

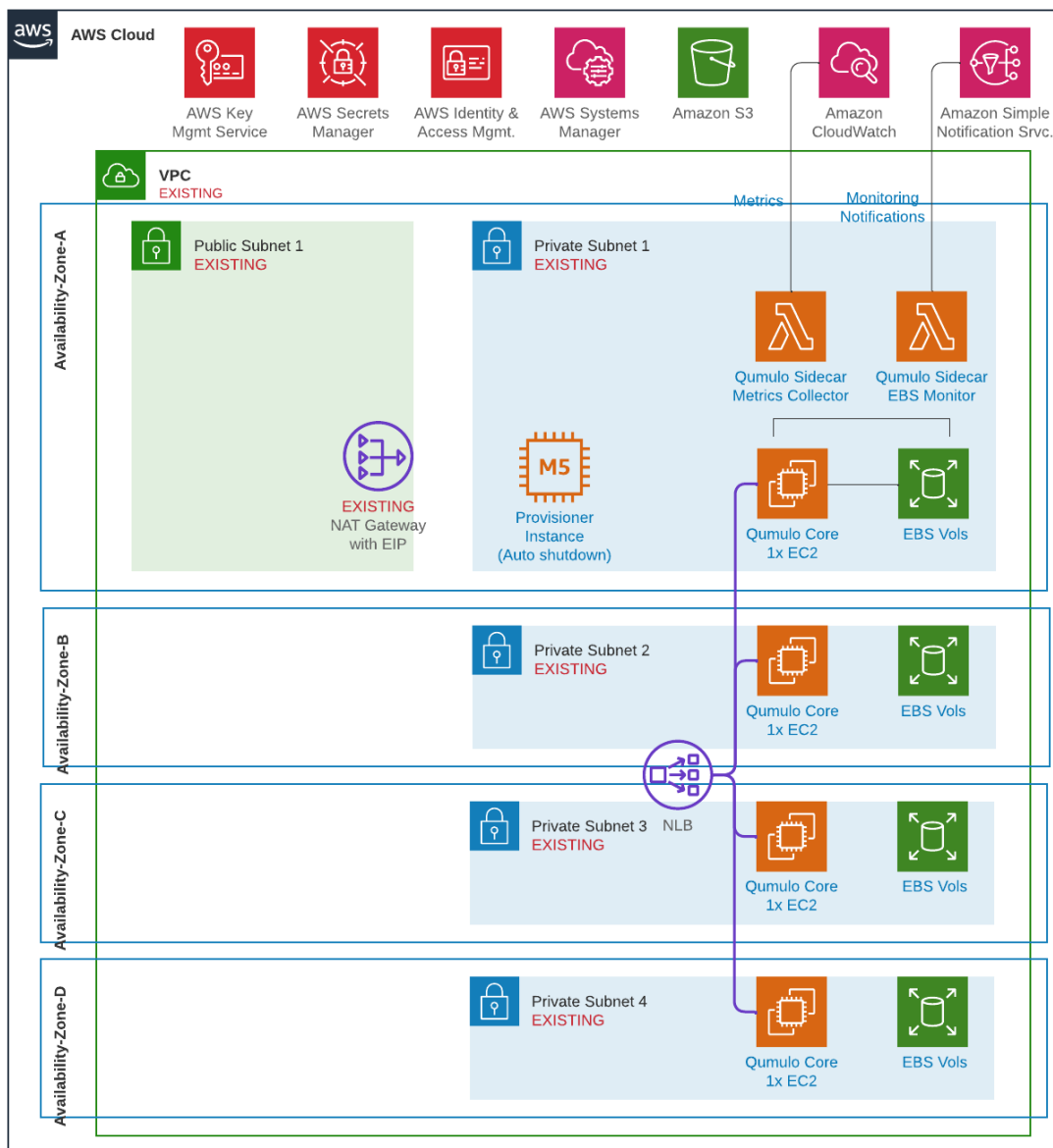
# Terraform - Cloud Q on AWS

## Deploying a Distributed Multi-AZ Cluster

Dack Busch/Gokul Kuppuraj, Nov 14th, 2022

### Overview

A Qumulo cluster may be deployed in a distributed multi-AZ fashion. The reference architecture for this deployment is shown below.



## Benefits of Distributed Multi-AZ

The primary benefit is the enhanced availability of the cluster during AZ outages. Since all Qumulo clusters are instantly consistent, there is no durability benefit, short of a tornado taking out a physical AZ.

- Enhanced availability with single AZ outages
- Simple client failover and restore with an AWS NLB
- Deploy a single cluster - no need to replicate to a second cluster for availability

## Detriments of Distributed Multi-AZ

The primary drawback of a distributed multi-AZ deployment is an impact to performance. The inter-AZ traffic has additional latency. This is most noticeable on small file workloads with up to a 40% impact at the extremes. Large file workloads are minimally impacted by the additional latency.

- Limited Regional Support
- Additional latency - very small file workload impacts
- NFSv3 locking is not reliable with an AWS NLB

## Costs of Distributed Multi-AZ

Generally speaking a distributed multi-AZ cluster is more cost effective than a pair of clusters with replication as long as the average sustained throughput is <500MB/s 24x7. Most workloads fall far below this bar even though they may have peaks that are much higher.

## Supported Regions

A Qumulo cluster requires a minimum of 4 nodes. Thus, 4 AZs are required. This limits the regional support for this architecture on AWS to three regions.

### *Americas*

- us-east-1, US East (N. Virginia) (5 AZs)
- us-west-2, US West (Oregon) (4 AZs)

### *Asia Pacific*

- ap-northeast-2, Asia Pacific (Seoul) (4 AZs)



## Requirements

- Number of AZs: 4
- Number of Nodes in Cluster: 4, 8, or 12
- Load Distribution: AWS NLB

## Deploying a Distributed Multi-AZ Cluster

### Terraform Variables

All the Terraform variables are used in the same manner except:

- Enter four subnets for `q_private_subnet`, comma delimited, one for each AZ
- Rather than specifying the `q_node_count`, specify `q_nodes_per_az`, 1, 2, or 3

### Terraform Deployment

Due to Terraform limitations the NLB cannot be deployed at the same time as the cluster. All the parameters required for the NLB are stored in the AWS SSM Parameter Store so NLB deployment is simple, but it is a second deployment.

- Terraform init and apply the top-level main Terraform code with your tfvars.
- Change to the 'nlb\_qumulo' directory, init and deploy this Terraform code.

## Adding Nodes to a Distributed Multi-AZ Cluster

Nodes must be added in increments of 4 nodes at a time. Further, a cluster may only be grown from 4 nodes to 8 nodes (`q_nodes_per_az=1` to 2). For 3 nodes per AZ a new cluster must be deployed. Here are the steps to add 4 nodes to an existing 4 node cluster, for a total of 8 nodes, 2 per AZ.

1. Set `q_nodes_per_az=2` in your tfvars file
2. Terraform apply
3. Change to the `nlb_qumulo` directory and terraform apply to pickup the new nodes
4. From an EC2 instance with connectivity to the cluster perform the following commands:
  - a. `ssh -i <your keypair.pem> admin@<ip address of any node>`
  - b. `sudo -i`
  - c. `cd /opt/qumulo`
  - d. `./qq_internal login -u admin`
  - e. Enter the admin password
  - f. `./qq_internal increase_node_fault_tolerance`



## Summary of Multi-AZ Configurations

q_nodes_per_az	AZs	# Nodes	Node Protection	AZ Protection	Notes
1	4	4	1	1	Cluster can grow to an 8 node
2	4	8	2	1	
3	4	12	3	1	

