Cisco Router Automation with Ansible

This repository contains Ansible playbooks and roles for automating VPN tunnel and direct link configurations on Cisco IOS routers across multiple datacenters (EQX, KKB) and AWS connections.

? Table of Contents

- Architecture Overview
- Prerequisites
- Quick Start
- Deployment Commands
- Network Topology
- Configuration Structure
- Troubleshooting

?? Architecture Overview

Datacenter Infrastructure

- **EQX Datacenter**: 2 routers (master/slave) running BGP AS 65401/65402
- **KKB Datacenter**: 2 routers (master/slave) running BGP AS 65501/65502
- **AWS Integration**: Direct Connect and IPsec VPN connections
- **Fortigate Firewalls**: Local firewalls at each datacenter for traffic filtering

Link Types

1. **IPsec VPN Tunnels**: Encrypted tunnels between datacenters and to AWS
2. **DWDM Direct Links**: High-speed fiber optic connections between EQX and KKB
3. **AWS Direct Connect**: Dedicated connections to AWS via IST and FR POPs
4. **Fortigate Links**: BGP peering with local firewalls

? Prerequisites

```
# Python environment (recommended: pyenv + virtualenv)
pyenv virtualenv 3.x routers
pyenv activate routers

# Install Ansible
pip install ansible

# Install required collections (if any)
ansible-galaxy collection install ansible.netcommon
```

? Quick Start

1. Clone and Setup

```
cd /path/to/cisco-automation-ansible
pyenv activate routers  # or your Python environment
```

2. Verify Inventory

```
# List all routers
ansible -i hosts.ini all_routers --list-hosts

# Test connectivity (dry-run)
ansible -i hosts.ini all_routers -m debug -a 'var=links' --connection=local
```

3. Deploy a Link

```
# Deploy specific link to a router
ansible-playbook -i hosts.ini deploy_tunnel.yml \
  --limit eqx-master \
  --extra-vars "link_name=TO-KKB-IPSEC-1"
```

? Deployment Commands by Router

? eqx-master (EQX Master - AS 65401)

```
# Inter-Datacenter Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-KKB-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-KKB-DWDM-1"

# Local Firewall
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-FW-FORTI-1"

# AWS Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-AWS-PROD-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-AWS-PROD-IPSEC-2"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-AWS-PROD-IST-DCON-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master --extra-vars "link_name=TO-AWS-PROD-FR-DCON-2"
```

View as table

Link Name	Type	Destination	Description
TO-KKB-IPSEC-1	IPsec VPN	KKB Datacenter	Primary backup tunnel to KKB (prepend 2)
TO-KKB-DWDM-1	Direct/DWDM	KKB Datacenter	High-speed fiber to KKB (primary)
TO-FW-FORTI-1	Direct/BGP	Local Firewall	AS 65001, receives KKB+AWS, distributes EQX
TO-AWS-PROD-IPSEC-1	IPsec VPN	AWS	Primary backup to AWS (prepend 3)
TO-AWS-PROD-IPSEC-2	IPsec VPN	AWS	Secondary backup to AWS (prepend 4)

Link Name	Type	Destination	Description
TO-AWS-PROD-IST-DCON-1	Direct Connect	AWS Istanbul	Direct Connect via Istanbul POP (prepend 1)
TO-AWS-PROD-FR-DCON-2	Direct Connect	AWS Frankfurt	Direct Connect via Frankfurt POP (prepend 2)

? eqx-slave (EQX Slave - AS 65402)

```
# Inter-Datacenter Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-KKB-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-KKB-DWDM-1"

# Local Firewall
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-FW-FORTI-1"

# AWS Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-AWS-PROD-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-AWS-PROD-IPSEC-2"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-AWS-PROD-IST-DCON-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-slave --extra-vars "link_name=TO-AWS-PROD-FR-DCON-2"
```

View as table

Link Name	Type	Destination	Description
TO-KKB-IPSEC-1	IPsec VPN	KKB Datacenter	Backup tunnel to KKB (prepend 3)
TO-KKB-DWDM-1	Direct/DWDM	KKB Datacenter	High-speed fiber to KKB (primary)
TO-FW-FORTI-1	Direct/BGP	Local Firewall	AS 65001, receives KKB+AWS, distributes EQX

Link Name	Type	Destination	Description
TO-AWS-PROD-IPSEC-1	IPsec VPN	AWS	Primary backup to AWS (prepend 3)
TO-AWS-PROD-IPSEC-2	IPsec VPN	AWS	Secondary backup to AWS (prepend 4)
TO-AWS-PROD-IST-DCON-1	Direct Connect	AWS Istanbul	Direct Connect via Istanbul POP (prepend 1)
TO-AWS-PROD-FR-DCON-2	Direct Connect	AWS Frankfurt	Direct Connect via Frankfurt POP (prepend 2)

? kkb-master (KKB Master - AS 65501)

```
# Inter-Datacenter Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-EQX-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-EQX-DWDM-1"

# Local Firewall
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-FW-FORTI-1"

# AWS Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-AWS-PROD-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-AWS-PROD-IPSEC-2"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-AWS-PROD-IST-DCON-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-master --extra-vars "link_name=TO-AWS-PROD-FR-DCON-2"
```

View as table

Link Name	Type	Destination	Description
TO-EQX-IPSEC-1	IPsec VPN	EQX Datacenter	Primary backup tunnel to EQX (prepend 2)

Link Name	Type	Destination	Description
TO-EQX-DWDM-1	Direct/DWDM	EQX Datacenter	High-speed fiber to EQX (primary)
TO-FW-FORTI-1	Direct/BGP	Local Firewall	AS 65000, receives EQX+AWS, distributes KKB
TO-AWS-PROD-IPSEC-1	IPsec VPN	AWS	Primary backup to AWS (prepend 3)
TO-AWS-PROD-IPSEC-2	IPsec VPN	AWS	Secondary backup to AWS (prepend 4)
TO-AWS-PROD-IST-DCON-1	Direct Connect	AWS Istanbul	Direct Connect via Istanbul POP (prepend 1)
TO-AWS-PROD-FR-DCON-2	Direct Connect	AWS Frankfurt	Direct Connect via Frankfurt POP (prepend 2)

? kkb-slave (KKB Slave - AS 65502)

```
# Inter-Datacenter Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-EQX-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-EQX-DWDM-1"

# Local Firewall
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-FW-FORTI-1"

# AWS Links
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-AWS-PROD-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-AWS-PROD-IPSEC-2"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-AWS-PROD-IST-DCON-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit kkb-slave --extra-vars "link_name=TO-AWS-PROD-FR-DCON-2"
```

View as table

Link Name	Type	Destination	Description
TO-EQX-IPSEC-1	IPsec VPN	EQX Datacenter	Backup tunnel to EQX (prepend 3)
TO-EQX-DWDM-1	Direct/DWDM	EQX Datacenter	High-speed fiber to EQX (primary)
TO-FW-FORTI-1	Direct/BGP	Local Firewall	AS 65000, receives EQX+AWS, distributes KKB
TO-AWS-PROD-IPSEC-1	IPsec VPN	AWS	Primary backup to AWS (prepend 3)
TO-AWS-PROD-IPSEC-2	IPsec VPN	AWS	Secondary backup to AWS (prepend 4)
TO-AWS-PROD-IST-DCON-1	Direct Connect	AWS Istanbul	Direct Connect via Istanbul POP (prepend 1)
TO-AWS-PROD-FR-DCON-2	Direct Connect	AWS Frankfurt	Direct Connect via Frankfurt POP (prepend 2)

? Development Routers

dev-eqx-master

```
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-eqx-master --extra-vars
"link_name=TO-KKB-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-eqx-master --extra-vars
"link_name=TO-KKB-DWDM-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-eqx-master --extra-vars
"link_name=TO-FW-FORTI-1"
```

dev-eqx-slave

```
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-eqx-slave --extra-vars
"link_name=TO-KKB-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-eqx-slave --extra-vars
"link_name=TO-KKB-DWDM-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-eqx-slave --extra-vars
"link_name=TO-FW-FORTI-1"
```

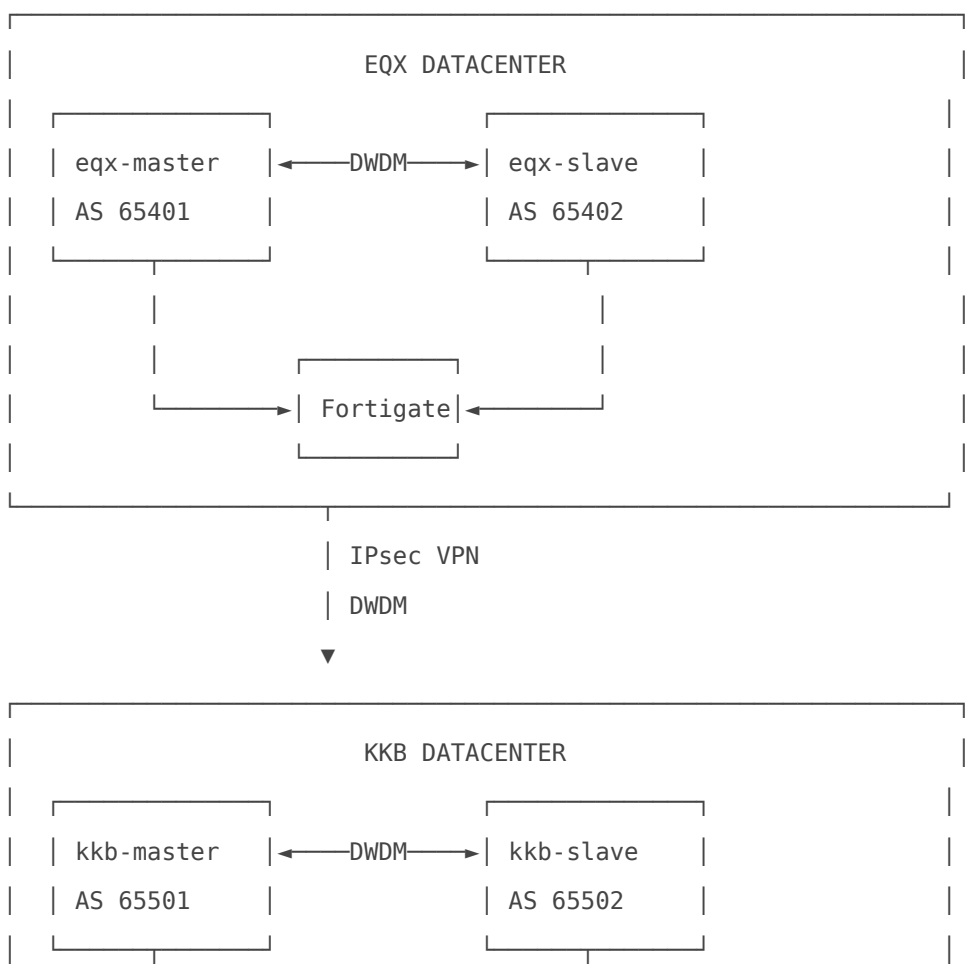
dev-kkb-master

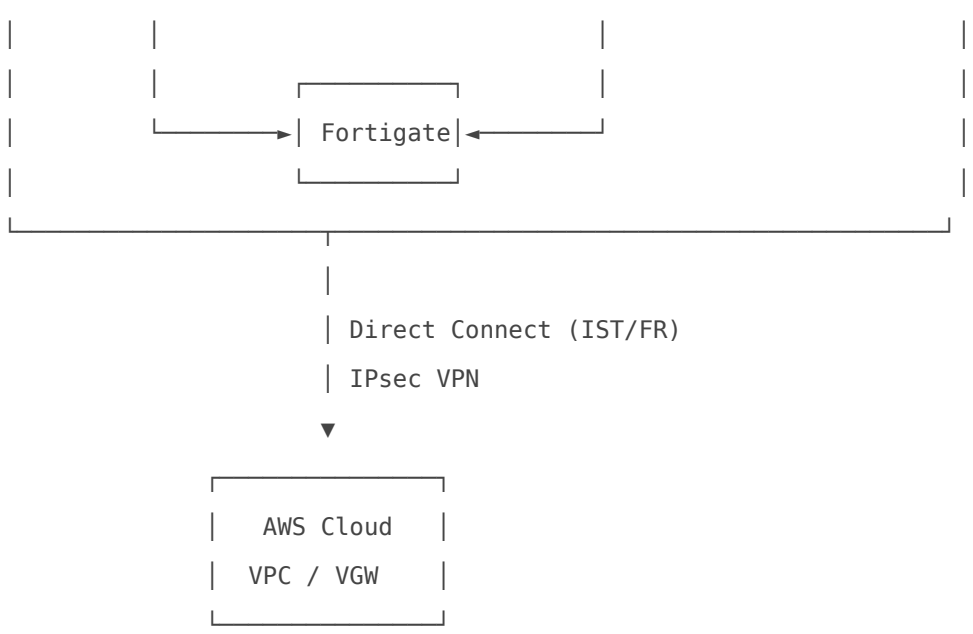
```
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-kkb-master --extra-vars
"link_name=TO-EQX-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-kkb-master --extra-vars
"link_name=TO-EQX-DWDM-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-kkb-master --extra-vars
"link_name=TO-FW-FORTI-1"
```

dev-kkb-slave

```
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-kkb-slave --extra-vars
"link_name=TO-EQX-IPSEC-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-kkb-slave --extra-vars
"link_name=TO-EQX-DWDM-1"
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit dev-kkb-slave --extra-vars
"link_name=TO-FW-FORTI-1"
```

? Network Topology





? Configuration Structure

```
cisco-automation-ansible/
├── ansible.cfg                 # Ansible configuration
├── hosts.ini                   # Inventory file with all routers
├── deploy_tunnel.yml           # Main playbook
├── group_vars/
│   ├── all.yml                # Shared network lists (centralized)
│   ├── eqx_masters.yml        # EQX master router link configs
│   ├── eqx_slaves.yml         # EQX slave router link configs
│   ├── kkb_masters.yml        # KKB master router link configs
│   └── kkb_slaves.yml         # KKB slave router link configs
├── host_vars/
│   ├── eqx-master.yml         # Per-host variables (prod)
│   ├── eqx-slave.yml
│   ├── kkb-master.yml
│   ├── kkb-slave.yml
│   ├── dev-eqx-master.yml     # Per-host variables (dev)
│   ├── dev-eqx-slave.yml
│   ├── dev-kkb-master.yml
│   └── dev-kkb-slave.yml
└── roles/
    └── vpn_tunnel/
        └── tasks/
```

```
├── main.yml       # Entry point with link selection
├── ipsec.yml      # IPsec tunnel configuration
├── bgp.yml        # BGP neighbor & routing config
└── interface.yml  # Interface configuration
```

Centralized Network Configuration

All network prefixes are defined once in `group_vars/all.yml`:

- **eqx_receive_networks / eqx_distribute_networks**: 18 EQX datacenter networks
- **kkb_distribute_networks**: 13 KKB datacenter networks
- **aws_receive_networks**: AWS VPC networks (172.16.110.0/24)
- **aws_distribute_networks**: Networks announced to AWS
- **forti_receive_in_eqx / forti_distribute_in_eqx**: Combined networks for EQX Fortigate
- **forti_receive_in_kkb / forti_distribute_in_kkb**: Combined networks for KKB Fortigate

This DRY (Don't Repeat Yourself) approach ensures consistency and easier maintenance.

? Advanced Usage

Deploy Multiple Links

```
# Deploy all links on a router (use with caution!)
ansible-playbook -i hosts.ini deploy_tunnel.yml --limit eqx-master
```

Deploy to Multiple Routers

```
# Deploy same link to all EQX routers
ansible-playbook -i hosts.ini deploy_tunnel.yml \
  --limit eqx_masters \
  --extra-vars "link_name=TO-KKB-IPSEC-1"

# Deploy to both datacenters
ansible-playbook -i hosts.ini deploy_tunnel.yml \
  --limit "eqx-master,kkb-master" \
  --extra-vars "link_name=TO-FW-FORTI-1"
```

Dry-Run / Check Mode

```
# Test without making changes
ansible-playbook -i hosts.ini deploy_tunnel.yml \
  --limit eqx-master \
  --extra-vars "link_name=TO-KKB-IPSEC-1" \
  --check
```

Selective Configuration

Links support per-component deployment flags:

```
configure:
  bgp: true        # Deploy BGP configuration
  ipsec: true      # Deploy IPsec configuration
  interface: true  # Deploy interface configuration
```

To skip specific components, set them to `false` in the link definition.

? Troubleshooting

Common Issues

Issue: "link_name is not defined"

```
# Solution: Always specify link_name
ansible-playbook -i hosts.ini deploy_tunnel.yml \
  --limit eqx-master \
  --extra-vars "link_name=TO-KKB-IPSEC-1"
```

Issue: "Unable to connect to router"

```
# Check SSH connectivity
ansible -i hosts.ini eqx-master -m ping

# Verify credentials in group_vars/all.yml
ansible_user: admin
ansible_password: admin
```

Issue: "Jinja2 template error"

```
# Validate configuration syntax
ansible -i hosts.ini eqx-master -m debug -a 'var=links' --connection=local
```

Debug Commands

```
# List all configured links for a router
ansible -i hosts.ini eqx-master -m debug -a 'var=links' --connection=local

# Check specific link configuration
ansible -i hosts.ini eqx-master -m debug \
  -a 'var=links[0]' \
  --connection=local

# Verify network lists
ansible -i hosts.ini eqx-master -m debug \
  -a 'var=eqx_receive_networks' \
  --connection=local
```

Validation

```
# Validate all routers can render their configurations
ansible -i hosts.ini all_routers -m debug \
  -a 'var=links[0].name' \
  --connection=local | grep SUCCESS
```

? Best Practices

1. **Always test in dev environment first**: Use dev routers before deploying to production
2. **Deploy one link at a time**: Easier to troubleshoot and rollback
3. **Use version control**: Commit changes before deploying to production
4. **Document changes**: Update this README when adding new links or routers
5. **Backup configurations**: Save router configs before making changes
6. **Monitor BGP sessions**: Verify BGP neighbors come up after deployment

? Security Notes

- Credentials are stored in `group_vars/all.yml` - ensure proper file permissions (600)
- Consider using Ansible Vault for sensitive data:

```
ansible-vault encrypt group_vars/all.yml
```

- IPsec pre-shared keys should be rotated regularly
- Limit SSH access to Ansible control node only

? BGP AS Numbers

Router	AS Number	Role
eqx-master	65401	EQX Primary
eqx-slave	65402	EQX Secondary
kkb-master	65501	KKB Primary
kkb-slave	65502	KKB Secondary
Fortigate (EQX)	65001	EQX Firewall
Fortigate (KKB)	65000	KKB Firewall

? Additional Resources

- Ansible Network Automation Guide
- Cisco IOS Configuration Guide
- BGP Best Practices

? License

Internal use only - Proprietary

? Support

For issues or questions, contact the Network Operations team.

Last Updated: October 2025

Revision #2
Created 2025-10-19 08:54:13 UTC by Mesut Bayrak
Updated 2025-10-19 08:56:42 UTC by Mesut Bayrak