

--	--	--	--

Doom  [rspress](#)  Alauda           

□ □

□ □ □ □

□□ Markdown □□□□ MDX

□ □ □ □

- □□□□□□□□□□□□□□□□
- □□□□□□□
- □□□□□

□□ **Markdown** □□□□ **MDX**

[illegible]



0000000000 MDX 00 ↗ 0

0000

0000000000 Doom 00

00

00

0000

000000

00

11

□□ `doom` □□□□

4/4

API □□□□

Page 10

Page 10 of 10

Page 10

□ □ □ □ □

Page 10

□ □ □ □ □ □ □ □ □

--	--	--	--	--	--	--	--

sites.yaml

□ □ □ □

□ □ □ □ □ □ □ □ □

□ □ □ □ □ □

11

4/4

333

11

11

Markdown

Callouts

Mermaid

MDX

00 MDX 000000000000000000

rspress 00

doom 00

00000000

000

0000000000000000

i18n.json

.ts/.tsx

.mdx

API 00

00 API

CRD

0000

00 openapi 00

000000

props

00

0000

000000

00

000000000000000000000000 ACP 00

000000

0000

000 ACP

📄📄

📄📄

📄📄📄

📄📄📄📄

📄📄📄📄📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄

📄📄📄

📄📄 PDF

📄📄📄

📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

```
mkdir my-docs && cd my-docs
```

📄 `npm init -y` 📄📄📄📄📄📄📄📄📄📄📄📄📄 `npm`📄`yarn` 📄 `pnpm` 📄📄 `doom:`

 `npm`  `yarn`  `pnpm`  `bun`

```
npm install -D @alauda/doom typescript
```

📄📄📄📄📄📄📄📄📄:

```
# 创建 docs 目录
```

```
mkdir docs/en && echo '# Hello World' > docs/en/index.md
```

```
mkdir docs/zh && echo '# 你好' > docs/zh/index.md
```

在 `package.json` 中添加脚本:

```
{
  "scripts": {
    "dev": "doom dev",
    "build": "doom build",
    "new": "doom new",
    "serve": "doom serve",
    "translate": "doom translate",
    "export": "doom export"
  }
}
```

在 `doom.config.yml` 中添加:

```
title: My Docs
```

在 `tsconfig.json` 中添加:

```
{
  "compilerOptions": {
    "jsx": "react-jsx",
    "module": "NodeNext",
    "moduleResolution": "NodeNext",
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "resolveJsonModule": true,
    "skipLibCheck": true,
    "strict": true,
    "target": "ESNext",
  },
  "mdx": {
    "checkMdx": true,
  },
}
```

global.d.ts

```
/// <reference types="@alauda/doom/runtime" />
```

.mdx doom


```
doom -h
```

```
# output
```

```
Usage: doom [options] [command]
```

```
Doctor Doom making docs.
```

Options:

-V, --version	output the version number
-c, --config <config>	Specify the path to the config file
-v <version>	Specify the version of the documentation, c
-b, --base <base>	Override the base of the documentation
-p, --prefix <prefix>	Specify the prefix of the documentation bas
-f, --force [boolean]	Force to
	1. fetch latest reference remotes or scaffo
	2. translate ignore hash equality check and
-i, --ignore [boolean]	Ignore internal routes (default: false)
-d, --download [boolean]	Display download pdf link on nav bar (defau
-e, --export [boolean]	Run or build in exporting PDF mode, `apis/*
-I, --include <language...>	Include **only** the specific language(s),
-E, --exclude <language...>	Include all languages except the specific l
-o, --out-dir <path>	Override the `outDir` defined in the config
-r, --redirect <enum>	Whether to redirect to the locale closest t
-R, --edit-repo [boolean url]	Whether to enable or override the `editRepo
-n, --no-open [boolean]	Do not open the browser after starting the
-h, --help	display help for command

Commands:

dev [options] [root]	Start the development server
build [root]	Build the documentation
preview serve [options] [root]	Preview the built documentation
new [template]	Generate scaffolding from templates
translate [options] [root]	Translate the documentation
export [options] [root]	Export the documentation as PDF, `apis/**`
help [command]	display help for command

```
📄📄📄📄📄📄📄
```

```
📄📄📄📄📄
```

```
📄 yarn dev 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄
```

```
doom dev -h
```

```
# output
```

```
Usage: doom dev [options] [root]
```

```
Start the development server
```

```
Arguments:
```

```
  root
```

```
Root directory of the documentation
```

```
Options:
```

```
  -H, --host [host]
```

```
Dev server host name
```

```
  -P, --port [port]
```

```
Dev server port number
```

```
  -l, --lazy [boolean]
```

```
Whether to enable `lazyCompilation` which could i
```

```
  -h, --help
```

```
display help for command
```

```
~~~~~
```

```
❯ yarn build ~~~~~ dist ~~~~~
```

```
~~~~~
```

```
❯ yarn serve ~~~~~ -b, -p ~~~~~ -b, -p
```

```
❯
```

```
~~~~~
```

```
❯ yarn new ~~~~~
```

```
~~~~~
```

```
doom translate -h
```

```
# output
```

```
Usage: doom translate [options] [root]
```

Translate the documentation

Arguments:

root Root directory of the documentation

Options:

-s, --source <language> Document source language, one of en, zh, ru (default)

-t, --target <language> Document target language, one of en, zh, ru (default)

-g, --glob <path...> Glob patterns for source dirs/files

-C, --copy [boolean] Whether to copy relative assets to the target directory

-h, --help display help for command

- -g, --glob 指定源目录的glob模式，glob 模式指定了源目录的搜索范围。

1. yarn translate -g abc xyz 指定 <root>/<source>/abc , <root>/<source>/xyz 为源目录，<root>/<target>/abc , <root>/<target>/xyz 为目标目录。

2. yarn translate -g '*' 指定 <root>/<source> 为源目录。

- -C, --copy 指定是否将源目录中的相对资产复制到目标目录，false 表示不复制。

- 示例

1. /<source>/abc.jpg 复制到 <root>/public/<source>/abc.jpg 并复制到 <root>/public/<target>/abc.jpg 并复制到 <root>/<target>/abc.jpg

2. <root>/<source>/abc.mdx 复制到 ./assets/xyz.jpg 并复制到 <root>/<source>/assets/xyz.jpg 并复制到 <root>/<target>/assets/xyz.jpg 并复制到 <root>/<target>/assets/xyz.jpg

3. <root>/<source>/abc.mdx 复制到 ./assets/<source>/xyz.jpg 并复制到 <root>/<source>/assets/<source>/xyz.jpg 并复制到 <root>/<target>/assets/<target>/xyz.jpg 并复制到 ./assets/<target>/xyz.jpg

- 📄📄📄📄📄📄
 1. `/<source>/abc.jpg` 📄📄📄 `<root>/public/<target>/abc.jpg` 📄📄📄📄📄📄
📄📄📄📄 `/<target>/abc.jpg` 📄📄📄📄📄📄📄📄📄📄
 2. `<root>/<source>/abc.mdx` 📄📄📄 `./assets/<source>/xyz.jpg` 📄📄📄📄
`<root>/<target>/assets/<target>/xyz.jpg` 📄📄📄📄📄📄📄📄📄📄
`./assets/<target>/xyz.jpg` 📄📄📄📄📄
`../<source>/assets/<target>/xyz.jpg`

WARNING

📄📄📄📄📄 `-g '*'` 📄📄📄📄📄📄📄 `source` 📄 `target` 📄📄📄📄📄
`internalRoutes` 📄📄📄📄📄 `target` 📄📄📄📄📄

TIP

📄📄📄📄📄📄 `AZURE_OPENAI_API_KEY` 📄📄📄📄📄📄📄 `Leader` 📄📄

📄📄📄📄📄📄📄📄📄📄📄📄

```
i18n:
  title:
    en: DevOps Connectors
  additionalPrompts: '📄📄📄 Connectors 📄📄📄📄📄📄'
  disableAutoTranslation: false
title: DevOps 📄📄
```

📄📄📄📄📄📄📄📄📄

📄📄 **PDF**

WARNING

📄📄📄📄📄📄📄📄 `yarn build` 📄📄📄

```
doom export -h
```

```
# output
```

Usage: doom export [options] [root]

Export the documentation as PDF, ``apis/**`` and ``*/apis/**`` routes will be ign

Arguments:

```
root      Root directory of the documentation
```

Options:

`-H, --host [host]` Serve host name

```
-P, --port [port]  Serve port number (default: "4173")
```

```
-h, --help      display help for command
```

❏ yarn export ❏❏❏❏ PDF ❏❏❏❏❏❏❏❏❏❏ -b, -p ❏❏❏❏❏❏❏❏❏❏❏❏❏❏❏❏ -b, -p

❏❏

playwright build-harbor.alauda.cn/frontend/playwright-runner:doom

```
PLAYWRIGHT_DOWNLOAD_HOST="https://cdn.npmmirror.com/binaries/playwright"
```

--	--	--	--

```
doom lint -h
```

```
# output
```

Usage: **doom lint** [options] [root]

Lint the documentation

Arguments:

root Root directory of the documentation

Options:

```
-h, --help  display help for command
```

0000000000000000



📄

📄 `doom` 📄📄

📄📄

📄📄

API 📄📄

📄📄📄📄📄

📄📄📄

📄📄📄

📄📄📄

📄📄📄📄📄

📄📄📄📄📄📄

📄📄📄📄📄

`sites.yaml` 📄

📄📄

📄📄📄📄📄📄

📄📄📄

📄

📄“📄📄📄📄”📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

📄📄

📄📄

📄

📄

Markdown

Callouts

Mermaid

MDX

📄 MDX 📄📄📄📄📄📄📄📄📄📄

rspress 📄

doom 📄

📄📄📄📄📄

📄📄

📄📄📄📄📄📄📄📄📄

i18n.json

.ts/.tsx

.mdx

API 📄

📄 API

CRD

📄📄📄

📄 openapi 📄

📄📄📄📄📄

props

📄

0000

000000

00

00000000000000000000 ACP 00

00000

0000

000 ACP

□ □ □ □ □ □

```
import { defineConfig } from '@alauda/doom/config'

export default defineConfig({})
```

--	--	--	--

- `lang` 指定する言語のコード。例: `en` (英語), `ja` (日本語)。
指定しない場合は `null` または `undefined`。
- `title` ページのタイトル。
- `logo` 公開するロゴのパス。例: `public/logo.png`。
指定しない場合は `doom` または `alauda logo`。
- `logoText` ロゴのテキスト。
- `icon` ファビコンのパス。例: `logo`。
- `base` ベースパス。例: `product-docs`。
指定しない場合は `/`。
- `outDir` 出力ディレクトリ。例: `dist/{base}/{version}`。
指定しない場合は `dist/{outDir}/{version}`。
例: `version` は `1.0.0`。

API ☐ ☐ ☐ ☐

```
# doo.config.* glob json/yaml
permission:
  functionresources:
    # `kubectl get functionresources`
    - docs/shared/functionresources/*.yaml
  roletemplates:
    # `kubectl get roletemplates -l auth.cpaas.io/roletemplate.official=true`
    - docs/shared/roletemplates/*.yaml
```

reference:

```

- repo: alauda-public/product-doc-guide # 
branch: # [string] 
publicBase: # [string] /images/xx.png do
sources:
  - name: anchor # 
    path: docs/index.mdx# # doom.conf
    ignoreHeading: # [boolean] true
    processors: 
      - type: ejsTemplate
        data: # ejs `<%= data.xx %>` 
    frontmatterMode: merge # frontmatter ignore ign

```

frontmatterMode

- **ignore** frontmatter frontmatter
- **merge** frontmatter key
- **replace** frontmatter frontmatter
- **remove** frontmatter

releaseNotes:**queryTemplates:**

```

fixed: # ejs jql 
unfixed:

```

```
<!-- release-notes-for-bugs?template=fixed&project=DevOps -->
```

```
{/* release-notes-for-bugs?template=fixed&project=DevOps */}
```

--	--	--	--	--

```
sidebar:
  collapsed: false # oooooooooooooooooooooooooooooooooooooo false
```

--	--	--	--	--	--	--	--

```
internalRoutes: # glob docs cli -i, --ignore`
- '*/internal/**'
```

--	--	--	--	--	--	--	--	--

```
onlyIncludeRoutes: # 匹配 glob 匹配的 docs 中的 cli 包，忽略 ` ` 匹配的
  - '*/internal/**'
internalRoutes:
  - '*/internal/overview.mdx'
```

--	--	--	--	--	--	--	--

plaintext

```
sites.yaml [REDACTED] [REDACTED]
```

```
[REDACTED]
```

--	--	--	--

translate:

```
# ===== `sourceLang`, `targetLang`, `userPrompt` o `additionalPrompts` =====
#   `sourceLang` o `targetLang` o `` o ``
#   `userPrompt`
#   `additionalPrompts` o `frontmatter.i18n` o `additionalPrompts` o
#
systemPrompt: |
##
===== <%= targetLang %>===== <%= sourceLang %>=====<%=
targetLang %>=====

##
- <%= sourceLang %>===== <%= targetLang %>=====
- MDX MDX MDX jsx <Overview />
-
- MDX MDX MDX
  - <Overview /> o Overview
  - <Tab label="value"></Tab>o label o key "value"
<%= terms %>
-
- {/* release-notes-for-bugs */}
- <!-- release-notes-for-bugs -->
-
- {/* reference-start */}
- {/* reference-end */}
- <!-- reference-start -->
- <!-- reference-end -->
- \< o \\{
- Markdown frontmatter, frontmatter.i18n

##
=====
1. <%= sourceLang %>===== <%= targetLang %>=====
2.
  - <%= targetLang %>=====
  -
  -
3. <===== <%= targetLang %>=====
4. <===== <%= targetLang %>===== <%= targetLang %>=====

=====

<%= userPrompt %>
```



```
<%= additionalPrompts %>
```

```
userPrompt: # `systemPrompt` `ejs`
```

```
editRepoBaseUrl: alauda/doom/tree/main/docs # https://github.com/
```

```
lint:
```

```
cspellOptions: # cspell https://github.com/streetsidesoftware/csp
```

11

11

index.mdx Overview

```
#  
  
<Overview />
```

Markdown

[gfm](#)
Doom
Markdown

Callouts

Mermaid

Callouts

NOTE

- ; , % , # , // , /** */ , --
<!-- -->
- [!\code callout]
- :::callouts
<div class="doom-callouts">
<Callouts>

```
```sh
```

```
Memory overhead per virtual machine ≈ (1.002 × requested memory) \
 + 218 MiB \
 + 8 MiB × (number of vCPUs) \
 + 16 MiB × (number of graphics devices) \
 + (additional memory overhead)
```

```
```
```

```
:::callouts
```

1. Required for the processes that run in the `virt-launcher` pod.
2. Number of virtual CPUs requested by the virtual machine.
3. Number of virtual graphics cards requested by the virtual machine.
4. Additional memory overhead:
 - If your environment includes a Single Root I/O Virtualization (SR-IOV) network device or a Graphics Processing Unit (GPU), allocate 1 GiB additional memory overhead for each device.
 - If Secure Encrypted Virtualization (SEV) is enabled, add 256 MiB.
 - If Trusted Platform Module (TPM) is enabled, add 53 MiB.

```
:::
```

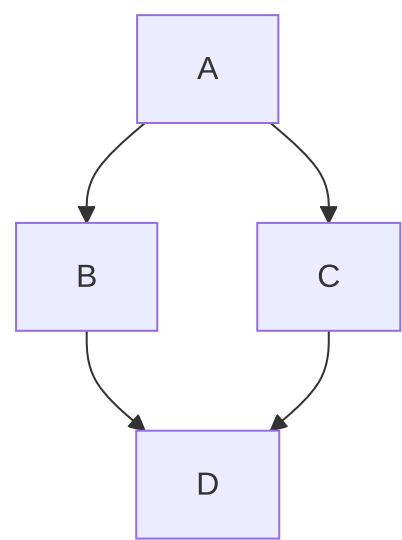
```
Memory overhead per virtual machine ≈ (1.002 × requested memory) \
    + 218 MiB \
    + 8 MiB × (number of vCPUs) \
    + 16 MiB × (number of graphics devices) \
    + (additional memory overhead)
```

- ① Required for the processes that run in the `virt-launcher` pod.
- ② Number of virtual CPUs requested by the virtual machine.
- ③ Number of virtual graphics cards requested by the virtual machine.
- ④ Additional memory overhead:
 - If your environment includes a Single Root I/O Virtualization (SR-IOV) network device or a Graphics Processing Unit (GPU), allocate 1 GiB additional memory overhead for each device.
 - If Secure Encrypted Virtualization (SEV) is enabled, add 256 MiB.
 - If Trusted Platform Module (TPM) is enabled, add 53 MiB.

Mermaid ↗

□□□□□

```
```mermaid
graph TD;
 A-->B;
 A-->C;
 B-->D;
 C-->D;
```
```



□□ [Markdown Preview Mermaid ↗](#) □□□ VSCode □□□□□

MDX

[MDX](#) [Markdown](#) [Markdown](#) [JSX](#) [rspress MDX](#)

rspress

doom

- Overview
- Directive
- ExternalSite
- ExternalSiteLink
- AcpApisOverview ExternalApisOverview

Term

- props
- TermsTable
- props
- JsonViewer

rspress

rspress [rspress](#) .mdx

- Badge
- Card

- [LinkCard](#)
- [PackageManagerTabs](#)
- [Steps](#)
- [Tab/Tabs](#)
- [Toc](#)

在 @rspress/core/theme 中

```
import { SourceCode } from '@rspress/core/theme'

<SourceCode href="/" />
```

doom 文档

doom 文档

Overview

文档

Directive

在文档中，[Directive](#) 用于

```
- doc(en) doc(zh)

<Directive type="danger" title="危险">
  doc(zh)
</Directive>
```

- doc(en) doc(zh) 文档

11

doc/zh

11

ExternalSite

--	--	--	--	--	--	--	--

```
<ExternalSite name="connectors" />
```

Note

DevOps DevOps

DevOps ☐ ☐ ☐ ☒

ExternalSiteLink

[illegible]

```
<ExternalSiteLink name="connectors" href="link.mdx#hash" children="Content" /
```

[Content ↗](#)

TIP

```
mdx <ExternalSiteLink name="connectors" href="link" children="Content" />
```

□ □ □ □ □ □ □ □ □ □

```
<ExternalSiteLink name="connectors" href="link">
  Content {/* □□□□ `p` □□□ */}
</ExternalSiteLink>
```

000000000000 **p** 00000000000000000000 **children** 0000

AcpApisOverview ExternalApisOverview

0000000 API 0000

```
<AcpApisOverview />
{/* same as following */}
<ExternalApisOverview name="acp" />

<ExternalApisOverview name="connectors" />
```

Note

00 ACP APIs 000000000000 [ACP APIs 00 ↗](#) 0

Note

00 DevOps 000 APIs 000000000000 [DevOps 000 APIs 00 ↗](#) 0

Term

00000000000000000000

```
<Term name="company" textCase="capitalize" />
<Term name="product" textCase="lower" />
<Term name="productShort" textCase="upper" />
```

000 00000000 ACP

props

- **name** : 00000000000000000000
- **textCase** : 0000000000000000 **lower** , **upper** , **capitalize**

TermsTable

```
<TermsTable />
```

company		-	Alauda	-	
product		-	Alauda Container Platform	-	
productShort	ACP	-	ACP	-	

props

- terms : NormalizedTermItem[]

JsonViewer

```
<JsonViewer value={{ key: 'value' }} />
```

```
yaml  json
```

key: value

shared

```
import CommonContent from './shared/CommonContent.mdx'
```

```
<CommonContent />
```

runtime ↗ API .jsx/.tsx .mdx

```
// shared/CommonContent.tsx
```

```
export const CommonContent = () => {
  const { page } = usePageData()
  return <div>{page.title}</div>
}
```

```
// showcase/content.mdx
```

```
import { CommonContent } from './shared/CommonContent'
; <CommonContent />
```

WARNING

.mdx props issue ↗ props
.jsx/.tsx

文档

alauda 文档 en / zh 文档
 public 文档 en / zh 文档

文档

i18n.json

.ts/.tsx

.mdx

i18n.json

文档 docs 文档 i18n.json 文档
 useI18n 文档

```

{
  "title": {
    "zh": "标题",
    "en": "Title"
  },
  "description": {
    "zh": "描述",
    "en": "description"
  }
}
```

.ts/.tsx

```
import { useI18n } from '@rspress/runtime'

export const CommonContent = () => {
  const t = useI18n()
  return <h1>{t('title')}</h1>
}
```

.mdx

```
import { useI18n } from '@rspress/runtime'

# {useI18n()('title')}

{useI18n()('description')}
```

API 文档

本页面提供 API 文档 API 文档 CRD (Custom Resource Definition) 文档

```
├─ apis
│   ├── advanced-apis # 高级 API
│   ├── crds # CRDs
│   └── references # 参考
```

API

- API
 - props
- CRD
 - props
- 文档
 - props
- openapi 文档

API

```
# CodeQualityTaskSummary

<OpenAPIPath path="/plugins/v1alpha1/template/codeQuality/task/{task-id}/summ
```

CodeQualityTaskSummary

props

- `path` : OpenAPI schema `paths` 下的路径
- `pathPrefix` : 路径前缀 `api.pathPrefix`
- `openapiPath` : 指向 `openapi` 的路径

CRD

```
# ArtifactCleanupRun
```

```
<K8sCrd name="artifactcleanupruns.artifacts.katanomi.dev" />
```

指向 `ArtifactCleanupRun`

props

- `name` : CRD `metadata.name`
- `crdPath` : 指向 `openapi` 下的 `CRD` 的路径

CRD

```
# CodeQuality
```

```
<OpenAPIRef schema="v1alpha1.CodeQuality" />
```

指向 `CodeQuality`

props

- `schema` : OpenAPI schema `definitions` (v2) or `component/schemas` (v3) 下的 schema

- `openapiPath` : 参照する `openapi` の ID

参照する `openapi` の ID

参照する `OpenAPIPath` の `OpenAPIRef` の参照する `openapi` の ID を指定する。
参照する `openapi` の ID を `openapiPath` で指定する。

```
<OpenAPIPath  
  path="/plugins/v1alpha1/template/codeQuality/task/{task-id}/summary"  
  openapiPath="shared/openapis/katanomi.json"  
>
```

权限管理

```
<K8sPermissionTable functions={['devops-testplans', 'devops-testmodules']} />
```

权限

props

权限

props

- functions: string[] - 权限资源 FunctionResource 列表

权限

权限	权限	权限 资源	权限 资源	权限 资源	权限 资源	权限 资源	权限 资源
权限 devops- testplans	权限	✓	✓	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
权限 devops-	权限	✓	✓	✓	✓	✓	✗

模块	测试	测试用例 通过	测试用例 失败	测试用例 通过	测试用例 通过	测试用例 通过	测试用例 通过
testmodules	测试	✓	✗	✓	✓	✓	✗
	测试	✓	✗	✓	✓	✓	✗
	测试	✓	✗	✓	✓	✓	✗

reference:

```

- repo: alauda-public/product-doc-guide # 
branch: # [string] 
publicBase: # [string] /images/xx.png do
sources:
  - name: anchor # 
    path: docs/index.mdx# # doom.conf
    ignoreHeading: # [boolean] true
    processors: 
      - type: ejTemplate
        data: # ejs `<%= data.xx %>` 
    frontmatterMode: merge # frontmatter ignore ign

```

frontmatterMode

- **ignore** frontmatter frontmatter
- **merge** frontmatter key
- **replace** frontmatter frontmatter
- **remove** frontmatter

0000000000

□ □ □ □ □ □ □ □

FROM build-harbor.alauda.cn/ops/alpine:latest

WORKDIR /docs

COPY . dist

□ □ □ ACP

--	--	--	--	--

```

$ doom build dist -v

```

```
# release-4.0 4.0
doom build -v 4.0 # 4.0 dist/4.0 {base}/4.0
doom build -v master # master dist/master {base}/master
doom build -v {other} # dist/{other} {base}/{other}

# unversioned unversioned-x.y
doom build -v unversioned # dist/unversioned {base}
doom build -v unversioned-4.0 # 4.0 dist/unversioned-4.0 {base}
```




```
|— console-platform
|   |— 4.0
|   |— 4.1
|   |— index.html
|   |— overrides.yaml
|   └─ versions.yaml
|— console-devops-docs
|   |— 4.0
|   |— 4.1
|   |— index.html
|   |— overrides.yaml
|   └─ versions.yaml
|— console-tekton-docs
|   |— 1.0
|   |— 1.1
|   |— index.html
|   |— overrides.yaml
|   └─ versions.yaml
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Redirecting...</title>
    <meta http-equiv="refresh" content="0; url=/console-docs/4.1" />
  </head>
  <body>
    <p>Redirecting to <a href="/console-docs/4.1">/console-docs/4.1</a></p>
  </body>
</html>
```

☐☐☐☐☐☐☐☐

```
# console-platform
# https://gitlab-ce.alauda.cn/idp/Doom/-/blob/master/src/terms.ts#L11
terms:
  company:
    en: Alauda
    zh:
  product:
    en: Alauda Container Platform
    zh:
  productShort:
    en: ACP

#
title:
  en: Doom - Alauda
  zh: Doom -
logoText:
  en: Doom - Alauda
  zh: Doom -
```

- '4.1'
- '4.0'

[chart-frontend](#) [alauda-docs](#) [check-alauda-docs](#)

[doom](#) [idp](#) [webapp](#)

INFO

PR gitops