CONTROLIO – ON-PREMISES DEPLOYMENT TUTORIAL

BENEFITS OF ON-PREMISES DEPLOYMENT

- Get full control over your data
- No data sent out of your network
- HIPAA, GDPR compliance
- Managed data retention
- Control your backups
- Private Cloud support (deploy Docker container in any environment)
- AWS S3 storage support (screen data, email attachment)

TECHNICAL REQUIREMENTS

SERVER COMPUTER

Controlio On-Premises server app is shipped as Docker container - multi-platform virtual machine supported by Windows OS and Linux OS. Linux OS is a preferable platform as Docker works on it natively.

List of OS supported versions*:

- Windows Server 2022**, Windows 10, 11
- CentOS 7 or 8
- Debian 12
- Fedora 40 or Fedora 41
- Ubuntu 22.04, 24.04, or the latest non-LTS version

**all supported platforms are 64-bit*

*** Support for this OS is experimental, as it is not officially supported by Docker Desktop, so the setup may involve many nuances*

Hardware requirements for the on-premises server

50..100 users:

- 1Ghz CPU
- 2Gb RAM
- HDD 7200rpm
- 2-months data retention storage estimates: 2 Gb for DB Storage, 250GB for Screen Storage
- 6-months data retention storage estimates: 5 Gb for DB Storage, 750GB for Screen Storage

100..500 users:

- 2 CPU cores
- 4Gb RAM
- SSD

- 2-months data retention storage estimates: 4 Gb for DB Storage, 500GB for Screen Storage
- 6-months data retention storage estimates: 10 Gb for DB Storage, 1500GB for Screen Storage

500..1000 users:

- 4 CPU cores
- 8Gb RAM
- SSD 200Mb/sec r/w speed
- 2-months data retention storage estimates: 15 Gb for DB Storage, 2.5Tb for Screen Storage
- 6-months data retention storage estimates: 50 Gb for DB Storage, 7.5Tb for Screen Storage

1000..2000 users:

- 8 CPU cores
- 16Gb RAM
- SSD 400Mb/sec r/w speed
- 2-months data retention storage estimates: 45 Gb for DB Storage, 7.5Tb for Screen Storage
- 6-months data retention storage estimates: 150 Gb for DB Storage, 22.5Tb for Screen Storage

Storage size requirements (all features on, emails with no MS Outlook attachments)*:

- Database Storage: 20 Mb per user/monthly
- Screens Storage**: 2.5 Gb per user/monthly (100hrs of active time per month, office apps, Full HD mp4)

**based on the average of real figures provided by our customers*

***screen recordings may be stored on an Amazon S3 Cloud as an alternative option*

**** MS Outlook attachments may add significant data size, highly depends on the customer and cannot be estimated*

AWS EC2 requirements

EC2 instance(s) should be provisioned based on the expected number of concurrent monitored sessions, according to the following table:

Up to 100 users: m4.large 2CPU/8GB (estimate 72$/month)

Up to 500 users: m4.xlarge 4CPU/16GB (estimate 144$/month)

Up to 1000 users: m4.2xlarge 8CPU/32GB (estimate 288$/month)

Larger deployments: equivalent to m4.2xlarge per each 1000 users (example: m4.4xlarge per 2000 users, m4.16xlarge per 8000 users)

**comparable t3 EC2 instances may be used as an analog but they have a less stable performance.*

***Controlio server has no limits for the number of client connections, all depends on the hardware performance.*

CLIENT AGENT COMPUTER

List of OS supported versions:

- Windows Vista, 7,8,10,11, Server 2016,2019,2022 - both 32bit and 64bit editions
- macOS 10.14 (Mojave), 10.15 (Catalina), 11 (Big Sur), 13 (Ventura), 14 (Sonoma), 15 (Sequoia)

Hardware requirements:

- 1Ghz CPU
- 2Gb RAM
- 1Gb free drive space
- 0.2 Mbit Network speed

DEPLOYMENT

WINDOWS/LINUX OS

- Install Docker: https://docs.docker.com/engine/install/
- Download .yml file from https://controlio.net/docker-compose-local-distr.yml

- Enter fields in .yml file marked with "*[!!!]*" (note: *PUBLIC_URL* parameter is required to run the web app, enter server's IP. If the server is published to the external network (e.g. Internet), use the external IP here. This is important for web consoles to connect correctly. Server publication and IP forwarding should be implemented by customer's network admin.). Example:
- ```
  # Version of docker-compose
  version: '3.5'
  # Containers we are going to run
  services:
    # Our Phoenix container
    on_premise:
      image: stingmaster/controlio_on_premise:1.0.5
      ## The build parameters for this container.
      #build:
      #  # Here we define that it should build from the current directory
      #  context: .
      environment:
        # Variables to connect to our Postgres server
        PUBLIC_URL: 192.168.1.1
  ```
- If you do not use SMTP server, use space in SMTP parameters (SMTP_SERVER, SMTP_USERNAME, SMTP_PASSWORD)
- Request the LICENSE (trial or registered) from support@controlio.net.
- Enter the license key value received from support@controlio.net to the LICENSE: parameter as a single string
  ```

- Save changes in .yml file, stay in .yml file's folder and run the Docker container with Terminal command:

  docker-compose -f docker-compose-local-distr.yml up -d

  *note: if you want to check the app log, please run **docker-compose logs** command or run the container without **-d** command (it will write log to the terminal). To stop the container app use Ctrl+C.*

- Open the web browser and go to the *PUBLIC_URL,* use login credentials from the License info (received from support@controlio.net). Password can be changed later from Settings - Logins.

PUBLISH VIA HTTPS

Controlio Dashboard can be accessed via HTTPS. You may use your certificate or generate a self-signed certificate on service like Let's Encrypt. It will require proxy server configuration to publish Controlio Dashboard with SSL connection, here are sample instructions for NGINX.

The idea is that you browse to https://controlio.yourcompany.com, it goes over HTTPS to NGINX on :443 which proxies it to http://127.0.0.1:4000 (on_premise container).

YML file sample:

```
services:
on_premise:
   …
   environment:
    PUBLIC_URL: controlio.yourcompany.com # domain the SSL certificate is issued for
    USE_SSL: "yes"
    PUBLIC_PORT: 443 # this command doesn't bind the container to port 443
    LIVE_STREAM_PORT: 444
   ports:
    - "4000:4000" # container binds to 4000, so NGINX should proxy traffic on this port
    - "4001:4001"
```

nginx.conf sample:

```
map $http_upgrade $connection_upgrade {
  default upgrade;
  ''      close;
}

# publish main app
server {
   server_name controlio.yourcompany.com;
   listen 443 ssl http2;
   listen [::]:443 ssl http2;
```

```
    # SSL settings
    ssl_certificate /etc/nginx/ssl/controlio.yourcompany.com/fullchain.pem;; # path to the certificate
    ssl_certificate_key /etc/nginx/ssl/controlio.yourcompany.com/privkey.pem # path to the private key

    location / {
        proxy_pass http://localhost:4000;
        proxy_http_version 1.1;  # recommended with keepalive connections
# https://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_http_version
        # WebSocket proxying - from https://nginx.org/en/docs/http/websocket.html
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_read_timeout 150;
        client_max_body_size 100M;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

# publish live streaming
server {
    server_name controlio.yourcompany.com;

    listen 444 ssl http2;
    listen [::]:444 ssl http2;
    # SSL settings
    ssl_certificate /etc/nginx/ssl/controlio.yourcompany.com/fullchain.pem;; # path to the certificate
    ssl_certificate_key /etc/nginx/ssl/controlio.yourcompany.com/privkey.pem # path to the private key

    location / {
        proxy_pass http://localhost:4001;
        proxy_http_version 1.1;  # recommended with keepalive connections
# https://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_http_version
        # WebSocket proxying - from https://nginx.org/en/docs/http/websocket.html
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_read_timeout 150;
        client_max_body_size 100M;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

TECHNICAL SUPPORT AND LICENSE TERMS

- Free assistance on remote deployment
- Licensed by concurrent user sessions
- No license/limits for Dashboard users (admins, managers)

- No data retention period limits (limited by your storage)
- 1 business day support via email
- Live chat support on the website
- Remote assistance tools

COMPANY

EfficientLab LLC (NY, USA) is an independent software vendor providing high-quality and robust employee monitoring technologies. The first product, Work Examiner, was released in 2005.

Controlio (released in 2017) is a web-based cloud system for employee surveillance on their work PCs that run Windows. You can easily monitor web and application usage and watch what's happening on your staff screens live or on-demand. Productivity optimization is one of the reasons why you need a surveillance system. With Controlio, you can see productivity issues and organizational bottlenecks at their early stages and take care of them before they become big.

CONTACTS

- Website: https://controlio.net
- E-mail: support@controlio.net
- Phone: +1 (800) 470 - 9165