

# Szkolenie

# Terraform: Dzień 4

---



## 1. Zadanie: Wysyłanie metryk z App Service oraz Key Vault do Log Analytics Workspace

Należy wykorzystać Key Vault i App Service z poprzednich zajęć (Terraform: Dzień 3) oraz stworzyć zasób Log Analytics Workspace. Następnie włączyć wysyłanie wszystkich metryk z App Serwis oraz Key Vault do Log Analytics Workspace z wykorzystaniem Diagnostic Settings.

Pamiętaj, że Diagnostic Settings muszą być konfigurowane w momencie istnienia zasobów, do których się odwołują. W tym przypadku jest to np. Key Vault, z którego chcemy wysyłać dane oraz Log Analytics Workspace do którego te dane chcemy wysyłać.

Zauważ, że Diagnostic Settings możemy wykorzystać do wysyłania zarówno metryk jak i logów. Różne zasoby mają różne kategorie logów i metryk, którymi jesteśmy w stanie zarządzać. Dostępne kategorie można znaleźć m.in. w dokumentacji Microsoft. W tym zadaniu potrzebujemy wskazać jedyną dostępną dla obu zasobów kategorię metryk: **AllMetrics**. Pamiętaj o wskazaniu id odpowiednich zasobów.

Zasoby które należy utworzyć (w kolejności):

- Log Analytics Workspace
- Diagnostic settings: Key Vault do Log Analytics workspace
- Diagnostic settings: App Service do Log Analytics workspace

Przydatne linki:

Terraform dokumentacja Log Analytics Workspace

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/log\\_analytics\\_workspace](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/log_analytics_workspace)

Terraform dokumentacja Diagnostic Settings

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/monitor\\_diagnostic\\_setting](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/monitor_diagnostic_setting)

Dostępne kategorie logów zasobów w Azure

<https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/resource-logs-categories>

Metryki wspierane przez Azure Monitor

<https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported>

## 2. Zadanie: Utworzenie Application Insights powiązane z Log Analytics Workspace i połączenie do aplikacji webowej

Należy utworzyć Application Insights oparte o Log Analytics Workspace, a następnie połączyć App Service z nowo utworzonym Application Insights.

Powołanie zasobu Application Insights powinno być stosunkowo proste i wymaga jedynie podania Log Analytics Workspace id pod który chcemy podpiąć usługę.

W przypadku podpięcia Application Insights do aplikacji webowej, należy zmodyfikować blok *app\_settings* dla App Service. Czynność ta wymaga podania wartości zmapowanych w następujący sposób:

```
"APPINSIGHTS_INSTRUMENTATIONKEY" : "${azurerm_application_insights.appinsights01.instrumentation_key}"
"APPLICATIONINSIGHTS_CONNECTION_STRING" : "${azurerm_application_insights.appinsights01.connection_string}"
"ApplicationInsightsAgent_EXTENSION_VERSION" : "~3"
"XDT_MicrosoftApplicationInsights_Mode" : "Recommended"
```

Zasoby które należy utworzyć (w kolejności):

- Application Insights
- Zmodyfikować blok *app\_settings* dla App Service

Przydatne linki:

Terraform dokumentacja Application Insights

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/application\\_insights](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/application_insights)

Terraform dokumentacja App Service

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux\\_web\\_app](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux_web_app)

## 3. Zadanie: Powołanie Azure Policy

Należy powołać definicje dwóch polityk, a następnie przypisać je do własnej Resource Group'y. W końcu należy przetestować działanie polityk np. poprzez utworzenie testowego zasobu.

Jedną politykę przygotuj tylko z wykorzystaniem kodu terraform, w drugiej polityce wykorzystaj plik .json przy pomocy funkcji file(), jsondecode(), jsonencode().

Należy stworzyć 2 polityki:

- Polityka z efektem *Deny*, która będzie wymuszać tworzenie zasobów w sprecyzowanych przez nas lokalizacjach np. *westeurope* i *northeurope*

[https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/General/AllowedLocations\\_Deny.json](https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/General/AllowedLocations_Deny.json)

- Polityka z efektem *Deny*, która będzie zezwalać na utworzenie dozwolonych rozmiarów Maszyn Wirtualnych

[https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/Compute/VMSkusAllowed\\_Deny.json](https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/Compute/VMSkusAllowed_Deny.json)

Zasoby które należy utworzyć (w kolejności):

- Policy definition
- Policy assignment
- Resource testowy

Przydatne linki:

Terraform dokumentacja Policy Definition

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/policy\\_definition](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/policy_definition)

Terraform dokumentacja Resource Group Policy Assignment

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/resource\\_group\\_policy\\_assignment](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/resource_group_policy_assignment)

Terraform dokumentacja funkcji file()

<https://www.terraform.io/language/functions/file>

Terraform dokumentacja funkcji jsonencode()

<https://www.terraform.io/language/functions/jsonencode>

Terraform dokumentacja funkcji jsondecode()

<https://www.terraform.io/language/functions/jsondecode>