

Docker 本地安装 Minio 对象存储

 犬小哈 ·  2023年10月28日 ·  2,102字 ·  8分钟 ·  20张图 ·  9,578



公众号: 小哈学Java 网站: www.quanxiaoha.com

因为博客设置模块涉及到图片上传，如上传博客 LOGO 图片、作者头像，以及后续发布文章也需要支持图片上传。所以，我们首先需要搭建一个图床服务，这里小哈选型的是 Minio 对象存储，它不光可以存储图片，还能存储文件、视频等，非常强大。

1. 什么是 MinIO?

MinIO 是一个开源的对象存储服务器。这意味着它允许你在互联网上存储大量数据，比如文件、图片、视频等，而不需要依赖传统的文件系统。MinIO 的特点在于它非常灵活、易于使用，同时也非常强大，可以在你的应用程序中方便地集成。

2. 为什么使用 MinIO?

- **可伸缩性和性能：** MinIO 允许你在需要时轻松地扩展存储容量，无需中断服务。它具有出色的性能，可以处理大量的并发读取和写入请求。
- **开源和自由：** MinIO 是开源软件，遵循 Apache License 2.0 许可证，这意味着你可以自由地使用、修改和分发它。
- **容器化部署：** MinIO 提供了容器化部署的支持，可以在各种平台上快速部署和运行，包括本地开发机、云服务器和容器编排环境（如 Docker）。
- **兼容性：** MinIO 提供了 S3 兼容的 API，这意味着它可以与任何兼容 Amazon S3 的应用程序无缝集成，为你的应用程序提供强大的对象存储能力。
- **易用性：** MinIO 的配置和管理非常简单，它提供了直观的Web控制台和命令行工具，帮助你方便地管理存储桶和对象。

总的来说，MinIO 是一个灵活、高性能、易用且开源的对象存储解决方案，适用于各种规模的应用程序，特别是那些需要大规模数据存储和访问的项目。

3. Docker 搭建 Minio 服务

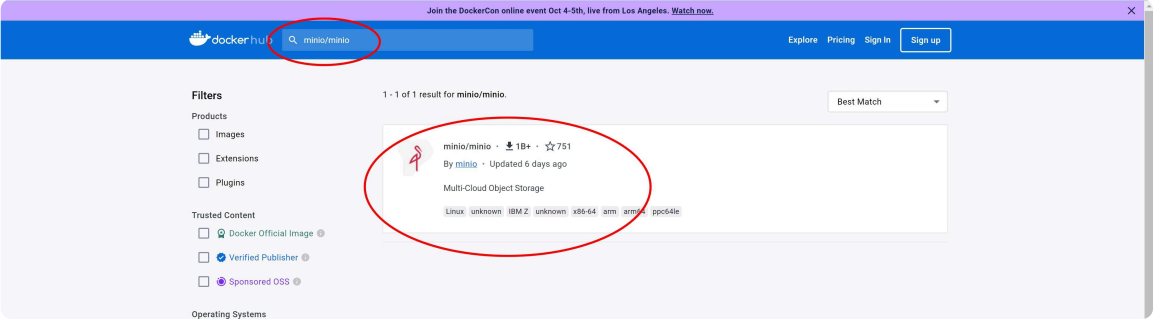


了解了 Minio 是什么后，接下来我们开始安装它。这里小哈使用的 Docker 来安装，更加简单一些。

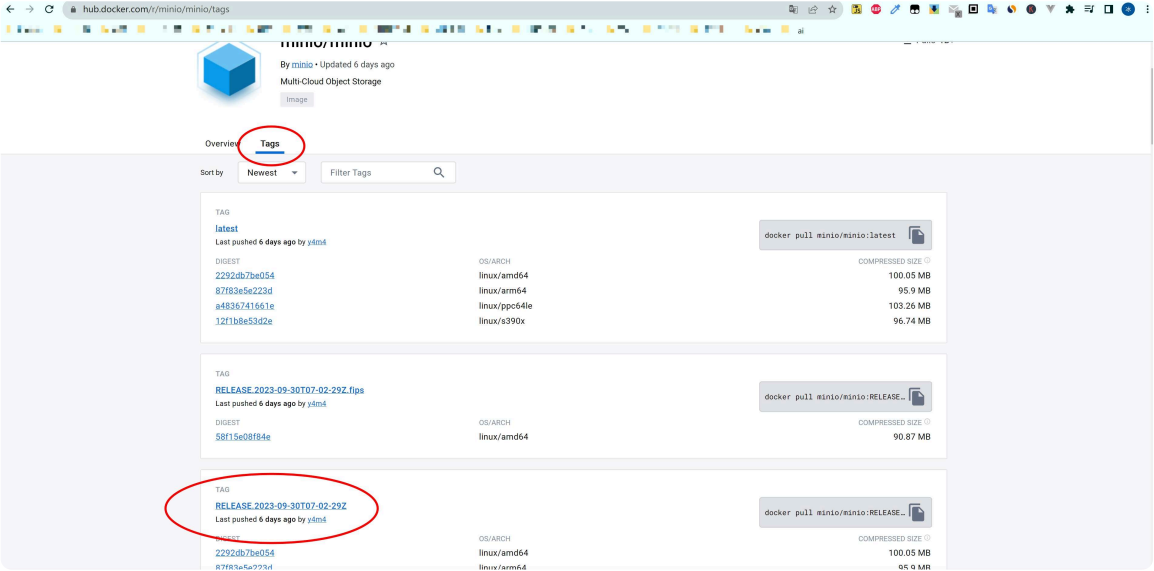
首先，你需要确保你的机器已经安装成功了 Docker，不清楚如何安装 Docker 的童鞋，可以翻阅前面的《后端环境安装》[📖](#) 小节。

3.1 选择一个 Minio 镜像

然后，我们在浏览器中访问地址：<https://hub.docker.com/> [🔗](#)，输入关键词 `minio/minio`，找到 Minio 镜像：



点击进去，点击 `Tags` 标签选项，小哈这里选择的是最新的一个发行版本：



3.2 下载 Minio 镜像

点击右侧复制命令，打开命令行，执行该命令拉取镜像：

```
docker pull minio/minio:RELEASE.2023-09-30T07-02-29Z
```

```
PS C:\Users\Administrator> docker pull minio/minio:RELEASE.2023-09-30T07-02-29Z
RELEASE.2023-09-30T07-02-29Z: Pulling from minio/minio
0cbafc6a7793: Already exists
0c06e955dc3b: Already exists
5bf3b024e1b0: Already exists
c3041f06b66f: Already exists
1ef03837ebc0: Already exists
9d74ffa4e082: Already exists
Digest: sha256:6262bc9a2730eeaf16be1bf436a3c2bca2ab76639f113778601a9f89c1485b56
Status: Downloaded newer image for minio/minio:RELEASE.2023-09-30T07-02-29Z
docker.io/minio/minio:RELEASE.2023-09-30T07-02-29Z
```

镜像下载成功后，执行 `docker images`，如果列表中有 `minio/minio` 镜像，则表示镜像下载成功了：

```
PS C:\Users\Administrator> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
minio/minio          RELEASE.2023-09-30T07-02-29Z  1330d18cbc2b       6 days ago         275MB
mysql                5.7                c20987f18b13       21 months ago      448MB
PS C:\Users\Administrator>
```

公众号: 小哈学Java 网站: www.quanxiaoha.com

3.3 新建数据挂载目录

下载镜像成功后，我们在某个盘下，小哈这里选择的是 E：盘，新建一个 /docker 文件夹，然后在该文件夹中再新建一个 /minio 文件夹，如下图所示：



新建该文件夹的目的是，后面通过镜像运行 Minio 容器时，可以将容器内的数据目录，挂载到宿主机的 E:\docker\minio 目录下，防止容器重启后，会导致数据丢失的问题。

3.4 运行 Docker Minio 容器

然后，通过该镜像运行 Minio 容器，命令如下：

```
docker run -d \
  -p 9000:9000 \
  -p 9090:9090 \
  --name minio \
  -v E:\docker\minio\data:/data \
  -e "MINIO_ROOT_USER=quanxiaoha" \
  -e "MINIO_ROOT_PASSWORD=quanxiaoha" \
  minio/minio:RELEASE.2023-09-30T07-02-29Z server /data --console-address ":9090"
```

注意，执行的时候需要将 \ 替换成空格，放到一行中来执行，最终命令如下：

```
docker run -d -p 9000:9000 -p 9090:9090 --name minio -v E:\docker\minio\data:/data -e "MINIO_ROOT_U:
```

解释一下上述命令各选项的含义：

- docker run：运行 Docker 容器的命令。
- -d：表示后台运行该容器；
- -p 9000:9000：将宿主机的 9000 端口映射到容器的 9000 端口。MinIO 默认的 HTTP API 端口是 9000。
- -p 9090:9090：将宿主机的 9090 端口映射到容器的 9090 端口。这是 MinIO 的 Web 控制台的端口。
- --name minio：给容器取了一个名字，这里是 "minio"。
- -v E:\docker\minio\data:/data：将主机上的 E:\docker\minio\data 目录映射到容器内的 /data 目录。

这是 MinIO 存储数据的地方。如果你希望数据在容器删除后仍然保存，可以将数据目录映射到宿主机。

- `-e "MINIO_ROOT_USER=quanxiaoha"`：设置 MinIO 的管理员用户名为 "quanxiaoha"。这是用于 MinIO Web 控制台和 API 的初始管理员用户名。
- `-e "MINIO_ROOT_PASSWORD=quanxiaoha"`：设置 MinIO 的管理员密码为 "quanxiaoha"。这是用于 MinIO Web 控制台和 API 的初始管理员密码。
- `minio/minio:RELEASE.2023-09-30T07-02-29Z`：这是 MinIO 的 Docker 镜像版本。
- `server /data --console-address ":9090"`：启动 MinIO 服务器，并将数据存储在容器内的 `/data` 目录。 `--console-address ":9090"` 表示 MinIO 的 Web 控制台将在容器的 9090 端口上运行。

```
PS C:\Users\Administrator> docker run -d -p 9000:9000 -p 9090:9090 --name minio -v E:\docker\minio\data:/data -e "MINIO_ROOT_USER=quanxiaoha" -e "MINIO_ROOT_PASSWORD=quanxiaoha" minio/minio:RELEASE.2023-09-30T07-02-29Z server /data --console-address ":9090"
4edfcf92dcb3ac74363dafc16bd978a85b81454f43f7d6e7cf0239521ef50768
PS C:\Users\Administrator>
```

公众号: 小哈学Java 网站: www.quanxiaoha.com

执行该命令后，再执行 `docker ps` 命令，可查看正在运行的容器，若如下图所示，容器列表中出现了 `minio`，则表示 `Minio` 后台运行成功了：

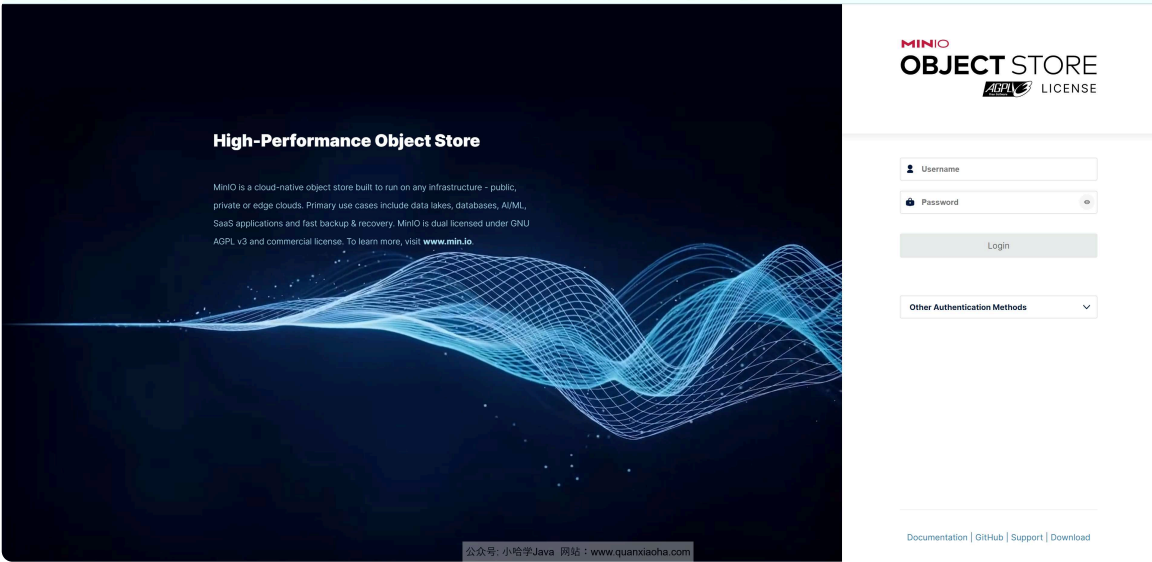
```
PS C:\Users\Administrator> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
NAME
4edfcf92dcb3   minio/minio:RELEASE.2023-09-30T07-02-29Z  "/usr/bin/docker-ent...  10 days ago   Up 5 hours   0.0.0.0:9000->9000/tcp, 0.0.0.0:9090->9090/tcp
4348c54bd89/   mysql:b. /                                docker-entrypoint.s...   3 weeks ago   Up 2 days    0.0.0.0:3306->3306/tcp, 33060/tcp
mysql
```

公众号: 小哈学Java 网站: www.quanxiaoha.com

TIP：如果 `Minio` 容器运行未成功，就需要通过日志来定位问题了，可以重新执行 `docker run` 命令，并将 `-d` 参数去掉，以前台的方式运行容器，即可看到启动日志了。

4. 访问 Minio 控制台

浏览器访问地址 `http://localhost:9090`，可访问 MinIO 的 Web 控制台：

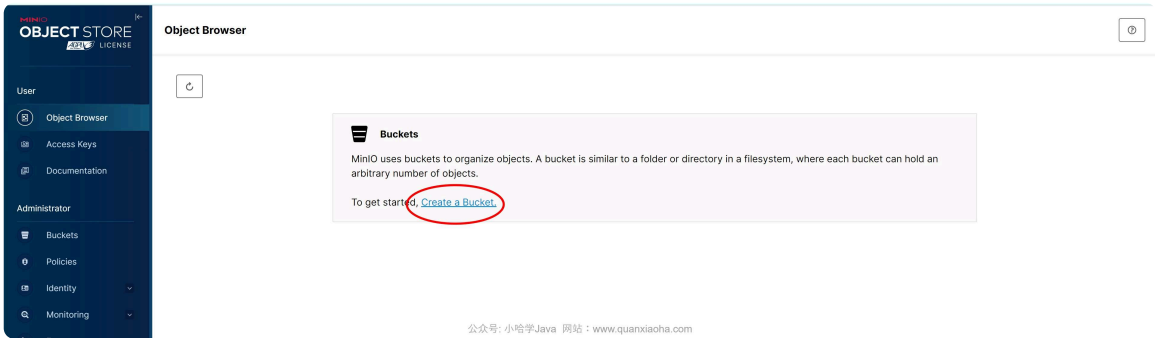


输入运行容器时，指定的用户名/密码：`quanxiaoha/quanxiaoha`，进入到 `Minio` 的管理后台：

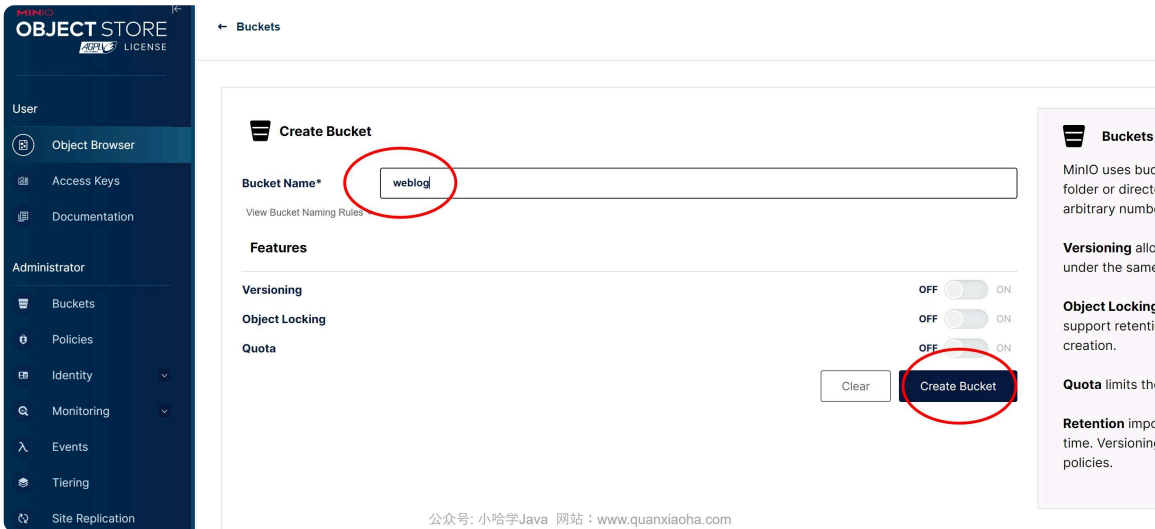


5. 新建一个 Bucket 桶

进入后台后, 点击 *Create a Bucket* 创建一个 *Bucket* 桶, 用于存储图片:



输入 Bucket Name, 我们将其命名为 *weblog*, 然后点击 *Create Bucket* 按钮:



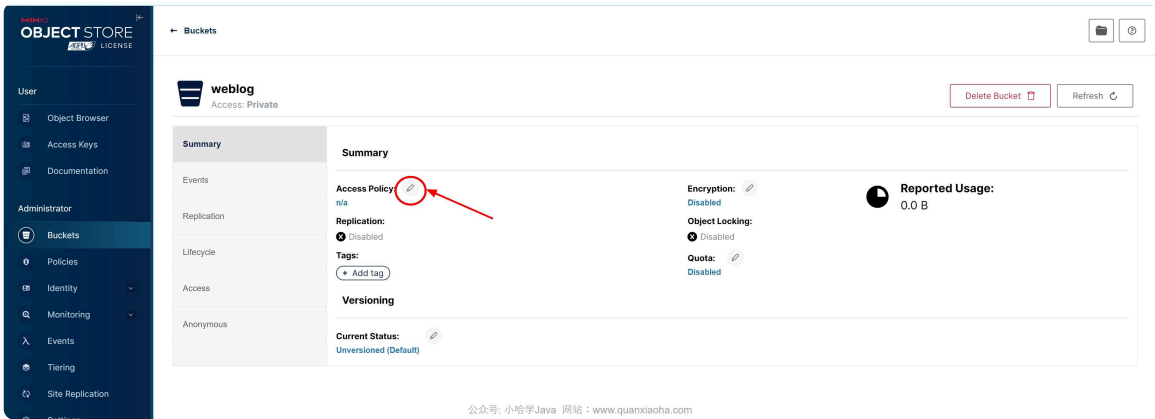
创建成功后, 在 Buckets 列表中就可以看到刚刚新建的桶了:





6. 设置 Bucket 为公共读

因为我们上传的图片需要被公网访问到，所以，还需要设置 Bucket 为公共读，默认为 Private 私有。点击想要设置的桶，然后编辑 Access Policy：

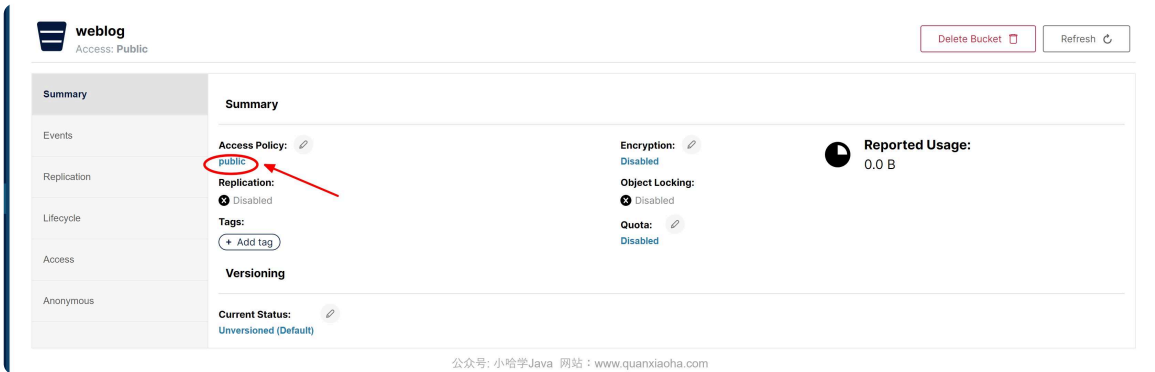


将 Access Policy 选项选择为 Public 公共读，点击 Set 设置按钮：

犬小哈实战专栏

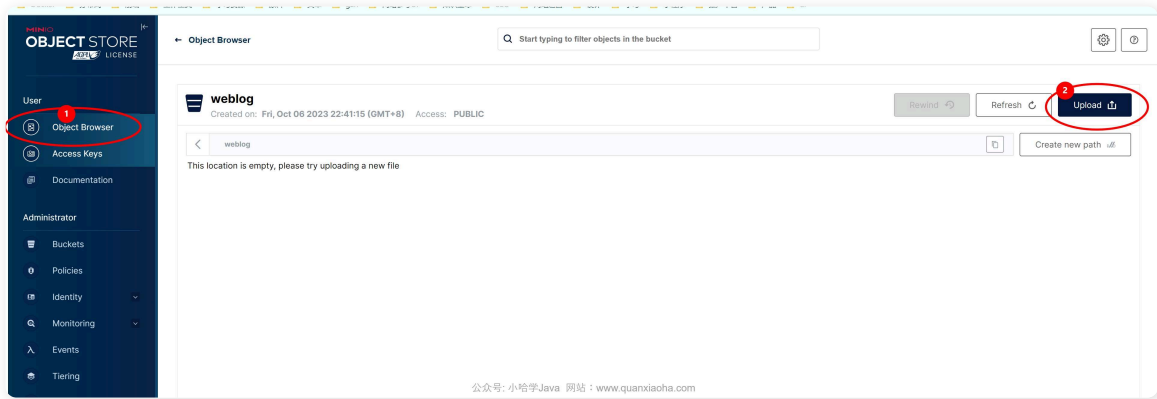


设置成功后，就可以看到 Access Policy 一栏变更为 Public 了，如下图所示：



7. 上传一张图片

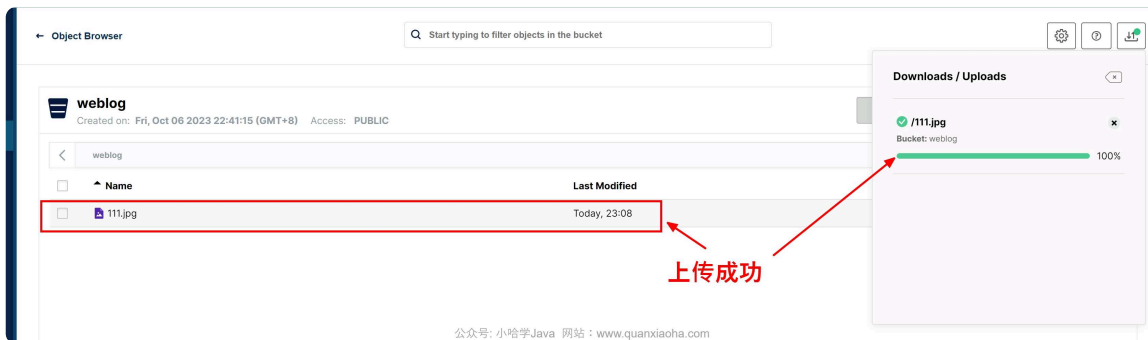
相关设置完成后，我们直接在后台上传一张本地的图片，测试看看能够正常上传成功。点击 *Object Browser -> Upload* 上传图片：



这里小哈选择了一张妹子图片来做测试：

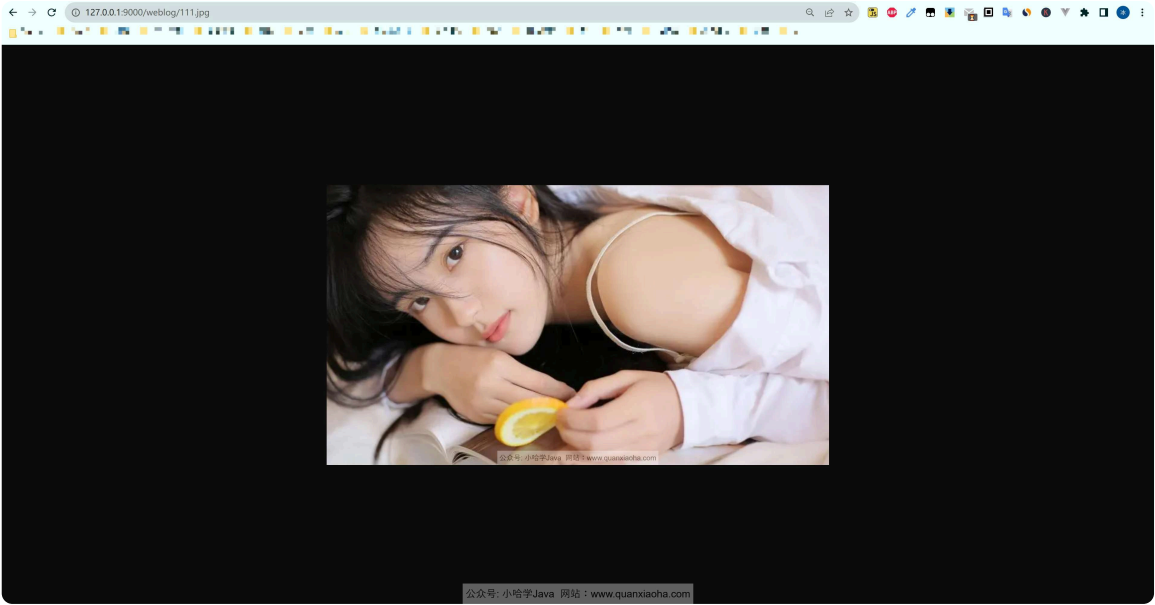


可以看到上传成功了：



接下来，我们直接在浏览器中，来访问该图片的直链：<http://127.0.0.1:9000/weblog/111.jpg>，看看能否被正常访问：

💡 TIP: 本地访问路径格式为 *请求地址:端口号 + 桶名称 + 图片的名称*，上线后会申请域名，格式为 *域名 + 桶名称 + 图片的名称*，例如 <https://img.quanxiaoha.com/weblog/111.jpg>。



OK，可以看到该图片能够被正常访问。到这里，本地的 **Minio** 对象存储服务就搭建好了，后续博客中的相关图片，都会上传到 **Minio** 中，然后，数据库中会直接存储图片的直链地址。

8. 结语

本小节中，小哈带着大家通过 Docker 容器，在本地环境中，将 Minio 对象存储服务搭建起来了，以作图床使用。这样，也可以方便的进行本地图片上传，当然，如果小伙伴们有服务器，也可以安装 Linux 服务器中来安装使用。在项目最终上线时，小哈会再次演示如何在 Linux 服务器中来安装它。

📅 发布于 2023年10月07日 | 08:59:18

🔔 **提示:** 点击下载小节源码时，若星球官方提示“无权限”，在**星球官方网站重新登录一下**即可。

◀ 上一节
9.1 博客设置模块功能分析、表设计

下一节 ▶
9.3 文件上传接口开发