



# Evolving real-world AsciiDoc<sup>®</sup> into a specification and how it will help the ecosystem

Alexander Schwartz, Principal Software Engineer @ Red Hat  
FOSDEM | Brussels, BE | 2024-02-01

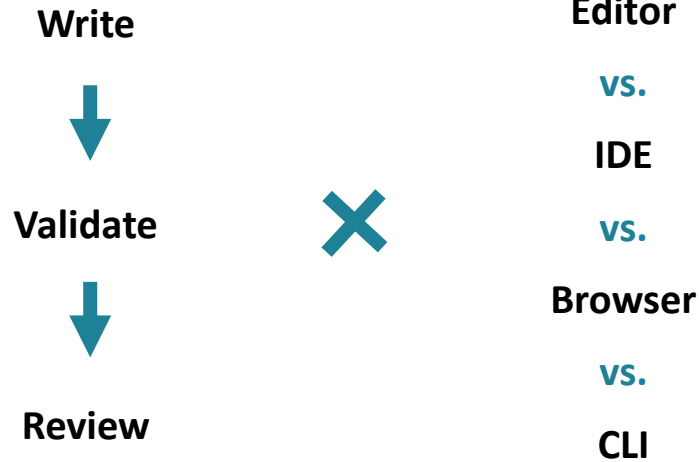
- 1 Mission: Get the docs out to the readers!
- 2 Workflow: Author, Convert, Publish
- 3 AsciiDoc Brand and Language Documentation
- 4 AsciiDoc Language Specification

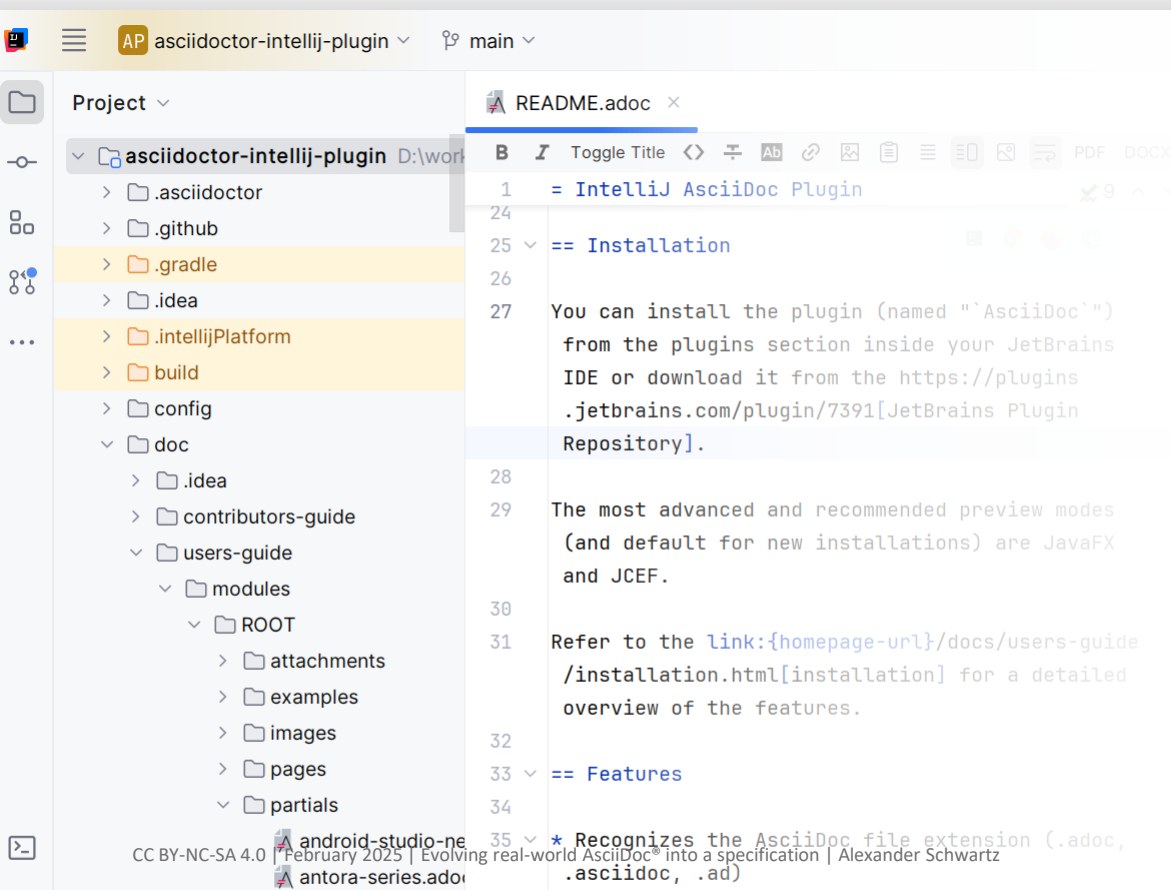
- 1 **Mission: Get the docs out to the readers!**
- 2 Workflow: Author, Convert, Publish
- 3 AsciiDoc Brand and Language Documentation
- 4 AsciiDoc Language Specification

- Authors should focus on content and its structure
- Conversion should support multiple targets from a single source
- Published documentation educates users and provides answers to their questions



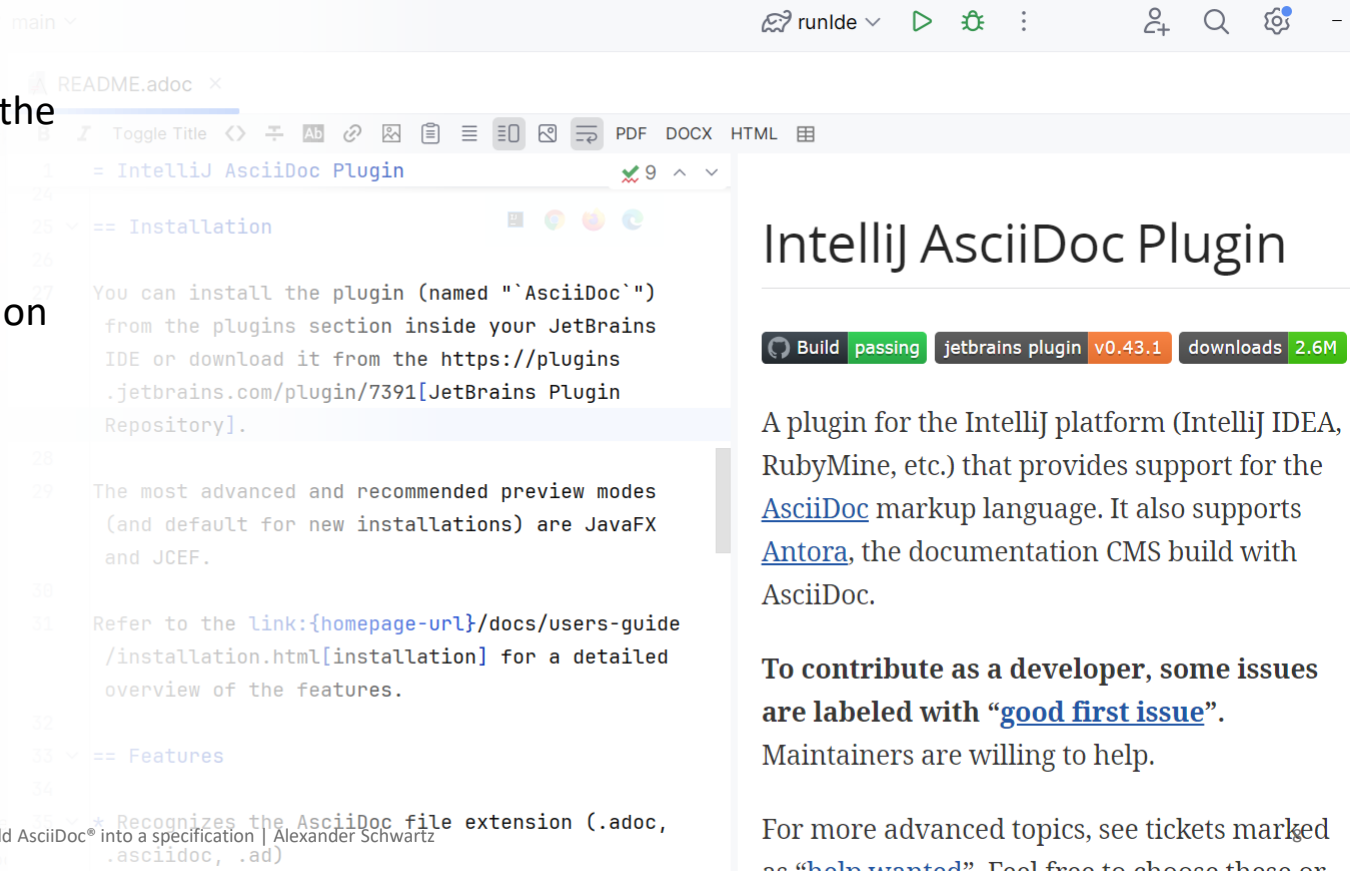
- 1 Mission: Get the docs out to the readers!
- 2 **Workflow: Author, Convert, Publish**
- 3 AsciiDoc Brand and Language Documentation
- 4 AsciiDoc Language Specification





- Write without distraction
- Syntax highlighting
- Autocompletion
- Grammar and style check

- Scrolling and navigate the context
- Styled Preview
- Collaborate using version control



The screenshot shows an IDE interface with a source editor on the left and a preview pane on the right. The source editor displays AsciiDoc code for a README file, including sections for installation and features. The preview pane shows the rendered HTML output, featuring a title, a build status bar, and a main text block.

**IntelliJ AsciiDoc Plugin**

Build passing jetbrains plugin v0.43.1 downloads 2.6M

A plugin for the IntelliJ platform (IntelliJ IDEA, RubyMine, etc.) that provides support for the [AsciiDoc](#) markup language. It also supports [Antora](#), the documentation CMS build with AsciiDoc.

**To contribute as a developer, some issues are labeled with “[good first issue](#)”.** Maintainers are willing to help.

For more advanced topics, see tickets marked as “[help wanted](#)”. Feel free to choose those or

\* Recognizes the AsciiDoc file extension (.adoc, .asciidoc, .ad)



Each sentence should be in a separate line.

Every blank line creates a new paragraph.

You can write **\*bold\*** text.

*\_Italic\_* works as well text.

Lists

- . work
- . with
- . dots

to make them numbered and

- \* asterisks
- \* to
- \* make

bullet points.

Each sentence should be in a separate line.

Every blank line creates a new paragraph.

You can write **bold** text. *Italic* works as well text.

Lists

1. work
2. with
3. dots

to make them numbered and

- asterisks
- to
- make

bullet points.

```
[IMPORTANT]
```

```
--
```

```
You can mark a paragraph to be important.
```

```
--
```

```
TIP: There is also the shorter one-line-  
syntax.
```

```
====
```

```
This is an example block. Use it to provide  
examples to your users.
```

```
====
```

```
----
```

```
This is a listing.  
You can provide syntax highlighting for  
XML/Java etc. here.
```

```
----
```

```
IMPORTANT
```

```
You can mark a paragraph to be important.
```

```
TIP
```

```
There is also the shorter one-line-syntax.
```

```
This is an example block. Use it to provide  
examples to your users.
```

```
This is a listing.  
You can provide syntax highlighting for XML/Java  
etc. here.
```

Tables are blocks, `|===` marks their beginning and end.`

```
.Table title
[cols="1,1,3a,1"]
```

```
|===
```

```
|H1 |H2 |H3 |H4
```

```
|Cell 1
```

```
|Cell 2
```

|Formatting **\*of words\*** works like in *\_before\_* if you switch the column or cell to AsciiDoc format.

```
|Cell 4
```

```
|===
```

Tables are blocks, `|===` marks their beginning and end.`

*Table 1. Table title*

H1	H2	H3	H4
Cell 1	Cell 2	Formatting <b>of words</b> works like in <i>before</i> if you switch the column or cell to AsciiDoc format.	Cell 4

.Standard include

[,java]

----

**include::**Example.java[]

----

**:icons: font**

.Pick a block with callout

[,java,indent=0]

----

**include::**Example.java[tag=sop]

----

<1> this is your first line of Java

### *Standard include*

```
public class Example {
    public static int main() {
        // tag::sop[]
        System.out.println("Hello world!"); // <1>
        // end::sop[]
    }
}
```

### *Pick a block with callout*

```
System.out.println("Hello world!"); ①
```

① this is your first line of Java

```
# .vale.ini
```

```
StylesPath = .github/styles
MinAlertLevel = suggestion
```

```
Packages = RedHat
```

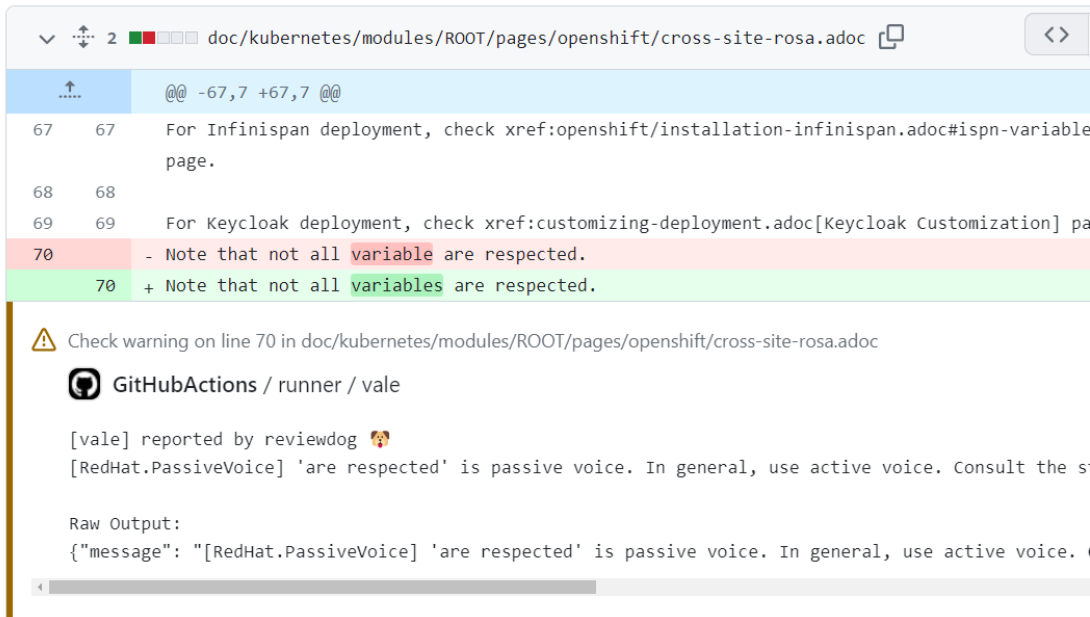
```
IgnoredClasses = error-message
```

```
[*.adoc]
```

```
BasedOnStyles = RedHat
```

```
RedHat.TermsErrors = warning
```

<https://github.com/errata-ai/vale-action>



doc/kubernetes/modules/ROOT/pages/openshift/cross-site-rosa.adoc

67 67 For Infinispan deployment, check xref:openshift/installation-infinispan.adoc#ispn-variable page.

68 68

69 69 For Keycloak deployment, check xref:customizing-deployment.adoc[Keycloak Customization] page.

70 - Note that not all variable are respected.

70 + Note that not all variables are respected.

⚠ Check warning on line 70 in doc/kubernetes/modules/ROOT/pages/openshift/cross-site-rosa.adoc

GitHubActions / runner / vale

[vale] reported by reviewdog 🐕

[RedHat.PassiveVoice] 'are respected' is passive voice. In general, use active voice. Consult the s

Raw Output:

{\"message\": \"[RedHat.PassiveVoice] 'are respected' is passive voice. In general, use active voice.\"" data-bbox="433 325 1000 896"/>

**HTML**

vs.

**PDF**

vs.

**DocBook**

vs.

...



**Maven**

vs.

**npm**

vs.

**CLI**

vs.

...



**Asciidoctor**

(Ruby, JavaScript, Java)

vs.

**libasciidoc**  
(Go)

vs.

**Asciidoc-hs**  
(Haskell)

vs.

...

```
$ asciidoctor test.adoc
asciidoctor: WARNING: test.adoc: line 2: multiple ids
detected in style attribute
```

```
= Hello, AsciiDoc!
```

```
This is an interactive editor.
Use it to try https://asciidoc.org[AsciiDoc].
```

```
== Section Title
```

```
* A list item
* Another list item|
```

```
[,ruby]
----
puts 'Hello, World!'
----
```

## Hello, AsciiDoc!

This is an interactive editor. Use it to try [AsciiDoc](https://asciidoc.org).

### Section Title

- A list item
- Another list item

```
puts 'Hello, World!'
```

Write extensions that

- Extend the AsciiDoc syntax by adding new macros
- Interact with a document on an AST level
- Generate AsciiDoc content that can be rendered in any supported output format
- ...







- ▶ What's New in Antora 3.1
- How Antora Can Help
- How Antora Works
- Install and Run Antora Quickstart
- ▶ Get Antora
- ▶ Set Up Content Sources
- ▶ Configure Component Versions
- ▶ Resources and Resource IDs
- ▶ Pages
- ▶ Images
- ▶ Partials
- ▶ Examples
  - Attachments
- ▶ AsciiDoc Primer
- ▶ Navigation
- ▶ Configure Your Site



# Antora Documentation

## The Static Site Generator for Tech Writers

This site hosts the technical documentation for Antora. Antora makes it easy for tech writers to create documentation and publish it to the web. As a tech writer, you focus on authoring content in [AsciiDoc](#), Antora's content markup language. You organize those files into a standard project structure and store them in one or more content (git) repositories. Antora picks up your content from there and transforms it into a website. That's all there is to it!

## Manage docs as code

With Antora, you manage **docs as code**. That means your documentation process benefits from the same practices used to produce successful software

### Contents

[Manage docs as code](#)

[Where to begin](#)

## Antora

- ▶ What's New in Antora 3.1
  - How Antora Can Help
  - How Antora Works
  - Install and Run Antora Quickstart
- ▶ Get Antora
- ▶ Set Up Content Sources
- ▶ Configure Component Versions
- ▶ Resources and Resource IDs
- ▶ Pages
- ▶ Images
- ▶ Partials
- ▶ Examples
  - Attachments
- ▶ AsciiDoc Primer
- ▶ Navigation
- ▶ Configure Your Site



Antora / Antora Documentation

# Antora Documentation

## The Static Site Generator for Tech Writers

This site hosts the technical documentation for Antora. It's easy for tech writers to create documentation and use Antora as a tech writer, you focus on authoring content in AsciiDoc markup language. You organize those files into a content repository and store them in one or more content (git) repositories. Antora pulls your content from there and transforms it into a website for you to it!

## Manage docs as code

With Antora, you manage docs as code. That means your build process benefits from the same practices used to manage code in software development.

- Static site publishing for AsciiDoc content
- Generates a simple to host website
- Fully automated publishing
- Customizable Theme to match your design
- Navigation outline
- Site Search

- Page outline
- Custom Navigation
- Page linking
- Redirects when pages have been renamed

The screenshot shows the top navigation bar of the Antora documentation website. It includes a search box labeled "Search the docs", and navigation links for "Projects", "Tooling", and "Community", each with a dropdown arrow. A Twitter icon is located on the far right. Below the navigation bar, the page title "Antora Documentation" is displayed on the left, and the version "3.1" and "Edit this Page" link are on the right. The main content area features a large heading "Antora Documentation" and a sub-heading "Antora: A Site Generator for Tech Writers". The text describes Antora's role in generating technical documentation for Antora, mentioning its use of AsciiDoc and its ability to pull content from git repositories. A right-hand sidebar titled "Contents" lists "Manage docs as code" and "Where to begin". At the bottom, a section titled "Manage docs as code" begins with the text "When you manage docs as code. That means your documentation benefits from the same practices used to produce successful".

Search the docs

Projects ▾

Tooling ▾

Community ▾

Twitter

Antora Documentation

3.1 ▾ Edit this Page

# Antora Documentation

## Antora: A Site Generator for Tech Writers

Antora generates the technical documentation for Antora. Antora makes it easy for tech writers to create documentation and publish it to the web. As a result, you focus on authoring content in [AsciiDoc](#), Antora's content authoring language. You organize those files into a standard project structure stored in one or more content (git) repositories. Antora picks up the content from there and transforms it into a website. That's all there is to it.

### Manage docs as code

When you manage docs as code. That means your documentation benefits from the same practices used to produce successful

#### Contents

- Manage docs as code
- Where to begin

### Author:

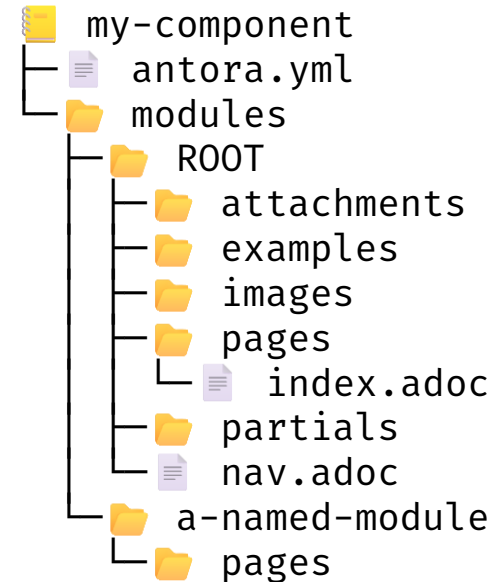
- Creates content in a well-defined folder structure
- Customizes navigation and versions

### Docu Ops:

- Prepares automation scripts for validation and publishing

### Web Developer:

- Customizes the UI theme



- 1 Mission: Get the docs out to the readers!
- 2 Workflow: Author, Convert, Publish
- 3 AsciiDoc Brand and Language Documentation**
- 4 AsciiDoc Language Specification

AsciiDoc® and AsciiDoc Language™ are trademarks of the Eclipse Foundation, Inc.

For questions related to these brand guidelines please reach out to the AsciiDoc Working Group.



Visit <https://asciidoc.org/> to learn about AsciiDoc and links to all resources.



About

Compare

Specifications

Tools

Docs

**Publish presentation-rich content from  
a concise and comprehensive  
authoring format.**



Visit <https://docs.asciidoctor.org/asciidoc/latest/> to learn about the language.



- ▶ Introduction
- ▶ Blocks
- ▶ Document Attributes
- ▶ Element Attributes
- ▶ Document Header
  - Document Type
- ▶ Sections
- ▶ Paragraphs
  - Discrete Headings
  - Breaks
- ▶ Text Formatting and

# AsciiDoc Language Documentation

## About AsciiDoc


AsciiDoc is a lightweight and semantic markup language primarily designed for writing technical documentation. The language can be used to produce a variety of presentation-rich output formats, all from content encoded in a concise, human-readable, plain text format.

### Contents

- [About AsciiDoc](#)
- [About this documentation](#)




- 1 Mission: Get the docs out to the readers!
- 2 Workflow: Author, Convert, Publish
- 3 AsciiDoc Brand and Language Documentation
- 4 **AsciiDoc Language Specification**

Visit <https://gitlab.eclipse.org/eclipse/asciidoc-lang> to get involved, and discuss at <https://chat.asciidoc.org/>




**A** **AsciiDoc Language** 

**Subgroups and projects**   Shared projects   Inactive




---

---

  **AsciiDoc Language**  ★ 20  
This project maintains the official specification, Technology Co...

---

  **AsciiDoc TCK**  ★ 7  
This project maintains the official Technology Compatibility Kit (T...

Community and experts agree on how to standardize the behavior in a best backwards compatible way.

## SDR-5: Describe Inline Syntax using Formal Grammar

---

Related issues

- [asciidoc-lang#16](#)
- [asciidoc-lang#22](#)

### Context

---

The inline syntax in the AsciiDoc language has never had a formal grammar, nor was it initially designed to have one. Inline syntax refers to the markup in the content of a non-verbatim leaf node (e.g., paragraph, heading, title, block content attribute, etc.). We herein refer to this content as regular text.

The inline markup in AsciiDoc is currently processed as a sequence of substitutions on regular text using

## TCK to check compatibility of implementations



= Document Title

body

```
{
  "name": "document",
  "type": "block",
  "attributes": {},
  "header": {
    "title": [
      {
        "name": "text",
        "type": "string",
        "value": "Document Title",
        "location": [{ "line": 1, "col": 3 }, { "line": 1, "col": 16 }]
      }
    ],
    "location": [{ "line": 1, "col": 1 }, { "line": 1, "col": 16 }]
  },
  ...
}
```

## AsciiDoc

<https://asciidoc.org>

## Asciidoctor

<https://asciidoctor.org>

## IntelliJ AsciiDoc Plugin

<https://ahus1.de/post/technical-writing-udemy>

## Antora

<https://antora.org/>

<https://www.ahus1.de/post/documentation-site-antora>

## Videos, slides and examples

<https://www.ahus1.de/post/asciidoctor-intro-and-deep-dive>

Slides:



[@ahus1@fosstodon.org](mailto:@ahus1@fosstodon.org)



[@ahus1.de](https://twitter.com/@ahus1.de)

# Contact



Alexander Schwartz  
Principal Software Engineer

[alexander.schwartz@gmx.net](mailto:alexander.schwartz@gmx.net)  
<https://www.ahus1.de>

 @ahus1.de

 @ahus1@fosstodon.org