmaintained, and dangerous—often breaking the host operating system by mixing Debian repositories with Ubuntu repositories (a "Franken-Debian" state).[21]

5.1 The Selection: Penetration Testers Framework (PTF)

To satisfy the requirement for a **maintained** and **stable** repository, we select the **Penetration Testers Framework (PTF)** developed by TrustedSec.

Why PTF?

- **Architecture:** Unlike simple scripts that just apt-get install, PTF fetches the latest source code from the developers' own GitHub repositories (e.g., cloning the metasploit-framework repo directly), compiles it, and links it. This ensures you have the absolute bleeding-edge version of every tool, often newer than what is in the standard Kali repositories.[22]
- **Isolation:** It installs tools into a dedicated /pentest/ directory, keeping the base Ubuntu system clean and stable.[24]
- **Maintenance:** It is actively maintained by the professional security community.[25]

Alternative Analysis:

- **Katoolin/Katoolin3:** While Katoolin3 exists as a Python 3 fork, the underlying logic of adding Kali repos to Ubuntu is inherently risky and prone to glibc dependency conflicts.[21]
- **Kali-on-Linux (Andrea055):** This is a lighter script [10], but lacks the granular module management and enterprise-grade reliability of PTF.

5.2 System Preparation and Dependency Resolution

Before deploying PTF, we must prepare the Ubuntu 24.04 environment.

Step 1: Update the Base System
Execute via PuTTY:

Bash

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Core Dependencies

PTF requires Python and Git.

Bash

```bash
sudo apt install git python3 python3-pip python3-venv build-essential -y
```

Step 3: The LibSSL Compatibility Fix (The Ubuntu 22.04/24.04 Hurdle)
Many older security tools rely on libssl1.1, which was deprecated in Ubuntu 22.04 in favor of libssl3. This causes installation failures for tools that haven't been updated.[26] To preempt this, we manually install the legacy library:

Bash

```bash
wget
http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2_amd64.deb
sudo dpkg -i libssl1.1_1.1.1f-1ubuntu2_amd64.deb
```

Note: If wget fails due to link rot (URLs change), search packages.ubuntu.com for the current libssl1.1 amd64 .deb link.

5.3 Installing and Configuring PTF

Now we deploy the framework.

Step 1: Clone the Repository
We adhere to the Filesystem Hierarchy Standard by placing third-party apps in /opt.

Bash

```bash
cd /opt
sudo git clone https://github.com/trustedsec/ptf.git
```

Step 2: Launching the Framework

Bash

```
cd ptf
sudo./ptf
```

The terminal will display the PTF banner and a ptf> prompt.

Step 3: Module Selection Strategy
PTF allows for an "Install All" approach, but this is resource-heavy and takes hours. For a focused lab, we use a surgical approach, installing only high-value tool clusters.

- **Command:** show modules (Lists all available tool categories).

Priority Installation List:

1. **The Exploitation Standard (Metasploit):**
   Bash
   ```
   ptf> use modules/exploitation/metasploit
   ptf> install
   ```
   Insight: PTF handles the Ruby dependencies and database setup automatically.
2. **Intelligence Gathering (OSINT):**
   Bash
   ```
   ptf> use modules/intelligence/theharvester
   ptf> install
   ```
3. **Network Enumeration:**
   Bash
   ```
   ptf> use modules/vulnerability-analysis/nmap
   ptf> install
   ```
4. **Web Application Assessment:**
   Bash
   ```
   ptf> use modules/vulnerability-analysis/burpsuite
   ptf> install
   ```
5. **Password Auditing:**
   Bash
   ```
   ptf> use modules/password-recovery/john
   ptf> install
   ```

Step 4: Verification

Exit PTF (exit). The tools are now installed. Verify functionality:

Bash

```
msfconsole
```

If the Metasploit console loads, the environment is successfully weaponized.

---

6. Operational Management and Cost Optimization

Building the lab is only the first step. Maintaining it requires operational discipline to ensure it remains cost-neutral and secure.

6.1 The "Stop" vs. "Terminate" Protocol

This distinction is the primary cause of unexpected bills for beginners.

- **Stopping an Instance:**
  - **Action:** Instance State > Stop.
  - **Effect:** The OS shuts down. You stop accruing **Compute Hours** (saving your credits/free tier allowance).
  - **Cost:** You **continue to pay for EBS Storage** (the 25GB disk).
  - *Verdict:* Since the Free Tier includes 30GB of EBS storage, you can leave the instance "Stopped" indefinitely without cost, provided you have only one such instance.[2]
- **Terminating an Instance:**
  - **Action:** Instance State > Terminate.
  - **Effect:** The instance is deleted. The storage volume is wiped.
  - *Verdict:* Use this only if you wish to destroy the lab permanently.

6.2 Managing the Dynamic IP (Elastic IP Warning)

When you stop and start an EC2 instance, its Public IP address changes. This breaks your saved PuTTY/WinSCP sessions.

- **The Trap:** You might be tempted to use an **Elastic IP (EIP)** to get a static address.
- **The Cost:** AWS offers one EIP for free *only if it is attached to a running instance*. If you stop your instance (to save compute credits), the EIP becomes "unattached," and AWS begins charging **$0.005 per hour** for holding the IP unused.
- **The Strategy:** Do **not** use an Elastic IP for a free lab. Simply update the Host Name in PuTTY each time you boot the server. It takes 10 seconds and saves roughly $3.60/month in potential idle fees.

6.3 Maintenance Updates

Unlike a static Kali VM, your cloud tools need updates.

- **System Updates:** Run sudo apt update && sudo apt upgrade monthly.
- **Tool Updates:** Launch PTF (sudo./ptf) and run update. This pulls the latest Git commits for all installed tools, ensuring you have the newest exploit modules and signatures.[27]

6.4 Student Alternative: AWS Educate

If the commercial Free Tier credits ($200) are exhausted or the 6-month window closes, academic users (MS/MPhil students) should pivot to **AWS Educate**.

- **Mechanism:** This program provides distinct "Starter Accounts" that do not require a credit card.
- **Benefits:** It offers a renewable pool of credits (typically $50-$100 per year) specifically for learning. While these accounts have some limitations (e.g., no heavy GPU instances), they are perfect for sustaining a Linux micro-instance for long-term research without financial risk.[5]

7. Conclusion

By executing the architecture detailed in this report, we have successfully circumvented the hardware limitations of local virtualization to deploy a professional-grade penetration testing environment on the cloud. We have navigated the complexities of the 2025 AWS billing restructuring, leveraging the credit model to emulate a 12-month free trial. We have bypassed the instability of deprecated scripts by adopting the robust Penetration Testers Framework, ensuring our toolkit remains relevant and maintainable.

For the information security researcher, this lab is more than a collection of tools; it is a scalable, disposable, and accessible dojo. It allows for the simulation of complex attack vectors against cloud-native infrastructure—a skill set that is increasingly mandatory in the modern threat landscape.

Disclaimer: *This guide is provided for educational and authorized research purposes only. The deployment of offensive security tools against networks or systems without explicit, written permission from the owner is a violation of federal and international law. The author and publisher assume no liability for the misuse of the infrastructure described herein.*

Appendix A: Troubleshooting Matrix

Symptom	Probable Cause	Corrective Action
PuTTY Network Timeout	Security Group Misconfiguration	Ensure Port 22 is open in the AWS Security Group. Verify the "Source" IP matches your current public IP (Google "What is my IP").
"Permission Denied (publickey)"	Wrong Username or Key Format	1. Use user ubuntu (not root or ec2-user). 2. Ensure the .ppk was generated from the specific .pem assigned to *this* instance.
WinSCP "Cannot overwrite" Error	Permission Issues	Ensure the SFTP server path in WinSCP Advanced Settings is set to sudo /usr/lib/openssh/sftp-server or sudo /usr/libexec/openssh/sftp-server depending on the exact Ubuntu sub-version.[19]
PTF Installation Failure	Missing Dependencies	Run sudo apt --fix-broken install. Manually install libssl1.1 as described in Section 5.2.
Unexpected AWS Charges	Idle Elastic IP or NAT Gateway	check the "Bills" page immediately. Release any unattached Elastic IPs. Ensure you are not running "T3 Unlimited" mode.