RHEL STIG RAG - Ansible Deployment Guide

This directory contains Ansible playbooks and configurations for automated deployment of the RHEL STIG RAG system.

Quick Start

```bash
# Clone the repository
git clone <repository-url>
cd rhel-stig-rag/ansible

# Install Ansible (if not already installed)
pip3 install ansible

# Configure your inventory
cp inventory/hosts.example inventory/hosts
# Edit inventory/hosts with your server details

# Run the deployment
ansible-playbook -i inventory/hosts site.yml

# Deploy to specific environment
ansible-playbook -i inventory/hosts site.yml --limit production
```

Directory Structure

```
ansible/
├── site.yml                 # Main playbook
├── ansible.cfg              # Ansible configuration
├── inventory/
│   └── hosts                # Inventory file
├── vars/
│   ├── main.yml             # Default variables
│   └── secrets.yml          # Encrypted secrets (vault)
├── tasks/
│   ├── dependencies.yml     # System dependencies
│   ├── user.yml             # User management
│   ├── application.yml      # Application deployment
│   ├── service.yml          # Service configuration
│   ├── stig_data.yml        # STIG data management
│   ├── monitoring.yml       # Monitoring setup
│   └── health_check.yml     # Health verification
├── templates/
│   ├── stig_rag_config.env.j2
│   ├── stig-rag.service.j2
│   └── stig_rag_start.sh.j2
└── files/
    └── sample_data/
```

Configuration

1. Inventory Setup

Edit `inventory/hosts` to match your environment:

```
ini

[stig_rag_servers]
stig-server-01 ansible_host=192.168.1.100

[production:vars]
stig_rag_download_official_stigs=true
stig_rag_backup_enabled=true
```

2. Variables Configuration

Key variables in `vars/main.yml`:

```yaml
# RHEL Version Priority
stig_rag_default_rhel_version: 9
stig_rag_supported_rhel_versions: [8, 9]

# Service Configuration
stig_rag_port: 8000
stig_rag_workers: 4

# LLM Provider (huggingface, openai, anthropic)
stig_rag_llm_provider: "huggingface"
```

3. Secrets Management

For sensitive data like API keys:

```bash
# Create encrypted secrets file
ansible-vault create vars/secrets.yml

# Edit existing secrets
ansible-vault edit vars/secrets.yml

# Run playbook with vault password
ansible-playbook -i inventory/hosts site.yml --ask-vault-pass
```

Deployment Options

Basic Deployment

```bash
# Deploy with default settings
ansible-playbook -i inventory/hosts site.yml
```

Development Deployment

```bash
# Deploy in development mode
ansible-playbook -i inventory/hosts site.yml -e "stig_rag_dev_mode=true"
```

Production Deployment

bash

```
# Deploy to production with all features
ansible-playbook -i inventory/hosts site.yml --limit production \
  -e "stig_rag_download_official_stigs=true" \
  -e "stig_rag_backup_enabled=true" \
  -e "stig_rag_setup_proxy=true"
```

Use OpenAI/Anthropic Models

bash

```
# Deploy with OpenAI
ansible-playbook -i inventory/hosts site.yml \
  -e "stig_rag_llm_provider=openai" \
  --ask-vault-pass

# Deploy with Anthropic Claude
ansible-playbook -i inventory/hosts site.yml \
  -e "stig_rag_llm_provider=anthropic" \
  --ask-vault-pass
```

Tags Usage

Run specific parts of the deployment:

bash

```
# Only install dependencies
ansible-playbook -i inventory/hosts site.yml --tags dependencies

# Only deploy application code
ansible-playbook -i inventory/hosts site.yml --tags application

# Only manage STIG data
ansible-playbook -i inventory/hosts site.yml --tags stig

# Only run health checks
ansible-playbook -i inventory/hosts site.yml --tags health
```

Environment-Specific Deployments

Development Environment

```bash
# Deploy to dev with sample data
ansible-playbook -i inventory/hosts site.yml --limit development
```

Configuration in inventory:

```ini
[development:vars]
stig_rag_dev_mode=true
stig_rag_debug_enabled=true
stig_rag_load_sample_data=true
stig_rag_download_official_stigs=false
```

Production Environment

```bash
# Deploy to production
ansible-playbook -i inventory/hosts site.yml --limit production
```

Configuration in inventory:

```ini
[production:vars]
stig_rag_dev_mode=false
stig_rag_load_sample_data=false
stig_rag_download_official_stigs=true
stig_rag_backup_enabled=true
stig_rag_ssl_enabled=true
```

Monitoring and Maintenance

Health Checks

bash

```bash
# Run health checks only
ansible-playbook -i inventory/hosts site.yml --tags health

# Check service status across all servers
ansible stig_rag_servers -i inventory/hosts -a "systemctl status stig-rag"
```

Updates

bash

```bash
# Update application code
ansible-playbook -i inventory/hosts site.yml --tags application

# Update STIG data
ansible-playbook -i inventory/hosts site.yml --tags stig

# Full update
ansible-playbook -i inventory/hosts site.yml
```

Ad-hoc Commands

bash

```bash
# Check service status
ansible stig_rag_servers -i inventory/hosts -a "systemctl status stig-rag"

# Restart service
ansible stig_rag_servers -i inventory/hosts -a "systemctl restart stig-rag" --become

# Check logs
ansible stig_rag_servers -i inventory/hosts -a "tail -n 50 /var/log/stig-rag/stig_rag.log"

# Test application
ansible stig_rag_servers -i inventory/hosts -m uri -a "url=http://localhost:8000/health"
```

Scaling and High Availability

Load Balancer Setup

yaml

```
# In vars/main.yml
stig_rag_setup_proxy: true
stig_rag_proxy_server: "nginx"

# In inventory/hosts
[load_balancers]
stig-lb-01 ansible_host=192.168.1.200

[stig_rag_servers]
stig-rag-01 ansible_host=192.168.1.101
stig-rag-02 ansible_host=192.168.1.102
stig-rag-03 ansible_host=192.168.1.103
```

Rolling Updates

bash

```
# Update servers one by one
ansible-playbook -i inventory/hosts site.yml --serial 1

# Update with maximum 2 servers at a time
ansible-playbook -i inventory/hosts site.yml --serial 2
```

Troubleshooting

Common Issues

1. **Service won't start**

   bash

   ```
   # Check service status
   ansible stig_rag_servers -i inventory/hosts -a "systemctl status stig-rag" --become

   # Check logs
   ansible stig_rag_servers -i inventory/hosts -a "journalctl -u stig-rag -n 50" --become
   ```

2. **Permission issues**

   bash

   ```
   # Fix ownership
   ansible stig_rag_servers -i inventory/hosts -a "chown -R stigrag:stigrag /opt/stig-rag" --b
   ```

3. **Connectivity issues**

   ```bash
   # Test connection
   ansible all -i inventory/hosts -m ping

   # Test sudo access
   ansible all -i inventory/hosts -a "whoami" --become
   ```

Debug Mode

```bash
# Run with increased verbosity
ansible-playbook -i inventory/hosts site.yml -vvv

# Check what would change (dry run)
ansible-playbook -i inventory/hosts site.yml --check --diff
```

Performance Tuning

For Large Deployments

```yaml
# In vars/main.yml
stig_rag_workers: "{{ ansible_processor_vcpus * 2 }}"
stig_rag_chunk_size: 1500
stig_rag_memory_limit_mb: 4096
stig_rag_cpu_limit_percent: 80
```

For Resource-Constrained Environments

```yaml
# In vars/main.yml
stig_rag_workers: 2
stig_rag_chunk_size: 500
stig_rag_embedding_model: "all-MiniLM-L6-v2"  # Smaller model
stig_rag_llm_model: "microsoft/DialoGPT-small"
```

Security Considerations

1. **Use Ansible Vault for secrets**:

   bash

   ```
   ansible-vault encrypt vars/secrets.yml
   ```

2. **SSH key authentication**:

   ini

   ```
   # In inventory/hosts
   [all:vars]
   ansible_ssh_private_key_file=~/.ssh/stig_rag_key
   ```

3. **Firewall configuration**:

   yaml

   ```
   # In vars/main.yml
   stig_rag_configure_firewall: true
   stig_rag_allowed_sources: ["192.168.1.0/24"]
   ```

Integration with CI/CD

GitLab CI Example

yaml

```
# .gitlab-ci.yml
deploy:
  stage: deploy
  script:
    - ansible-playbook -i inventory/hosts site.yml
  only:
    - main
```

Jenkins Pipeline Example

```groovy
// Jenkinsfile
pipeline {
    agent any
    stages {
        stage('Deploy') {
            steps {
                ansiblePlaybook(
                    playbook: 'site.yml',
                    inventory: 'inventory/hosts'
                )
            }
        }
    }
}
```

Support

- Check the main project documentation
- Review Ansible logs: `tail -f ansible.log`
- Use verbose mode for debugging: `-vvv`
- Test individual tasks with `--start-at-task`