

[matrix]

发 开 基 博 常 矩 店 现在
现 发 础 客 见 阵 铺 试试

文档 规格 API游乐场 代码

自托管地图切片服务器以进行位置共享

多亏了 [MSC3488](#)，共享您的位置的能力即将提供给各种 Matrix 客户端，包括 FluffyChat（已经有一段时间了）和 Element。

当我们共享有关我们位置的信息时，我们需要格外小心，不要将其保密。因为 Matrix 提供了强大的、经过公开审核的端到端加密，所以当我们共享一个位置时，我们可以确信我们发送的消息只会被房间里的人阅读。

但是，除了我们发送的消息之外，我们还需要考虑我们的信息可能泄露的其他方式。特别是，当我们在屏幕上绘制地图时，我们需要与额外的服务器通信以获取地图信息，而我们正在寻找的地图部分（称为“图块”）可以显示我们的大致位置，如果他们得到落入坏人之手。

Element 正在为 Element 客户端默认使用的磁贴服务器的托管提供资金，并且不会与您的主服务器共享信息，这意味着查看正在请求哪些磁贴的人和拥有访问元数据，例如 Matrix 活动的模式。

但 Matrix 旨在让您控制自己的信息和基础设施，因此这里有一个指南，可帮助您在 Element 客户端中享受功能齐全的位置共享，同时确保所有磁贴请求都发送到您控制的服务器。

我们将描述如何运行 [OSM 瓦片服务器堆栈](#)，以及如何添加 config 元素需要使用它来显示地图。

🔗 未准备好生产

注意：这不是关于如何为许多用户托管强大的、可用于生产的地图切片服务器的指南。要做到这一点，您将需要认真的硬件，并进一步研究如何在您自己的托管环境中提供高可用性、监控以及您需要的一切。

本指南应该为您提供在本地运行切片服务器所需的信息，以便您可以开始学习在生产中启动切片服务器所需的内容。

（有关生产就绪所需的更多信息，请参阅 [Self-Host a Matrix Map Server](#)，其中包括一些 ansible 脚本和硬件要求。）

🔗 先决条件

本指南基于 [Docker](#) 构建，因此您需要一个支持 Docker 的环境或兼容的容器引擎。

[matrix]

发 开 基 博 常 矩 店 现在
现 发 础 客 问 阵 铺 试试

运行瓦片服务器

[OSM 瓦片服务器堆栈](#)是一个地图瓦片服务器，它通过 HTTP 提供瓦片，并提供一个简单的 Web UI 来浏览地图数据：



下载原始数据

为了运行瓦片服务器，我们需要一些地图数据，然后我们将处理这些数据并将其作为地图瓦片提供。（图块是地图特定区域的图像，而原始数据包含道路、建筑物等的确切位置）

您可以从download.geofabrik.de下载地图数据（PBF 文件）。例如，要下载英国的数据，请运行以下命令：

```
wget 'https://download.geofabrik.de/europe/great-britain-latest.osm'
```

导入它

获得数据后，我们需要将其导入到切片服务器中。运行这些命令：

（更改“英国”以匹配您下载的内容。）

```
docker volume create openstreetmap-data  
docker volume create openstreetmap-rendered-tiles  
docker run \
```

[matrix]

发 开 基 博 常 矩 店 现在
现 发 础 客 问 阵 铺 试试

这期望下载的 PFB 文件位于当前目录中，并将导入的数据保存到前两个命令创建的两个 docker 卷中。

导入过程可能需要相当长的时间。在相当强大的 2022 笔记本电脑上，英国数据需要 39 分钟才能导入。

如果您将切片服务器作为服务运行，您可能希望自动执行下载和导入新数据的过程。更新应该包括从“下载原始数据”重新运行下载命令，然后重新运行 `import` 上面的命令。

运行磁贴服务器

导入数据后，您可以像这样运行服务器：

```
docker run \  
  -p 8080:80 \  
  -v openstreetmap-data:/var/lib/postgresql/12/main \  
  -v openstreetmap-rendered-tiles:/var/lib/mod_tile \  
  -e THREADS=24 \  
  -e ALLOW_CORS=enabled \  
  -d overv/openstreetmap-tile-server:1.3.10 \  
run
```

这应该会在后台启动 docker 容器，您可以使用 `docker ps`。

测试一下

服务器运行后，您可以通过访问 <http://127.0.0.1:8080/tile/0/0/0.png> 来获取单个文件，或者在 <http://127.0.0.1> 与 Web UI 交互：`8080`。

当您第一次启动该服务时，它会占用大量 CPU 并且非常迟钝，但几分钟后它应该会相当顺利地运行。

提供客户端需要的 JSON 文件

一旦 tile 服务器运行，我们需要提供配置以便 Element（和任何类似的客户端）可以使用它。

Element 使用 [MapLibre GL](#) 库来渲染地图，而 MapLibre 需要一个样式文件来描述如何查找地图图块。这是它的外观：

Element Location Sharing demo

[matrix]

发 开 基 博 常 矩 店 现在
 现 发 础 客 问 阵 铺 试试

样式文件非常简单。为了描述这台机器上的瓦片服务器，我们需要创建一个 `style.json` 像这样的文件：

```
{
  "version": 8,
  "sources": {
    "localsource": {
      "type": "raster",
      "tiles": [
        "http://127.0.0.1:8080/tile/{z}/{x}/{y}.png"
      ],
      "tileSize": 256,
      "attribution": "Maps Copyright 2022 &lt;a href=\"http://www.ge"
    }
  },
  "layers": [
    {
      "id": "locallayer",
      "source": "localsource",
      "type": "raster"
    }
  ]
}
```

我们需要这个文件由瓦片服务器提供，但这很容易通过稍微修改我们用来运行它的命令来完成。

首先停止你的瓷砖服务器。用于 `docker ps` 查找您的 tile 服务器的 CONTAINER ID，然后使用 `docker kill <CID>`（替换为 CONTAINER ID 而不是 `<CID>`）停止它。

现在重新启动 tile 服务器，要求它也提供我们的样式文件：

[matrix]

发 开 基 博 常 矩 店 现在
现 发 础 客 问 阵 铺 试试

```
-e ALLOW_CORS=enabled \  
-d overv/openstreetmap-tile-server:1.3.10 \  
run
```

(注意提到的额外行 `style.json`。这假设 `style.json` 在当前目录中。)

现在，如果我编辑 Element 客户端的配置以指向该样式文件，它将使用我们的图块服务器来显示其地图。

`config.json` 如果您是自托管 Element Web，则可以将其添加到其文件的顶层：

```
{  
  ... other config ...  
  "map_style_url": "http://127.0.0.1:8080/style.json",  
  ... other config ...  
}
```

此外，如果您运行家庭服务器，Element 客户端将更喜欢您在您所在 `.well-known` 区域指定的切片服务器，而不是其自己的配置设置。您应该从服务器根目录提供一个文件，该文件 `/.well-known/matrix/client` 包含如下信息：

```
{  
  ... other info ...  
  "m.tile_server": {  
    "map_style_url": "http://mytileserver.example.com/style.json"  
  }  
  ... other info ...  
}
```

一旦你有了这些设置之一，你的 Element 客户端应该在每次显示地图时使用你自己的图块服务器！

🔗 进一步阅读和致谢

julian 在此基础上发布了一个出色的指南，其中包括 ansible 脚本和对所需硬件的一些评论：

- [自托管矩阵地图服务器](#)

本指南是通过阅读以下链接构建的。感谢他们的作者：

- [Switch2OSM - 基础知识](#)- 一些概念的解释

[matrix]

发 开 基 博 常 矩 店 现在
现 发 础 客 见 阵 铺 试试

- [位置保护 Firefox 插件](#)
 - 允许设置硬编码位置而不是共享您的真实位置。
- [在 Libshumate 中实现矢量平铺支持](#)
 - 詹姆斯韦斯特曼的一些关键思想的解释

内容

未准备好生产

先决条件

运行瓦片服务器

提供客户端需要的 JSON 文件

进一步阅读和致谢

其他指南

介绍

介绍

我怎样才能参与？

矩阵中的适度

应用服务

为 Matrix 徽章指南而生

矩阵工作室

使用开源通信工具运行在线活动

服务器设置

安装突触

自托管地图切片服务器以进行位置共享

使用 Oracle 云的免费 Matrix 服务器

加密

[matrix]

发现 开发 基础 博客 常见问题 矩阵直播 店铺 现在试试

从客户端服务器 API v1 迁移

客户端服务器 API

使用 Matrix 使 1960 年代的聊天机器人软件在 2018 年可用

支持 SSO 的客户端开发人员指南

创建一个简单的只读 Matrix 客户端

matrix-bot-sdk 的使用

matrix-bot-sdk 房间管理功能

matrix-nio (Python Matrix 客户端库) 的使用

桥接

桥接类型

桥接矩阵与在 VM 上运行的 WhatsApp

家庭服务器开发

绝望的非数学国家决议 v2

发现

试试矩阵

客户

机器人

SDK

托管

指南

入门

客户端-服务器 API

安装突触

桥梁

所有指南

开发文档

规格

API游乐场

代码

博客

所有帖子

本周在矩阵

安全

[matrix]

发现 开发 基础 博客 常见问题 矩阵直播 店铺 现在试试

[常见问题](#)

[矩阵直播](#)

[安全披露政策](#)

[Matrix.org 行为准则](#)

[合法的](#)

[接触](#)

[网站来源](#)



© 2021 Matrix.org 基金会 CIC