mas… 2 Branches 97 Tags Go to file Go to file Code

budtmo Updated appium-docker-android image ✓ 5688826 · 3 months ago 651 Commits

.github	Added Android 14 in pipeline	last year
cli	[pip] Bump coverage from 7.5.4 to 7.6.1	4 months ago
docker	Updated appium-docker-android image	3 months ago
documentations	Updated documentation	15 days ago
example/genymotion	Big restructuring by moving to python	last year
images	Resized the image	last year
mixins	Big restructuring by moving to python	last year
.dockerignore	Big restructuring by moving to python	last year
.gitignore	Big restructuring by moving to python	last year
LICENSE.md	Big restructuring by moving to python	last year
MAINTAINERS.md	Big restructuring by moving to python	last year
README.md	Update README.md	5 months ago
app.sh	Added Android 14	last year

README License

paypal donate PRs welcome codecov 67% chat on gitter release v2.11.4-p0

Docker-Android is a docker image built to be used for everything related to Android. It can be used for Application development and testing (native, web and hybrid-app).

Advantages of using this project

1. Emulator with different device profile and skins, such as Samsung Galaxy S6, LG Nexus 4, HTC Nexus One and more.
2. Support vnc to be able to see what happen inside docker container
3. Support log sharing feature where all logs can be accessed from web-UI
4. Ability to control emulator from outside container by using adb connect
5. Integrated with other cloud solutions, e.g. Genymotion Cloud
6. It can be used to build Android project
7. It can be used to run unit and UI-Test with different test-frameworks, e.g. Appium, Espresso, etc.

List of Docker-Images

Android	API	Image with latest release version	Image with specific release version
9.0	28	budtmo/docker-android:emulator_9.0	budtmo/docker-android:emulator_9.0_<release_version>
10.0	29	budtmo/docker-android:emulator_10.0	budtmo/docker-android:emulator_10.0_<release_version>
11.0	30	budtmo/docker-android:emulator_11.0	budtmo/docker-android:emulator_11.0_<release_version>
12.0	32	budtmo/docker-android:emulator_12.0	budtmo/docker-android:emulator_12.0_<release_version>
13.0	33	budtmo/docker-android:emulator_13.0	budtmo/docker-android:emulator_13.0_<release_version>
14.0	34	budtmo/docker-android:emulator_14.0	budtmo/docker-android:emulator_14.0_<release_version>
-	-	budtmo/docker-android:genymotion	budtmo/docker-android:genymotion_<release_version>

About

Android in docker solution with noVNC supported and video recording

#android #emulator #docker #kubernetes #jenkins #aws #cloud #azure #terraform #selenium #gcp #mobile-app #novnc #saltstack #android-emulator #selenium-grid #mobile-web #docker-android #genymotion #alibabacloud

Readme
View license
Activity
9.9k stars
169 watching
1.3k forks
Report repository

Releases 93

v2.11.4-p0 Latest
on Sep 26

+ 92 releases

paypal.me/budtmo

Learn more about GitHub Sponsors

Packages

No packages published

Contributors 48

+ 34 contributors

Languages

Python 91.7% Shell 8.3%

List of Devices

Type	Device Name
Phone	Samsung Galaxy S10
Phone	Samsung Galaxy S9
Phone	Samsung Galaxy S8
Phone	Samsung Galaxy S7 Edge
Phone	Samsung Galaxy S7
Phone	Samsung Galaxy S6
Phone	Nexus 4
Phone	Nexus 5
Phone	Nexus One
Phone	Nexus S
Tablet	Nexus 7

Requirements

1. Docker is installed on your system.

Quick Start

1. If you use ***Ubuntu OS*** on your host machine, you can skip this step. For ***OSX*** and ***Windows OS*** user, you need to use Virtual Machine that support Virtualization with Ubuntu OS because the image can be run under ***Ubuntu OS only***.

2. Your machine should support virtualization. To check if the virtualization is enabled is:

```
sudo apt install cpu-checker
kvm-ok
```

3. Run Docker-Android container

```
docker run -d -p 6080:6080 -e EMULATOR_DEVICE="Samsung Galaxy S10" -e WEB_VNC=true --device /dev/kvm --name
android-container budtmo/docker-android:emulator_11.0
```

4. Open ***http://localhost:6080*** to see inside running container.

5. To check the status of the emulator

```
docker exec -it android-container cat device_status
```

Persisting data

The default behaviour is to destroy the emulated device on container restart. To persist data, you need to mount a volume at `/home/androidusr`: `docker run -v data:/home/androidusr budtmo/docker-android:emulator_11.0`

WSL2 Hardware acceleration (Windows 11 only)

Credit goes to Guillaume - The Parallel Interface blog

Microsoft - Advanced settings configuration in WSL

1. Add yourself to the `kvm` usergroup.

```
sudo usermod -a -G kvm ${USER}
```

2. Add necessary flags to `/etc/wsl2.conf` to their respective sections.

```
[boot]
command = /bin/bash -c 'chown -v root:kvm /dev/kvm && chmod 660 /dev/kvm'

[wsl2]
nestedVirtualization=true
```

3. Restart WSL2 via CMD prompt or Powershell

```
wsl --shutdown
```

`command = /bin/bash -c 'chown -v root:kvm /dev/kvm && chmod 660 /dev/kvm'` sets `/dev/kvm` to `kvm` usergroup rather than the default `root` usergroup on WSL2 startup.

`nestedVirtualization` flag is only available to Windows 11.

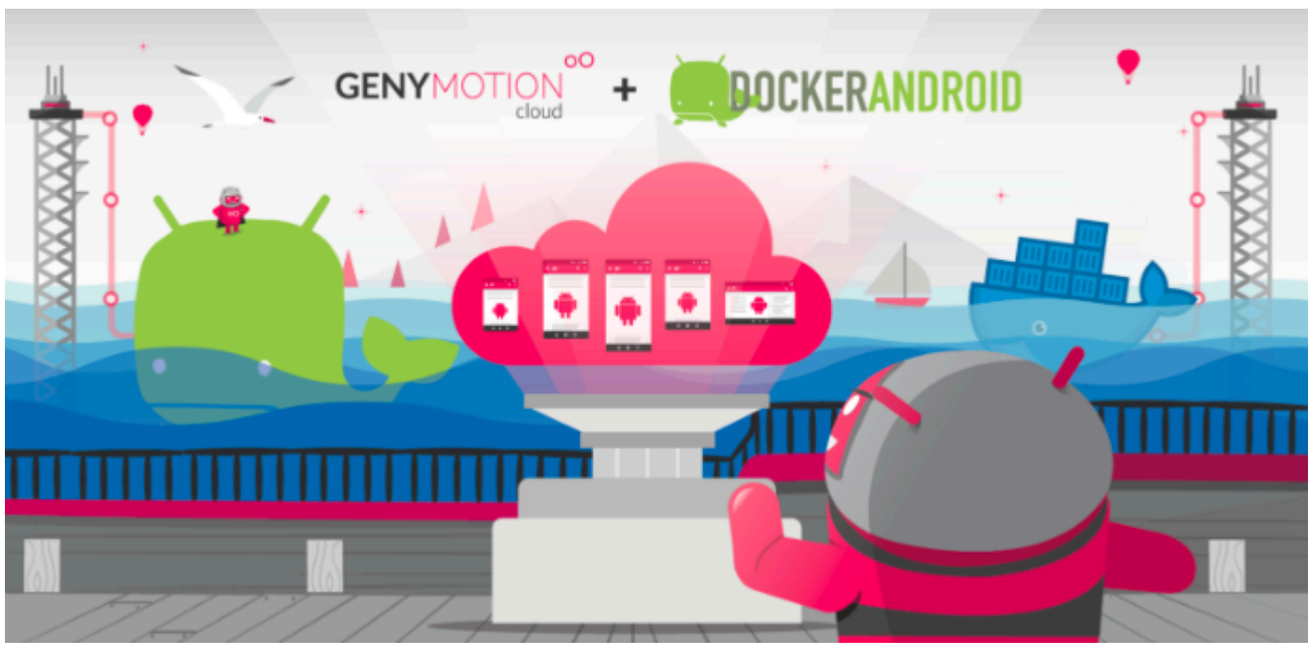
Use-Cases

1. Build Android project
2. UI-Test with Appium
3. Control Android emulator on host machine
4. SMS Simulation
5. Jenkins
6. Deploying on cloud (Azure, AWS, GCP)

Custom-Configurations

This document contains information about configurations that can be used to enable some features, e.g. log-sharing, etc.

Genymotion



For you who do not have ressources to maintain the simulator or to buy machines or need different device profiles, you can give a try by using Genymotion SAAS. Docker-Android is integrated with Genymotion on different cloud services, e.g. Genymotion SAAS, AWS, GCP, Alibaba Cloud. Please follow this document for more detail.

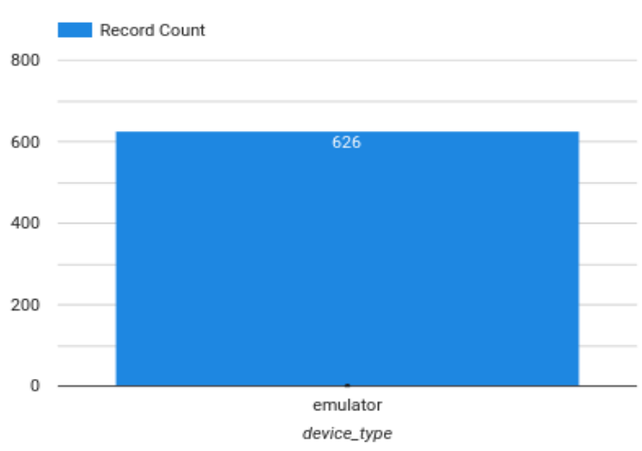
Emulator Skins

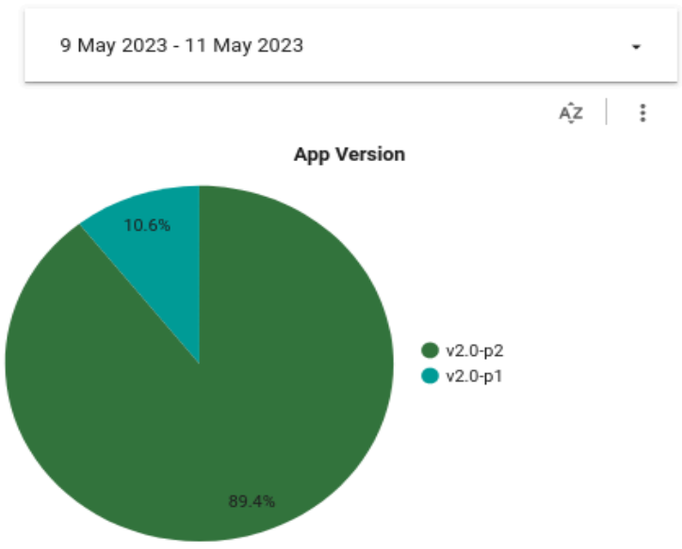
The Emulator skins are taken from Android Studio IDE and Samsung Developer Website

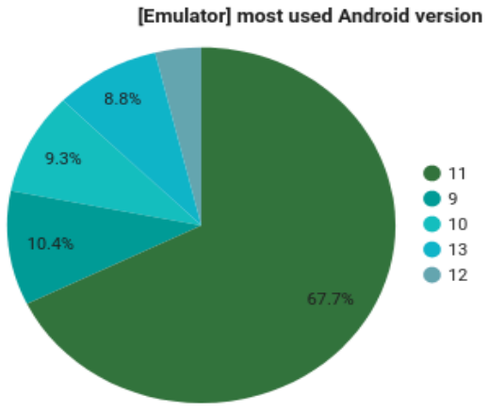
USERS

	country	city	user
1.	CA	Toronto	4
2.	DE	Frankfurt am Main	3
3.	RU	Vologda	3
4.	VN	Ho Chi Minh City	3
5.	IE	Dublin	3
6.	DE	Berlin	3
7.	ES	Valladolid	2
8.	TR	Istanbul	2
9.	NL	Amsterdam	2
10.	VN	Hanoi	2

1 - 20 / 50







PRO VERSION

Due to high requests for help and to be able to actively maintain the projects, the creator has decided to create docker-android-pro. Docker-Android-Pro is a sponsor based project which mean that the docker image of pro-version can be pulled only by active sponsor.

The differences between normal version and pro version are:

Feature	Normal	Pro	Comment
user-behavior-analytics	Yes	No	-
proxy	No	Yes	Set up company proxy on Android emulator on fly
language	No	Yes	Set up language on Android emulator on fly
root-privileged	No	Yes	Able to run command with security privileged
headless-mode	No	Yes	Save resources by using headless mode
Selenium 4.x integration	No	Yes	Running Appium UI-Tests againt one (Selenium Hub) endpoint for Android- and iOS emulator(s) / device(s)
multiple Android-Simulators	No	Yes (soon)	Save resources by having multiple Android-Simulators on one docker-container
Google Play Store	No	Yes (soon)	-
Video Recording	No	Yes (soon)	Helpful for debugging

This document contains detail information about how to use docker-android-pro.

LICENSE