DevOps VM Setup with GitHub Actions

This repository contains a secure VM setup script that can be deployed automatically using GitHub Actions.

🚀 Features

- **Conditional Deployment**: Only runs when the setup script actually changes
- **Manual Trigger**: Force deployment when needed
- **Multi-Environment Support**: Deploy to production or staging environments
- **Secure SSH Deployment**: Uses GitHub Secrets for secure access
- **Automatic Backups**: Creates backups before deployment

📋 Prerequisites

1. VM Setup

- Ubuntu-based VM (tested on Ubuntu 20.04+)
- SSH access configured
- User with sudo privileges

2. GitHub Secrets Configuration

Add these secrets to your GitHub repository (`Settings` → `Secrets and variables` → `Actions`):

Required Secrets:

- `VM_SSH_PRIVATE_KEY`: Your private SSH key for VM access
- `PRODUCTION_VM_IP`: IP address of your production VM
- `STAGING_VM_IP`: IP address of your staging VM (optional)

How to generate SSH key:

```bash
# Generate SSH key pair
ssh-keygen -t rsa -b 4096 -C "github-actions@your-domain.com"

# Copy public key to VM
ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@YOUR_VM_IP

# Copy private key content to GitHub secret
cat ~/.ssh/id_rsa
```

🔧 Usage

Automatic Deployment

The workflow automatically triggers when:

- You push changes to `scripts/setup-vm-and-docker.sh`
- You push changes to the workflow file itself
- Changes are made to the `main` branch

Manual Deployment

1. Go to your repository on GitHub
2. Navigate to `Actions` tab
3. Select "Deploy VM Setup Script"
4. Click "Run workflow"
5. Choose your options:
   - **Force deployment**: Skip change detection
   - **Target environment**: Choose production or staging

🛡️ Security Features

The setup script includes:

- **Firewall Configuration**: UFW with secure defaults
- **SSH Hardening**: Disabled root login and password auth
- **Fail2Ban**: Protection against brute force attacks
- **Automatic Updates**: Security updates enabled
- **Docker Installation**: Latest Docker with Swarm mode

📁 File Structure

```
devops/
├── .github/
│   └── workflows/
│       └── deploy-vm-setup.yml    # GitHub Actions workflow
├── scripts/
│   └── setup-vm-and-docker.sh     # VM setup script
└── README.md                       # This file
```

🔍 Monitoring

Check Deployment Status

- View workflow runs in GitHub Actions tab
- Check VM logs: `sudo journalctl -u docker`
- Verify Docker: `docker info`

Troubleshooting

1. **SSH Connection Issues**: Verify your SSH key is correct
2. **Permission Denied**: Ensure the VM user has sudo privileges
3. **Script Fails**: Check the workflow logs for detailed error messages

⚠️ Important Notes

- **Backup Your VM**: Always backup your VM before running the setup
- **SSH Key Security**: Keep your private key secure and never commit it to the repository
- **Firewall Rules**: The script opens ports 22 (SSH), 80 (HTTP), and 443 (HTTPS)
- **Docker Swarm**: The script initializes Docker Swarm mode

🚨 Emergency Access

If you lose SSH access after deployment:

1. Use your cloud provider's console access
2. Temporarily disable the firewall: `sudo ufw disable`
3. Check SSH configuration: `sudo nano /etc/ssh/sshd_config`
4. Restart SSH: `sudo systemctl restart ssh`