[Getting Started with Redash](https://www.youtube.com/watch?v=Yn3_QDk2qQM&t=291s)

**docker pull redash/redash**

## **How to deploy Redash**

Normally, I’d first walk you through the process of installing Docker, but the developers behind Redash have created a very easy-to-use setup script that installs everything you need (docker-ce and docker-compose) and then deploys everything necessary for Redash to function.

The first thing you should do is make certain your system is up-to-date. Log into Ubuntu Server and issue the following two commands:

sudo apt-get update

sudo apt-get upgrade -y

Remember, if the kernel gets upgraded, you'll need to reboot the machine for the changes to take effect.

Once the upgrade is complete, download the installer script with the command:

curl -O <https://raw.githubusercontent.com/getredash/setup/master/setup.sh>

Give the installer executable permissions with:

chmod u+x setup.sh

Finally, run the installer with:

sudo ./setup.sh

—----------------------------------------------------------------------------------------------------------------------------

# Environment Variables Settings

Many aspects of the functionality of Redash can be changed with settings. Settings are read by redash.settings from environment variables which (for most installs) can be set in /opt/redash/.env. When developing with Docker, create the .env file in the root directory.

The follow is a list of settings and what they control:

| **Name** | **Description** | **Default Value** |
| --- | --- | --- |
| REDASH\_REDIS\_URL | URL Redash services will use to read and write to redis | “redis://localhost:6379/0” |
| REDASH\_DISABLE\_PUBLIC\_URLS | Whether to disable access to public URLs | “false” |
| REDASH\_BLOCKED\_DOMAINS | Comma separated of email domains that cannot create user accounts | “qq.com” |
| REDASH\_PROXIES\_COUNT | Passed to Werkzeug’s [ProxyFix](https://werkzeug.palletsprojects.com/en/2.1.x/middleware/proxy_fix/) to make sure we get the right referral address even behind proxies like nginx. | 1 |
| REDASH\_STATSD\_HOST | Host for STATSD daemon | 127.0.0.1 |
| REDASH\_STATSD\_PORT | Port for STATSD daemon | 8125 |
| REDASH\_STATSD\_PREFIX | Prefix for metrics sent to STATSD daemon | redash |
| REDASH\_STATSD\_USE\_TAGS | Whether to use tags in StatsD metrics (InfluxDB’s format) | false |
| REDASH\_DATABASE\_URL | URL the Redash server and worker services will use to access the metadata database | postgresql://postgres |
| REDASH\_QUERY\_RESULTS\_CLEANUP\_ENABLED | Whether the job to cleanup unused query results should run automatically | true |
| REDASH\_QUERY\_RESULTS\_CLEANUP\_COUNT | The number of results to cleanup during each call to cleanup\_query\_results | 100 |
| REDASH\_QUERY\_RESULTS\_CLEANUP\_MAX\_AGE | Integer number of days, past which a query result may be cleaned up if it is not referenced by another query. | 7 |
| REDASH\_SCHEMAS\_REFRESH\_SCHEDULE | How often to refresh the data source schemas (in minutes) | 30 |
| REDASH\_AUTH\_TYPE | May be either api\_key or hmac. Implemented [here](https://github.com/getredash/redash/pull/385) | api\_key |
| REDASH\_ENFORCE\_HTTPS | This is passed to [Flask-Talisman](https://github.com/GoogleCloudPlatform/flask-talisman) | false |
| REDASH\_INVITATION\_TOKEN\_MAX\_AGE | An integer number of seconds after which an invitation link will expire | 60 \_ 60 \_ 24 \* 7 |
| REDASH\_MULTI\_ORG | Whether to enable multi-org mode. Note: Multi-org mode is not documented or supported at this time | false |
| REDASH\_GOOGLE\_CLIENT\_ID | The client ID to use for Google Login, be sure to set a client secret as well |  |
| REDASH\_GOOGLE\_CLIENT\_SECRET | The client secret to use for Google Login, be sure to set a client id as well |  |
| REDASH\_REMOTE\_USER\_LOGIN\_ENABLED | Enables the use of an externally-provided and trusted remote user via an HTTP header. The “user” must be an email address. [More details](https://github.com/getredash/redash/blob/e6ebef1e5ab866ce1e706eaee6260edaffdc2bd7/redash/settings/__init__.py#L185-L207) | false |
| REDASH\_REMOTE\_USER\_HEADER | Name of header to use if REDASH\_REMOTE\_USER\_LOGIN\_ENABLED is true | X-Forwarded-Remote-User |
| REDASH\_LDAP\_LOGIN\_ENABLED |  | false |
| REDASH\_LDAP\_URL |  | None |
| REDASH\_LDAP\_BIND\_DN |  | None |
| REDASH\_LDAP\_BIND\_DN\_PASSWORD |  |  |
| REDASH\_LDAP\_DISPLAY\_NAME\_KEY |  | displayName |
| REDASH\_LDAP\_EMAIL\_KEY |  | mail |
| REDASH\_LDAP\_CUSTOM\_USERNAME\_PROMPT |  | LDAP/AD/SSO username: |
| REDASH\_LDAP\_SEARCH\_TEMPLATE |  | (cn=%(username)s) |
| REDASH\_LDAP\_SEARCH\_DN |  | REDASH*SEARCH*DN |
| REDASH\_STATIC\_ASSETS\_PATH | Directory that contains all front-end assets. Relative to the ./redash directory | ”../client/dist/” |
| REDASH\_JOB\_EXPIRY\_TIME | TTL in seconds for jobs placed in queue. If a job is not picked up by a worker within this TTL it will expire. | 3600 \* 12 |
| REDASH\_COOKIE\_SECRET | Required. Used for various cryptographic features of the web server. Read more about how Redash uses secret keys [here](https://redash.io/help/open-source/admin-guide/secrets) |  |
| REDASH\_LOG\_LEVEL | Logging messages which are less severe than level will be ignored. Read more about Python logging [here](https://docs.python.org/3/library/logging.html) | INFO |
| REDASH\_MAIL\_SERVER |  | localhost |
| REDASH\_MAIL\_PORT |  | 25 |
| REDASH\_MAIL\_USE\_TLS |  | false |
| REDASH\_MAIL\_USE\_SSL |  | false |
| REDASH\_MAIL\_USERNAME |  | None |
| REDASH\_MAIL\_PASSWORD |  | None |
| REDASH\_MAIL\_DEFAULT\_SENDER |  | None |
| REDASH\_MAIL\_MAX\_EMAILS |  | None |
| REDASH\_MAIL\_ASCII\_ATTACHMENTS |  | false |
| REDASH\_HOST | The URL host used in emails sent to users (invites, alerts, notifications) etc. |  |
| REDASH\_ALERTS\_DEFAULT\_MAIL\_SUBJECT\_TEMPLATE | Subject used for email alert notifications. {state} and {alert\_name} are the only available variables. | ({state}) {alert\_name} |
| REDASH\_THROTTLE\_LOGIN\_PATTERN | The [Flask-Limiter](https://flask-limiter.readthedocs.io/en/stable/) string pattern used to rate limit requests to the /login route. | 50/hour |
| REDASH\_LIMITER\_STORAGE | Mapped directly to Flask-Limiter’s [storage\_uri](https://flask-limiter.readthedocs.io/en/stable/configuration.html?highlight=limiter_storage#RATELIMIT_STORAGE_URI). | REDIS\_URL |
| REDASH\_CORS\_ACCESS\_CONTROL\_ALLOW\_ORIGIN |  |  |
| REDASH\_CORS\_ACCESS\_CONTROL\_ALLOW\_CREDENTIALS |  | false |
| REDASH\_CORS\_ACCESS\_CONTROL\_REQUEST\_METHOD |  | GET, POST, PUT |
| REDASH\_CORS\_ACCESS\_CONTROL\_ALLOW\_HEADERS |  | Content-Type |
| REDASH\_ENABLED\_QUERY\_RUNNERS | Comma-separated list of query runners to be enabled (e.g. redash.query\_runner.pg,redash.query\_runner.mysql) | ”,”.join(default*query*runners) |
| REDASH\_ADDITIONAL\_QUERY\_RUNNERS | Comma-separated list of non-default query runners to be enabled |  |
| REDASH\_DISABLED\_QUERY\_RUNNERS | Comma-separated list of query runners that will not appear in Redash |  |
| REDASH\_SCHEDULED\_QUERY\_TIME\_LIMIT | Time limit for scheduled queries (in seconds) | None |
| REDASH\_ADHOC\_QUERY\_TIME\_LIMIT | Time limit for adhoc queries (in seconds) | None |
| REDASH\_ENABLED\_DESTINATIONS | Comma-separated list of alert destinations to be enabled (e.g. redash.destinations.email,redash.destinations.slack ) | ”,”.join(default\_destinations) |
| REDASH\_ADDITIONAL\_DESTINATIONS | Comma-separated list of non-default alert destinations to be enabled |  |
| REDASH\_EVENT\_REPORTING\_WEBHOOKS | Comma-separated list of webhook URLs that to which events will be forwarded |  |
| REDASH\_SENTRY\_DSN |  |  |
| REDASH\_ALLOW\_SCRIPTS\_IN\_USER\_INPUT | Disable sanitization of text input, allowing full HTML | false |
| REDASH\_DASHBOARD\_REFRESH\_INTERVALS | Comma-separated list of integer seconds that will be allowed for dashboard refresh | 60,300,600,1800,3600,43200,86400 |
| REDASH\_QUERY\_REFRESH\_INTERVALS | Comma-separated list of integer seconds that will be allowed for scheduled query refresh | 60, 300, 600, 900, 1800, 3600, 7200, 10800, 14400, 18000, 21600, 25200, 28800, 32400, 36000, 39600, 43200, 86400, 604800, 1209600, 2592000 |
| REDASH\_PASSWORD\_LOGIN\_ENABLED | Toggle to allow password login. Often disabled if Google Login, SAML, LDAP, or REMOTE*USER*LOGIN are enabled. | true |
| REDASH\_SAML\_METADATA\_URL |  |  |
| REDASH\_SAML\_ENTITY\_ID |  |  |
| REDASH\_SAML\_NAMEID\_FORMAT |  |  |
| REDASH\_DATE\_FORMAT | Moment.js format to be used throughout the web application | DD/MM/YY |
| REDASH\_JWT\_LOGIN\_ENABLED |  | false |
| REDASH\_JWT\_AUTH\_ISSUER |  |  |
| REDASH\_JWT\_AUTH\_PUBLIC\_CERTS\_URL |  |  |
| REDASH\_JWT\_AUTH\_AUDIENCE |  |  |
| REDASH\_JWT\_AUTH\_ALGORITHMS |  | HS256,RS256,ES256 |
| REDASH\_JWT\_AUTH\_COOKIE\_NAME |  |  |
| REDASH\_JWT\_AUTH\_HEADER\_NAME |  |  |
| REDASH\_FEATURE\_SHOW\_QUERY\_RESULTS\_COUNT | Disable/enable showing count of query results in status | true |
| REDASH\_VERSION\_CHECK | Toggle whether to periodically check if a newer version of Redash is available. | true |
| REDASH\_FEATURE\_DISABLE\_REFRESH\_QUERIES | Disable scheduled query execution | false |
| REDASH\_FEATURE\_SHOW\_PERMISSIONS\_CONTROL | Enable experimental multiple owners support | false |
| REDASH\_FEATURE\_ALLOW\_CUSTOM\_JS\_VISUALIZATIONS | Enable the custom visualization option. This appears as a sub-type of “Chart” visualizations in the UI. | false |
| REDASH\_FEATURE\_AUTO\_PUBLISH\_NAMED\_QUERIES | Automatically publish a new query after its name is changed from “New Query” | true |
| REDASH\_BIGQUERY\_HTTP\_TIMEOUT | HTTP timeout for the BigQuery query runner | 600 |
| REDASH\_SCHEMA\_RUN\_TABLE\_SIZE\_CALCULATIONS | Enable showing the size of each table in the schema browser. | false |
| REDASH\_WEB\_WORKERS | How many processes will gunicorn spawn to handle web requests | 4 |

—----------------------------------------------------------------------------------------------------------------------------

#!/usr/bin/env bash

# This script setups dockerized Redash on Ubuntu 18.04.

set -eu

REDASH\_BASE\_PATH=/opt/redash

install\_docker(){

# Install Docker

export DEBIAN\_FRONTEND=noninteractive

sudo apt-get -qqy update

DEBIAN\_FRONTEND=noninteractive sudo -E apt-get -qqy -o Dpkg::Options::="--force-confdef" -o Dpkg::Options::="--force-confold" upgrade

sudo apt-get -yy install apt-transport-https ca-certificates curl software-properties-common wget pwgen

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update && sudo apt-get -y install docker-ce

# Install Docker Compose

sudo curl -L https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

# Allow current user to run Docker commands

sudo usermod -aG docker $USER

}

create\_directories() {

if [[ ! -e $REDASH\_BASE\_PATH ]]; then

sudo mkdir -p $REDASH\_BASE\_PATH

sudo chown $USER:$USER $REDASH\_BASE\_PATH

fi

if [[ ! -e $REDASH\_BASE\_PATH/postgres-data ]]; then

mkdir $REDASH\_BASE\_PATH/postgres-data

fi

}

create\_config() {

if [[ -e $REDASH\_BASE\_PATH/env ]]; then

rm $REDASH\_BASE\_PATH/env

touch $REDASH\_BASE\_PATH/env

fi

COOKIE\_SECRET=$(pwgen -1s 32)

SECRET\_KEY=$(pwgen -1s 32)

POSTGRES\_PASSWORD=$(pwgen -1s 32)

REDASH\_DATABASE\_URL="postgresql://postgres:${POSTGRES\_PASSWORD}@postgres/postgres"

echo "PYTHONUNBUFFERED=0" >> $REDASH\_BASE\_PATH/env

echo "REDASH\_LOG\_LEVEL=INFO" >> $REDASH\_BASE\_PATH/env

echo "REDASH\_REDIS\_URL=redis://redis:6379/0" >> $REDASH\_BASE\_PATH/env

echo "POSTGRES\_PASSWORD=$POSTGRES\_PASSWORD" >> $REDASH\_BASE\_PATH/env

echo "REDASH\_COOKIE\_SECRET=$COOKIE\_SECRET" >> $REDASH\_BASE\_PATH/env

echo "REDASH\_SECRET\_KEY=$SECRET\_KEY" >> $REDASH\_BASE\_PATH/env

echo "REDASH\_DATABASE\_URL=$REDASH\_DATABASE\_URL" >> $REDASH\_BASE\_PATH/env

}

setup\_compose() {

REQUESTED\_CHANNEL=stable

LATEST\_VERSION=`curl -s "https://version.redash.io/api/releases?channel=$REQUESTED\_CHANNEL" | json\_pp | grep "docker\_image" | head -n 1 | awk 'BEGIN{FS=":"}{print $3}' | awk 'BEGIN{FS="\""}{print $1}'`

cd $REDASH\_BASE\_PATH

GIT\_BRANCH="${REDASH\_BRANCH:-master}" # Default branch/version to master if not specified in REDASH\_BRANCH env var

wget https://raw.githubusercontent.com/getredash/setup/${GIT\_BRANCH}/data/docker-compose.yml

sed -ri "s/image: redash\/redash:([A-Za-z0-9.-]\*)/image: redash\/redash:$LATEST\_VERSION/" docker-compose.yml

echo "export COMPOSE\_PROJECT\_NAME=redash" >> ~/.profile

echo "export COMPOSE\_FILE=/opt/redash/docker-compose.yml" >> ~/.profile

export COMPOSE\_PROJECT\_NAME=redash

export COMPOSE\_FILE=/opt/redash/docker-compose.yml

sudo docker-compose run --rm server create\_db

sudo docker-compose up -d

}

install\_docker

create\_directories

create\_config

setup\_compose