

Lab 5.2

Manual Fixes and Establishing a Governance Baseline

Toepasbare Cloud Security voor IT-Professionals

Contents

1. Manual Fixes and Establishing a Governance Baseline	3
1.1. Context	3
1.2. Learning goals	3
1.3. Estimated time	3
1.4. Before you start	3
2. Review and Add Required Tags	5
2.1. Inspect current tags	5
2.2. Add or update missing tags	5
3. Review Storage Account Exposure	7
3.1. Review the frontend storage account	7
3.2. Review the private or data storage account	7
3.3. Optional: disable public network access on the private data storage account	8
4. Enable Diagnostic Settings on Key Vault	9
4.1. Navigate to Key Vault diagnostic settings	9
4.2. Configure diagnostics	9
4.3. Verify diagnostic settings	9
5. Verify HTTPS Enforcement on App Services	11
6. Document the Governance Baseline	12
7. Manual Fixes and Terraform Drift	13
8. Reflection Questions	14
9. Summary	15

1. Manual Fixes and Establishing a Governance Baseline

1.1. Context

In Lab 5.1, you inspected the Day 5 deployment and identified governance and security gaps.

In this lab, you will fix selected findings manually. You will add missing tags, review storage exposure, enable Key Vault diagnostic settings, and verify HTTPS enforcement on the App Services.

The goal is not to make manual work the long-term solution. The goal is to understand what each setting means before enforcing it with Azure Policy in Lab 5.3.

Note

Manual fixes are useful for learning and for small one-off corrections. They do not scale well. In the next labs, you will use Azure Policy and remediation tasks to make these requirements more consistent.

1.2. Learning goals

By the end of this lab, you should be able to explain:

- What a governance baseline is.
- How tags support ownership, cost control, and operations.
- The difference between anonymous blob access and public network access on storage accounts.
- Why disabling public network access can break workloads if private connectivity is not ready.
- Why Key Vault diagnostic settings are important for security monitoring.
- Why manual fixes should eventually be replaced or supported by policy and infrastructure-as-code.

1.3. Estimated time

45 minutes

Suggested timing:

Time	Activity
10 minutes	Review and add required tags
10 minutes	Review storage account exposure
10 minutes	Enable Key Vault diagnostic settings
5 minutes	Verify HTTPS enforcement on App Services
5 minutes	Document the governance baseline
5 minutes	Reflection and discussion

1.4. Before you start

Make sure you have:

- ☐ Lab 5.1 compliance audit completed.
- ☐ Contributor access on your lab resource group.
- ☐ Access to the Azure portal.
- ☐ Access to the Log Analytics workspace.
- ☐ The names of your Day 5 resource group, storage accounts, Key Vault, and App Services.

Note

Some changes in this lab may affect how the demo application works. Only apply the network-related changes to resources that your instructor has identified as safe to modify.

2. Review and Add Required Tags

Tags are metadata that help connect cloud resources to ownership, cost, lifecycle, and governance processes.

Your Day 5 resources should have a consistent minimum tag set.

Recommended baseline tags:

Tag	Example value	Purpose
environment	dev	Shows which environment the resource belongs to.
student	<your_prefix>	Identifies the student or group that owns the lab resources.
managed_by	terraform	Shows that the resource is expected to be managed by infrastructure-as-code.
day	day-5	Connects the resource to the course day.

Note

Azure supports tags on resources, resource groups, and subscriptions, but not all resource types or child resources behave the same way. Some child resources may not show the same tags as the parent resource.

2.1. Inspect current tags

In the Azure portal:

1. Open your Day 5 resource group.
2. Open the **Tags** blade.
3. Write down the tags on the resource group.
4. Open several resources and check whether they have the same tag set.

You can also use Azure CLI:

```
az resource list `
  --resource-group <RESOURCE_GROUP_NAME> `
  --query "[].{Name:name, Type:type, Tags:tags}" `
  --output table
```

Note

The examples use PowerShell-style line continuation with a backtick. If you use Bash, replace the backtick with a backslash.

2.2. Add or update missing tags

In the portal:

1. Open a resource that is missing required tags.
2. Open **Tags**.
3. Add the required tags.
4. Save the changes.

Optional CLI approach:

```
az tag update `
  --resource-id <RESOURCE_ID> `
  --operation Merge `
  --tags environment=dev student=<your_prefix> managed_by=terraform day=day-5
```

Note

Use `Merge` when adding tags with Azure CLI. Replacing all tags can accidentally remove tags that were added by Terraform, Azure Policy, or another process.

Question

What could go wrong if a manually changed tag conflicts with what Terraform expects?

3. Review Storage Account Exposure

Storage exposure has multiple layers. Do not treat all “public” settings as the same thing.

Setting	Meaning
Anonymous blob access	Controls whether blob data can be made readable without authentication.
Container anonymous access level	Controls whether a specific container allows anonymous read access, if account-level anonymous access permits it.
Public network access	Controls whether the storage account public endpoint can be reached from public networks.
Firewall and virtual network rules	Restrict which public IP ranges, virtual networks, or trusted Azure services may reach the public endpoint.
Private endpoint	Provides private connectivity to the storage account through a virtual network.

Note

A storage account can have public network access enabled without allowing anonymous blob reads. It can also disallow anonymous blob access while still being reachable over the public endpoint for authenticated clients.

3.1. Review the frontend storage account

The frontend storage account may be used for static website hosting.

1. Open the frontend storage account.
2. Review **Data management** or **Configuration**.
3. Check whether **Allow Blob anonymous access** is enabled or disabled.
4. Review **Static website** configuration if this account hosts the frontend.
5. Review **Networking** and note the public network access setting.

Note

Static website hosting uses the storage static website endpoint. Do not blindly disable settings on the frontend storage account unless you know how the frontend is being served and whether the lab still needs public access.

Question

Why might the frontend storage account need different exposure settings than a storage account used for private application data?

3.2. Review the private or data storage account

If your Day 5 deployment includes a storage account intended for private application data, review its exposure more strictly.

1. Open the storage account.

2. Go to **Networking**.
3. Review **Public network access**.
4. Review firewall rules and private endpoint configuration.
5. Check whether the backend application has a valid network path to the storage account.

Note

Disabling public network access is safer only when the required private connectivity is already in place. A managed identity solves authentication and authorization, but it does not by itself solve network reachability.

3.3. Optional: disable public network access on the private data storage account

Only perform this step if your instructor confirms that the application is ready for private connectivity.

1. Open the private data storage account.
2. Go to **Networking**.
3. Set **Public network access** to **Disabled**.
4. Save the change.
5. Test the application path that depends on this storage account.

Question

If the backend uses a managed identity but the storage account blocks public network access, what else must be true for the backend to access the storage account?

4. Enable Diagnostic Settings on Key Vault

Key Vault contains security-sensitive operations. You need audit logs to investigate who accessed secrets, keys, or certificates.

4.1. Navigate to Key Vault diagnostic settings

1. Open your Key Vault.
2. Go to **Monitoring**.
3. Open **Diagnostic settings**.
4. Click **Add diagnostic setting**.

4.2. Configure diagnostics

Use a clear name:

diag-kv-audit

Select log categories such as:

- AuditEvent or the available Key Vault audit category shown in your tenant.

Select metrics if available and relevant:

- AllMetrics

Destination:

- Send to Log Analytics workspace.
- Select the Day 5 Log Analytics workspace.

Save the diagnostic setting.

Note

Key Vault audit logs help answer questions such as who accessed the vault, which operation was performed, whether it succeeded, and from where. In Log Analytics, these records commonly appear in `AZKVAuditLogs` when resource-specific tables are used.

4.3. Verify diagnostic settings

After saving:

1. Confirm the diagnostic setting appears in the Key Vault.
2. Wait a few minutes.
3. Trigger a simple Key Vault operation if your instructor asks you to.
4. Query Log Analytics later to confirm that data arrives.

Example query for later use:

```
AZKVAuditLogs
| where TimeGenerated > ago(1h)
| project TimeGenerated, OperationName, ResultType, CallerIpAddress, Identity,
ResourceId
```

```
| sort by TimeGenerated desc  
| take 20
```

Question

Why is Log Analytics more useful for investigation than only checking settings in the Key Vault portal?

5. Verify HTTPS Enforcement on App Services

The App Services should enforce HTTPS.

1. Open the public backend App Service.
2. Go to **Settings** or **Configuration**.
3. Verify that **HTTPS Only** is enabled.
4. Repeat for the internal backend App Service.

Optional CLI check:

```
az webapp show `
  --resource-group <RESOURCE_GROUP_NAME> `
  --name <APP_SERVICE_NAME> `
  --query "{name:name, httpsOnly:httpsOnly}" `
  --output table
```

Note

If Terraform sets `https\_only = true`, the portal and CLI should show HTTPS-only as enabled. If a manual change disables it, Terraform may try to correct the drift on the next apply.

6. Document the Governance Baseline

Complete the baseline table.

Requirement	Current state	Required state
Required tags on resource group and main resources		environment, student, managed_by, day present
Frontend storage anonymous blob access reviewed		Known and documented
Private data storage public network access reviewed		Disabled only if private connectivity is ready
Storage accounts deny anonymous blob access unless explicitly required		Disabled by default
Key Vault diagnostic settings		Audit logs sent to Log Analytics
App Services HTTPS only		On
Terraform drift risk		Manual changes documented before next apply

Note

A baseline should be specific enough to verify, but flexible enough to avoid breaking valid workloads. For example, frontend hosting and private data storage should not always have the same network exposure requirements.

Question

Which baseline requirements are safe to enforce automatically? Which ones require more context before enforcing?

7. Manual Fixes and Terraform Drift

Manual changes can create drift from your Terraform configuration.

Drift means:

“The real Azure resource no longer matches what the infrastructure-as-code configuration describes.”

This matters because a future Terraform apply may:

revert your manual change, detect unexpected differences, **fail because of a conflicting setting**, or hide the fact that the source code is no longer the source of truth.

Note

Manual fixes are useful in a lab because they make the setting visible. In real environments, long-term fixes should usually be captured in Terraform, Bicep, ARM, or Azure Policy.

Question

After completing a manual fix, where should the permanent fix live: only in the portal, in Terraform, in Azure Policy, or in more than one place?

8. Reflection Questions

Your answer:

1. Which manual fix was easiest to understand?
2. Which manual fix had the highest risk of breaking the application?
3. Why is it important to understand a setting manually before enforcing it with Azure Policy?
4. What problems would manual fixes create if you had 50 or 500 resources?
5. Which findings from this lab should become Terraform changes?
6. Which findings from this lab should become Azure Policy guardrails?

9. Summary

In this lab, you manually improved the Day 5 governance baseline.

You reviewed and updated tags, inspected storage exposure, enabled Key Vault diagnostic settings, verified HTTPS enforcement on App Services, and documented which requirements should become policy or infrastructure-as-code.

The key lesson is:

“Manual fixes help you understand the control. Governance baselines help you standardize it. Policy and infrastructure-as-code help you keep it consistent.”

Next, you will use Azure Policy to enforce selected guardrails automatically.