+++ title = "Template variables in CloudWatch query" description = "Template variables in CloudWatch queryh" weight = 10 aliases = ["/docs/grafana/latest/datasources/cloudwatch"] +++

Using template variables in CloudWatch queries

Instead of hard-coding server, application, and sensor names in your metric queries, you can use variables. The variables are listed as dropdown select boxes at the top of the dashboard. These dropdowns make it easy to change the display of data in your dashboard.

For an introduction to templating and template variables, refer to the [Templating]({{< relref "../../variables/_index.md" >}}) documentation.

Query variable

The CloudWatch data source provides the following queries that you can specify in the `Query` field in the Variable edit view. They allow you to fill a variable's options list with things like `region`, `namespaces`, `metric names` and `dimension keys/values`.

In place of `region` you can specify `default` to use the default region configured in the data source for the query, e.g. `metrics(AWS/DynamoDB, default)` or `dimension_values(default, ..., ..., ...)`.

Read more about the available dimensions in the CloudWatch Metrics and Dimensions Reference.

Name	Description
`regions()`	Returns a list of all AWS regions
`namespaces()`	Returns a list of namespaces CloudWatch support.
`metrics(namespace, [region])`	Returns a list of metrics in the namespace. (specify region or use "default" for custom metrics)
`dimension_keys(namespace)`	Returns a list of dimension keys in the namespace.
`dimension_values(region, namespace, metric, dimension_key, [filters])`	Returns a list of dimension values matching the specified `region`, `namespace`, `metric`, `dimension_key` or you can use dimension `filters` to get more specific result as well.
`ebs_volume_ids(region, instance_id)`	Returns a list of volume ids matching the specified `region`, `instance_id`.
`ec2_instance_attribute(region, attribute_name, filters)`	Returns a list of attributes matching the specified `region`, `attribute_name`, `filters`.
`resource_arns(region, resource_type, tags)`	Returns a list of ARNs matching the specified `region`, `resource_type` and `tags`.
`statistics()`	Returns a list of all the standard statistics

For details about the metrics CloudWatch provides, please refer to the CloudWatch documentation.

Example of templated queries

Here is an example of the dimension queries which will return list of resources for individual AWS Services:

Query	Service
`dimension_values(us-east-1,AWS/ELB,RequestCount,LoadBalancerName)`	ELB
`dimension_values(us-east-1,AWS/ElastiCache,CPUUtilization,CacheClusterId)`	ElastiCache
`dimension_values(us-east-1,AWS/Redshift,CPUUtilization,ClusterIdentifier)`	RedShift
`dimension_values(us-east-1,AWS/RDS,CPUUtilization,DBInstanceIdentifier)`	RDS
`dimension_values(us-east-1,AWS/S3,BucketSizeBytes,BucketName)`	S3
`dimension_values(us-east-1,CWAgent,disk_used_percent,device,{"InstanceId":"$instance_id"})`	CloudWatch Agent
`resource_arns(eu-west-1,elasticloadbalancing:loadbalancer,{"elasticbeanstalk:environment-name":["myApp-dev","myApp-prod"]})`	ELB
`resource_arns(eu-west-1,elasticloadbalancing:loadbalancer,{"Component":["$service"],"Environment":["$environment"]})`	ELB
`resource_arns(eu-west-1,ec2:instance,{"elasticbeanstalk:environment-name":["myApp-dev","myApp-prod"]})`	EC2

Using JSON format template variables

Some queries accept filters in JSON format and Grafana supports the conversion of template variables to JSON.

If `env = 'production', 'staging'`, following query will return ARNs of EC2 instances which `Environment` tag is `production` or `staging`.

```
resource_arns(us-east-1, ec2:instance, {"Environment":${env:json}})
```

ec2_instance_attribute examples

JSON filters

The `ec2_instance_attribute` query takes `filters` in JSON format. You can specify pre-defined filters of ec2:DescribeInstances. Note that the actual filtering takes place on Amazon's servers, not in Grafana.

Filters syntax:

```
{ "filter_name1": [ "filter_value1" ], "filter_name2": [ "filter_value2" ] }
```

Example `ec2_instance_attribute()` query

```
ec2_instance_attribute(us - east - 1, InstanceId, { 'tag:Environment':
['production'] });
```

Selecting attributes

Only 1 attribute per instance can be returned. Any flat attribute can be selected (i.e. if the attribute has a single value and isn't an object or array). Below is a list of available flat attributes:

- `AmiLaunchIndex`
- `Architecture`
- `ClientToken`
- `EbsOptimized`
- `EnaSupport`
- `Hypervisor`
- `IamInstanceProfile`
- `ImageId`
- `InstanceId`
- `InstanceLifecycle`
- `InstanceType`
- `KernelId`
- `KeyName`
- `LaunchTime`
- `Platform`
- `PrivateDnsName`
- `PrivateIpAddress`
- `PublicDnsName`
- `PublicIpAddress`
- `RamdiskId`
- `RootDeviceName`
- `RootDeviceType`
- `SourceDestCheck`
- `SpotInstanceRequestId`
- `SriovNetSupport`
- `SubnetId`
- `VirtualizationType`
- `VpcId`

Tags can be selected by prepending the tag name with `Tags.`

Example `ec2_instance_attribute()` query

```
ec2_instance_attribute(us - east - 1, Tags.Name, { 'tag:Team': ['sysops'] });
```