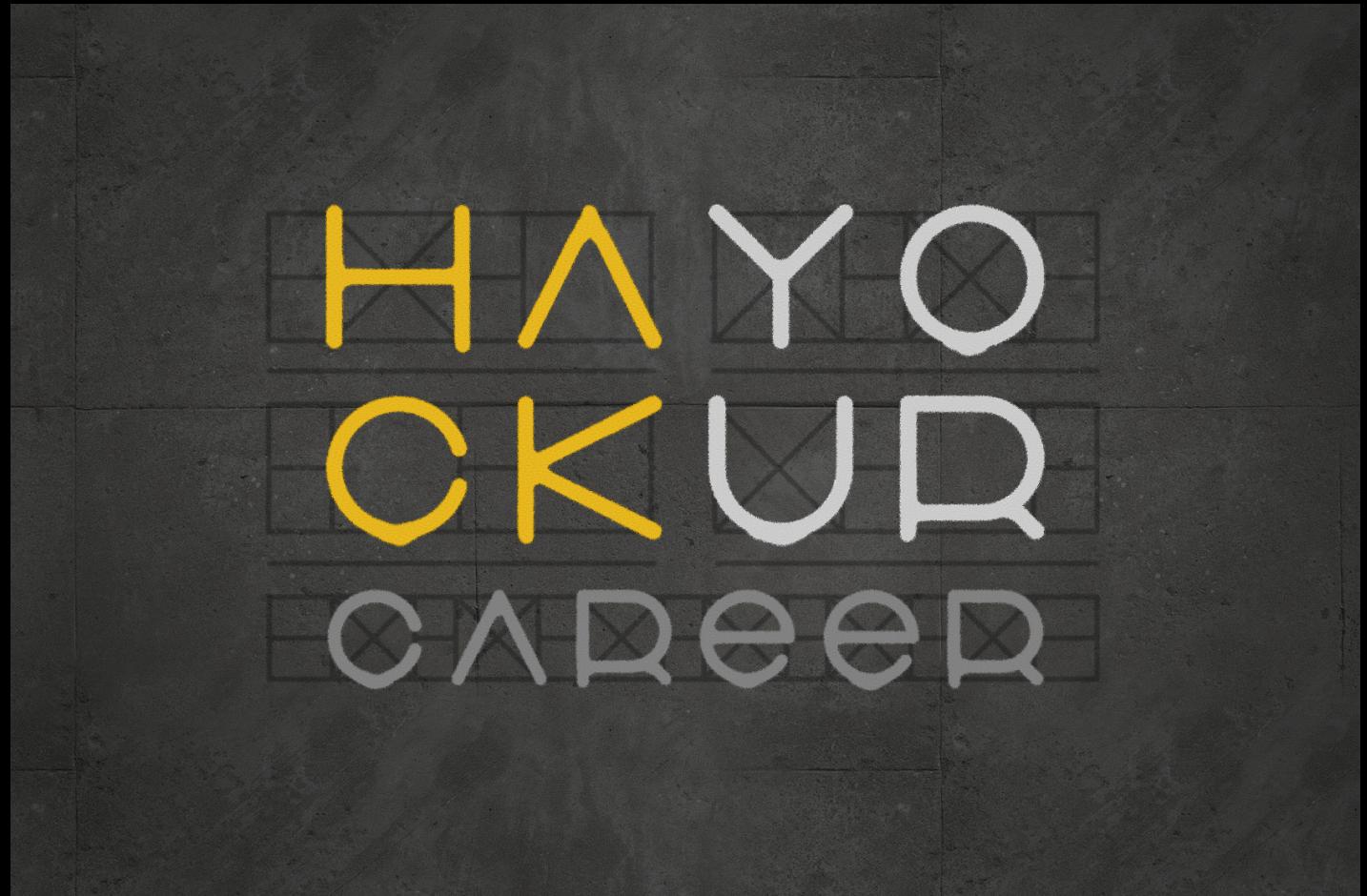


**Gajdlajny, GitHuby, riwju,  
czyli o co chodzi z tą  
dokumentacją.**

**Praca Technical Writera w  
Kymie.**

Maja Kurcius, SAP  
Dec 8, 2020

PUBLIC



# Agenda



# Hack Your Career

HAYO  
CKUR  
CAREER



# SAP Labs Poland

## Różnorodność domen

E-commerce, social commerce, cloud platform, open source,...

**Development:** Go, Java, Angular, Cloud Native solutions, ...

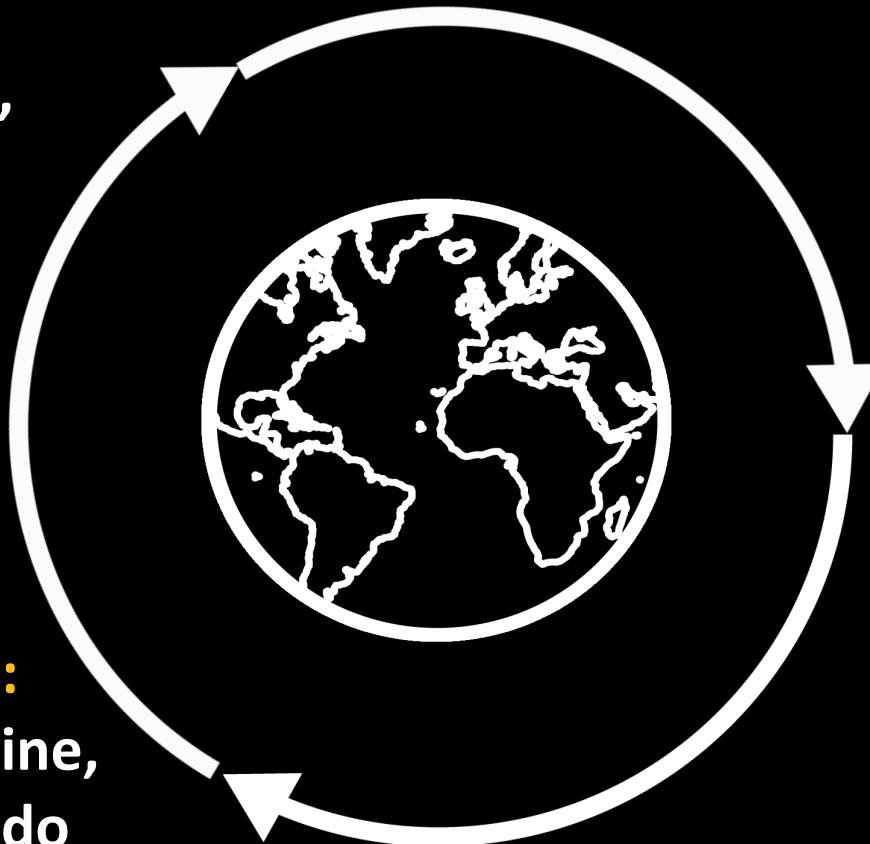
## Możliwość nauki i rozwoju:

Platformy do edukacji on-line, certyfikacje, łatwy dostęp do ekspertów na całym świecie

> 350 pracowników

**Różnorodność specjalizacji**  
Product teams, project teams, innovation projects teams, support teams

Jedno z 21 centrów SAP's Labs Network





Docs Blog Community Roadmap ⚙️ 🔎

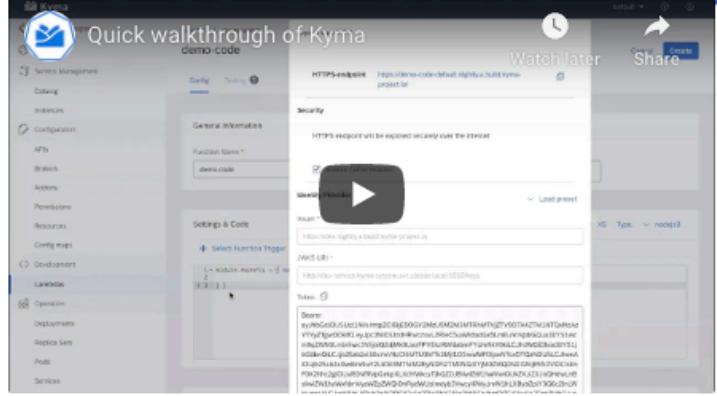
Star Fork 288

## Extend your applications with Kubernetes

Kyma /kee-ma/ is a platform for extending applications with serverless functions and microservices. It provides a selection of cloud-native projects glued together to simplify the creation and management of extensions.

```
bash
$ brew install kyma-cli
$ kyma provision minikube
$ kyma install
Installation successful! Happy Kyma-ing :)
```

[Install](#)



**Create extensions...**

Don't waste time rewriting your monoliths but focus on their extensibility instead. Make sure your application exposes an API and is able to send events to Kyma. Use our Console UI or CLI to connect, extend, and integrate applications.

**...with tools at hand**

Kyma gives you a properly configured, monitored, and secured Kubernetes cluster. It combines various open-source projects for authentication, logging, eventing, alerting, tracing, and many more. We ensured they play well together so you can only focus on coding.



# Maja Kurcius

Technical Writer,  
SAP Labs Poland.

- BA in English,  
University of Silesia
- BEng in Computer Science,  
Silesian University of Technology
- Former English Teacher,  
Poland & China
- Writer at Jezykowa Silka,  
JezykowaSilka.pl



# What you'll learn

- Technical Writer – the job
- Documentation – what & how
- **Documentation as code** approach + demo
- Useful tools: Git, GitHub, Markdown
- Useful resources: links

# Technical Writer

- NOT a ~~translator or editor~~
- Part of the development team
- Co-responsible for the whole product
- Creates, reviews, and maintains documentation and other technical content



# Technical Writer

## CREATES

- Working with a subject-matter expert (developer, ...)
- Based on input
- According to guidelines

# Technical Writer

## REVIEWS

- Checking others' content
- Language
- Style and formatting
- Content
- Accuracy and correctness
- Testing, if testable

# Technical Writer

## MAINTAINS

- Updating existing docs
- Fixing bugs
- Spotting and logging issues
- Estimating and prioritizing issues
- Solving existing issues/tasks

# Technical Writer

CONTENT developed

# Technical Writer

CONTENT developed

- Documentation

# Technical Writer

CONTENT developed

- Documentation
  - User-facing

# Technical Writer

CONTENT developed

- Documentation
  - User-facing
  - Internal

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
  - Internal

## Monitoring

### Overview ↴

Kyma comes bundled with third-party applications like Prometheus, Alertmanager, and Grafana, that offer a monitoring functionality for all Kyma resources. These applications are deployed during the Kyma cluster installation, along with a set of pre-defined alerting rules, Grafana dashboards, and Prometheus configuration.

The whole installation package provides the end-to-end Kubernetes cluster monitoring that allows you to:

- View metrics exposed by the Pods.
- Use the metrics to create descriptive dashboards that monitor any Pod anomalies.
- Manage the default alert rules and create new ones.
- Set up channels for notifications informing of any detected alerts.

**NOTE:** The monitoring component is available by default in the cluster installation, but disabled in the Kyma Lite local installation on Minikube. [Enable the component](#) to install it with the [local profile](#).

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
  - Internal

## Configuration

### Application Operator sub-chart

To configure the Application Operator (AO) sub-chart, override the default values of its `values.yaml` file. This document describes parameters that you can configure.

TIP: To learn more about how to use overrides in Kyma, see the following documents:

- [Sub-charts overrides](#)

#### Configurable parameters

This table lists the configurable parameters, their descriptions, and default values:

Parameter	Description	Default value
<code>controller.args.installationTimeout</code>	Specifies a period of time provided for the Application Gateway, Application Connectivity Validator, and Event Service installation. The Application requires these services to be operational. The value is provided in seconds.	240
<code>controller.args.helmDriver</code>	Specifies the backend storage driver used by Helm 3 to store release data. Possible values are <code>configmap</code> , <code>secret</code> and <code>memory</code> .	secret
<code>global.disableLegacyConnectivity</code>	Disables the default legacy AO work mode and enables the Compass mode.	false

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
  - Internal

### Revoke a client certificate

After you have established a secure connection with Compass and generated a client certificate, you may want to revoke this certificate at some point. To revoke a client certificate, follow the steps in this tutorial.

**NOTE:** A revoked client certificate remains valid until it expires, but it cannot be renewed.

#### Prerequisites

- [OpenSSL toolkit](#) to create a Certificate Signing Request (CSR), keys, and certificates which meet high security standards
- [Compass](#)
- Registered Application
- Runtime connected to Compass
- [Established secure connection with Compass](#)

**NOTE:** To see how to maintain a secure connection with Compass and renew a client certificate, read the [tutorial](#).

#### Steps

##### 1. Revoke the client certificate

To revoke a client certificate, make a call to the Certificate-Secured Connector URL using the client certificate. The Certificate-Secured Connector URL is the `certificateSecuredConnectorURL` obtained when establishing a secure connection with Compass. Send this mutation with the call:

```
mutation { result: revokeCertificate }
```

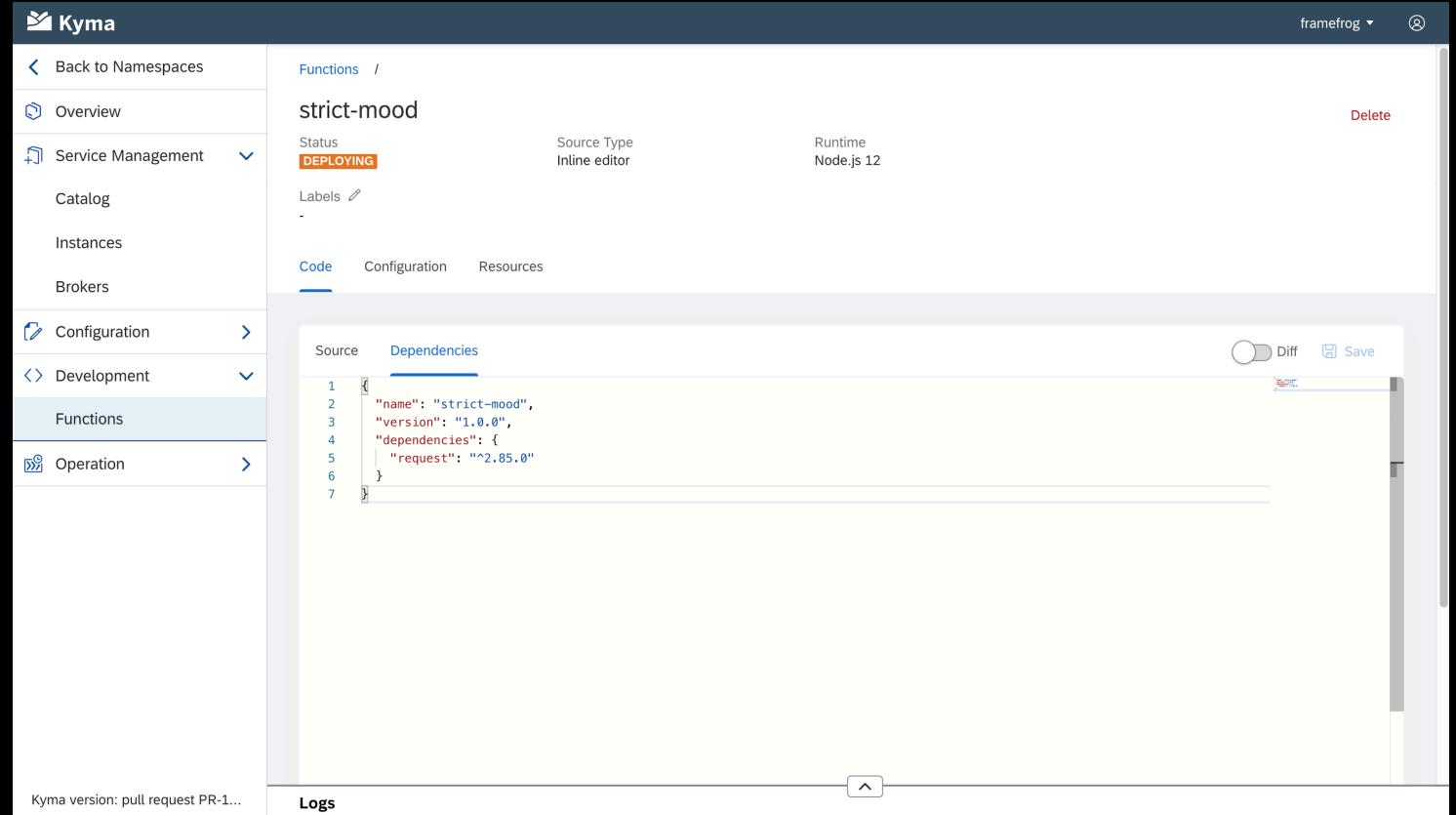
A successful call returns the following response:

```
{"data": {"result": true}}
```

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal



The screenshot shows the Kyma Platform interface for managing functions. The left sidebar navigation includes Back to Namespaces, Overview, Service Management (Catalog, Instances, Brokers), Configuration, Development (Functions selected), and Operation. The main content area displays the 'strict-mood' function details: Status (DEPLOYING), Source Type (Inline editor), and Runtime (Node.js 12). A 'Delete' button is visible. Below these details are tabs for Code, Configuration, and Resources, with the Code tab currently active. The code editor shows the following JSON configuration:

```
1 "name": "strict-mood",
2 "version": "1.0.0",
3 "dependencies": {
4   "request": "^2.85.0"
5 }
```

A 'Logs' tab is at the bottom of the main content area.

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes
- Release notes (RNs)

# Technical Writer

## Kyma 1.15 Vienna

Maja Kurcius, Technical Writer @Kyma on August 28, 2020



Same language, different city! After leaving Ulm, we take a short trip to the City of Music, Vienna. We all know Vienna for the famous musicians that were either born there or went there to work. But did you know that Vienna was the largest German-speaking city in the world up to the 20th century, and to date remains the second-largest one, after Berlin? Not to mention the beautiful architecture of the city! Inspired by the views, we worked on improving our Console UI. We also added more documentation sources to the website. Lastly, we worked on tweaks and improvements for monitoring, Kyma CLI, and the ORY stack. Read the notes to find out more.

See the overview of all changes in this release:

- [Monitoring](#) - Jaeger data source in Grafana
- [CLI](#) - Installation from a local checkout to a remote cluster, improved resilience of the `install` command
- [ORY stack](#) - Reduced Hydra installation time, increased availability of Hydra during updates, improved Oathkeeper stability
- [Console](#) - Managing OAuth2 clients from the Console UI, collapsible navigation categories, configuration of event triggers for services
- [Website](#) - Documentation from various repository sources displayed on the website

### Monitoring

#### Jaeger data source in Grafana

We took the chance and preconfigured the new Jaeger data source in Grafana. Now you can also explore all traces available in the Jaeger UI by selecting this new data source in the `Explore` section of Grafana.

### CLI

#### Installation from a local checkout to a remote cluster

Kyma installation on a remote cluster from a local Kyma GitHub checkout requires building and pushing the `kyma-installer` image to a remote Docker registry. From now on, CLI takes care of this process. When the `install` command detects such a setup, it will prompt you to pass the `--custom-image` parameter specifying the full image name to be used. The CLI will then build and push the image automatically, leveraging the authentication details retrieved from the local Docker client.

- ## Release notes (RNs)

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes
- Release notes (RNs)
- Deprecation notes

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes
- Release notes (RNs)
- Deprecation notes
- Migration guides

# Technical Writer

## Migrate from 1.15 to 1.16

### Istio

Kyma 1.16 comes with a new Istio version: 1.5.10, which introduces a new custom resource for Istio installation. If you use any overrides provided by Helm or IstioControlPlane, migrate them to the new [IstioOperator](#). See [an example of IstioOperator](#) for more information.

All Istio-related objects in Kyma are updated, but the migration may be necessary for the Istio objects created manually in your cluster. Please read the [Istio upgrade notes](#) for the details. Please ensure you're not using Istio RBAC, which is [deprecated](#) and may not work after the upgrade. Use the [Authorization Policy](#) to configure authorization for your services. In addition, consider migrating existing Authentication Policy to equivalent PeerAuthentication and RequestAuthentication objects.

### Eventing

In Release 1.16, we remove the support for Knative-Serving from Kyma. After the upgrade Knative-Serving is still installed and fully functional in the cluster, but can be removed easily in order to free resources. Removing Knative-Serving will not affect supported Kyma functionality.

Before you remove Knative-Serving from the cluster, make sure that you create a backup of any of your own resources that used Knative-Serving. You can do this by running:

```
kubectl get serving -A -oyaml > <backup-file-name>
```

To remove the `knative-serving` chart, execute:

```
helm delete knative-serving -n knative-serving
```

**NOTE:** This requires Helm version 3.x

This command renders Knative-Serving inactive, but all user-created configuration is still available in the cluster. To remove the CustomResourceDefinitions for Knative-Serving along with the respective CustomResources, run:

```
kubectl delete crd \
certificates.networking.internal.knative.dev \
configurations.serving.knative.dev \
images.caching.internal.knative.dev \
```

- **Release notes (RNs)**
- **Deprecation notes**
- **Migration guides**

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes
- Release notes (RNs)
- Deprecation notes
- Migration guides
- Blog posts, videos, ...

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes
- Release notes (RNs)
- Deprecation notes
- Migration guides
- Blog posts, videos, ...
- Other

# Technical Writer

## CONTENT developed

- Documentation
  - User-facing
    - Functionality
    - Configuration
    - Tutorials
    - UI text
  - Internal
    - Dev guides & how to's
    - Processes
- Release notes (RNs)
- Deprecation notes
- Migration guides
- Blog posts, videos, ...
- Other

# Documentation

## CONTEXT

- Recipient – who for
- Purpose – what for



# Documentation

## WHAT to document

- User-facing functionality



# Documentation

## WHAT to document

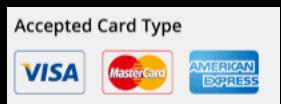
- User-facing functionality
- Configurable parameters, components



# Documentation

## WHAT to document

- User-facing functionality
- Configurable parameters, components
- Prerequisites



# Documentation

## WHAT to document

- User-facing functionality
- Configurable parameters, components
- Prerequisites
- Possible values



# Documentation

## WHAT to document

- User-facing functionality
- Configurable parameters, components
- Prerequisites
- Possible values
- How-to – tutorials, guides



# Documentation

## HOW to document

- Guidelines
  - Structure
  - Templates
  - Language
  - Grammar & punctuation
  - Formatting
  - Diagrams & screenshots
  - Naming

# Documentation

title
Style and terminology

These are the guidelines for the grammar, preferred word choices, and the list of terms to capitalize when writing Kyma documentation. The goal of these guidelines is to have all contributors write in the same way to ensure a uniform flow throughout the whole Kyma documentation.

## Grammar

These are the generally accepted grammar rules for writing Kyma documentation.

### Active voice

Use active voice whenever possible. Active voice is clear, concise, and it avoids misinterpretation. It is also easier for non-native speakers to understand. Passive voice is indirect, uses more words, and can be misleading because it reverses the logical order of events.

The endpoint path includes your service name.

Your service name is to be included in the endpoint path.

### Voice and tone

There are different tones for different types of technical documentation. The documentation can range from instructional to somewhat conversational, but always with the goal of helping users understand how to use the product for practical purposes and, in blogs and release notes, also helping business users understand changes. While writing Kyma documentation, use semi-formal style and imperative mood. The imperative mood tells the reader directly to do something. Use the imperative mood to write instructional documentation such as procedures and tutorials. Other moods can imply optional behavior.

**NOTE:** Avoid using unnecessary words such as "please" or "remember."

Click Add.

Click the Add button.

Please, click Add.

Remember to click Add.

Click Upload and select one or more documents.

If you want to upload a document, you can click Upload.

### Tenses

title
Formatting

These are the guidelines for formatting text and using specific elements such as lists, tables, headings, and links. For guidelines regarding the creation of diagrams, see [this document](#).

## Code formatting

It is important to consistently format items such as code or filenames to quickly distinguish them while reading technical documentation. The following tables outline when to use **bold** font and when to use `code` font:

### Use bold font for these items:

Items	Examples
Parameters	The <code>env</code> attribute is optional.
HTTP headers	The Authorization Proxy validates the JWT token passed in the <code>Authorization Bearer</code> request header.
Events	The service publishes an <code>order.created</code> event.
Roles	Only the users with the <code>kyma_admin</code> role can list Pods in the Kyma system Namespaces.
UI elements	Click <code>Subscribe</code> .
Variables and placeholders	Click <code>Project &gt; {YOUR_PROJECT_NAME}</code> .

### Use code font for these items:

Items	Examples
Code examples	Get the list of all Pods in a Namespace using the <code>kubectl get pods -n {NAMESPACE}</code> command.
Values	Set the <code>partial</code> attribute to <code>true</code> to perform a partial replacement.

# Documentation

## HOW to document

- Guidelines
  - Structure
  - Templates
  - Language
  - Grammar & punctuation
  - Formatting
  - Diagrams & screenshots
  - Naming
- Process
  - Scrum
  - Review
  - Approval from content owner required
- Tools and deployment
  - Markdown
  - GitHub
  - Website

# Documentation

## QUALITIES of good documentation

- Simple
  - Vocabulary
  - Grammar
- Clear
- User-friendly
- Coherent & consistent
- Relevant
- Helpful

# Documentation as code

- Docs and code developed and maintained equally
- The same processes and tools
  - IDE
  - VCS
  - Testing
  - Building and deployment
- Code and docs released together as integral parts of released functionality

# VCS, Git, GitHub, IDE

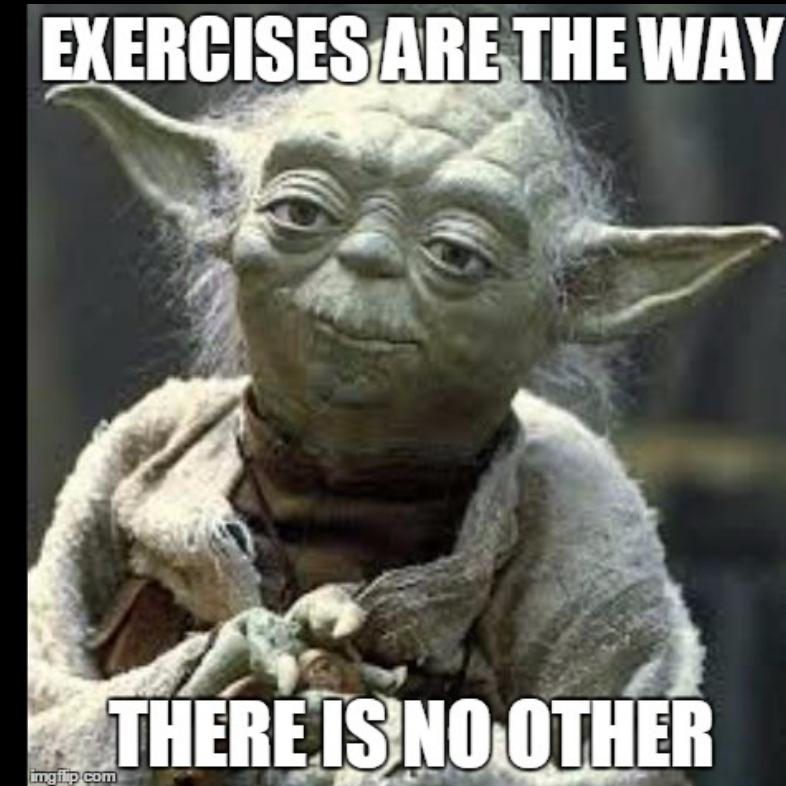
- VCS (Version Control System) – a system that keeps track of every change made to your files over time (not like Dropbox)
- Git – a free, fast, efficient, open-source VCS
- GitHub – a platform for hosting Git repositories with extra bells and whistles
- IDE (Integrated Development Environment) – software for developing applications that consolidates different development tools.

Popular free ones: [Visual Studio Code](#), [Atom](#).

# Demo

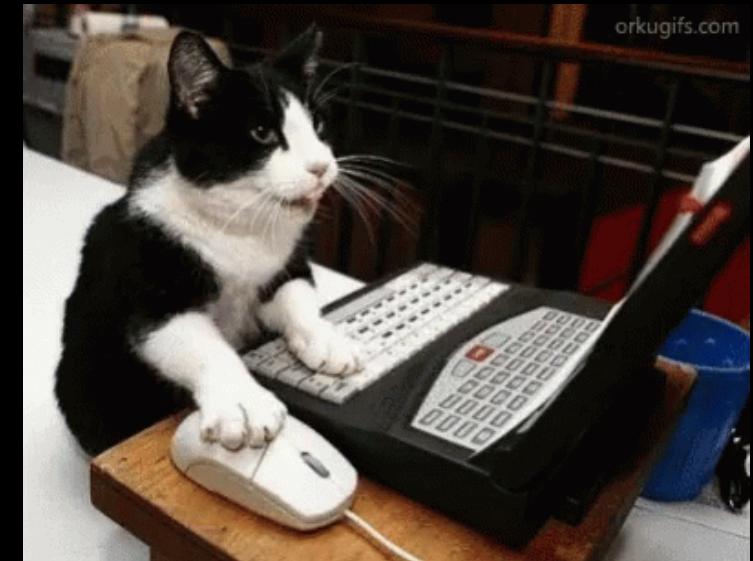


Try it!



# Try it!

- [Learn Markdown basics in 5 mins](#)
- [Learn how to add a new GitHub issue template & create your own pull request \(PR\)](#)
- [Work on real tasks and contribute to Kyma](#)
  - Good first issues: [Kyma](#), [Community](#)



# Summary

Now you know

- Documentation process @ Kyma
- Content types @ Kyma
- **Documentation as code** approach
- VCS: Git
- GitHub



# Resources

- [Kyma Website](#)
- [Kyma docs \(GitHub\)](#)
- [Kyma blog & release notes](#)
- [Kyma content guidelines](#)
- [WriteTheDocs.org](#)
- [TechWriter.pl](#)

# Questions?

# Hack Your Career

Current job offers at SAP:

- <https://jobs.sap.com/search/?locationsearch=gliwice>  
(short URL: <https://url.sap/3vywdy>)

Follow us:

- <https://hackyourcareer.github.io/>
- <https://www.facebook.com/Hack.your.Career/>

# Survey

Please fill out the survey to let us know how you liked the presentation:

<https://www.surveymonkey.com/r/5SJ2FKY>

Thank you!

# Thank you.

Feel free to reach out to me

Email: [maja.kurcius@sap.com](mailto:maja.kurcius@sap.com)

GitHub: [@majakurcius](https://github.com/majakurcius)

