TECHNICAL WHITE PAPER

# Rubrik Exocompute: Granular File Recovery for AWS

Chris Wahl
September 2023
RWP-0528

# Table of Contents

## INTRODUCTION

Indexing and file recovery is an important part of recovering customer data. Rubrik uses the Exocompute framework to extract filesystem index metadata from the Elastic Block Storage (EBS) snapshots created when protecting Elastic Compute Cloud (EC2) instances and EBS volumes in Amazon Web Services (AWS). This metadata is then used when searching for, and recovering, individual files and folders from within these snapshots. In AWS, Exocompute leverages Amazon Elastic Kubernetes Service (EKS) which is a managed Kubernetes service.

### AUDIENCE

This guide is for anyone who wants to better understand the setup, configuration, and troubleshooting of the Exocompute framework for AWS. This includes architects, engineers, and administrators responsible for AWS infrastructure and data protection operations as well as individuals with a vested interest in security, compliance, or governance.

### OBJECTIVES

The goal of this guide is to provide clear guidance on how to design an AWS environment to successfully deploy the file indexing feature from Rubrik. After reading this document, the reader should be able to answer the following questions regarding file indexing for AWS:

- What designs does Rubrik recommend when deploying Exocompute into AWS?
- How does one automate the creation of AWS resources to prepare for Exocompute?
- What steps are necessary to successfully setup Exocompute for a customer deployment?
- How does one see the status and health of Exocompute for file indexing jobs?
- What are common troubleshooting scenarios when dealing with Exocompute in AWS?

## ARCHITECTURE

The Exocompute framework is designed to provide indexing and file recovery features to cloud native workloads. By using an ephemeral cluster of containers to attach, scan, read, and store index metadata, Rubrik is able to offer granular file and folder recovery for a variety of workloads in the public cloud.

> **Note:** For more details on Cloud-Native Protection for AWS on Rubrik's platform and the technical architectures that underpin those capabilities, see *How It Works: Cloud-Native Protection for AWS (RWP-0512)*.

Within AWS, Exocompute uses an EKS Cluster within an Auto Scaling Group to launch worker nodes. These nodes process snapshots to acquire indexing metadata when required. Upon the completion of all indexing tasks, the EKS Cluster is terminated to save on resource costs. The on-demand nature of Exocompute is truly built for the cloud.

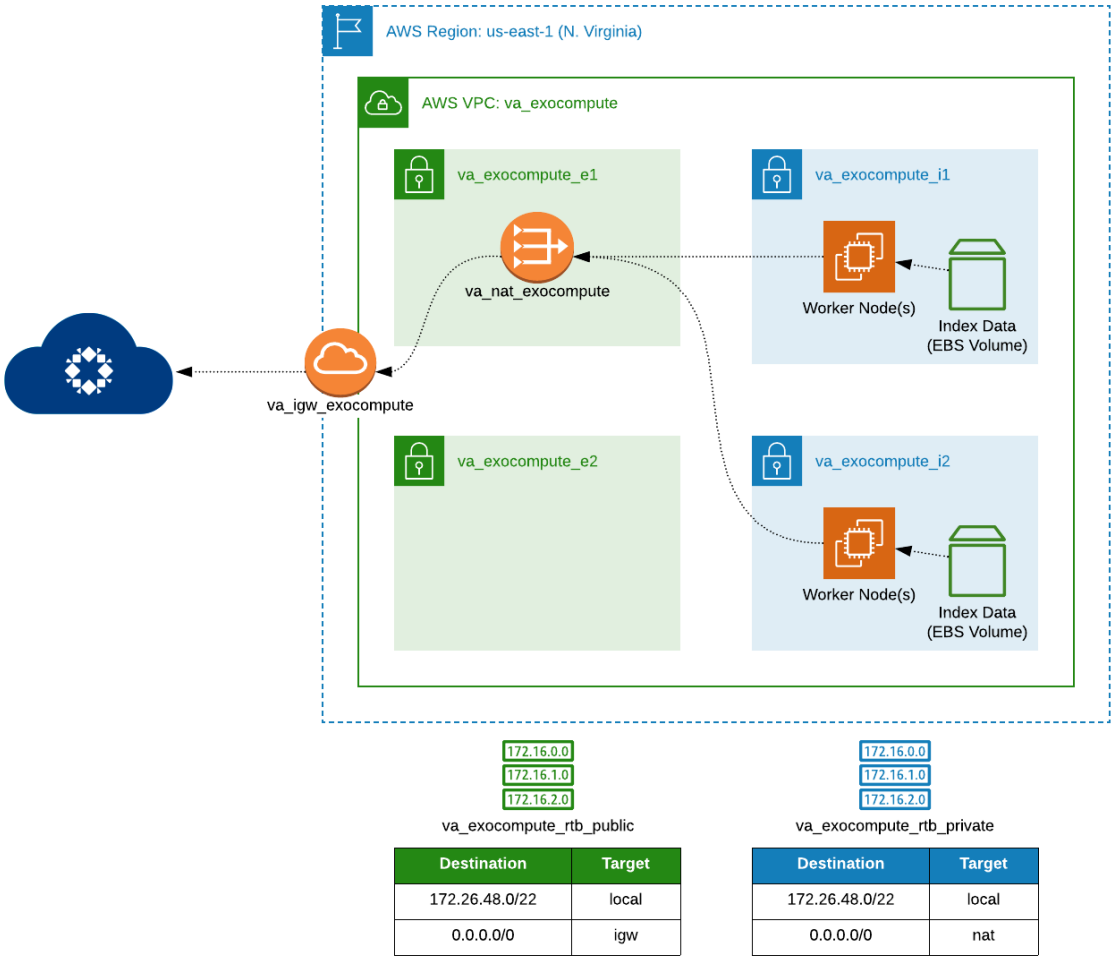
Exocompute can be deployed into a customer AWS account using an existing or new Virtual Private Cloud (VPC). It is recommended that the customer create a new VPC for each region that is expected to host Exocompute resources.

This has several benefits:

1. Network CIDR range can overlap with other deployments in the customer's AWS account(s).
2. Because each EKS worker node can have up to 8 ENIs, and each ENI can have up to 30 IP addresses, a private VPC eliminates the concern over shared VPC CIDR exhaustion when using SNAT.
3. For customers that consider file recovery to be a critical feature, having a dedicated VPC reduces the risk of potential errors from other applications and services.

### SCENARIO: SINGLE VPC

An example customer design scenario for a single Exocompute VPC is shown below.

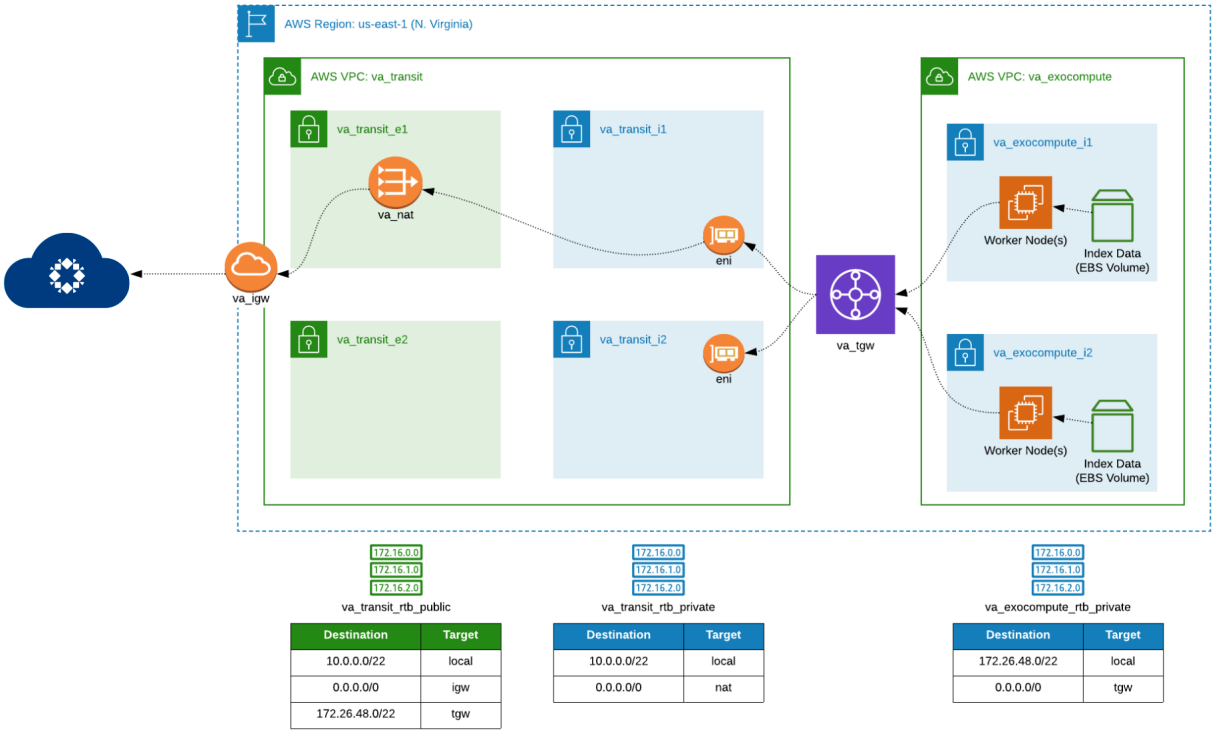



This is a simple design in which all network traffic is completely isolated into a single VPC. Internet connectivity is provided by a NAT gateway located in the `va_exocompute_e1` subnet and associated Availability Zone.

> **Note:** The NAT gateway is highly available and implemented with redundancy by AWS. While the scenario below uses a single NAT gateway to reduce costs, it is suggested that a NAT gateway configured in each Availability Zone to ensure zone-independent architecture in production environments.

## SCENARIO: MULTIPLE VPCS WITH AWS TRANSIT GATEWAY

An example customer design scenario for a single Exocompute VPC attached to an AWS Transit Gateway (TGW) and transit VPC is shown below.




This is a more complex design in which all network traffic is sent to a transit VPC for external access. Internet connectivity is provided by a NAT gateway located in the `va_transit_e1` subnet and associated Availability Zone.

> **Note:** The NAT gateway is highly available and implemented with redundancy by AWS. While the scenario below uses a single NAT gateway to reduce costs, it is suggested that a NAT gateway configured in each Availability Zone to ensure zone-independent architecture in production environments.

## INFRASTRUCTURE AS CODE

It is recommended that customers use an automated tool to generate infrastructure as code (IaC) such as AWS CloudFormation or HashiCorp Terraform.

An example Terraform configuration file for the Single VPC scenario can be found on GitHub here. This will provision all of the AWS resources necessary for the customer to begin the configuration and setup of Exocompute. This includes:

- Creation of the VPC.
- Creation of the public and private subnets.
- Creation of the Internet Gateway (IGW), Elastic IP (EIP), and NAT Gateway (NAT).
- Creation of a private route table with NAT and private subnet association.
- Creation of a public route table with IGW and public subnet association.
- Customer specific tagging.

## PREREQUISITES

To complete the setup of Exocompute, you'll need:

- A Rubrik account.
- An administrative user with access to Rubrik Remote Settings.
- An AWS Cloud Account added to Rubrik for AWS Native Protection.
- An AWS VPC with two subnets that have some form of access to the Internet.

## AWS SETUP

For this paper we are assuming the Single VPC model has been selected. The customer can change any CIDR blocks to meet their corporate networking requirements so long as sufficient private IPv4 addresses are available within the VPC. Example CIDR blocks will be used to demonstrate the scenario.

### VPC AND SUBNET CONFIGURATION

Once an AWS Region has been selected, create the VPC and subnets required by Exocompute.

1. VPC name: `va_exocompute`
    a. IPv4 CIDR block: `172.26.48.0/22`
    b. Tenancy: `Default`

2. Private Subnet 1: `va_exocompute_i1`
    a. IPv4 CIDR block: `172.26.48.0/24`
    b. Availability Zone: `us-east-1c`

3. Private Subnet 2: `va_exocompute_i2`
    a. IPv4 CIDR block: `172.26.49.0/24`
    b. Availability Zone: `us-east-1f`

4. Public Subnet 1: `va_exocompute_e1`
    a. IPv4 CIDR block: `172.26.50.0/24`
    b. Availability Zone: `us-east-1c`

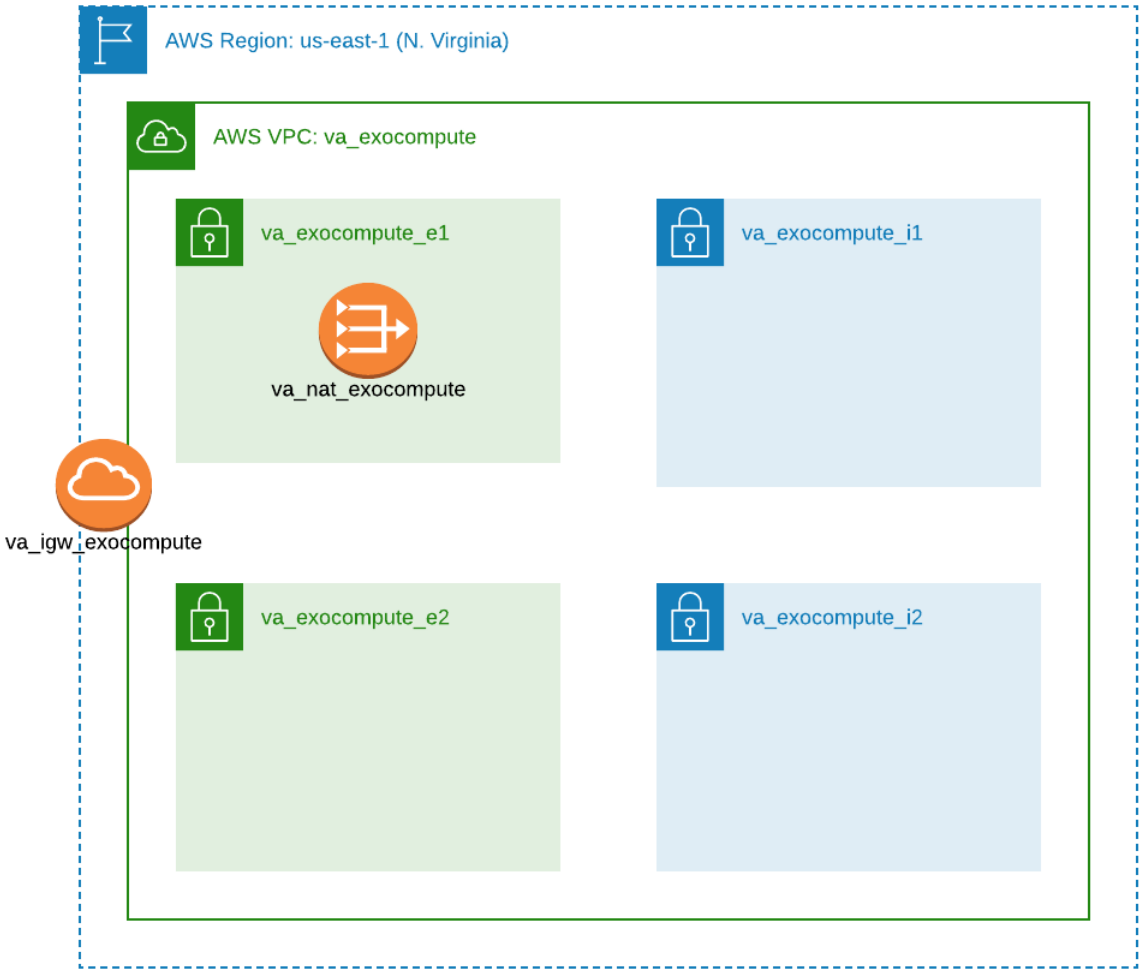
5. Public Subnet 1: `va_exocompute_e2`
    a. IPv4 CIDR block: `172.26.51.0/24`
    b. Availability Zone: `us-east-1f`

The results should be similar to the image below:

| Name | Subnet ID | State | VPC | IPv4 CIDR |
|---|---|---|---|---|
| va_exocompute_i1 | subnet-0ec0c38346716debc | available | vpc-0a1ccf2e527e77a0e \| va_exocompute | 172.26.48.0/24 |
| va_exocompute_i2 | subnet-00ffe5b8a71afad7d | available | vpc-0a1ccf2e527e77a0e \| va_exocompute | 172.26.49.0/24 |
| va_exocompute_e1 | subnet-000ed30a7ef943701 | available | vpc-0a1ccf2e527e77a0e \| va_exocompute | 172.26.50.0/24 |
| va_exocompute_e2 | subnet-08a934c0e120ce88a | available | vpc-0a1ccf2e527e77a0e \| va_exocompute | 172.26.51.0/24 |

### INTERNET GATEWAY, ELASTIC IP, AND NAT GATEWAY SETUP

The next step is to provide Internet access to the VPC. Because the private subnet IPv4 CIDR block contains private RFC 1918 addresses, a NAT gateway is deployed in the public subnet for IPv4 translation purposes.




1. Internet Gateway: `va_igw_exocompute`
    a. Attach: `va_exocompute`
2. Elastic IP: `va_eip_exocompute`
    a. Pool: Amazon pool
3. NAT Gateway:
    a. Subnet: `va_exocompute_e1`
    b. Elastic IP: `va_eip_exocompute`

## ROUTE TABLE CONFIGURATION

The final configuration step is to supply route tables so that traffic understands how to flow through the VPC subnets.

172.16.0.0
172.16.1.0
172.16.2.0

va_exocompute_rtb_public

| Destination | Target |
|---|---|
| 172.26.48.0/22 | local |
| 0.0.0.0/0 | igw |

172.16.0.0
172.16.1.0
172.16.2.0

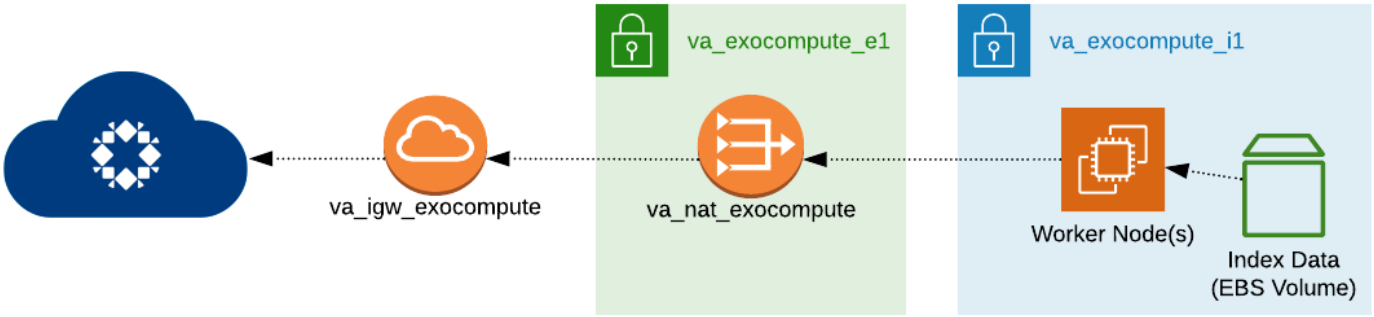
va_exocompute_rtb_private

| Destination | Target |
|---|---|
| 172.26.48.0/22 | local |
| 0.0.0.0/0 | nat |

1. Private Route Table: `va_exocompute_rtb_private`
    a. Subnet Associations: `va_exocompute_i1` & `va_exocompute_i2`
    b. Route: Add destination `0.0.0.0/0` to target `va_nat_exocompute`

2. Private Route Table: `va_exocompute_rtb_public`
    a. Subnet Associations: `va_exocompute_e1` & `va_exocompute_e2`
    b. Route: Add destination `0.0.0.0/0` to target `va_igw_exocompute`

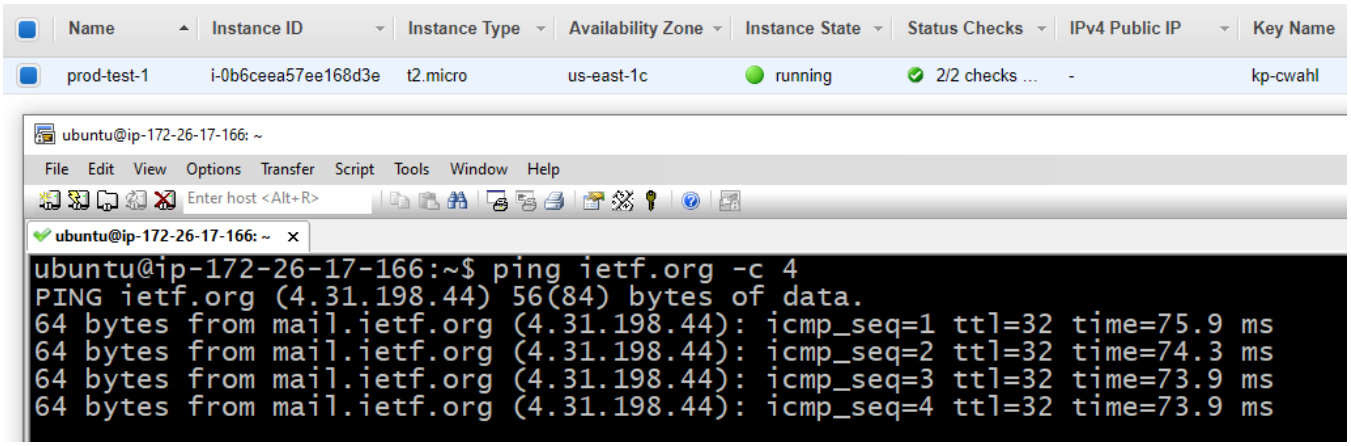
This will connect the VPC subnets together via the routing tables and allow traffic to flow in and out of the VPC.




Traffic will still be filtered by security groups, NACLs, and other forms of security controls found in AWS.

### SIMPLE NETWORK TEST

Deploy a t2.micro EC2 instance into the `va_exocompute_i1` subnet. Temporarily apply a security group to the instance that allows for inbound traffic, such as allowing `0.0.0.0/0` inbound. Try pinging an Internet host that responds to pings, such as `ietf.org`.



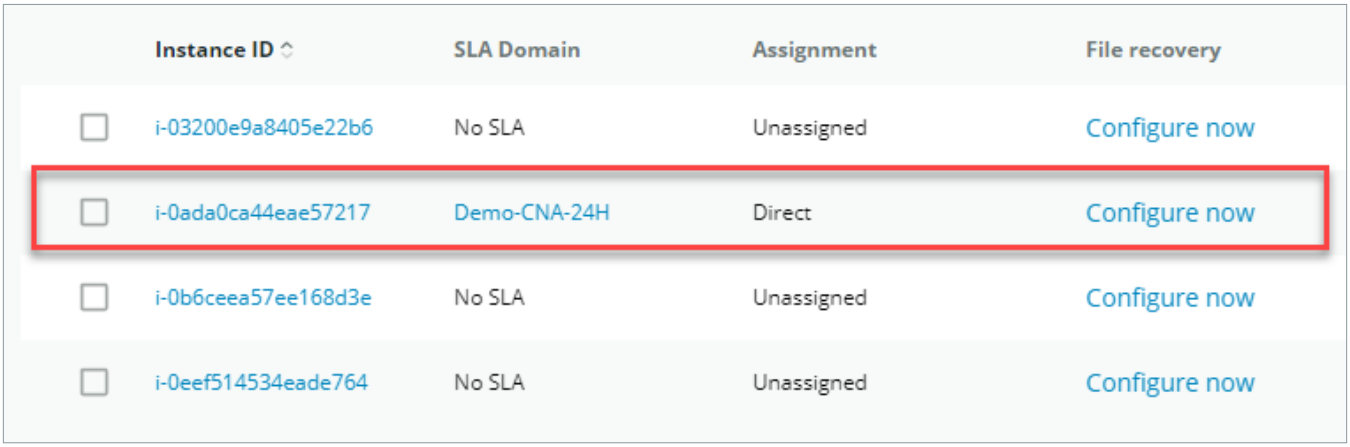

## EXOCOMPUTE SETUP

The setup of Exocompute involves the following steps:

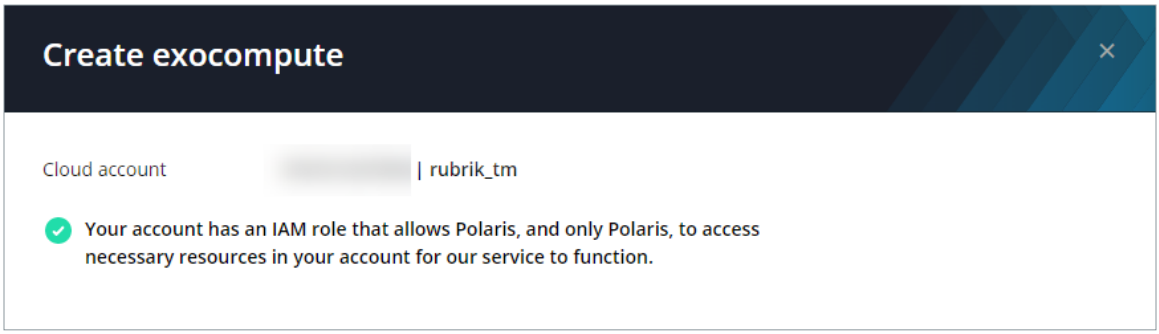
1. Select the VPC(s) for deployment.
2. Provide two subnets for each VPC.
3. Enable File Recovery for any applicable EC2 instances and EBS volumes.

### VPC DEPLOYMENT CONFIGURATION

Customers who look at their Rubrik inventory of **AWS - EC2** objects will see a **File Recovery** status of **Configure Now**. This indicates that resources in AWS can be indexed for file recovery upon the successful configuration of Exocompute.



Clicking on the **Configure Now** link will send the customer to the **Remote Settings > Exocompute Settings** page. The initial configuration screen validates that the Cloud Account is properly configured. Ensure that the desired Cloud Account ID and name match the desired AWS account for file recovery.



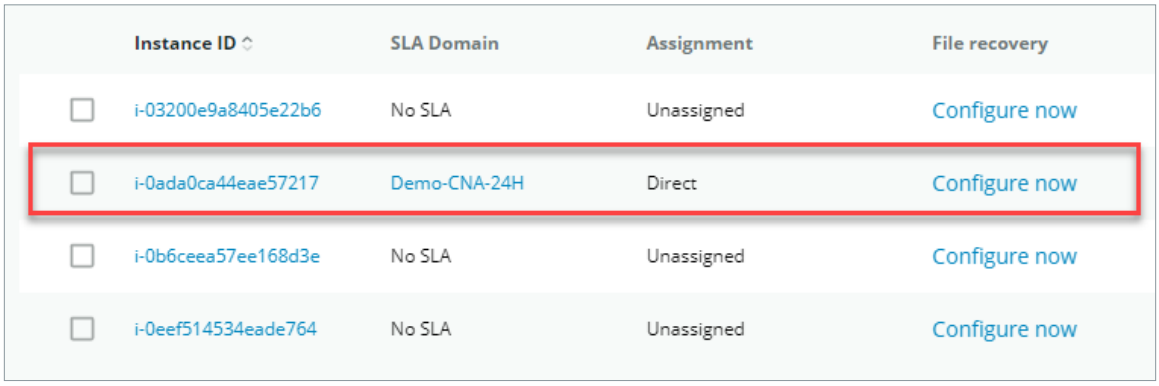

Each AWS Region where Exocompute should be deployed requires a VPC and two subnet values. In the example below, the customer is using the dedicated `va_exocompute` VPC and two internal (private) subnets that will use a NAT gateway to reach the Internet. Rubrik will automatically retrieve the ID values and tagged names.


### EXOCOMPUTE CONFIGURATION VALIDATION

Upon the completion of the Exocompute setup workflow, new settings for Exocompute appear for each AWS Region configured in **Remote Settings > Exocompute Settings**.

| Region | VPC | Subnet 1 | Subnet 2 |
|---|---|---|---|
| US East (N. Virginia) | vpc-0a1ccf2e527e77a0e | subnet-00ffe5b8a71afad7d<br>us-east-1f | subnet-0ec0c38346716debc<br>us-east-1c |

Additionally, the File Recovery column for resources located in the AWS Regions previously configured will display the **Disabled** since the setting has not been configured for these instances.



| | Instance ID | SLA Domain | Assignment | File recovery |
|---|---|---|---|---|
| ☐ | i-03200e9a8405e22b6 | No SLA | Unassigned | Configure now |
| ☐ | i-0ada0ca44eae57217 | Demo-CNA-24H | Direct | Configure now |
| ☐ | i-0b6ceea57ee168d3e | No SLA | Unassigned | Configure now |
| ☐ | i-0eef514534eade764 | No SLA | Unassigned | Configure now |

## EXOCOMPUTE INDEXING

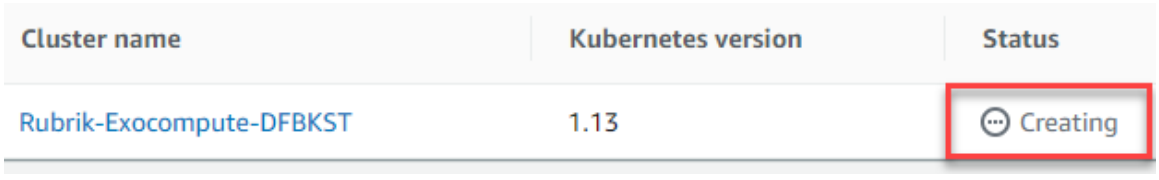
Every two hours, the filesystem indexing workflow begins with a task responsible for identifying snapshots in scope and eligible for indexing. A snapshot is considered eligible if:

- Indexing is enabled on its source EC2 instance or EBS volume.
- Exocompute is configured and functional for the snapshot's account and region.
- The snapshot is unexpired and does not have an expiry hint set.
- The snapshot is currently unindexed.
- The snapshot is not currently marked as unindexable.

Rubrik will not create an EKS Cluster until the above requirements are met. This is by design to reduce resource costs.

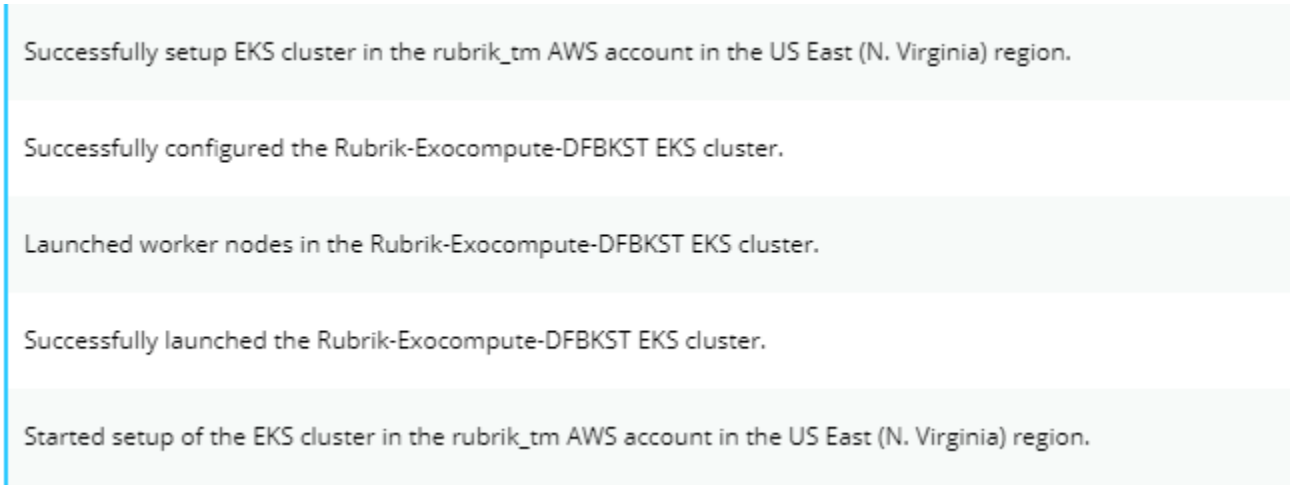
### EXOCOMPUTE EKS CLUSTER CREATION

Once the requirements for an EKS Cluster are met, Exocompute will create a new cluster using the configuration values from the previous section. Each EKS Cluster is configured with a randomized suffix and requires zero customer effort to setup, maintain, or terminate. The EKS Cluster begins in the **Creating** state and will transition into the **Active** state in about 10-15 minutes. This is the AWS perspective of the EKS Cluster health and does not reflect the ability to begin indexing snapshot data.



| Cluster name | Kubernetes version | Status |
|---|---|---|
| Rubrik-Exocompute-DFBKST | 1.13 | Creating |

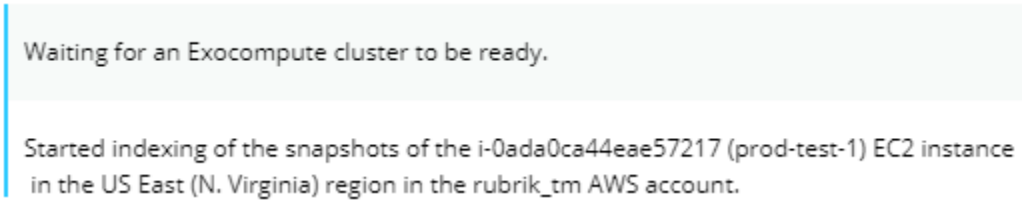
The Rubrik Event Log will show the status of the Exocompute object as it is configured. Use the **Exocompute** object type filter to easily find the logs. The logs below show that the customer environment launched a new EKS Cluster named `Rubrik-Exocompute-DFBKST` based on the need to index one or more snapshots. Once the EKS Cluster is **Active**, Exocompute launches the required quantity of worker nodes and configures the cluster to begin indexing work.




**Note:** Just because an EKS Cluster has a status of **Active** in the AWS console does not mean that it is ready to perform indexing. Until the Rubrik Event Log states "*Successfully setup EKS cluster in the <customer account> AWS account in the <AWS region> region*" the EKS Cluster is not yet ready for indexing work.

### INDEXING SNAPSHOTS

For each resource that has snapshots eligible for indexing, associated events will appear under the **Index** event type. The event status will remain at *Waiting for an Exocompute cluster to be ready* until Exocompute marks the cluster as ready.




Once the EKS Cluster is ready, the awaiting index job associates to the available EKS Cluster. Indexing work proceeds with details shown in the Event Log.

The Index Event Logs include:

- The name of the Exocompute cluster being used.
- The quantity of EBS snapshots being mapped to the cluster's worker node(s).
- Success or failure on a per-snapshot basis.

The example below shows Exocompute indexing a snapshot for an EC2 instance.

Successfully indexed 1 snapshot(s) of the i-0ada0ca44eae57217 (prod-test-1) EC2 instance in the US East (N. Virginia) region in the rubrik_tm AWS account.

1 snapshot(s) are available for file recovery.

Deleted EBS volume(s) for 1 snapshot(s).

Successfully indexed snapshot taken at 08 Apr 20 4:30 PM UTC.

Launched EBS volume(s) for 1 snapshot(s).

Using the Exocompute cluster Rubrik-Exocompute-DFBKST.

Started indexing of the snapshots of the i-0ada0ca44eae57217 (prod-test-1) EC2 instance in the US East (N. Virginia) region in the rubrik_tm AWS account.

### EXOCOMPUTE EKS CLUSTER TERMINATION

Once all indexing work is completed across all eligible snapshots in the customer's AWS Region, Exocompute will terminate the EKS Cluster.

Successfully terminated the Rubrik-Exocompute-DFBKST EKS cluster in the rubrik_tm AWS account in the US East (N. Virginia) region.

Terminating the Rubrik-Exocompute-DFBKST EKS cluster in the rubrik_tm AWS account in the US East (N. Virginia) region.

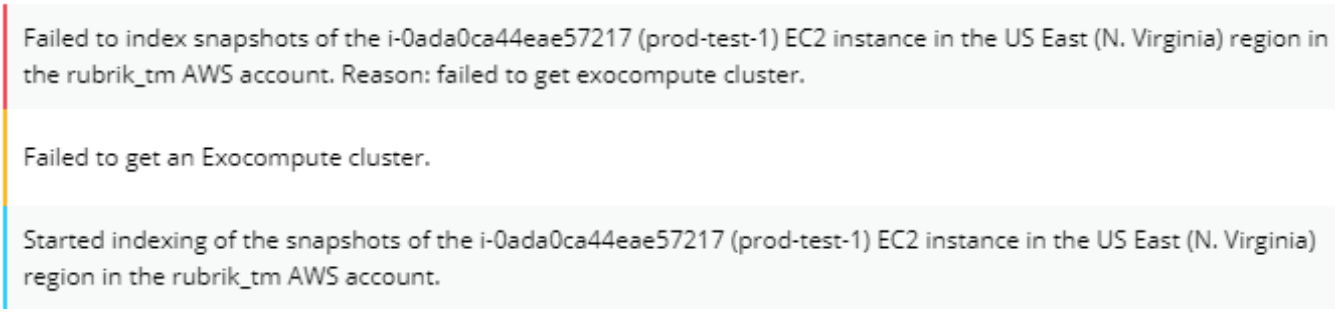
## APPENDIX A: TROUBLESHOOTING

Common issues that are encountered in the field.

### NETWORK CONNECTIVITY FAILURE

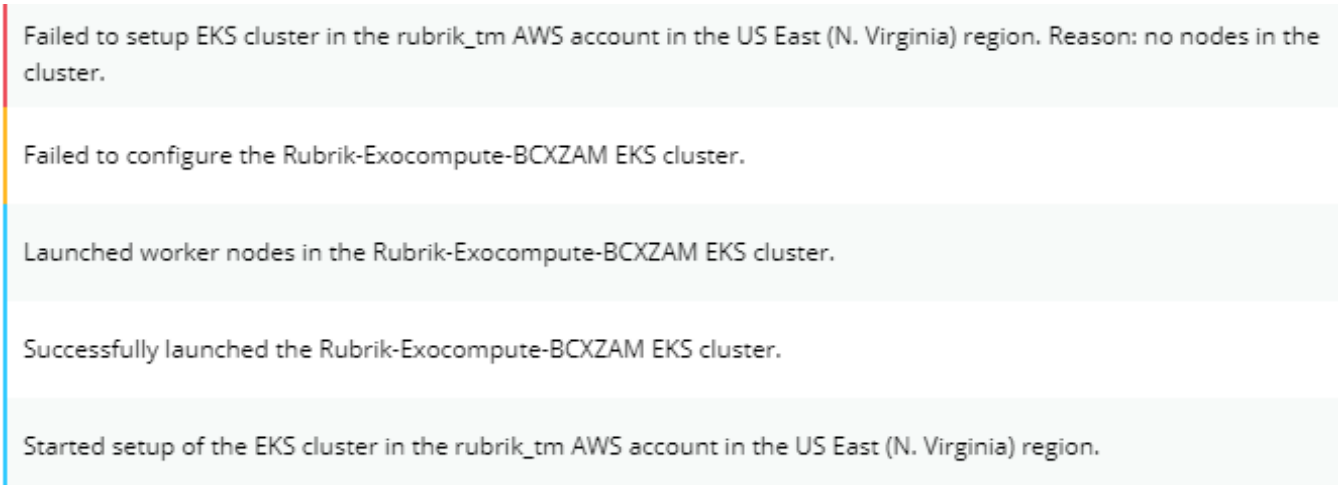
When a resource is unable to acquire an Exocompute EKS Cluster, several caution and error messages will appear under the **Index** event type for the affected resource.

| | |
|---|---|
| Caution | *Failed to get an Exocompute cluster.* |
| Error | *Failed to index snapshots of the <id> EC2 instance in the <AWS region> region in the <customer account> AWS account. Reason: failed to get exocompute cluster.* |




Additionally, there is a Failure status for the Exocompute setup job.

| | |
|---|---|
| Caution | *Failed to configure the Rubrik-Exocompute-ABCDEF EKS cluster.* |
| Error | *Failed to setup EKS cluster in the <customer account> AWS account in the <AWS region> region. Reason: no nodes in the cluster.* |



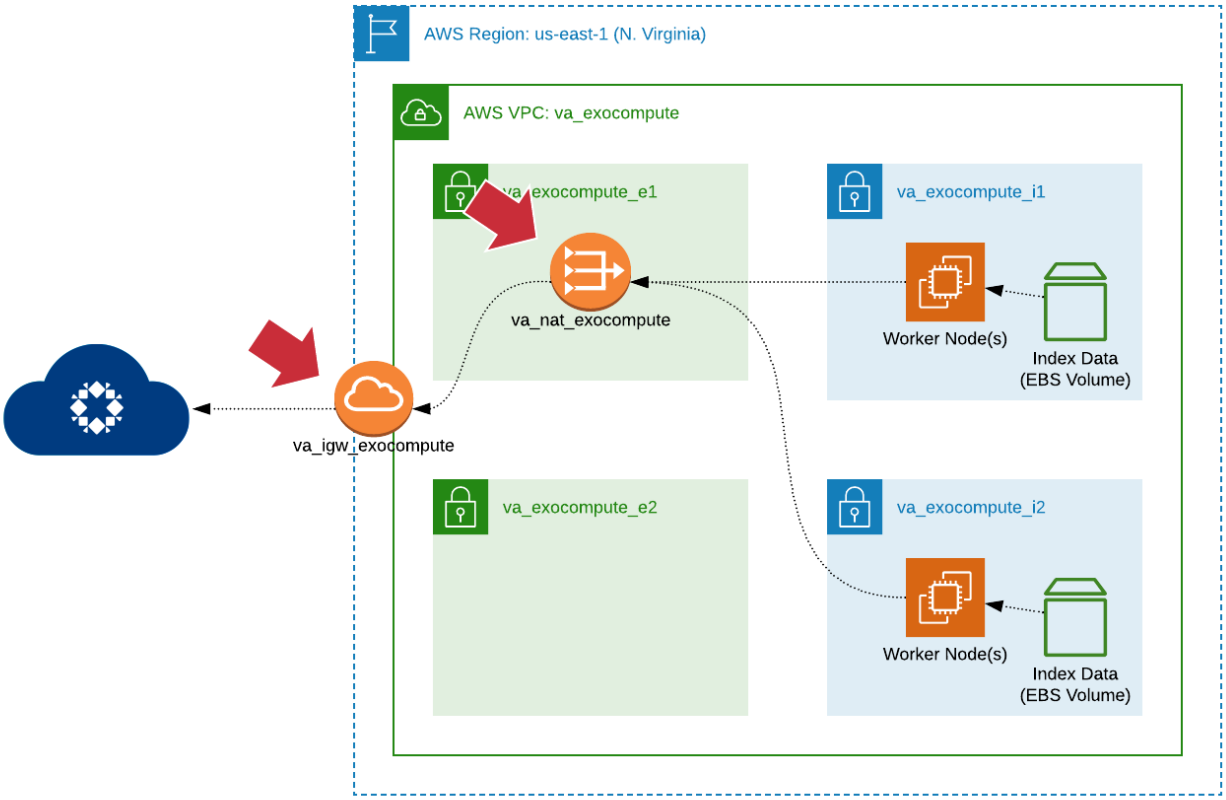
### Resolution

These errors are typically related to a networking misconfiguration in the customer's AWS environment. The worker node(s) are unable to communicate with the Rubrik platform to receive instructions.

The checklists below identify configurations that should be validated to assist with troubleshooting.

### Single VPC Checklist

Troubleshooting for a single, isolated Exocompute VPC.




172.16.0.0
172.16.1.0
172.16.2.0

va_exocompute_rtb_public

| Destination | Target |
|---|---|
| 172.26.48.0/22 | local |
| 0.0.0.0/0 | igw |

172.16.0.0
172.16.1.0
172.16.2.0

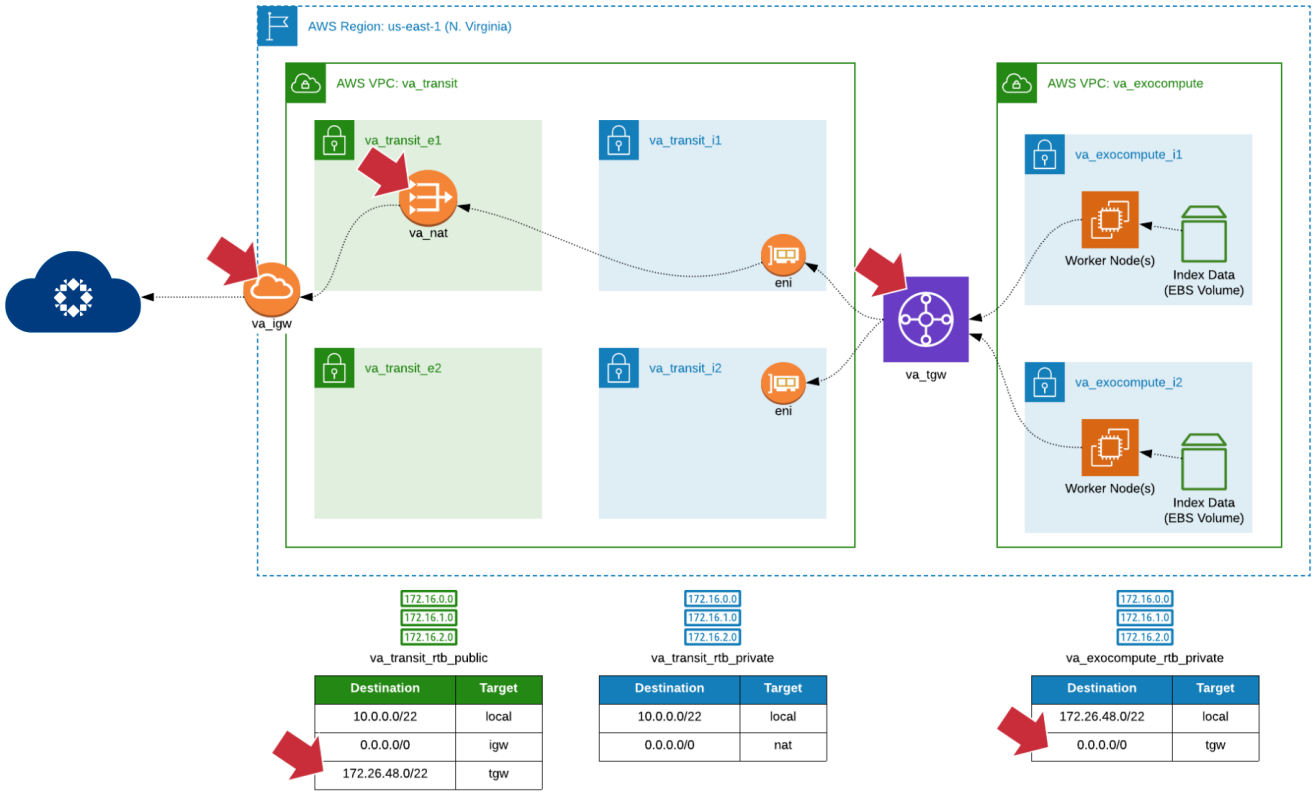
va_exocompute_rtb_private

| Destination | Target |
|---|---|
| 172.26.48.0/22 | local |
| 0.0.0.0/0 | nat |

| Validation | Details |
|---|---|
| Validate that the VPC has an Internet Gateway (IGW) attached to the Exocompute VPC's **public** subnet(s). | Without an IGW, no traffic is allowed to flow to the Internet. There must always be some method for communication with the Rubrik platform. |
| Validate that the VPC has a NAT Gateway (NAT) attached to the Exocompute VPC's **public** subnets. | If the customer places the NAT in the private subnet(s), it will not be able to forward traffic to the Internet. The NAT gateway should live in the public subnets. |
| Validate that the customer's routing tables, including the default gateways, are correctly configured.<br>• The private subnet(s) should forward 0.0.0.0/0 to the NAT.<br>• The public subnet(s) should forward 0.0.0.0/0 to the IGW. | Without a route leading to the IGW, there is no way to establish connectivity to the Rubrik platform and receive indexing instructions.<br>• Private subnets use private IPv4 addresses and must have a default route (0.0.0.0/0) to the NAT gateway.<br>• Public subnets use public IPv4 addresses and must have a default route (0.0.0.0/0) to the IGW gateway. |

**Multiple VPCs with AWS Transit Gateway Checklist**

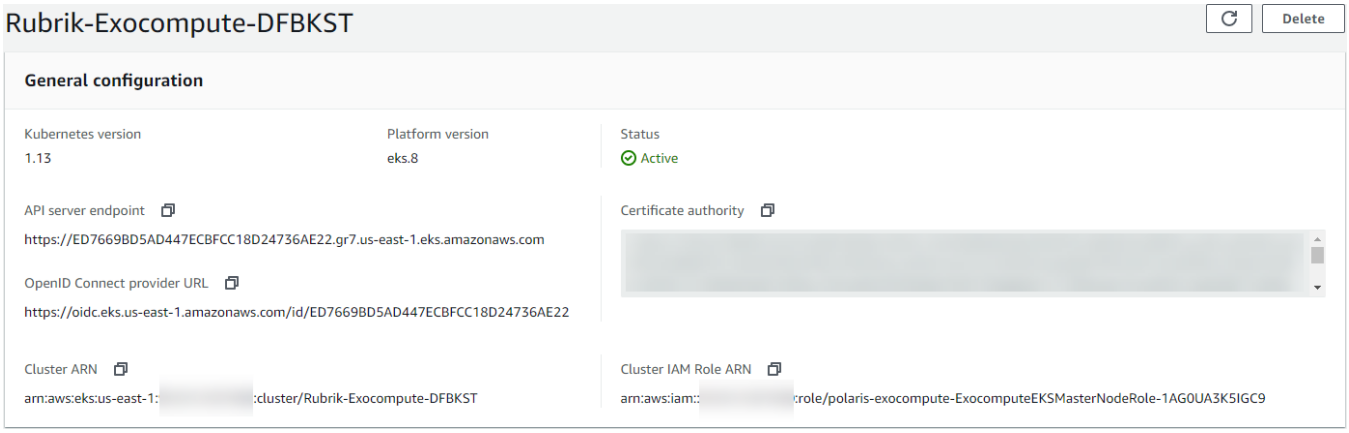
Troubleshooting for a single Exocompute VPC attached to an AWS Transit Gateway and transit VPC.



| Validation | Details |
|---|---|
| Validate that the transit VPC has an operational IGW and NAT attached. | In most environments, the transit VPC is already forwarding traffic for other associated VPCs, making this an easy error to rule out. |
| Validate that the Exocompute VPC is attached to the Transit Gateway (TGW). | If the Exocompute VPC is not attached to the TGW, it will not have a network connection out to the Internet. |
| Validate that the customer's TGW Route Tables are configured to allow routing of traffic from the Exocompute VPC to the transit VPC. | There are two types of route tables in this design: TGW route tables and subnet route tables. The TGW route tables provide routes to the VPCs attached to the TGW. |
| Validate that the customer's Exocompute VPC's **private** subnet(s) are forwarding 0.0.0.0/0 to the TGW. | Because EKS worker nodes only need to communicate with each other, the EKS control plane service, and Rubrik, the default gateway (0.0.0.0/0) should be configured as the TGW. The TGW will then determine the next hop based on the destination address. |
| Validate that the customer's transit VPC's **public** subnet(s) are forwarding the Exocompute VPC network(s) back to the TGW. | Traffic returning from Rubrik must have a route to reach the EKS worker node(s). The TGW route table must contain an entry forwarding traffic back to the Exocompute subnets. |

## APPENDIX B: EKS CLUSTER

An example of an Exocompute EKS Cluster general configuration page is shown below:



## VERSION HISTORY

| Version | Date | Summary of Changes |
|---|---|---|
| 1.0 | April 2020 | Initial Release |
| 1.1 | October 2021 | Updated title and boilerplate |
| 1.2 | September 2023 | Product naming and boilerplate updates |

rubrik

**Global HQ**
3495 Deer Creek Road
Palo Alto, CA 94304
United States

1-844-4RUBRIK
inquiries@rubrik.com
**www.rubrik.com**



rwp-rubrik-exocompute-granular-file-recovery-for-aws / 20230920