DB Stack

Plug-and-play database stack with MongoDB 7, InfluxDB 2, and MySQL 8 on a single EBS volume. Auto-configures via user-data or interactive setup. Production-ready with Docker, backups, and CLI management.

- Version 1.1.0 - ami-0da7080bd1e12b6b8

Version 1.1.0 - ami-0da7080bd1e12b6b8

Database Stack AMI

MongoDB + InfluxDB + MySQL - A plug-and-play multi-database stack for AWS EC2 with persistent EBS storage.

Overview

This AMI provides a production-ready database stack with three popular databases:

Database	Version	Port	Use Case
MongoDB	7.x	27017	Document/NoSQL database
InfluxDB	2.x	8086	Time-series database
MySQL	8.x	3306	Relational database

All databases share a single EBS volume for persistent storage, making it easy to backup, snapshot, and manage your data.

Requirements

Instance Sizing

Workload	Instance Type	vCPU	RAM	Notes
Development	t3.small	2	2 GB	Light testing
Small Production	t3.medium	2	4 GB	Recommended minimum
Medium Production	t3.large	2	8 GB	Better performance
Large Production	t3.xlarge+	4+	16+ GB	Heavy workloads

Storage

- **Minimum EBS**: 20 GB (gp3)
- **Recommended**: 50-100 GB+ depending on data volume
- **Type**: gp3 for best price/performance

Network (Security Group)

Port	Protocol	Source	Description
22	TCP	Your IP	SSH access
27017	TCP	Your App/VPC	MongoDB
8086	TCP	Your App/VPC	InfluxDB
3306	TCP	Your App/VPC	MySQL

⚠ **Security Note**: Never expose database ports (27017, 8086, 3306) to 0.0.0.0/0 in production!

Getting Started

1. Launch Instance

- Launch an EC2 instance from this AMI
- Recommended: t3.medium or larger
- Configure security group with required ports

2. Attach EBS Volume

- Create a gp3 EBS volume (20 GB minimum)
- **Important**: Same Availability Zone as your instance
- Attach to your instance as `/dev/sdf`

3. Configure Databases

SSH into your instance and run:

```
sudo /opt/dbstack/configure-dbstack.sh
```

The wizard will prompt you for:

MongoDB:

- Root username and password
- Application database name
- Application user credentials

InfluxDB:

- Organization name
- Bucket name
- Admin credentials

MySQL:

- Root password
- Application database name
- Application user credentials

4. Verify Installation

```
dbstack-cli status
```

Management Commands

```
# Service management
dbstack-cli status          # Show status of all databases
dbstack-cli start           # Start all services
dbstack-cli stop            # Stop all services
dbstack-cli restart         # Restart all services

# Logs
dbstack-cli logs            # All container logs
dbstack-cli logs mongodb    # MongoDB logs only
dbstack-cli logs influxdb   # InfluxDB logs only
dbstack-cli logs mysql      # MySQL logs only

# Configuration
dbstack-cli info            # Show configuration info
sudo dbstack-cli credentials # Show passwords and connection strings

# Maintenance
```

```
dbstack-cli update          # Pull latest images and restart
sudo dbstack-cli backup     # Backup all databases
sudo dbstack-cli restore    # Restore from backup

# Database shells
dbstack-cli mongo-shell     # MongoDB shell
dbstack-cli mysql-shell     # MySQL shell
dbstack-cli influx-shell    # InfluxDB CLI
```

Connection Examples

MongoDB

```
# Python (pymongo)
from pymongo import MongoClient

# Using root credentials
client = MongoClient("mongodb://admin:password@your-ip:27017")

# Using app credentials
client = MongoClient("mongodb://appuser:password@your-ip:27017/mydb")
db = client.mydb
```

```
// Node.js
const { MongoClient } = require('mongodb');
const client = new MongoClient('mongodb://appuser:password@your-ip:27017/mydb');
```

InfluxDB

```
# Python (influxdb-client)
from influxdb_client import InfluxDBClient

client = InfluxDBClient(
    url="http://your-ip:8086",
    token="your-api-token",
    org="myorg"
)
```

```
# CLI
influx write -b mybucket -o myorg -t your-token \
  'temperature,location=office value=72.5'
```

MySQL

```
# Python (mysql-connector)
import mysql.connector

conn = mysql.connector.connect(
    host="your-ip",
    port=3306,
    user="appuser",
    password="password",
    database="mydb"
)
```

```
# CLI
mysql -h your-ip -P 3306 -u appuser -p mydb
```

Backup & Recovery

Manual Backup

```
# Create backup of all databases
sudo dbstack-cli backup

# Backups stored in: /mnt/dbstack-data/backups/
```

EBS Snapshots (Recommended)

For production, use EBS snapshots for point-in-time recovery:

```
# Stop services before snapshot for consistency
dbstack-cli stop
```

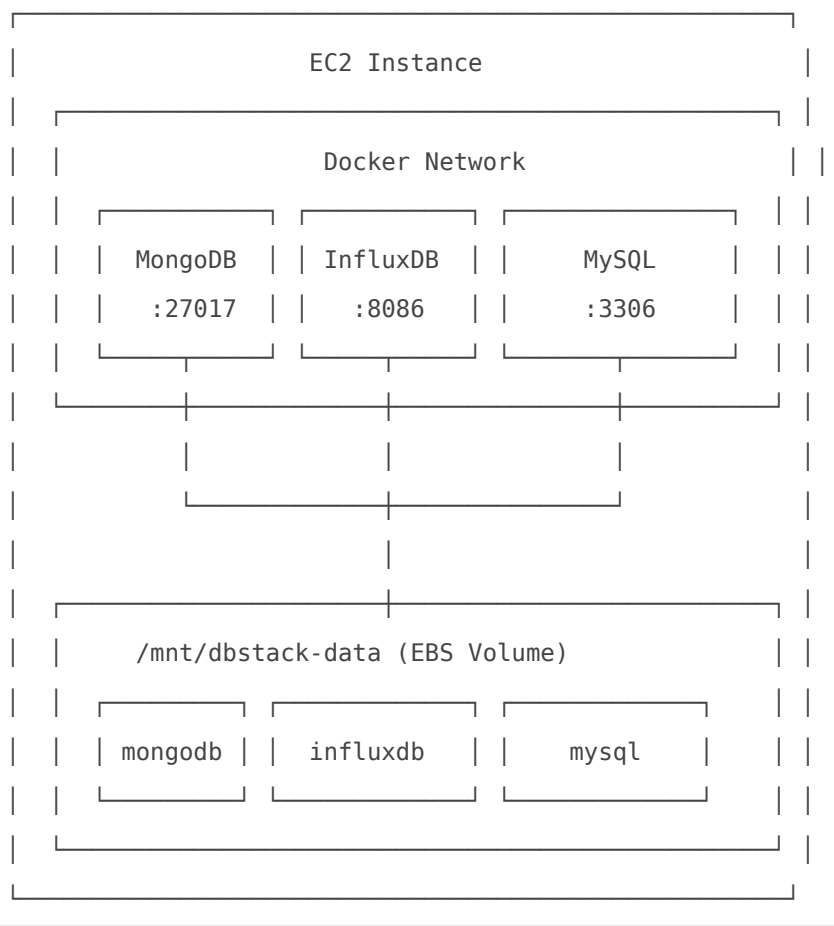
```
# Create snapshot via AWS Console or CLI
aws ec2 create-snapshot --volume-id vol-xxxxx --description "dbstack-backup-$(date +%Y%m%d)"

# Start services
dbstack-cli start
```

Restore from Backup

```
sudo dbstack-cli restore
# Follow prompts to select backup timestamp
```

Architecture



Troubleshooting

Service Won't Start

```
# Check Docker status
systemctl status docker

# Check container logs
docker logs dbstack-mongodb
docker logs dbstack-influxdb
docker logs dbstack-mysql

# Check EBS mount
df -h /mnt/dbstack-data
```

Connection Refused

1. Check security group allows the port
2. Verify UFW rules: `sudo ufw status`
3. Check service is running: `dbstack-cli status`

Out of Disk Space

```
# Check disk usage
dbstack-cli status

# Extend EBS volume in AWS Console, then:
sudo growpart /dev/xvdf 1
sudo resize2fs /dev/xvdf
```

Reset Everything

```
# Complete reset (DESTROYS ALL DATA)
sudo /opt/dbstack/configure-dbstack.sh
# Select "yes" when asked to reconfigure
```

Data Locations

Path	Description
`/mnt/dbstack-data/mongodb`	MongoDB data files

Path	Description
`/mnt/dbstack-data/influxdb`	InfluxDB data and config
`/mnt/dbstack-data/mysql`	MySQL data files
`/mnt/dbstack-data/backups`	Backup files
`/opt/dbstack/.credentials`	Saved credentials
`/opt/dbstack/config-info.txt`	Configuration summary

Security Recommendations

1. **Use VPC**: Keep databases in private subnet
2. **Security Groups**: Restrict ports to application servers only
3. **Strong Passwords**: Use 16+ character passwords
4. **Enable Encryption**: Use encrypted EBS volumes
5. **Regular Backups**: Automate EBS snapshots
6. **Update Regularly**: `dbstack-cli update` for security patches

Support

- **Credentials**: `sudo cat /opt/dbstack/.credentials`
- **Configuration**: `cat /opt/dbstack/config-info.txt`
- **Logs**: `dbstack-cli logs`

Version History

- **1.0.0**: Initial release with MongoDB 7, InfluxDB 2, MySQL 8

Built for AWS Marketplace | Ubuntu 24.04 | Docker-based