

[Open in app](#)**Jonathan**[Follow](#)

29 Followers

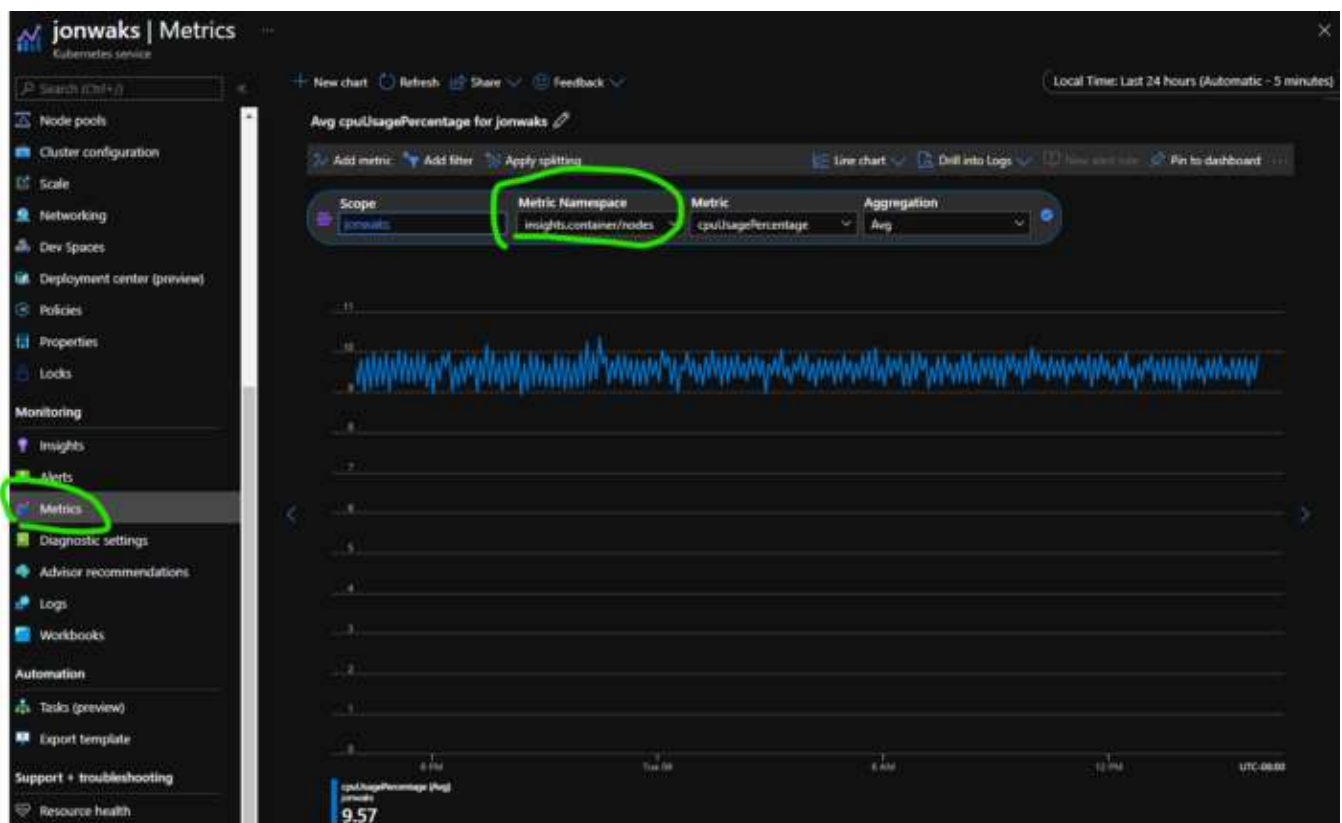
[About](#)

Use Azure Monitor REST API to Read Azure Kubernetes Service (AKS) Metrics



Jonathan Apr 4 · 4 min read

I happened to be researching how to use REST API for getting the custom metrics on AKS. The image below would show more what I am talking about here. This is not just normal metrics we could consume under Azure Monitor default REST API endpoints. These are actually metrics under Azure Monitor custom metric namespaces and it would require users to hit the right REST API endpoints for getting the data needed.



[Open in app](#)

Create Azure Active Directory (AAD) service principal (SP)

We would need to create an AAD SP as it would be the representative for consuming all sorts of Azure REST API endpoints. After following through [the first part of the PowerShell script](#), we should now have an AAD SP with sufficient permissions on reading Azure Monitor data.

```
$subscriptionId = "< your subscription ID>"
$resourceGroupName = "<resource group name>"

# Authenticate to a specific Azure subscription.
Connect-AzAccount -SubscriptionId $subscriptionId

# Password for the service principal
$pwd = "<create a password>"
$secureStringPassword = ConvertTo-SecureString -String $pwd -
AsPlainText -Force

# Create a new Azure AD application
$azureAdApplication = New-AzADApplication `
  -DisplayName "AzMonitorSP" `
  -HomePage "https://localhost/jonw-azure-monitor" `
  -IdentifierUri "https://localhost/jonw-azure-monitor" `
  -Password $secureStringPassword

# Create a new service principal associated with the designated
application
New-AzADServicePrincipal -ApplicationId
$azureAdApplication.ApplicationId

# Assign Reader role to the newly created service principal
New-AzRoleAssignment -RoleDefinitionName Reader `
  -ServicePrincipalName $azureAdApplication.ApplicationId.Guid
```

The next action would be using the newly created AAD SP and leverage client credentials grant for AAD authentication. If everything works as expected, we would retrieve an access token. Client credentials grant would not provide refresh token as it would increase the security risk. However, each access token acquired by client credentials grant could last 86400 seconds.

You can use the [OAuth 2.0 client credentials grant](#) specified in RFC 6749, sometimes called two-legged OAuth, to access web-hosted resources by using the identity of an application.

[Open in app](#)

often referred to as daemons or service accounts.

— Quoted from [here](#)

```
$subscriptionId = "<your subscription ID>"
$resourceGroupName = "<resource group name>"
$pwd = "<create a password>"
$azureAdApplication = Get-AzADApplication -IdentifierUri
"https://localhost/jonw-azure-monitor"
$subscription = Get-AzSubscription -SubscriptionId $subscriptionId
$clientId = $azureAdApplication.ApplicationId.Guid
$tenantId = $subscription.TenantId

$Uri =
"https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token";
$Body = @{
grant_type = "client_credentials"
client_id = $clientId
client_secret = $pwd
scope = 'https://management.azure.com/.default'
redirect_uri = 'https://localhost/'
}
$authResult = Invoke-RestMethod -Method Post -Uri $Uri -ContentType
"application/x-www-form-urlencoded" -Body $body
$accessToken=$authResult.access_token
$authHeader = @{ 'Authorization'="Bearer $accessToken" }
```

List Azure Monitor Custom Namespaces

If you would like to know more about how to use REST API endpoints for getting Azure Monitor namespaces or custom namespaces, please check [here](#).

```
#List AKS metrics custom namespace

$subscriptionId = "<your subscription ID>"
$resourceGroupName = "<resource group name>"
$clusterName = "<cluster name>"

$request =
"https://management.azure.com/subscriptions/$subscriptionId/resource
Groups/$resourceGroupName/providers/Microsoft.ContainerService/manag
edClusters/$clusterName/providers/microsoft.insights/metricNamespac
e?api-version=2017-12-01-preview"

$result = Invoke-RestMethod -Uri $request `
-ContentType application/json `
```

[Open in app](#)

```
Write-Output $result.value.properties.metricNamespaceName
```

The expected result from PowerShell script above should be

```
Insights.Container/nodes
Insights.Container/pods
insights.container/containers
insights.container/persistentvolumes
Microsoft.ContainerService/managedClusters
```

List Azure Monitor Custom Namespace Metrics

There are several parameters you could tweak.

- **timespan:** it means when to when is your target
- **interval:** it means how frequent you would like to get an average, max or min
- **metricnamespace:** reuse the result retrieved from the above script
- **metricnames:** we could get from Azure portal or with this REST API endpoint without specifying one metric name (by default, it would list all metric names under the namespace)
- **aggregation:** this could be average, count, max or min
- **top:** only could be used with count, max or min
- **orderby:** only could be used with count, max or min

```
$subscriptionId = "<your subscription ID>"
$resourceGroupName = "<resource group name>"
$clusterName = "<cluster name>"

$request =
"https://management.azure.com/subscriptions/$subscriptionId/resource
Groups/$resourceGroupName/providers/Microsoft.ContainerService/manag
edClusters/$clusterName/providers/microsoft.insights/metrics?
timespan=2021-03-08T02:20:00Z/2021-03-
08T04:20:00Z&interval=PT5M&metricnamespace=Insights.Container/nodes&
```

[Open in app](#)

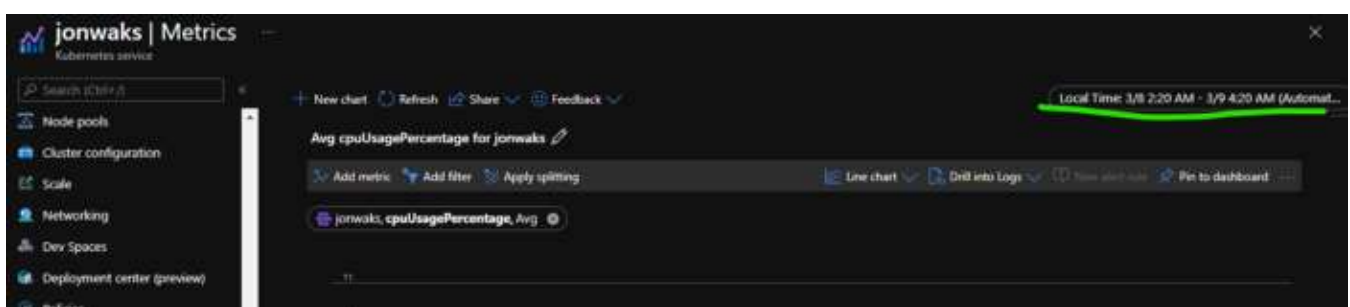

```
$result = Invoke-RestMethod -Uri $request
-ContentType application/json `
-Headers $authHeader `
-Method Get
```

```
Write-Output $result.value.timeseries.data
```

The expected outcome from the PowerShell script above.

timeStamp	average
-----	-----
2021-03-08T02:20:00Z	9.7368010876666666
2021-03-08T02:25:00Z	10.684912887875
2021-03-08T02:30:00Z	9.7218887111249987
2021-03-08T02:35:00Z	9.9403485090833321
2021-03-08T02:40:00Z	9.4864689833333333
2021-03-08T02:45:00Z	10.310866700791667
2021-03-08T02:50:00Z	9.5301839500833339
2021-03-08T02:55:00Z	11.033740746166668
2021-03-08T03:00:00Z	10.359540908416665
2021-03-08T03:05:00Z	9.7360018555416676
2021-03-08T03:10:00Z	9.0768887417916666
2021-03-08T03:15:00Z	9.738803412625
2021-03-08T03:20:00Z	9.3522125777083343
2021-03-08T03:25:00Z	9.8368563266666662
2021-03-08T03:30:00Z	9.352385603375
2021-03-08T03:35:00Z	9.797668792291665
2021-03-08T03:40:00Z	9.1457025600416664
2021-03-08T03:45:00Z	9.6116043901666668
2021-03-08T03:50:00Z	9.085582542458333
2021-03-08T03:55:00Z	9.7894768375
2021-03-08T04:00:00Z	9.2014669925416666
2021-03-08T04:05:00Z	9.741368539583334
2021-03-08T04:10:00Z	9.37893837775
2021-03-08T04:15:00Z	9.741896410625

If you do not believe the result is accurate, you could always double confirm by adjusting the timeline Azure portal visual to the queried timeline and check.





That is it! This would provide you some ideas on how to hit REST API endpoint with all programmatic approach. We would be having other articles about all the OAuth grant Azure AD provides in the future! Happy learning!

Azure Kubernetes Service

Monitoring

Rest Api

Powershell

Script

AboutHelpLegal

Get the Medium app

Download on the App Store

GET IT ON Google Play