

A close up of a sign

Description generated with high confidence

InternalOperations-ServiceNow-ProdGeoHierarchy-V1 Azure API Management DevOps Change Request Documentation

Procter and Gamble Enterprise API Management

Enterprise Application and Integration

# Purpose

This document is for the use of the API Management Support Team. This will aid in smoother processing of API deployment. It should be clear here all the necessary changes that need to be implemented during the process. It also serves as the storage for the test documents and approvals.

**API Name:** **InternalOperations-ServiceNow-ProdGeoHierarchy-V1**

**Version:** **1**

**Author:** **Heidi Embat**

**Creation Date: July 02, 2025**

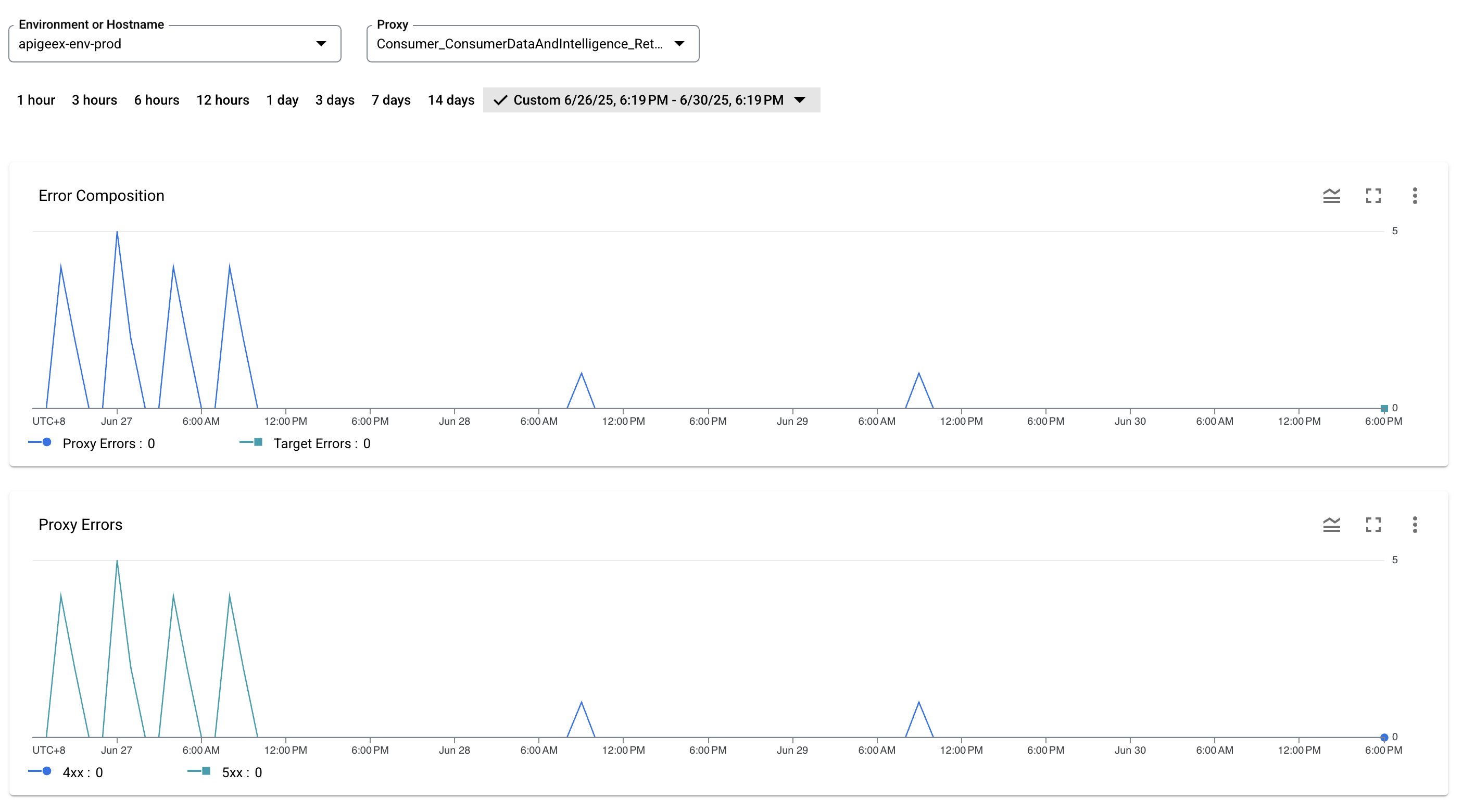
**Document Information**

|  |  |
| --- | --- |
| Application Area | APIM Team/API Owner should add value here based on New Engagement Ticket/Mega ID/ leanix |
| Owners Email | APIM Team/API Owner should add value here based on New Engagement Ticket |
| Region | APIM Team/API Owner should add value here based on New Engagement Ticket/Mega ID/ leanix |

**Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Approved by** | **Name** | **Role** | **Date** |
| **Process Team Lead** | TBA | TBA | TBA |
| **Development Team Lead** | TBA | TBA | TBA |
| **Governance Team** | TBA | TBA | TBA |

# Data Flow Diagram



**Data Flow Diagram Structure**

**(P&G Client -> WAF/Cloud Armor -> Authentication Method-> API Policies -> Target/Backend)**

|  |  |
| --- | --- |
| # | **API Policies** |
| **1** | **Base** |
| **2** | **Basic Auth** |
| **3** | **Conditional Logic** |
| **4** | **Forward Request** |
| **5** | **Header Control** |
| **6** | **Outbound** |
| **7** | **Quota** |
| **8** | **SpikeArrest** |
| **9** | **Variable Assignment** |

# Test Documentation

[Attach the updated Test Case Excel sheet OR the URL to the document in P&G SharePoint. If this has already been attached to the Change Request ticket, indicate the file name here.]

# Business Sign-off

[Attach the most recent Business Sign-off OR the URL to the document in P&G SharePoint. If this has already been attached to the Change Request ticket, indicate the file name here.]

***https://github.com/procter-gamble/EAI-APIM-Ext-InternalOperations-ServiceNow-ProdGeoHierarchy-V1/pull/41***

# Technical Specifications and Configurations

## General Details

|  |  |
| --- | --- |
| **Category** | **Description** |
| **API Questionnaire** |  |
| **API URL** | *https://API.pgcloud.com/it/itserviceoperation/v1/rds\_productandgeositehierarchy* |
| **Path** | *it/itserviceoperation/v1/rds\_productandgeositehierarchy* |
| **Backend URL** | *https://servicesboomiazureqart.pg.com:9093/ws/simple* |
| **Criticality of API** | TBA |

## Policy Configuration and Proxy/Target Endpoints

|  |  |  |
| --- | --- | --- |
| # | **API Policies** | **Reference** |
| **1** | **outbound** | **<outbound>, <backend> section elements.  - To apply a policy to the incoming request (before it is forwarded to the backend service), place a corresponding policy element within the <inbound> section element.  - To apply a policy to the outgoing response (before it is sent back to the caller), place a corresponding policy element within the <outbound> section element.  - To add a policy, place the cursor at the desired insertion point and select a policy from the sidebar.  - To remove a policy, delete the corresponding policy statement from the policy document.  - Position the <base> element within a section element to inherit all policies from the corresponding section element in the enclosing scope.  - Remove the <base> element to prevent inheriting policies from the corresponding section element in the enclosing scope.  - Policies are applied in the order of their appearance, from the top down.  - Comments within policy elements are not supported and may disappear. Place your comments between policy elements or at a higher level scope.  ##########################################  ################ Metadata ################  ##########################################  ## API Name: InternalOperations\_ServiceNow\_ProdGeoHierarchy\_V1  ## API Short Name: InternalOperations-ServiceNow-ProdGeoHierarchy-V1  ## Service Now Ticket No: RITM6851374  ########################################## --> <policies>  <inbound>  <base />** |
| **2** | **authentication-basic** | **<authentication-basic username="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-username}}" password="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-password}}" />** |
| **3** | **set-variable** | **<set-variable name="callerIp" value="@{  string[] value;  if (context.Request.Headers.TryGetValue("X-Forwarded-For", out value)) {</>** |
| **4** | **rate-limit-by-key** | **<rate-limit-by-key  calls="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-ratelimit-calls}}"  renewal-period="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-ratelimit-renewal-period}}"  counter-key="@(context.Variables.GetValueOrDefault<string>("callerIp"))"  increment-condition="@(context.Response.StatusCode >= 200 && context.Response.StatusCode < 300)"  />** |
| **5** | **quota-by-key** | **<quota-by-key  calls="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-quota-calls}}"  renewal-period="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-quota-renewal-period}}"  counter-key="@(context.Variables.GetValueOrDefault<string>("callerIp"))"  increment-condition="@(context.Response.StatusCode >= 200 && context.Response.StatusCode < 400)"  />** |
| **6** | **set-variable** | **<set-variable name="allowedIps" value="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-allowed-ips}}" />** |
| **7** | **set-variable** | **<set-variable name="appGatewayIpAdress" value="@(context.Request.IpAddress)" />** |
| **8** | **set-variable** | **<set-variable name="validCIDRs" value="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-allowed-cidrs}}" />** |
| **9** | **set-variable** | **<set-variable name="callerIpInValidRange" value="@{  int HostToNetworkOrder(int host)  {  return (((int)HostToNetworkOrderShort((short)host) & 0xFFFF) << 16) | ((int)HostToNetworkOrderShort((short)(host >> 16)) & 0xFFFF);  }  short HostToNetworkOrderShort(short host)  {  return (short)((((int)host & 0xFF) << 8) | (int)((host >> 8) & 0xFF));  }  string ipAddress = context.Variables.GetValueOrDefault<string>("callerIp");  List<string> cidrList = context.Variables.GetValueOrDefault<string>("validCIDRs").Split(',').ToList();  string ret = "false";  if (cidrList.Contains("0.0.0.0/0"))  {  ret = "true";  }  else if (!string.IsNullOrEmpty(ipAddress))  {  foreach (string cidrAddress in cidrList)  {  string[] cidrParts = cidrAddress.Split('/');  string[] inputIPParts = ipAddress.Split('.');  string[] cidrIPArray = cidrParts[0].Split('.');  if (inputIPParts.Length == 4 && cidrIPArray.Length == 4)  {  byte[] inputIPBytes = new byte[] {  Convert.ToByte(int.Parse(inputIPParts[0])),   Convert.ToByte(int.Parse(inputIPParts[1])),   Convert.ToByte(int.Parse(inputIPParts[2])),  Convert.ToByte(int.Parse(inputIPParts[3]))  };  byte[] cidrIPBytes = new byte[] {  Convert.ToByte(int.Parse(cidrIPArray[0])),   Convert.ToByte(int.Parse(cidrIPArray[1])),   Convert.ToByte(int.Parse(cidrIPArray[2])),  Convert.ToByte(int.Parse(cidrIPArray[3]))  };  int cidrAddr = BitConverter.ToInt32(cidrIPBytes,0);  int ipAddr = BitConverter.ToInt32(inputIPBytes,0);  var host = int.Parse(cidrParts[1]);  host = -1 << (32-host);  var mask = HostToNetworkOrder(host);  if (((ipAddr & mask) == (cidrAddr & mask)))  {  ret = "true";  }  }  }  }  return ret;  }" />** |
| **10** | **choose** | **<choose>  <when condition="@(context.Variables.GetValueOrDefault<string>("callerIpInValidRange")=="false")">  <choose>  <when condition="@(context.Variables.GetValueOrDefault<string>("allowedIps").Split(',').Contains(context.Variables["callerIp"]))" />  <otherwise>  <return-response>  <set-status code="401" reason="Forbidden" />  <set-header name="ErrorMessage" exists-action="override">  <value />  </set-header>  <set-header name="Content-Type" exists-action="override">  <value>application/json</value>  </set-header>  <set-body>{"statusCode": "401" ,"Message": "This Ip Address is forbidden"}</set-body>  </return-response>  </otherwise>  </choose>** |
| **11** | **set-variable** | **<set-variable name="queryparmregexp" value="@{  String s\_queryparm = "", s\_queryval = "";  int i = 0;  String currurl = context.Request.Url.ToString();  bool s = false;  string regstr = @"[\s]\*((delete)|(exec)|(drop\s\*table)|(insert)|(shutdown)|(update)|(\bor\b))";  Regex reg = new Regex(regstr);  if (currurl != "") {  var matches = Regex.Matches(currurl, @"[\?&](([^&=]+)=([^&=#]\*))", RegexOptions.Compiled);  foreach (Match m in matches) {  s\_queryparm = m.Groups[2].Value;//query parameter  s\_queryval = m.Groups[3].Value;//query parameter value  if (s == false)  {  if (reg.IsMatch(s\_queryval.ToLower()))//query parameter value  {  s = true;  }  }  if (s == false)  {  if (reg.IsMatch(s\_queryparm.ToLower()))//query name  {  s = true;  }  }  }  }  return s;  }" />** |
| **12** | **choose** | **<choose>  <when condition="@(context.Variables.GetValueOrDefault<bool>("queryparmregexp"))">  <return-response>  <set-status code="400" reason="Bad Request" />  <set-header name="ErrorMessage" exists-action="override">  <value>SQL Injection</value>  </set-header>  <set-header name="Content-Type" exists-action="override">  <value>application/json</value>  </set-header>  <set-body template="liquid">{"statusCode": "400" ,"Message": "SQL Injection detected"}</set-body>  </return-response>  </when>  <otherwise />  </choose>** |
| **13** | **choose** | **<choose>  <when condition="@(context.Request.Method.Equals("POST") || context.Request.Method.Equals("PUT"))">  <set-variable name="bodySize" value="@(context.Request.Headers["Content-Length"][0])" />  <choose>  <!-- Bodysize is expressed in Bytes -->  <when condition="@(int.Parse(context.Variables.GetValueOrDefault<string>("bodySize"))>{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-request-size}})">  <!-- one or more policy statements to be applied if the above condition is true -->  <return-response>  <set-status code="400" reason="Bad Request" />  <set-header name="ErrorMessage" exists-action="override">  <value>Message too large</value>  </set-header>  <set-header name="Content-Type" exists-action="override">  <value>application/json</value>  </set-header>  <set-body template="liquid">{"statusCode": "400" ,"Message": "Message request too large"}</set-body>  </return-response>  </when>  <otherwise />  </choose>** |
| **14** | **set-variable** | **<set-variable name="Operation-ID" value="@(context.Operation.Id)" />** |
| **15** | **set-variable** | **<set-variable name="geosite-backend" value="{{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-geosite-backend}}" />** |
| **16** | **choose** | **<choose>  <when condition="@(context.Operation.Id == "GET-GeoSiteHierarchy")">  <set-backend-service base-url="@(context.Variables.GetValueOrDefault<string>("geosite-backend"))" />  </when>  </choose>** |
| **17** | **set-header** | **<set-header name="Ocp-Apim-Subscription-Key" exists-action="delete" />** |
| **18** | **set-header** | **<set-header name="Ocp-Apim-Trace" exists-action="delete" />** |
| **19** | **forward-request** | **<forward-request />** |
| **20** | **base** | **<base />** |
| **21** | **choose** | **<choose>  <when condition="@(context.Response.Body.As<string>(preserveContent: true).Length > ({{InternalOperations-ServiceNow-ProdGeoHierarchy-V1-response-size}}))">  <return-response>  <set-status code="400" reason="Bad Request" />  <set-header name="ErrorMessage" exists-action="override">  <value>Message response too large.</value>  </set-header>  <set-header name="Content-Type" exists-action="override">  <value>application/json</value>  </set-header>  <set-body>{"statusCode": "400" ,"Message": "Message response too large"}</set-body>  </return-response>  </when>  <otherwise />  </choose>** |
| **22** | **set-header** | **<set-header name="Cache-Control" exists-action="override">  <value>no-cache</value>  </set-header>** |
| **23** | **set-header** | **<set-header name="Content-Security-Policy" exists-action="override">  <value>default-src 'self'</value>  </set-header>** |
| **24** | **set-header** | **<set-header name="Strict-Transport-Security" exists-action="override">  <value>max-age=31536000; includeSubDomains</value>  </set-header>** |
| **25** | **set-header** | **<set-header name="X-XSS-Protection" exists-action="override">  <value>1; mode=block</value>  </set-header>** |
| **26** | **set-header** | **<set-header name="X-Content-Type-Options" exists-action="override">  <value>nosniff</value>  </set-header>** |
| **27** | **set-header** | **<set-header name="X-Frame-Options" exists-action="override">  <value>deny</value>  </set-header>** |
| **28** | **base** | **<base />** |

# Deployment Steps

Fill in the missing details if needed. Otherwise, indicate N/A. If there are additional steps, please feel free to add them. **Ensure ALL required steps are indicated in this document.**

**1. Approvals**

**Before deploying production, the following approvals must be obtained:**

**APIM Team:**

**Apigee X:** procter-gamble/eai\_apigeex\_platform

**Azure:** procter-gamble/eai-azureapim-adminproduction

**Governance Team:** procter-gamble/cnf-gcpcl-its-eai-eng-gov

**2. Merge & Deploy**

**Once approved:**

* The pull request is merged into the main branch.
* CI/CD pipelines are triggered.
* Configuration is deployed to Azure APIM or Apigee X Production Environment.
* A ServiceNow DevOps change request is auto-created—check the Actions tab in GitHub for the request number.

**3. Post-Deployment**

* Begin testing your production API.
* Complete the Post Implementation Review using the provided template.
* The ServiceNow DevOps change will close automatically—no further action needed.

## Deploy the KVM/Name Values

[For the Description and Purpose of Change, please be as descriptive as possible]

|  |  |  |
| --- | --- | --- |
| # | **File Name** | **Kvm Entry** |
| **1** | **ratelimit-calls** | **1500** |
| **2** | **ratelimit-renewal-period** | **60** |
| **3** | **quota-calls** | **834** |
| **4** | **quota-renewal-period** | **300** |
| **5** | **request-size** | **200000** |
| **6** | **response-size** | **400000** |
| **7** | **allowed-ips** | **199.91.139.160,199.91.141.12,149.96.6.9,149.96.6.10,149.96.6.11,149.96.6.12,149.96.6.13,149.96.6.14,149.96.5.9,149.96.5.10,149.96.5.11,149.96.5.12,149.96.5.13,149.96.5.14** |
| **8** | **allowed-cidrs** | **199.91.136.8/29,199.91.140.8/29,149.96.5.8/29,149.96.6.8/29** |
| **9** | **geosite-backend** | **https://servicesboomiazureqart.pg.com:9093/ws/simple** |
| **10** | **username** | **DC7908** |
| **11** | **password** |  |

## Configure IP of Production for whitelisting:

|  |  |
| --- | --- |
| # | **IP** |
| **1** | **149.96.5.10** |
| **2** | **149.96.5.11** |
| **3** | **149.96.5.12** |
| **4** | **149.96.5.13** |
| **5** | **149.96.5.14** |
| **6** | **149.96.5.8/29** |
| **7** | **149.96.5.9** |
| **8** | **149.96.6.10** |
| **9** | **149.96.6.11** |
| **10** | **149.96.6.12** |
| **11** | **149.96.6.13** |
| **12** | **149.96.6.14** |
| **13** | **149.96.6.8/29** |
| **14** | **149.96.6.9** |
| **15** | **199.91.136.8/29** |
| **16** | **199.91.139.160** |
| **17** | **199.91.140.8/29** |
| **18** | **199.91.141.12** |

## API Products

|  |  |
| --- | --- |
| # | Name |
| 1 | **InternalOperations-ServiceNow-ProdGeoHierarchy-V1** |

# Backout Plan

**Apigee X**

1. Open the GCP Console and navigate to Apigee X.
2. Go to the specific API.
3. Check for an existing revision:
4. If no revision exists: Undeploy the API.
5. Re-deploy previous version using CI/CD:
   1. Trigger pipeline with the last known working version (rollback commit).

**Azure APIM**

1. Open Azure Portal and navigate to API Management (APIM).
2. Go to the API under the relevant Product/Service.
3. Determine the last stable configuration:
   1. Use GitHub commit history or a previous ARM template (if versioned).
4. Re-deploy previous version using CI/CD:
   1. Trigger pipeline with the last known working version (rollback commit).

**Post-Rollback Actions**

* Notify stakeholders of the rollback.
* Document of the incident (RCA).
* Re-deploy after issue resolution and ServiceNow DevOps change request is auto created.