2025 International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3315-7426-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICETA67772.2025.11280031

Process of sharing knowledge through private cloud documentation

Lubica Zidekova
Faculty of Management Science and Informatics
University of Zilina
Zilina, Slovakia
lubica.zidekova@fri.uniza.sk

Ivana Bridova
Faculty of Management Science and Informatics
University of Zilina
Zilina, Slovakia
ivana.bridova@fri.uniza.sk

Marek Moravcik
Faculty of Management Science and Informatics
University of Zilina
Zilina, Slovakia
marek.moravcik@fri.uniza.sk

Pavel Segec
Faculty of Management Science and Informatics
University of Zilina
Zilina, Slovakia
pavel.segec@fri.uniza.sk

***Abstract*—Documentation is a key prerequisite for sustainable management of projects and entire organizations – it unifies concepts, processes, and procedures and reduces operational risks. The article focuses on the selection and implementation of a tool for web documentation of the departmental private cloud at the Department of Information Networks of the University of Žilina in Žilina (KIS FRI UNIZA). In the introduction, we broadly distinguish between technical and user documentation and emphasize the benefits of the web form (accessibility, easy updating, multimedia, interactivity). Based on the defined multi-criteria requirements, Sphinx, MkDocs, and Docusaurus were compared. The analysis shows that MkDocs with the Material for MkDocs template, deployed in containers via Docker Compose, is the most suitable for the needs of the department. The solution brings a clear content structure in Markdown format, quick editing, support for copyable code blocks, and easy integration of student guides for OpenStack. We describe the basic project structure, key elements of the docker-compose.yaml configuration and a plan for future expansions. The result is a central, scalable, and practically maintainable knowledge repository that supports teaching, research, and private cloud operations.**

***Index Terms*—Cloud Computing, documentation, academic environment, knowledge sharing**

I. Introduction

Documentation is one of the key components of successful management of any project or management of the entire organization. It defines various important terms necessary for understanding the functioning of the entire system. Documentation provides written guidelines, information, instructions and processes important for the correct execution of individual project activities.

In the field of IT and software development, technical as well as user documentation plays a special role. Technical documentation assists in all steps of project management, from the installation and deployment of specialized software, through the development and maintenance of systems, as well as in the management of information and communication security risks.

It plays an important role in project-related communication, facilitating interaction between individual employees of the organization, and is a key element in ensuring the long-term sustainability of systems. User documentation often makes it easier for laypeople to better understand the software and technologies they are interested in using.

Documentation can take various forms, for example, it can be in the classic printed form, but also in electronic form in the form of a website. Both forms have their pros and cons. Some people need to have documentation in printed form because it gives them better clarity, to view the content and browse a specific chapter, rather than having to ”click through the web” and look for information in individual tabs.

In today’s digital world, web documentation is of particular importance. Its undisputed greatest advantage is its accessibility, as users only need access to the Internet and a digital device, which can be a computer, tablet or mobile phone. In addition to immediate access to the necessary information, it allows for easy editing or updating of content, ensuring that the documentation is always up-to-date and contains relevant information. Another advantage of electronic documentation over its printed version is the possibility of using multimedia elements, such as videos, GIF images or links to additional information available on another website. Higher interactivity can help to better visualize information, and thus to understand it more quickly.

Web documentation is used by large software application providers, including cloud service and infrastructure providers such as Amazon AWS or OpenStack, as well as smaller businesses, organizations, and various institutions.

Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:36:37 UTC from IEEE Xplore. Restrictions apply.

KIS FRI UNIZA, which has been offering cloud services to students for ten years, lacked a source of information about the functionalities and possibilities of using the cloud, which would be processed in the form of clear documentation. For this reason, a requirement arose to create web documentation where students, teachers, but also enthusiasts for cloud technologies would obtain the necessary information. For example, about how a private cloud was deployed for the academic environment using automation and virtualization tools, since the KIS FRI UNIZA department is the only one of its kind in Slovakia that has its own private cloud solution.

Students will find interesting information and instructions on how to create a virtual instance and other virtual network components via:

- GUI Horizon [1].
- **Terraform** [2], an open-source tool that enables declarative creation of cloud Infrastructure as a Code (IaaC).
- **YAML file** [3], a file format that is also involved in creating and defining configuration, data, and network structures.

The documentation will continue to evolve since its launch, and its content will also be updated to cover information requirements to the greatest extent possible.

II. Requirements for choosing a documentation creation tool

A summary of the requirements that were set for selecting a suitable tool for documentation creation were as follows:

- open-source tool,
- easy deployment, ideally using containerization tools,
- easy management and clear source code,
- prefer a tool whose use is not burdened by complex technologies such as JavaScript,
- active community and support for developers and users,
- creation of documentation without complex functions,
- ability to submit forms,
- in the future the possibility of adding administrator authentication using the Lightweight Directory Access Protocol (LDAP).

According to these preferences, the three most suitable software tools were selected for creating web documentation: Sphinx, MkDocs, and Docusaurus. The individual parameters for choosing a specific web documentation software are described in the table I. Each of these tools has its own advantages and each of them has a different suitability for different use cases. Sphinx is a good choice if you need a robust tool with lots of plugins and extensions, the problem is that you need to know the RST (reStructuredText) language.

Docusaurus is suitable for modern and aesthetically pleasing web documentation design, using Markdown as the language for creating documentation content, but the source code is extensive and more difficult to understand.

MkDocs is an effective tool for quickly creating clear documentation with clear and easy-to-create source code, just like Docusaurus, MkDocs uses the Markdown language to create documentation content.

TABLE I: Comparison of open-source tools for creating web documentation.

Properties	Sphinx	MkDocs	Docusaurus
Software Proprietary	Open-source	Open-source	Open-source
Language and syntax	reStructuredText (RST), similar to Markdown	Markdown language	Markdown language
Extensibility and Plugins	Wide range of plugins and extensions	Custom Python plugins possible	Custom JS plugins and extensions
Appearance and Design	Simple with customizable templates	Default themes with customization options	Modern and aesthetically pleasing design
Community and Support	Large and active community, available documentation	Active community with sufficient support	Growing community and support
User-Friendliness	Advanced tool, requires knowledge of RST	Simple tool, easy to use	Easy to use with modern tools
Deployment via Containerization	Possible, requires Docker configuration	Possible, easy to implement	Possible, more complex to implement
Ease of Deployment	Depends on the environment, not always easy	Simple, fast deployment across multiple platforms	Simple, ready for quick deployment
Creating Documentation Without JS and HTML Knowledge	Requires knowledge of RST and configuration	Relatively simple, based only on Markdown	Documentation based on Markdown, but many files are in JS or HTML
Code Readability	Well-organized but highly complex code	Simple code, suitable for smaller projects	Well-structured and intuitive code, but overly complex for small projects
Possibility to Create/Link Google Forms	Possible, but requires extensions or customization	Possible using plugins or external tools	Possible using plugins or external tools
Extension - LDAP Authentication	Possible, but requires extensions or customization	Possible using plugins or external tools	Possible using plugins or external tools

Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:36:37 UTC from IEEE Xplore. Restrictions apply.

Of the selected documentation creation tools, the MkDocs software was selected, the reasons for its selection are as follows:

- Easy deployment using containerization.
- It offers a clear structure from the start, which each user can customize.
- Unless you are interested in creating your own plugins, no knowledge of programming or scripting languages is required.
- MkDocs is a frequently used tool for creating web documentation, and therefore has a wide community of users who provide sufficient support in case of problems.

A. Practice of other institutions and universities in using web documentation

Experience from academic institutions suggests three predominant approaches to creating web documentation.

1) Statically generated pages with Markdown content are used primarily for how-to materials focused on quick introductions and repeatable procedures.
2) Tools primarily intended for extensive technical references are used where it is necessary to systematically generate and maintain a large amount of reference content.
3) Portal solutions combine documentation with blog posts, multilingual support, and versioning.

A representative example of the first approach is the documentation portal of the University of Florida Research Computing Center (HiPerGator), which uses MkDocs with the Material template. The structure is oriented towards concise ”quick starts” and thematically structured tutorials, which serve as a single point of departure for user procedures and training materials [4, 5]. A comparable model is also used by the Research IT department at the University of California, Berkeley: statically generated documentation with MkDocs/Material is managed in a repository with the possibility of proposing changes and publishing them after expert review, which supports ongoing maintenance and involvement of the author community [6, 7].

For spaces with a high proportion of reference content, Sphinx is a common choice in the academic and open community. OpenStack ecosystem projects maintain their own theme and extensions that unify the look and structure across subprojects; this approach is suitable where stable production and versioning of large references is required [8, 9]. An example of a third approach is portal solutions such as Docusaurus, which combine documentation with multilingualism, versioning, and accompanying articles; this model is particularly applicable to long-term projects with clearly separated versions of materials [10, 11].

The above findings provide a framework for tool selection in the next section: how-to materials and the need for rapid content updates favor statically generated Markdown pages (e.g. MkDocs/Material), while extensive reference sections can benefit from Sphinx; portal solutions make sense when requiring multilingualism and formal versioning of entire document sets.

III. MkDocs

MkDocs is a simple, fast, sophisticated and easily customizable tool for generating web documentation in the form of static web pages [12]. The entire content of the web documentation, and therefore all source files, are written in Markdown, which is a simple tool for converting text into HTML (Hyper Text Markup Language), designed for creating web pages [13]. For easier installation and a more beautiful appearance of the documentation, the Material for MkDocs template [14] was chosen, which is inspired by the Material Design design language developed by Google, which focuses on the simplicity and cleanliness of web pages.

The Material for MkDocs template provides a modern and aesthetically pleasing design for project documentation. It features interactive elements such as drop-down sections, sidebars, refined navigation elements, and Material Design-compliant color schemes. This template also supports various customization options for appearance and functionality to reflect the specific needs of the project.

A. Installing MkDocs using Docker

One of the requirements for installing the web documentation software was the ability to run this software using containerization and automation. For this reason, the containerization tools Docker and Docker compose were installed on the deployed server. The difference between them is that docker starts the container with the run command on the command line, in which individual parameters such as port definitions are also defined. Docker compose starts the container with the docker-compose up command, with the configuration data being loaded from the YAML file docker-compose.yaml.

An indisputable advantage is that the Material for MkDocs template itself has several Docker projects created by fan-developers, which can be downloaded from the official DockerHub repository [15], [16]. There are also several suggestions for instructions on how to install MkDocs on a server written on the Internet. In our case, the installation was carried out according to the instructions [17], which uses only the docker-compose.yaml file for installation. This file is clear and you can easily make changes in it, which are immediately applied after saving and restarting the container.

1) Description of configuration and configuration file: At the beginning, it is necessary to create a new folder in which all files belonging to the documentation will be concentrated. This folder can be named arbitrarily, in our case it was named cloud_mkdocs, because it covers the cloud documentation. The entire documentation schema is stored in the main folder. It consists of the newly created folder (/)docs, which stores the markdown files that form the content of the web documentation and the mkdocs.yaml file, which is responsible for the structure and appearance of the entire documentation. The main folder also contains the docker-compose.yaml file, which runs the MkDocs application in a container.

979-8-3315-7426-0/25/$31.00 ©2025 IEEE University of Zilina. Downloaded on December 22,2025 at 10:36:37 UTC from IEEE Xplore. Restrictions apply.

2) Structure and description of the docker-compose file: The docker-compose file has a predefined structure that can be applied to most applications with some modifications. The file content structure for cloud documentation is defined as follows [17]:

- version -– defines the version of the docker compose file format,
- services — defines the service (container) for the application. In our case, there is one service called mkdocs,
- mkdocs -– the name of the service that we defined in the services section,
  - image — this parameter determines which image is used to create the container for this service,
  - ports -– maps port 80 on the physical machine to port 8000 in the container,
  - volumes -– appends the current directory (.) to the directory (/docs) in the container. This allows MkDocs to access the documentation and configuration files.
  - stdin_open: true and tty: true -– these two parameters enable interactivity when loading the documentation after saving changes, i.e. any change will take effect immediately after saving this change in the IDE development environment.

B. Documentation content

The content of the documentation will reflect current information requirements regarding the use and operation of the departmental cloud. It will mainly include instructions for students on how to work with OpenStack technology. It will also serve to share resources and information from the results of project teaching, as well as bachelor's and diploma theses, which brought new deployments of technologies in OpenStack to support increased productivity and availability of the cloud. The documentation will also include an Evaluation Form for evaluating the service by its users, which was created in 2023 as part of the diploma thesis Systemic approach to cloud service development. The Terms of Use and Lease of the Departmental Cloud will be available, by accepting which the lessor assumes full responsibility for the computing resources that will be allocated to him.

1) Tutorials for students: Cloud infrastructure services are also used by students who are not primarily involved in cloud technologies during their studies, or those who are just learning to work with this technology. For this reason, a User Guide for managing basic OpenStack administration and creating networks and instances was created by students of the Applied Network Engineering (ASI) study program.

This user guide serves to facilitate the handling of OpenStack and its GUI. Until now, it has been processed in text form and was available through the Microsoft Teams (MS Teams) communication tool. However, a common problem with this type of data sharing is that it is often difficult to find the document you are looking for among the large number of shared documents.

For this reason, all available manuals and tutorials intended for students to work with OpenStack will be moved to the web documentation. It is assumed that this documentation will become part of the project teaching, to constantly reflect the current state, and to be managed in such a way that it is clear and will be a quality source of information for students during the following academic years.

2) Description of how to deploy and operate a private cloud solution on KIS: When creating the content of the documentation, the need to maintain all information related to the method of deploying and improving the operation of the departmental cloud in one place was also taken into account, so that students, as well as teachers, have a better overview of the activities performed in the cloud, their results, and the possibilities of building on this knowledge when writing new topics for final theses and choosing further scientific direction and progress in CC research.

Given the fact that KIS FRI UNIZA is the only department in Slovakia that operates its own private cloud solution, the web documentation and instructions on how to deploy, operate and troubleshoot a private cloud written in it will serve as inspiration and a guide to deploying a private cloud solution for cloud technology enthusiasts, whether for academic or any other purposes. One of the advantages of MkDocs is the ability to add copyable code blocks to the documentation, so the user does not have to rewrite individual lines of code manually or mark a long block of code for copying. The content of the entire block is copied to the clipboard after pressing the copy button.

3) Evaluation form and other questionnaires: Improving the provision of cloud services at KIS FRI UNIZA cannot be done without the cooperation of the departmental cloud operators and its users, who would provide feedback. Feedback and evaluation of the quality of cloud service provision will undoubtedly contribute to improving the cloud operations at KIS. Any constructive criticism that will help provide better quality services and facilitate teaching will be fully welcomed, however, it is necessary to take into account when evaluating the form that KIS provides a non-guaranteed service, since its guarantee is not yet possible for both personnel and technical reasons. The form's home page is shown inthe figure 1.

4) Terms of use for the departmental cloud: Providing cloud infrastructure as a service at KIS FRI UNIZA is providing an unguaranteed service in terms of the inability to guarantee a high level of its availability and the inability to maintain a certain level of quality of this service. These guarantees cannot be provided for both personnel and technical reasons.

Despite these facts, a requirement arose to create a set of user conditions, or rather business conditions, through which, under the pre-determined conditions of the provider, which is the management of KIS FRI UNIZA, the allocation of computing resources of the departmental cloud to lessors according to their needs will be ensured.

The terms of use are a form of binding agreement, by accepting which the user expresses his consent to their content

979-8-3315-7426-0/25/$31.00 ©2025 IEEE
Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:36:37 UTC from IEEE Xplore. Restrictions apply.

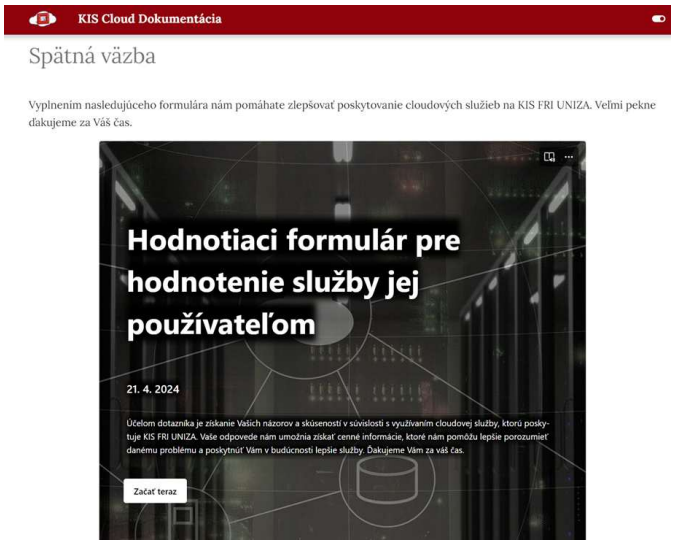


Fig. 1: Cloud service evaluation form on KIS

and, after accepting them, is obliged to comply with them throughout the entire period of use of the service.

The main reason for introducing these conditions into the operational cycle of providing departmental cloud services is to establish the rights and obligations, as well as to protect the rights and interests of both interested parties, both the lessor and the provider. Pre-established conditions of use of services can lead to accelerated resolution of disputes, or even their complete prevention. In the case of digital services that serve to process and store data, it is important to introduce a clause into the terms and conditions regarding the method of protection and processing of this data.

Terms of use are undoubtedly an important legal tool that provides assurance of maintaining accountability and transparency in both the provision and use of various types of digital services and products.

The developed user conditions for the use of departmental cloud services are part of the diploma thesis developed at KIS FRI UNIZA . It is also assumed that their content will be part of the ManageIQ tool, which is an important element in the deployment of OpenStack, and through which computing resources will be requested according to the specific needs of the lessor while adhering to these written conditions.

Until the cloud services are fully leased using the ManageIQ tool, which is currently under development, the wording of these terms and conditions will be published in the web documentation in the form of an MS Office or MS Forms form. Through this form, it will be possible to order computing resources with specific requirements from the ”comfort of home”, or online and from anywhere, i.e., without the need to physically go to the administrators and ask them for this action. The form can be filled out directly on the documentation page without needing redirection.

IV. Discussion

The observations suggest that deploying web documentation based on MkDocs and using containerization can reduce barriers to content management of cloud documentation and shorten the time required for editing and publishing. In the departmental private cloud environment, this approach proved to be particularly suitable for instructional-oriented materials, which corresponds to the practice of academic workplaces that prefer statically generated pages for quick deployment. The differences compared to works using Sphinx mainly concern the scope of reference content; while Sphinx is advantageous for large API documentation, in our context, simplicity and low demands on developer skills prevailed.

This leads to three recommendations for practice:

- maintain a tutorial structure with copyable code blocks,
- integrate short feedback with satisfaction when using cloud services
- formalize the contribution process – and involve students in it

The expected benefit is a reduction in students' training in working with the cloud environment and a decrease in repeated inquiries in support channels or during personal consultations outside of class hours.

Currently, all information published on the website http://docs.cc.uniza.sk is fully accessible to the public, however, there is an assumption and plan to add a custom extension to the documentation that would allow administrator login to the documentation using LDAP authentication. After administrator authentication, special content would be displayed, such as the departmental cloud address plan, internal documents, among which we can include security documentation, which will be supplemented with policies and guidelines for ensuring information and cyber security.

While network courses are also provided for foreign students studying at KIS FRI UNIZA on the basis of ERASMUS+ student mobility, efforts will be made to provide the option of choosing the language in which the documentation will be displayed.

V. Conclusion

The presented paper shows that consolidating knowledge about the private cloud into web documentation significantly increases process transparency, facilitates collaboration and reduces the time needed to train users. A comparison of open-source tools (Sphinx, MkDocs, Docusaurus) confirmed that MkDocs best meets the defined criteria: it is easily deployed in containers, contains readable source code, does not require advanced JavaScript knowledge and has a strong community. The chosen Material for MkDocs template brings modern design and user-friendliness without complex maintenance. Implementation via Docker Compose simplifies versioning, updates and portability between environments.

By centralizing instructions and adding copyable code blocks, the documentation becomes a practical working tool for students and educators. The integration of forms (service

979-8-3315-7426-0/25/$31.00 ©2025 IEEE 383
Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:36:37 UTC from IEEE Xplore. Restrictions apply.

evaluation, terms of use and rental of computing resources) improves feedback and formalizes the capacity allocation process. Planned extensions, in particular LDAP authentication for internal materials and multilingual support, will further strengthen the security, openness and usability of the solution.

The result is a living documentation ecosystem that meets the specifics of KIS FRI UNIZA as a unique academic workplace with its own private cloud in Slovakia. The solution is sustainable, scalable, and ready for continuous development of content and functionalities in the coming academic years.

ACKNOWLEDGMENT

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00010.

REFERENCES

[1] *Horizon: The OpenStack Dashboard Project*. 2024. URL: https://docs.openstack.org/horizon/latest/ (visited on 04/21/2024).

[2] *Terraform by HashiCorp*. 2024. URL: https://www.terraform.io/ (visited on 04/21/2024).

[3] *The Official YAML Web Site*. 2024. URL: https://yaml.org/ (visited on 04/21/2024).

[4] *UFIT Research Computing User Documentation (HiPerGator)*. 2025. URL: https://docs.rc.ufl.edu/ (visited on 09/24/2025).

[5] *Introduction to UFIT Research Computing*. 2025. URL: https://docs.rc.ufl.edu/quickstart/ (visited on 09/24/2025).

[6] *UC Berkeley Research IT Documentation*. 2025. URL: https://docs-research-it.berkeley.edu/ (visited on 09/24/2025).

[7] *Research IT Docs — Community Contributions Guide*. 2025. URL: https://docs-research-it.berkeley.edu/contributing (visited on 09/24/2025).

[8] *OpenStack Sphinx themes*. 2017. URL: https://docs.openstack.org/openstackdocstheme/latest/ (visited on 09/24/2025).

[9] *OpenStack Sphinx Extensions*. 2018. URL: https://docs.openstack.org/oslosphinx/pike/ (visited on 09/24/2025).

[10] *Docusaurus — Documentation*. 2025. URL: https://docusaurus.io/docs (visited on 09/24/2025).

[11] *Docusaurus — Site Showcase*. 2025. URL: https://docusaurus.io/showcase (visited on 09/24/2025).

[12] *MkDocs*. 2024. URL: https://www.mkdocs.org/ (visited on 04/21/2024).

[13] *Introduction to HTML*. 2024. URL: https://www.w3schools.com/html/html_intro.asp (visited on 04/21/2024).

[14] *Material for MkDocs*. 2024. URL: https://squidfunk.github.io/mkdocs-material/ (visited on 04/21/2024).

[15] *squidfunk/mkdocs-material*. 2024. URL: https://hub.docker.com/r/squidfunk/mkdocs-material (visited on 04/14/2024).

[16] *photoprism/mkdocs-material*. 2024. URL: https://hub.docker.com/r/photoprism/mkdocs-material (visited on 04/14/2024).

[17] *Setting Up MkDocs with Docker Compose - Techdox Docs*. 2024. URL: https://docs.techdox.nz/mkdocs/ (visited on 04/14/2024).

979-8-3315-7426-0/25/$31.00 ©2025 IEEE
Authorized licensed use limited to: University of Zilina. Downloaded on December 22,2025 at 10:36:37 UTC from IEEE Xplore. Restrictions apply.