

【harbor 学习笔记】docker push 接口详情

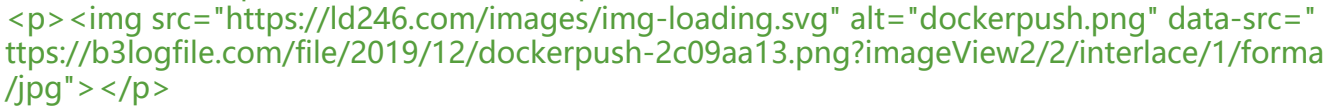
作者: [Archer1A](#)

原文链接: <https://ld246.com/article/1575706751094>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

docker push 接口详情



step 1. Api 版本确认

registry 服务器会通过 `/v2/` 接口根据返回的状态码来提供 api 的版本信息, 请求格式如下

```
GET /v2
```


如果返回 `200 OK` 则表明 registry 实现了 V2 接口, 并且客户端可以安全的去用其他所有的 `v2` 接口

如果返回 `401 Unauthorized`, 则表示客户端需要根据 `WWW-Authenticate` header 重新请求接口。根据访问控制设置, 用户即使这里通过验证, 在访问不同的资源的时候可能仍需身份验证

如果返回 `404 NOT FOUND` ,则客户端应认为 registry 未实现 v2 接口

step 2. 校验 layer 是否存在

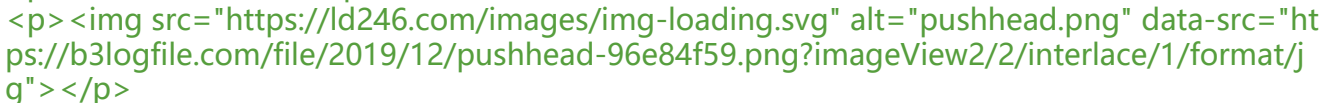
可以通过一个 `HEAD` 请求来校验一个 layer 是否存在。请求格式如下:

```
HEAD /v2/<name>/blobs/<digest>
```

如果返回的状态码为 `200 OK` ,则表示指定的 layer 已存在, 若已存在则 client 则跳过该 layer 的上传。由于 `HEAD` 请求根据 HTTP 规范没有 body,因此 HEADER 中带了以下信息

```
200
Content-Length: <length of blob>
Docker-Content-Digest: <digest>
```

以下为抓包的结果:



step 3. 发送 post 请求

该请求的参数是镜像命名空间, layer 将在该命名空间被链接。该请求的格式是:

```
POST /v2/<name>/blobs/uploads/
```

如果返回 `202 Accepted` 则表明请求成功, 其 HEADER 携带了以下信息:

```
202 Accepted
Location: /v2/<name>/blobs/uploads/<uuid>
Range: bytes=0-&offset&
Content-Length: 0
Docker-Upload-ID: <uuid>
```


nary Data>

</code></pre>

<p>抓包结果: </p>

<pre><code class="highlight-chroma">PATCH /v2/hello-world/blobs/uploads/87399c24-26ba-4f39-a1db-ca8b5469133d?_state_uog2S7ReISUOCxLS2uU3Z3S2h1_aV4Sq8T5CmA09B_p7Ik5hbWUiOiJoZWxsby13b3JsZCilIIVVUQiOi4NzM5OWMyNC0yNmJhLTRmMzktYTFkYi1jYThiNTQ2OTEzM2QiLCJmZmZmZXQjOjAsIl0YXJ0ZWRBdCI6IjwMTktMTEtMThUMTY6MzA6MzMUMTI4NzA5NTE1WiJ9 HTTP/1.1</code></pre>

Host: hub.vic.comUser-Agent: docker/17.12.0-ce go/go1.9.2 git-commit/c97c6d6 kernel/3.10.0-693.el7.x86_64 os/linux arch/amd64 UpstreamClient(Docker-Client/17.12.0-ce \((linux)\))

Content-Length: 77

Accept-Encoding: gzip

X-Forwarded-For: 172.17.0.1, 192.168.79.136

X-Forwarded-Prot: https

X-Real-Ip: 172.17.0.1

.....U.k.U.>.l.0.....cY..f2.IS U..q.T../.....].=.M|.!>....W|..

....,? D..'C`...lg....A....}.1s.F<.%<]:.w....

.....ng.J...w

~.....\Wn.&......G..w...5.'.9x.Lt.....=?...

hB...|..Q..x.~...0.l.?...+}.K..~.....K.....c..

.Y....w.../|.v..`..o)y.....*...Ea.....Mk...7...t.....?.....7&.....Z\.....m:].....K....

'.W...i.....7...J&x]..\...DBc..fC.:c.1.}).&.....L.F).\$.8.jW.C.R<x.u...C.r.#Ut.6....H....c96|M|.7.....L.....d...Od.:6C).q...._...X.8.."1...W.....Y.nt_wp.4.@qf.`..qQ.H.*..Y.q\$.U..)\... #.....|.... S.....,u.{Z=..gl.'.R.d..b m|.6...WD.x..U"R..O.>j.p.cl...._...n...5.!K....U.1..m:....np+.0+.>...|.&.l.6..db^..*

b96giX.6z.p.IC.]t0.....(.. " ...d.q.*.}|.%... \K...6..ZP...@W.CN+f...W.'.....}.....s,,,m.w.y.;;N_..cs.....r.....;L3..D....\{...!#...>{(pp....l....v.....Y....Y.._43..r.{...j..Z...2...Z...Y<.....5.b-..gsecv.....~.q.....k.....g.....

HTTP/1.1 202 Accepted

Content-Length: 0Docker-Distribution-Api-Version: registry/2.0

Docker-Upload-Id: 87399c24-26ba-4f39-a1db-ca8b5469133d

Location: https://hub.vic.com/v2/hello-world/blobs/uploads/87399c24-26ba-4f39-a1db-ca8b5469133d?_state=IoT88kcYkqkNriHRetc7JVpbPvK690THF_WixElh4J7Ik5hbWUiOiJoZWxsby13b3JsZCilIIVVVSUQiOi4NzM5OWMyNC0yNmJhLTRmMzktYTFkYi1jYThiNTQ2OTEzM2QiLCJmZmZmZXQjOjAsIl0YXJ0ZWRBdCI6IjwMTktMTEtMThUMTY6MzA6MzMUMTI4NzA5NTE1WiJ9 hcnRlZEF0ljoimjAxOS0xMS0xOFQxNjozMDozM1oifQ%3D%3D

```

Range: 0-976
X-Content-Type-
options: nosniff
Date: Mon, 18 No
2019 16:30:33 GMT

```

服务器必须顺序的接受每一个分块，但是不限制每个 layer 是如何分割的。服务器可能设置了一最小块的大小。如果服务器无法接收这个 chunk，则会返回 `416` 的状态码，并在 H ADER 中返回一个 `Range` 标识当前的状态

```

416 Requested Range Not Satisfiable
Location: /v2/&
ame&
;blobs/uploads/&
;uuid&
;
Range: 0-&
last
alid range&
;
Content-Length: 0
Docker-Upload-U
ID: &
;uuid&
;

```

如果返回的值为 `202 Accepted` 客户端需要从最后一个有效的 range 开始上后续的分块。

```

202 Accepted
Location: /v2/&
ame&
;blobs/uploads/&
;uuid&
;
Range: bytes=0-&
;offset&
;
Content-Length: 0
Docker-Upload-U
ID: &
;uuid&
;

```

step 5. 完成 yayer 上传

客户端发送一个 `PUT` 请求并携带 `digest` 参数来告诉服务器上传完成。如果发送该接口，上传将会被视为没有完成：

```

PUT /v2/&
;name&
;blobs/uploads/&
;uuid&
;?digest=&
;digest&
;
Content-Length: &
;size of chunk&
;
Content-Range: &
;start of range&
;-&
;end of range&
;
Content-Type: app
lication/octet-stream
&
;Last Layer Ch
nk Binary Data&
;

```

如果对应的 layer 已经存在，可以发送一个 Content-Length 为 0 的请求。当最后的 chunk 被收并验证通过将会返回 `302 Created` 状态：

```

201 Created
Location: /v2/&
ame&
;blobs/&
;digest&
;
Content-Length: 0

```



```
</span></span><span class="highlight-line"><span class="highlight-cl">Docker-Content-Digest: &lt;digest&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>HEADER 的 <code>Location</code> 将会携带一个可以验证当前接收的 layer 存在的地址。<code>Docker-Content-Digest</code> 将会提供当前 blob 的 digest,它可能会和提供的 digest 不
</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">PUT /v2/hello-world/blobs/uploads/83f294f5-2886-4d3b-bc9a-cc6c1ab93101?_state=EM
iJZM0HP1kEIDmi_luN3GbJ1JfEnn6XcYp_3FYILI7Ik5hbWUiOiJoZWxsby13b3JsZCIsIlVVSUQioil
M2YyOTRmNS0yODg2LTRkM2ltYmM5YS1jYzZjMWFiOTMxMDEiLCJZmZzZXQiOjE1MTAsIlN
YXJ0ZWRBdCI6IjwMTktMTEtMThUMTY6MzA6MzNaln0%3D&digest=sha256%3Afce289
99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc1741675f587e HTTP/1.1
</span></span><span class="highlight-line"><span class="highlight-cl">Host: hub.vic.com
</span></span><span class="highlight-line"><span class="highlight-cl">User-Agent: dock
r/17.12.0-ce go/go1.9.2 git-commit/c97c6d6 kernel/3.10.0-693.el7.x86_64 os/linux arch/amd6
UpstreamClient(Docker-Client/17.12.0-ce \(\linux\))
</span></span><span class="highlight-line"><span class="highlight-cl">Content-Length: 0
</span></span><span class="highlight-line"><span class="highlight-cl">Accept-Encoding:
gzip
</span></span><span class="highlight-line"><span class="highlight-cl">X-Forwarded-For:
172.17.0.1, 192.168.79.136
</span></span><span class="highlight-line"><span class="highlight-cl">X-Forwarded-Prot
: https
</span></span><span class="highlight-line"><span class="highlight-cl">X-Real-Ip: 172.17.0
1
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">HTTP/1.1 201 Crea
ed
</span></span><span class="highlight-line"><span class="highlight-cl">Content-Length: 0
</span></span><span class="highlight-line"><span class="highlight-cl">Docker-Content-D
igest: sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc1741675f587e
</span></span><span class="highlight-line"><span class="highlight-cl">Docker-Distributi
on-API-Version: registry/2.0
</span></span><span class="highlight-line"><span class="highlight-cl">Location: https://
ub.vic.com/v2/hello-world/blobs/sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c37247
c9235adc1741675f587e
</span></span><span class="highlight-line"><span class="highlight-cl">X-Content-Type-
ptions: nosniff
</span></span><span class="highlight-line"><span class="highlight-cl">Date: Mon, 18 No
2019 16:30:33 GMT
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<h2 id="step-6--push-镜像的manifest">step 6. push 镜像的 manifest</h2>
<p>当所有的 layer 都上传完成了, 客户端需要发送一个 <code>PUT</code> 请求并提供镜像的
anifest:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">PUT /v2/&lt;name&gt;/manifests/&lt;reference&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">Content-Type: &lt;
manifest media type&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "name": &lt;na
e&gt;;
</span></span><span class="highlight-line"><span class="highlight-cl">  "tag": &lt;tag&g
```

```

,,
</span></span><span class="highlight-line"><span class="highlight-cl"> "fsLayers": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "blobSum":
lt;digest>>
</span></span><span class="highlight-line"><span class="highlight-cl">   },
</span></span><span class="highlight-line"><span class="highlight-cl">   ...
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> ],
</span></span><span class="highlight-line"><span class="highlight-cl"> "history": &lt;v1
mages>&gt;,
</span></span><span class="highlight-line"><span class="highlight-cl"> "signature": &lt;
WS>&gt;,
</span></span><span class="highlight-line"><span class="highlight-cl"> ...
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>url 中的 <code>name</code> 和 <code>reference</code> 必须和下面的 body 中的字段
应。<code>reference</code> 可以是 <code>tag</code> 或者 <code>digest</code></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">PUT /v2/hello-world/manifests/1.0 HTTP/1.1
</span></span><span class="highlight-line"><span class="highlight-cl">Host: hub.vic.com
</span></span><span class="highlight-line"><span class="highlight-cl">User-Agent: dock
r/17.12.0-ce go/go1.9.2 git-commit/c97c6d6 kernel/3.10.0-693.el7.x86_64 os/linux arch/amd6
UpstreamClient(Docker-Client/17.12.0-ce \(\linux\))
</span></span><span class="highlight-line"><span class="highlight-cl">Content-Length:
24
</span></span><span class="highlight-line"><span class="highlight-cl">Accept-Encoding:
gzip
</span></span><span class="highlight-line"><span class="highlight-cl">Content-Type: app
ication/vnd.docker.distribution.manifest.v2+json
</span></span><span class="highlight-line"><span class="highlight-cl">X-Forwarded-For:
172.17.0.1, 192.168.79.136
</span></span><span class="highlight-line"><span class="highlight-cl">X-Forwarded-Prot
: https
</span></span><span class="highlight-line"><span class="highlight-cl">X-Real-Ip: 172.17.0
1
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "schemaVersion"
2,
</span></span><span class="highlight-line"><span class="highlight-cl">  "mediaType": "a
plication/vnd.docker.distribution.manifest.v2+json",
</span></span><span class="highlight-line"><span class="highlight-cl">  "config": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "mediaType":
application/vnd.docker.container.image.v1+json",
</span></span><span class="highlight-line"><span class="highlight-cl">    "size": 1510,
</span></span><span class="highlight-line"><span class="highlight-cl">    "digest": "sha
56:fce289e99eb9bca977dae136fbc2a82b6b7d4c372474c9235adc1741675f587e"
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "layers": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "mediaType"
"application/vnd.docker.image.rootfs.diff.tar.gzip",
</span></span><span class="highlight-line"><span class="highlight-cl">      "size": 977,

```

```
</span></span><span class="highlight-line"><span class="highlight-cl">    "digest": "sh
256:1b930d010525941c1d56ec53b97bd057a67ae1865eebf042686d2a2d18271ced"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl"></span></span>
</span></span><span class="highlight-line"><span class="highlight-cl">HTTP/1.1 201 Crea
ed
</span></span><span class="highlight-line"><span class="highlight-cl">Docker-Content-D
gest: sha256:92c7f9c92844bbbb5d0a101b22f7c2a7949e40f8ea90c8b3bc396879d95e899a
</span></span><span class="highlight-line"><span class="highlight-cl">Docker-Distributi
n-API-Version: registry/2.0
</span></span><span class="highlight-line"><span class="highlight-cl">Location: https://
ub.vic.com/v2/hello-world/manifests/sha256:92c7f9c92844bbbb5d0a101b22f7c2a7949e40f8
a90c8b3bc396879d95e899a
</span></span><span class="highlight-line"><span class="highlight-cl">X-Content-Type-
ptions: nosniff
</span></span><span class="highlight-line"><span class="highlight-cl">Date: Mon, 18 No
2019 16:30:33 GMT
</span></span><span class="highlight-line"><span class="highlight-cl">Content-Length: 0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>这个接口返回 <code>201</code> 后整个 push 流程就结束了~</p>
```