☰ Menu

□□□□

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 □□

□□□□□

□□□□□

□□□□□□

□□□□□□□□

Menu   □□□□ ›

□□

□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

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

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

 npm   yarn   pnpm   bun
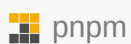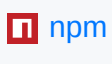
```
npm install -D @alauda/doom typescript
```

□□□□□□□□□□□□□□:

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

□ `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` :

```
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,
  },
}
```

然后添加 `global.d.ts` 文件，内容如下：

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

这将为你提供 `.mdx` 文件的类型声明以及 doom 运行时的全局类型。

配置项

```
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 [boolean|alauda] Whether to enable or use the alauda (docs.a
  -S, --site-url              Whether to enable the siteUrl for sitemap g
  -n, --no-open               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
```

全局选项及命令说明

□□□□□□

□□ `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
  --no-lazy                 Do not enable `lazyCompilation`
  -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 (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` 用于指定要翻译的源文档目录或文件，支持 `glob` 匹配模式，所有的匹配路径必须为相对于源文档目录的路径。例如：

  1. `yarn translate -g abc xyz` 会翻译 `<root>/<source>/abc` , `<root>/<source>/xyz` 目录下的文档并输出到 `<root>/<target>/abc` , `<root>/<target>/xyz` 目录下

  2. `yarn translate -g '*'` 会翻译 `<root>/<source>` 目录下的所有文档

- `-C, --copy` 用于指定是否将文档中相对引用的资源文件复制到目标目录下，默认为 `false` 。当此选项被启用时，复制行为如下：

  - 对于以下引用：

    1. `/<source>/abc.jpg` 会被解析为 `<root>/public/<source>/abc.jpg` 和 `<root>/public/<target>/abc.jpg` 的引用，并自动替换为 `/<target>/abc.jpg`

    2. `<root>/<source>/abc.mdx` 中引用的 `./assets/xyz.jpg` 会被解析为 `<root>/<source>/assets/xyz.jpg` 和 `<root>/<target>/assets/xyz.jpg` 的引用，并自动复制

    3. `<root>/<source>/abc.mdx` 中引用的 `./assets/<source>/xyz.jpg` 会被解析为 `<root>/<source>/assets/<source>/xyz.jpg` 和 `<root>/<target>/assets/<target>/xyz.jpg` 的引用，并自动替换为 `./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
```

运行 `yarn export` 即可导出 PDF 文档，它也支持传递 `-b` , `-p` 等参数，对应的参数同样为 `-b` , `-p` 等。

导出时会调用 `playwright ↗`，为了保证其与 `build-harbor.alauda.cn/frontend/playwright-runner:doom` 镜像中的浏览器版本一致，你可能 需要设置如下环境变量后再安装依赖：

.env.yarn

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

命令列表

```
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
```

更多文档信息请参阅官方文档。

☰ Menu

□□

□□

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

□□□□

□□□

□□

□□

Markdown

Callouts

Mermaid

MDX

□□ MDX □□□□□□□□□□□□□□□□□□□□

rspress □□

doom □□

□□□□□□□

□□□

□□□□□□□□□□□□□□□□

`i18n.json`

`.ts/.tsx`

`.mdx`

API □□

□□ API

CRD

□□□□

□□ openapi □□

□□□□□□

`props`

□□

□□□□

□□□□□

□□

□□□□□□□□□□□□□□□□ ACP □□

□□□□□

□□□□□

□□□□□□

□□□□□□□□

□□

□□

□□□□

□□□□□□□□□□□□□□□□ `yaml` □□□□□□□□□□ `doom.config.yaml` □ `doom.config.yml` □□□□□□□□□□□□□□□□□□□□ `rspress` □□□□□□□□ `js/ts` □□□

配置文件 `.js/.ts/.mjs/.mts/.cjs/.cts` 扩展名是支持的。

对于 `js/ts` 类型的配置文件，我们建议你使用来自 `@alauda/doom/config` 中导出的 `defineConfig` 来定义你的配置。

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

export default defineConfig({})
```

站点配置

- `lang` 配置当前文档的默认语言，支持中英文两种语言切换，默认为 `en`。注意当配置为多语言站点时，该配置应为 `null` 或 `undefined`
- `title` 配置当前文档的标题，主要用于首页等。
- `logo` 配置当前文档的 logo，支持本地文件和远程地址，本地文件应置于 `public` 目录下，具体参考静态资源处理，默认使用内置的 `doom` 图标，即 alauda logo
- `logoText` 配置当前文档的标题，展示于 logo 旁
- `icon` 配置为 favicon，默认同 `logo`
- `base` 配置当前文档的基础路径，默认为内置的 `product-docs`，不能为 `/`
- `outDir` 配置当前文档的输出目录，默认为 `dist/{base}/{version}`，自定义时输出目录则为 `dist/{outDir}/{version}`，其中 `version` 详见多版本部署

API 文档生成

```
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** □□

□□□□□□□□

```
# □□□□□□□□□□□ 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

□□□□□□□□□□□□

版本说明集成

```
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 */}
```

□□□ `template=fixed&project=DevOps` □□□ `fixed` □ `queryTemplates` □□□□□□□□□□□□□ `query` □□ `project=DevOps` □□□ `ejs ↗` □□□□□□□□ `fixed` □□□□□□□□□ jira `jql ↗` □□ `https://jira.alauda.cn/rest/api/2/search?jql=<jql>` □□□□ API □□□□□□□□□ `JIRA_USERNAME` □ `JIRA_PASSWORD` □□□□□□□□□□□

□□□□□

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

□□□□□□□□□

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

□□□□□□□□□□

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

□□□□□□□□

```yaml
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` □□
□□□□

```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 # □□□□□□□□□□□□□□□□□□
```

□□□□

```yaml
translate:
  # □□□□□□ejs □□□□□□□□□ `sourceLang`, `targetLang`, `userPrompt`, `additional
  # □□ `sourceLang` □ `targetLang` □ `□□` □ `□□` □□□□□□
  #     `userPrompt` □□□□□□□□□□□□□□
  #     `additionalPrompts` □□□ `frontmatter.i18n` □□ `additionalPrompts` □□□
  #     `terms` □ `titleTranslationPrompt` □□□□□□□□□□□□□□□□□□□□□□□ AI □□□
  # □□□□□□□□□□□□□□□□□□□□□
  systemPrompt: |
    You are a professional technical documentation engineer, skilled in writing h

    ## Baseline Requirements
    - Sentences should be fluent and conform to the expression habits of the <%=
    - Input format is MDX; output format must also retain the original MDX format
    - **CRITICAL**: Do not translate or modify ANY link content in the document.
      - URLs in markdown links: [text](URL) - keep URL exactly as is
      - Reference-style links: [text][ref] and [ref]: URL - keep both ref and URL
      - Inline URLs: https://example.com - keep completely unchanged
      - Image links: ![alt](src) - keep src unchanged, but alt text can be transl
      - Anchor links: [text](#anchor) - keep #anchor unchanged
      - Any href attributes in HTML tags - keep unchanged
    - Do not translate professional technical terms and proper nouns, including b
    - The title field and description field in frontmatter should be translated,
    - Content within MDX components needs to be translated, whereas MDX component
    - Do not modify or translate any placeholders in the format of __ANCHOR_N__ (
    - Keep original escape characters like backslash, angle brackets, etc. unchan
    - Do not add any escape characters to special characters like [], (), {}, etc
      - If source has "Architecture [Optional]", keep it as "Architecture [Option
      - If source has "Function (param)", keep it as "Function (param)" (not "Fun
      - Only add escape characters if they were present in the original text
    - Preserve and do not translate the following comments, nor modify their cont
      - {/* release-notes-for-bugs */}
      - <!-- release-notes-for-bugs -->
    - Remove and do not retain the following comments:
      - {/* reference-start */}
      - {/* reference-end */}
      - <!-- reference-start -->
      - <!-- reference-end -->
    - Ensure the original Markdown format remains intact during translation, such
    - Do not translate the content of the code block.
    <% if (titleTranslationPrompt) { %>
    <%- titleTranslationPrompt %>
    <% } %>
    <% if (terms) { %>
```

```
<%- terms %>
<% } %>


<% if (userPrompt || additionalPrompts) { %>
## Additional Requirements
These are additional requirements for the translation. They should be met alo

The text for translation is provided below, within triple quotes:
"""
<% if (userPrompt) { %>
<%- userPrompt %>
<% } %>


<% if (additionalPrompts) { %>
<%- additionalPrompts %>
<% } %>
"""
<% } %>
```

□□□□□□□□□

```
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 □□□□

> **INFO**
>
> □□ `rspress` □□□□□□□□□□ Algolia □□□□□□□□□[□□□□□↗] □□□□□□□□□□□□□□□□□□□□□□□□□□□
> `@alauda/doom/theme` □□□□□□□□□□□□□□□□□□□□□
>
> ```
> export * from '@alauda/doom/theme'
> ```

Sitemap □□

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

☰ Menu

□□

□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `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 />
```

□□□□□□□□□□□□

```
# □□


<Overview />
```

☰ Menu

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.
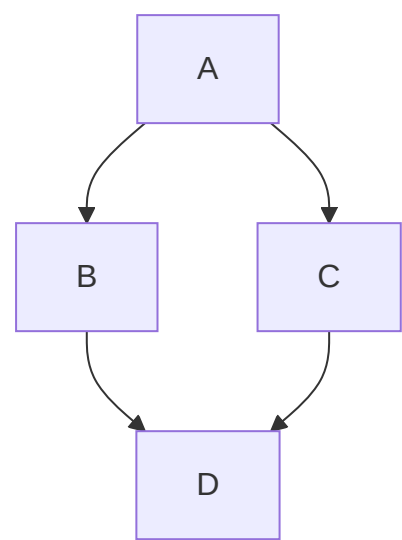
:::

```
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 □□□□□□

Menu   🔲🔲🔲 ›

MDX

MDX↗ 是一种 Markdown 的扩展语法，允许在 Markdown 中使用 JSX 语法。更多信息，请参考 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>
```

- 英文文档（ `doc/en` ）有变更时，会根据 `doc/zh` 中是否已有对应的翻译文档来判断是否需要创建或更新翻译文档。

> □□
>
> 如果没有对应翻译文档，会创建一个新的翻译文档；如果有对应翻译文档，会覆盖 `doc/zh` 中的对应翻译文档内容。

ExternalSite

用于嵌入外部站点

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

> Note
>
> 这是 DevOps 的通用组件，具体的使用方式请参考 DevOps 的文档。有关组件的更多信息，请参考 [DevOps 文档 ↗]。

ExternalSiteLink

用于链接到外部站点

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

[Content ↗]

> TIP
>
> 在 mdx 中 `<ExternalSiteLink name="connectors" href="link" children="Content" />` 会被解析成一个链接。

```
  <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

```
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` □□□□□□

☰ Menu

国际化 ›

国际化

`alauda` 支持多种形式的国际化配置。目前默认支持的语言为 `en` / `zh` 两种语言，默认语言为英文。本网站内的 `public` 目录下也有 `en` / `zh` 两种语言的配置，可以根据实际情况进行调整，以适配不同的语言环境。

概念

`i18n.json`

`.ts/.tsx`

`.mdx`

---

i18n.json

对于本框架自身的国际化配置，我们使用了一种简单的配置方式，在 `docs` 目录下新建 `i18n.json` 文件，其中包含了 `useI18n` 来实现配置国际化的一种方案。

```
docs/i18n.json

{
  "title": {
    "zh": "□□",
    "en": "Title"
  },
  "description": {
    "zh": "□□",
    "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')}
```

☰ Menu □□□□ ›

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
```

查看 CodeQualityTaskSummary。

props

- `path` : OpenAPI schema `paths` 中的路径
- `pathPrefix` : 路径前缀，一般用于覆盖 `api.pathPrefix`
- `openapiPath` : 用于指定 openapi 文件

CRD

crds/ArtifactCleanupRun.mdx

```
# ArtifactCleanupRun

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

查看 ArtifactCleanupRun。

props

- `name` : CRD `metadata.name`
- `crdPath` : 用于指定 openapi 文件，该文件中应包含 CRD 信息

属性配置

> 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`
	□□	✓	✓	✓	✓	✓	✕

	□□	✓	✕	✓	✓	✓	✕
	□□	✓	✕	✓	✓	✓	✕
	□□	✓	✕	✓	✓	✓	✕

☰ Menu

□□□□ ›

□□□□

□ 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

□□□□□□□□□□□□□

☰ Menu

□□□□ ›

□□

□□

□□□□□

□□□

```
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'
```