

Documentation Style Guide

This style guide ensures consistency, clarity, and maintainability across all documentation in the HX Infrastructure Ansible project.



General Principles

Clarity and Accessibility

- Write for your audience - assume basic Ansible knowledge but explain complex concepts
- Use active voice and present tense
- Avoid jargon and acronyms without explanation
- Provide context and examples for all procedures

Consistency

- Follow established patterns and templates
- Use consistent terminology throughout
- Maintain uniform formatting and structure
- Apply standardized naming conventions

Maintainability

- Keep documentation close to code
- Use version control for all changes
- Include update dates and version information
- Provide clear ownership and contact information



Markdown Standards

Headings

Use hierarchical heading structure with proper nesting:

```
# Page Title (H1) - One per document
## Major Section (H2)
### Subsection (H3)
#### Detail Section (H4) - Avoid deeper nesting
```

Guidelines:

- Use sentence case for headings: “Getting started” not “Getting Started”
- Include emoji icons for major sections when appropriate
- Avoid skipping heading levels
- Keep headings descriptive and scannable

Code Blocks

Always specify language for syntax highlighting:

```
# Good - with language specification
---
- name: Example task
  debug:
    msg: "Hello World"
```

```
# Avoid - no language specification
---
- name: Example task
  debug:
    msg: "Hello World"
```

Best Practices:

- Use `yaml` for Ansible content
- Use `bash` for shell commands
- Use `json` for JSON data
- Include comments in code examples
- Test all code examples for accuracy

Links and References

Use descriptive link text and relative paths:

```
# Good
See the [deployment guide](../operations/deployment.md) for details.
Learn about [role variables](roles/hx_ca_trust.md#variables).

# Avoid
Click [here](../operations/deployment.md) for more info.
See [this link](https://example.com/docs).
```

Guidelines:

- Use relative paths for internal links
- Include section anchors when linking to specific content
- Verify all links work correctly
- Provide context for external links

Lists and Tables

Use consistent formatting for lists and tables:

Unordered lists

- First item with proper spacing
- Second item
 - Nested item with 2-space indentation
 - Another nested item

Ordered lists

1. First step
2. Second step
3. Third step

Tables with proper alignment

Column 1	Column 2	Column 3
-----	-----	-----
Value 1	Value 2	Value 3
Data A	Data B	Data C

Admonitions and Callouts

Use admonitions for important information:

```
!!! note "Information"
    This provides additional context or helpful information.

!!! warning "Important"
    This highlights something that requires attention.

!!! danger "Critical"
    This indicates potential risks or critical issues.

!!! tip "Best Practice"
    This shares recommended approaches or optimizations.
```

Content Structure

Document Templates

Each document type should follow a consistent structure:

Role Documentation

```
# Role Name

## Overview
Brief description and purpose

## Requirements
Prerequisites and dependencies

## Variables
Detailed variable documentation with examples

## Usage
Examples and common use cases

## Testing
How to test the role

## Troubleshooting
Common issues and solutions
```

Operational Procedures

```
# Procedure Name

## Purpose
What this procedure accomplishes

## Prerequisites
Required access, tools, and conditions

## Steps
Detailed step-by-step instructions

## Verification
How to confirm success

## Rollback
How to undo changes if needed

## Troubleshooting
Common issues and solutions
```

Variable Documentation

Document all variables with consistent format:

```
# Variable name with clear description
variable_name: default_value
# Type: string|boolean|integer|list|dict
# Required: yes|no
# Description: Detailed explanation of purpose and usage
# Example: variable_name: "example_value"
# Valid values: list of acceptable values if applicable
```

Code Examples

Provide complete, working examples:

```
# Complete playbook example
---
- name: Example playbook
  hosts: all
  become: yes
  vars:
    example_var: "value"

  tasks:
    - name: Example task with clear description
      debug:
        msg: "{{ example_var }}"
      tags:
        - example
```

Formatting Standards

File Naming

Use consistent naming conventions:

- **Files:** snake_case.md (e.g., deployment_guide.md)
- **Directories:** kebab-case (e.g., user-guides)
- **Images:** descriptive-name.png (e.g., architecture-diagram.png)

Line Length

- **Maximum:** 120 characters per line
- **Preferred:** 80-100 characters for readability
- **Code blocks:** Can exceed for readability
- **Tables:** Can exceed when necessary

Whitespace and Formatting

- Use 2 spaces for indentation in YAML
- Include blank lines around headings and sections
- Use consistent spacing in lists and tables
- Remove trailing whitespace

Cross-References

Internal Links

Create a web of interconnected documentation:

```
# Reference related content
For more details, see:
- [Security hardening](../security/hardening.md)
- [Backup procedures](../operations/backup.md)
- [Troubleshooting guide](../operations/troubleshooting.md)

# Link to specific sections
Review the [variable configuration](roles/common.md#variables) section.
```

External References

Provide authoritative external sources:

Reference official documentation

Based on [Ansible best practices](https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html).

Include version-specific links when relevant

Compatible with [Ansible 2.12+](https://docs.ansible.com/ansible/2.12/).

Quality Checklist

Before publishing documentation, verify:

Content Quality

- ☐ Information is accurate and up-to-date
- ☐ Examples are tested and working
- ☐ All links are functional
- ☐ Content follows logical flow
- ☐ Audience needs are addressed

Technical Accuracy

- ☐ Code examples are syntactically correct
- ☐ Variable names match actual implementation
- ☐ Procedures have been tested
- ☐ Version compatibility is specified
- ☐ Dependencies are documented

Style Compliance

- ☐ Headings follow hierarchy rules
- ☐ Code blocks specify language
- ☐ Links use descriptive text
- ☐ Formatting is consistent
- ☐ Grammar and spelling are correct

Accessibility

- ☐ Alt text provided for images
- ☐ Tables include headers
- ☐ Color is not the only way to convey information
- ☐ Content is screen reader friendly
- ☐ Language is clear and inclusive

Maintenance

Regular Reviews

- **Monthly:** Check for outdated information
- **Quarterly:** Review and update examples

- **Annually:** Comprehensive style guide review
- **As needed:** Update for major changes

Version Control

- Use meaningful commit messages
- Tag documentation versions
- Maintain changelog for major updates
- Archive obsolete documentation

Feedback Integration

- Monitor user feedback and questions
- Update documentation based on common issues
- Incorporate suggestions from team members
- Track documentation usage and effectiveness

Note: This style guide is a living document. Suggest improvements through the standard contribution process.