6: Borgmatic - Enterprise Backup Solution

Borgmatic is a configuration-driven wrapper for BorgBackup that provides enterprise-grade backup capabilities with deduplication, encryption, and automated scheduling. It implements a dual-backup strategy separating file and database backups with different retention policies. For comprehensive configuration options, advanced features, and API documentation, please refer to the official Borgmatic documentation.

Interdependencies

Required dependency: Apprise must be installed and running for notification delivery (success/failure/security alerts). Install via □□ *APPLICATIONS → Apprise → Install* before enabling Borgmatic.

Prerequisites

- □ **Docker running** (Chapter 3)
- □ **Apprise installed** (Chapter 5) for notifications
- □ **Storage capacity** (backups require free disk space)
- □ **Optional:** Traefik (not required for Borgmatic)

Architecture Overview

Borgmatic in Infinity Tools implements a sophisticated backup architecture:

- □ **Dual Backup Strategy** - Separate file and database backup schedules
- □ **Deduplication** - Efficient storage using Borg's chunking algorithm
- □ **Encryption** - AES-256 encryption with repokey mode
- □ **Ransomware Protection** - Canary file monitoring system
- □ **Multi-Database Support** - MariaDB, PostgreSQL, SQLite, MongoDB
- □ **Notification Integration** - Apprise integration for alerts

Installation Methods

Via Infinity Tools Menu

Navigate to the Infinity Tools menu and select:

```
□□BACKUP MANAGEMENT → Install Borgmatic
```

Command Line Installation

```
# Direct script execution
sudo bash /opt/InfinityTools/Solutions/setup-borgmatic.sh --install

# With environment variables
export BORGMATIC_SCHEDULE="daily"
export BORGMATIC_RETENTION="7,4,6"
export BORGMATIC_COMPRESSION="zstd"
export BORGMATIC_ENCRYPTION="repokey"
sudo -E bash /opt/InfinityTools/Solutions/setup-borgmatic.sh --install
```

Configuration Parameters

Schedule Configuration

Borgmatic supports multiple schedule options:

```
# Environment Variables
export BORGMATIC_SCHEDULE="daily"        # daily, twice-daily, weekly
export BORGMATIC_RETENTION="7,4,6"       # daily,weekly,monthly
export BORGMATIC_COMPRESSION="zstd"      # lz4, zlib, lzma, zstd
export BORGMATIC_ENCRYPTION="repokey"    # none, keyfile, repokey
```

Dual Backup Strategy

The system implements two separate backup configurations:

- **File Backups:** Daily schedule, long-term retention
- **Database Backups:** High-frequency schedule, short-term retention

Generated Configuration

File Backup Configuration

Location: `/opt/speedbits/borgmatic/borgmatic-files.yml`

```
# Borgmatic Files Configuration
source_directories:
    - /opt/speedbits

repositories:
    - path: /backups/borgmatic-repo
      label: speedbits-repo

exclude_patterns:
    - '*.tmp'
    - '*.log'
    - '*/logs/*'
    - '*/cache/*'
    - '*/tmp/*'
    - '*/.git/*'
    - '*/node_modules/*'
    - '*/venv/*'
    - '*/__pycache__/*'
    - '*/database-dumps/*'
    - '*/netdata/lib/*'
    - '*/netdata/cache/*'

compression: zstd
archive_name_format: 'speedbits-files-{hostname}-{now:%Y-%m-%d-%H%M%S}'

keep_daily: 7
keep_weekly: 4
keep_monthly: 6

checks:
    - name: repository
      frequency: 2 weeks
```

```
    - name: archives
      frequency: 2 weeks
```

Database Backup Configuration

Location: `/opt/speedbits/borgmatic/borgmatic-databases.yml`

```
# Borgmatic Database Configuration
source_directories:
    - /backups/database-dumps

repositories:
    - path: /backups/borgmatic-repo
      label: speedbits-repo

compression: zstd,9
archive_name_format: 'speedbits-databases-{hostname}-{now:%Y-%m-%d-%H%M%S}'

keep_hourly: 48
keep_daily: 7

# Database preparation hooks
hooks:
    before_backup:
        - /usr/local/bin/backup-databases.sh
    after_backup:
        - /usr/local/bin/cleanup-database-dumps.sh
```

Security Features

Ransomware Protection

Borgmatic implements canary file monitoring:

```
# Canary file system
mkdir -p /opt/speedbits/dont-touch-this-folder
echo "This is a test" > /opt/speedbits/dont-touch-this-folder/dont-change-this-file-critical-
```

```
data.txt
chmod 644 /opt/speedbits/dont-touch-this-folder/dont-change-this-file-critical-data.txt

# Security monitoring in borgmatic-files.yml
commands:
    - before: action
      when: [create]
      run:
          - if [ ! -f /opt/speedbits/dont-touch-this-folder/dont-change-this-file-critical-
data.txt ]; then echo "CANARY FILE MISSING!" && curl -X POST http://apprise:8000/notify -d
"body=SECURITY ALERT Canary file is MISSING." -d "title=CANARY FILE MISSING" -d "tag=security"
2>/dev/null || true && exit 1; fi
```

Encryption Configuration

Borgmatic uses repokey encryption mode:

```
# Repository initialization
borg init --encryption=repokey --make-parent-dirs /backups/borgmatic-repo

# Passphrase management
BORG_PASSCOMMAND="cat /etc/borgmatic/repo-passphrase.txt"
export BORG_PASSCOMMAND
```

Database Integration

Multi-Database Support

Borgmatic automatically discovers and backs up multiple database types:

```
# Database discovery script
#!/bin/bash
# /usr/local/bin/backup-databases.sh

# MariaDB/MySQL
if docker ps --format '{{.Names}}' | grep -q "mariadb\|mysql"; then
    docker exec mariadb mysqldump --all-databases > /backups/database-dumps/mariadb-$(date
```

```
+%Y%m%d-%H%M%S).sql
fi

# PostgreSQL
if docker ps --format '{{.Names}}' | grep -q "postgres"; then
    docker exec postgres pg_dumpall > /backups/database-dumps/postgres-$(date +%Y%m%d-
%H%M%S).sql
fi

# SQLite
find /opt/speedbits -name "*.db" -o -name "*.sqlite" -o -name "*.sqlite3" | while read db; do
    cp "$db" "/backups/database-dumps/sqlite-$(basename "$db")-$(date +%Y%m%d-%H%M%S)"
done
```

Docker Compose Configuration

Container Setup

Location: `/opt/speedbits/borgmatic/docker-compose.yml`

```
version: '3.8'

services:
  borgmatic:
    image: borgmatic/borgmatic:latest
    container_name: borgmatic
    restart: unless-stopped
    environment:
      BORG_PASSCOMMAND: "cat /etc/borgmatic/repo-passphrase.txt"
    volumes:
      - /opt/speedbits:/opt/speedbits:ro
      - /opt/speedbits-backup:/backups
      - /opt/speedbits/borgmatic:/etc/borgmatic:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
    networks:
      - borgmatic-network
      - borgmatic-db
```

```
    command: |
      /bin/sh -c "
        echo 'Initializing Borgmatic...'

        # Install rclone for cloud storage support
        if ! command -v rclone >/dev/null 2>&1; then
          echo 'Installing rclone for cloud storage support...'
          apk add --no-cache rclone || echo 'rclone installation failed, continuing...'
        fi

        # Initialize repository if needed
        if ! borg info /backups/borgmatic-repo >/dev/null 2>&1; then
          echo 'Creating new Borg repository...'
          borg init --encryption=repokey --make-parent-dirs /backups/borgmatic-repo
        fi

        # Set up cron jobs
        {
          echo 'BORG_PASSCOMMAND=\"cat /etc/borgmatic/repo-passphrase.txt\"'
          echo '0 2 * * * borgmatic --config /etc/borgmatic/borgmatic-files.yml create --
verbosity 1'
          echo '0 */6 * * * borgmatic --config /etc/borgmatic/borgmatic-databases.yml create
--verbosity 1'
        } | crontab -

        # Start cron daemon
        crond -f
      "

networks:
  borgmatic-network:
    driver: bridge
  borgmatic-db:
    external: true
```

Monitoring and Notifications

Apprise Integration

Borgmatic integrates with Apprise for notifications:

```
# Notification configuration in borgmatic-files.yml
commands:
    - before: action
      when: [create]
      run:
          - curl -X POST http://apprise:8000/notify -d "body=Starting scheduled file backup" -
d "title=File Backup Started" -d "tag=backup" 2>/dev/null || true

    - after: action
      when: [create]
      states: [finish]
      run:
          - curl -X POST http://apprise:8000/notify -d "body=File backup completed
successfully" -d "title=File Backup Complete" -d "tag=backup" 2>/dev/null || true

    - after: action
      when: [create]
      states: [fail]
      run:
          - curl -X POST http://apprise:8000/notify -d "body=File backup FAILED" -d
"title=File Backup FAILED" -d "tag=backup,error" 2>/dev/null || true
```

Health Monitoring

```
# Check backup status
docker exec borgmatic borg list /backups/borgmatic-repo

# Check repository integrity
docker exec borgmatic borg check /backups/borgmatic-repo

# View backup logs
docker logs borgmatic

# Check cron jobs
docker exec borgmatic crontab -l
```

Advanced Configuration

Cloud Storage Integration

Borgmatic supports cloud storage via rclone:

```
# rclone configuration
rclone config create remote s3 \
  provider=AWS \
  access_key_id=your_access_key \
  secret_access_key=your_secret_key \
  region=us-east-1

# Cloud backup command
rclone sync /opt/speedbits-backup/borgmatic-repo remote:backups/borgmatic-repo
```

Custom Hooks

Implement custom backup hooks:

```
# Custom pre-backup hook
hooks:
  before_backup:
    - /usr/local/bin/pre-backup.sh
    - /usr/local/bin/backup-databases.sh
  after_backup:
    - /usr/local/bin/post-backup.sh
    - /usr/local/bin/cleanup-temp-files.sh
  on_error:
    - /usr/local/bin/backup-error-handler.sh
```

Performance Optimization

Compression Settings

Choose compression based on your needs:

- **lz4:** Fastest, least compression
- **zstd:** Balanced speed and compression
- **zlib:** Good compression, moderate speed
- **lzma:** Best compression, slowest

Resource Management

```
# Resource limits in docker-compose.yml
services:
  borgmatic:
    deploy:
      resources:
        limits:
          memory: 1G
          cpus: '1.0'
        reservations:
          memory: 512M
          cpus: '0.5'
```

Disaster Recovery

Backup Restoration

```
# List available archives
docker exec borgmatic borg list /backups/borgmatic-repo

# Extract specific archive
docker exec borgmatic borg extract /backups/borgmatic-repo::speedbits-files-server-2024-01-15-020000

# Extract to specific location
docker exec borgmatic borg extract /backups/borgmatic-repo::speedbits-files-server-2024-01-15-020000 /restore/path
```

Repository Recovery

```
# Check repository integrity
docker exec borgmatic borg check /backups/borgmatic-repo

# Repair repository if needed
docker exec borgmatic borg check --repair /backups/borgmatic-repo

# Recover from cloud storage
rclone sync remote:backups/borgmatic-repo /opt/speedbits-backup/borgmatic-repo
```

Troubleshooting

Common Issues

Backup Failures:

```
# Check container logs
docker logs borgmatic

# Check cron jobs
docker exec borgmatic crontab -l

# Test manual backup
docker exec borgmatic borgmatic --config /etc/borgmatic/borgmatic-files.yml create --verbosity
2
```

Repository Issues:

```
# Check repository status
docker exec borgmatic borg info /backups/borgmatic-repo

# Verify passphrase
docker exec borgmatic cat /etc/borgmatic/repo-passphrase.txt

# Check disk space
df -h /opt/speedbits-backup
```

Debug Commands

```
# Container status
docker ps | grep borgmatic

# Container exec
docker exec -it borgmatic /bin/sh

# Check mounted volumes
docker inspect borgmatic | grep -A 10 Mounts

# View configuration
docker exec borgmatic cat /etc/borgmatic/borgmatic-files.yml
```

Best Practices

Security

- Use strong, unique passphrases
- Store passphrases securely (password manager)
- Enable canary file monitoring
- Regular backup testing and restoration

Operational

- Monitor backup success/failure notifications
- Regular repository integrity checks
- Test disaster recovery procedures
- Document restoration procedures

Next Steps

With Borgmatic installed and configured, you have enterprise-grade backup protection ready for all your services. This infrastructure will automatically protect any applications you install going forward.

Verification Checklist

- □ Borgmatic container running and healthy
- □ File and database backup schedules active

- Repository initialized and encrypted
- Canary file monitoring enabled
- Notification system configured
- Backup restoration tested

Next: Installing Vaultwarden - Password Management Solution (Chapter 6)

Revision #2
Created 31 October 2025 13:10:21 by bjoern
Updated 17 November 2025 16:33:45 by bjoern