

گام اول

در قسمت اول، باید یک ایمج بر پایه لینوکس را از داکرهاب بگیریم و روی آن امکان استفاده از دستور `curl` را ایجاد کنیم. برای این کار ابتدا باید ایمج داکر ابونتو موردنظر خود را از داکرهاب بگیریم. برای این کار از دستور `pull` استفاده میکنیم:

```
docker pull ubuntu:focal
```

وقتی ایمج مورد نظر گرفته شد، با دستور زیر آن را اجرا میکنیم:

```
docker run -it --name ubuntu ubuntu:focal bash
```

در دستور بالا با استفاده از `-it` وقتی ابونتو بالا می آید، محیط اجرایی و امکان اجرای دستورات در ترمینال آن را به ما میدهد درحالیکه اگر فقط از دستور `docker run ubuntu:focal` استفاده کنیم، چون خود لینوکس کاری برای اجرا ندارد، کانتینر آن بالا می آید و اجرا شده و تمام میشود و امکان استفاده از ترمینال آن را به ما نمیدهد. در ترمینال ایمج لینوکس برای ایجاد امکان استفاده از دستور `curl` از دستور زیر استفاده میکنیم:

```
apt-get update && apt-get install curl
```

`apt-get` پکیج منجر لینوکس است که اول پکیج های ایمج را آپدیت میکنیم و بعد `curl` را نصب میکنیم. اگر در این حالت از ایمج `exit` کنیم، دستور `docker ps` خالی است چون ایمجی در حال اجرا نیست اما اگر دستور `docker ps -a` را بزنیم، خروجی به صورت زیر است:

```
PS D:\lectures\Cloud Computing\9731702_HW2\DockeizedProject> docker ps -a
```

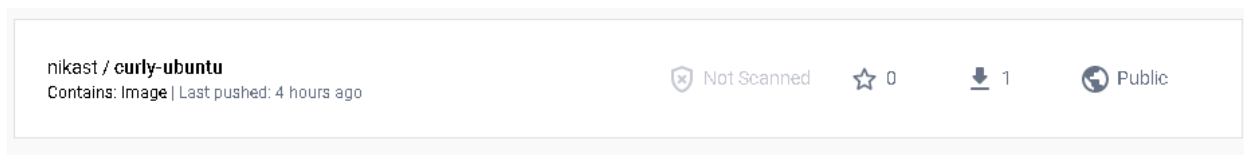
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
bf9eeb82b4a7	ubuntu:focal	"bash"	About an hour ago	Exited (0)	About an hour ago
ubuntu					
fa81adfa74ba	ubuntu:20.04	"bash"	About an hour ago	Exited (0)	About an hour ago
xenodochial_shirley					
0037efc732a6	dockerizedproject-web	"/bin/sh -c 'python ..."	3 hours ago	Exited (0)	About an hour ago
web					
2613fc9512ed	redislabs/redismod	"redis-server --load..."	3 hours ago	Exited (0)	About an hour ago
redis					

که نشان داده که ایمج ابونتو یک ساعت پیش ساخته شده است. حال برای اینکه این ایمج را روی داکرهاب قرار دهیم، ابتدا به آن یک تگ میدهم. برای این کار از دستور زیر استفاده میکنیم:

```
docker tag curly-ubuntu:1.0 nikast/curly-ubuntu:1.0
```

سپس آن را با دستور زیر روی داکرهاب پوش میکنیم:

```
docker image push nikast/curly-ubuntu:1.0
```



اکنون این ایمج را با دستور زیر از داکرهاب گرفته و آن را اجرا میکنیم:

docker run -it nikast/curly-ubuntu:1.0 bash

اکنون دستور curl google.com را اگر بزنیم، خروجی زیر را میگیریم:

```
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
```

```
<TITLE>301 Moved</TITLE></HEAD><BODY>
```

```
<H1>301 Moved</H1>
```

The document has moved

```
<A HREF="http://www.google.com/">here</A>.
```

```
</BODY></HTML>
```

و اگر دستور curl www.google.com را بزنیم، خروجی زیر را میگیریم:

```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IR"><head><meta content="Search the world's i
nformation, including webpages, images, videos and more. Google has many special features to help you find exactly what you'r
e looking for." name="description"><meta content="noodp" name="robots"><meta content="text/html; charset=UTF-8" http-equiv="C
ontent-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</tit
le><script nonce="aghFB-_lIQeqKeZSMXhgDw">(function(){window.google={kEI:'Z3OTY6TaN-uI9u8Pr8ensAo',kEXPI:'0,1359409,6059,206,
4804,2316,383,246,5,1129120,1197723,678,380090,16114,28684,22431,1361,283,12032,17584,4998,13228,3847,10623,22740,5081,887,70
6,1279,2742,149,1103,840,2197,4100,109,3405,606,2023,2297,14670,3227,2846,6,29074,4696,1850,15757,3,576,1014,1,5444,149,11323
,2652,4,1528,2304,27348,7422,7357,13658,4437,16786,5821,2536,4094,4052,3,3541,1,11943,30211,2,16737,3534,19490,5670,1021,2378
,28743,4568,6259,23416,1254,5835,14968,4332,7484,445,2,2,1,10957,15675,8155,7381,2,1477,14491,873,19632,8,1922,5784,3995,1388
0,6760,3053,7629,4832,26504,927,14678,4531,14,82,3890,751,13384,2181,1341,281,1779,669,1556,1626,1125,8130,779,81,246,1959,12
85,2411,1741,814,1224,10,308,2318,82,1419,2,566,988,1388,707,267,83,90,398,97,427,937,95,43,291,3836,82,399,470,961,3,261,523
,3,312,2318,135,962,342,246,930,1,861,172,1206,947,857,301,18,130,2,2984,229,79,3,442,96,937,184,186,913,6,190,83,374,390,187
,394,806,281,1252,5,269,173,1077,130,108,139,323,3,300,59,101,78,114,10,79,63,292,13,3,144,35,28,2,76,151,74,536,233,646,69,1
70,341,500,61,1,506,59,1623,441,263,5,3,247,530,462,890,251,36,70,115,38,135,326,120,1251,224,164,20,2,7,237,506,380,37,358,8
24,606,424,621,1,155,1979,1597,708,5281024,134,115,82,98,5993775,1660,2803414,3311,141,795,19735,1,303,44,2759,43,401,2,1,3,1
18,30,7,4,2,5,2,2,1,3,1,3,2,1,3,4,1,3,1,3,1,3,53,23947023,512,18,2769637,1271975,1964,16672,3406,5595,11,3835,3637,1494,544,1
521973',kBL:'oABN';google.sn='webhp';google.kHL='en-IR';})();(function(){
var f=this||self;var h,k=[];function l(a){for(var b;a&&(la.getAttribute||l(b=a.getAttribute("eid"))));a=a.parentNode;return b
||h}function m(a){for(var b=null;a&&(la.getAttribute||l(b=a.getAttribute("leid"))));a=a.parentNode;return b}
function n(a,b,c,d,g){var e="";c||-1!=b.search("&ei=")||e="&ei="+d(),-1==b.search("&lei=")&&(d=m(d))&&(e+="&lei="+d));d="
";l&&f._cschid&&-1==b.search("&cschid=")&&"slh"!=a&&(d="&cschid="+f._cschid);c=c||"/"+(g||"gen_204")+"?atyp=i&ict="+a+"&cad="+b
+e+"&zx="+Date.now()+d;/^http:\/i.test(c)&&"https:"==window.location.protocol&&(google.ml&&google.ml(Error("a"),l1,{src:c,glm
m:1}),c="");return c};h=google.kEI;google.getEI=l;google.getLEI=m;google.ml=function(){return null};google.log=function(a,b,c
,d,g){if(c=n(a,b,c,d,g)){a=new Image;var e=k.length;k[e]=a;a.onerror=a.onload=a.onabort=function(){delete k[e];a.src=c}};goo
gle.logUrl=n;}.call(this);(function(){google.y={};google.sy=[];google.x=function(a,b){if(a)var c=a.id;else{do c=Math.random(
);while(google.y[c])}google.y[c]=[a,b];return l1};google.sx=function(a){google.sy.push(a)};google.lm=[];google.plm=function(a)
{google.lm.push.apply(google.lm,a)};google.lq=[];google.load=function(a,b,c){google.lq.push([[a],b,c)};google.loadAll=functi
on(a,b){google.lq.push([a,b]);google.bx=l1;google.lx=function(){}}.call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");
a="1"===c||"q"===c&&la.elements.q.value!0:!!1}else a=l1;a&&(b.preventDefault(),b.stopPropagation()),l0);document.documentEle
ment.addEventListener("click",function(b){var a;a:{for(a=b.target;a&&a!=document.documentElement;a=a.parentElement)if("A"===
a.tagName){a="1"===a.getAttribute("data-nohref");break a;a=l1}a&&b.preventDefault(),l0;}}.call(this);</script><style>#gbar,
#guser{font-size:13px;padding-top:1px !important;}#gbar{height:22px}#guser{padding-bottom:7px !important;text-align:right}.gb
```

Figure ۱- بخشی از خروجی curl www.google.com

لیست ایمیج های موجود در سیستم خود به صورت زیر است:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
bdd85e621fc3	nikast/curly-ubuntu:1.0	"bash"	14 minutes ago	Exited (0)	About a minute ago
charming_bartik	curly-ubuntu:1.0	"bash"	4 hours ago	Exited (2)	4 hours ago
01a4acef4a7b	ubuntu				

گام دوم

در گام دوم باید سروری برای دریافت قیمت ارز های دیجیتال می ساختیم. در این بخش باید از قابلیت ردیس برای کش کردن اطلاعات استفاده میکردیم. برای این کار، در فایل config این زمان انقضا تنظیم میشود و اگر ارز موردنظر قبلا درخواست برای آن ارسال نشده بود، در ردیس ذخیره میشود و زمان انقضای آن از آن لحظه شروع میشود. اگر این ارز قبلا درخواست داده شده بود، دیگر به api سایت coinapi کال زده نمیشود و اطلاعات ارز موردنظر از ردیس خوانده و نمایش داده میشود.

```
if (redis.exists(config.coin_name) == False):
    response = requests.get('https://rest.coinapi.io/v1/assets/'+ config.coin_name , headers=headers)
    data_json = json.loads(response.text)
    redis.set(config.coin_name, json.dumps(data_json))
    redis.expire(config.coin_name, config.cache_timeout)
    print('does not exist')
else:
    data_json = json.loads(redis.get(config.coin_name))

return {
    'name': data_json[0]['name'],
    'price': data_json[0]['price_usd']
}
```

دستور docker ps برای نمایش container های موجود به صورت زیر است:

```
PS D:\lectures\Cloud Computing\9731702_HW2\ DockerizedProject> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NA
MES
96110c57ab4a   secondphase-web  "/bin/sh -c 'python ..." 31 seconds ago Up 25 seconds  0.0.0.0:5000->5000/tcp  we
200b164756db   redislabs/redismod  "redis-server --load..." 33 seconds ago Up 27 seconds  0.0.0.0:6379->6379/tcp  re
dis
```

همانطور که مشاهده میشود، برنامه دارای دو کانیتینر وب و ردیس می باشد. همچنین دستور docker inspect برای ایمچ سرور و کانیتینر وب به صورت زیر است:

```
PS D:\lectures\Cloud Computing\9731702_HW2\ DockerizedProject> docker inspect 96110c57ab4a
[
  {
    "Id": "96110c57ab4a45e0559db6c4c150df7ffce7eb1bb517f0ff267e07f7e6ecd2aa",
    "Created": "2022-12-15T20:26:53.0018603Z",
    "Path": "/bin/sh",
    "Args": [
      "-c",
      "python app.py"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 6714,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2022-12-15T20:26:59.6685568Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:3f7379c4aad4d897feb10eae06f6d9fbc92e86da0ffa3decdd0578f9f45d0ca5",
    "ResolvConfPath": "/var/lib/docker/containers/96110c57ab4a45e0559db6c4c150df7ffce7eb1bb517f0ff267e07f7e6ecd2aa/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/96110c57ab4a45e0559db6c4c150df7ffce7eb1bb517f0ff267e07f7e6ecd2aa/hostname",
    "HostsPath": "/var/lib/docker/containers/96110c57ab4a45e0559db6c4c150df7ffce7eb1bb517f0ff267e07f7e6ecd2aa/hosts",
    "LogPath": "/var/lib/docker/containers/96110c57ab4a45e0559db6c4c150df7ffce7eb1bb517f0ff267e07f7e6ecd2aa/96110c57ab4a45e0559db6c4c150df7ffce7eb1bb517f0ff267e07f7e6ecd2aa-json.log",
    "Name": "/web",
  }
]
```

و برای کانتینر ردیس به صورت زیر می باشد:

```
PS D:\lectures\Cloud Computing\9731702_HW2\DockerizedProject> docker inspect 200b164756db
[
  {
    "Id": "200b164756dbc75b55246317afc3ca9d2d004a410c841db6b331543c0d95c577",
    "Created": "2022-12-15T20:26:51.2806588Z",
    "Path": "redis-server",
    "Args": [
      "--loadmodule",
      "/usr/lib/redis/modules/redisai.so",
      "--loadmodule",
      "/usr/lib/redis/modules/redisearch.so",
      "--loadmodule",
      "/usr/lib/redis/modules/redisgraph.so",
      "--loadmodule",
      "/usr/lib/redis/modules/redistimeseries.so",
      "--loadmodule",
      "/usr/lib/redis/modules/rejson.so",
      "--loadmodule",
      "/usr/lib/redis/modules/redisbloom.so",
      "--loadmodule",
      "/usr/lib/redis/modules/redisgears.so",
      "Plugin",
      "/var/opt/redislabs/modules/rg/plugin/gears_python.so"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 6605,
      "ExitCode": 0,

```

خروجی دستور `docker stats` برای نمایش منابع در حال استفاده به صورت زیر می باشد:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
96110c57ab4a	web	1.89%	56.42MiB / 6.101GiB	0.90%	936B / 0B	0B / 0B	3
200b164756db	redis	3.16%	20.09MiB / 6.101GiB	0.32%	1.12kB / 0B	0B / 0B	26

آدرس زیر نیز برای `persist` کردن اطلاعات به عنوان `volume` استفاده میشود:

```
volumes:
  - ./redis.conf:/usr/local/etc/redis/redis.conf
```

گام سوم

در گام سوم برای هر یک از سرویس های وب و ردیس فایل های مربوطه را ساخته و با `apply` کردن آن ها در `kubectl` کانتینتر مربوط به آنها را در `minikube` ایجاد میکنیم.

فایل های مربوطه در پروژه قرار گرفته اند.