The fundamentals of technical writing

Tooling

The Red Hat Customer Content Services team

About the authors:

Alexandra Nikandrova
Technical Writer for Red Hat
Former Devops

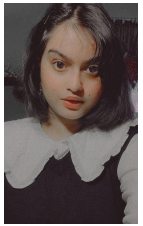
Jana Vrbkova
Technical Writer for Red Hat

Srikanth R
Technical Writer for Red Hat

About the authors:

Alexandra Nikandrova
Technical Writer.
Former Devops

Ioanna Gkioka
Technical Writing Manager
Support technical writers
Former university teacher



Kalyani Desai
I am a Technical Writer, Open source contributor,
Fun fact: I tried some strawberry shampoo.
It doesn't taste as good as it smells :P

Arati Ajit Belgaonkar
I am a Technical Writer by profession and personally, I am an athlete. My hobbies are gardening and swimming.

Dominika Vagnerova
Technical Writer
Former journalist

About the authors:

Vendula Ferschmannova
Global Manager

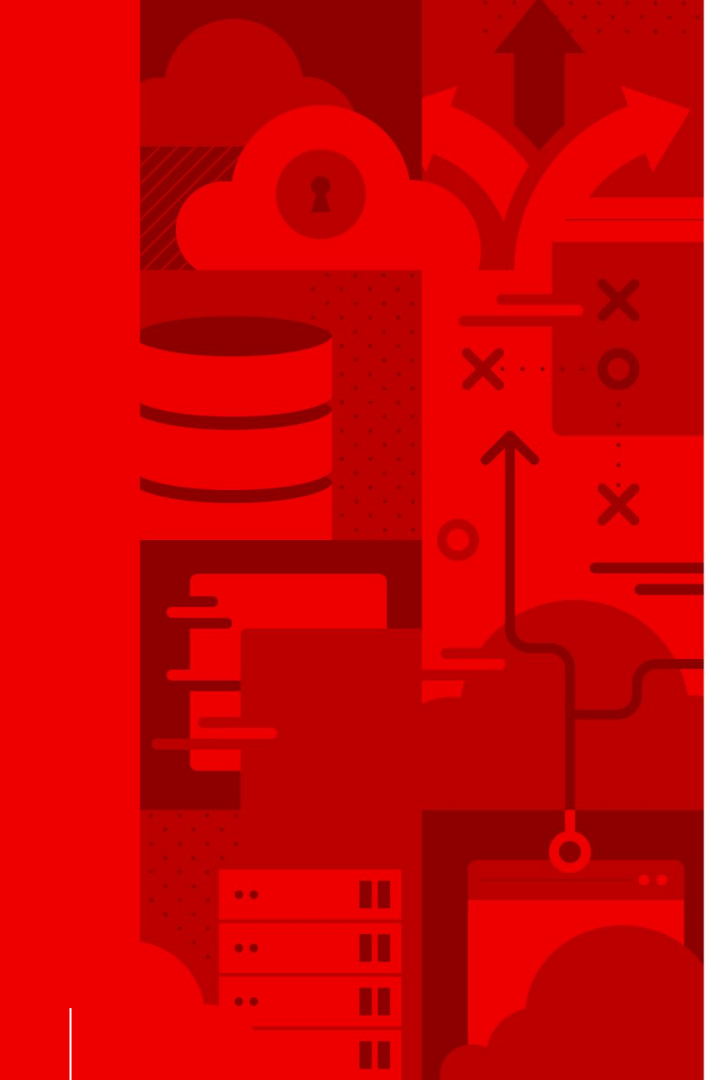
Mayur Patil
Technical Writer 2
RHEL Networking and RDMA

Petr Kovář
Documentation Program Manager

Open source documentation writer and translator.

What we'll discuss today

- Introduction
- Editors and word processors
- Languages
  - XML, DocBook
  - Dita toolkit
  - reStructuredText (RTS), Sphinx
- Collaboration tools
- Version control tools (git)
- Graphics



Introduction

Why?

Single sourcing and content reuse

- "*Write once, publish everywhere*"

Platform independence

- Consistent output across different platforms

Efficiency and consistency

- Unified structure, templating, and terminology

Why?

Docs as code

- Alignment with coding practices, versioning, and collaborative workflows

Automation and productivity

- Save time and reduce repetitive tasks

Focus on content

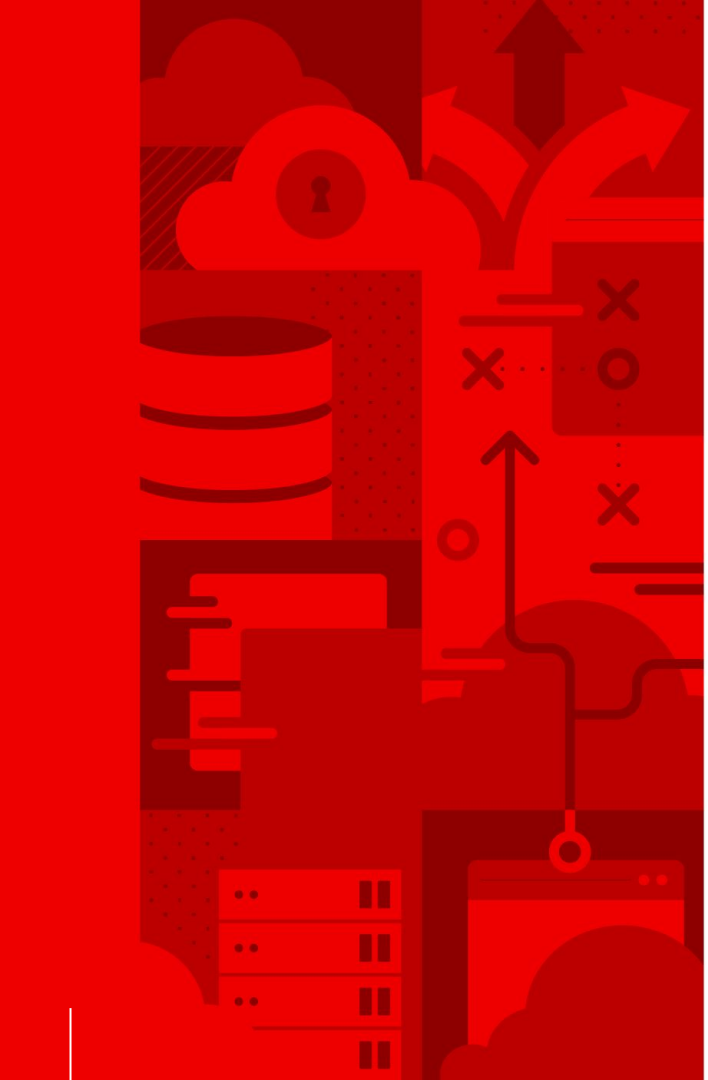
- Separation of content and formatting

Why

- Single source → multiple outputs
- Reusing text
- Unified structure and templating
  - Usability: Structure of each article in documentation must look similar. Must have the same font, highlighting, levelling, and use blank space

How

- Languages and structure created specifically for technical writers
- Templates
- Simplified English



Editors
Word processor

Editors and word processors

- Multiple editors: vi/vim,Visual Studio Code, Notepad, Pages, Emacs, MS Word, Google Docs, etc.
- Choose whatever is comfortable and suitable for the project.

VS Code preview

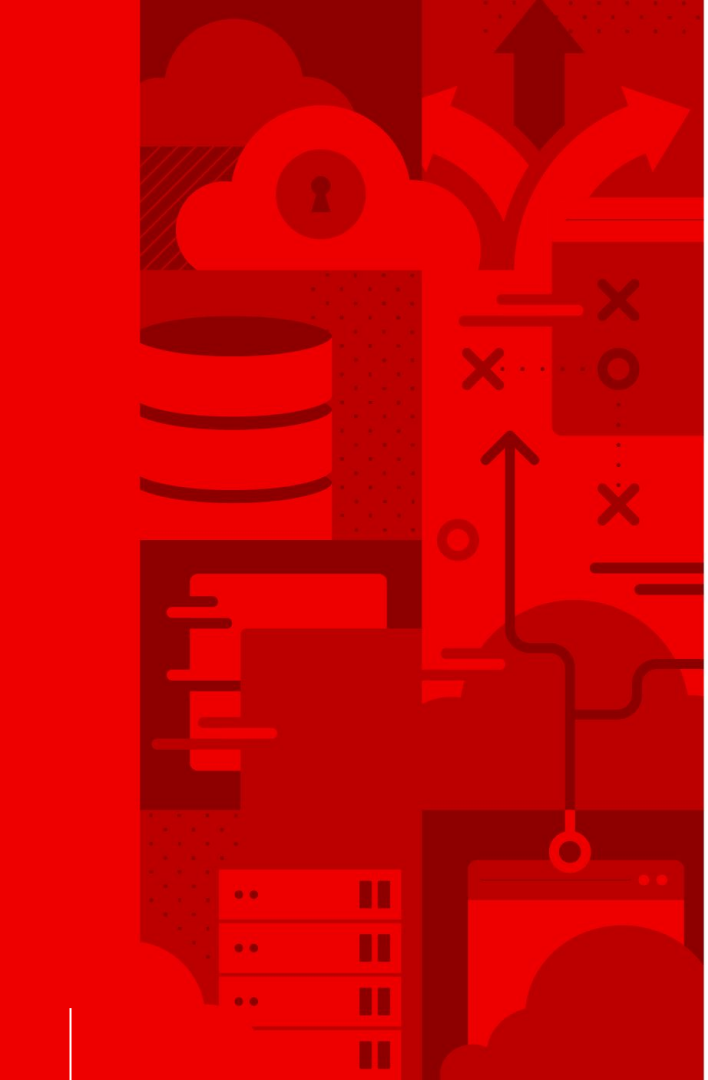
Vim editor

There are modes: **normal** (default), **insert** and **command line**.

Command	Description
vim FILE_NAME	Create or modify the FILE_NAME in vim.
:q or :ZQ	Quit the file without saving. Perform in **command line** mode.
:x or :qw!	Save and quit file. Perform in **command line** mode.
dd	Delete the highlighted text or the current line. Perform in **normal** mode.
v	Highlight the text. Use *left* and *right* arrows to expand the text area. Perform in **normal** mode.
y	Copy the highlighted text or the current line. Perform in **normal** mode.
p	Paste the highlighted text or the current line. Perform in **normal** mode.

Vim tutorial: *vimtutor*

Useful links: *vim.org*



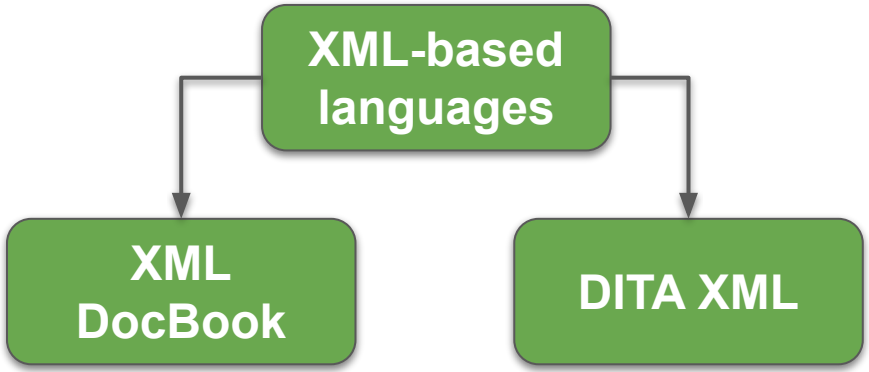
Languages

What is a markup language?

"A **markup language** is a text-encoding system which specifies the structure and formatting of a document and potentially the relationships among its parts."

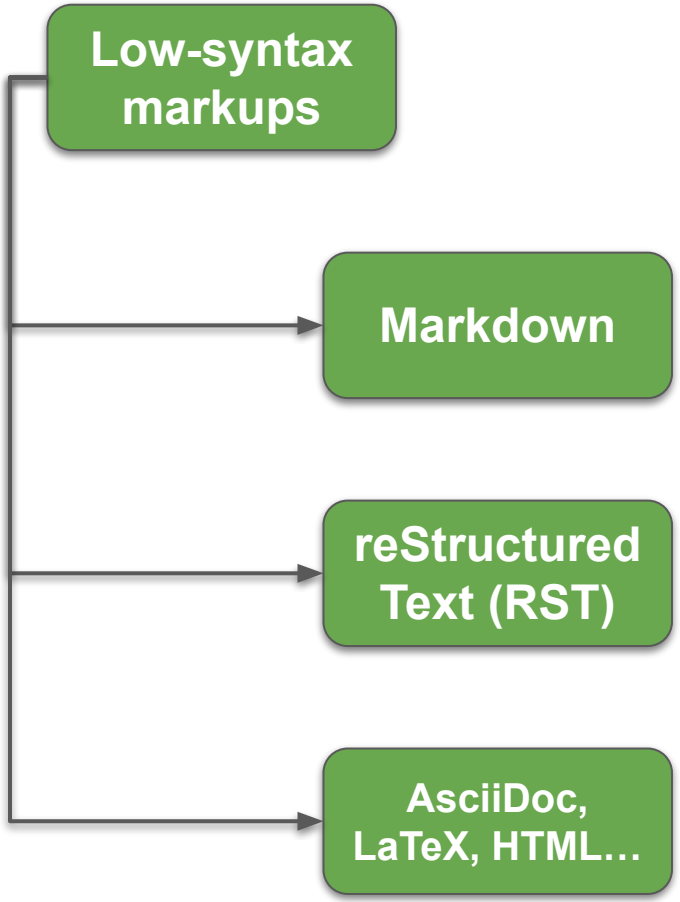
Wikipedia

https://en.wikipedia.org/wiki/Markup_language



- Robust Content Management Systems
- Multi-language support
- Single source
- Modular content management
- Commercial/corporate world

- Easy adoption by community members
- Single source
- Modular content management
- Natural connection with GitHub
- Popular in open source world



XML

Example

```
<section>

  <title>Extensible Markup
  Language</title>

  <para>Extensible Markup Language
  (XML) is a markup language that marks
  up data content with tags. XML
uses
  tags to define content
  structure.</para>

</section>
```

- Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data (Wikipedia)
- Define data with pair of tags
  - Human-readable
  - Machine-readable
- Each XML document needs to contain XML header (XML declaration): `<?xml version="1.0" encoding="UTF-8"?>`
- Nesting

DocBook

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook" version="5.0">
  <title>Very simple book</title>
  <chapter xml:id="chapter_1">
    <title>Chapter 1</title>
    <para>Hello world!</para>
    <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</para>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Chapter 2</title>
    <para>Hello again, world!</para>
  </chapter>
</book>
```

- Semantic markup language created for the purpose of technical writing
- XML uses tags to define content structure
- It consists of:
  - Elements - the basic building blocks
  - Attributes - information related to a specific element
  - Entities - representing an item of data instead of using the data itself
- Advantages: self-describing, widely used, extensibility
- Disadvantages: more lengthy, less readable compared to other markup languages, strict syntax rules
- Learn more:

  https://tdg.docbook.org/tdg/4.5/docbook.html

DocBook

- Single source, multiple outputs (HTML, PDF, CHM)
- Templates
- Tooling: Oxygen, xMetaL, etc.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook" version="5.0">
  <title>Very simple book</title>
  <chapter xml:id="chapter_1">
    <title>Chapter 1</title>
    <para>Hello world!</para>
    <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</para>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Chapter 2</title>
    <para>Hello again, world!</para>
  </chapter>
</book>
```

Darwin Information Typing Infrastructure (DITA)

- DITA is an XML-based architecture for creating modular content that allows you to reuse content and interchange content from various sources: topic-based authoring
- Advantages:
  - CMS support
  - WYSIWYG
  - Compounding guides based on versions
- DITA Open Toolkit – the open-source publishing engine: https://github.com/dita-ot/dita-ot/

Example

```
<topic xml:lang="en"
id="sample">
  <title>Sample
  title</title>
  <body>
  <p audience="teacher">
  This text is for the
  teacher.</p>
  <p audience="student">
  This text is for the
  student.</p>
  </body>
</topic>
```

Comparison of languages

Language	DocBook	DITA	Markdown	RST
Single Source	x	x	x	x
Modular content		x		x
Reusing content	x	x		x
CMS	x	x		x
Proprietary / open source	x	x	x	x

Markdown

Example

```
Heading
=======

# Alternative heading

Sub-heading
-----------

Block of text with _italic_,
**bold** and `monospace`
formatting. This is a
[link](http://example.com).

 1. numbered list
     * bulleted list
     * another bulleted list
  2. another list item

![Image](some-picture.png
"picture")
```

- *"Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML)."* John Gruber

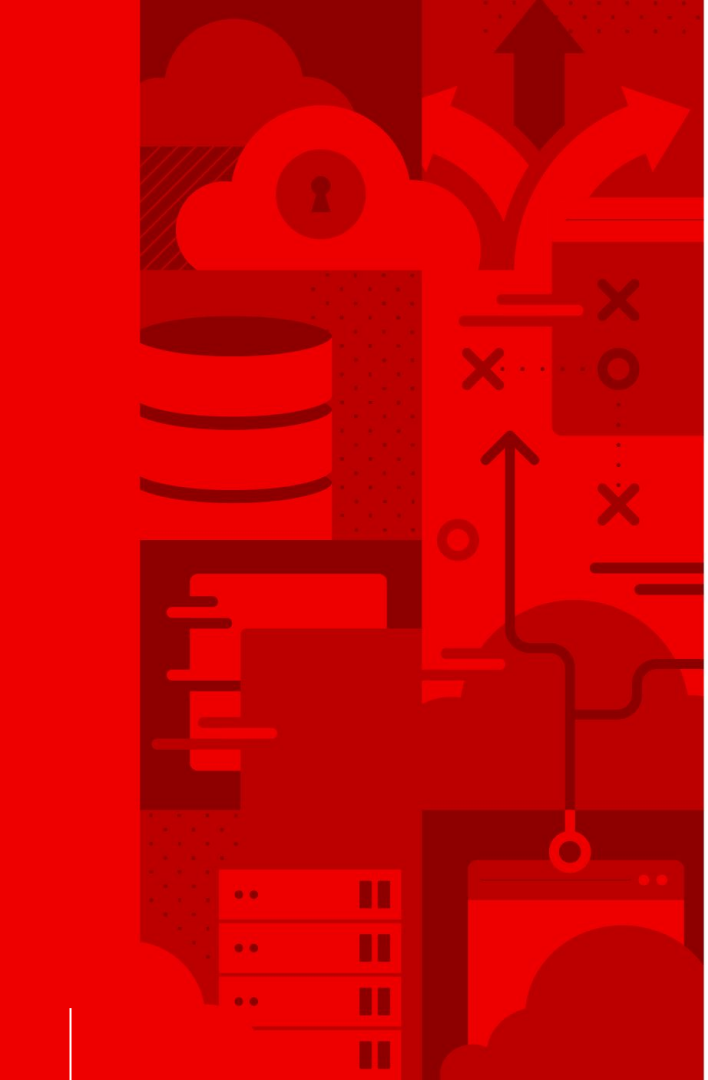
- Simplicity: plain text with Markdown syntax but very basic options (e.g. support for tables, modularity)
- Created in 2004, it became the first popular lightweight markup language, especially for blogging, online forums and hosting platforms like GitHub
- Many markup flavors, limited success in standardization
- Learn more: https://daringfireball.net/projects/markdown/

reStructuredText (RST) + Sphinx

Example

```
Chapter 1 Title
===============

.. toctree::
   :maxdepth: 2

   other_documents/included_document

This is a paragraph.

Section 1.1 Title
-----------------

   This is an indented block of text.

::

    This is a block of preformatted text.

* a bullet point

  - a sub-list item

* another bullet point

.. image:: image_folder/included_image.png
```

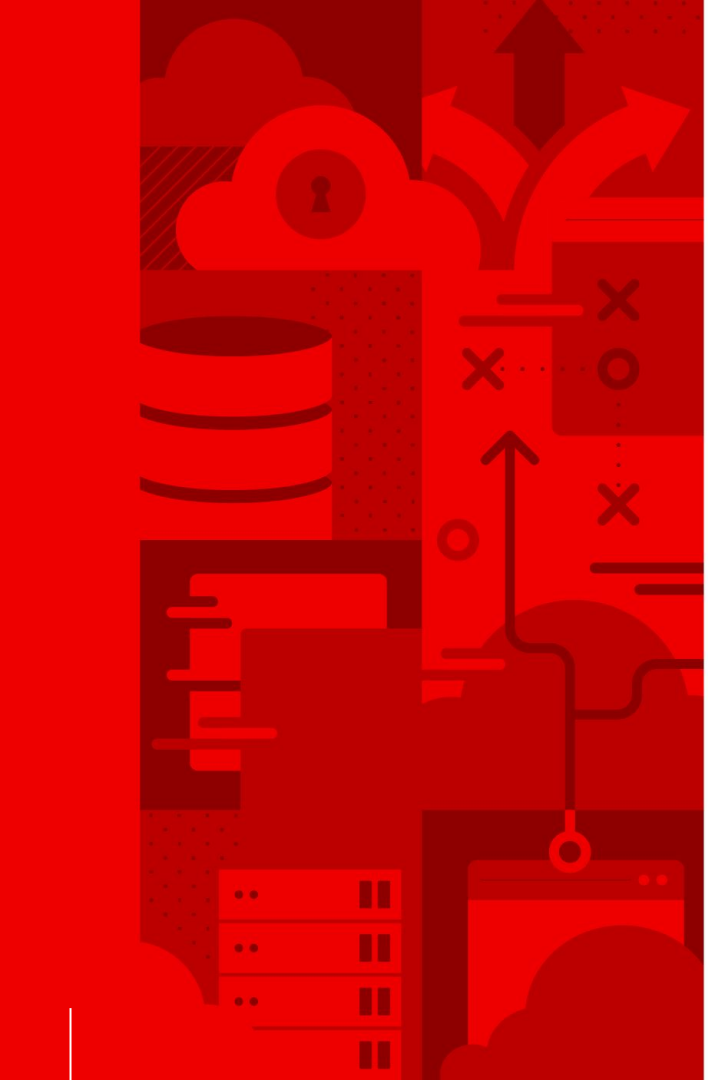
- reStructuredText (RST) is a lightweight markup language popular in Python-based communities
- Similar ly to Python, RST uses indentations
- Sphinx generates documentation from RST sources using the Docutils Python tools
- Sphinx supports cross-references and including documents in a hierarchy with automated linking, or generating indices
- Supports multiple output formats including HTML, PDF, man pages, etc.
- Can build whole documentation sites, including translations and custom themes
- Supported on GitHub and used by documentation hosting platforms such as https://readthedocs.org/



Collaboration tools

Collaboration

- Communication (Slack, IRC, Google Meet, Zoom)
- Whiteboards (Miro)
- Project and task management tools (Jira, Confluence)
- Version control (Github, Gitlab, Bitbucket)

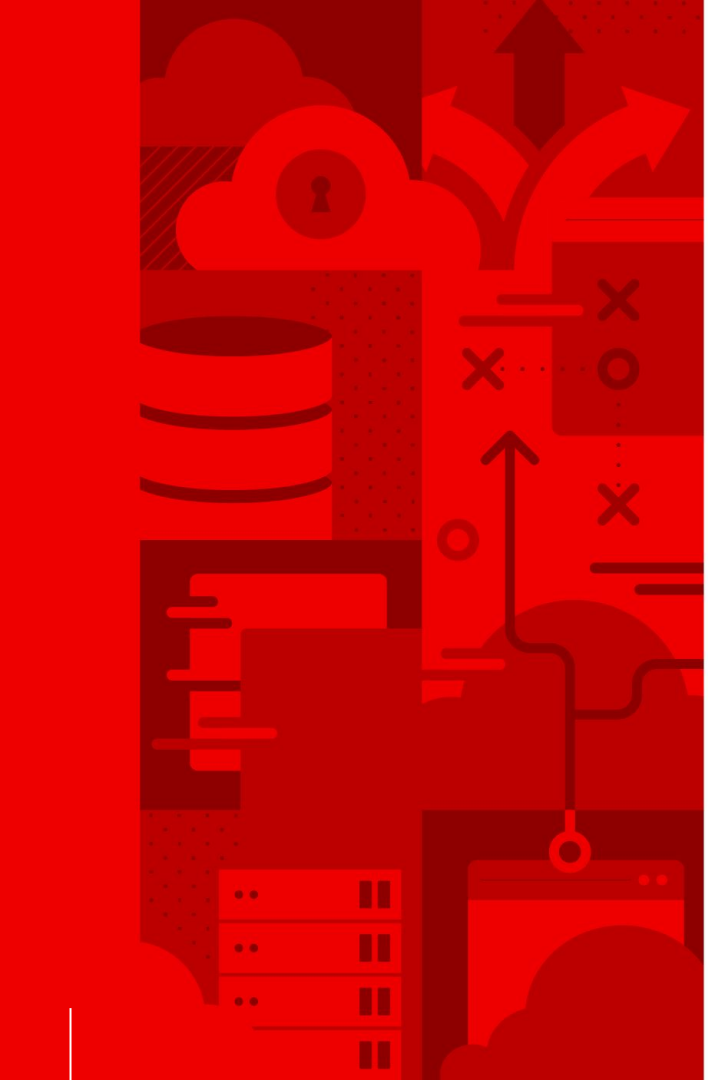


Version control

Version control

- Tracking and management of content changes
- Makes software development faster and more efficient
- Every modification is tracked in a special database
- Makes fixing issues easier
- The most widely used version control system is Git

https://www.atlassian.com/git/tutorials/what-is-version-control#:~:text=Version%20control%2C%20also%20known%20as%20source%20control%2C,manage%20changes%20to%20source%20code%20over%20time
https://www.atlassian.com/git/tutorials/what-is-git



Tools for graphics

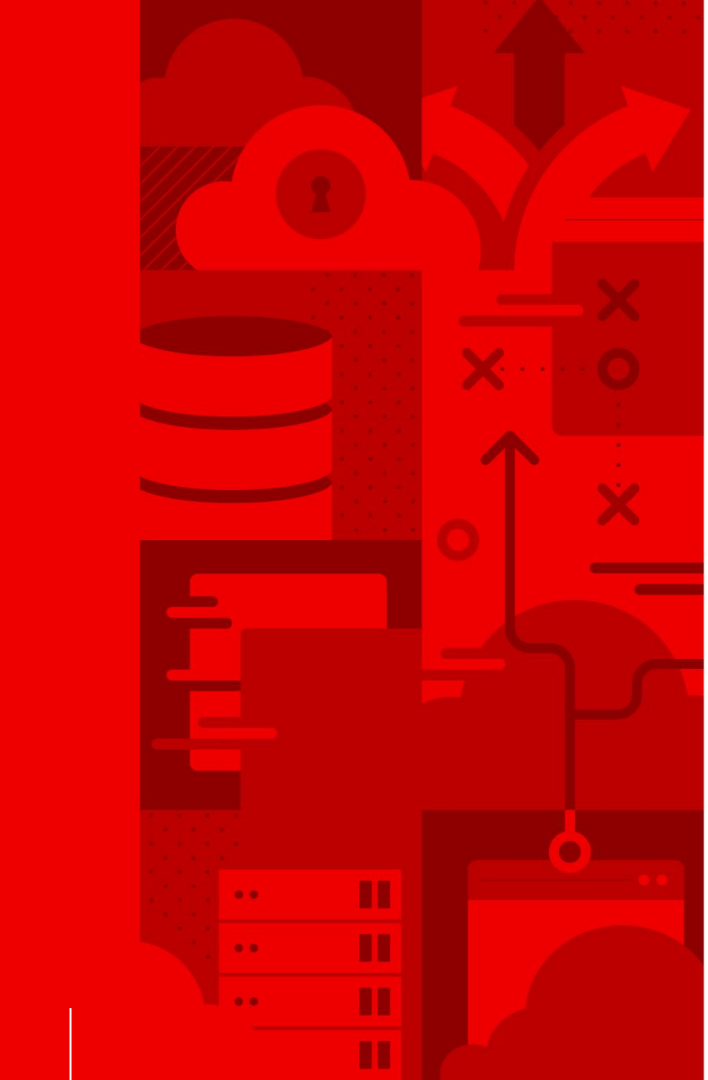
Graphics

- Screenshot capture
- Image manipulation (GIMP)
- Diagrams (Inkscape*, diagrams.net, mermaid.js)

* Inkscape is a vector graphics software with extended capabilities

Takeaways

- Tools help you do your work efficiently
- Choose tools based on your specific needs, project requirements, and personal preferences
- Most popular language in the open source is Markdown
- Use a text editor that supports syntax highlighting and has a preview



Exercise

Document a technology/feature

- https://dillinger.io/
- Write a short introduction using markdown
- Explain for who it is useful and describe benefits
- Document how to use the feature/application
- Include: heading, paragraph, list, picture, text style (bold, italic, etc)

Thank you

Prerequisites for the next class (14th March)

- Install AsciiDoctor on your computer (https://rh-writers.github.io/technical-writing-course-brno/#_introduction_to_asciidoctor)
- Sign up for a GitHub account.
- Set up Git and SSH keys on your computer if you work from the command line (https://rh-writers.github.io/technical-writing-course-brno/#_git_installation .
- Alternatively, use the GitHub web interface.