Quark Apps Server

AWS Configuration Guide

By: speedy403

Overview:

This guide provides an in-depth overview of how to deploy the Quark Apps Server to AWS using ECS and RDS. There is some additional information regarding EFS, which would be recommended if this application were to be deployed to an actual production environment, due to EFS being persistent. Setup image guides can be found in the GitHub repository.

However, this demo will only use the Ephemeral storage included in ECS. This means upon container shutdown, any data (and therefore apps stored on the server) may be lost. Optionally, feel free to modify the source code in the GitHub repository to store files in S3 opposed to in the container's filesystem. If you don't wish to use EFS (as used in the example deployment), simply skip any sections labeled for EFS and/or refer to the images, which don't use EFS. Additionally, no instructional images will be included for using EFS. You are essentially on your own to set this up as you see fit.

That said, let's get started with the demo setup. →

Security Group for ECS Instance

Inbound Rules:

1. **Web Access**:
    - Protocol: TCP
    - Port Range: 8080
    - Source: 0.0.0.0/0 (or a specific IP range if you want to restrict access)
    - Description: Allow web access on port 8080
    - Protocol: TCP
    - Port Range: 9000
    - Source: (Restrict to a certain IP, i.e. 123.123.123.123/32)
    - Description: Allow web access on port 9000
2. **RDS Access**:
    - Protocol: TCP
    - Port Range: 3306
    - Source: Security group ID of the RDS instance
    - Description: Allow MySQL access from ECS
3. **EFS Access**:
    - Protocol: TCP
    - Port Range: 2049
    - Source: Security group ID of the EFS mount targets
    - Description: Allow NFS access from ECS

Outbound Rules:

1. **Internet Access**:
    - Protocol: All
    - Port Range: All
    - Destination: 0.0.0.0/0
    - Description: Allow all outbound traffic to the internet
2. **RDS Access**:
    - Protocol: TCP
    - Port Range: 3306
    - Destination: Security group ID of the RDS instance
    - Description: Allow outbound MySQL traffic to RDS
3. **EFS Access**:
    - Protocol: TCP
    - Port Range: 2049
    - Destination: Security group ID of the EFS mount targets
    - Description: Allow outbound NFS traffic to EFS

Security Group for RDS Instance

Inbound Rules:

1. **MySQL Access**:
   - Protocol: TCP
   - Port Range: 3306
   - Source: Security group ID of the ECS instance
   - Description: Allow MySQL access from ECS
2. **MySQL Access Public:**
   - Protocol: TCP
   - Port Range: 3306
   - Source: My IP (Set this to your public IP)
   - Description: Allow MySQL from my IP

Outbound Rules:

1. **Default Rule**:
   - Protocol: All
   - Port Range: All
   - Destination: 0.0.0.0/0
   - Description: Allow all outbound traffic

Security Group for EFS

Inbound Rules:

1. **NFS Access**:
   - Protocol: TCP
   - Port Range: 2049
   - Source: Security group ID of the ECS instance
   - Description: Allow NFS access from ECS

Outbound Rules:

1. **Default Rule**:
   - Protocol: All
   - Port Range: All
   - Destination: 0.0.0.0/0
   - Description: Allow all outbound traffic

IAM Roles

Note:

- Due to permissions to use the EFSfile system from the Elastic Container Service,we need to create a new IAM role. However, even without EFS,I recommend adding this role to ECS.

Select trusted entity:

- Trusted entity type: AWSservice
- Use case:
    - Service or use case:Elastic Container Service
    - Use Case:Elastic Container Service Task

Add permissions (Only if using EFS):

- AmazonElasticFileSystemClientFullAccess
- AmazonElasticFileSystemFullAccess

Name, review, and create:

- Role details:
    - Role name: IAMQuarkECSRole
    - Details: Allows ECStasks to connect to the EFSshare.

Creating the Relational Database (RDS)

Database creation method:

- Standard create

Engine options:

- MySQL
- Engine Version
    - MySQL 8.0.40

Templates:

- Free tier (or your choice)

Settings:

- DB instance identifier: quark-db-region (i.e. quark-db-us-east-1)
- Credentials
    - Master username: quark
    - Self managed
    - Master password: (save for later)

Instance configuration:

- db.t4g.micro or similar is fine for most use cases

Storage:

- General Purpose SSD(gp2)
- 20 GiB should be plenty, since the only data being stored is a small amount of information about each file
- Additional Storage configuration:
    - Storage autoscaling: Off

Connectivity:

- Don't connect to an EC2 compute resource
- Network type: IPv4
- Public access: No
- VPC security group
    - Choose existing
    - Quark-RDS-Security-Group

Availability Zone:

- No preference (or your choice)

Database authentication:

- Password authentication

Additional Configuration:

- Initial database name: quark
- Automated backups: Off
- Encryption: Off

Creating the Elastic File System (EFS) - *Optional*

1. Create file system → customize

File system settings:

General:

- Name: quark-filesystem
- File system type: One Zone
  - Availability Zone: Same as before
- Automatic backups: Off
- Transition into Infrequent Access(IA): 30 days
- Transition into Archive: None
- Enable encryption of data at rest: Off

Performance:

- Throughput mode: Enhanced
- Elastic (Recommended)

Network access:

- **Network:**
  - Virtual Private Cloud (VPC):default
  - Mount targets:
    - Availability zone: same as before
    - Subnet ID: leave default
    - IP address: automatic
    - Security groups: Quark-EFS-Security-Group

File system policy

- **Policy options:**
  - Leave all unchecked

Setting Up The Elastic Container Service (ECS)

Create cluster:

Cluster configuration:

- Cluster name: QuarkCluster
- Default namespace:leave default

Infrastructure:

- AWS Fargate (serverless)

Create Task Definition:

Task definition configuration:

- Task definition family: QuarkTaskDefinition

Infrastructure requirements:

- Launch type: AWS Fargate
- OS,Architecture, Network mode:
    - Operating system/Architecture: Linux/X86_64
- Task size:
    - CPU: 1 vCPU
    - Memory: 2 GB
- Task roles:
    - Task role: -
    - Task execution role: IAMQuarkECSRole

Container - 1:

- Container details:
    - Name: quark-webui
    - Image URI:ghcr.io/justinv403/quark-apps-server:latest
    - Essential container: Yes
- Port mappings:
    - Mapping 1:
        - Container port: 8080
        - Protocol: TCP
        - Port name: quark-webui
        - App protocol: HTTP
    - Mapping 2:
        - Container port: 9000
        - Protocol: TCP
        - Port name: quark-admin-webui
        - App protocol: HTTP
    - Mapping 3:
        - Container port: 3306
        - Protocol: TCP
        - Port name: mysql
        - App protocol: none

- Environment variables:
    - Variable 1:
        - Key: PORT
        - Value type: Value
        - Value: 8080
    - Variable 2:
        - Key: ADMIN_PORT
        - Value type: Value
        - Value: 9000
    - Variable 3:
        - Key: HOST
        - Value type: Value
        - Value: 0.0.0.0
    - Variable 4:
        - Key: QUARK_DB_NAME
        - Value type: Value
        - Value: quark
    - Variable 5:
        - Key: QUARK_DB_USER
        - Value type: Value
        - Value: quark
    - Variable 6:
        - Key: QUARK_DB_HOST
        - Value type: Value
        - Value: (same as DB name)
    - Variable 7:
        - Key: QUARK_DB_PASS
        - Value type: Value
        - Value: (the password used in the RDSdatabase)
- Logging:
    - Log collection:
        - Uselog collection: Off

Storage: (Assuming Bind mount method, modify for EFS)

- Volumes:
    - Volume 1:
        - Volume name: QuarkVolume
        - Configuration type: Configure at task definition creation
        - Volume type: Bind mount
- Container mount points:
    - Container: quark-webui
    - Source volume: QuarkVolume
    - Container path: /usr/share/nginx/html/apps
    - Read only: Off

Create Service:

Environment:

- Compute options: Launch type
    - Launch type: FARGATE
    - Platform version: LATEST

Deployment configuration:

- Application type: Service
- Family: QuarkTaskDefinition
- Revision: # (LATEST)
- Service name: QuarkService
- Service type: Replica
- Desired tasks: 1
- Availability Zone rebalancing: Off
- Deployment failure detection:
    - Use the Amazon ECS deployment circuit breaker: On
    - Rollback on failures: Off

Networking:

- VPC: default
- Subnets: leave default
- Security group: Quark-ECS-Security-Group
- Public IP: Turned on