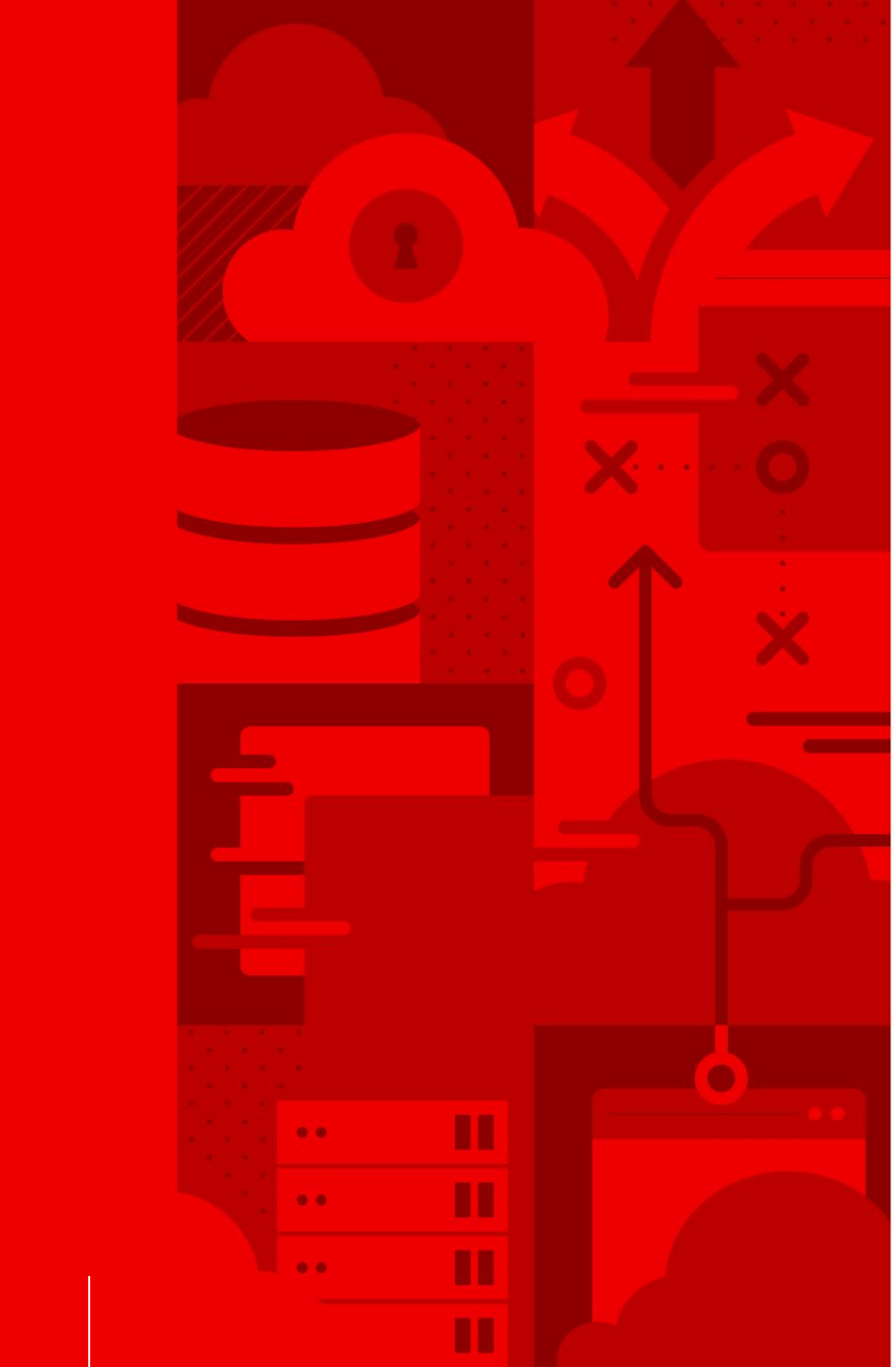
An Introduction to Tooling for Technical Documentation

Dominika Borges
Technical Writer

Eliska Romanova
Technical Writer

What we'll discuss today

- Introduction
- Languages
  - XML based vs low syntax markups
  - Markdown
- Editors
- Demo
- Exercises
- Takeaways
- Prerequisites for the next lesson



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

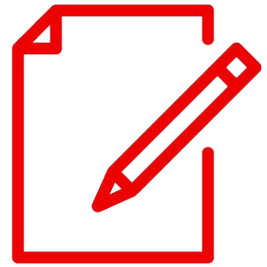
- Alignment with coding practices, versioning, and collaborative workflows

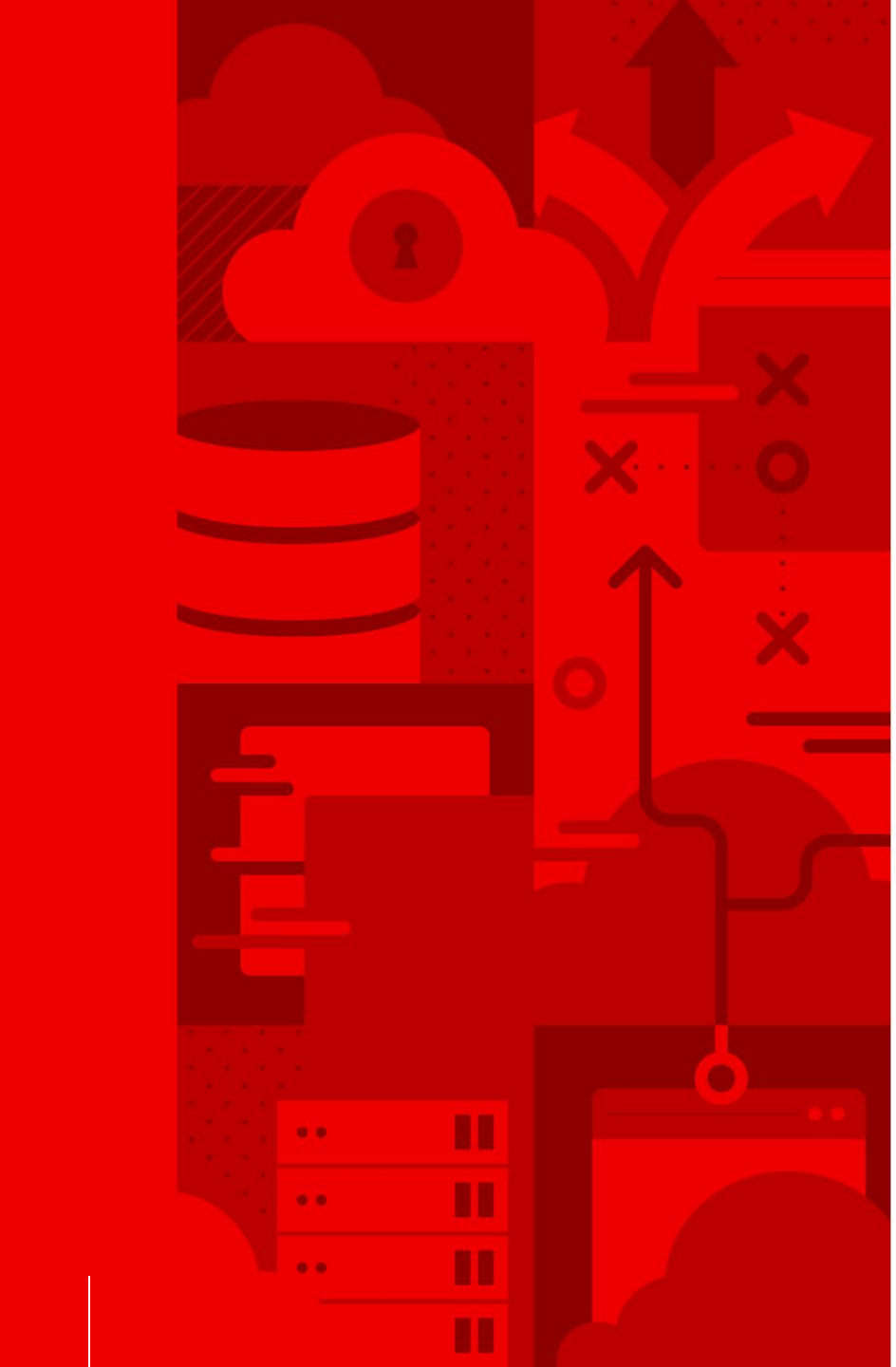
Automation and productivity

- Save time and reduce repetitive tasks

Focus on content

- Separation of content and formatting



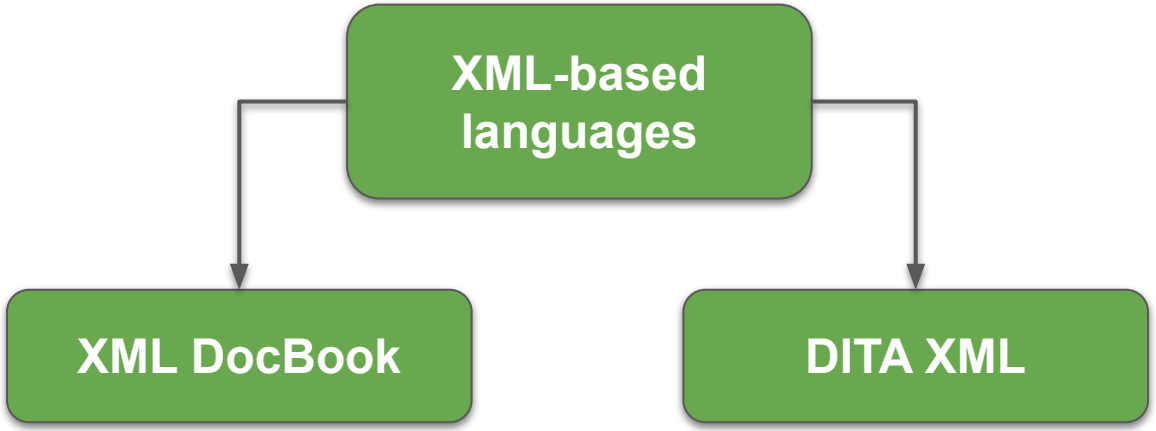


Languages

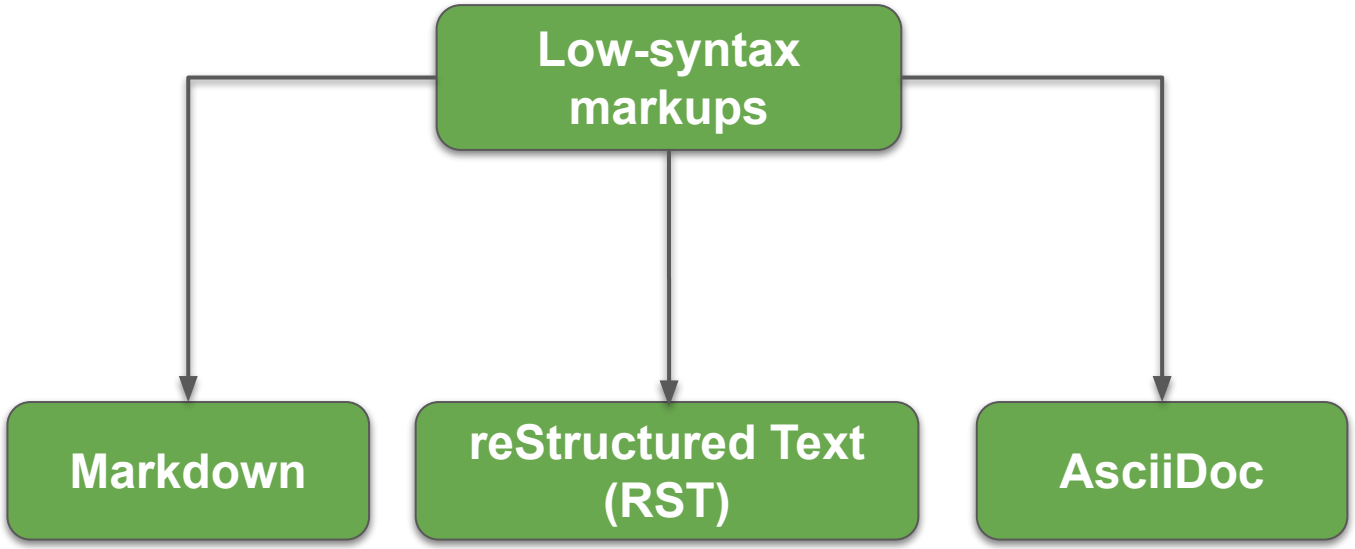
Markup languages

A tool to structure and format documents in a consistent and organized way

- separate content from formatting
- provide semantic meaning to content (accessibility)
- consistent output across different platforms
- unified structure and templating



- Robust content management systems
- Multi-language support
- Single source
- Modular content management
- Commercial/corporate world



- Simple and readable
- Single source
- Modular content management
- Natural connection with GitHub
- Popular in open source world

Markdown

```
### Description List

Term 1
: Definition 1

Term 2
: Definition 2

Term 3
: Definition 3
```

XML

```
<DescriptionList>
  <Term>
    <Name>Term 1</Name>
    <Definition>Definition 1</Definition>
  </Term>
  <Term>
    <Name>Term 2</Name>
    <Definition>Definition 2</Definition>
  </Term>
  <Term>
    <Name>Term 3</Name>
    <Definition>Definition 3</Definition>
  </Term>
</DescriptionList>
```

Xml-based vs low-syntax

```
<info>
<title>Creating {brandname} users</title>
<date>2024-02-20</date>
</info>
<simpara>Add credentials to authenticate with {brandname} Server
deployments through Hot Rod and REST endpoints.
Before you can access the {brandname} Console or perform cache
operations you must create at least one user with the {brandname}
command line interface (CLI).</simpara>
<tip>
<simpara>{brandname} enforces security authorization with
role-based access control (RBAC).
Create an <literal>admin</literal> user the first time you add
credentials to gain full <literal>ADMIN</literal> permissions to
your {brandname} deployment.</simpara>
</tip>
<itemizedlist>
<title>Prerequisites</title>
<listitem>
<simpara>Download and install {brandname} Server.</simpara>
</listitem>
</itemizedlist>
<orderedlist numeration="arabic">
<title>Procedure</title>
<listitem>
<simpara>Open a terminal in
<literal>{server_home}</literal>.</simpara>
</listitem>
<listitem>
<simpara>Create an <literal>admin</literal> user with the
<literal role="command">user create</literal> command.</simpara>
<programlisting language="sh"
linenumbering="unnumbered">include::cmd_examples/user_tool_nix.ad
oc[]</programlisting>
```

1.6. Creating Data Grid users

Add credentials to authenticate with Data Grid Server deployments through Hot Rod and REST endpoints. Before you can access the Data Grid Console or perform cache operations you must create at least one user with the Data Grid command line interface (CLI).

Tip

Data Grid enforces security authorization with role-based access control (RBAC). Create an `admin` user the first time you add credentials to gain full `ADMIN` permissions to your Data Grid deployment.

Prerequisites

- Download and install Data Grid Server.

Procedure

1. Open a terminal in `$RHDG_HOME`.
2. Create an `admin` user with the `user create` command.

```
bin/cli.sh user create admin -p changeme
```

Xml-based vs low-syntax

```
[id='creating-users_{context}']
= Creating {brandname} users

Add credentials to authenticate with {brandname} Server
deployments through Hot Rod and REST endpoints.
Before you can access the {brandname} Console or perform
cache operations you must create at least one user with the
{brandname} command line interface (CLI).

[TIP]
====
{brandname} enforces security authorization with role-based
access control (RBAC).
Create an `admin` user the first time you add credentials
to gain full `ADMIN` permissions to your {brandname}
deployment.
====

.Prerequisites

* Download and install {brandname} Server.

.Procedure

. Open a terminal in `{server_home}`.
. Create an `admin` user with the [command]`user create`
command.
+
[source,sh,options="nowrap",subs=attributes+]
----
include::cmd_examples/user_tool_nix.adoc[]
----
```

1.6. Creating Data Grid users

Add credentials to authenticate with Data Grid Server deployments through Hot Rod and REST endpoints. Before you can access the Data Grid Console or perform cache operations you must create at least one user with the Data Grid command line interface (CLI).

Tip

Data Grid enforces security authorization with role-based access control (RBAC). Create an `admin` user the first time you add credentials to gain full `ADMIN` permissions to your Data Grid deployment.

Prerequisites

- Download and install Data Grid Server.

Procedure

1. Open a terminal in `$RHDG_HOME`.
2. Create an `admin` user with the `user create` command.

```
bin/cli.sh user create admin -p changeme
```

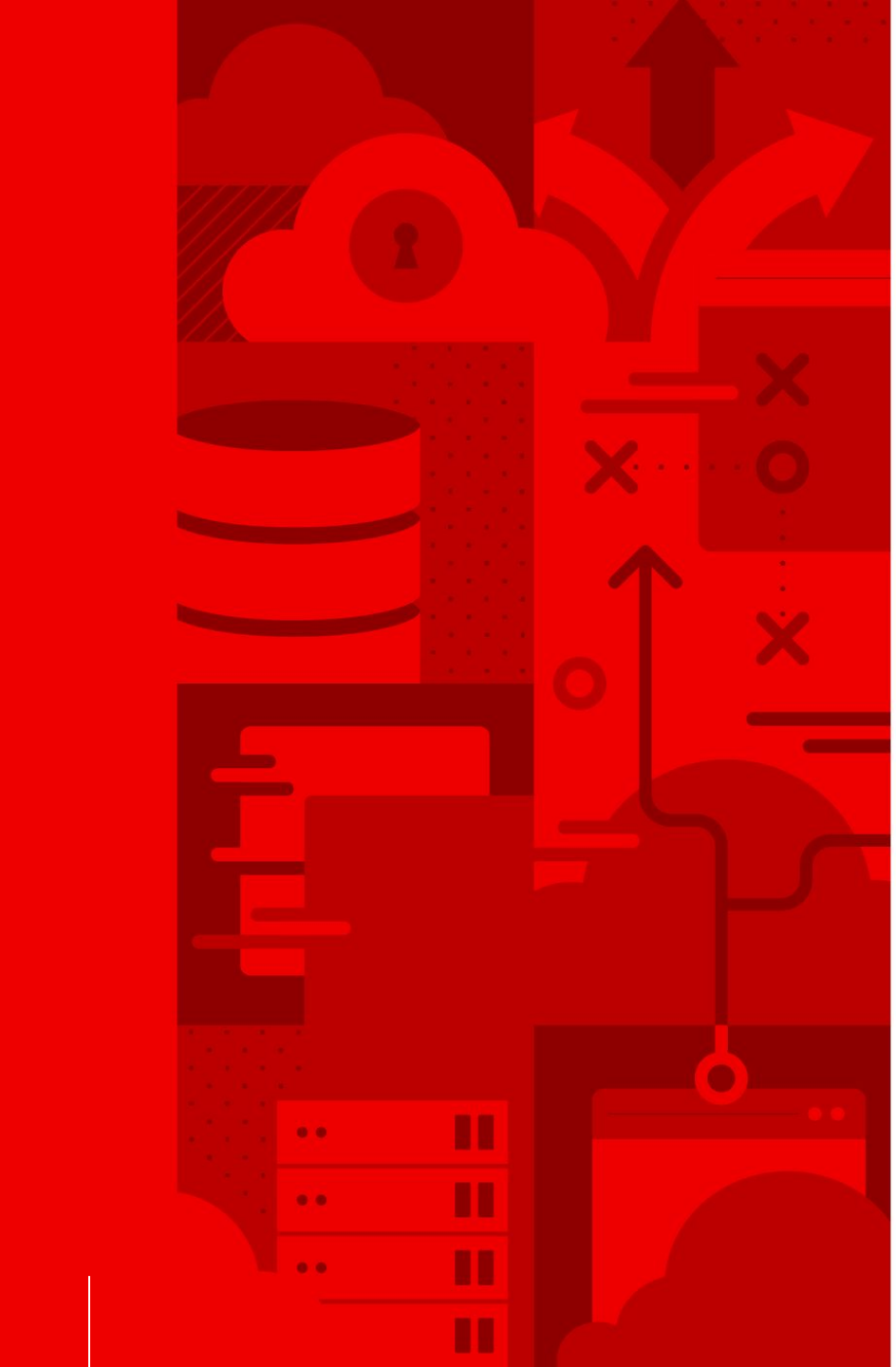
Markdown

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 docs. 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 collaboration platforms like GitHub.
- Many Markdown flavors, limited success in standardization
- Resource: https://daringfireball.net/projects/markdown/



Editors

Editors

• **Standalone:**
  ○ Vim
  ○ VSCode
  ○ Notepad++
  ○ Apple Pages
  ○ Emacs
  ○ Sublime Text
	• **Web-Based:**
	  ○ Stackedit
	  ○ Pandao
	  ○ Dillinger.IO
		• **Extensions:**
		  ○ Markdown Viewer
		  ○ Markdown Here

- Choose whatever is comfortable

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.

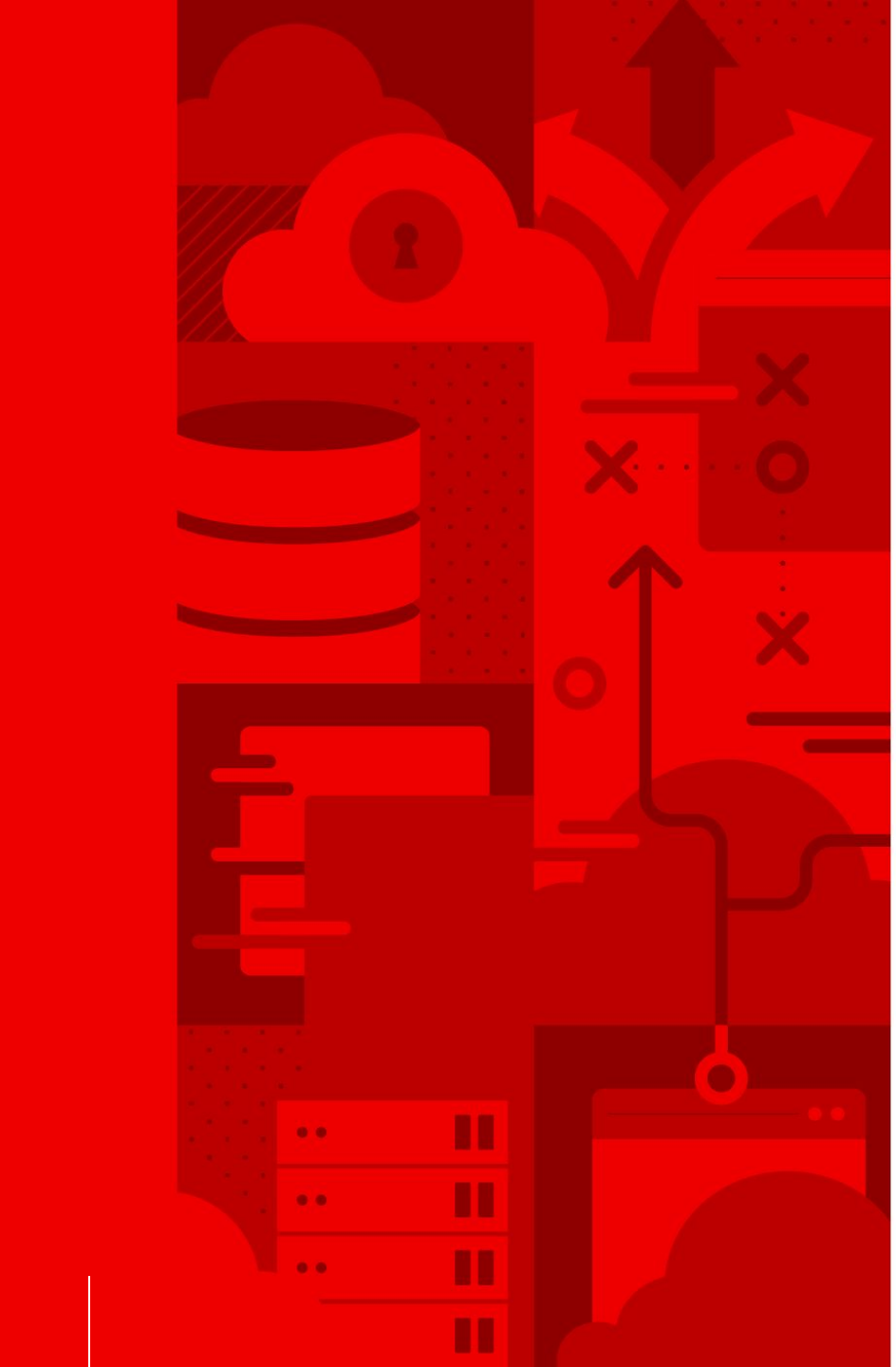
Vim tutorial: *vimtutor*

Useful links: *vim.org*

Demo: VSCode preview

Demo: Markdown syntax in VS Code

Demo: VSCode preview



Exercise

Exercise I

Getting Started with Markdown

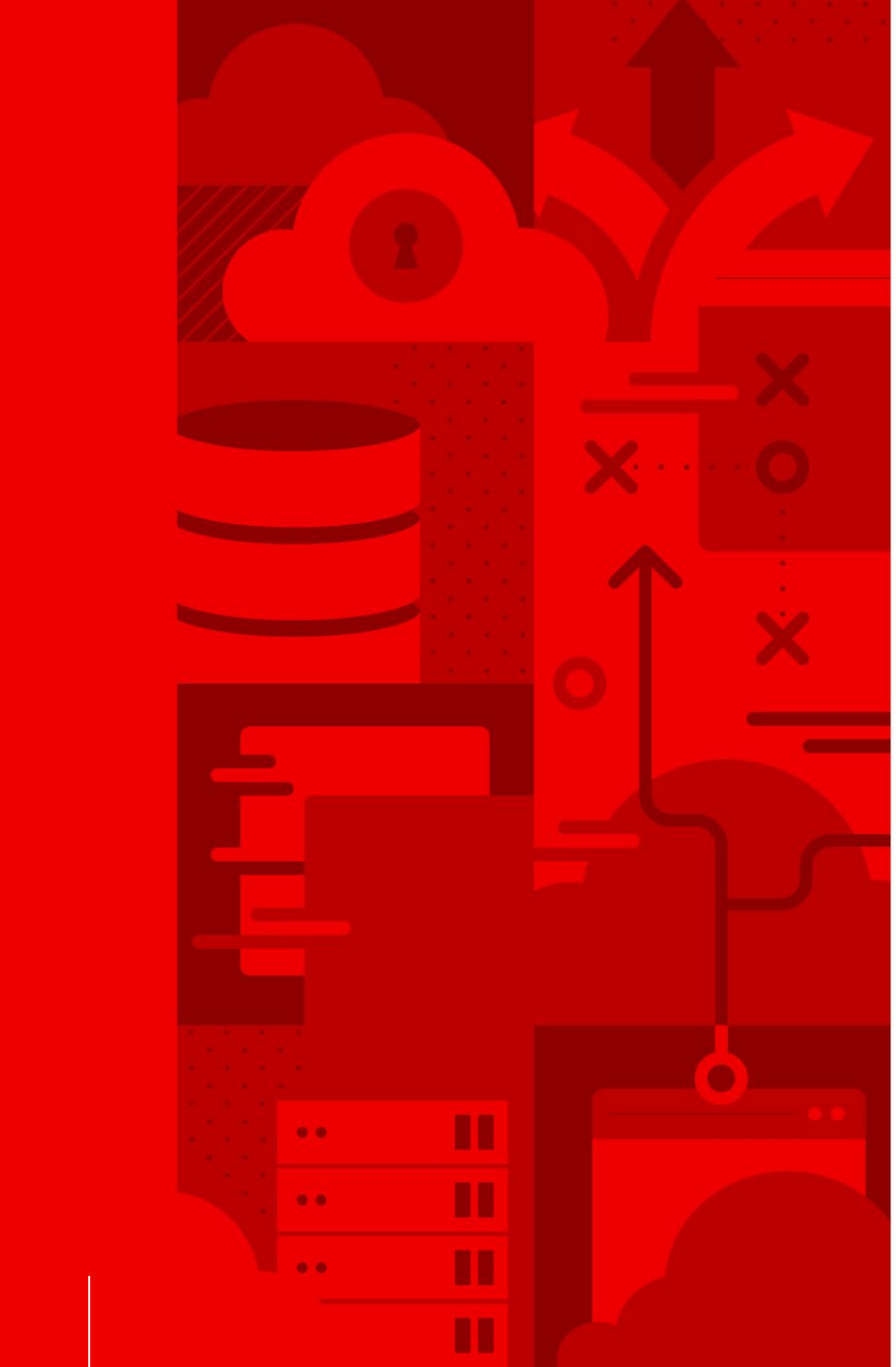
To start writing in **Markdown**, follow these steps:
0. Download and install VSCode
1. Open a new file in your text editor.
2. Save the file with a `.md` extension to indicate that it is written in Markdown format.
3. Write your text content using Markdown syntax, such as headings, lists, links, and code blocks.

TIP: If you are using VS Code, you can press `Ctrl+K V` (or the preview icon) to have a preview side-by-side with the file you are editing and see changes reflected in real-time as you edit.

Using emphasis

Emphasis	Usage
Bold	Interface controls and keywords
Italics	Variables and citations
`Monospace`	Code, code examples, and command names

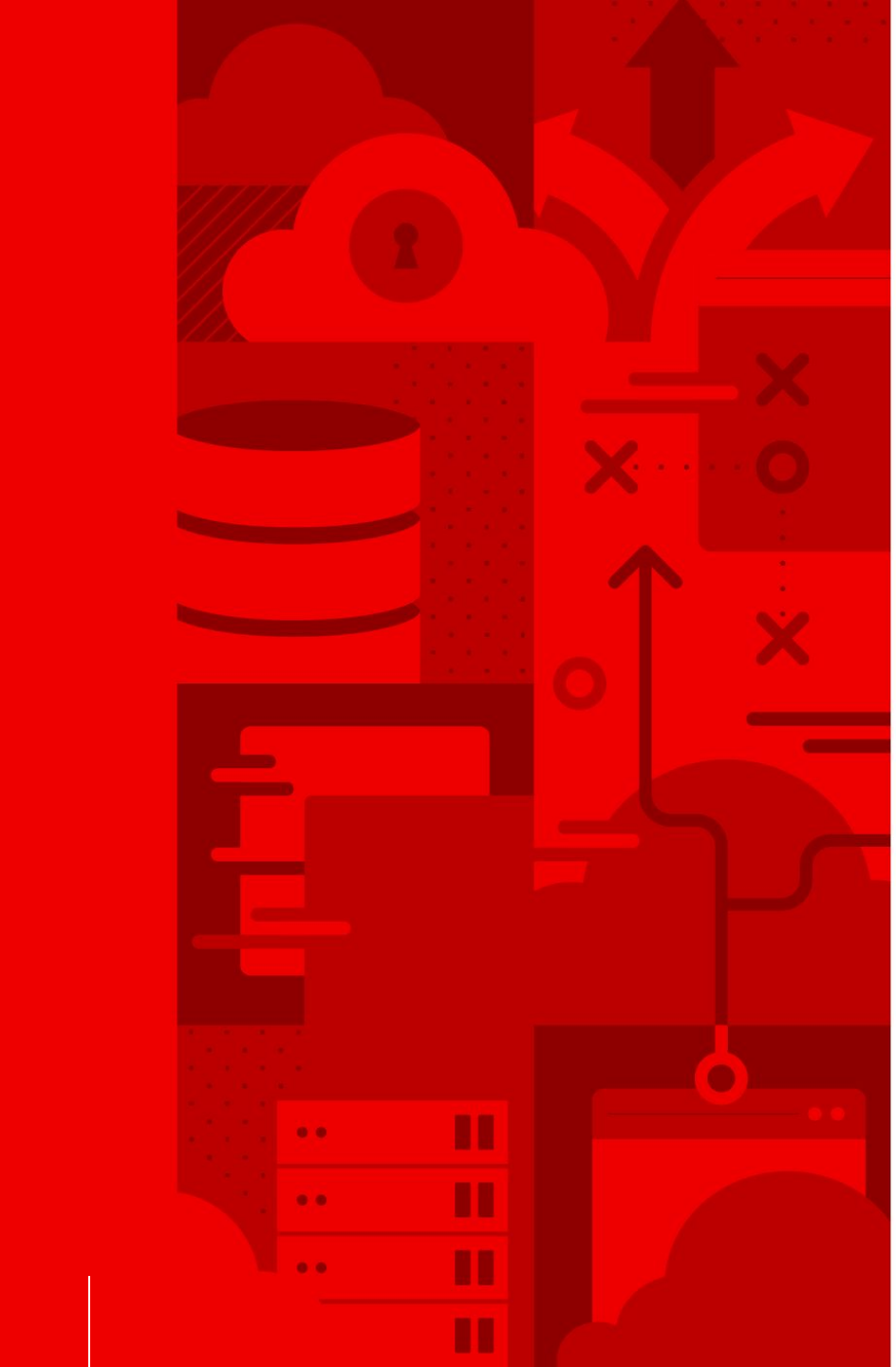
Need help?
markdownguide.org/basic-syntax/



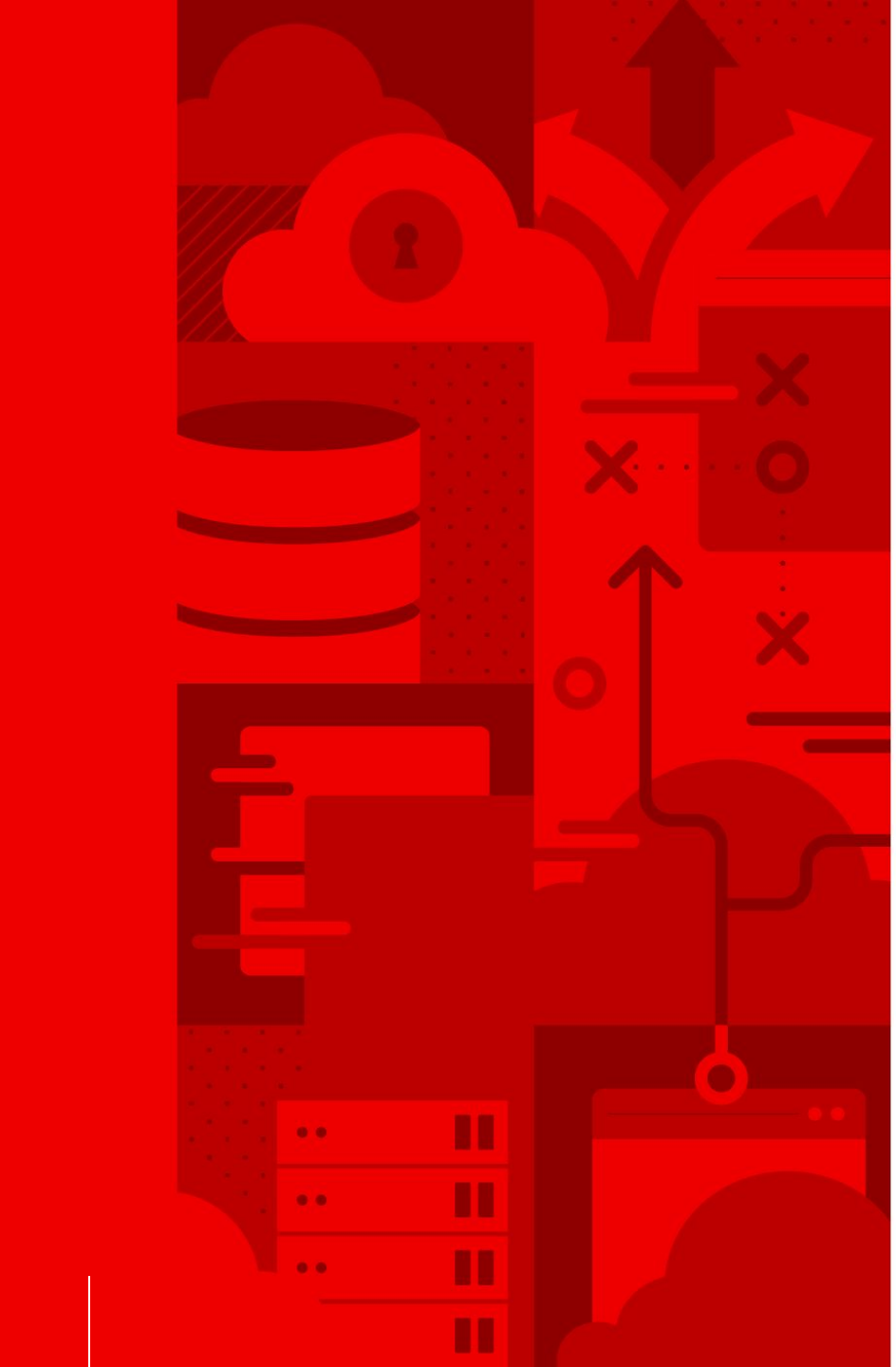
Exercise I review

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



Q&A



Post-lesson exercise

Time to practice Markdown

Rewrite and restructure the following text using **Markdown**

Exercise text

Duration **60 minutes**
Save the file as .md file, include the image file and send all the files to dvagnero@redhat.com by 12:15 PM

Need help?
markdownguide.org/basic-syntax/
developers.google.com/style/text-formatting

Prerequisites for the next lesson

- Create a Github account

Signing up for a new personal account

1. Navigate to https://github.com/.
2. Click **Sign up**.
3. Follow the prompts to create your personal account.

During sign up, you'll be asked to verify your email address. Without a verified email address, you won't be able to complete some basic GitHub tasks, such as creating a repository.

If you're having problems verifying your email address, there are some troubleshooting steps you can take. For more information, see "Verifying your email address."

Prerequisites for the next lesson

Install git

- If you are installing git on Windows, during the setup, just keep all the default configurations (you can change the destination location, of course)
- Verify that git works correctly:
  - Windows: Open *Git CMD*, Linux: Open terminal
  - Run `**git**` command.
  - You should get a list of git commands you can use:

```
eromanov@eromanov-thinkpadt14sgen1 ~ $ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]
```