Analysis of Google Cloud Build Authorization Failure for Identity Platform API

1. Introduction:
   A Google Cloud Build process has encountered an error while attempting to enable the Identity Platform API, as indicated by a "PERMISSION_DENIED" message within the build logs. This report aims to provide a comprehensive analysis of this error, identify the underlying missing authorization, and offer detailed steps for debugging and resolving the issue. The objective is to successfully enable the Identity Platform API within the specified Google Cloud project using the Cloud Build service.
2. **Detailed Analysis of the Permission Denied Error:**
   - Error Message Breakdown:
     The Cloud Build log clearly states the encountered problem with the error message: ERROR: (gcloud.services.enable) PERMISSION_DENIED: Permission denied to enable service [identityplatform.googleapis.com].1 This message indicates that the attempt to enable the Identity Platform API was rejected due to insufficient permissions. The log further specifies the identity under which the command was executed: This command is authenticated as cloud-build-sa@org-service-account-001.iam.gserviceaccount.com which is the active account specified by the [core/account] property.1 This confirms that the Cloud Build service account is the entity attempting to perform the enablement.
     The error details include structured information in the form of @type: type.googleapis.com/google.rpc.PreconditionFailure and type.googleapis.com/google.rpc.ErrorInfo.[1] These indicate that a precondition for the operation was not met, specifically related to authorization. The violation is detailed as: subject:?error_code=110002&service=servicemanagement.googleapis.com&permission=servicemanagement.services.bind&resource=soccer-dev-1744055311.[1] This pinpointed violation reveals that the missing authorization is the servicemanagement.services.bind permission for the identityplatform.googleapis.com service within the soccer-dev-1744055311 project. The reason: AUTH_PERMISSION_DENIED and the metadata further emphasize that the lack of the permission: servicemanagement.services.bind on the resource: soccer-dev-1744055311 for the service: servicemanagement.googleapis.com is the direct cause of the failure.[1] The Service Management API is responsible for managing Google Cloud services, and the services.bind permission is crucial for associating and enabling a

particular service within a project. Therefore, the Cloud Build service account is being denied the ability to enable the Identity Platform API because it lacks this specific permission.

   - Involved Service Account and Project:
     The service account attempting to enable the API is identified as cloud-build-sa@org-service-account-001.iam.gserviceaccount.com.1 The target Google Cloud project where this operation is being performed is soccer-dev-1744055311.1

3. **Understanding the Required IAM Permissions for Enabling Services:**
   - **serviceusage.services.enable Permission:** Generally, enabling a service on a Google Cloud project necessitates the serviceusage.services.enable permission at the project level.[2] This permission empowers an identity to alter the operational status of a service for a specific project. While the immediate error message highlights the absence of servicemanagement.services.bind, the serviceusage.services.enable permission serves as a fundamental prerequisite for any service enablement process. An identity might possess the permission to interact with the service management framework but still lack the necessary project-level authorization to activate the service within that project. Therefore, both permissions are typically required for successful service enablement.
   - **servicemanagement.services.bind Permission (Crucial for this error):** Enabling a particular service also mandates the servicemanagement.services.bind permission specifically for that service.[2] This permission grants the ability to "view and enable the service on projects the caller controls".[4] The error message explicitly points to the absence of this permission [1], indicating that the Cloud Build service account is failing at this specific authorization check for the Identity Platform API within the soccer-dev-1744055311 project. The Service Management API governs the management of Google Cloud services, and this binding permission is essential for associating and activating a service within a given project context.
4. **Identifying IAM Roles with Necessary Permissions:**
   - **Analysis of Existing Roles Assigned to the Cloud Build Service Account:** The Cloud Build service account, cloud-build-sa@org-service-account-001.iam.gserviceaccount.com, currently holds the following IAM roles: roles/editor, roles/firebase.admin, roles/firebase.developAdmin, roles/iam.serviceAccountTokenCreator, roles/identityplatform.admin, roles/owner, roles/servicemanagement.admin, roles/serviceusage.apiKeysAdmin, and

roles/serviceusage.serviceUsageAdmin.[1] This collection of roles includes several with broad permissions, such as roles/owner and roles/servicemanagement.admin. The presence of roles/identityplatform.admin also suggests an intent to manage the Identity Platform service. The fact that the "PERMISSION_DENIED" error persists despite these roles indicates a potential issue beyond simply lacking a basic permission. This could involve how these roles are being interpreted in the Cloud Build environment, the presence of an organizational policy that overrides these permissions, or a specific requirement for service enablement that isn't fully met by these roles in this context.

- **Determining if Listed Roles Include servicemanagement.services.bind and serviceusage.services.enable:** Research indicates that the roles/editor and roles/owner roles include the serviceusage.services.enable permission.[2] Furthermore, these roles also inherently grant the servicemanagement.services.bind permission.[4] The roles/serviceusage.serviceUsageAdmin role also includes the serviceusage.services.enable permission [2] and grants the ability to enable and disable services.[2] Snippet [6] also shows that roles/servicemanagement.admin includes the servicemanagement.services.bind permission. Given that the Cloud Build service account already possesses roles/owner, roles/editor, and roles/servicemanagement.admin, it should theoretically have both the servicemanagement.services.bind and serviceusage.services.enable permissions. This contradiction with the "PERMISSION_DENIED" error suggests that the underlying cause might be a higher-level restriction or a temporary issue within the Google Cloud infrastructure affecting the application of these roles in this specific scenario.

Role Name	**Includes serviceusage.services.enable?**	**Includes servicemanagement.services.bind?**
roles/owner	Yes [2]	Yes [4]
roles/editor	Yes [2]	Yes [4]
roles/serviceusage.serviceUsageAdmin	Yes [2]	No (based on direct evidence)

roles/servicemanagement.admin	No	Yes [6]

*   **Other Predefined Roles Granting These Permissions:**
    The `roles/serviceusage.serviceUsageAdmin` role is explicitly designed for managing service usage and includes the `serviceusage.services.enable` permission [2, 5], granting the ability to enable and disable services.[2, 3] While the Cloud Build service account has this role, it might not be sufficient on its own for the `servicemanagement.services.bind` permission. Snippet [22] indicates that the `Service Config Editor` role might be necessary in some cases for API enablement, although this is less likely given the presence of `roles/owner` and `roles/servicemanagement.admin`. The `roles/serviceusage.serviceUsageConsumer` role grants the `serviceusage.services.use` permission, which is related to using services but not enabling them.[2] The `roles/servicemanagement.serviceConsumer` role is mentioned in research but is more focused on consuming services rather than enabling them.[4]

5. **Debugging and Resolving the Missing Authorization:**
    - **Granting the roles/serviceusage.serviceUsageAdmin Role (as a potential explicit fix):**
        - **Using Google Cloud Console:** Navigate to the IAM & Admin section in the Google Cloud Console, then select IAM. Locate the Cloud Build service account (cloud-build-sa@org-service-account-001.iam.gserviceaccount.com) in the list of principals. Click on the "Edit principal" icon (pencil icon) next to it. In the edit pane, click the "Add another role" button. Search for "Service Usage Admin" in the role selection dropdown and choose it. Finally, click the "Save" button to apply the changes.
        - **Using gcloud Command:** Open the Google Cloud CLI and execute the following command, replacing soccer-dev-1744055311 with your project ID if it's different: gcloud projects add-iam-policy-binding soccer-dev-1744055311 --member='serviceAccount:cloud-build-sa@org-service-account-001.iam.gserviceaccount.com' --role='roles/serviceusage.serviceUsageAdmin' This command explicitly grants the

roles/serviceusage.serviceUsageAdmin role to the Cloud Build service account on the specified project.

- Investigating Organizational Policies (A likely culprit given existing roles): Organizational policies can supersede IAM permissions and prevent specific actions, including the enablement of certain services.7 It is possible that an organizational policy is configured at the organization, folder, or project level that is preventing the Cloud Build service account from enabling the Identity Platform API, despite the seemingly sufficient IAM roles it possesses. A constraint such as constraints/serviceusage.allowedServices might be in place, which restricts the services that can be enabled within the project or organization.11
  To check for organizational policies:
  - **Using Google Cloud Console:** Navigate to the IAM & Admin section and select "Organization policies".[12] Ensure that the correct project, folder, or organization is selected in the project picker. Filter the list of policies by searching for "serviceusage" or "identityplatform" to find relevant policies. Review the enforcement status and configuration of any policies that appear relevant. Look for any policies that might explicitly deny the enablement of new services or specifically the Identity Platform API.
  - **Using gcloud CLI:** To list the organizational policies applied to the soccer-dev-1744055311 project, execute the command: gcloud resource-manager org-policies list --project=soccer-dev-1744055311.[10] To get detailed information about a specific policy, use its name in the following command: gcloud resource-manager org-policies describe POLICY_NAME --project=soccer-dev-1744055311.[10] It is also crucial to check for policies applied at the folder and organization levels if the project is within a folder or organization. You can do this by using the --folder or --organization flags instead of --project in the above commands, providing the respective folder or organization ID.
- Verifying Cloud Build Service Account Configuration:
  Ensure that the Cloud Build service account (cloud-build-sa@org-service-account-001.iam.gserviceaccount.com) exists within the soccer-dev-1744055311 project. Verify that the service account has not been accidentally disabled or deleted.14 Although the build log indicates that Cloud Build is running, it's worth confirming that the Cloud Build API is enabled for the soccer-dev-1744055311 project 15 by navigating to the "APIs & Services" > "Dashboard" section in the Google Cloud Console and checking if the Cloud Build API is listed as enabled.
- **Considering Alternative Enablement Methods:**

- **Manual Enablement via Google Cloud Console:** Navigate to the "APIs & Services" section in the Google Cloud Console and select "Library". Search for "Identity Platform API" and select it. If the API is not already enabled, click the "Enable" button. This action requires a user with sufficient permissions on the project, such as Project Owner or a user with both serviceusage.services.enable and servicemanagement.services.bind permissions. If manual enablement by a user with broad permissions also fails with a similar error, it strongly suggests an organizational policy is in effect.
- **Enabling via Terraform (if applicable):** If the project's infrastructure is managed using Terraform, the Identity Platform API can be enabled by adding a google_project_service resource to the Terraform configuration.[17]

6. Best Practices for Service Account Permissions in Cloud Build:
   It is crucial to adhere to the principle of least privilege when granting permissions to service accounts in Cloud Build.18 Avoid assigning overly broad roles like Owner if more specific roles can fulfill the necessary requirements. Consider utilizing dedicated service accounts for different Cloud Build jobs or stages to further limit the potential impact of compromised credentials. Regularly review and audit the permissions assigned to Cloud Build service accounts to ensure they remain appropriate and necessary as the project evolves. Leverage the predefined Cloud Build IAM roles, such as roles/cloudbuild.builds.builder and roles/cloudbuild.builds.editor 19, for managing build-related permissions, while carefully considering the specific permissions needed for actions like enabling services. Be aware of the security implications of granting broad permissions to service accounts, as highlighted in research regarding potential privilege escalation.21
7. Conclusion:
   The Cloud Build process encountered a "PERMISSION_DENIED" error while attempting to enable the Identity Platform API, specifically indicating a missing servicemanagement.services.bind permission. However, the Cloud Build service account in question already holds roles, including roles/owner, roles/editor, and roles/servicemanagement.admin, which should inherently grant this permission. This discrepancy strongly suggests that an organizational policy is likely the primary factor preventing the service account from enabling the Identity Platform API.
   The recommended course of action is to first try explicitly granting the roles/serviceusage.serviceUsageAdmin role to the Cloud Build service account as a precautionary measure. The next critical step is to thoroughly investigate

organizational policies applied at the project, folder, and organization levels, specifically looking for any policies that might be restricting the enablement of the Identity Platform API or services in general. As a means of verification and a potential temporary solution, attempting to enable the Identity Platform API manually via the Google Cloud Console by a user with Project Owner or similar broad permissions is advisable. If this manual attempt also fails, it would further solidify the likelihood of an organizational policy restriction. Adhering to best practices for managing service account permissions in Cloud Build is essential for maintaining a secure and efficient development pipeline.

Works cited

1. Re: How to accurately determine which Services/API need to be enabled given a list of permissions? - Google Cloud Community, accessed on April 8, 2025, https://www.googlecloudcommunity.com/gc/Developer-Tools/How-to-accurately-determine-which-Services-API-need-to-be/m-p/177928
2. Access Control with IAM | Service Usage Documentation - Google Cloud, accessed on April 8, 2025, https://cloud.google.com/service-usage/docs/access-control
3. Solved: Im facing issue while working with AML AI model,an... - Google Cloud Community, accessed on April 8, 2025, https://www.googlecloudcommunity.com/gc/Gemini-Code-Assist/Im-facing-issue-while-working-with-AML-AI-model-any-ideas-on-how/m-p/863289
4. Service Management API Access Control - Google Cloud, accessed on April 8, 2025, https://cloud.google.com/service-infrastructure/docs/service-management/access-control
5. Service Usage - Permissions Reference for Google Cloud IAM, accessed on April 8, 2025, https://gcp.permissions.cloud/iam/serviceusage
6. Service Management - Permissions Reference for Google Cloud IAM, accessed on April 8, 2025, https://gcp.permissions.cloud/iam/servicemanagement
7. Introduction to the Organization Policy Service | Resource Manager Documentation, accessed on April 8, 2025, https://cloud.google.com/resource-manager/docs/organization-policy/overview
8. A Guide to GCP Organization Policy: Managing Access - Sonrai Security, accessed on April 8, 2025, https://sonraisecurity.com/blog/a-guide-to-gcp-organization-policy-managing-access/
9. Comprehensive Guide to Organization Policies in Google Cloud - Medium, accessed on April 8, 2025, https://medium.com/google-cloud/comprehensive-guide-to-organization-policies-in-google-cloud-a81ff9e1c9eb
10. List Organization Policies with gcloud: A Quick and Easy Guide | by Anita Gutta | Google Cloud - Medium, accessed on April 8, 2025,

https://medium.com/google-cloud/list-organization-policies-with-gcloud-a-quick-and-easy-guide-edcedafe9806

11. Organization policy constraints | Resource Manager Documentation - Google Cloud, accessed on April 8, 2025, https://cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints
12. cloud.google.com, accessed on April 8, 2025, https://cloud.google.com/resource-manager/docs/organization-policy/creating-managing-policies#:~:text=In%20the%20Google%20Cloud%20console%2C%20go%20to%20the%20Organization%20policies%20page.&text=From%20the%20project%20picker%2C%20select,are%20available%20for%20this%20resource.
13. Creating and managing organization policies - Google Cloud, accessed on April 8, 2025, https://cloud.google.com/resource-manager/docs/organization-policy/creating-managing-policies
14. Service accounts permissions - google cloud platform - Stack Overflow, accessed on April 8, 2025, https://stackoverflow.com/questions/55725726/service-accounts-permissions
15. Cloud Build: Trigger uses default service account - Google Cloud Community, accessed on April 8, 2025, https://www.googlecloudcommunity.com/gc/Developer-Tools/Cloud-Build-Trigger-uses-default-service-account-when-user/td-p/515992
16. Cloud Build Error - Permission 'cloudbuild.builds.create' denied - Google Cloud Community, accessed on April 8, 2025, https://www.googlecloudcommunity.com/gc/Developer-Tools/Cloud-Build-Error-Permission-cloudbuild-builds-create-denied/m-p/729684
17. How to enable Identity and Access Management (IAM) API programmatically for a Google Cloud Project? - Stack Overflow, accessed on April 8, 2025, https://stackoverflow.com/questions/63358802/how-to-enable-identity-and-access-management-iam-api-programmatically-for-a-go
18. IAM basic and predefined roles reference | IAM Documentation - Google Cloud, accessed on April 8, 2025, https://cloud.google.com/iam/docs/understanding-roles
19. IAM roles and permissions | Cloud Build Documentation, accessed on April 8, 2025, https://cloud.google.com/build/docs/iam-roles-permissions
20. Cloud Build - Permissions Reference for Google Cloud IAM | gcp.permissions.cloud, accessed on April 8, 2025, https://gcp.permissions.cloud/iam/cloudbuild
21. Working-As-Intended: RCE to IAM Privilege Escalation in GCP Cloud Build, accessed on April 8, 2025, https://rhinosecuritylabs.com/gcp/iam-privilege-escalation-gcp-cloudbuild/
22. GCP: ERROR: (gcloud.services.enable) PERMISSION_DENIED: The caller does not have permission when enabling API - Stack Overflow, accessed on April 8, 2025, https://stackoverflow.com/questions/63787883/gcp-error-gcloud-services-enable-permission-denied-the-caller-does-not-have