

Prisma™ Cloud Resource Query Language (RQL) Reference

Contact Information

Corporate Headquarters:

Palo Alto Networks

3000 Tannery Way

Santa Clara, CA 95054

www.paloaltonetworks.com/company/contact-support

About the Documentation

- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal www.paloaltonetworks.com/documentation.
- To search for a specific topic, go to our search page www.paloaltonetworks.com/documentation/document-search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Palo Alto Networks, Inc.

www.paloaltonetworks.com

© 2019-2019 Palo Alto Networks, Inc. Palo Alto Networks is a registered trademark of Palo Alto Networks. A list of our trademarks can be found at www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

Last Revised

September 17, 2019

Table of Contents

Prisma Cloud RQL Reference.....	5
Prisma Cloud Resource Query Language (RQL).....	7
Config Query.....	8
Config Query Attributes.....	8
Config Query Examples.....	12
Event Query.....	16
Event Query Attributes.....	16
Event Query Examples.....	19
Network Query.....	22
Network Query Attributes.....	22
Network Query Examples.....	28
AWS APIs Ingested by Prisma Cloud.....	29
GCP APIs Ingested by Prisma Cloud.....	33
Microsoft Azure APIs Ingested by Prisma Cloud.....	35
RQL Operators.....	37
Operators Within JSON Arrays.....	37
Config and Event Operators.....	37
Joins.....	45
RQL FAQs.....	50
RQL Example Library.....	53
AWS Examples.....	53
Azure Examples.....	58
GCP Examples.....	60
Common Useful Query Examples.....	61

Prisma Cloud RQL Reference

Use the Prisma Cloud Resource Query Language (RQL) to perform configuration checks on resources deployed on different cloud platforms and to gain visibility and insights into user and network events.

- > [Prisma Cloud Resource Query Language \(RQL\)](#)
- > [Config Query](#)
- > [Event Query](#)
- > [Network Query](#)
- > [AWS APIs Ingested by Prisma Cloud](#)
- > [GCP APIs Ingested by Prisma Cloud](#)
- > [Microsoft Azure APIs Ingested by Prisma Cloud](#)
- > [RQL Operators](#)
- > [RQL FAQs](#)
- > [RQL Example Library](#)

Prisma Cloud Resource Query Language (RQL)

Prisma Cloud Resource Query Language (RQL) is a powerful and flexible tool that helps you gain security and operational insights about your deployments in public cloud environments. You can use RQL to perform configuration checks on resources deployed on different cloud platforms and to gain visibility and insights into user and network events. You can use these security insights to create policy guardrails that secure your cloud environments.

RQL is a structured query language that resembles Structured Query Language (SQL). RQL supports the following types of queries:

- **Config**—Use [Config Query](#) to search for the configuration of the cloud resources.
- **Event**—Use [Event Query](#) to search and audit all the console and API access events in your cloud environment.
- **Network**—Use [Network Query](#) to search real-time network events in your environment.

Use RQL to find answers to fundamental questions that help you understand what is happening on your network. For example, you can find answers to the following questions:

- Do I have S3 buckets with encryption disabled?
- Do I have databases that are directly accessible from the internet?
- Who uses a root account to manage day-to-day administrative activities on my network?
- Which cloud resources are missing critical patches that make them exploitable?

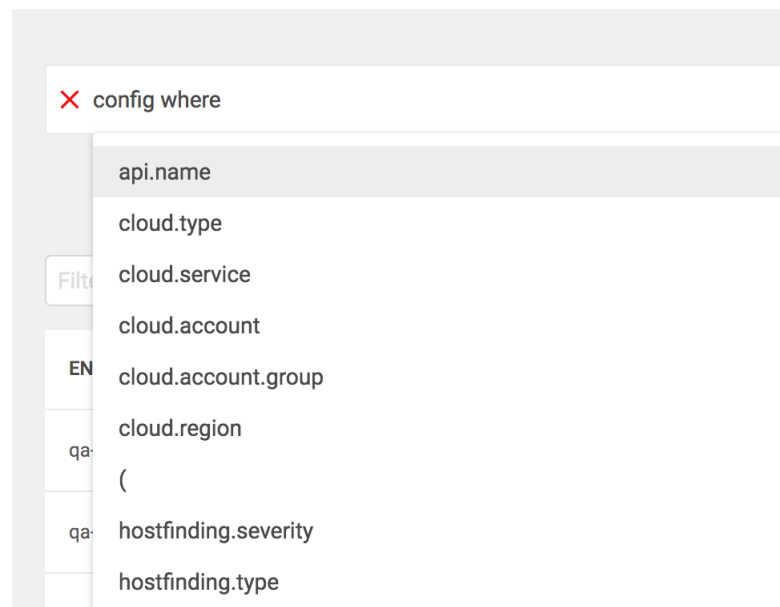
Config Query

Prisma Cloud ingests various services and associated configuration data from AWS, Azure, and GCP cloud environments. Use Config Query to retrieve resource information and identify misconfigurations, compliance violations, and cloud identity, access, and password management policies. To investigate configuration issues, use the **config where** query in the search box on the **Investigate** tab of the Prisma Cloud admin console.

- [Config Query Attributes](#)
- [Config Query Examples](#)

Config Query Attributes

Review your options when using **config where** on the **Investigate** tab of the Prisma Cloud administrative console.



Each attribute allows you to narrow your search criteria. The auto-suggest feature displays expressions and [Operators](#) available for each attribute.

- **api.name**

Cloud APIs are part of cloud platforms and they enable the development of applications and services used for provisioning resources, virtual machines, platforms, and software. Cloud APIs are generally based on a REST framework.

For each cloud platform, depending on the resource, there are several APIs available. You can use the **api.name** attribute to identify a specific configuration for the resource. For a list of all API names available for each cloud platform, see [AWS APIs Ingested by Prisma Cloud](#), [Microsoft Azure APIs Ingested by Prisma Cloud](#), and [GCP APIs Ingested by Prisma Cloud](#).

The **api.name** attribute is required in configuration queries except when you are querying the configuration for [host findings](#).

When used with the [cloud.type](#) attribute, auto-complete displays only the API names that pertain to the cloud type you selected.


```
config where cloud.type = 'gcp' AND api.name = |
```

```
'gcloud-compute-interfaces-list'  
'gcloud-compute-networks-list'  
'gcloud-compute-networks-subnets-list'  
'gcloud-iam-service-accounts-keys-list'  
'gcloud-iam-service-accounts-list'  
'gcloud-projects-get-iam-policy'  
'gcloud-projects-get-iam-user'  
'gcloud-sql-instances-list'  
'gcloud-storage-buckets-list'
```

For example, you can list SQL instances on Google Cloud:

```
config where cloud.type = 'gcp' AND api.name = 'gcloud-sql-instances-list'
```

- **addcolumn**

Use the **addcolumn** attribute to add columns to the results displayed on screen. This enables you to view the JSON data for the resources that correspond to your query.

```
config where api.name = 'aws-cloudwatch-describe-alarms' addcolumn |
```

```
actionsEnabled  
alarmActions[*]  
alarmArn  
alarmConfigurationUpdatedTimestamp  
alarmDescription  
alarmName  
comparisonOperator  
datapointsToAlarm  
dimensions[*]
```

For example, you can add columns for key name and image ID for EC2 instances:

```
config where api.name = 'aws-ec2-describe-instances' addcolumn keyName  
hypervisor imageId
```

- **cloud.type**

Use the **cloud.type** attribute to narrow down your search option to specific clouds. Supported options are AWS, Azure, and GCP.

For example, you can list all EC2 instances in your AWS cloud accounts:

```
config where cloud.type = 'aws' AND api.name = 'aws-ec2-describe-instances'
```

- **cloud.service**

Use the **cloud.service** attribute to query configuration for a specific cloud service, such as IAM, S3, or Virtual Machines.

For example, you can list all S3 storage bucket access control lists (ACLs) in your AWS cloud accounts:

```
config where cloud.type = 'aws' AND cloud.service = 'S3' AND api.name = 'aws-s3api-get-bucketacl'
```

- **cloud.account**

Use the **cloud.account** attribute to narrow down a configuration search to one or more cloud accounts that you connected to the Prisma Cloud.

For example, you can list EC2 instances in your Production AWS account:

```
config where cloud.type = 'aws' AND cloud.account = 'Production' AND api.name = 'aws-ec2-describe-instances'
```

- **cloud.region**

Use the **cloud.region** attribute to narrow down a configuration search to one or more cloud regions.

For example, you can list all virtual machine instances in your Azure account in the Central US region:

```
config where cloud.type = 'azure' and cloud.account = 'RedLock - Azure Subscription' AND cloud.region = 'Azure Central US' AND api.name = 'azure-vm-list'
```

- **cloud.account.group**

Use the **cloud.account.group** attribute to narrow down the configuration to the cloud account in your cloud account group.

For example, you can list all the Amazon RDS instances in all your AWS accounts:

```
config where cloud.account.group = 'All my AWS accounts' AND cloud.region = 'AWS Virginia' AND api.name = 'aws-rds-describe-db-instances'
```

- **hostfinding.type, hostfinding.severity, hostfinding.source**

Use host finding attributes to query for vulnerabilities on workloads—destination or source resources—that have one or more host-related security findings. Prisma Cloud ingests host vulnerability data from external sources, such as Qualys, Tenable.io, and AWS Inspector and ingests host and IAM users security-related alerts from AWS GuardDuty.



*To leverage **hostfinding** attributes, first enable the integration with the host vulnerability providers.*

```
config where hostfinding.type =|
```

```
'RedLock'  
'Host Vulnerability'  
'Compliance'  
'AWS Inspector Runtime Behavior Analysis'  
'AWS Inspector Security Best Practices'  
'AWS GuardDuty Host'
```

For example, you can list all the hosts with a critical host vulnerability:

```
config where hostfinding.type = 'Host Vulnerability' AND  
hostfinding.severity = 'critical'
```

Or find potential security issues by source:

```
config where hostfinding.source = 'AWS Guard Duty' AND hostfinding.type =  
'AWS GuardDuty IAM ' AND api.name= 'aws-iam-list-users'
```

Host finding attributes support the following resource types:

- **Prisma Cloud**—Fetches all resources that have one or more open alerts generated by Prisma Cloud.
- **Host Vulnerability**—Fetches all resources that have one or more of the host vulnerabilities (such as CVE-2016-8655) reported by external providers such as AWS Inspector, Qualys, or Tenable.io.
- **Compliance**—Fetches all resources that are in violation of one or more compliance issues reported by external compliance host-scanning systems.
- **AWS Inspector Runtime Behavior Analysis**—Fetches all resources which are in violation of one or more rules reported by the AWS Runtime Behavior Analysis package.
- **AWS Inspector Security Best Practices**—Fetches all resources which are in violation of one or more rules reported by the AWS Inspector Security best practices package.
- **AWS GuardDuty**—Fetches all resources which have one or more findings reported by AWS GuardDuty.
- **hostfinding.name**

Use the **hostfinding.name** attribute and enter a string value to find a host vulnerability by the name defined on your host vulnerability provider. Specify the **hostfinding.type** for the autocomplete suggestion to specify a **hostfinding name**.

config where hostfinding.type = 'Host Vulnerability' AND

```
api.name  
cloud.type  
cloud.service  
cloud.account  
cloud.region  
(  
hostfinding.severity  
hostfinding.name
```

For example, you can list all the hosts with the CVE-2016-8399 vulnerability:

```
config where hostfinding.type = 'Host Vulnerability' AND hostfinding.name =  
'CVE-2016-8399'
```

or,

```
config where hostfinding.type = 'AWS GuardDuty IAM' AND hostfinding.name=  
'Recon:IAM/TorIPCaller'
```

- **json.rule**

Prisma Cloud ingests data and updates events in the JSON format.

Use the **json.rule** attribute to query or filter specific elements included in the JSON configuration related to a cloud resource. The **json.rule** attribute enables you to look for specific configurations: parse JSON-encoded values, extract data from JSON, or search for value within any configuration policy for cloud accounts that you are monitoring using Prisma Cloud. This **json.rule** attribute allows you to create boolean combinations and find data in selected fields within the JSON data that represents the resource.

When you include the **json.rule** attribute in a configuration query, the auto-complete displays the elements or resources that match your search criteria. Because JSON has a nested structure, you can search for elements at the root level, inside the JSON tree, or in an array object.

For example, you can list all Azure Linux Virtual Machines where password authentication is disabled:

```
config where api.name = 'azure-vm-list' AND json.rule =  
['properties.osProfile'].linuxConfiguration.disablePasswordAuthentication  
is true
```

Config Query Examples

Use this section for some examples that show you how to use [Config Query Attributes](#) in RQL for investigating issues on each cloud platform:

- [AWS—Config Query Examples](#)
- [Azure—Config Query Examples](#)
- [GCP—Config Query Examples](#)

AWS—Config Query Examples

DESCRIPTION	RQL
View users who enabled console access with both access keys and passwords.	<pre>config where api.name = 'aws-iam-get-credential-report' AND json.rule = access_key_1_active is true or access_key_2_active is true and password_enabled equals true</pre>
List root accounts that do not have MFA enabled.	<pre>config where api.name = 'aws-iam-get-account-summary' AND json.rule='not AccountMFAEnabled equals 1'</pre>
List active access keys.	<pre>config where api.name = 'aws-iam-list-access-keys' AND json.rule = status equals Active</pre>
View all S3 buckets that are accessible to the public through bucket ACLs.	<pre>config where api.name='aws-s3api-get-bucket-acl' AND json.rule="(acl.grants[? (@.grantee=='AllUsers')]] size > 0)"</pre>
View all S3 buckets that are accessible to the public through bucket policy.	<pre>config where api.name = 'aws-s3api-get-bucket-acl' and json.rule = "policy.Statement exists and policy.Statement[? (@.Action=='s3:GetObject' && @.Effect=='Allow')].Principal contains *"</pre>
Check for S3:GetObject operations.	<p>You can include other operations related to S3 buckets, such as s3:PutObject, s3:*, s3:GetBucketAcl, s3:ListBucket, s3:ListAllMyBuckets, s3:PutObjectAcl, s3:GetObjectAcl, and s3:GetObjectVersion.</p> <pre>config where api.name = 'aws-s3api-get-bucket-acl' and json.rule = "policy.Statement exists and policy.Statement[? (@.Action=='s3:GetObject' && @.Effect=='Allow' @.Action=='s3:ListBucket' && @.Effect=='Allow' @.Action=='s3:*' && @.Effect=='Allow' @.Action=='s3:GetBucketAcl' && @.Effect=='Allow' @.Action=='s3:PutObject' && @.Effect=='Allow' @.Action=='s3:GetObjectAcl'</pre>

DESCRIPTION	RQL
	<pre> && @.Effect=='Allow' @.Action=='s3:GetObjectVersion' && @.Effect=='Allow')].Principal contains *</pre>

Azure—Config Query Examples

DESCRIPTION	RQL
View SQL Server firewall rules that allow access to any Azure internal resources.	<pre> config where api.name = 'azure-sql-server-list' AND json.rule = "firewallRules[*] size > 0 and firewallRules[*].['endIpAddress'] contains 0.0.0.0 and firewallRules. [*].['startIpAddress'] contains 0.0.0.0"</pre>
List security center resource groups in Azure that do not specify a security contact email address.	<pre> config where cloud.type = 'azure' AND api.name = 'azure- security-center' AND json.rule = 'name == default and (properties.securityContactConfiguration.securi isEmpty or properties.securityContactConfiguration exists)'</pre>
View SQL databases where encryption is disabled.	<pre> config where api.name = 'azure-sql-db-list' AND json.rule='transparentDataEncryption is false'</pre>
List Azure VNETs that are peered successfully.	<pre> config where cloud.type = 'azure' AND api.name = 'azure-network- vnet-list' AND json.rule = " ['properties.virtualNetworkPeerings'] [*]. ['properties.provisioningState'] contains Succeeded "</pre>

GCP—Config Query Examples

DESCRIPTION	RQL
View firewall rules that allow internet traffic through the MongoDB port (27017).	<pre> config where api.name='gcloud- compute-firewall-rules-list' AND json.rule='sourceRanges[*] contains</pre>

DESCRIPTION	RQL
	<code>0.0.0.0/0 and allowed[*].ports[*] == 27017'</code>
List SQL Instances where SSL is not configured.	<code>config where api.name='gcloud-sql-instances-list' and json.rule = 'settings.ipConfiguration.requireSsl is true'</code>
List virtual machine (VM) instances where preemptive termination is enabled.	<code>config where api.name = 'gcloud-compute-instances-list' AND json.rule = 'scheduling.preemptible is true'</code>
View all storage buckets or objects that are publicly accessible.	<code>config where cloud.type = 'gcp' AND cloud.service = 'Google Cloud Storage' AND api.name = 'gcloud-storage-buckets-list' AND json.rule = 'acl[*].entity contains allUsers or acl[*].entity contains allAuthenticatedUsers'</code>

Event Query

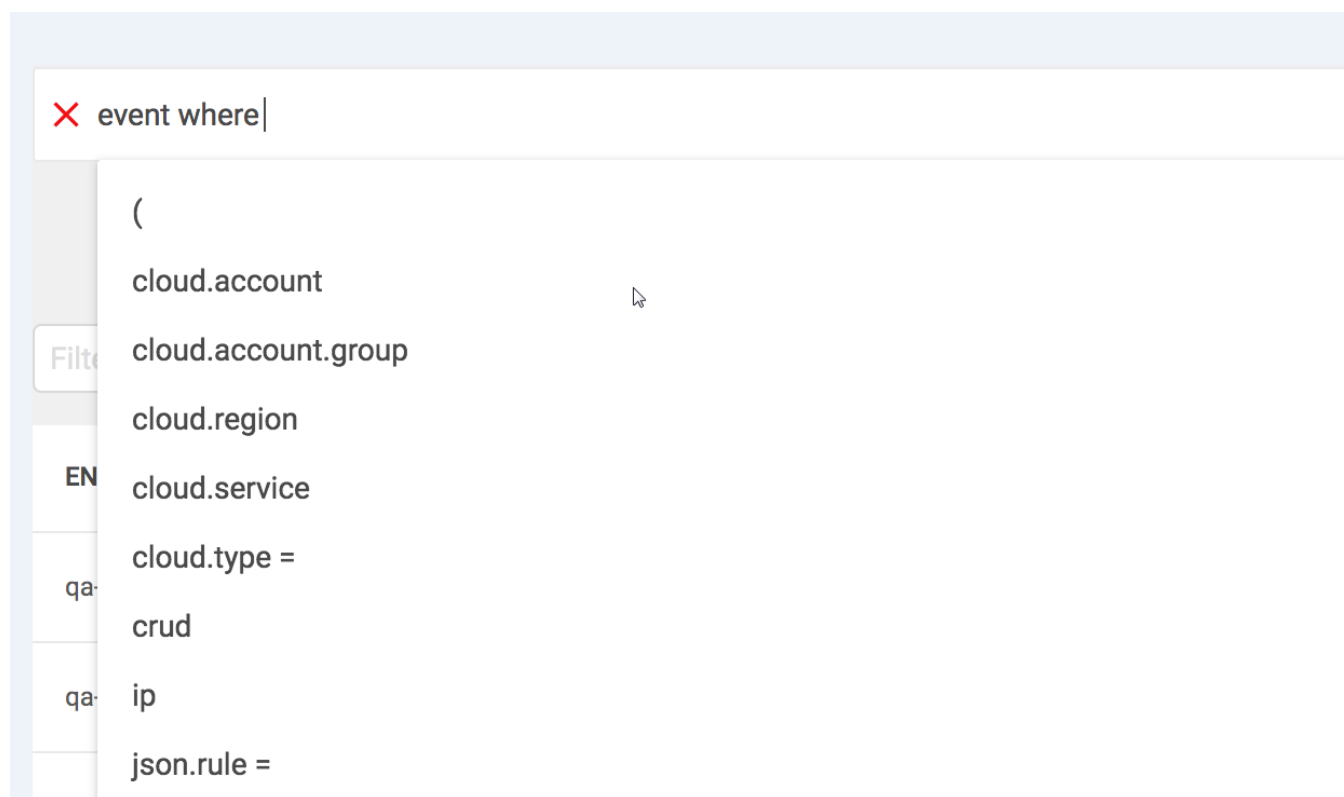
Event queries help you to detect and investigate console and API access events, monitor privileged activities, detect account compromise, and detect unusual user behavior in your cloud environments.

To investigate events, compose **event where** queries in the search box on the **Investigate** tab of the Prisma Cloud administrative console.

- [Event Query Attributes](#)
- [Event Query Examples](#)

Event Query Attributes

Review your options when using **event where** on the **Investigate** tab of the Prisma Cloud administrative console:



Each attribute allows you to narrow your search criteria. As you use these attributes, the auto-suggestion feature shows the available expressions and the [Operators](#) that are applicable for each attribute.

- **alert.id**

Use the **alert.id** attribute to view alert details on the **Investigate** tab.

For example, you can visualize the alert details for a set of alerts such as P-8444, P-8421, P-8420.

```
event where alert.id IN ('P-8444', 'P-8421', 'P-8420')
```

- **cloud.account**

Use the **cloud.account** attribute to narrow down audit search to one or more cloud accounts that you connected to Prisma Cloud.

For example, you can list entities or users who deleted security groups from a given cloud account:

```
event where cloud.account = 'Developer Sandbox' AND operation IN  
( 'DeleteSecurityGroup' )
```

- **cloud.account.group**

Use the **cloud.account.group** attribute to narrow your search to only the cloud accounts in your cloud account group.

For example, you can list entities or users who deleted Virtual Private Clouds in all your AWS accounts:

```
event where operation = 'DeleteVpc' AND cloud.account.group = 'All my AWS  
accounts'
```

```
event where cloud.account.group = 'All my AWS accounts' AND cloud.service =  
'autoscaling.amazonaws.com' AND user = 'maxusertest__gahplTho'
```

- **cloud.type**

Use the **cloud.type** attribute to narrow your search to a specific cloud platform. Supported options are AWS, Azure, and GCP.

For example, you can list all users who deleted S3 buckets:

```
event where cloud.type = 'aws' AND cloud.service = 's3.amazonaws.com' AND  
operation = 'DeleteBucket'
```

- **cloud.region**

Use the **cloud.region** attribute to narrow down audit search to one or more cloud regions.

For example, you can list entities or users who deleted access keys from a given cloud account:

```
event where cloud.account = 'Developer Sandbox' AND cloud.region = 'AWS  
Canada' AND operation IN ( 'DeleteAccessKey' )
```

- **cloud.service**

Use the **cloud.service** attribute to search for information using a specific service name in your cloud accounts.

```
event where cloud.service =
```

```
'acm.amazonaws.com'  
'apigateway.amazonaws.com'  
'appengine.googleapis.com'  
'audited_resource'  
'autoscaling.amazonaws.com'  
'bigquery.googleapis.com'  
'bigtableadmin.googleapis.com'  
'clientauthconfig.googleapis.com'  
'cloudbilling.googleapis.com'
```

For example, you can review details for users who performed operations, such as deleting cloud trail logs or disabling or stopping logging events:

```
event where system = 'cloudtrail.amazonaws.com' AND operation IN  
( 'DeleteTrail' , 'DisableLogging' , 'StopLogging' )
```

- **crud**

Use the **crud** attribute to search for information for users or entities who performed Create, Read, Update, or Delete operations.

You can list all Azure resources that were deleted:

```
event where cloud.account in ( 'Azure - Microsoft Azure Sponsorship' ) and  
crud = 'delete'
```

- **operation**

An operation is an action performed by users on resources in a cloud account. Use the **operation** attribute to start typing the name of the operation in which you are interested and Prisma Cloud auto-completes a list of operations that match your search criteria.

For example, you can list details of delete operations on VPCs, VPC endpoints, and VPC peering connections:

```
event where operation in ( 'DeleteVpc' , 'DeleteVpcEndpoints'  
'DeleteVpcPeeringConnection' )
```

- **subject**

Use this attribute to search for actions that a specific user or an instance performed on your cloud account.

For example, you can list console login operations by Ben:

```
event where operation = 'ConsoleLogin' AND subject = 'ben'
```

- **role**

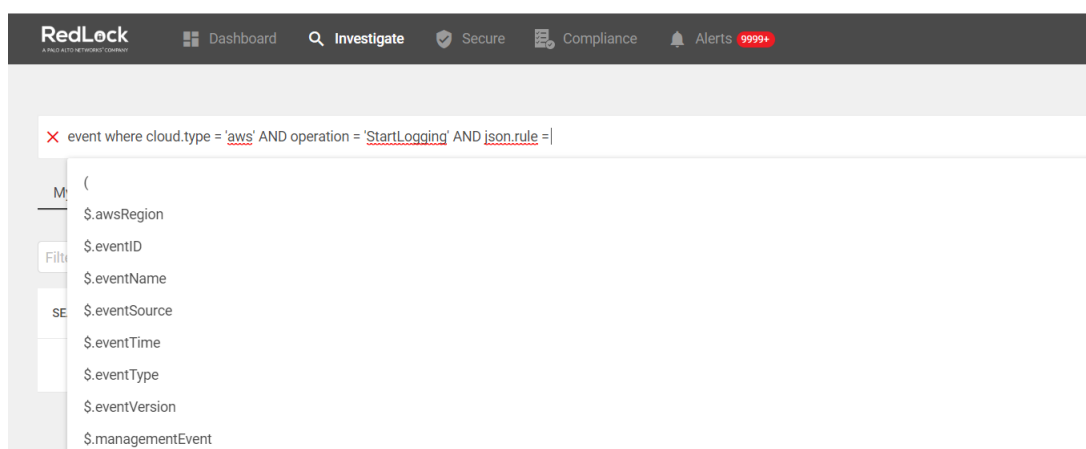
Use this attribute to filter the search results by role.

For example, you can look for events performed by the Okta role:


```
Event where role = 'OktaDevReadWriteRole'
```

- **json.rule**

Use this attribute to filter specific elements included in the JSON configuration related to a cloud resource. The **json.rule** attribute enables you to look for specific configurations—parse JSON-encoded values, extract data from JSON, search for value within any configuration policy for cloud accounts that you are monitoring using Prisma Cloud.



Use the automatic suggest feature to see the available values for `json.rule`.

 The auto suggest works with the operators `=` and `IN`. It is not supported for array objects.
Use `cloud.type` attribute to refine the search results.

For example, you can check for login failures on the console:

```
event where cloud.account = 'Sandbox' AND json.rule =
$.responseElements.ConsoleLogin != 'Success'
```

Event Query Examples

Use this section to review examples that show you how to use [Event Query Attributes](#) in RQL for investigating issues on each cloud platform:

- [AWS—Event Query Examples](#)
- [Azure—Event Query Examples](#)
- [GCP—Event Query Examples](#)

AWS—Event Query Examples

DESCRIPTION	RQL
Detect activities from non-automated events and from specific IP addresses.	<pre>event where ip EXISTS AND ip IN (152.1.1.1)</pre>
Detect potentially sensitive or suspicious changes to the network configuration that impact your Security posture.	<pre>event where operation IN ('AuthorizeSecurityGroupEgress', 'AuthorizeSecurityGroupIngress', 'CreateVpc', 'DeleteFlowLogs', 'DeleteVpc', 'ModifyVpcAttribute', 'RevokeSecurityGroupIngress')</pre>
Detect potentially sensitive or suspicious changes to configuration settings.	<pre>event where operation IN ('DeleteBucket', 'DeleteConfigRule', 'DeleteTrail',</pre>

DESCRIPTION	RQL
	<pre>'PutBucketAcl', 'PutBucketLogging', 'PutBucketPolicy') Sensitive Activities by User where operation IN ('AddUserToGroup', 'AttachGroupPolicy', 'AttachGroupPolicy', 'AttachUserPolicy' , 'AttachRolePolicy' , 'CreateAccessKey', 'CreateKeyPair', 'DeleteKeyPair', 'DeleteLogGroup')</pre>
Detect risky changes executed by a root user.	<pre>event where operation IN ('ChangePassword', 'ConsoleLogin', 'DeactivateMFADevice', 'DeleteAccessKey' , 'DeleteAlarms') AND user = 'root'</pre>

Azure—Event Query Examples

DESCRIPTION	RQL
List specific operations performed on a specific Microsoft Azure account.	<pre>event where cloud.account = 'RedLock - Azure Subscription' AND operation IN ('AttachRolePolicy', 'AttachLoadBalancers')</pre>
List Classic compute register operations performed by a specific user on a specific Microsoft Azure account.	<pre>event where cloud.account in ('RedLock - Azure Subscription') and user = 'abc@redlock.io' and operation IN ('Microsoft.ClassicCompute/ register/action (BeginRequest)')</pre>

GCP—Event Query Examples

DESCRIPTION	RQL
View sensitive network configuration updates on GCP	<pre>event where operation IN ('v1.compute.networks.delete', 'beta.compute.net</pre>
View sensitive SQL instance updates in GCP.	<pre>event where operation IN ('cloudsql.instances.update', 'cloudsql.sslCerts.create',</pre>

DESCRIPTION	RQL
	<code>cloudsql.instances.create', 'cloudsql.instances.delete')</code>
List all events with sensitive user actions on GCP.	<code>event where operation IN ('CreateCryptoKey', 'DestroyCryptoKeyVersion', 'v</code>

Network Query

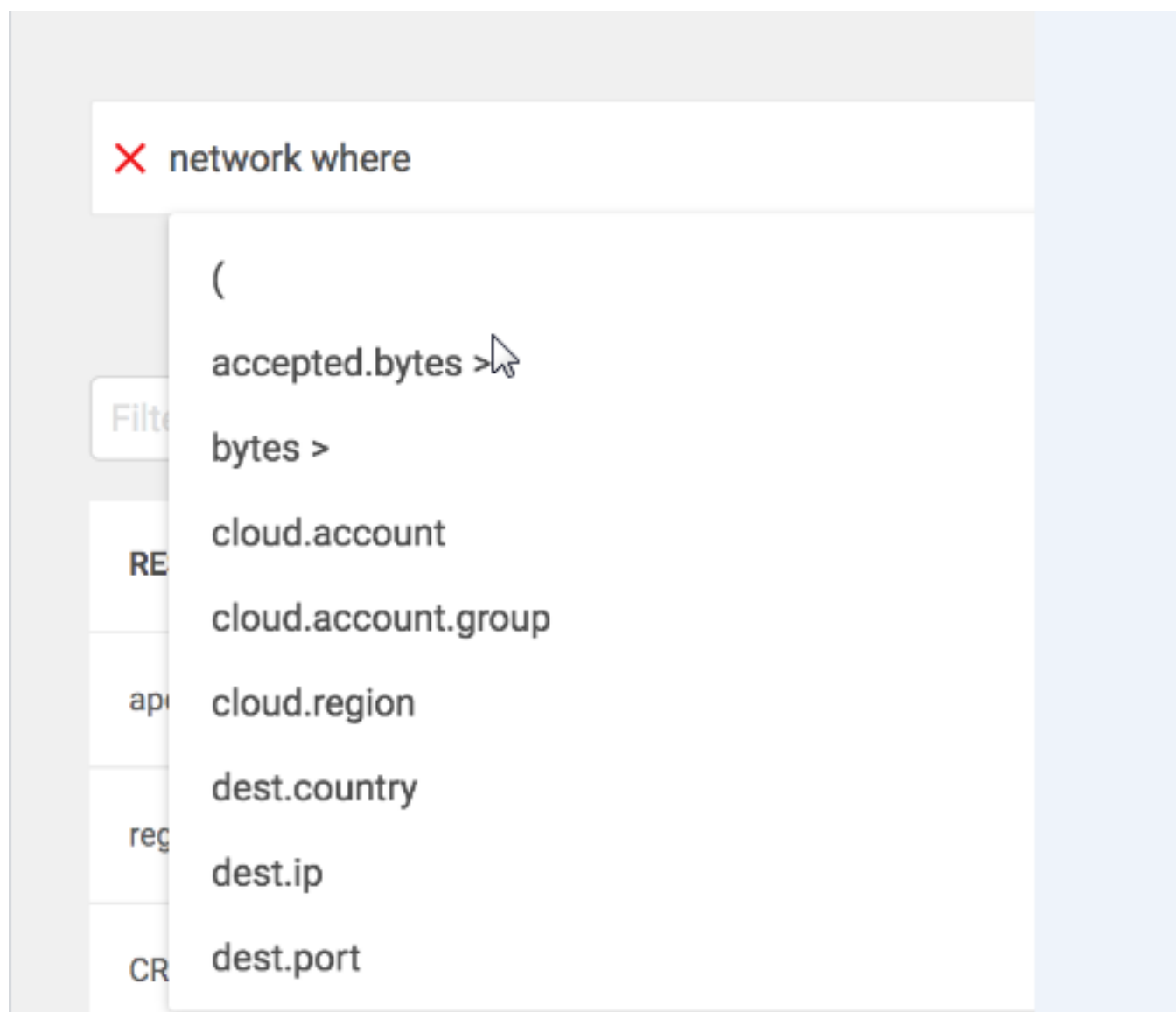
When you onboard your cloud accounts to Prisma Cloud, it monitors network traffic from and to your assets deployed on the cloud environment, and you can then use this data to find previously unidentified network security risks. With Network queries, **network where**, you can for example, detect when services, applications or databases are exposed to the Internet and fix risky configuration issues, or search for assets that are receiving traffic and connections from suspicious IP addresses to prevent data exfiltration attempts before it is too late.

Network queries are currently supported on AWS and Azure cloud environments. GCP is in beta.

- [Network Query Attributes](#)
- [Network Query Examples](#)

Network Query Attributes

Review your options when using **Network where** on the **Investigate** tab of the RedLock administrative console:



Each attribute allows you to narrow your search criteria. As you use these attributes, the auto-suggestion capability shows the available expressions, and [Operators](#) that are applicable for each attribute.

- **cloud.account**

Use the **cloud.account** attribute to search for network activity in one or more cloud accounts that you connected to the RedLock service.

For example, you can view network activity in a cloud account with > 1MB traffic:

```
network where cloud.account = 'Developer Sandbox' AND bytes > 1048576
```

- **cloud.region**

Use the **cloud.region** attribute to search for network activity in your cloud regions.

For example, you can view network activity in Developer sandbox account for AWS Oregon region:

```
network where cloud.account = 'Developer Sandbox' AND cloud.region = 'AWS Oregon' AND bytes > 0
```

- **cloud.account.group**

Use the **cloud.account.group** attribute to search for network activity within a group of cloud accounts that you have connected to the RedLock service.

For example, you can view network activity across your AWS accounts that belong to the Oregon region where more than 100000 packets were transmitted:

```
network where cloud.account.group = 'All my AWS accounts' AND cloud.region = 'AWS Oregon' AND packets > 100000
```

- **dest.ip, source.ip**

Use the **dest.ip**, **source.ip** attribute to filter your network to view traffic from an originating or a receiving IP address.



The value 0.0.0.0 does not mean any IP address, it means any public IP address.

For example, you can view network traffic to a public IP address to which more than 1000000 bytes were transmitted:

```
network where dest.ip = 0.0.0.0 AND bytes > 1000000
```

For example, you can view SSH traffic from any public IP address on the internet:

```
network where source.ip = 0.0.0.0 and dest.port = 22
```

- **dest.port**

Use the **dest.port** attribute to filter your network activity to view traffic from a destination port.

For example, you can view network traffic for any public IP address where the destination port is 27017:

```
network where dest.port = 27017 AND source.ip = 0.0.0.0
```

- **dest.outboundpeers, source.outboundpeers**

Use the **dest.outboundpeers** and **source.outboundpeers** attributes for a count of distinct IP addresses to which this asset establishes a connection. These network attributes enable you to aggregate connection counts for both ingress and egress traffic to help detect account compromise or identify hosts that are establishing multiple SSH connections from one or more external IP addresses.

- **dest.outboundports, source.outboundports**

Use the **dest.outboundports** and **source.outboundports** attributes for a count of distinct destination ports to which this asset establishes a connection. These network attributes enable you to aggregate connection counts for both ingress and egress traffic. For example, you can detect an attempt to perform a port scan or port sweep, or detect an attempt to set up a number of egress connections on the crypto ports.

- **dest.publicnetwork, source.publicnetwork**

Use the **source.publicnetwork** and **dest.publicnetwork** attributes to query traffic from and to pre-defined networks. For example, **Internet IPs** represent all public IPs, **Suspicious IPs** represent all suspicious IPs.



You can also define your own network with a set of IP addresses/CIDRs to see traffic from/to your internal networks. If you belong to the System Admin permission group, you can set it up in Settings > IP Whitelisting.

For example, you can view traffic on the destination port 3389 and that are classified as internet IPs or suspicious IPs:

```
network where dest.port IN (3389) and dest.publicnetwork IN ( 'Internet  
IPs' , 'Suspicious IPs' ) and bytes > 0
```

- **dest.resource, source.resource**

Use the **dest.resource, source.resource** attributes to search and filter the network by a destination or a source resource for host findings, roles, security groups, tags, and virtual networks.

```
dest.resource IN or source.resource IN;
```

displays more options:

```
network where dest.resource IN ( resource where  
(  
  hostfinding.severity  
  hostfinding.type  
  role  
  securitygroup.name  
  tag (  
  virtualnetwork.name
```

- **hostfinding.severity, hostfinding.type, hostfinding.source**

Use hostfinding attributes to query for vulnerabilities on destination or source resources that have one or more host related security findings. RedLock ingests host vulnerability data from external sources such as Qualys, Tenable.io, AWS Inspector, and host security related alerts from AWS GuardDuty.



*To leverage **hostfinding** attributes, first enable the integration with the host vulnerability providers.*

For example, you can list hostfinding events from AWS Guard duty on destination resource which have severity as critical:

```
network where dest.resource IN ( resource where hostfinding.type = 'AWS  
GuardDuty Host' AND hostfinding.severity = 'critical' ) AND bytes > 0
```

For example, you can list host vulnerability events on the destination resource:

```
network where dest.resource IN ( resource where hostfinding.type IN ( 'Host  
Vulnerability' ) ) and bytes > 0
```

- **securitygroup.name**

Use the **securitygroup.name** attribute to filter the network traffic by security group name.

For example, you can view the network traffic which is hitting the security groups with names AWS-OpsWorks-Java-App-Server and AWS-OpsWorks-Blank-Server:

```
network where source.ip = 0.0.0.0 and dest.resource IN ( resource where
  securitygroup.name IN ( 'AWS-OpsWorks-Java-App-Server' , 'AWS-OpsWorks-
  Blank-Server' ))
```

- **virtualnetwork.name**

Use the **virtualnetwork.name** attribute to filter the network traffic by virtual network names.

For example, you can view the network traffic which is hitting the virtual network ICHS_FLORENCE:

```
network where dest.resource IN ( resource where virtualnetwork.name IN
  ( 'ICHS_FLORENCE' ))
```

- **dest.state, source.state**

Use **dest.state** or **source.state** attributes to view traffic originating from or destined to a specific state within a country.

For example, you can view network traffic to Karnataka in India:

```
network where cloud.account = 'Developer Sandbox' AND dest.country = 'India'
  AND dest.state = 'Karnataka'
```

For example, you can view network traffic from Karnataka in India:

```
network where cloud.account = 'Developer Sandbox' AND source.country =
  'India' AND source.state = 'Karnataka'
```

- **dest.country, source.country**

Use the **dest.country**, **source.country** attributes to filter your network to view traffic from the country of its origin or the country where the traffic is received.

For example, you can view network activity where the destination of the traffic is in China and Russia:

```
network where dest.country IN ( 'China' , 'Russia' ) and bytes > 0
```

To view network activity where the source of the traffic is in China:

```
network where source.country = 'China' AND bytes > 0
```

- **bytes**

Use the **bytes** attribute to search for network related information by the aggregate byte volume while the transmission lasts.

For example, you can search for network traffic by internet IPs, suspicious IPs and bytes:

```
network where source.publicnetwork IN ( 'Internet IPs' , 'Suspicious IPs' )
  and bytes > 0
```

- **response.bytes**

Use the **dest.country**, **source.country** attribute to search for network related information by the aggregate response byte volume.

For example, you can search for network traffic with response bytes more than 1,00,000:

```
network where response.bytes > 100000 AND cloud.account = 'Sandbox Account'
```

- **accepted.bytes**

Use the **accepted.bytes** attribute to search for the network related information by the aggregate accepted byte volume.

For example, you can search for network traffic with accepted bytes more than 1,00,000:

```
network where accepted.bytes > 100000 AND cloud.account = 'Sandbox Account'
```

- **packets**

Use the **packets** attribute to search for network related information by the aggregate packet volume while the transmission lasts.

For example, you can identify traffic from internal workloads to internet IPs on ports 8545,30303 that are known to mine Ethereum:

```
network where dest.port IN (8545,30303) and dest.publicnetwork IN ('Internet  
IPs' , 'Suspicious IPs' ) and packets> 0
```

- **protocol**

Use the **protocol** attribute to search for network-related information in relation to network protocols.

For example, you can search for network information by TCP protocol and where the destination port is 21:

```
network where src.ip=0.0.0.0 AND protocol='TCP' AND dest.port IN (21)
```

- **role**

Use the **role** attribute to filter the network traffic by roles.

For example, you can view all network traffic in RedLock account where the destination resource role is not AWS NAT Gateway and AWS ELB:

```
network where cloud.account = 'RedLock' AND source.ip = 0.0.0.0 AND  
dest.resource IN ( resource where role NOT IN ( 'AWS NAT Gateway' , 'AWS  
ELB' ) )
```

For example, you can view traffic originating from suspicious IPs and internet IPS which are hitting the resource roles AWS RDS and Database:

```
network where source.publicnetwork IN ( 'Suspicious IPs' , 'Internet IPs' )  
and dest.resource IN ( resource where role IN ( 'AWS RDS' , 'Database' ) )
```

- **tag**

Use **tag** attribute to filter the network traffic by tags.

For example, you can view network traffic which is hitting the resources that are tagged as NISP:

```
network where dest.resource IN ( resource where tag ( 'name' ) = 'NISP' )
```

- **traffic.type IN**

Use **traffic.type** IN attribute to view how entities within your cloud environment have accepted and rejected traffic.

For example,

```
NETWORK WHERE src.publicnetwork IN ('Suspicious IPs','Internet IPs') AND
dest.resource IN (resource WHERE virtualnetwork.name IN ( 'vpc-323cda49' ))
AND dest.ip IN (172.31.12.172 ) AND traffic.type IN ('REJECTED')
```

Network Query Examples

Use this section for some examples that show you how to use [Network Query Attributes](#) in RQL for investigating Network issues:

DESCRIPTION	RQL
View traffic originating from the Internet & suspicious IPs to resource with Database role.	<pre>network where source.publicnetwork IN ('Suspicious IPs' , 'Internet IPs') and dest.resource IN (resource where role IN ('AWS RDS' , 'Database'))</pre>
Find instances that are accessible over the internet using insecure ports.	<pre>network where source.publicnetwork IN ('Internet IPs') and protocol = 'TCP' AND dest.port IN (21,23,80)</pre>
Find hosts with Meltdown and Spectre vulnerabilities receiving network traffic.	<pre>network where dest.resource IN (resource where hostfinding.type IN ('Host Vulnerability') AND hostfinding.name IN ('CVE-2017-5754', 'CVE-2017-5753', 'CVE-2017-5715')) and bytes > 0</pre>
Look for traffic from internet to any instance outside of Web servers, NAT Gateways or ELBs.	<pre>Network where src.publicnetwork IN ('Suspicious IPs','Internet IPs') AND dest.resource IN (resource where role not in ('AWS NAT Gateway' , 'AWS ELB')) and protocol not in ('ICMP' , 'ICMP6')</pre>
Look for source entities which are AWS ELBs with connections to more than 10 unique peer IP addresses, but those peer IPs are not endpoints that function as Databases.	<pre>Network where src.resource IN (RESOURCE WHERE role = ('AWS ELB') AND source.outboundpeers > 10) AND dest.resource IN (RESOURCE WHERE role != ('Database'))</pre>

AWS APIs Ingested by Prisma Cloud

The following are AWS APIs that are ingested by Prisma Cloud.

SERVICE	API NAME IN PRISMA CLOUD
Account Attributes	aws-describe-account-attributes
API Gateway	<ul style="list-style-type: none">aws-apigateway-get-rest-apisaws-apigateway-get-stages
AWS AutoScaling	aws-describe-auto-scaling-groups
AWS Certificate Manager	aws-acm-describe-certificate
AWS CloudFormation	aws-cloudformation-describe-stacks
AWS CloudFront	aws-cloudfront-list-distributions
AWS CloudTrail	<ul style="list-style-type: none">aws-cloudtrail-describe-trailsaws-cloudtrail-get-event-selectorsaws-cloudtrail-get-trail-status
AWS CloudWatch	<ul style="list-style-type: none">aws-cloudwatch-describe-alarmsaws-logs-describe-metric-filters
Amazon Cognito	<ul style="list-style-type: none">aws-cognito-identity-poolaws-cognito-user-pool
Amazon MQ	aws-mq-broker
Config	aws-configservice-describe-configuration-records
Delivery Channels	aws-describe-delivery-channels
DynamoDB	aws-dynamodb-describe-table
EC2	<ul style="list-style-type: none">aws-ec2-describe-snapshotsaws-ec2-describe-volumesaws-elb-describe-load-balancersaws-elbv2-describe-load-balancersaws-ec2-describe-flow-logsaws-ec2-describe-instancesaws-ec2-describe-imagesaws-ec2-describe-internet-gatewaysaws-ec2-describe-network-interfacesaws-ec2-describe-network-aclsaws-ecr-get-repository-policyaws-ecs-describe-task-definition

SERVICE	API NAME IN PRISMA CLOUD
	<ul style="list-style-type: none"> aws-describe-ssl-policies aws-ec2-autoscaling-launch-configuration
AWS Elastic Beanstalk	aws-elasticbeanstalk-environment
Amazon Elastic Container Registry (ECR)	aws-ecr-get-repository-policy
AWS Elastic File System (EFS)	aws-describe-mount-targets
Amazon Elastic Container Service for Kubernetes (EKS)	aws-eks-describe-cluster
ElastiCache	<ul style="list-style-type: none"> aws-cache-engine-versions aws-elasticache-cache-clusters aws-elasticache-parameter-groups aws-elasticache-describe-replication-groups aws-elasticache-reserved-cache-node-offering aws-elasticache-reserved-cache-nodes aws-elasticache-snapshots aws-elasticache-subnet-groups
Amazon ElasticSearch	aws-es-describe-elasticsearch-domain
Amazon Elastic MapReduce (EMR)	aws-emr-describe-cluster
Amazon S3 Glacier	<ul style="list-style-type: none"> aws-glacier-get-vault-access-policy aws-glacier-get-vault-lock
Amazon GuardDuty	aws-guardduty-detector
AWS Identity and Access Management (IAM)	<ul style="list-style-type: none"> aws-iam-list-access-keys aws-iam-get-account-summary aws-iam-list-server-certificates aws-iam-get-credential-report aws-iam-list-mfa-devices aws-iam-list-virtual-mfa-devices aws-iam-get-account-password-policy aws-iam-get-policy-version aws-iam-list-users aws-iam-list-user-policies aws-iam-list-roles aws-iam-list-groups aws-iam-list-attached-user-policies aws-iam-list-ssh-public-keys aws-iam-saml-provider
AWS Key Management Service (KMS)	aws-kms-get-key-rotation-status

SERVICE	API NAME IN PRISMA CLOUD
Amazon Kinesis Data Streams (KDS)	aws-kinesis-list-streams
AWS Lambda	<ul style="list-style-type: none"> aws-lambda-list-functions aws-lambda-get-region-summary
Amazon Relational Database Service (RDS)	<ul style="list-style-type: none"> aws-rds-describe-db-instances aws-rds-describe-db-snapshots aws-rds-describe-event-subscriptions aws-rds-db-cluster-snapshots aws-rds-db-clusters
Amazon RedShift	aws-redshift-describe-clusters
AWS Route53	aws-route53-list-hosted-zones
Amazon RDS	aws-rds-describe-db-parameter-groups
AWS Secrets Manager	aws-secretsmanager-describe-secret
AWS Systems Manager	aws-ssm-parameter
Amazon Simple Storage Service (Amazon S3)	<ul style="list-style-type: none"> aws-s3api-get-bucket-acl <p>The list of APIs associated with this API name are:</p> <ul style="list-style-type: none"> listBuckets getS3AccountOwner getRegionName getBucketLocation getBucketAcl getBucketPolicy getBucketPolicyStatus getBucketVersioningConfiguration aws-s3control-public-access-block
Amazon Simple Notification Service (SNS)	<ul style="list-style-type: none"> aws-sns-get-subscription-attributes aws-sns-get-topic-attributes
Amazon Simple Queue Service (SQS)	aws-sqs-get-queue-attributes
Amazon VPC	<ul style="list-style-type: none"> aws-ec2-describe-security-groups aws-ec2-describe-route-tables aws-ec2-describe-subnets aws-ec2-describe-vpcs aws-ec2-describe-vpc-peering-connections aws-ec2-customer-gateways-summary aws-ec2-describe-customer-gateways aws-describe-vpc-endpoints aws-ec2-vpn-connections-summary

SERVICE	API NAME IN PRISMA CLOUD
	<ul style="list-style-type: none">• aws-ec2-describe-vpn-connections• aws-ec2-describe-vpn-gateways• aws-ec2-describe-vpn-gateways-summary• aws-vpc-dhcp-options
Amazon Workspaces	<ul style="list-style-type: none">• aws-describe-workspace-directories• aws-workspaces-describe-workspaces

GCP APIs Ingested by Prisma Cloud

The following are GCP APIs that have been ingested by Prisma Cloud.

SERVICE	API NAME IN PRISMA CLOUD
Google Compute Engine (GCE) --	<ul style="list-style-type: none">gcloud-compute-networks-listgcloud-compute-networks-subnets-listgcloud-compute-firewall-rules-listgcloud-compute-instances-listgcloud-compute-interfaces-list
Google App Engine	gcloud-app-engine-firewall-rule
Google Stackdriver	<ul style="list-style-type: none">gcloud-monitoring-policies-list
SQL Databases	gcloud-sql-instances-list
Google Cloud Storage	gcloud-storage-buckets-list
Cloud Identity & Access Management (Cloud IAM)	<ul style="list-style-type: none">gcloud-iam-service-accounts-listgcloud-iam-service-accounts-keys-listgcloud-projects-get-iam-policygcloud-projects-get-iam-usergcloud-iam-get-audit-config
Cloud SQL	gcloud-sql-instances-list
BigQuery	gcloud-bigquery-dataset-list
Google Cloud Kubernetes	gcloud-container-describe-clusters
Cloud Spanner	gcloud-cloud-spanner-instance-list
Compute Disk	gcp-compute-disk-list
DNS	<ul style="list-style-type: none">gcloud-dns-project-infogcloud-dns-managed-zone
Big Table	gcloud_bigtable-instance-list
Cloud Key Management Service (Cloud KMS)	gcloud-kms-keyring-list
Logging sinks	gcloud-logging-sinks-list
Compute	<ul style="list-style-type: none">gcloud-compute-project-infogcloud-compute-load-balancer
Google Cloud Load Balancer	<ul style="list-style-type: none">gcloud-compute-internal-lb-backend-servicegcloud-lb-virtual-network-interface

SERVICE	API NAME IN PRISMA CLOUD
	<ul style="list-style-type: none"> gcloud-compute-target-pools gcloud-compute-target-https-proxies gcloud-compute-url-maps gcloud-compute-global-forwarding-rule
Google Cloud Memorystore for Redis	gcloud-redis-instances-list

Microsoft Azure APIs Ingested by Prisma Cloud

The following APIs are ingested by Prisma Cloud

SERVICE	API NAME IN PRISMA CLOUD
Azure Active Directory (for IAM)	azure-ad-user-list
Azure Disk	azure-disk-list
Azure Key Vault	azure-key-vault-list azure-key-vault-certificate
Azure Application Gateway	azure-application-gateway
Azure Load Balancer	azure-network-lb-list azure-lb-virtual-network-interface
Azure Container Registry	azure-container-registry
Azure Cosmos DB	azure-cosmos-db
Azure Resource Group	azure-resource-group
Azure Resource Locks	azure-lock-list
Network Usage	azure-network-usage
Azure Security Center	<ul style="list-style-type: none">• azure-security-center• azure-security-center-settings
Azure Virtual Machines	azure-vm-list
Azure Virtual Network	<ul style="list-style-type: none">• azure-network-vnet-list• azure-network-subnet-list• azure-network-route-table• azure-network-lb-list• azure-network-peering• azure-network-nic-list• azure-network-nsg-list• azure-network-vpn-connection-list
Databases	<ul style="list-style-type: none">• azure-sql-db-list• azure-sql-server-list
Monitoring	<ul style="list-style-type: none">• azure-activity-log-alerts• azure-monitor-log-profiles-list• azure-network-watcher-list

SERVICE	API NAME IN PRISMA CLOUD
Policy Assignments	azure-policy-assignments
Storage Account	azure-storage-account-list
Subnets	azure-network-subnet-list
Azure Kubernetes Service	azure-kubernetes-cluster
App service	azure-app-service

RQL Operators

An operator in RQL is one or more symbols or words that compare the value of a field on its left with one or more values on its right, such that only valid results are retrieved and displayed to you. You can use an RQL operator to find a specific term included as a value within an object or an array in a JSON.

The following operators and conditions that you can use to compare or validate results:

- [Operators Within JSON Arrays](#)
- [Config and Event Operators](#)
- [Joins](#)
- [Functions](#)

Operators Within JSON Arrays

OPERATOR	DESCRIPTION	RQL EXAMPLE
@ and ?	@ and ? are expressions used to filter arrays. <ul style="list-style-type: none">• ? opens the array.• @ represents the current item being processed. It is used to hone in on a particular block in the json object so that you are only matching that block and no others.	<pre>config where api.name='aws- ec2-describe- security-groups' AND json.rule='ipPermissions[? (@.fromPort==0)].ipRanges[*] contains 0.0.0.0/0'</pre>
&& and	Combine conditions within json.rule using && and .	<pre>config where api.name = 'aws-s3api- get-bucket-acl' and json.rule = "policy.Statement exists and policy.Statement[? (@.Action=='s3:GetObject' && @.Effect=='Allow' @.Action=='s3:ListBucket' && @.Effect=='Allow')].Principal contains *"</pre>

Config and Event Operators

OPERATOR	DESCRIPTION	RQL EXAMPLE
Greater than	Compares a path on left-hand side against either a numeric	<pre>config where api.name = 'aws-iam-get- account-password- policy' AND json.rule</pre>

OPERATOR	DESCRIPTION	RQL EXAMPLE
	value or another path on the right-hand side.	<code>= maxPasswordAge greater than 20</code>
Less than	Compares a path on left-hand side against either a numeric value or another path on the right-hand side.	<code>config where api.name = 'aws-iam-get- account-password- policy' AND json.rule = maxPasswordAge less than 100</code>
Equals	Compares a path on left-hand side against either a numeric value or another path on the right-hand side.	<code>config where api.name = 'aws-iam-get- account-password- policy' AND json.rule = maxPasswordAge equals 90</code>
Does not equal	Compares a path on left-hand side against either a numeric value or another path on the right-hand side.	<code>config where api.name = 'aws-iam-get- account-password- policy' AND json.rule = maxPasswordAge does not equal 90</code>
Starts with	Left-hand side must be a path with a string value.	<code>config where api.name = 'aws-iam-list- users' and json.rule = userName starts with y</code>
Does not start with	Left-hand side must be a path with a string value.	<code>config where api.name = 'aws-iam-list- users' and json.rule = userName does not start with y</code>
Ends with	Left-hand side must be a path with a string value.	<code>config where api.name = 'aws-iam-list- users' and json.rule = userName ends with i</code>
Does not end with	Left-hand side must be a path with a string value.	<code>config where api.name = 'aws-iam-list- users' and json.rule = userName does not end with i</code>
Contains	The left-hand side may be a single path or a set of paths with numeric or string value.	<code>config where api.name = 'azure-network-nsg- list' AND json.rule =</code>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<code>defaultSecurityRules[*].direction contains outbound</code>
Does not contain	The left-hand side may be a single path or a set of paths with numeric or string value.	<code>config where cloud.type = 'azure' AND api.name = 'azure-vm-list' AND json.rule = powerState does not contain allocated</code>
Is empty	The left-hand side must be a path leading to a string value.	<code>config where api.name = 'aws-ec2- describe-instances' and json.rule = publicIpAddress is empty</code>
Is not empty	The left-hand side must be a path leading to a string value.	<code>config where api.name = 'aws-ec2- describe-instances' and json.rule = publicIpAddress is not empty</code>
Exists	The left-hand side must be a path.	<code>config where api.name = 'aws-ec2-describe- network-interfaces' AND json.rule = 'association.publicIp exists'</code>
Does not exist	The left-hand side must be a path.	<code>config where cloud.type = 'gcp' AND cloud.service = 'Compute Engine' and api.name = 'gcloud- compute-instances- list' AND json.rule = metadata.kind does not exist</code>
Any start with	The left-hand side must be a set of paths leading to string values.	<code>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId any start with vpc-3</code>
None start with	The left-hand side must be a set of paths leading to string values.	<code>config where api.name = 'aws-ec2-</code>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<pre>describe-instances' AND json.rule = networkInterfaces[*].vpcId none start with vpc-323cda</pre>
All start with	The left-hand side must be a set of paths leading to string values.	<pre>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId all start with vpc-323cda</pre>
Any end with	The left-hand side must be a set of paths leading to string values.	<pre>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId any end with 49</pre>
None end with	The left-hand side must be a set of paths leading to string values.	<pre>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId none end with 49</pre>
All end with	The left-hand side must be a set of paths leading to string values.	<pre>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId all end with 49</pre>
Any equal	The left-hand side must be a set of paths leading to string or numeric values.	<pre>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId any equal vpc-323cda49</pre>
None equal	The left-hand side must be a set of paths leading to string or numeric values.	<pre>config where api.name = 'aws-ec2- describe-instances' AND json.rule = networkInterfaces[*].vpcId</pre>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<pre>none equal vpc-323cda49</pre>
All equal	The left-hand side must be a set of paths leading to string or numeric values.	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = networkInterfaces[*].vpcId all equal vpc-323cda49</pre>
Size equals	<p>The left-hand side must be an array.</p> <p>The right-hand side must be an integer.</p>	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = tags[*] size equals 0</pre>
Size does not equal	<p>The left-hand side must be an array.</p> <p>The right-hand side must be an integer.</p>	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = tags[*] size does not equal 0</pre>
Size greater than	<p>The left-hand side must be an array.</p> <p>The right-hand side must be an integer.</p>	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = tags[*] size greater than 1</pre>
Size less than	<p>The left-hand side must be an array.</p> <p>The right-hand side must be an integer.</p>	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = tags[*] size less than 1</pre>
Length equals	<p>The left-hand side is a path with a string value.</p> <p>The right-hand side must be an integer.</p>	<pre>config where api.name = 'aws-rds-describe-db-snapshots' AND json.rule = snapshot.storageType length equals 3</pre>
Length does not equal	<p>The left-hand side is a path with a string value.</p> <p>The right-hand side must be an integer.</p>	<pre>config where api.name = 'aws-rds-describe-db-snapshots' AND json.rule = snapshot.storageType</pre>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<code>length does not equal 3</code>
Length greater than	The left-hand side is a path with a string value. The right-hand side must be an integer.	<code>config where api.name = 'aws-rds-describe-db-snapshots' AND json.rule = snapshot.storageType length greater than 3</code>
Length less than	The left-hand side is a path with a string value. The right-hand side must be an integer.	<code>config where api.name = 'aws-rds-describe-db-snapshots' and json.rule = snapshot.storageType length less than 4</code>
Number of words equals	The left-hand side is a path with a string value.	<code>config where cloud.type = 'gcp' AND cloud.service = 'Compute Engine' and api.name = 'gcloud-compute-instances-list' AND json.rule = cpuPlatform number of words equals 3</code>
Number of words does not equal	The left-hand side is a path with a string value.	<code>config where cloud.type = 'gcp' AND cloud.service = 'Compute Engine' and api.name = 'gcloud-compute-instances-list' AND json.rule = cpuPlatform number of words does not equal 3</code>
Number of words greater than	The left-hand side is a path with a string value.	<code>config where cloud.type = 'gcp' AND cloud.service = 'Compute Engine' and api.name = 'gcloud-compute-instances-list' AND json.rule = cpuPlatform number of words greater than 2</code>
Number of words less than	The left-hand side is a path with a string value.	<code>config where cloud.type = 'gcp' AND cloud.service = 'Compute Engine' and api.name = 'gcloud-</code>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<code>compute-instances-list' AND json.rule = cpuPlatform number of words less than 3</code>
Any True	The left-hand side is a set of paths with Boolean values.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nic-list' AND json.rule = " ['properties.ipConfigurations'] [*]. ['properties.primary'] any true "</code>
None True	The left-hand side is a set of paths with Boolean values.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nic-list' AND json.rule = " ['properties.ipConfigurations'] [*]. ['properties.primary'] none true"</code>
All True	The left-hand side is a set of paths with Boolean values.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nic-list' AND json.rule = " ['properties.ipConfigurations'] [*]. ['properties.primary'] all true</code>
Any False	The left-hand side is a set of paths with Boolean values.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nic-list' AND json.rule = " ['properties.ipConfigurations'] [*]. ['properties.primary'] any false"</code>
None False	The left-hand side is a set of paths with Boolean values.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nic-list' AND json.rule = " ['properties.ipConfigurations'] [*].</code>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<code>['properties.primary'] none false"</code>
All False	The left-hand side is a set of paths with Boolean values.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nic- list' AND json.rule = " ['properties.ipConfigurations'] [*]. ['properties.primary'] all false"</code>
Is True	The left-hand side is a Path with Boolean value.	<code>config where api.name = 'azure-storage- account-list' AND json.rule = encryptionStatuses.Blob is true</code>
Is False	The left-hand side is a Path with Boolean value.	<code>config where api.name = 'azure-storage- account-list' AND json.rule = encryptionStatuses.Blob is false</code>
Is Not Member of	The left-hand side is a Path with string value, and the right-hand side is a set of values in parentheses and separated with commas	<code>Config where api.name = 'aws-ec2-describe- security-groups' AND json.rule = ipPermissions[*].toPort exists and ipPermissions[*].fromPort is not member of (22)</code>
Is Member of	The left-hand side is a Path with string value, and the right-hand side is a set of values in parentheses and separated with commas	<code>config where api.name = 'aws-ec2-describe- security-groups' AND json.rule = ipPermissions[*].toPort exists and ipPermissions[*].toPort is member of (3389,22,5432)</code> <code>config where api.name = 'aws-ec2-describe- security-groups' AND json.rule = ipPermissions[*].ipProtocol exists and</code>

OPERATOR	DESCRIPTION	RQL EXAMPLE
		<code>ipPermissions[*].ipProtocol is member of (tcp)</code>
Matches Does not Match	For Event queries, use the boolean operators Matches and Does not Match to match or not match field values against simple patterns and not full REGEX. Patterns can have substrings and * for wild character search.	In the following example, the operation MATCHES 'c*login' enables you to list activities that match <code>clogin</code> , <code>cloudlogin</code> , or <code>consolelogin</code> . <code>event where cloud.type = 'aws' AND cloud.account = 'RedLock Sandbox' AND operation MATCHES 'c*login'</code>

Joins

Joins allow you to get data from two different APIs where you have combined different conditions. You can use Joins only across **config where** queries, and can include up to three configuration API resources with alias X, Y and Z.

Joins across event, network, and config are not supported.

Join basic syntax:

```
config where api.name=".." as X; config where api.name="..." as Y; filter
"$X... <operator> $.Y"; show (X;|Y;)
```

To find EC2 instances that have public IP addresses assigned at launch use this query:

Steps:

1. List EC2 instances as X:

```
config where api.name = 'aws-ec2-describe-instances' as X;
```

2. List subnets as Y:

```
config where api.name = 'aws-ec2-describe-subnets' as Y;
```

3. Set the filter:

```
filter '$X.subnetId == $.Y.subnetId and $.Y.mapPublicIpOnLaunch is true';  
show X;
```

4. Complete the query to list instances in subnets which have public IP addresses auto-assigned:

```
config where api.name = 'aws-ec2-describe-instances' as X; config where  
api.name = 'aws-ec2-describe-subnets' as Y; filter '$X.subnetId ==  
$.Y.subnetId and $.Y.mapPublicIpOnLaunch is true'; show X;
```

Examples of Joins:

DESCRIPTION	RQL EXAMPLE
VPCs that are connected to internet gateways.	<pre>config where api.name = 'aws-ec2-describe-internet-gateways' as X; config where api.name = 'aws-ec2-describe-vpcs' as Y; filter '\$.X.attachments[*].vpcId == \$.Y.vpcId and \$.Y.tags[*].key contains IsConnected and \$.Y.tags[*].value contains true'; show Y;</pre>
CloudTrail logs that are integrated with CloudWatch for all regions.	<pre>config where api.name = 'aws-cloudtrail-describe-trails' as X; config where api.name = 'aws-cloudtrail-get-trail-status' as Y; filter '\$.X.cloudWatchLogsLogGroupArn != null and (\$.Y.status.latestCloudWatchLogsDeliveryTime != null and _DateTime.ageInDays(\$.Y.status.latestCloudWatchLogsDeliveryTime, _DateTime.now()) > 1) and (\$.X.rnrn == \$.Y.rnrn)'; show X;</pre>

Functions

A function performs a calculation on specific data that matches the clause contained in the function and displays true results. Functions support auto-complete when you enter the prefix `_` in a `json.rule` or `addColumn` attribute.

Prisma Cloud supports following functions:

- [_DateTime Examples](#)
- [_AWSAccount.isRedLockMonitored Examples](#)
- [_IPAddress.inRange Examples](#)
- [_Port.inRange Examples](#)

DateTime Examples

Query time ranges are not part of RQL grammar, and the query time window is passed as a separate argument to the query APIs. The selection of the attributes or columns for a category are not part of the RQL grammar.

The query time ranges that are available are `_DateTime.ageInDays`, `_DateTime.ageInMonths`, `_DateTime.ageInYears`, and `_DateTime.daysBetween`. The `_DateTime.daysBetween` function looks for any information that falls in between two dates and takes two dates as arguments.



When using the `_DateTime` function all json parameters are available as auto-complete options, you must select only parameters that have timestamps. Also, the syntax for a

function does not support spaces. Remove empty spaces before or after parenthesis, and between comma-separated parameters.

DESCRIPTION	RQL EXAMPLE
List EC2 instances with age greater than 2 days.	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = '_DateTime.ageInDays(launchTime) > 2'</pre>
List resource names where access keys are not rotated for 90 days.	<pre>config where api.name = 'aws-iam-get-credential-report' AND json.rule = '(access_key_1_active is true and access_key_1_last_rotated != N/A and _DateTime.ageInDays(access_key_1_last_rotated) > 90) or (access_key_2_active is true and access_key_2_last_rotated != N/A and _DateTime.ageInDays(access_key_2_last_rotated) > 90)'</pre>
Use the function today() to return the current day's date.	<pre>config where cloud.type = 'aws' and api.name = 'aws-cloudtrail-get-trail-status' AND json.rule = '"_DateTime.daysBetween(\$.latestDeliveryTime,today()) > 2"'</pre>

_AWSCloudAccount.isRedLockMonitored Examples

DESCRIPTION	RQL EXAMPLE
List any snapshots that are shared publicly and are not monitored by Prisma Cloud.	<pre>config where api.name = 'aws-ec2-describe-snapshots' AND json.rule = 'createVolumePermissions[*].size != 0 and _AWSCloudAccount.isRedLockMonitored(createVolumePermissions[*].size) is false'</pre>

_IPAddress.inRange Examples

To check if a particular IP address is part of an IP address range, use `_IPAddress.inRange` and in the argument specify the octets, along with the `<fromInteger>`, `<toInteger>`. For example ("172.%d",16,31) or ("172.10.%d",10,255).

DESCRIPTION	RQL EXAMPLE
List AWS Route53 Public Zones that have Private Records.	<p>In this example, the <code>IPAddress.inRange("172.%d",16,31)</code> allows you to search for IP addresses that are in the range "172.16.x.x" to "172.31.x.x":</p> <pre>config where cloud.type = 'aws' AND api.name = 'aws-route53-list-hosted-zones' AND json.rule = resourceRecordSet[*].resourceRecords[*].value any start with _IPAddress.inRange("172.%d",16,31)</pre>

Port.inRange Examples

To check if a particular port number is part of a specific range, use class **Port** and method **inRange**. This method takes three arguments **<fromInteger>**, **<toInteger>**, and you can optionally include an **<offset>**.



By default, the **<offset>** is 1.

DESCRIPTION	RQL EXAMPLE
<p>Use the inRange function with the contains and does not contain operators to check for conditions on a port range.</p> <p>Specify <fromInteger> and <toInteger> to find all ports within the specified range.</p>	<p>Example using contains to check for ports numbers between 22 and 33 with an offset of 1:</p> <pre>config where api.name = 'aws-ec2-describe-security-groups' AND json.rule = ipPermissions[*].toPort exists and ipPermissions[*].toPort contains _Port.inRange(22,33,1)</pre> <p>The example above checks for all ports between 22 and 33.</p>
	<p>Example using Does not Contain:</p> <pre>config where api.name = 'azure-network-nsg-list' AND json.rule = securityRules[*].sourcePortRanges[*] does not contain _Port.inRange(350,5400,5)</pre> <p>The example above checks for ports 350, 355, 360,5390, 5395, 5600.</p>
	<p>Example using no offset, to find all ports within the specified range:</p> <pre>config where api.name = 'aws-ec2-describe-security-groups' AND json.rule = ipPermissions[*].toPort</pre>

DESCRIPTION	RQL EXAMPLE
	<pre>exists and ipPermissions[*].toPort contains _Port.inRange(400,500)</pre>

RQL FAQs

The following are answers to frequently asked questions.

- **Is time range part of grammar?**

Query time ranges are not part of grammar. Instead, the query time window is passed as a separate argument to the query APIs. The selection of what attributes or columns of a category are returned are not part of the RQL grammar.

- **Is aggregating of query results supported?**

Aggregating, limiting, or formatting of query results are not supported.

- **Are cross joins supported?**

Joins across Event, Network, and Config are not supported. Joins are supported across Config queries only, and you can include up to three joins.

- **Why can't I see auto-complete suggestions when using functions?**

Autocomplete suggestions are not supported for function parameters. Autocomplete suggestions are also not supported for some attributes.

- **What is the correct way of grouping or breaking negated clauses within the `json.rule`?**

Use De Morgan Laws when grouping or breaking negated clauses within the `json.rule`.

For example, to separate the conditions, use `(not ($.x is true)) or (not ($.y is true))` instead of `(not ($.x is true and $.y is true))`

Similarly, if you want to use separate clauses use `(not ($.x is true)) and (not ($.y is true))` instead of `(not ($.x is true or $.y is true))`.

- **How do I use quotation marks in a JSON rule?**

When you use the `json.rule` attribute, you can use specify the match conditions in the expression using single quotation marks, double quotation marks, or without any quotation marks.

You can for example, say:

`json.rule = encrypted is true`, or

`json.rule = 'encrypted is true'`, or

`json.rule = "encrypted is true"`

The query output is the same whether you enclose it in quotation marks or not, the main difference is that the expression is validated only when you build the query without quotation marks.

If you use functions or are matching data within an array, you must use single or double quotation marks. And if you have used a single quote in an array condition, use a double quote to enclose the expression. For example, `json.rule = "ipAddress[?(@.x=='a')].port"`.

- **Can we apply primitive operators to a set of values?**

[RQL Operators](#) allow you to specify a condition using an integer, string, or boolean in an RQL query to a set of values within an array or a list. The list of primitive operators supported are `==`, `!=`, `>`, `<`, `is true`, `is false`.

On the Left Hand side (LHS) of an RQL expression, you can include a value or a set of values, and on the Right Hand Side (RHS) provide the value at the leaf which is typically a value that can be an integer, character, boolean or null.

PRIMITIVE OPERATOR	WHEN APPLIED TO LHS CONTAINING SET OF VALUES
<code>==</code>	is any one of them exactly equal to
<code>!=</code>	is any one of them not equal to
<code>></code>	any greater than
<code><</code>	any less than
<code>is true</code>	any true
<code>is false</code>	any false

- **How do we use the operator `MATCHES` in Event RQL?**

Use the boolean operators **`Matches`** and **`Does not Match`** to match or not match field values against simple patterns (and do not use regular expressions). A pattern can include substrings and `*` to support wild characters.

In the following example, the operation **`matches 'c*login'`** enables you to list all activities that match **`clogin`**, **`cloudlogin`**, or **`consolelogin`**:

```
event where cloud.type = 'aws' AND cloud.account = 'RedLock Sandbox' AND
operation MATCHES 'c*login'
```

- **Which are some constructs that are used across config, network, and event queries?**

- **`bytes` or `packets`**

Use **`bytes`** to search for network related information by the aggregate byte or packet volume while the transmission lasts. For example, to search for network traffic generated by IP addresses that are coming from the public internet or from suspicious IP addresses:

```
network where source.publicnetwork IN ( 'Internet IPs' , 'Suspicious
IPs' ) and bytes > 0
```

To identify traffic from internal workloads to public IP addresses on ports 8545,30303 that are known to mine Ethereum:

```
network where dest.port IN (8545,30303) and dest.publicnetwork IN
('Internet IPs' , 'Suspicious IPs' ) and packets> 0
```

- **`operation`**

An **`operation`** is an action performed by users on resources in a cloud account. When you type the name of the operation that you are interested in, auto-suggest displays the options that match the search criteria. For example, to view all operations such as deletion of VPCs, VPC endpoints, and VPC peering connections:

```
event where operation in ( 'DeleteVpc' , 'DeleteVpcEndpoints'
'DeleteVpcPeeringConnection' )
```

- **`protocol`**

You can search for network traffic based on protocols. For example, to search for network traffic from any public IP address that uses the TCP protocol and where the destination port is 21:

```
network where source.ip=0.0.0.0 AND protocol='TCP' AND dest.port IN (21)
```

- **role**

Use **role** to filter the network traffic by resource roles.

For example, to show all network traffic from any public IP address in a specific cloud account where the destination resource role is not AWS NAT Gateway and AWS ELB:

```
network where cloud.account = 'Redlock' AND source.ip = 0.0.0.0 AND  
dest.resource IN ( resource where role NOT IN ( 'AWS NAT Gateway' ,  
'AWS ELB' ))
```

To view traffic originating from suspicious IPs and internet IPs which are hitting the resource roles AWS RDS and Database:

```
network where source.publicnetwork IN ( 'Suspicious IPs' , 'Internet  
IPs' ) and dest.resource IN ( resource where role IN ( 'AWS RDS' ,  
'Database' ))
```

- **tag**

Use **tag** to filter the network traffic with a specific tag. For example, to find all resources that are tagged as NISP to which you network traffic:

```
network where dest.resource IN ( resource where tag ('name') = 'NISP' )
```

- **user**

To search for operations performed by specific users, use **user**. For example, to view all console login operations by Ben:

```
event where operation = 'ConsoleLogin' AND user = 'ben'
```

- **addcolumn**

Use **addcolumn** to dynamically display columns for the Config queries results that are displayed on screen.

To add columns for key name and image ID for EC2 instances, for example:

```
config where api.name = 'aws-ec2-describe-instances' addcolumn keyName  
hypervisor imageId
```

RQL Example Library

Use the Resource Query Language (RQL) examples in this section to learn how to monitor and detect issues on your cloud resources.

- [AWS Examples](#)
- [Azure Examples](#)
- [GCP Examples](#)
- [Common Useful Query Examples](#)

AWS Examples

DESCRIPTION	RQL
List EC2 instances with a public IP address.	<pre>config where api.name = 'aws-ec2-describe-instances' and json.rule = publicIpAddress exists</pre> <pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = publicIpAddress exists and publicIpAddress is not empty</pre>
List EC2 instances that are attached to a Security Group named 'allow-all'.	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = 'securityGroups contains allow-all'</pre>
List all EC2 instances that have a publicly accessible hostname.	<pre>config where api.name = 'aws-ec2-describe-instances' and json.rule = 'publicDnsName exists '</pre>
List all EC2 instances that have a public IP address and allows any IP address to connect to it.	<pre>config where api.name = 'aws-ec2-describe-security-groups' AND json.rule = '\$.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0'</pre>
List all EC2 instances that associated with a specific security group.	<pre>config where api.name = 'aws-ec2-describe-instances' AND json.rule = securityGroups[*].groupId contains "sg-c57910b7"</pre>
List all EC2 instances that have a public IP address and are publicly accessible (The IP range is not restricted to a set of specific IP addresses).	<pre>config where api.name = 'aws-ec2-describe-instances' as X; config where api.name = 'aws-ec2-describe-security-groups' as Y; filter '\$.X.publicIpAddress exists and not \$.X.publicIpAddress is empty and \$.X.securityGroups[*].groupName</pre>

DESCRIPTION	RQL
	<pre> == \$.Y.groupName and \$.Y.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0 and \$.Y.ipPermissions[*].ipProtocol == -1'; show X; </pre>
List all EC2 instances that are not in a specified destination security group and have traffic flowing from a resource that does not have a specified tag. (uses the NOT IN operator for negation)	<pre> network where accepted.bytes > 10000 AND dest.resource NOT IN (resource where securitygroup.name = '2nd_hong_kong_sg') AND source.resource NOT IN (resource where tag (ANY) IN ('HelloWorld')) </pre>
Find EC2 instances where launch time is more than 30 days.	<pre> config where api.name = 'aws-ec2- describe-instances' AND json.rule = '_DateTime.ageInDays(\$.launchTime) > 30' </pre>
Find all EBS volumes that do not have a Data Classification tag.	<pre> config where api.name = 'aws-ec2- describe-volumes' AND json.rule = 'tags[*].key != DataClassification </pre>
Find all RDS snapshots that are shared with cloud accounts that Prisma Cloud is not monitoring.	<pre> config where api.name = 'aws- rds-describe-db-snapshots' AND json.rule = "\$.attributes[? (@.attributeName=='restore')].attributeValues[*] size != 0 and _AWSCloudAccount.isRedLockMonitored(\$.attribute (@.attributeName=='restore')].attributeValues) is false" </pre>
Find all Security Groups that opens port 22 to the internet (and are attached to an EC2 instance).	<pre> config where api.name='aws-ec2- describe-security-groups' as X; config where api.name = 'aws-ec2- describe-instances' as Y;filter '\$.X.ipPermissions[*].toPort == 22 and \$.X.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0 and \$.Y.securityGroups[*].groupId == \$.X.groupId' ;show X; </pre>
List RDS instances with a public IP address.	<pre> config where api.name = 'aws-rds- describe-db-instances' and json.rule = publiclyAccessible is true </pre>
List workloads with no tags.	<pre> config where api.name = 'aws- ec2-describe-instances' and </pre>

DESCRIPTION	RQL
	<code>json.rule='\$.tags[*] size == 1 and \$.tags[*].key contains Name'</code>
List Security Groups with egress 0.0.0.0/0 and with no port limitations.	<code>config where api.name = 'aws-ec2-describe-security-groups' AND json.rule = "\$.ipPermissionsEgress[*].ipRanges[*] contains 0.0.0.0/0 and \$.ipPermissions[*].toPort !exists"</code>
List Security Groups with egress 0.0.0.0/0 with fromPort =9009 and no toPort.	<code>config where api.name = 'aws-ec2-describe-security-groups' AND json.rule = "\$.ipPermissionsEgress[*].ipRanges[*] contains 0.0.0.0/0 and \$.ipPermissions[? (@.fromPort==9009)].toPort !exists"</code>
Identify Security Groups with 0.0.0.0/0 configured where toPort is NOT 443.	<code>config where api.name = 'aws-ec2-describe-security-groups' AND json.rule = "\$.ipPermissions[*].ipRanges[*] size > 0 and \$.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0 and (not \$.ipPermissions[? (@.toPort==443)].ipRanges[*] contains 0.0.0.0/0)"</code>
List non-encrypted sda1 and xvda volumes.	<code>config where api.name = 'aws-ec2-describe-volumes' AND json.rule = ' encrypted is false and attachment[*].device !contains sda1'</code> <code>config where api.name = 'aws-ec2-describe-volumes' AND json.rule = ' encrypted is false and attachment[*].device !contains xvda'</code> <code>config where api.name = 'aws-ec2-describe-volumes' AND json.rule = ' encrypted is false and attachment[*].device !contains sda1 and attachment[*].device !contains xvda'</code>
Identify VPC's with Internet Gateway attached.	<code>config where api.name = 'aws-ec2-describe-internet-gateways' as X; config where api.name = 'aws-ec2-describe-vpcs' as Y; filter</code>

DESCRIPTION	RQL
	<pre>'\$.X.attachments[*].vpcId == \$.Y.vpcId and \$.Y.tags[*].key contains IsConnected and \$.Y.tags[*].value contains true'; show Y;</pre>
Find traffic from public IP addresses and in CIDR 169.254.0.0/16, and exclude ICMP and ICMP6 traffic.	<pre>network where src.publicnetwork IN ('Suspicious IPs','Internet IPs') AND source.ip IN 169.254.0.0/16 and bytes > 0 and protocol NOT IN ('ICMP' , 'ICMP6')</pre>
Find workloads with vulnerability 'CVE-2015-5600'.	<pre>network where dest.resource IN (resource where hostfinding.type IN ('Host Vulnerability') AND hostfinding.name = 'CVE-2015-5600') and bytes > 0</pre>
Find membership status of items, such as Redshift nodes that are tagged as members of the stage or production environments.	<pre>config where api.name = 'aws-redshift-describe- clusters' AND json.rule = clusterNodes[*].nodeRole is member of ("stage","prod")</pre>
Find EC2 security groups with IP permissions that allow access to ports other than 443 and 80.	<pre>config where api.name = 'aws-ec2-describe-security- groups' AND json.rule = ipPermissions[*].toPort is not member of (443,80)</pre>
Find "real users" logging in from an IP address to perform root activities; these are not activities performed by automation tasks.	<pre>event where user = 'root' and IP EXISTS</pre>
Find instances that are in subnets that have public IPs auto-assigned.	<pre>config where api.name = 'aws- ec2-describe-instances' as X; config where api.name = 'aws-ec2- describe-subnets' as Y; filter '\$.X.subnetId == \$.Y.subnetId and \$.Y.mapPublicIpOnLaunch is true'; show X;</pre>
Check for bucket exposed publicly that does not have a "Data Classification" tag with a value of "Public".	<pre>config where cloud.type = 'aws' AND api.name='aws-s3api-get-bucket- acl' AND json.rule="(\$.acl.grants[? (@.grantee=='AllUsers')]) size > 0) and websiteConfiguration does not exist and tagSets.DataClassification != Public"</pre>

DESCRIPTION	RQL
Verify that all S3 buckets have a "Data Classification" tag with a valid value.	<p>Custom query to find buckets with no Data Classification tag:</p> <pre>config where cloud.type = 'aws' AND api.name='aws-s3api-get-bucket-acl' AND json.rule= tagSets.DataClassification !exists</pre> <p>Custom query to find buckets with invalid Data Classification tag(s)</p> <pre>config where cloud.type = 'aws' AND api.name='aws-s3api-get-bucket-acl' AND json.rule= tagSets.DataClassification exists and tagSets.DataClassification != Public and tagSets.DataClassification != Private</pre>
Alert on S3 buckets open to AllUsers except for ones with a tagSet of: Data Security: Public or Data Security: blank.	<pre>config where cloud.type = 'aws' AND api.name='aws-s3api-get-bucket-acl' AND json.rule="(\$.acl.grants[? (@.grantee=='AllUsers')]) size > 0) and websiteConfiguration does not exist and (['tagSets']['Data Security'] does not exist or ['tagSets']['Data Security'] does not contain Public)"</pre>
<p>Identify S3 bucket policies that enable write access to a principal who does not belong to an account in your organization.</p> <p>This query helps you find all S3 buckets that allow write action (s3:put) where the Principal Org ID is anything except what you specify in the query.</p>	<pre>config where cloud.type = 'aws' AND api.name = 'aws-s3api-get-bucket-acl' AND json.rule = "policy.Statement[*].Condition.StringEquals.aws does not equal \"o-e9mdyuma56\" and (policy.Statement[? (@.Principal=='*' && @.Effect=='Allow')].Action contains s3:* or policy.Statement[? (@.Principal=='*' && @.Effect=='Allow')].Action contains s3:Put)"</pre>
Alert on all Amazon ELB's (Elastic Load Balancing) that have an expiring certificate.	<p>Custom query for ELBs with certificates that'll expire in less than 90 days:</p> <pre>config where api.name = 'aws-acm-describe-certificate' as X;config where api.name = 'aws-elb-describe-load-balancers' as Y;filter '_DateTime.ageInDays(\$.X.notAfter) > -90 and \$.Y.listenerDescriptions contains \$.X.certificateArn' ; show Y;</pre>

DESCRIPTION	RQL
	<p>Custom query for ELBs with certificates that'll expire in less than 90 days, and with instances attached to ELB:</p> <pre>config where api.name = 'aws-acm-describe-certificate' as X;config where api.name = 'aws-elb-describe-load-balancers' as Y;filter '_DateTime.ageInDays(\$.X.notAfter) > -90 and \$.Y.listenerDescriptions contains \$.X.certificateArn and \$.Y.instances exists' ; show Y;</pre>
Query that looks for SG with 0.0.0.0/0 access and is connected to the running instance.	<pre>config where api.name = 'aws-ec2-describe-instances' as X; config where api.name = 'aws-ec2-describe-security-groups' as Y; filter '\$.X.state.name equals running and \$.X.securityGroups[*].groupId contains \$.Y.groupId and (\$.Y.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0 or \$.Y.ipPermissions[*].ipv6Ranges[*].cidrIpv6 contains ::/0)' ; show X;</pre>
List any AWS instances with GuardDuty or Inspector Vulnerabilities.	<pre>config where hostfinding.type IN ('AWS Inspector Runtime Behavior Analysis', 'AWS Inspector Security Best Practices', 'AWS GuardDuty Host')</pre>
Find someone accessing a specific cloud account, who has assuming a specific role that includes a specific email address.	<p>The account in this example is encsharedtest, the role is AdminSSO and the User email is davidhoffman@abc.com:</p> <pre>event where cloud.account = 'encsharedtest' AND json.rule = \$.userIdentity.arn = 'arn:aws:sts::786215072930:assumed-role/AdminSSO/davidhoffman@abc.com'</pre>

Azure Examples

DESCRIPTION	RQL
Azure workloads with no tags.	<pre>config where api.name = 'azure-vm-list' and json.rule='\$.tags[*] size</pre>

DESCRIPTION	RQL
	<code>== 1 and \$.tags[*].key contains Name'</code>
Azure SQL DB's with Transparent Data Encryption disabled.	<code>config where api.name = 'azure-sql-db-list' and json.rule = transparentDataEncryption is false</code>
Azure SQL instances that allow any IP address to connect to it.	<code>config where cloud.service = 'Azure SQL' AND api.name = 'azure-sql-server-list' AND json.rule = firewallRules[*] contains "0.0.0.0"</code>
Display Azure storage accounts that do not require HTTPS for access.	<code>config where cloud.account = 'Azure-RedLock-public-demo' AND api.name = 'azure-storage-account-list' AND json.rule = ['properties.supportsHttpsTrafficOnly'] is false</code>
Display Azure VM's with Linux OS type in storage profile.	<code>config where cloud.account = 'Azure-RedLock-public-demo' AND api.name = 'azure-vm-list' AND json.rule = ['properties.storageProfile'].osDisk.osType contains "Linux"</code>
List Azure Network Watchers (can be used for Azure flow log checks).	<code>config where cloud.service = 'Azure Network Watcher' AND api.name = 'azure-network-watcher-list' addcolumn provisioningState</code>
List Azure NSGs (can be used for Azure flow log checks).	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nsg-list' addcolumn provisioningState</code>
List Azure Storage accounts (can be used for Azure flow log checks).	<code>config where cloud.type = 'azure' AND api.name = 'azure-storage-account-list' addcolumn location</code>
Show NSGs.	<code>config where cloud.type = 'azure' AND api.name = 'azure-network-nsg-list' addcolumn location name provisioningState securityRules[*]</code>
Instances/VMs Public IP check on Azure.	<code>config where api.name = 'azure-vm-list' AND json.rule = ['properties.networkProfile'].networkInterfaces contains publicIpAddress and</code>

DESCRIPTION	RQL
	<code>['properties.networkProfile'].networkInterfaces none empty</code>
Find all VMs within a specific cloud account that are not running.	<p>This query will include instances that are deallocated, stopped starting, or unknown:</p> <pre>config where cloud.account = 'Azure-RedLock-public-demo' AND api.name = 'azure-vm-list' AND json.rule = powerState does not contain "running"</pre>
Find Azure NSGs that allow inbound traffic.	<pre>config where api.name = 'azure-network-nsg-list' AND json.rule = "securityRules[? (@.sourceAddressPrefix=='*' && @.access=='Allow')].direction contains Inbound"</pre>
Find SQL databases deployed on Azure that are not in the East-US location.	<pre>config where cloud.type = 'azure' AND api.name = 'azure-sql-db-list' AND json.rule = sqlDatabase is not member of ("East US")</pre>

GCP Examples

DESCRIPTION	RQL
GCP (Google Cloud Platform) workloads with no tags.	<pre>config where api.name = 'gcloud-compute-instances-list' and json.rule = '\$.tags[*] size == 1 and \$.tags[*].key contains Name'</pre>
GCP terminated compute instances.	<pre>config where api.name = 'gcloud-compute-instances-list' and json.rule = status contains TERMINATED</pre>
List all VM (Google compute engine) instances that have a public IP address.	<pre>config where api.name = 'gcloud-compute-instances-list' AND json.rule = networkInterfaces[*].accessConfigs[*].natIP size greater than 0 and networkInterfaces[*].accessConfigs[*].natIP none empty</pre>

Common Useful Query Examples

The following are useful queries that can be used as a good base or when you are looking for examples on how complex to make an RQL.

DESCRIPTION	RQL
List all network traffic from the Internet or from Suspicious IPs with over 100Kb data transferred to a network interface (on any cloud environment).	<pre>network where source.publicnetwork IN ('Internet IPs', 'Suspicious IPs') AND bytes > 100000</pre>
All network traffic that is greater than 1GB and destined to Internet or Suspicious IPs (allows you to identify data exfiltration attempt on any cloud environment).	<pre>network where dest.publicnetwork IN ('Internet IPs', 'Suspicious IPs') AND bytes > 1000000000</pre>
All network traffic from Suspicious IPs to instances that have Host Vulnerabilities.	<pre>network where source.publicnetwork = 'Suspicious IPs' AND dest.resource IN (resource where hostfinding.type IN ('AWS GuardDuty Host', 'AWS Inspector Runtime Behavior Analysis', 'AWS Inspector Security Best Practices', 'Host Vulnerability')) AND bytes > 0</pre>
List VPCs that do not have Flow Logs enabled.	<pre>config where api.name = 'aws-ec2- describe-vpcs' as X; config where api.name = 'aws-ec2-describe- flow-logs' as Y; filter ' not (\$.Y.resourceId equals \$.X.vpcId)'; show X;</pre>
List all instances that have a Public IP assigned, and are associated to an NSG that is open to the public.	<pre>config where api.name = 'aws- ec2-describe-instances' as X; config where api.name = 'aws- ec2-describe-security-groups' as Y; filter '(\$.X.publicIpAddress exists and \$.X.publicIpAddress is not empty) and (\$.X.securityGroups[*].groupName == \$.Y.groupName) and (\$.Y.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0 or \$.Y.ipPermissions[*].ipv6Ranges[*].cidrIpv6 contains ::/0)'; show X;</pre>
List all security groups that are open to the public on port 3389 that are on a VPC that contains an IGW.	<pre>config where api.name = 'aws-ec2-describe-security- groups' as X; config where api.name = 'aws-ec2-describe- internet-gateways' as Y; filter '\$.Y.attachments[*].vpcId contains \$.X.vpcId and (\$.X.ipPermissions[?</pre>

DESCRIPTION	RQL
	<pre>(@.toPort==3389 @.fromPort==3389)].ipv6Ranges[*].cidrIpv6 contains ::/0 or \$.X.ipPermissions[? (@.toPort==3389 @.fromPort==3389)].ipRanges[*] contains 0.0.0.0/0 or \$.X.ipPermissions[? (@.toPort>3389&&@.fromPort<3389)].ipv6Ranges[*].cidrIpv6 contains ::/0 or \$.X.ipPermissions[? (@.toPort>3389&&@.fromPort<3389)].ipRanges[*] contains 0.0.0.0/0)'; show X;</pre>
List all security groups that are open to the public on port 22 that are on a VPC that contains an IGW with an EC2 instance attached.	<pre>config where api.name = 'aws-ec2- describe-security-groups' as X; config where api.name = 'aws-ec2- describe-internet-gateways' as Y; config where api.name = 'aws-ec2- describe-instances' as Z; filter '\$.Z.securityGroups[*].groupId contains \$.X.groupId and \$.Y.attachments[*].vpcId contains \$.X.vpcId and (\$.X.ipPermissions[?(@.toPort==22 @.fromPort==22)].ipv6Ranges[*].cidrIpv6 contains ::/0 or \$.X.ipPermissions[?(@.toPort==22 @.fromPort==22)].ipRanges[*] contains 0.0.0.0/0 or \$.X.ipPermissions[? (@.toPort>22&&@.fromPort<22)].ipv6Ranges[*].cidrIpv6 contains ::/0 or \$.X.ipPermissions[? (@.toPort>22&&@.fromPort<22)].ipRanges[*] contains 0.0.0.0/0)'; show X;</pre>
List all security groups that are open to the public, unless they are Tagged as a Mailserver and are open on ports 25, 110, or 443.	<pre>config where api.name = 'aws-ec2-describe-security- groups' AND json.rule = ((ipPermissions[*].ipRanges[*] contains 0.0.0.0/0 or ipPermissions[*].ipv6Ranges[*].cidrIpv6 contains ::/0) and (not (tags[? (@.key=='TYPE')].value contains MAILSERVER AND (((ipPermissions[? (@.toPort>25&&@.fromPort<25)].ipRanges[*] contains 0.0.0.0/0) or (ipPermissions[?(@.toPort==25 @.fromPort==25)].ipRanges[*] contains 0.0.0.0/0)) or ((ipPermissions[? (@.toPort>25&&@.fromPort<25)].ipv6Ranges[*].cidrIpv6 contains ::/0) or (ipPermissions[?(@.toPort==25 @.fromPort==25)].ipv6Ranges[*].cidrIpv6 contains ::/0)) or ((ipPermissions[?</pre>

DESCRIPTION	RQL
	<pre>(@.toPort>443&&@.fromPort<443)].ipRanges[*] contains 0.0.0.0/0) or (ipPermissions[?(@.toPort==443 @.fromPort==443)].ipRanges[*] contains 0.0.0.0/0)) or ((ipPermissions[? (@.toPort>443&&@.fromPort<443)].ipv6Ranges[*].cidrIpv6 contains ::/0) or (ipPermissions[?(@.toPort==443 @.fromPort==443)].ipv6Ranges[*].cidrIpv6 contains ::/0)) or ((ipPermissions[? (@.toPort>110&&@.fromPort<110)].ipRanges[*] contains 0.0.0.0/0) or (ipPermissions[?(@.toPort==110 @.fromPort==110)].ipRanges[*] contains 0.0.0.0/0)) or ((ipPermissions[? (@.toPort>110&&@.fromPort<110)].ipv6Ranges[*].cidrIpv6 contains ::/0) or (ipPermissions[?(@.toPort==110 @.fromPort==110)].ipv6Ranges[*].cidrIpv6 contains ::/0))))))</pre>
Detect AMI images older than 90 days.	<pre>config where cloud.type = 'aws' AND cloud.service = 'EC2' AND api.name = 'aws-ec2- describe-images' AND json.rule = '_DateTime.ageInDays(image.creationDate) > 90'</pre>
Detect EC2 instances running AMIs older than 30 days.	<pre>config where api.name = 'aws- ec2-describe-instances' as X; config where api.name = 'aws- ec2-describe-images' as Y; filter '\$X.imageId==\$Y.image.imageId and _DateTime.ageInDays(\$.Y.image.creationDate) > 30' ; show X; addcolumn launchTime state</pre>
Detect KMS keys with no key rotation.	<pre>config where cloud.type = 'aws' AND api.name = 'aws-kms-get-key- rotation-status' AND json.rule = keyMetadata.keyState does not equal "PendingDeletion" and rotation_status.keyRotationEnabled is false</pre>
Detect CloudFormation Templates (CFTs) that created public Security Groups.	<pre>config where api.name = 'aws- cloudformation-describe- stacks' as X; config where api.name = 'aws-ec2-describe- security-groups' as Y; filter '\$X.stackResources[*].physicalResourceId == \$.Y.groupId and</pre>

DESCRIPTION	RQL
	<pre>(\$.Y.ipPermissions[*].ipv6Ranges[*].cidrIpv6 contains ::/0 or \$.Y.ipPermissions[*].ipRanges[*] contains 0.0.0.0/0)"; show X;</pre>
Detect S3 buckets that are open to Internet but don't contain specific tag key/value pairs.	<pre>config where cloud.type = 'aws' AND api.name='aws-s3api-get-bucket- acl' AND json.rule="(\$.acl.grants[? (@.grantee=='AllUsers')] size > 0) and websiteConfiguration does not exist and (['tagSets']['Name'] does not exist or ['tagSets']. ['Name'] does not contain Value)"</pre>
Detect security groups except for specific tag key/value pairs.	<pre>config where api.name = 'aws- ec2-describe-security-groups' AND json.rule = "tags[? (@.key=='Name')].value does not contain public"</pre>
Find VPC Flow Logs of VPCs that have EC2 instances in it (to verify if there should be network flowlog or not).	<pre>config where api.name = 'aws- ec2-describe-flow-logs' as X; config where api.name = 'aws-ec2- describe-instances' as Y; filter "\$X.resourceId==\$Y.vpcId"; show X;</pre>
Find EC2 instances that are not attached to security groups.	<pre>config where cloud.type = 'aws' AND api.name = 'aws-ec2-describe- security-groups' as X; config where api.name = 'aws-ec2-describe- instances' as Y; filter 'not (\$.Y.securityGroups[*].groupId contains \$.X.groupId)'; show X;</pre>
Find ENIs that are not associated with security groups.	<pre>config where api.name = 'aws-ec2- describe-security-groups' as X; config where api.name = 'aws-ec2- describe-network-interfaces' as Y; filter 'not(\$.Y.groups[*].groupId contains \$.X.groupId or \$.X.groupName == default) '; show X;</pre>