

Terraform - Cloud Q on AWS

Supported Updates using Terraform

Dack Busch/Gokul Kuppuraj, Feb 20th, 2022

Terraform Update Options

Note: Carefully review the Terraform plan before committing the apply when doing any updates to the infrastructure.. If you've left **term_protection = true** then you are protected from changes that would harm the cluster. This is why termination protection is on by default.

If you deployed using standard.tfvars a limited set of variable updates are supported with a subsequent Terraform apply denoted by the (*) below. If you want access to all potential variable updates switch to advanced.tfvars.

*** standard.tfvars and advanced.tfvars**

	Add	Del	Change
Total Number of Qumulo EC2 Instances*			increase
Floating IPs for IP Failover			✓
Provision Qumulo Sidecar Lambdas	✓		
Qumulo Sidecar Software Version*			✓
Qumulo Security Group CIDRS	✓	✓	✓
Enable Termination Protection*	✓	✓	✓
OPTIONAL: SNS Topics for automated EC2 and EBS recovery	✓	✓	✓
OPTIONAL: Provision Public IP for Qumulo Management	✓	✓	✓
OPTIONAL: Enable Replication Port for Qumulo Public IP	✓	✓	✓
OPTIONAL: FQDN for R53 Private Hosted Zone	✓	✓	✓
OPTIONAL: R53 Record Name for Qumulo RR DNS	✓	✓	✓
OPTIONAL: Send Qumulo Audit Log messages to CloudWatch Logs?	✓	✓	✓
OPTIONAL: AWS Permissions Boundary Policy Name	✓	✓	✓



Adding Node(s) to the Cluster

A Qumulo cluster may be grown in both capacity and performance by adding additional nodes (EC2 instances) to the cluster. This Terraform supports adding as many as 16 nodes in one subsequent apply for a maximum of 20 nodes total in the cluster. Each node added increases compute, networking, and storage capacity. To add nodes to a cluster follow the procedure below. Note, total instance count may only be increased, not decreased. If total instance count is decreased the Terraform apply will fail assuming termination protection is enabled. **IF YOU DISABLE TERMINATION PROTECTION and DECREASE THE NUMBER OF NODES YOU WILL LOOSE DATA. THIS IS EQUIVALENT TO DESTROYING THE CLUSTER..**

IF you have upgraded the software on the cluster after initial deployment, leave the software version for the cluster in the template as it was originally provisioned. The stack is unaware of this update and the software version field for the cluster can not be used for upgrades after initial deployment.

1. Simply increase `q_node_count` to the total number of EC2 instances desired
2. Terraform plan
3. Review the plan
4. Terraform apply

Terraform will build the new node(s) and the provisioning instance will add them to the cluster. This is not service impacting as the existing nodes are left untouched. There is a brief quorum bounce to add the four node(s) to the cluster. Validate that additional nodes are spinning up in the **EC2 Console**.

Notice that the Provisioning instance is also being rebuilt.. This is by design. The Provisioner will query the latest version of software running on the cluster and upgrade all new node(s) to this version of software before joining them to the cluster. It also updates the floating IPs.

At the completion of node addition you may review any and all of the AWS infrastructure referencing the former section. As a final check make sure the Provisioning node shutdown, which indicates success of all secondary provisioning, and then login to the Qumulo UI and verify the additional nodes are in quorum.



Changing the number of Floating IPs

Terraform may be used to change the number of floating IPs per EC2 instance. Follow the same steps as a Node Addition, but change the **q_floating_ips_per_node** variable to the desired number of floating IPs per instance, 1-4, instead of changing the number of EC2 instances (step 1 above). Note, if DNS for the floating IPs is being managed outside of the Terraform project, the UNC path for clients mounting the cluster will be impacted until DNS is manually updated. To avoid this use the R53 Private Hosted Zone module of this Terraform.

Updating the Sidecar Software Version

Terraform may be used to update the Sidecar software version. Follow the same steps as a Node Addition, but change the **q_sidecar_version** field to the desired version instead of changing the number of EC2 instances (step 1 above). This is typically done after updating the cluster software via the Qumulo UI.

Adding or Removing Qumulo Security Group CIDRs

Terraform may be used to provision additional CIDRs for the Qumulo security group. If a CIDR change is desired modify **q_cluster_additional_sg_cidrs**, which is a comma delimited string in CIDR notation, and execute the Terraform apply. For every CIDR added, all ports in the security group are provisioned with ingress rules. Services allowed are SSH, HTTPS, HTTP, SMB, NFS, FTP, REST, and Qumulo Replication.

Adding or Removing Route53 DNS Private Hosted Zone

It is possible to change the R53 FQDN with Terraform, but AWS requires the deletion of the current Private Hosted Zone and a new one rebuilt if the FQDN is modified.. To remove the private hosted zone, set **q_route53_provision = false**.

Enabling or Disabling Audit Logging

Terraform may be used to enable or disable Qumulo audit logging. These logs are stored in a CloudWatch Logs log group. If an update is used to disable audit logging the log group will be deleted. Likewise, if audit logging is enabled with a Terraform apply, a log group will be created with the name **/qumulo/[deployment-unique-name]-audit-log**.

Adding the Qumulo Sidecar Lambdas



If the Sidecar was not deployed with the Cluster originally, it may be added subsequently with a Terraform apply.. Follow the same steps as a Node Addition, but change the **q_sidecar_provision = true**.

Enabling or Disabling Termination Protection

Terraform may be used to enable or disable Termination Protection for the EC2 instances. Termination protection should be enabled in all production environments. Only disable it with a Terraform apply prior to deleting the stack with a Terraform destroy.

Adding or Removing SNS Topics for recovery alarms

Terraform may be used to add SNS topic ARNs for the EC2 Instance Recovery alarm and the EBS Volume Recovery alarm. These notification ARNs can be added, removed, or changed with using Terraform apply.

Changing EC2 Instance Types

Terraform will allow you to change the EC2 instance type for the cluster. This WILL STOP AND RESTART ALL instances in the cluster. THIS IS PRODUCTION IMPACTING. Downtime is simply the time to reboot the EC2 instances. If you choose to change EC2 instance types in this manner CAREFULLY REVIEW the Terraform plan and triple check that instances are being modified in place. Qumulo recommends shutting down an instance at a time so the cluster can leverage floating IPs and maintain the production workload if you want to change instance types without downtime. Qumulo offers a script for this purpose.

Changing EBS Volume Types

Due to the permutations of EBS volume configurations the likelihood of user error is high attempting to change EBS volume types within Terraform. Rather than risk data loss this is not supported. For EBS volume type changes Qumulo offers a script that is production friendly.

Changing EBS Volume Configurations

If, after initial deployment, you change **q_marketplace_type** and/or **q_disk_config** Terraform will simply ignore these changes. Permitting these changes requires EBS volume and EC2 instance replacement which would result in complete data loss. If you need a different disk configuration, build a new cluster and use [Qumulo Replication](#) to copy the filesystem from the previous cluster.

