EcoShower - AWS Installation Guide

מדריך התקנה מלא

תוכן העניינים

1. דרישות מקדימות

1.1 חשבון AWS

- חשבון AWS פעיל עם הרשאות Administrator
- AWS CLI מותקן ומוגדר
- Node.js 18+ ו-Python 3.11+ מותקנים

1.2 כלים נדרשים

```
# התקנת AWS CLI
pip install awscli

# הגדרת credentials
aws configure
# AWS Access Key ID: [your-key]
# AWS Secret Access Key: [your-secret]
# Default region: eu-north-1 (Stockholm)
# Account ID: From aws sts get-caller-identity
# Bucket Name: Unique name
# User Pool Name: EcoShower-Users

# 1. Set Environment Variables
export AWS_REGION=eu-north-1
export PROJECT_NAME=ecoshower
export ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output
text)
```

1.3 משתנים גלובליים

```
export AWS_REGION=eu-north-1
export PROJECT_NAME=ecoshower
export ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output
text)
```

---

2. יצירת DynamoDB Tables

2.1 טבלת Users

```
aws dynamodb create-table \
    --table-name EcoShower-Users \
    --attribute-definitions \
        AttributeName=user_id,AttributeType=S \
        AttributeName=email,AttributeType=S \
    --key-schema \
        AttributeName=user_id,KeyType=HASH \
    --global-secondary-indexes \
        "[{
            \"IndexName\": \"email-index\",
            \"KeySchema\":
[{\"AttributeName\":\"email\",\"KeyType\":\"HASH\"}],
            \"Projection\": {\"ProjectionType\":\"ALL\"}
        }]" \
    --billing-mode PAY_PER_REQUEST \
    --region $AWS_REGION
```

2.2 טבלת Devices

```
aws dynamodb create-table \
    --table-name EcoShower-Devices \
    --attribute-definitions \
        AttributeName=device_id,AttributeType=S \
        AttributeName=user_id,AttributeType=S \
    --key-schema \
        AttributeName=device_id,KeyType=HASH \
    --global-secondary-indexes \
        "[{
            \"IndexName\": \"user-index\",
            \"KeySchema\":
[{\"AttributeName\":\"user_id\",\"KeyType\":\"HASH\"}],
            \"Projection\": {\"ProjectionType\":\"ALL\"}
        }]" \
    --billing-mode PAY_PER_REQUEST \
    --region $AWS_REGION
```

2.3 טבלת Sessions

```
aws dynamodb create-table \
    --table-name EcoShower-Sessions \
    --attribute-definitions \
        AttributeName=session_id,AttributeType=S \
        AttributeName=device_id,AttributeType=S \
        AttributeName=start_time,AttributeType=S \
    --key-schema \
        AttributeName=session_id,KeyType=HASH \
    --global-secondary-indexes \
        "[{
            \"IndexName\": \"device-index\",
            \"KeySchema\": [
                {\"AttributeName\":\"device_id\",\"KeyType\":\"HASH\"},
                {\"AttributeName\":\"start_time\",\"KeyType\":\"RANGE\"}
            ],
            \"Projection\": {\"ProjectionType\":\"ALL\"}
        }]" \
    --billing-mode PAY_PER_REQUEST \
    --region $AWS_REGION
```

2.4 טבלת Telemetry

```
aws dynamodb create-table \
    --table-name EcoShower-Telemetry \
    --attribute-definitions \
        AttributeName=device_id,AttributeType=S \
        AttributeName=timestamp,AttributeType=S \
    --key-schema \
        AttributeName=device_id,KeyType=HASH \
        AttributeName=timestamp,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --region $AWS_REGION
```

2.5 הפעלת Point-in-Time Recovery

```
for table in Users Devices Sessions Telemetry; do
    aws dynamodb update-continuous-backups \
        --table-name EcoShower-$table \
        --point-in-time-recovery-specification
PointInTimeRecoveryEnabled=true \
        --region $AWS_REGION
done
```

---

3. הגדרת Cognito

3.1 יצירת User Pool

```bash
# יצירת User Pool
aws cognito-idp create-user-pool \
    --pool-name EcoShower-Users \
    --policies '{
        "PasswordPolicy": {
            "MinimumLength": 8,
            "RequireUppercase": true,
            "RequireLowercase": true,
            "RequireNumbers": true,
            "RequireSymbols": true
        }
    }' \
    --auto-verified-attributes email \
    --username-attributes email \
    --schema '[
        {"Name": "email", "Required": true, "Mutable": true},
        {"Name": "name", "Required": true, "Mutable": true},
        {"Name": "custom:role", "AttributeDataType": "String", "Mutable":
true}
    ]' \
    --region $AWS_REGION

# שמור את ה-Pool ID
export USER_POOL_ID=$(aws cognito-idp list-user-pools --max-results 10 \
    --query "UserPools[?Name=='EcoShower-Users'].Id" --output text)

echo "User Pool ID: $USER_POOL_ID"
```

3.2 יצירת App Client

```bash
aws cognito-idp create-user-pool-client \
    --user-pool-id $USER_POOL_ID \
    --client-name EcoShower-WebApp \
    --generate-secret false \
    --explicit-auth-flows \
        ALLOW_USER_PASSWORD_AUTH \
        ALLOW_REFRESH_TOKEN_AUTH \
        ALLOW_USER_SRP_AUTH \
    --supported-identity-providers COGNITO \
    --region $AWS_REGION

# שמור את ה-Client ID
export CLIENT_ID=$(aws cognito-idp list-user-pool-clients \
    --user-pool-id $USER_POOL_ID \
    --query "UserPoolClients[0].ClientId" --output text)
```

```
echo "Client ID: $CLIENT_ID"
```

3.3 יצירת קבוצות

```
# קבוצת מנהלים
aws cognito-idp create-group \
    --user-pool-id $USER_POOL_ID \
    --group-name admins \
    --description "System administrators" \
    --region $AWS_REGION

# קבוצת משתמשים
aws cognito-idp create-group \
    --user-pool-id $USER_POOL_ID \
    --group-name users \
    --description "Regular users" \
    --region $AWS_REGION
```

3.4 יצירת משתמש מנהל ראשוני

```
# יצירת משתמש
aws cognito-idp admin-create-user \
    --user-pool-id $USER_POOL_ID \
    --username admin@ecoshower.com \
    --user-attributes \
        Name=email,Value=admin@ecoshower.com \
        Name=name,Value="System Admin" \
        Name=email_verified,Value=true \
        Name=custom:role,Value=admin \
    --temporary-password "TempPass123!" \
    --region $AWS_REGION

# הוספה לקבוצת מנהלים
aws cognito-idp admin-add-user-to-group \
    --user-pool-id $USER_POOL_ID \
    --username admin@ecoshower.com \
    --group-name admins \
    --region $AWS_REGION
```

---

4. יצירת Lambda Functions

4.1 יצירת IAM Role ל-Lambda

```
```

```
# יצירת trust policy
cat > lambda-trust-policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {"Service": "lambda.amazonaws.com"},
        "Action": "sts:AssumeRole"
    }]
}
EOF

# יצירת Role
aws iam create-role \
    --role-name EcoShower-LambdaRole \
    --assume-role-policy-document file://lambda-trust-policy.json

# הוספת policies
aws iam attach-role-policy \
    --role-name EcoShower-LambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-
role/AWSLambdaBasicExecutionRole

aws iam attach-role-policy \
    --role-name EcoShower-LambdaRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess

aws iam attach-role-policy \
    --role-name EcoShower-LambdaRole \
    --policy-arn arn:aws:iam::aws:policy/AWSIoTDataAccess

aws iam attach-role-policy \
    --role-name EcoShower-LambdaRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSNSFullAccess

export LAMBDA_ROLE_ARN=$(aws iam get-role --role-name EcoShower-LambdaRole
\
    --query 'Role.Arn' --output text)
```

4.2 יצירת Lambda - Process Telemetry

```
# ארוז את הקוד
cd src/lambda
zip process_telemetry.zip process_telemetry.py

# צור את ה-Lambda
aws lambda create-function \
    --function-name EcoShower-ProcessTelemetry \
    --runtime python3.11 \
    --role $LAMBDA_ROLE_ARN \
    --handler process_telemetry.lambda_handler \
    --zip-file fileb://process_telemetry.zip \
    --timeout 30 \
    --memory-size 256 \
    --environment "Variables={
        TELEMETRY_TABLE=EcoShower-Telemetry,
        DEVICES_TABLE=EcoShower-Devices,
        SESSIONS_TABLE=EcoShower-Sessions,
        SNS_TOPIC_ARN=arn:aws:sns:$AWS_REGION:$ACCOUNT_ID:EcoShower-
Notifications
    }" \
    --region $AWS_REGION
```

4.3 יצירת Lambda - API Handler

```
zip api_handler.zip api_handler.py

aws lambda create-function \
    --function-name EcoShower-API \
    --runtime python3.11 \
    --role $LAMBDA_ROLE_ARN \
    --handler api_handler.lambda_handler \
    --zip-file fileb://api_handler.zip \
    --timeout 30 \
    --memory-size 256 \
    --environment "Variables={
        USERS_TABLE=EcoShower-Users,
        DEVICES_TABLE=EcoShower-Devices,
        SESSIONS_TABLE=EcoShower-Sessions,
        TELEMETRY_TABLE=EcoShower-Telemetry
    }" \
    --region $AWS_REGION
```

---

5. הגדרת API Gateway

5.1 יצירת REST API

```
# יצירת API
aws apigateway create-rest-api \
    --name EcoShower-API \
    --description "EcoShower REST API" \
    --endpoint-configuration types=REGIONAL \
    --region $AWS_REGION

export API_ID=$(aws apigateway get-rest-apis \
    --query "items[?name=='EcoShower-API'].id" --output text)

export ROOT_ID=$(aws apigateway get-resources --rest-api-id $API_ID \
    --query 'items[?path==`/`].id' --output text)

echo "API ID: $API_ID"
```

5.2 יצירת Cognito Authorizer

```
aws apigateway create-authorizer \
    --rest-api-id $API_ID \
    --name CognitoAuth \
    --type COGNITO_USER_POOLS \
    --provider-arns "arn:aws:cognito-
idp:$AWS_REGION:$ACCOUNT_ID:userpool/$USER_POOL_ID" \
    --identity-source 'method.request.header.Authorization' \
    --region $AWS_REGION

export AUTHORIZER_ID=$(aws apigateway get-authorizers --rest-api-id
$API_ID \
    --query 'items[0].id' --output text)
```

5.3 יצירת Resources ו-Methods

```
# פונקציה ליצירת resource ו-method
create_resource() {
    local path=$1
    local parent_id=$2

    # יצירת resource
    aws apigateway create-resource \
        --rest-api-id $API_ID \
        --parent-id $parent_id \
        --path-part "$path" \
        --region $AWS_REGION
}

# /devices resource
DEVICES_ID=$(aws apigateway create-resource \
    --rest-api-id $API_ID \
    --parent-id $ROOT_ID \
    --path-part "devices" \
    --query 'id' --output text)

# /devices/{device_id}
DEVICE_ID_RESOURCE=$(aws apigateway create-resource \
    --rest-api-id $API_ID \
    --parent-id $DEVICES_ID \
    --path-part "{device_id}" \
    --query 'id' --output text)

# /dashboard resource
```

```bash
DASHBOARD_ID=$(aws apigateway create-resource \
    --rest-api-id $API_ID \
    --parent-id $ROOT_ID \
    --path-part "dashboard" \
    --query 'id' --output text)

# הוסף methods...
# (ראה את הסקריפט המלא בקובץ setup_api.sh)
```

5.4 Deploy API

```bash
aws apigateway create-deployment \
    --rest-api-id $API_ID \
    --stage-name prod \
    --region $AWS_REGION

export API_URL="https://$API_ID.execute-
api.$AWS_REGION.amazonaws.com/prod"
echo "API URL: $API_URL"
```

---

6. הגדרת IoT Core

6.1 יצירת Thing Type

```bash
aws iot create-thing-type \
    --thing-type-name EcoShowerDevice \
    --thing-type-properties "thingTypeDescription=EcoShower smart shower
controller" \
    --region $AWS_REGION
```

6.2 יצירת Policy למכשירים

```bash
cat > iot-device-policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iot:Connect",
            "Resource":
"arn:aws:iot:*:*:client/${iot:Connection.Thing.ThingName}"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Publish",
            "Resource":
"arn:aws:iot:*:*:topic/ecoshower/${iot:Connection.Thing.ThingName}/*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource":
"arn:aws:iot:*:*:topicfilter/ecoshower/${iot:Connection.Thing.ThingName}/*
"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Receive",
            "Resource":
"arn:aws:iot:*:*:topic/ecoshower/${iot:Connection.Thing.ThingName}/*"
        }
    ]
}
EOF

aws iot create-policy \
    --policy-name EcoShower-DevicePolicy \
    --policy-document file://iot-device-policy.json \
    --region $AWS_REGION
```

6.3 יצירת IoT Rule

```
# קבל את ARN של Lambda
LAMBDA_ARN=$(aws lambda get-function --function-name EcoShower-
ProcessTelemetry \
    --query 'Configuration.FunctionArn' --output text)

# הוסף permission ל-Lambda
aws lambda add-permission \
    --function-name EcoShower-ProcessTelemetry \
    --statement-id iot-rule \
    --action lambda:InvokeFunction \
    --principal iot.amazonaws.com \
    --region $AWS_REGION

# צור Rule
aws iot create-topic-rule \
    --rule-name EcoShower_ProcessTelemetry \
    --topic-rule-payload "{
        \"sql\": \"SELECT * FROM 'ecoshower/+/telemetry'\",
        \"actions\": [{
            \"lambda\": {
                \"functionArn\": \"$LAMBDA_ARN\"
            }
        }],
        \"ruleDisabled\": false,
        \"awsIotSqlVersion\": \"2016-03-23\"
    }" \
    --region $AWS_REGION
```

---

7. הגדרת SNS

7.1 יצירת Topic

```
aws sns create-topic \
    --name EcoShower-Notifications \
    --region $AWS_REGION

export SNS_TOPIC_ARN=$(aws sns list-topics \
    --query "Topics[?contains(TopicArn, 'EcoShower-
Notifications')].TopicArn" \
    --output text)

echo "SNS Topic ARN: $SNS_TOPIC_ARN"
```

7.2 הוספת Email Subscription (לבדיקה)

```
aws sns subscribe \
    --topic-arn $SNS_TOPIC_ARN \
    --protocol email \
    --notification-endpoint your-email@example.com \
    --region $AWS_REGION
```

---

8. העלאת Frontend ל-S3

8.1 יצירת S3 Bucket

```
export BUCKET_NAME="ecoshower-frontend-$ACCOUNT_ID"

aws s3 mb s3://$BUCKET_NAME --region $AWS_REGION

# הגדרת Static Website Hosting
aws s3 website s3://$BUCKET_NAME \
    --index-document index.html \
    --error-document error.html
```

8.2 Build ו-Upload Frontend

```
cd src/frontend

# עדכון config ה עם-API URL
cat > src/config.js << EOF
export const config = {
    API_URL: '$API_URL',
    COGNITO_USER_POOL_ID: '$USER_POOL_ID',
    COGNITO_CLIENT_ID: '$CLIENT_ID',
    AWS_REGION: '$AWS_REGION'
};
EOF

# Build
npm install
npm run build

# Upload
aws s3 sync dist/ s3://$BUCKET_NAME --delete
```

---

9. הגדרת CloudFront

9.1 יצירת Distribution

```
cat > cloudfront-config.json << EOF
{
    "CallerReference": "ecoshower-$(date +%s)",
    "Origins": {
        "Quantity": 1,
        "Items": [{
            "Id": "S3-$BUCKET_NAME",
            "DomainName": "$BUCKET_NAME.s3.$AWS_REGION.amazonaws.com",
            "S3OriginConfig": {
                "OriginAccessIdentity": ""
            }
        }]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "S3-$BUCKET_NAME",
        "ViewerProtocolPolicy": "redirect-to-https",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["GET", "HEAD"]
        },
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {"Forward": "none"}
        },
        "MinTTL": 0,
        "DefaultTTL": 86400
    },
    "DefaultRootObject": "index.html",
    "Enabled": true,
    "Comment": "EcoShower Frontend"
}
EOF

aws cloudfront create-distribution \
    --distribution-config file://cloudfront-config.json
```

---

10. בדיקת המערכת

10.1 בדיקת DynamoDB Tables

```
for table in Users Devices Sessions Telemetry; do
    echo "Checking EcoShower-$table..."
    aws dynamodb describe-table --table-name EcoShower-$table \
        --query 'Table.TableStatus' --output text
done
```

10.2 בדיקת Lambda Functions

```
# Test ProcessTelemetry
aws lambda invoke \
    --function-name EcoShower-ProcessTelemetry \
    --payload
'{"device_id":"test123","temperature":35,"status":"heating"}' \
    response.json

cat response.json
```

10.3 בדיקת API

```
# Test health endpoint
curl -X GET "$API_URL/health"
```

10.4 כתובות גישה סופיות

```
echo "========================================"
echo "EcoShower Installation Complete!"
echo "========================================"
echo "Frontend URL: https://$(aws cloudfront list-distributions \
    --query 'DistributionList.Items[0].DomainName' --output text)"
```

```
  echo "API URL: $API_URL"
  echo "Cognito User Pool: $USER_POOL_ID"
  echo "Cognito Client ID: $CLIENT_ID"
  echo "=========================================="
  echo ""
  echo "Admin Login:"
  echo "Email: admin@ecoshower.com"
  echo "Password: TempPass123! (change on first login)"
  echo "=========================================="
```

סיכום

לאחר השלמת כל השלבים, המערכת תכלול:

- 4 טבלאות DynamoDB
- 2 Lambda Functions
- REST API עם Cognito Auth
- IoT Core להתחברות מכשירים
- SNS להתראות
- S3 + CloudFront לאחסון והפצת Frontend

זמן התקנה משוער: 30-45 דקות