□□□□

Doom □□□ [rspress↗](#) □□□□□□□ Alauda □□□

□□

□□□□

□□ Markdown □□□□ MDX

□□□□

□□□□

- □□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□

□□ Markdown □□□□ MDX

MDX □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Markdown □□□□□□□□ Markdown □□□□□□□ React □□□

```
// docs/index.mdx
import { CustomComponent } from './custom';

# Hello World

<CustomComponent />
```

□□□□□□□□□ MDX□ □□ ↗□

□□□□

□□□□□□□□□□□ Doom □□

□□

□□
□□□□
□□□□□

□□

□□

□□ `doom` □□□□

□□□□

□□□□

API □□□□

□□□□□□□□

□□□□□□

□□□□□□

□□□□□

□□□□□□□

□□□□□□□□

□□□□□□□

`sites.yaml` □□

□□□□

□□□□□□□□

□□□□□□

Algolia □□□□

Sitemap □□

□□

□□"□□□□□□"□□

□□□□

□□□

□□

□□

Markdown

Callouts

Mermaid

MDX

□□ MDX □□□□□□□□□□□□□□□□□□□□□

rspress □□

doom □□

□□□□□□□□

□□□

□□□□□□□□□□□□□□□□

`i18n.json`

`.ts/.tsx`

`.mdx`

API □□

□□ API

CRD

□□□□

□□ openapi □□

□□□□□□□

`props`

□□

□□□□

□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□ ACP □□

□□□□□

□□□□□

□□□□□□

□□□□□□□

□□

□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

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

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

npm   yarn   pnpm   bun

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

□□□□□□□□□□□□□□□□:

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

在 `package.json` □□□□□□□□:

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

□□□□□□□□□□ `doom.config.yml`:

```yml
title: My Docs
```

□□□□ `tsconfig.json` □□□□□□

```
{
  "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
  -a, --algolia                    Whether to enable Algolia search (default:
  -S, --site-url                   Whether to enable the siteUrl for sitemap g
  -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/**`
  lint [root]                    Lint the documentation
  help [command]                 display help for command
```

命令行帮助信息输出

□□□□□□

使用 `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
```

□□□□□□

使用 `yarn build` □□□□□□□□□□□□□□□□□□ `dist` □□□□□□□□□□

□□□□

使用 `yarn serve` □□□□□□□□□□□□□□□□□□□□ `-b`, `-p` □□□□□□□□□□□□□□□□ `-b`, `-p` □
□

□□□□□□□

使用 `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 (defau
  -t, --target <language>  Document target language, one of en, zh, ru (defau
  -g, --glob <path...>     Glob patterns for source dirs/files
  -C, --copy [boolean]     Wether to copy relative assets to the target direc
  -h, --help               display help for command
```

- `-g, --glob` 　□□□□□□□□□□□□□□□□□□□□□□□□ `glob` □□

  1. `yarn translate -g abc xyz` □□□□ `<root>/<source>/abc`, `<root>/<source>/xyz`
     □□□□□□□□□□□ `<root>/<target>/abc`, `<root>/<target>/xyz` □□□

  2. `yarn translate -g '*'` □□□□ `<root>/<source>` □□□□□□□□□

- `-C, --copy` □□ `false` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

  - □□□□□□□

    1. `/<source>/abc.jpg` □□□□□□ `<root>/public/<source>/abc.jpg` □
       `<root>/public/<target>/abc.jpg`□□□□□□□□□□□□ `/<target>/abc.jpg`

    2. `<root>/<source>/abc.mdx` □□□□ `./assets/xyz.jpg` □□□□□□
       `<root>/<source>/assets/xyz.jpg` □ `<root>/<target>/assets/xyz.jpg`□□□□
       □□□□□□□

    3. `<root>/<source>/abc.mdx` □□□□ `./assets/<source>/xyz.jpg` □□□□□□
       `<root>/<source>/assets/<source>/xyz.jpg` □
       `<root>/<target>/assets/<target>/xyz.jpg`□□□□□□□□□□□□□
       `./assets/<target>/xyz.jpg`

- 对于相对路径资源：

  1. `/<source>/abc.jpg` 会被解析为 `<root>/public/<target>/abc.jpg`，并在生成的文件中
     替换为对应的 `/<target>/abc.jpg`，以确保正确的引用关系。

  2. `<root>/<source>/abc.mdx` 中引用的 `./assets/<source>/xyz.jpg` 会被解析为
     `<root>/<target>/assets/<target>/xyz.jpg`，并在生成的文件中替换为
     `./assets/<target>/xyz.jpg` 或对应的相对路径
     `../<source>/assets/<target>/xyz.jpg`

> **WARNING**
>
> 如果你在使用 `-g '*'` 的方式匹配所有文件时，`source` 和 `target` 的值不要使用默认的
> `internalRoutes`，否则会导致 `target` 目录下的文件被错误处理。

> **TIP**
>
> 如果你还没有设置 `AZURE_OPENAI_API_KEY`，请参考获取密钥的相关说明向 Leader 申请。

对于多语言站点，你还可以这样配置：

```
i18n:
  title:
    en: DevOps Connectors
  additionalPrompts: '这是关于 Connectors 的配置说明内容'
  disableAutoTranslation: false
title: DevOps 连接器
```

更多详细信息请参考文档说明

生成 PDF

> **WARNING**
>
> 此功能目前仅支持在执行 `yarn build` 时生效。

```
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
```

执行 `yarn export` 命令会将 PDF 文档导出到当前目录的 `-b`, `-p` 目录下，如果需要指定目录可以使用 `-b`, `-p` 参数

该功能依赖于 `playwright ↗`，我们已经将其 `build-harbor.alauda.cn/frontend/playwright-runner:doom` 镜像中预置了相关依赖环境 请确保你的环境可以访问该镜像。

```
.env.yarn

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

国际化

```
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
```

□□□□□□□□□□□□□

□□

□□

□□ `doom` □□□□

□□□□

□□□□

API □□□□

□□□□□□□□

□□□□□□

□□□□□□

□□□□□

□□□□□□□

□□□□□□□□

□□□□□□□

`sites.yaml` □□

□□□□

□□□□□□□□

□□□□□□

Algolia □□□□

Sitemap □□

□□

□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□

□□

□□

Markdown

Callouts

Mermaid

MDX

使用 MDX 如何在文档中编写代码、组件等内容

rspress 组件

doom 组件

业务组件的封装

国际化

如何切换不同语言的文档内容

`i18n.json`

`.ts/.tsx`

`.mdx`

API 文档

生成 API

CRD

资源列表

导入 openapi 文档

可视化组件库

`props`

示例

□□□□

□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□ ACP □□

□□□□□

□□□□□

□□□□□□

□□□□□□□□

□□

□□

□□□□

□□□□□□□□□□□□□□□□ `yaml` □□□□□□□□□□ `doom.config.yaml` □ `doom.config.yml` □□□□□□□□□□□□□□□□□□□□□□ `rspress` □□□□□□□□□□ `js/ts` □□□□□□□□ `.js/.ts/.mjs/.mts/.cjs/.cts` □□□□□□□□

一个 `js/ts` 文件（推荐使用后者），从 `@alauda/doom/config` 中导出 `defineConfig` 以 获取更好的类型提示：

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

export default defineConfig({})
```

基础配置

- `lang` 配置站点的默认语言，影响文档的默认语言和一些内置组件的文案，默认为 `en`，可通过命令行参数覆盖此配置，但不能为 `null` 或 `undefined`

- `title` 配置站点的标题，影响浏览器标签页的标题

- `logo` 配置站点的 logo，影响导航栏和文档的标题，值为 `public` 目录下的相对路径或完整的外链地址，默认使用 `doom` 内置的 alauda logo

- `logoText` 配置站点的标题文案，展示在 logo 旁

- `icon` 配置站点 favicon，默认同 `logo`

- `base` 配置站点的基础路径，影响静态资源的加载，默认为 `product-docs` 可以为 `/`

- `outDir` 配置站点的输出目录，默认为 `dist/{base}/{version}`，配置后的输出目录为 `dist/{outDir}/{version}`，其中 `version` 相关请参考版本化章节

API 文档渲染

```yaml
api:
  # CRD □□□□□□□□□ doom.config.* □□□□□□□ glob □□□json/yaml □□
  crds:
    - docs/shared/crds/*.yaml
  # OpenAPI □□□□□□□□□ doom.config.* □□□□□□□ glob □□□json/yaml □□
  openapis:
    - docs/shared/openapis/*.json
  # □□ openapi □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  # □□ https://doom.alauda.cn/apis/references/CodeQuality.html#v1alpha1.CodeQ
  references:
    v1alpha1.CodeQualityBranch: /apis/references/CodeQualityBranch#v1alpha1.C
  # □□□API □□□□□□□□□□□□□□ gateway □□□□□□□□□□□□
  pathPrefix: /apis
```

□□□□□□□ API □□

□□□□□□□□□

```yaml
# □□□□□□□□□□□ doom.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.confi
        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.md

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

release-notes.mdx

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

其中， `template=fixed&project=DevOps` 表示以 `fixed` 为 `queryTemplates` 模板，其余的均作为
变量 `query` 传入 `project=DevOps` 用于在 `ejs` ↗ 模板中渲染。 `fixed` 模板的内容为 jira `jql` ↗ 语
句， https://jira.alauda.cn/rest/api/2/search?jql=<jql> 用于调用 API 获取数据，需要通过
`JIRA_USERNAME` 和 `JIRA_PASSWORD` 传入必要的认证信息。

□□□□□

```
sidebar:
  collapsed: false # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ false
```

□□□□□□□□

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

□□□□□□□□□

```
onlyIncludeRoutes: # □□□□□ glob □□□□□□ docs □□□□ cli □□ `-i, --ignore` □□□□□□
  - '*/internal/**'
internalRoutes:
  - '*/internal/overview.mdx'
```

□□□□□□□□

```
shiki:
  theme: # optional, https://shiki.style/themes
  langs: # optional, https://shiki.style/languages
  transformers: # optional, only available in js/ts config, https://shiki.sty
```

> **WARNING**
>
> □□□□□□□□□□□□□□□□□□□□ `plaintext` □□

`sites.yaml` □□

`sites.yaml` □□□
□□□□□

```
- name: connectors # □□□□□□
  base: /devops-connectors # □□□□□□□□
  version: v1.1 # □□□□□□□□□ ExternalSite/ExternalSiteLink □□□□□

  displayName: # □□□□□□□□□□□□□□□□□□□□□□□□□ name
    en: DevOps Connectors
    zh: DevOps □□□

  # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  # □□□□□□□□□□□□□□□□□□□□□□□□□□
  repo: https://github.com/AlaudaDevops/connectors-operator # □□□□□□□□□□□ gi
  image: devops/connectors-docs # □□□□□□□□□□□□□□□□□□□□
```

□□□□

```
translate:
  # □□□□□□ejs □□□□□□□□□ `sourceLang`, `targetLang`, `userPrompt` □ `additiona
  # □□ `sourceLang` □ `targetLang` □ `□□` □ `□□` □□□□□□
  #     `userPrompt` □□□□□□□□□□□□□□
  #     `additionalPrompts` □□□ `frontmatter.i18n` □□ `additionalPrompts` □□□
  # □□□□□□□□□□□□□□□□□□□□
  systemPrompt: |
## □□
□□□□□□□□□□□□□□□□□□□□□<%= targetLang %>□□□□□□□□□□□□□□□<%= sourceLang %>□□□<%

## □□
- □□□□□□□□□□□□<%= sourceLang %>□□□□□□□□□□□□□□□□□□□<%= targetLang %>□□□□□
- □□□□□ MDX □□□□□□□□□□□□□ MDX □□□□□□□□□□ jsx □□□□□□ <Overview />□□□□□□□□□□□
- □□□□□□□□□□□□□□□
- MDX □□□□□□□□□□□□□MDX □□□□□□□□□□□□□□□ MDX □□□□□□□□□□□
  - <Overview /> □□ Overview □□□□□□□□□
  - <Tab label="value">□□□□□□□□</Tab>□label □ key □□□□□"value" □□□□□□□□
<%= terms %>
- □□□□□□□□□□□□□□□□□□□□□□
  - {/* release-notes-for-bugs */}
  - <!-- release-notes-for-bugs -->
- □□□□□□□□□□□□□□□
  - {/* reference-start */}
  - {/* reference-end */}
  - <!-- reference-start -->
  - <!-- reference-end -->
- □□□□□□□□□□□□ \\< □ \\{ □□□□□□□□□□□□
- □□□□□□□□□□□□ Markdown □□□□ frontmatter, □□□□□□□□□□□□ frontmatter.ii8n □□□□

## □□
□□□□□□□□□□
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
```

Algolia □□□□

```
algolia: # □□□Algolia □□□□□□□□□ `-a, --algolia` □□□□□□□□□
  appId: # Algolia □□ ID
  apiKey: # Algolia API Key
  indexName: # Algolia □□□□
```

□□□ `public/robots.txt` □□ Algolia □□□□

Sitemap □□

```
siteUrl: https://docs.alauda.cn # □□□□□ URL□□□□ sitemap□□□□ `-S, --site-url
```

□□

□□

□□□□

□□□

□□

□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□ `index` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `index.md` □ `index.mdx` □□

```
├── index.md
├── start.mdx
└── usage
    ├── index.mdx
    └── convention.md
```

□□□□□□

1. `public` □□□□□□□□□□□□□□□□□□□□□□□

2. `public/_remotes` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `*/public/_remotes` □□ `.gitignore` □□□□□□□□□□

3. `shared` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□ `frontmatter` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
---
title: □□
description: □□
author: □□
category: □□
---
```

□□□□□□□□□□ MDX □□ `.mdx` □□□□□□□□ `frontmatter` □□□□□□□□□□

□□

□ `index.md` □ `index.mdx` □□□□□□□□□□□□□□□□□□□□□□□□□ `frontmatter` □□ `weight` □□□□□□□□□□□□□□□□ `weight` □□□□□□□□□□□

```
---
weight: 1
---
```

> **WARNING**
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `index.mdx` □□□□□□ `Overview` □□□

```
# □□


<Overview />
```

□□□□□□□□□□□

Markdown

□□□□□ [gfm ↗](#) □□□□Doom □□□□□□□□□ Markdown □□□□□□

□□

Callouts

Mermaid

Callouts

□□□□□□

> **NOTE**
>
> 1. □□□□□□□□□□□□□□□□□ `;` , `%` , `#` , `//` , `/** */` , `--` □ `<!-- -->` □
>
> 2. □□□□□□□□□□□□□□□ `[\!code callout]` □□□□
>
> 3. □□□ `:::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) n
   - 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.

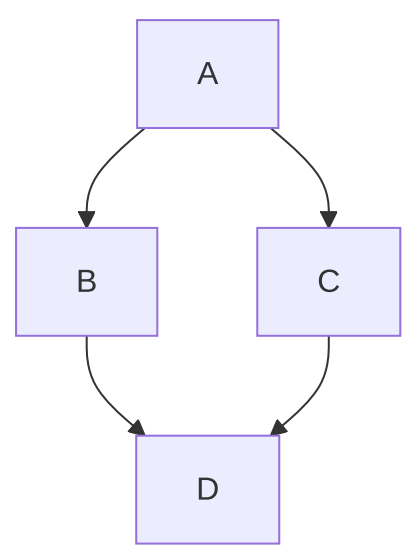- If Trusted Platform Module (TPM) is enabled, add 53 MiB.

🔲🔲🔲🔲🔲🔲🔲🔲🔲 Shiki Transformers ↗🔲

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` □□□□□□□[□□□↗](#)□□□□□□□□□□□□□ `.mdx` □□□□□□□□□□□□□□□□

- `Badge`
- `Card`

- `LinkCard`
- `PackageManagerTabs`
- `Steps`
- `Tab/Tabs`
- `Toc`

□□□□□□□□□□□□ `@rspress/core/theme` □□□□□□□□

> preview.mdx
>
> ```
> import { SourceCode } from '@rspress/core/theme'
>
> <SourceCode href="/" />
> ```

---

doom □□

`doom` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Overview

□□□□□□□□□□□□□□

Directive

□□□□□□□□□□□□□□□□↗□□□□□□□□□□□□ `Directive` □□□□

```
- □□□□□(`doc/en`)□□□□□□□□ `doc/zh` □□□□□□□□□□□□□□□□□□□□□□□□□□

  <Directive type="danger" title="□□">
    □□□□□□□□□□□□□□□□□□□□□□□□□
    `doc/zh` □□□□□□□□□□□□
  </Directive>
```

- 口口口口口(`doc/en`)口口口口口口口口 `doc/zh` 口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口
口口口口口

> 口口
>
> 口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口口 `doc/zh` 口口口口口口口口口口口口口口
口口

ExternalSite

口口口口口口口口

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

> Note
>
> 口口 DevOps 口口口口口口口口口口口口口口口口口口口口口口 DevOps 口口口口口口口口口口口口口口口口口口口口口口口
> [DevOps 口口口 ↗]口

ExternalSiteLink

口口口口口口口口口口

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

[Content ↗]

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

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

□□□□□□□□□□ `p` □□□□□□□□□□□□□□□ `children` □□□□

`AcpApisOverview` □ `ExternalApisOverview`

□□□□□□ API □□□□

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

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

> **Note**
>
> □□ ACP APIs □□□□□□□□□□□ ACP APIs □□ ↗□

> **Note**
>
> □□ DevOps □□□ APIs □□□□□□□□□□ DevOps □□□ APIs □□ ↗□

Term

□□□□□□□□□□□□□□□

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

□□□ □□□□□□□□ ACP

`props`

- `name` : □□□□□□□□□□□□□□□□□□□

- `textCase` : □□□□□□□□□□□□□ `lower` , `upper` , `capitalize`

TermsTable

□□□□□□□□□□□

```
<TermsTable />
```

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

props

- `terms` : `NormalizedTermItem[]` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

JsonViewer

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

yaml   json

```yaml
key: value
```

口口口口口口

口口口口口口口口口口口口口口口口口口口 `shared` 口口口口口口口口口口口口口口口口口口口

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

<CommonContent />
```

口口口口口口口口 runtime↗ 口口口 API口口口口口 `.jsx/.tsx` 口口口口口口口口口 `.mdx` 口口口口口口口

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

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

> **WARNING**
>
> 口口口口口 `.mdx` 口口口口口口口口 `props` 口口口口口口 issue↗口口口口口口口 `props` 口口口口口口 `.jsx/.tsx` 口口口口口口

国际化

`alauda` 对国际化做了深度定制和优化，当前仅支持 `en` / `zh` 两种语言，其实现是通过目录约定（即以 `public` 目录下的 `en` / `zh` 目录区分）和文件名后缀约定，下面将分别介绍其使用方式。

约定

`i18n.json`

`.ts/.tsx`

`.mdx`

---

i18n.json

该文件用于存放所有语言的翻译文本，在运行时会自动根据当前语言环境加载对应的翻译文本，`docs` 目录下的 `i18n.json` 会自动被加载，通过 `useI18n` 钩子来获取翻译文本，例如：

```
docs/i18n.json

{
  "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 □□

□□□□□□□□□□□□□□ API □□□□ API □ CRD (Custom Resource Definition) □□□□□□□□□□ □□□□□□

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

□□

□□ API

`props`

CRD

`props`

□□□□

`props`

□□ openapi □□

□□ API

advanced-apis/codeQualityTaskSummary.mdx

```
# CodeQualityTaskSummary

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

□□ [CodeQualityTaskSummary](#)□

props

- `path` : OpenAPI schema `paths` □□□□
- `pathPrefix` : □□□□□□□□□□□□ `api.pathPrefix`
- `openapiPath` : □□□□ [openapi](#) □□

CRD

```
crds/ArtifactCleanupRun.mdx

# ArtifactCleanupRun

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

□□ [ArtifactCleanupRun](#)□

props

- `name` : CRD `metadata.name`
- `crdPath` : □□□□ [openapi](#) □□□□□□□□□□□ CRD □□

□□□□

```
references/CodeQuality.mdx

# CodeQuality

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

□□ CodeQuality□

props

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

- `openapiPath` : □□□□ openapi □□

□□ openapi □□

□□ `OpenAPIPath` □ `OpenAPIRef` □□□□□□□□□□ openapi □□□□□□□□□□□□□□□□□□□□□□□ openapi □□□□□□□□ `openapiPath` □□□□□□

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

□□□□□□

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

□□

`props` □□

props

- `functions` : `string[]` - □□□□□□□□ `FunctionResource` □□□□□□

□□

□□	□□	□□□ □□	□□□ □□□	□□□ □□	□□□□ □□□	□□ □□	□□□ □□
	□□	✓	✓	✓	✓	✓	✕
□□□□ `devops-testplans`	□□	✓	✕	✓	✓	✓	✕
	□□	✓	✕	✓	✓	✓	✕
	□□	✓	✕	✓	✓	✓	✕
□□□□ `devops-`	□□	✓	✓	✓	✓	✓	✕

□□	□□	□□□ □□	□□□ □□□	□□□ □□	□□□□ □□□	□□ □□	□□□ □□
testmodules	□□	✓	✗	✓	✓	✓	✗
	□□	✓	✗	✓	✓	✓	✗
	□□	✓	✗	✓	✓	✓	✗

□□□□

□ Markdown □□□□

```
<!-- reference-start#name -->

<!-- reference-end -->
```

□ MDX □□□□

```
{/* reference-start#name */}

{/* reference-end */}
```

□□ `name` □□□□□□□□□□□□□[□□□□□]□□□ `<root>/public/_remotes/<name>` □□□□

□□□□□□ `<!-- reference-start#ref -->` □□□□□

□□

□□□□□□
`frontmatterMode`

□□□□□□

```
reference:
  - repo: alauda-public/product-doc-guide # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    branch: # [string] □□□□□□□□□□□□
    publicBase: # [string] □□□□□□□□□□□□□□□□□ /images/xx.png □□□□□□□□□□□□□□□□ do
    sources:
      - name: anchor # □□□□□□□□□□□□□□□□□□□□□□□
        path: docs/index.mdx#□□ # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ doom.confi
        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

□□□□□□□□□□□□□□□

□□

□□

□□□□□

□□□□□

□□□□□□

□□□□□□□□

□□□□□

□□□

```
doom build # □□□□□□
doom serve # □□□□□□□□□□□
```

□□□□□

□□□□□□ `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/unversio
```

□□□□□□

```
|— 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
```

index.html

```html
<!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>
```

□□□□□□□□

overrides.yaml

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

versions.yaml

```yaml
- '4.1'
- '4.0'
```