ThreatWerk - ECS Fargate Deployment Guide

Deploying ThreatWerk on ECS Fargate

July 2026

1 Overview

This guide covers deploying ThreatWerk using Amazon ECS Fargate via a single CloudFormation template. This is the simplest deployment path - no Kubernetes, no Helm, no kubectl required.

Time to deploy: Under 1 hour end-to-end.

What the template provisions:

- VPC with public, private, and database subnets (2 AZs)
- Application Load Balancer with TLS termination
- ECS Fargate cluster with backend and frontend services
- ECR repositories for container images
- RDS PostgreSQL 16 (Multi-AZ configurable)
- Secrets Manager secret
- IAM roles (task execution + application runtime)
- CloudWatch log groups

Prerequisites:

- AWS account with permissions for CloudFormation, VPC, EC2, ECS, ECR, RDS, ACM, IAM, Secrets Manager, ELB
- A domain name and ability to create DNS records
- Docker installed locally (to build and push images)
- `aws` CLI v2 configured (`aws sts get-caller-identity` shows the correct account)

2 Step 1: Deploy the CloudFormation Stack

```
aws cloudformation create-stack \
  --stack-name threatwerk \
  --template-body file://cloudformation/threatwerk-ecs.yaml \
  --capabilities CAPABILITY_NAMED_IAM \
  --parameters \
    ParameterKey=Environment,ParameterValue=production \
    ParameterKey=DomainName,ParameterValue=threatwerk.yourcompany.com \
    ParameterKey=AllowedInboundCidr,ParameterValue=203.0.113.0/24 \
    ParameterKey=ALBScheme,ParameterValue=internet-facing \
    ParameterKey=DBPassword,ParameterValue='YOUR_MASTER_PASSWORD' \
    ParameterKey=RDSMultiAZ,ParameterValue=true
```

Important: The ACM certificate validation is a blocking dependency. Open the ACM console immediately after starting the stack, find the pending certificate, and add the DNS CNAME record. The stack will not proceed until validation completes.

Wait for completion (15-25 minutes):

```
aws cloudformation wait stack-create-complete --stack-name threatwerk
```

Retrieve outputs:

```
aws cloudformation describe-stacks --stack-name threatwerk \
  --query "Stacks[0].Outputs" --output table
```

2.1 ALB Inbound Access

The `AllowedInboundCidr` parameter controls who can reach the ALB:

- **Restricted CIDR (recommended):** Set to your corporate network or VPN egress CIDR
- **Internal ALB + VPN:** Set `ALBScheme=internal` - ALB not reachable from internet

Do not leave `AllowedInboundCidr=0.0.0.0/0` with `ALBScheme=internet-facing` in production.

3 Step 2: Push Container Images to ECR

```
# Get ECR URIs from stack outputs
BACKEND_ECR=$(aws cloudformation describe-stacks --stack-name threatwerk \
  --query "Stacks[0].Outputs[?OutputKey=='BackendECRUri'].OutputValue" --output text)
FRONTEND_ECR=$(aws cloudformation describe-stacks --stack-name threatwerk \
  --query "Stacks[0].Outputs[?OutputKey=='FrontendECRUri'].OutputValue" --output text)

# Authenticate Docker to ECR
aws ecr get-login-password --region REGION | \
  docker login --username AWS --password-stdin ACCOUNT_ID.dkr.ecr.REGION.amazonaws.com

# Build and push
docker build --target prod -t $BACKEND_ECR:latest -f Dockerfile.backend .
docker push $BACKEND_ECR:latest

docker build --target prod -t $FRONTEND_ECR:latest -f Dockerfile.frontend ./frontend
docker push $FRONTEND_ECR:latest
```

4 Step 3: Update Secrets Manager

The CloudFormation stack creates the `threatwerk` database automatically and pre-configures the `DATABASE_URL` in Secrets Manager using the RDS master user. You only need to replace the placeholder `JWT_SECRET` and `ENCRYPTION_KEY`:

```
SECRET_ARN=$(aws cloudformation describe-stacks --stack-name threatwerk \
  --query "Stacks[0].Outputs[?OutputKey=='SecretsManagerARN'].OutputValue" --output text)

RDS_ENDPOINT=$(aws cloudformation describe-stacks --stack-name threatwerk \
  --query "Stacks[0].Outputs[?OutputKey=='RDSEndpoint'].OutputValue" --output text)

aws secretsmanager put-secret-value \
  --secret-id "$SECRET_ARN" \
  --secret-string '{
    "DATABASE_URL": "postgres://masteruser:MASTER_PASSWORD@'$RDS_ENDPOINT':5432/threatwerk?sslmode=require",
    "JWT_SECRET": "'$(openssl rand -base64 48)'",
    "ENCRYPTION_KEY": "'$(openssl rand -base64 32)'"
  }'
```

Replace `MASTER_PASSWORD` with the DB password you provided during stack creation.

5 Step 4: Point DNS to the ALB

Get the ALB DNS name:

```
aws cloudformation describe-stacks --stack-name threatwerk \
  --query "Stacks[0].Outputs[?OutputKey=='ALBDNSName'].OutputValue" --output text
```

Create a CNAME record (or Route 53 alias) pointing your domain to this value.

6 Step 5: Force a New Deployment

The initial tasks may have failed (no database credentials yet). Force a fresh deployment:

```
aws ecs update-service --cluster threatwerk-production \
  --service threatwerk-production-backend --force-new-deployment
aws ecs update-service --cluster threatwerk-production \
  --service threatwerk-production-frontend --force-new-deployment
```

7 Step 6: Verify

```
curl -s https://threatwerk.yourcompany.com/health
# Expected: {"status":"ok","version":"..."}
```

Navigate to `https://threatwerk.yourcompany.com` - the first user to register becomes the admin.

8 Configuration Reference

8.1 Environment Variables

Set these in the ECS task definition (the CloudFormation template pre-configures the required ones):

8.1.1 Required

Variable	Description
`SECRETS_MANAGER_ARN`	Secrets Manager secret ARN
`AWS_REGION`	AWS region (e.g. `eu-central-1`)
`APP_URL`	Public URL (used for emails, CORS, WebAuthn)

8.1.2 Optional

Variable	Default	Description
`LICENSE_MODE`	`aws_marketplace`	`aws_marketplace` or `jwt`
`LISTEN_ADDR`	`:8080`	Backend listen address
`MIGRATIONS_DIR`	`/migrations`	SQL migration files path
`NOTIFY_BACKEND`	unset	`ses` or `smtp` to enable email
`NOTIFY_FROM`	unset	Sender email address
`SES_REGION`	unset	AWS region for SES
`DB_MAX_OPEN_CONNS`	`20`	Max open DB connections per task
`DB_MAX_IDLE_CONNS`	`5`	Max idle DB connections
`INTEL_POLL_INTERVAL`	`12h`	Intel feed poll interval
`LOG_LEVEL`	`info`	`debug`, `info`, `warn`, `error`
`RATE_LIMIT_BURST`	`60`	Rate limiter burst size
`RATE_LIMIT_RPS`	`10`	Sustained requests/second

8.2 Secrets Manager Keys

Key	Required	Description
`DATABASE_URL`	Yes	PostgreSQL connection string with `sslmode=require`
`JWT_SECRET`	Yes	HMAC signing key for sessions (min 32 chars)
`ENCRYPTION_KEY`	Yes	AES-GCM key for credential encryption (min 32 chars)

Intel feed API keys (NVD, GHSA, OTX) are configured through the admin UI after installation.

9 Deploying Updates

```
# Build and push new images
docker build --target prod -t $BACKEND_ECR:v26.06 -f Dockerfile.backend .
docker push $BACKEND_ECR:v26.06

# Force new deployment
aws ecs update-service --cluster threatwerk-production \
  --service threatwerk-production-backend --force-new-deployment
aws ecs update-service --cluster threatwerk-production \
  --service threatwerk-production-frontend --force-new-deployment
```

Database migrations run automatically on startup. No manual migration steps needed.

10 Scaling

```
# Scale backend to 4 tasks
aws ecs update-service --cluster threatwerk-production \
  --service threatwerk-production-backend --desired-count 4
```

Or update the `BackendDesiredCount` CloudFormation parameter.

Database connection math: Each task opens up to `DB_MAX_OPEN_CONNS` (default: 20). With N tasks, ensure RDS `max_connections` is at least `N x 20 + 10`.

11 Operations

11.1 Logs

View in CloudWatch: - Backend: `/ecs/threatwerk-production/backend` - Frontend: `/ecs/threatwerk-production/frontend`

11.2 Debugging

Shell into a running task:

```
aws ecs execute-command --cluster threatwerk-production \
  --task TASK_ID --container backend \
  --interactive --command /bin/sh
```

11.3 Backup and Recovery

RDS handles automated daily backups. To restore:

1. Restore RDS from snapshot or point-in-time
2. Update `DATABASE_URL` in Secrets Manager
3. Force new backend deployment

11.4 Secret Rotation

Secret	Impact	Procedure
`JWT_SECRET`	Invalidates all sessions	Update in SM, force-new-deployment
`ENCRYPTION_KEY`	SBOM creds unreadable	Update, redeploy, re-enter SBOM creds
DB password	Connection failure	Update in RDS + SM, redeploy

12 Troubleshooting

12.1 Tasks not starting

Check CloudWatch logs (`/ecs/threatwerk-production/backend`):

Log message	Cause	Fix
`failed to resolve secrets`	Task role missing SM permissions	Check IAM role attached to task definition
`database not reachable`	Security group blocks 5432	Allow compute SG to reach RDS SG
`license validation failed`	License Manager perms missing	Add LM actions to task role policy
`JWT_SECRET must be at least 32 characters`	Value too short	Update in Secrets Manager

12.2 502 Bad Gateway

- Tasks not healthy - check target group health in EC2 console
- Health check path must be `/health`
- Target group port must be 8080

12.3 WebSocket disconnects

- ALB idle timeout too low - needs 180+ seconds
- Check the ALB idle timeout attribute in the CloudFormation template

12.4 No intel feed data

- No outbound internet - verify NAT Gateway exists and route table is correct
- NVD rate limiting - add `NVD_API_KEY` to Secrets Manager
- First poll happens after `INTEL_POLL_INTERVAL` (default 12h) - set to `5m` temporarily to test

13 Estimated Monthly Costs

13.1 Small Team (5-20 users)

Service	Configuration	Cost
ECS Fargate (backend)	2 tasks, 0.5 vCPU / 1 GB each	$45
ECS Fargate (frontend)	1 task, 0.25 vCPU / 0.5 GB	$12
RDS PostgreSQL	`db.t3.medium`, Multi-AZ, 50 GB	$134
Application Load Balancer	1 ALB	$27
NAT Gateway	1 gateway + ~30 GB data	$47
Secrets Manager, ECR, CloudWatch	Minimal	$5
Total		**~$270/month**

13.2 Medium Team (20-50 users)

Service	Configuration	Cost
ECS Fargate (backend)	3 tasks, 1 vCPU / 2 GB each	$135
ECS Fargate (frontend)	2 tasks, 0.25 vCPU / 0.5 GB	$24
RDS PostgreSQL	`db.r6g.large`, Multi-AZ, 100 GB	$380
Application Load Balancer	1 ALB	$38
NAT Gateway	1 gateway + ~60 GB data	$60
Secrets Manager, ECR, CloudWatch	Minimal	$11
Total		**~$648/month**

13.3 Test/Staging (cost-optimized)

Service	Configuration	Cost
ECS Fargate (backend)	1 task, 0.5 vCPU / 1 GB	$22
ECS Fargate (frontend)	1 task, 0.25 vCPU / 0.5 GB	$12
RDS PostgreSQL	`db.t3.micro`, Single-AZ, 20 GB	$15
ALB + NAT + misc	Minimal	$61
Total		**~$110/month**

Software license fee (AWS Marketplace) is separate and not included above.