**Nexus IQ Server HA Deployment on AWS EKS with Auto Scaling**

**1. Objective**

Deploy **Sonatype Nexus IQ Server** on **Amazon EKS** in **high availability (HA)** mode with **auto scaling**, using:

* **Amazon RDS (PostgreSQL)** for shared DB
* **Amazon EFS** for shared file system
* **EKS Horizontal Pod Autoscaler (HPA)**
* **AWS ALB Ingress Controller** for routing
* **Secrets** for secure credentials management

**2. Architecture Overview**

**High-Level Components:**

* **EKS**: Runs scalable Nexus IQ server pods
* **RDS (PostgreSQL)**: Shared persistent database
* **EFS**: Shared file storage for IQ data
* **ALB**: Load balances incoming traffic
* **HPA**: Scales IQ pods based on CPU/memory usage

sql

CopyEdit

+-------------+ +-------------------+

| Route 53 |-------> | AWS ALB Ingress |

+-------------+ +-------------------+

|

+----------------+----------------+

| |

+--------v--------+ +--------v--------+

| Nexus IQ Pod(s) | <------------> | Nexus IQ Pod(s) |

+-----------------+ +------------------+

| |

+-------+---------+ +---------+--------+

| EFS (RWX) | | RDS (HA) |

+------------------+ +------------------+

**3. Prerequisites**

* AWS CLI, kubectl, and eksctl
* Existing EKS cluster
* Helm v3
* Nexus IQ license
* AWS Load Balancer Controller installed
* EFS CSI driver installed

**4. Provisioning Resources**

**4.1 Amazon RDS (PostgreSQL)**

* Engine: PostgreSQL 14+
* Enable Multi-AZ
* Example:

bash

CopyEdit

aws rds create-db-instance \

--db-instance-identifier nexus-iq-db \

--engine postgres \

--db-instance-class db.t3.medium \

--allocated-storage 20 \

--master-username iq\_user \

--master-user-password yourpassword \

--vpc-security-group-ids sg-xxxxxxx \

--multi-az \

--backup-retention-period 7

**4.2 Amazon EFS**

* Create EFS & access points across all subnets.
* Enable encryption.
* Use EFS CSI driver to provision RWX volume.

**5. Kubernetes Deployment**

**5.1 Create Namespace**

bash

CopyEdit

kubectl create namespace nexus-iq

**5.2 Create Secrets**

bash

CopyEdit

kubectl create secret generic nexus-iq-license \

--from-file=license.lic -n nexus-iq

kubectl create secret generic db-creds \

--from-literal=username=iq\_user \

--from-literal=password=yourpassword -n nexus-iq

**5.3 Persistent Volume Claim for EFS**

yaml

CopyEdit

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: nexus-iq-pvc

namespace: nexus-iq

spec:

accessModes:

- ReadWriteMany

resources:

requests:

storage: 5Gi

storageClassName: efs-sc

Ensure efs-sc is provisioned using the EFS CSI driver.

**6. Deploy Nexus IQ with Auto Scaling**

**6.1 Deployment YAML**

yaml

CopyEdit

apiVersion: apps/v1

kind: Deployment

metadata:

name: nexus-iq

namespace: nexus-iq

spec:

replicas: 2

selector:

matchLabels:

app: nexus-iq

template:

metadata:

labels:

app: nexus-iq

spec:

containers:

- name: nexus-iq

image: sonatype/nexus-iq-server:latest

ports:

- containerPort: 8070

env:

- name: DATABASE\_URL

value: jdbc:postgresql://<rds-endpoint>:5432/nexus\_iq

- name: DATABASE\_USERNAME

valueFrom:

secretKeyRef:

name: db-creds

key: username

- name: DATABASE\_PASSWORD

valueFrom:

secretKeyRef:

name: db-creds

key: password

volumeMounts:

- name: data

mountPath: /sonatype-work

- name: license

mountPath: /etc/nexus-iq/license.lic

subPath: license.lic

resources:

requests:

cpu: "500m"

memory: "1Gi"

limits:

cpu: "1"

memory: "2Gi"

volumes:

- name: data

persistentVolumeClaim:

claimName: nexus-iq-pvc

- name: license

secret:

secretName: nexus-iq-license

**6.2 Service YAML (with ALB ingress)**

yaml

CopyEdit

apiVersion: v1

kind: Service

metadata:

name: nexus-iq

namespace: nexus-iq

spec:

selector:

app: nexus-iq

ports:

- protocol: TCP

port: 80

targetPort: 8070

type: NodePort

**6.3 Ingress Resource (ALB)**

yaml

CopyEdit

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: nexus-iq-ingress

namespace: nexus-iq

annotations:

alb.ingress.kubernetes.io/scheme: internet-facing

alb.ingress.kubernetes.io/target-type: ip

alb.ingress.kubernetes.io/backend-protocol: HTTP

alb.ingress.kubernetes.io/listen-ports: '[{"HTTP": 80}]'

spec:

ingressClassName: alb

rules:

- host: nexus.yourdomain.com

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: nexus-iq

port:

number: 80

**7. Enable Auto Scaling**

**7.1 Metrics Server (if not installed)**

bash

CopyEdit

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

**7.2 Horizontal Pod Autoscaler (HPA)**

yaml

CopyEdit

apiVersion: autoscaling/v2

kind: HorizontalPodAutoscaler

metadata:

name: nexus-iq-hpa

namespace: nexus-iq

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: nexus-iq

minReplicas: 2

maxReplicas: 5

metrics:

- type: Resource

resource:

name: cpu

target:

type: Utilization

averageUtilization: 60

You can scale based on memory or custom metrics using Prometheus Adapter.

**8. Validation**

* Deploy and access IQ via the ALB DNS or Route53 domain.
* Perform load testing to trigger auto scaling.
* Confirm EFS shared volume and DB persist across pods.
* Kill a pod — verify failover works and session continuity (depending on app logic).

**9. Backup & DR**

* **RDS**: Enable daily snapshots and PITR.
* **EFS**: Use AWS Backup or scheduled lifecycle rules.
* **YAML/Helm**: Store all K8s manifests in Git (GitOps).
* **License**: Back up license.lic in a secure vault.

**10. Security**

* Enable HTTPS (via Ingress + ACM TLS cert)
* Restrict access via NACLs and Security Groups
* Use IRSA for fine-grained IAM
* Encrypt EFS, RDS, and secrets

**11. Monitoring & Observability**

* **CPU/Memory metrics**: HPA via Metrics Server
* **Application Logs**: Send to CloudWatch or ELK
* **Health Checks**: Use /status endpoint
* **Prometheus & Grafana**: For IQ + cluster metrics

Would you like this document as a downloadable PDF/Markdown or want me to generate a **Helm chart**, **Terraform module**, or GitOps-style deployment repo?

**EC2 Instance Recommendations for Nexus IQ HA on AWS EKS**

**⚙️ Instance Selection Strategy**

When deploying **Nexus IQ Server** in HA mode on EKS, the choice of EC2 instance types is critical to ensure consistent performance, scalability, and cost-efficiency. The following factors influence the instance selection:

* **CPU-intensive workloads** (e.g., policy evaluations, reports)
* **Memory requirements** (Nexus IQ caches and scans)
* **IO performance** (EFS, logging, and container startup)
* **Auto scaling support**

**✅ Recommended EC2 Instance Types**

| **Instance Type** | **vCPU** | **Memory (GiB)** | **Notes** |
| --- | --- | --- | --- |
| **m6i.large** | 2 | 8 GiB | Great for small HA clusters, cost-efficient |
| **m6i.xlarge** | 4 | 16 GiB | Balanced compute/memory for medium workloads |
| **c6i.xlarge** | 4 | 8 GiB | More CPU-intensive workloads, faster scans |
| **r6i.xlarge** | 4 | 32 GiB | For larger in-memory IQ caches |
| **m6a or m6g** | - | - | Use Graviton (ARM) or AMD for cost savings — ensure IQ image support |

**💡 Tip:** Prefer m6i or c6i families (latest generation Intel) for best performance and cost balance.

**📊 Resource Guidelines per Pod**

Typical resource requests and limits per **Nexus IQ Pod**:

yaml

CopyEdit

resources:

requests:

cpu: "500m"

memory: "1Gi"

limits:

cpu: "2"

memory: "4Gi"

* You can safely run **2-3 IQ pods** on an m6i.xlarge
* Ensure EKS nodes have enough headroom for OS and kube-system pods

**🔁 Auto Scaling Integration**

Use **Cluster Autoscaler** with **HPA** for optimal elasticity:

* Configure **EKS Managed Node Groups** with **Auto Scaling Groups (ASGs)**
* Enable --balance-similar-node-groups in the Cluster Autoscaler
* Tag ASGs with:

json

CopyEdit

k8s.io/cluster-autoscaler/enabled = true

k8s.io/cluster-autoscaler/<cluster-name> = owned

**🔒 Security & Best Practices**

* Use **EBS volumes with encryption** for kubelet disk storage
* **Limit public IPs** unless necessary
* Use **EC2 Launch Templates** for consistent bootstrap
* Attach **IAM roles to node groups** for EFS/RDS access

**⚖️ Example Recommendation by Load Size**

| **Load Profile** | **Suggested Instance Type** | **# Pods** | **Notes** |
| --- | --- | --- | --- |
| Small Team (1–2k scans/month) | m6i.large | 2 | Cost-optimized |
| Mid-Sized Org (10k scans/month) | m6i.xlarge | 3–4 | Balanced CPU/memory |
| Large Enterprise (20k+ scans/month) | r6i.xlarge | 4–6 | High-memory for report generation |

Would you like a **Terraform module** or **eksctl configuration YAML** with these instance types and autoscaling preconfigured?

**Nexus IQ HA Implementation on AWS EKS (Terraform + Helm + Autoscaling)**

**1. 🎯 Overview**

This document provides a production-grade Nexus IQ Server High Availability (HA) deployment on AWS EKS using:

* **EKS Cluster** (via Terraform)
* **Amazon RDS (PostgreSQL)** for shared DB
* **Amazon EFS** for shared file system
* **Nexus IQ Helm Chart** for deployment
* **HPA (Horizontal Pod Autoscaler)** for workload scaling
* **AWS ALB Ingress** for traffic routing

**2. 🛠 Infrastructure Architecture**

**Components:**

* EKS with managed node groups (EC2-based)
* RDS PostgreSQL (Multi-AZ)
* EFS for /sonatype-work storage
* ALB for ingress
* HPA for workload scaling
* IRSA (IAM Roles for Service Accounts)

**3. 📦 Terraform Modules**

Directory layout:

css

CopyEdit

nexus-iq-ha/

├── main.tf

├── variables.tf

├── outputs.tf

├── eks/

├── rds/

├── efs/

**3.1 EKS Cluster (Terraform)**

Use terraform-aws-eks module:

hcl

CopyEdit

module "eks" {

source = "terraform-aws-modules/eks/aws"

cluster\_name = "nexus-iq-cluster"

cluster\_version = "1.29"

subnet\_ids = module.vpc.private\_subnets

vpc\_id = module.vpc.vpc\_id

node\_groups = {

default = {

desired\_capacity = 3

max\_capacity = 6

instance\_types = ["m6i.xlarge"]

min\_capacity = 2

}

}

}

**3.2 RDS PostgreSQL (Terraform)**

hcl

CopyEdit

resource "aws\_db\_instance" "nexus" {

engine = "postgres"

instance\_class = "db.t3.medium"

allocated\_storage = 20

name = "nexusiq"

username = "iq\_user"

password = var.db\_password

multi\_az = true

skip\_final\_snapshot = true

vpc\_security\_group\_ids = [aws\_security\_group.rds.id]

}

**3.3 EFS + Mount Targets**

hcl

CopyEdit

resource "aws\_efs\_file\_system" "efs" {

encrypted = true

lifecycle\_policy {

transition\_to\_ia = "AFTER\_7\_DAYS"

}

}

Add EFS CSI driver via Helm:

bash

CopyEdit

helm upgrade --install aws-efs-csi-driver \

aws-efs-csi-driver/aws-efs-csi-driver \

--namespace kube-system \

--set controller.serviceAccount.create=false \

--set controller.serviceAccount.name=efs-csi-controller-sa

**4. 🎛 Helm Chart Deployment**

**4.1 Helm Values File (values.yaml)**

yaml

CopyEdit

replicaCount: 2

resources:

limits:

cpu: 2

memory: 4Gi

requests:

cpu: 500m

memory: 1Gi

persistence:

enabled: true

accessMode: ReadWriteMany

storageClass: "efs-sc"

size: 5Gi

env:

- name: DATABASE\_URL

value: jdbc:postgresql://<rds-endpoint>:5432/nexusiq

- name: DATABASE\_USERNAME

valueFrom:

secretKeyRef:

name: db-creds

key: username

- name: DATABASE\_PASSWORD

valueFrom:

secretKeyRef:

name: db-creds

key: password

autoscaling:

enabled: true

minReplicas: 2

maxReplicas: 5

targetCPUUtilizationPercentage: 60

Add efs-sc storage class in your EKS cluster.

**4.2 Deploy with Helm**

bash

CopyEdit

helm upgrade --install nexus-iq sonatype/nexus-iq-server \

--namespace nexus-iq --create-namespace \

-f values.yaml

**5. 📈 Horizontal Pod Autoscaler (HPA)**

Included in values.yaml via Helm:

yaml

CopyEdit

autoscaling:

enabled: true

minReplicas: 2

maxReplicas: 5

targetCPUUtilizationPercentage: 60

Ensure Metrics Server is installed:

bash

CopyEdit

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

**6. 🌐 Ingress (ALB Controller)**

**6.1 Install ALB Controller**

bash

CopyEdit

helm upgrade --install aws-load-balancer-controller \

eks/aws-load-balancer-controller \

--set clusterName=nexus-iq-cluster \

--set serviceAccount.create=false \

--set serviceAccount.name=aws-load-balancer-controller \

-n kube-system

**6.2 Ingress YAML**

yaml

CopyEdit

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: nexus-iq-ingress

namespace: nexus-iq

annotations:

alb.ingress.kubernetes.io/scheme: internet-facing

alb.ingress.kubernetes.io/target-type: ip

spec:

ingressClassName: alb

rules:

- host: nexus.example.com

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: nexus-iq

port:

number: 80

**7. 🛡 IAM & Security**

* Use **IRSA** for EFS + ALB Controller
* Enable TLS via ACM + Ingress
* Store secrets in **AWS Secrets Manager** or K8s secrets
* Enable logging on all components (EKS, RDS, EFS, ALB)

**8. 📊 Monitoring & Alerts**

* Use Prometheus + Grafana
* Enable Container Insights on EKS
* Configure alerts for:
  + Pod restarts
  + HPA trigger events
  + EFS/RDS I/O spikes
  + Ingress failures

**✅ Summary**

| **Component** | **Managed via** |
| --- | --- |
| EKS Cluster | Terraform |
| RDS PostgreSQL | Terraform |
| EFS | Terraform |
| IQ Deployment | Helm |
| Autoscaling | Helm + HPA |
| Ingress (ALB) | Helm + YAML |

**➕ Bonus**

Would you like me to:

* Package this into a **ready-to-deploy GitHub repo**?
* Add a **GitOps-style ArgoCD setup**?
* Include a **CICD pipeline for automatic Helm upgrades**?