

GEDOPLAN

QUARKUS S3-CLIENT

AMAZON S3

- Simple Storage Service (S3)
- objektbasierter Cloud-Speicher von AWS
- 14. März 2006 gestartet
- Objekte werden in sogenannten Buckets gespeichert
- Objekt setzt sich zusammen aus: Datei + Metadaten + ID



ZUGRIFFSVERFAHREN

Blocking

- URL Connection HttpClient
- Apache HttpClient

```
@Inject  
S3Client s3Client;
```

Non-Blocking (Asynchron)

- Netty HTTP Client
- AWS CRT-HTTP Client

```
@Inject  
S3AsyncClient s3AsyncClient;
```

DEV SERVICES

- Docker-Container mit LocalStack
- AWS-Cloud-Services lokal auf dem Rechner testen
 - S3
 - Lambda
 - DynamoDB
 - ...
- Keine Gebühren während der Entwicklung
- Schnelle Entwicklungszyklen durch sofortiges Feedback

The screenshot shows the documentation page for 'Dev services for Amazon Services' in the Quarkiverse Docs. The page title is 'Dev services for Amazon Services'. A warning box states: 'Dev Services for Amazon Services does not verify if the chosen stack actually implements a mock for a given service. It will start the client to connect to it, regardless of whether the service is actually supported by that stack provider.' Below this, a code block shows configuration examples for global and per-service settings.

Amazon Services

- ACM
- Api Gateway Management Api
- Api Gateway
- Api Gateway v2
- App Config
- App Config Data
- Bedrock
- Bedrock Runtime
- CloudWatch
- CloudWatch Logs
- Cognito User Pools
- DynamoDB
- ECR
- Elastic Load Balancing
- Elastic Load Balancing v2
- EventBridge
- EventBridge Scheduler
- IAM
- Inspector
- Inspector2
- IoT
- Kinesis
- KMS
- Lambda
- Neptune
- Payment Cryptography
- Payment Cryptography Data
- Amazon Services

Dev services for Amazon Services

Quarkus Amazon Services automatically starts a containerized AWS stack in dev mode and when running tests. So, you don't need a client is configured automatically.

Choosing a Stack Provider

Dev Services for Amazon Services supports multiple stack providers:

- LocalStack: The default stack provider, using the LocalStack container
- MiniStack: An alternative stack provider
- Moto: A lightweight AWS service mock library, using the motoserver container
- Floci: Another alternative stack provider

WARNING

Dev Services for Amazon Services does not verify if the chosen stack actually implements a mock for a given service. It will start the client to connect to it, regardless of whether the service is actually supported by that stack provider.

You can configure the stack provider globally or per-service:

```
# Global configuration (default: localstack)
quarkus.aws.devservices.provider=localstack

# Per-service configuration (overrides global setting)
quarkus.dynamodb.devservices.provider=ministack
quarkus.cognito-user-pools.devservices.provider=moto
quarkus.s3.devservices.provider=floci
```

VERWENDUNG VON REQUESTS

- Kommunikation mit AWS S3 erfolgt über Requests
- Request zum Hochladen eines Objekts in einen S3-Bucket

```
PutObjectRequest request = PutObjectRequest.builder()  
    .bucket(bucketName)  
    .key(objectKey)  
    .build();
```

- Verwendung des Clients zum Senden des Requests
- Übergabe der zu speichernden Objekte

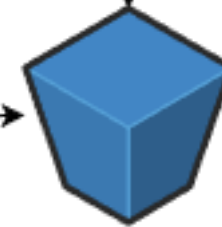
```
s3Client.putObject(request, RequestBody.fromBytes(content));
```

METADATEN



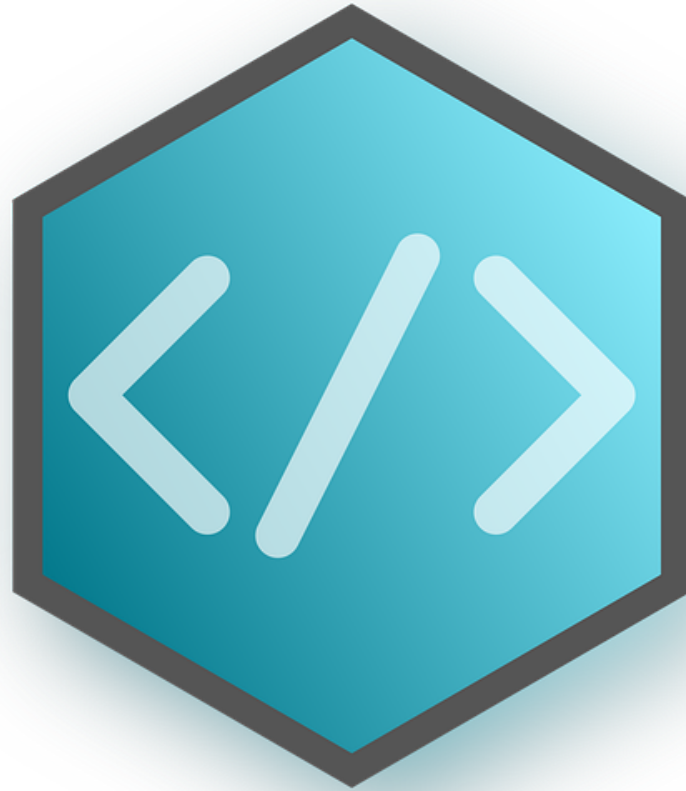
Metadaten

Ort:	Schweden
erstellt:	07.07.2022 16:10:58



S3 Bucket

DEMO



WEITERE MÖGLICHKEITEN

- Verwendung von min.io im Kubernetes-Cluster
- Nutzung von Amazon S3 Transfer Manager
- Event-getriebene Verarbeitung: S3-Events (SNS/SQS/Lambda)
- Asynchrone Verarbeitung mit S3AsyncClient
- ...

FRAGEN?

