**Conditional Access as Code Lab: Prerequisites Guide**

This guide covers all the prerequisites required to implement Conditional Access as Code using Terraform, Azure AD, and GitHub Actions.

**1. Azure Requirements**

**1.1 Azure Subscription**

* You must have access to at least one active Azure Subscription.
* The service principal (app registration) must be assigned the **Contributor** role in that subscription.

**1.2 Azure AD Premium P1 or P2 License**

* Required for Conditional Access policies.
* Ensure the test tenant is licensed.

**2. Azure Active Directory (AAD) Configuration**

**2.1 Create a Test Group**

* **Name**: TestGroup-CaPolicy
* **Membership type**: Assigned
* Add a **test user** to this group

**2.2 Create a Test User**

* **Name**: testuser@yourtenant.onmicrosoft.com
* Assign the user to the test group created above.
* Ensure the user has a valid license (P1 or P2).

**2.3 Trusted Location (Named Location)**

* Go to **Azure Active Directory → Security → Conditional Access → Named Locations**
* Create a new **IP range location**:
  + **Name**: OfficeLocation
  + Add your current IP address or a range (e.g., 203.0.113.0/24)
  + Mark as trusted location

**3. App Registration (Service Principal)**

**3.1 Register an App**

* **Name**: Terraform-Deployment-App
* **Supported account types**: Single tenant

**3.2 Create Client Secret**

* Go to **Certificates & secrets → Client secrets**
* Add a new secret and **copy the value**

**3.3 Assign API Permissions**

Go to **API permissions → Add a permission → Microsoft Graph → Application**

* Directory.ReadWrite.All
* Policy.ReadWrite.ConditionalAccess
* User.Read

Ensure all are **Application** type, except User.Read (can remain Delegated).

Click **Grant admin consent** after adding permissions.

**4. Assign Subscription Role to App**

Go to **Subscriptions → Access control (IAM) → Add role assignment**

* **Role**: Contributor
* **Assign access to**: User, group, or service principal
* Select: Terraform-Deployment-App

**5. GitHub Repository Setup**

**5.1 Add Secrets in GitHub**

Go to **GitHub → Repo → Settings → Secrets and variables → Actions → New repository secret**:

| **Name** | **Value from App Registration** |
| --- | --- |
| ARM\_CLIENT\_ID | Application (client) ID |
| ARM\_CLIENT\_SECRET | Client secret **value** |
| ARM\_TENANT\_ID | Directory (tenant) ID |
| ARM\_SUBSCRIPTION\_ID | Azure subscription ID |

**You’re now ready to run the Conditional Access as Code lab using Terraform and GitHub Actions.**

**Conditional Access Policy Testing Checklist**

**Purpose:**

Ensure the Conditional Access (CA) policy deployed via Terraform and GitHub Actions is applied correctly and enforces the intended security logic.

**Use Case Summary:**

**Block Office 365 access from non-compliant devices outside trusted locations unless MFA is used.**

**Pre-Deployment Checklist:**

* Azure AD Premium P1 license is active
* Target group is created and Object ID is available
* Trusted location is configured in Azure AD > Named Locations
* Test user is part of the included group in the policy
* GitHub secrets are configured (ARM\_CLIENT\_ID, etc.)

**Post-Deployment Verification:**

* Terraform deployment completed successfully
* Policy appears in Azure AD > Conditional Access
* Policy display name matches: "Block Non-Compliant Devices"
* Conditions match the configuration
* Grant control requires MFA

**Manual Testing Scenarios**

| **Scenario ID** | **Test Description** | **Expected Behavior** |
| --- | --- | --- |
| TC-001 | Login from a compliant device inside trusted location | Allow access |
| TC-002 | Login from non-compliant device inside trusted location | Block or MFA prompt |
| TC-003 | Login from compliant device outside trusted location | MFA prompt |
| TC-004 | Login from non-compliant device outside trusted location | Block access |
| TC-005 | Login from a user not in the policy group | No policy applied |

**Sign-in Log Review**

* Go to Azure AD > Sign-in Logs
* Filter by test user
* Check "Conditional Access" tab for each test case
* Confirm applied policy name and result (Success, Failure, Not Applied)

**Notes & Observations**

(Add notes during testing)

* TC-001: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* TC-002: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* TC-003: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* TC-004: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* TC-005: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Intern's Checklist Completion**

* I tested each scenario with proper documentation
* I reviewed the logs and results match expectations
* I reported any policy mismatch or error to the team lead
* I understand how to debug policy issues in Azure AD