Obliczenia: ARM i Bicep, maszyny wirtualne, dostępność, App Service i kontenery

Poziom: średnio

Dotyczy: AZ-104, domena „Deploy and manage Azure compute resources” (20 do 25%), szablony ARM i Bicep, maszyny wirtualne, zestawy skalowania, App Service, Azure Container Registry, Container Instances, Container Apps, Azure CLI 2.91, moduł Az

Czas: 6 dni po 1,5 do 2 godzin

Dni 12 do 17 programu: czytanie i zmiana szablonów ARM i plików Bicep, maszyny wirtualne z dyskami i szyfrowaniem na hoście, strefy, zestawy dostępności i zestawy skalowania, App Service ze slotami i domeną oraz kontenery w ACR, ACI i Container Apps. To najszerszy zakres egzaminu, a w laboratorium najdroższy, więc tu najbardziej pilnujemy sprzątania.

ZAŁOŻENIA SCENARIUSZA

maszyny stoją w grupie `rg-lab-obliczenia`, a sieć z pliku `lab-siec.bicep` w `rg-lab-siec`, więc podsieć podajemy pełnym identyfikatorem. Maszyny WWW mają adresy 10.32.16.4 i 10.32.16.5 w `snet-web`, baza 10.32.32.4 w `snet-dane`. Obraz w przykładach to alias `Ubuntu2204` z pomocy Azure CLI; produkcyjne maszyny Zielonej chodzą na Rocky Linux 9, którego obraz z Marketplace wymaga akceptacji warunków wydawcy i nazwy w formacie URN (sprawdzamy `az vm image list --publisher NazwaWydawcy --all`).

Skróty użyte na tej stronie (25)

Skrót	Co znaczy
ACI	Azure Container Instances: usługa uruchamiająca pojedyncze grupy kontenerów w Azure bez zarządzania serwerem ani klastrem

Skrót	Co znaczy
ACR	Azure Container Registry: prywatny rejestr obrazów kontenerów w Azure, z którego pobierają obrazy ACI, Container Apps, App Service i AKS
AKS	Azure Kubernetes Service: Kubernetes w chmurze Microsoftu, w którym płaszczyznę sterowania utrzymuje dostawca
ARM	Azure Resource Manager: warstwa zarządzania Azure, przez którą przechodzi każde utworzenie i zmiana zasobu; także szablony ARM, czyli pliki JSON opisujące zasoby
AZ-104	numer egzaminu Microsoft Azure Administrator: egzamin, którego zdanie daje certyfikat Microsoft Certified: Azure Administrator Associate, ważny rok i odnawiany bezpłatną oceną online
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
CNAME	Canonical Name: rekord DNS będący aliasem, drugą nazwą wskazującą na już istniejący rekord A
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
GRS	Geo-Redundant Storage: nadmiarowość w Azure, która dodatkowo kopiuje dane do drugiego, odległego regionu
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
JSON	JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami
LRS	Locally Redundant Storage: nadmiarowość w Azure, która trzyma trzy kopie danych w jednym centrum danych
LVM	Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny
RA-GRS	Read-Access Geo-Redundant Storage: nadmiarowość GRS z możliwością odczytu danych z regionu dodatkowego
RA-GZRS	Read-Access Geo-Zone-Redundant Storage: nadmiarowość GZRS z możliwością odczytu danych z regionu dodatkowego
SAS	Serial Attached SCSI: złącze i magistrala dyskowa w serwerach, następca równoległego SCSI, pozwala podłączyć wiele dysków i macierzy przez jeden kontroler z większą przepustowością niż SATA; w Azure także Shared Access Signature: podpisany adres z tokenem, który daje czasowy dostęp do danych w koncie magazynu
SNI	Server Name Indication: pole w pierwszym komunikacie TLS, w którym klient podaje nazwę serwera, z którym chce rozmawiać, co pozwala HAProxy wybrać backend bez rozszyfrowywania ruchu
SSL	Secure Sockets Layer: dawna nazwa szyfrowania, dziś formalnie TLS
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS
TXT	typ rekordu DNS przechowującego dowolny tekst, wykorzystywany między innymi do publikowania reguł SPF, DKIM i DMARC

Skrót	Co znaczy
URN	Uniform Resource Name: tu pełna nazwa obrazu maszyny w Azure w formacie Wydawca:Oferta:SKU:Wersja
VAT	Podatek od towarów i usług: polski podatek, którego ewidencje i faktury przedsiębiorca przechowuje do upływu przedawnienia zobowiązania podatkowego
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS
ZRS	Zone-Redundant Storage: nadmiarowość w Azure, która rozkłada kopie danych na kilka stref dostępności w jednym regionie

Fiszka

Komendy wykonujemy po `az account set --subscription zielona-lab`. Rozmiar maszyny (`NazwaRozmiaru`) wybieramy z `az vm list-sizes --location polandcentral`, bo dostępność rozmiarów zależy od regionu i limitów subskrypcji. Nazwy aplikacji App Service i rejestrów kontenerów są unikatowe w całym Azure. Parametry sprawdziliśmy w pomocy Azure CLI 2.91.0 i w dokumentacji modułu Az (19 października 2026).

□□ ZMIENIA SYSTEM

Eksport grupy zasobów do pliku Bicep: az group export

```
az group export --name rg-lab-siec --export-format bicep > rg-lab-siec.bicep
```

Format `json` (domyślny) daje szablon ARM, `bicep` od razu plik Bicep. Eksport jest punktem wyjścia, a nie gotowym plikiem: zwykle trzeba usunąć zbędne właściwości i dopisać parametry. W PowerShell: `Export-AzResourceGroup -ResourceGroupName rg-lab-siec -Path ./rg-lab-siec.json`.

□□ **ARM** = Azure Resource Manager: warstwa zarządzania Azure, przez którą przechodzi każde utworzenie i zmiana zasobu; także szablony ARM, czyli pliki JSON opisujące zasoby ·**az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Zamiana szablonu ARM na Bicep: az bicep decompile

```
az bicep decompile --file szablon.json
```

Tworzy obok plik `szablon.bicep`. Dekompilacja nie zawsze jest doskonała, więc wynik przeglądamy i sprawdzamy `az bicep build`.

ARM = Azure Resource Manager: warstwa zarządzania Azure, przez którą przechodzi każde utworzenie i zmiana zasobu; także szablony ARM, czyli pliki JSON opisujące zasoby ·**az** = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Kompilacja pliku Bicep do szablonu ARM: az bicep build

```
az bicep build --file konto-magazynu.bicep --stdout
```

Pokazuje szablon ARM w JSON, który powstaje z pliku Bicep. Tak czytamy, co naprawdę wyśle wdrożenie.

ARM = Azure Resource Manager: warstwa zarządzania Azure, przez którą przechodzi każde utworzenie i zmiana zasobu; także szablony ARM, czyli pliki JSON opisujące zasoby ·**JSON** = JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami ·**az** = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Plik konta magazynu do ćwiczeń z parametrami: konto-magazynu.bicep

```
@minLength(3)
@maxLength(24)
@description('Nazwa konta: male litery i cyfry, unikatowa w calym Azure')
param nazwaKonta string

param lokalizacja string = resourceGroup().location

@allowed([
  'Standard_LRS'
  'Standard_ZRS'
])
param sku string = 'Standard_LRS'

resource konto 'Microsoft.Storage/storageAccounts@2023-05-01' = {
  name: nazwaKonta
```

```
  location: lokalizacja
  sku: {
    name: sku
  }
  kind: 'StorageV2'
  properties: {
    accessTier: 'Hot'
    minimumTlsVersion: 'TLS1_2'
    allowBlobPublicAccess: false
    supportsHttpsTrafficOnly: true
  }
}

output punktKoncowyBlob string = konto.properties.primaryEndpoints.blob
```

Plik przechodzi `bicep build` i `bicep lint` w wersji 0.38.33. Odpowiadający mu szablon ARM pokazujemy w dniu 12.

ARM = Azure Resource Manager: warstwa zarządzania Azure, przez którą przechodzi każde utworzenie i zmiana zasobu; także szablony ARM, czyli pliki JSON opisujące zasoby

ZMIENIA SYSTEM

Wdrożenie pliku z parametrem z linii poleceń: az deployment group create --parameters

```
az deployment group create --resource-group rg-lab-magazyn --template-file konto-magazynu.bicep \
  --parameters nazwaKonta=NazwaKonta sku=Standard_ZRS
```

Parametry podajemy jako `nazwa=wartość` albo plikiem `@parametry.json`. Wartość spoza listy `@allowed` zatrzyma wdrożenie przed utworzeniem czegokolwiek.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Utworzenie maszyny w istniejącej podsieci w innej grupie zasobów: az vm create

```
PODSIEC=$(az network vnet subnet show --resource-group rg-lab-siec --vnet-name vnet-lab-app --name snet-web --query id -o tsv)
az vm create --resource-group rg-lab-obliczenia --name vm-lab-web01 --image Ubuntu2204 --size NazwaRozmiaru \
  --zone 1 --subnet "$PODSIEC" --private-ip-address 10.32.16.4 --public-ip-address "" --nsg "" \
  --admin-username tomek --generate-ssh-keys --tags projekt=az104-lab sprzatanie=tak
```

Pusty `--public-ip-address ""` i `--nsg ""` oznaczają maszynę bez publicznego adresu i bez własnej grupy zabezpieczeń na karcie (podsieć ma już `nsg-lab-web`). Dostęp do niej mamy przez Bastion w dniu 19. W PowerShell prostą maszynę tworzy `New-AzVM -ResourceGroupName rg-lab-obliczenia -Name NazwaMaszyny -Location polandcentral -Image Ubuntu2204 -Size NazwaRozmiaru -Credential (Get-Credential)`.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ **ZMIENIA SYSTEM**

Lista rozmiarów, na które można zmienić maszynę: az vm list-vm-resize-options

```
az vm list-vm-resize-options --resource-group rg-lab-obliczenia --name vm-lab-web01 -o table
```

Pokazuje rozmiary dostępne na sprzęcie, na którym maszyna teraz stoi. Rozmiaru spoza listy nie ustawimy bez dealokacji.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ **ZMIENIA SYSTEM**

Zmiana rozmiaru maszyny: az vm resize

```
az vm resize --resource-group rg-lab-obliczenia --name vm-lab-web01 --size NazwaRozmiaru
```

Zmiana rozmiaru restartuje maszynę. Gdy docelowego rozmiaru nie ma na liście, najpierw `az vm deallocate`, potem `az vm resize`.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Dołączenie nowego dysku danych: az vm disk attach --new

```
az vm disk attach --resource-group rg-lab-obliczenia --vm-name vm-lab-web01 --name dysk-web01-dane --new --size-gb 32 --sku StandardSSD_LRS
```

Tworzy dysk zarządzany i dołącza go do maszyny. W systemie trzeba go jeszcze podzielić na partycje i sformatować, tak jak na stronie o LVM w książce Linux.

□□ **LVM** = Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny · **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Zatrzymanie maszyny ze zwolnieniem sprzętu: az vm deallocate

```
az vm deallocate --resource-group rg-lab-obliczenia --name vm-lab-web01
```

Maszyna w stanie „Stopped (deallocated)” nie jest rozliczana za moc obliczeniową, za dyski tak. W PowerShell: `Stop-AzVM -ResourceGroupName rg-lab-obliczenia -Name vm-lab-web01 -Force`.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Rejestracja funkcji szyfrowania na hoście: az feature register

```
az feature register --namespace Microsoft.Compute --name EncryptionAtHost
```

Raz na subskrypcję, zanim włączymy szyfrowanie na hoście. Stan sprawdza `az feature show --namespace Microsoft.Compute --name EncryptionAtHost`.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ ZMIENIA SYSTEM

Włączenie szyfrowania na hoście istniejącej maszyny: az vm update --set

```
az vm deallocate --resource-group rg-lab-obliczenia --name vm-lab-web01
az vm update --resource-group rg-lab-obliczenia --name vm-lab-web01 --set
securityProfile.encryptionAtHost=true
az vm start --resource-group rg-lab-obliczenia --name vm-lab-web01
```

Przy nowej maszynie wystarczy parametr `--encryption-at-host` w `az vm create`. W PowerShell: `Update-AzVM -ResourceGroupName rg-lab-obliczenia -VM $vm -EncryptionAtHost $true` na zdealokowanej maszynie.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ ZMIENIA SYSTEM

Utworzenie zestawu dostępności: az vm availability-set create

```
az vm availability-set create --resource-group rg-lab-obliczenia --name avset-lab-web \
  --platform-fault-domain-count 2 --platform-update-domain-count 5
```

Maszynę dodajemy do zestawu tylko przy tworzeniu (`az vm create --availability-set avset-lab-web`). Istniejącej maszyny nie da się do niego przenieść bez ponownego utworzenia.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Przeniesienie maszyny do innej grupy zasobów: az resource move

```
az resource move --destination-group rg-lab-dr --ids $(az vm show --resource-group rg-
lab-obliczenia --name vm-lab-web01 --query id -o tsv)
```

Razem z maszyną przenosimy jej dyski i kartę sieciową. Do innego regionu maszynę przenosi Azure Resource Mover albo Site Recovery, a nie `az resource move`.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ ZMIENIA SYSTEM

Utworzenie zestawu skalowania w trzech strefach: az vmss create

```
az vmss create --resource-group rg-lab-obliczenia --name vmss-lab-web --image Ubuntu2204
--vm-sku NazwaRozmiaru \
```

```
  --instance-count 2 --zones 1 2 3 --orchestration-mode Flexible \
  --subnet "$PODSIEC" --admin-username tomek --generate-ssh-keys
```

W CLI 2.91 domyślnym trybem jest `Flexible`. Tryb `Uniform` tworzy identyczne instancje z jednego modelu.

CLI = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Ręczna zmiana liczby instancji: az vmss scale

```
az vmss scale --resource-group rg-lab-obliczenia --name vmss-lab-web --new-capacity 3
```

W PowerShell: `Update-AzVmss -ResourceGroupName rg-lab-obliczenia -VMScaleSetName vmss-lab-web -SkuCapacity 3`.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Ustawienie autoskalowania zestawu: az monitor autoscale create

```
az monitor autoscale create --resource-group rg-lab-obliczenia --name autoskala-web \
  --resource vmss-lab-web --resource-type Microsoft.Compute/virtualMachineScaleSets \
  --min-count 2 --max-count 5 --count 2
```

Tworzy ustawienia autoskalowania z granicami. Same granice niczego nie skalują, potrzebne są reguły.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Dodanie reguł skalowania w górę i w dół: az monitor autoscale rule create

```
az monitor autoscale rule create --resource-group rg-lab-obliczenia --autoscale-name
autoskala-web \
  --scale out 1 --condition "Percentage CPU > 75 avg 10m"
```

```
az monitor autoscale rule create --resource-group rg-lab-obliczenia --autoscale-name autoskala-web \
  --scale in 1 --condition "Percentage CPU < 25 avg 10m"
```

Zawsze dodajemy parę reguł: bez reguły w dół zestaw urośnie i zostanie duży.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ **ZMIENIA SYSTEM**

Utworzenie planu App Service: az appservice plan create

```
az appservice plan create --resource-group rg-lab-obliczenia --name asp-lab --sku S1 --is-linux
```

Plan to maszyny, na których działają aplikacje, i to plan jest rozliczany, nie aplikacja. Poziom S1 (Standard) wybieramy, bo sloty wdrożeniowe wymagają co najmniej Standard. W PowerShell: `New-AzAppServicePlan -ResourceGroupName rg-lab-obliczenia -Name asp-lab -Location polandcentral -Tier Standard -Linux`.

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ **ZMIENIA SYSTEM**

Utworzenie aplikacji: az webapp create

```
az webapp create --resource-group rg-lab-obliczenia --plan asp-lab --name NazwaAplikacji --runtime "NazwaSrodowiska"
```

(gdzie `NazwaSrodowiska` to wartość z `az webapp list-runtimes --os-type linux`, w pomocy CLI 2.91 przykładem jest `PYTHON:3.14`)

Aplikacja dostaje adres `NazwaAplikacji.azurewebsites.net`. W PowerShell: `New-AzWebApp -ResourceGroupName rg-lab-obliczenia -Name NazwaAplikacji -AppServicePlan asp-lab`.

□□ **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

□□ **ZMIENIA SYSTEM**

Ręczne skalowanie planu w poziomie i w pionie: az appservice plan update

```
az appservice plan update --resource-group rg-lab-obliczenia --name asp-lab --number-of-workers 2
az appservice plan update --resource-group rg-lab-obliczenia --name asp-lab --sku P1V3
```

Pierwsza komenda dokłada instancję (w poziomie), druga zmienia poziom planu (w pionie). Autoskalowanie planu ustawiamy tak jak dla zestawu skalowania, z `--resource-type Microsoft.Web/serverfarms`.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Utworzenie slotu wdrożeniowego: az webapp deployment slot create

```
az webapp deployment slot create --resource-group rg-lab-obliczenia --name NazwaAplikacji --slot staging
```

Slot to druga aplikacja z własnym adresem `NazwaAplikacji-staging.azurewebsites.net` na tym samym planie. W PowerShell: `New-AzWebAppSlot -ResourceGroupName rg-lab-obliczenia -Name NazwaAplikacji -Slot staging`.

az = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Zamiana slotu z produkcją: az webapp deployment slot swap

```
az webapp deployment slot swap --resource-group rg-lab-obliczenia --name NazwaAplikacji --slot staging --target-slot production
```

Zamiana przełącza ruch bez przerwy, a wycofanie to ta sama komenda jeszcze raz. W PowerShell: `Switch-AzWebAppSlot -ResourceGroupName rg-lab-obliczenia -Name NazwaAplikacji -SourceSlotName staging -DestinationSlotName production`.

az = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Przypisanie własnej nazwy DNS: az webapp config hostname add

```
az webapp config hostname add --resource-group rg-lab-obliczenia --webapp-name NazwaAplikacji --hostname www.azlab.zielona.example
```

Przed tym poleceniem w strefie DNS musi stać rekord CNAME na `NazwaAplikacji.azurewebsites.net` (subdomena) albo rekord A z rekordem TXT `asuid` (domena główna). Plan musi być płatny, nie Free (F1).

DNS = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **CNAME** = Canonical Name: rekord DNS będący aliasem, drugą nazwą wskazującą na już istniejący rekord A · **TXT** = typ rekordu DNS przechowującego dowolny tekst, wykorzystywany między innymi do publikowania reguł SPF, DKIM i DMARC · **az** = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Utworzenie certyfikatu zarządzanego dla nazwy: az webapp config ssl create

```
az webapp config ssl create --resource-group rg-lab-obliczenia --name NazwaAplikacji --hostname www.azlab.zielona.example
```

Certyfikat zarządzany przez App Service jest bezpłatny od poziomu Basic. Powiązanie z nazwą robi `az webapp config ssl bind --certificate-thumbprint OdciskCertyfikatu --ssl-type SNI --name NazwaAplikacji --resource-group rg-lab-obliczenia`. Polecenie `ssl create` jest w CLI 2.91 oznaczone jako wersja zapoznawcza.

SNI = Server Name Indication: pole w pierwszym komunikacie TLS, w którym klient podaje nazwę serwera, z którym chce rozmawiać, co pozwala HAProxy wybrać backend bez rozszyfrowywania ruchu · **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Wymuszenie HTTPS i minimalnej wersji TLS: az webapp update, az webapp config set

```
az webapp update --resource-group rg-lab-obliczenia --name NazwaAplikacji --https-only true
az webapp config set --resource-group rg-lab-obliczenia --name NazwaAplikacji --min-tls-version 1.2
```

Ustawienia protokołów zostają przy slocie i nie przechodzą przy zamianie.

HTTPS = HTTP zabezpieczony szyfrowaniem TLS · **TLS** = Transport Layer Security: szyfrowanie połączeń, np. w HTTPS · **az** = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Utworzenie kopii aplikacji na żądanie: az webapp config backup create

```
az webapp config backup create --resource-group rg-lab-obliczenia --webapp-name
NazwaAplikacji \
  --backup-name kopia-przed-zmiana --container-url "AdresKonteneraZSas"
```

Kopia własna wymaga konta magazynu z autoryzacją SAS. Kopie automatyczne działają bez konta, od poziomu Basic.

SAS = Serial Attached SCSI: złącze i magistrala dyskowa w serwerach, następca równoległego SCSI, pozwala podłączyć wiele dysków i macierzy przez jeden kontroler z większą przepustowością niż SATA; w Azure także Shared Access Signature: podpisany adres z tokenem, który daje czasowy dostęp do danych w koncie magazynu · **az** = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Integracja aplikacji z siecią wirtualną: az webapp vnet-integration add

```
az webapp vnet-integration add --resource-group rg-lab-obliczenia --name NazwaAplikacji
\
  --vnet /subscriptions/IdSubskrypcji/resourceGroups/rg-lab-
siec/providers/Microsoft.Network/virtualNetworks/vnet-lab-app \
  --subnet snet-appsvc
```

Ruch wychodzący aplikacji idzie wtedy przez podsieć `snet-appsvc` (delegowaną do `Microsoft.Web/serverFarms`), więc aplikacja dosięgnie bazy w `vnet-lab-dane` przez peering i trasy. Ruchu przychodzącego to nie zmienia: do tego służy prywatny punkt końcowy aplikacji.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Utworzenie rejestru kontenerów: az acr create

```
az acr create --resource-group rg-lab-obliczenia --name NazwaRejestru --sku Basic
```

Poziomy to Basic, Standard i Premium. Konto administratora rejestru jest domyślnie wyłączone. W PowerShell: `New-AzContainerRegistry -ResourceGroupName rg-lab-obliczenia -Name NazwaRejestru -Location polandcentral -Sku Basic`.

az = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Zbudowanie obrazu w rejestrze bez lokalnego Dockera: az acr build

```
az acr build --registry NazwaRejestru --image zielona-www:1.0 .
```

Wysyła bieżący katalog z plikiem `Dockerfile` do rejestru i buduje obraz w Azure. W laboratorium `Dockerfile` ma jedną linię `FROM nginx:1.27-alpine` i kopiuje plik `index.html`.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Uruchomienie kontenera w Azure Container Instances: az container create

```
az container create --resource-group rg-lab-obliczenia --name aci-lab-www --image
NazwaRejestru.azurecr.io/zielona-www:1.0 \
  --os-type Linux --cpu 1 --memory 1.5 --ports 80 --dns-name-label NazwaEtykiety --
restart-policy Always \
  --registry-login-server NazwaRejestru.azurecr.io --registry-username NazwaUzytkownika
--registry-password HasloRejestru
```

Rozmiar grupy kontenerów ustawiamy procesorem i pamięcią przy tworzeniu. ACI nie ma autoskalowania: więcej mocy to większa grupa albo więcej grup.

ACI = Azure Container Instances: usługa uruchamiająca pojedyncze grupy kontenerów w Azure bez zarządzania serwerem ani klastrem · **az** = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Uruchomienie kontenera w ACI z PowerShell: New-AzContainerGroup

```
$port = New-AzContainerInstancePortObject -Port 80 -Protocol TCP
$kontener = New-AzContainerInstanceObject -Name www -Image
NazwaRejestru.azurecr.io/zielona-www:1.0 -RequestCpu 1 -RequestMemoryInGb 1.5 -Port
@($port)
New-AzContainerGroup -ResourceGroupName rg-lab-obliczenia -Name aci-lab-www -Location
polandcentral `
  -Container $kontener -OsType Linux -IpAddressType Public
```

Moduł `Az.ContainerInstance` buduje grupę z obiektów: najpierw port, potem kontener, na końcu grupa. Do prywatnego rejestru dochodzi obiekt poświadczeń z `New-AzContainerGroupImageRegistryCredentialObject`.

ACI = Azure Container Instances: usługa uruchamiająca pojedyncze grupy kontenerów w Azure bez zarządzania serwerem ani klastrem · **TCP** = Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia

ZMIENIA SYSTEM

Utworzenie środowiska Container Apps: az containerapp env create

```
az containerapp env create --resource-group rg-lab-obliczenia --name cae-lab --location
polandcentral
```

Środowisko to wspólna granica sieci i logów dla kilku aplikacji kontenerowych.

az = Azure CLI: narzędzie chmury Microsoftu

ZMIENIA SYSTEM

Utworzenie aplikacji kontenerowej ze skalowaniem do zera: az containerapp create

```
az containerapp create --resource-group rg-lab-obliczenia --name ca-lab-www --
environment cae-lab \
  --image NazwaRejestru.azurecr.io/zielona-www:1.0 --target-port 80 --ingress external \
  --min-replicas 0 --max-replicas 5 --cpu 0.5 --memory 1.0Gi \
  --registry-server NazwaRejestru.azurecr.io --registry-username NazwaUzytkownika --
registry-password HasloRejestru
```

Przy `--min-replicas 0` aplikacja bez ruchu nie zużywa replik. Granice zmieniamy później przez `az containerapp update --min-replicas 1 --max-replicas 10`. Dopuszczalne pary procesora i pamięci sprawdzamy w dokumentacji Container Apps.

az = Azure CLI: narzędzie chmury Microsoftu

Dzień 12: szablony ARM i pliki Bicep

Z ŻYCIA WZIĘTE

Poniedziałek, 9:15. Ola przysyła Tomkowi szablon ARM z innego projektu i prosi, żeby wdrożył z niego konto magazynu z nadmiarowością strefową. Tomek otwiera 300 linii JSON i szuka, gdzie wpisać ZRS. Pokazujemy mu sekcję `parameters` z listą `allowedValues` i parametr `--parameters sku=Standard_ZRS`, potem `az bicep decompile`, po którym ten sam szablon ma 40 linii. Tomek wdraża, a w dzienniku zapisuje: najpierw czytam parametry, potem zasoby.

Cele egzaminu

- interpretacja szablonu ARM albo pliku Bicep
- modyfikacja istniejącego szablonu ARM
- modyfikacja istniejącego pliku Bicep
- wdrażanie zasobów z szablonu ARM albo pliku Bicep
- eksport wdrożenia jako szablonu ARM albo konwersja szablonu ARM na plik Bicep

Jak to działa

Pojęcie: szablon ARM. Plik JSON opisujący zasoby, które mają istnieć, a nie kroki do wykonania. Azure Resource Manager porównuje opis ze stanem i tworzy albo zmienia to, czego brakuje. Bicep jest krótszym zapisem tego samego: przed wdrożeniem kompiluje się do szablonu ARM.

Tak wygląda szablon ARM, który powstaje z pliku `konto-magazynu.bicep` (wynik `az bicep build --stdout`, bez sekcji `metadata`):

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "nazwaKonta": {
      "type": "string",
      "minLength": 3,
      "maxLength": 24,
      "metadata": {
        "description": "Nazwa konta: male litery i cyfry, unikatowa w calym Azure"
      }
    },
    "lokalizacja": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]"
    },
    "sku": {
      "type": "string",
      "defaultValue": "Standard_LRS",
      "allowedValues": [
        "Standard_LRS",
        "Standard_ZRS"
      ]
    }
  },
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2023-05-01",
      "name": "[parameters('nazwaKonta')]",
      "location": "[parameters('lokalizacja')]",
      "sku": {
        "name": "[parameters('sku')]"
      },
      "kind": "StorageV2",
      "properties": {
        "accessTier": "Hot",
        "minimumTlsVersion": "TLS1_2",
        "allowBlobPublicAccess": false,
```

```
        "supportsHttpsTrafficOnly": true
      }
    }
  ],
  "outputs": {
    "punktKoncowyBlob": {
      "type": "string",
      "value": "[reference(resourceId('Microsoft.Storage/storageAccounts',
parameters('nazwaKonta')), '2023-05-01').primaryEndpoints.blob]"
    }
  }
}
```

Sekcja szablonu ARM	Odpowiednik w Bicepie	Na co patrzymy w pytaniu
`parameters`	`param` z dekoratorami `@allowed`, `@minLength`	jakie wartości przejdą, jaka jest domyślna
`variables`	`var`	wartości wyliczane w szablonie
`resources`	`resource nazwa 'Typ@wersja'`	typ zasobu, `dependsOn` (w Bicepie zwykle niejawne)
`outputs`	`output`	co wdrożenie zwraca
funkcje `[...]`	wyrażenia bez nawiasów kwadratowych	`resourceGroup().location`, `concat`, `uniqueString`

Tryb wdrożenia `Incremental` (domyślny) dokłada i zmienia zasoby z pliku, a pozostałe w grupie zostawia. Tryb `Complete` usuwa z grupy wszystko, czego nie ma w pliku, dlatego w laboratorium używamy go tylko świadomie i zawsze po `what-if`.

Zadania laboratoryjne

1. Wdrażamy `konto-magazynu.bicep` z parametrem `sku=Standard_ZRS`, potem próbujemy `sku=Standard_GRS` i zapisujemy komunikat błędu.
2. Dopisujemy w pliku Bicep tag `projekt: 'az104-lab'` i parametr `@allowed` z wartością `Standard_GRS`, sprawdzamy `az deployment group what-if`, wdrażamy.
3. Eksportujemy `rg-lab-siec` w formacie `json` i `bicep`, porównujemy oba pliki z naszym `lab-siec.bicep`.
4. Wynik eksportu JSON zamieniamy `az bicep decompile` i kompilujemy z powrotem `az bicep build`.

Pułapki egzaminacyjne

- W pytaniu z fragmentem szablonu najpierw szukamy `allowedValues` i `defaultValue`: często cała odpowiedź tam jest.
- Tryb `Complete` usuwa zasoby spoza szablonu. Tryb domyślny tego nie robi.
- Szablon wdrażany na zakres grupy zasobów nie utworzy grupy zarządzania ani subskrypcji; do zasobów na poziomie subskrypcji służy `az deployment sub create`.
- `az group export` może nie wyeksportować wszystkich typów zasobów i zgłasza to ostrzeżeniem.

Ćwiczenia

1. W szablonie parametr `vmSize` ma `allowedValues` z trzema rozmiarami i `defaultValue` równe pierwszemu. Wdrażamy bez podania parametru. Co się stanie, a co przy podaniu czwartego rozmiaru?

Odpowiedź

Bez parametru wdrożenie użyje wartości domyślnej, czyli pierwszego rozmiaru. Podanie rozmiaru spoza listy zatrzyma wdrożenie na walidacji, zanim powstanie jakikolwiek zasób.

Dzień 13: maszyny wirtualne, rozmiary, dyski i szyfrowanie

> **Z ŻYCIA WZIĘTE**
>
> Wtorek, 11:20. Tomek tworzy `vm-lab-web01` i chce zmienić jej rozmiar na większy, ale `az vm resize` zgłasza, że rozmiar jest niedostępny. Sprawdzamy `az vm list-vm-resize-options`: rozmiaru nie ma na liście, bo nie ma go na klastrze, na którym maszyna stoi. Po `az vm deallocate` maszyna może wylądować na innym sprzęcie i zmiana się udaje. Tomek zapisuje, że dealokacja zmienia też publiczny adres dynamiczny, gdyby maszyna go miała.

Cele egzaminu

- tworzenie maszyny wirtualnej
- konfiguracja szyfrowania na hoście
- przenoszenie maszyny do innej grupy zasobów, subskrypcji albo regionu
- zarządzanie rozmiarami maszyn
- zarządzanie dyskami maszyn

Jak to działa

> **Pojęcie: maszyna wirtualna w Azure.** Zestaw kilku zasobów, a nie jeden obiekt: sama maszyna (rozmiar, obraz), dysk systemowy, dyski danych, karta sieciowa, opcjonalnie publiczny adres i grupa zabezpieczeń. Dlatego usunięcie samej maszyny może zostawić dyski i kartę, o ile nie ustawiono opcji usuwania razem z maszyną.

Dysk zarządzany	Do czego	Uwagi
Standard HDD	kopie, rzadki dostęp	najtańszy
Standard SSD	serwery WWW, laboratorium	dobry wybór domyślny
Premium SSD	bazy, produkcja	wpływa na SLA pojedynczej maszyny
Premium SSD v2, Ultra	wymagające bazy	wydajność ustawiana osobno

Szyfrowanie	Co szyfruje	Uwagi
Szyfrowanie po stronie serwera (SSE)	dane dysków w magazynie	zawsze włączone, klucze Microsoftu albo własne
Szyfrowanie na hoście	dodatkowo dyski tymczasowe i pamięć podręczną na hoście	rejestracja funkcji `EncryptionAtHost`, włączanie na zdealokowanej maszynie
Azure Disk Encryption	szyfrowanie w systemie (BitLocker, dm-crypt)	starsze rozwiązanie, Microsoft wskazuje szyfrowanie na hoście jako kierunek (do potwierdzenia w aktualnej dokumentacji)

Przenoszenie: do innej grupy zasobów i subskrypcji służy operacja przeniesienia (portal, `az resource move`), a do innego regionu Azure Resource Mover albo Site Recovery. Zasady przenoszenia maszyn z dyskami i kartami opisuje dokumentacja „Move operation support for resources”.

Zadania laboratoryjne

1. Tworzymy `vm-lab-web01` w strefie 1 bez publicznego adresu (fiszka), dołączamy dysk danych 32 GB i w systemie sprawdzamy go `lsblk`.
2. Sprawdzamy listę rozmiarów i zmieniamy rozmiar maszyny.
3. Rejestrujemy `EncryptionAtHost`, dealokujemy maszynę i włączamy szyfrowanie na hoście.
4. Przenosimy maszynę do `rg-lab-dr` razem z dyskiem i kartą, sprawdzamy nowy identyfikator i przenosimy z powrotem.
5. Wieczorem dealokujemy maszynę, jeśli ma przeżyć do dnia 15, albo usuwamy grupę.

Pułapki egzaminacyjne

- Rozmiar spoza listy `list-vm-resize-options` wymaga dealokacji.
- Szyfrowania na hoście nie włączymy na działającej maszynie.
- Usunięcie maszyny nie musi usuwać dysków i karty sieciowej.
- Przeniesienie do innego regionu to nie `az resource move`.

Ćwiczenia

1. Maszyna ma rozmiar, którego nie ma już na liście zmian. Jak zmienić rozmiar i co stracimy po drodze?

Odpowiedź

Dealokujemy maszynę (`az vm deallocate`), zmieniamy rozmiar i uruchamiamy. Tracimy zawartość dysku tymczasowego i dynamiczny publiczny adres, jeżeli maszyna go miała, a maszyna jest niedostępna na czas operacji.

Dzień 14: powtórka tygodnia 2

Wracamy do dzienników błędów z dni 8 do 13. Mini sprawdzian, dziesięć pytań na 15 minut:

1. Który wariant nadmiarowości daje stały odczyt z drugiego regionu?
2. Na jakich kontach działa warstwa Archive?
3. Jak cofnąć SAS konta?
4. Czym różni się wersjonowanie od miękkiego usuwania?
5. Co jest potrzebne do replikacji obiektów?
6. Jaki tryb wdrożenia usuwa zasoby spoza szablonu?
7. Gdzie w szablonie ARM szukamy dozwolonych wartości?
8. Co trzeba zrobić przed zmianą rozmiaru na niedostępny?
9. Czy maszyna zatrzymana z systemu jest rozliczana za moc obliczeniową?
10. Czym przeniesiemy maszynę do innego regionu?

Odpowiedzi

1. RA-GRS i RA-GZRS. 2. LRS, GRS, RA-GRS. 3. Tylko odnowieniem klucza, którym go podpisano. 4. Wersjonowanie zachowuje poprzednie treści przy każdej zmianie, miękkie usuwanie pozwala odzyskać usunięty blob przez określony czas. 5. Wersjonowanie na obu kontach i kanał zmian na źródłowym. 6. Complete. 7. `allowedValues` w sekcji `parameters`. 8. Dealokację. 9. Tak, dopóki nie jest

zdealokowana. 10. Azure Resource Mover albo Site Recovery.

Dzień 15: strefy, zestawy dostępności i zestawy skalowania

Z ŻYCIA WZIĘTE

Środa, 6:05. Marek dzwoni z magazynu, że portal zamówień dla przewoźników nie odpowiada, a w laboratorium Tomek właśnie przećwiczył dzień wcześniej awarię strefy: wyłączył `vm-lab-web01` w strefie 1, a `vm-lab-web02` w strefie 2 obsługiwała ruch za Load Balancerem. Produkcja Zielonej to jeszcze nie Azure, ale Tomek wie już, o co zapytać, gdy przyjdzie czas: czy dwie maszyny stoją w dwóch strefach, czy tylko w dwóch szafach.

Cele egzaminu

- wdrażanie maszyn w strefach dostępności i zestawach dostępności
- wdrażanie i konfiguracja zestawów skalowania maszyn wirtualnych

Jak to działa



Rysunek 1. Dostępność maszyn: zestaw dostępności rozkłada maszyny na domeny błędów i aktualizacji w jednym centrum danych, strefy rozdzielają je na fizycznie oddzielne centra danych, a

zestaw skalowania tworzy wiele jednakowych instancji z autoskalowaniem, także w strefach (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: domena błędów i domena aktualizacji.** Domena błędów to grupa sprzętu ze wspólnym zasilaniem i przełącznikiem sieciowym, domena aktualizacji to grupa hostów restartowanych razem podczas prac Microsoftu. Zestaw dostępności rozkłada maszyny tak, żeby jedna awaria szafy i jedno okno aktualizacji nie zabrały wszystkich naraz.

Sposób	Przed czym chroni	SLA według umowy dla maszyn (sprawdzamy aktualną wersję)	Uwagi
Pojedyncza maszyna	nic ponad sprzęt	zależne od dysków (najwyższe przy Premium SSD)	
Zestaw dostępności	awaria szafy, aktualizacja hosta	99,95% dla dwóch i więcej maszyn	jedno centrum danych
Strefy dostępności	awaria całej strefy	99,99% dla dwóch i więcej maszyn w dwóch strefach	wymaga regionu ze strefami
Zestaw skalowania	jak wyżej, zależnie od rozmieszczenia		dodaje autoskalowanie

> **Pojęcie: zestaw skalowania (VMSS).** Grupa maszyn zarządzana jako całość, z liczbą instancji zmienianą ręcznie albo przez autoskalowanie według metryk i harmonogramów. W trybie Uniform wszystkie instancje są z jednego modelu, w trybie Flexible można łączyć różne maszyny i stosować te same mechanizmy co dla zwykłych maszyn.

Zadania laboratoryjne

1. Tworzymy `vm-lab-web02` w strefie 2 z adresem 10.32.16.5. Instalujemy na obu maszynach nginx z różną stroną startową.
2. Tworzymy zestaw dostępności `avset-lab-web` i próbujemy dodać do niego istniejącą maszynę: zapisujemy, dlaczego się nie da.
3. Tworzymy `vmss-lab-web` z dwiema instancjami w trzech strefach, ustawiamy autoskalowanie 2 do 5 z parą reguł.
4. Obciążamy procesor jednej instancji (na przykład `stress-ng` albo pętla `sha256sum /dev/zero`) i obserwujemy w portalu historię skalowania.
5. Usuwamy zestaw skalowania tego samego dnia.

Pułapki egzaminacyjne

- Maszynę dodajemy do zestawu dostępności tylko przy tworzeniu.
- Jedna maszyna nie może być jednocześnie w zestawie dostępności i w strefie.
- Autoskalowanie bez reguły w dół nie zmniejsza zestawu.
- Okres wyciszenia (cooldown) po skalowaniu sprawia, że kolejna zmiana nie przychodzi od razu.

Ćwiczenia

1. Aplikacja ma przetrwać awarię jednego centrum danych w regionie `polandcentral`. Wybieramy zestaw dostępności czy strefy?

Odpowiedź

Strefy dostępności, z maszynami w co najmniej dwóch strefach i równoważeniem ruchu (Load Balancer Standard). Zestaw dostępności chroni tylko w obrębie jednego centrum danych.

Dzień 16: App Service

> **Z ŻYCIA WZIĘTE**
>
> Czwartek, 15:30. Ola chce wypuścić nową wersję strony z cennikiem dla odbiorców hurtowych, ale boi się przerwy w piątek po południu. Tomek wdraża nową wersję do slotu `staging`, Ola klika po niej przez kwadrans, a potem jedna zamiana slotów przełącza ruch. Po pięciu minutach pani Halina zgłasza, że ceny pokazują się bez VAT, więc Tomek wykonuje zamianę jeszcze raz i stara wersja wraca. Ola notuje, że ustawienie z adresem bazy cenowej musi być przyklejone do slotu.

Cele egzaminu

- tworzenie planu App Service
- konfiguracja skalowania planu App Service
- tworzenie aplikacji App Service
- konfiguracja certyfikatów i TLS dla aplikacji
- mapowanie własnej nazwy DNS na aplikację
- konfiguracja kopii zapasowej aplikacji
- konfiguracja ustawień sieciowych aplikacji
- konfiguracja slotów wdrożeniowych

Jak to działa

> **Pojęcie: plan App Service.** Zestaw maszyn (instancji) o określonym poziomie, na którym działa jedna albo wiele aplikacji. Wszystkie aplikacje w planie dzielą jego moc i skalują się razem z nim.

Funkcja	Od jakiego poziomu (dokumentacja Microsoftu)
Własna nazwa DNS	płatny poziom, nie Free (F1)
Certyfikat zarządzany App Service	Basic
Powiązanie TLS oparte na IP	Standard
Kopie automatyczne i własne	Basic (w Basic tylko slot produkcyjny)
Integracja z siecią wirtualną	Basic, Standard, Premium (v2, v3, v4), Elastic Premium
Sloty wdrożeniowe	Standard (w Standard pięć slotów), Premium, Isolated
Autoskalowanie według reguł	Standard (do potwierdzenia na stronie limitów App Service)

Przy zamianie slotów przechodzą między innymi ustawienia aplikacji i parametry połączeń (o ile nie są przyklejone do slotu), wersja środowiska i mapowania ścieżek. Zostają przy slocie: nazwy domen, certyfikaty niepubliczne i ustawienia TLS, skalowanie, ograniczenia IP, Always On, integracja z siecią wirtualną i tożsamości zarządzane. Ustawienie przyklejone do slotu („Deployment slot setting”) też zostaje.

Kopie automatyczne trzymane są 30 dni co godzinę, potem rzadziej, do 30 GB. Kopie własne mają limit 10 GB, w tym 4 GB bazy, i wymagają konta magazynu z SAS (tożsamość zarządzana nie jest obsługiwana).

Sieć ma dwa kierunki. Ruch wychodzący z aplikacji do sieci wirtualnej daje integracja z siecią (podsieć delegowana, minimum /28, zalecane /26). Ruch przychodzący tylko z sieci prywatnej daje prywatny punkt końcowy aplikacji, a ograniczenia dostępu filtrują ruch publiczny po adresach.

Zadania laboratoryjne

1. Tworzymy plan `asp-lab` (S1, Linux) i aplikację, wdrażamy do niej prostą stronę.
2. Tworzymy slot `staging`, wdrażamy inną stronę, dodajemy ustawienie aplikacji `BAZA_CEN` z zaznaczonym „Deployment slot setting” i zamieniamy sloty.
3. Skalujemy plan do dwóch instancji, potem ustawiamy autoskalowanie planu według procesora.
4. W strefie `azlab.zielona.example` (dzień 20) dodajemy CNAME `www` i przypisujemy nazwę do aplikacji, potem certyfikat zarządzany i powiązanie SNI.

5. Integrujemy aplikację z `snet-appsvc` i z konsoli aplikacji sprawdzamy połączenie z 10.32.32.4.
6. Robimy kopię na żądanie do kontenera z SAS. Wieczorem usuwamy plan razem z aplikacją.

Pułapki egzaminacyjne

- Sloty nie istnieją w Free, Shared i Basic.
- Integracja z siecią dotyczy ruchu wychodzącego, prywatny punkt końcowy przychodzącego.
- Plan jest rozliczany także wtedy, gdy aplikacje są zatrzymane.
- Własna domena główna wymaga rekordu A i TXT `asuid`, subdomena rekordu CNAME.

Ćwiczenia

1. Po zamianie slotów aplikacja produkcyjna łączy się z testową bazą danych. Co poszło nie tak i jak zapobiec?

Odpowiedź

Parametr połączenia nie był oznaczony jako ustawienie slotu, więc przeszedł razem z kodem ze `staging` do produkcji. Zaznaczamy go jako „Deployment slot setting” w obu slotach, wtedy przy zamianie zostaje na miejscu.

Dzień 17: kontenery w ACR, ACI i Container Apps

> **□□ Z ŻYCIA WZIĘTE**
>
> Piątek, 10:00. Ola ma mały kontener z narzędziem do przeliczania palet, uruchamiany raz dziennie, i pyta, czy musi stawiać dla niego maszynę. Tomek buduje obraz w rejestrze `az acr build`, a potem uruchamia go w ACI z polityką restartu `Never`: kontener robi swoje i się kończy. Do strony z podglądem stanu magazynu, która ma działać stale i rosnąć przy ruchu, wybiera Container Apps z minimum jedną repliką.

Cele egzaminu

- tworzenie rejestru Azure Container Registry i zarządzanie nim
- uruchamianie kontenera w Azure Container Instances
- uruchamianie kontenera w Azure Container Apps
- zarządzanie rozmiarem i skalowaniem kontenerów, w tym ACI i Container Apps

Jak to działa

Usługa	Do czego	Skalowanie	Uwagi
ACR	prywatny rejestr obrazów	nie dotyczy	Basic, Standard, Premium (replikacja geograficzna i Private Link w Premium)
ACI	pojedyncza grupa kontenerów, zadania, krótkie procesy	rozmiar przy tworzeniu, brak autoskalowania	polityka restartu Always, OnFailure, Never
Container Apps	aplikacje i mikrousługi, HTTP, zdarzenia	reguły skalowania, także do zera replik	środowisko, rewizje, wejście HTTP
AKS	pełny Kubernetes	węzły i pody	poza listą umiejętności z 17 kwietnia 2026

AKS opisujemy w bazie na stronie Nomad, Podman z Quadlet i zarządzany Kubernetes: AKS i EKS, a samego Kubernetesa w książce Kubernetes. Na egzaminie wystarczy rozpoznać, kiedy prostszą odpowiedzią jest ACI albo Container Apps.

Zadania laboratoryjne

1. Tworzymy rejestr Basic, budujemy obraz `zielona-www:1.0` przez `az acr build` i oglądamy repozytoria `az acr repository list --name NazwaRejestru`.
2. Uruchamiamy obraz w ACI z etykietą DNS i otwieramy stronę w przeglądarce. Powtarzamy to samo w PowerShell.
3. Tworzymy środowisko Container Apps i aplikację z wejściem zewnętrznym, minimum 0 i maksimum 5 replik. Zmieniamy minimum na 1 i obserwujemy czas pierwszej odpowiedzi.
4. Usuwamy grupę kontenerów i aplikację, rejestr zostawiamy do końca dnia.

Pułapki egzaminacyjne

- ACI nie skaluje się samo: pytanie o automatyczne zwiększanie liczby kontenerów według ruchu prowadzi do Container Apps.
- Rozmiar grupy ACI ustawiamy przy tworzeniu.
- Konto administratora ACR jest domyślnie wyłączone; zalecane jest logowanie tożsamością.
- Skalowanie do zera oznacza opóźnienie pierwszej odpowiedzi po okresie bez ruchu.

Ćwiczenia

1. Raport ma się wygenerować raz na noc w kontenerze i zakończyć, bez płacenia za czas bezczynności. Którą usługę i jaką politykę restartu wybieramy?

Odpowiedź

Azure Container Instances z polityką restartu `Never` (albo `OnFailure`, gdy chcemy ponowienia po błędzie), uruchamiane harmonogramem z zewnątrz. Container Apps też to potrafi (zadania), ale na liście umiejętności dla jednorazowego kontenera ACI jest najprostszą odpowiedzią.

Częste błędy

Objaw	Przyczyna	Co robimy
`az vm create` nie znajduje podsieci	sieć w innej grupie zasobów, podana sama nazwa	podajemy pełny identyfikator podsieci
Błąd limitu przy tworzeniu zestawu skalowania	za mały limit vCPU w regionie	`az vm list-usage`, mniej instancji albo mniejszy rozmiar
Nie da się zmienić rozmiaru	rozmiaru nie ma na obecnym sprzęcie	dealokacja i ponowna próba
Szyfrowanie na hoście się nie włącza	funkcja niezarejestrowana albo maszyna działa	`az feature register`, dealokacja
Slot nie daje się utworzyć	plan poniżej Standard	zmiana planu na S1 lub wyżej
Własna nazwa DNS odrzucona	brak CNAME lub TXT `asuid`, plan Free	poprawiamy rekordy, zmieniamy plan
ACI nie pobiera obrazu	brak poświadczeń do prywatnego rejestru	`--registry-login-server`, nazwa i hasło albo tożsamość
Rachunek za plan App Service rośnie, choć aplikacja zatrzymana	rozliczany jest plan, nie aplikacja	usuwamy plan albo zmieniamy na niższy poziom

Powiązane: Magazyn: konta, nadmiarowość, SAS, Blob, Azure Files i zabezpieczenia · Sieć: VNet, peering, trasy, NSG, Bastion, prywatne punkty końcowe, DNS i Load Balancer · Plan 4 tygodni i laboratorium Azure w Zielonej · Nomad, Podman z Quadlet i zarządzany Kubernetes: AKS i EKS

Źródła: Microsoft Learn: przewodnik do egzaminu AZ-104 (lista od 17 kwietnia 2026), „Set up staging environments in Azure App Service” (Standard, Premium, Isolated, pięć slotów w Standard, ustawienia zamieniane i przyklejone), „Back up and restore your app in Azure App Service” (poziomy, retencja, limity, SAS), „Integrate your app with an Azure virtual network” (poziomy, /28, /26), „Set up an existing custom domain in Azure App Service” (płatny poziom, A i TXT `asuid`, certyfikat zarządzany od Basic, IP SSL od Standard); pomoc Azure CLI 2.91.0 (`az group export`, `az bicep build`

i `decompile`, `az deployment group create`, `az vm create`, `resize`, `list-vm-resize-options`, `disk attach`, `deallocate`, `update`, `availability-set create`, `az feature register`, `az resource move`, `az vmss create` i `scale`, `az monitor autoscale create` i `rule create`, `az appservice plan create` i `update`, `az webapp create`, `deployment slot create` i `swap`, `config hostname add`, `config ssl create` i `bind`, `config backup create`, `vnet-integration add`, `update`, `config set`, `az acr create` i `build`, `az container create`, `az containerapp env create`, `create`, `update`); dokumentacja modułu Az (`Export-AzResourceGroup`, `New-AzVM`, `Stop-AzVM`, `Update-AzVM`, `Update-AzVmss`, `New-AzAppServicePlan`, `New-AzWebApp`, `New-AzWebAppSlot`, `Switch-AzWebAppSlot`, `New-AzContainerRegistry`, `New-AzContainerInstancePortObject`, `New-AzContainerInstanceObject`, `New-AzContainerGroup`), stan na 19 października 2026; Bicep CLI 0.38.33.

Revision #3
Created 2026-10-18 23:00:22 UTC by Tomasz Konieczny
Updated 2026-10-26 14:27:19 UTC by Tomasz Konieczny