**Project Progress and Review**

You have successfully completed the **network connectivity setup** between AWS and Azure, configured **network security groups (NSGs) and firewall rules**, and attempted **monitoring and alerting setup** in Azure. However, based on your initial plan, your progress has slightly deviated. Below is a structured review:

**Current Progress**

✅ **Network Connectivity Between AWS and Azure**

* WireGuard-based VPN setup between AWS and Azure.
* Security groups configured for SSH, WireGuard, HTTP, and HTTPS.
* Verified traffic between AWS and Azure.

✅ **Firewall & Network Security Group Rules**

* Configured inbound and outbound rules for required ports.
* Debugged issues related to ping failures and connectivity.

✅ **Monitoring & Alerts (Azure)**

* Created a metric alert in Azure for CPU usage.
* Verified alert creation via CLI but not visible in VM.
* Tested high CPU load using the stress command, but alerts were not triggered.

**Areas Where Progress is Not Aligned with Your Plan**

❌ **IAM Roles and Policies for AWS and Azure**

* Not yet implemented.
* Required for access control, least privilege enforcement, and compliance.

❌ **S3 Bucket Setup & Configuration**

* Not yet implemented.
* Needed for storage, logging, and security configurations.

❌ **Logging & Threat Detection**

* AWS CloudTrail, GuardDuty, and Splunk integration not yet implemented.
* Required for monitoring AWS/Azure environments for suspicious activities.

❌ **Automated Incident Response**

* No work done yet on SOAR tools (Security Orchestration, Automation, and Response).
* Automating response actions to security events in AWS and Azure.

❌ **Penetration Testing**

* Not yet conducted.
* Required for evaluating security defenses and identifying vulnerabilities.

**Next Steps (Based on Your Plan)**

1️⃣ **Set Up IAM Roles & Policies (AWS & Azure)**

* Create IAM roles with least privilege access in AWS & Azure.
* Assign necessary permissions for CloudTrail, GuardDuty, S3, and other services.

2️⃣ **Set Up S3 Buckets (AWS) & Storage (Azure Blob Storage)**

* Create S3 buckets for log storage, backups, and access control.
* Enable encryption, versioning, and logging for security.

3️⃣ **Implement Logging & Threat Detection**

* Enable **AWS CloudTrail** for auditing AWS API calls.
* Activate **AWS GuardDuty** for continuous monitoring.
* Integrate logs from AWS/Azure into **Splunk** for analysis.

4️⃣ **Automate Incident Response with SOAR**

* Set up an open-source SOAR tool like TheHive or Wazuh.
* Automate responses to detected security incidents.

5️⃣ **Conduct Penetration Testing**

* Perform security testing on the AWS-Azure environment.
* Identify and mitigate vulnerabilities in cloud infrastructure.

**Conclusion**

Your project is progressing well in **network setup and security rules**, but **logging, automation, and security analysis** still need to be implemented. To stay on track, you should now focus on **IAM roles, logging, and automation** before penetration testing.

Would you like a **detailed execution plan** for the next steps, starting with IAM roles?

**Execution Plan for Next Steps**

**Step 1: Set Up IAM Roles & Policies (AWS & Azure)**

**Why?**

* IAM roles ensure **least privilege access** for services like CloudTrail, GuardDuty, and Splunk.
* Prevents unauthorized access to logs and security data.

**Actions (AWS)**

✅ **Create IAM Role for CloudTrail & S3 Access**

1. Go to **AWS IAM Console** → **Roles** → **Create Role**.
2. Select **AWS Service** → **S3 & CloudTrail**.
3. Attach the policy **AWSCloudTrailFullAccess** and **AmazonS3FullAccess** (or least privilege access).
4. Name it **CloudTrail-S3-Role** and create the role.

✅ **Create IAM Role for GuardDuty & Security Monitoring**

1. Go to **AWS IAM Console** → **Roles** → **Create Role**.
2. Select **AWS Service** → **GuardDuty**.
3. Attach **AmazonGuardDutyFullAccess** policy.
4. Name it **GuardDuty-Role** and create the role.

✅ **IAM Setup in Azure**

1. Go to **Azure Portal** → **Azure Active Directory** → **Roles and Administrators**.
2. Assign **Monitoring Contributor** role for **Log Analytics & Threat Detection**.

**Step 2: Set Up S3 Buckets & Configure Logging**

**Why?**

* Store logs from **CloudTrail**, **GuardDuty**, and **Splunk**.
* Helps in **security auditing** and **threat detection**.

**Actions (AWS)**

✅ **Create S3 Bucket for Logs**

1. Go to **AWS S3 Console** → **Create Bucket**.
2. Name: **security-logs-bucket**.
3. Enable **Versioning & Encryption (SSE-S3 or SSE-KMS)**.

✅ **Enable CloudTrail Logging to S3**

1. Go to **AWS CloudTrail** → **Create Trail**.
2. Select **Write to S3** → Choose **security-logs-bucket**.
3. Enable **Event Logging** (Read/Write events).

✅ **Enable GuardDuty Logging to S3**

1. Go to **AWS GuardDuty** → **Settings** → **Enable Findings Export to S3**.
2. Choose **security-logs-bucket** for storage.

✅ **Splunk Integration (Optional, If Using S3 for Logs)**

1. Install **Splunk Add-on for AWS** in Splunk.
2. Configure **S3 input** to read logs from **security-logs-bucket**.

**Step 3: Implement Logging & Threat Detection**

**Why?**

* Monitors **AWS activity logs** for security breaches.
* Detects threats using **GuardDuty**.

**Actions (AWS)**

✅ **Enable AWS CloudTrail**

* Go to **CloudTrail Console** → **Enable Event History**.
* Ensure it logs to **S3 & CloudWatch**.

✅ **Enable AWS GuardDuty**

* Go to **GuardDuty Console** → **Enable**.
* Activate **Threat Intelligence** (Malware detection, anomaly detection).

✅ **Configure Splunk to Monitor Logs**

* Connect **CloudTrail logs** and **GuardDuty logs** in Splunk.
* Create **custom dashboards for security analysis**.

**Step 4: Automate Incident Response with SOAR**

**Why?**

* Reduces **manual effort** in responding to security threats.
* Automates security actions (like blocking IPs or sending alerts).

**Actions (Using Free SOAR Tools like TheHive, Wazuh, or AWS Lambda)**

✅ **Option 1: AWS Lambda for Auto Response**

* Write a **Lambda function** to **block IPs** in Security Groups based on GuardDuty alerts.
* Trigger this function using **CloudWatch Events**.

✅ **Option 2: Use Wazuh (Open-Source SOAR)**

* Install **Wazuh** for real-time security alerts.
* Configure automatic actions (like disabling a compromised IAM user).

**Step 5: Conduct Penetration Testing**

**Why?**

* To identify **vulnerabilities in the AWS-Azure infrastructure**.

**Actions**

✅ **Run Security Scans**

* Use **Nmap**, **Burp Suite**, and **Metasploit** to scan AWS and Azure networks.
* Check for **open ports, misconfigurations, and weak authentication**.

✅ **Simulate Attacks (Ethical Hacking)**

* Perform **brute-force attacks** on SSH to test IAM policies.
* Check for **misconfigured S3 buckets** (public access issues).

**Conclusion**

🔹 **Your progress is strong on networking, but IAM, Logging, SOAR, and Pentesting are pending.**  
🔹 **Next Immediate Steps:** IAM Roles & S3 Buckets → CloudTrail & GuardDuty → Automated Response → Penetration Testing.

Would you like me to provide exact **commands/scripts** for IAM setup or logging?