



Essentials MDM

Server installation guide

Table of contents

1. System requirements	3
1.1 Operating systems	3
1.2 Hardware resources	3
1.3 Networking	4
2. Installation	8
2.1 Referential deployment examples	8
2.2 Application server installation	9
2.3 Proxy server installation	10
2.4 Post installation tasks	10
3. Updating Essentials MDM	15
3.1 Standalone Server	15
3.2 Server with Proxy	15
4. Backup	16
5. Migration	17
5.1 Preparation	17
5.2 Migration procedure	17

1. System requirements

1.1 Operating systems

Essentials MDM development and support follows a subset of RHEL-derived Linux distributions.

Operating system	Version
Oracle Linux	8
Red Hat Enterprise Linux	8
Rocky Linux	8

Maintenance and lifecycle policy is to align support with operating system lifecycle. When OS platform reaches end of active support by its maintainers or its end of life – whichever comes first – after adjustment periods Essentials MDM installations require upgrade to an actively supported platform for access to new features and security updates.

1.2 Hardware resources

Primary factor for Essentials MDM infrastructure requirements is upper limit of actively managed devices – requirements assessed based on this metric are henceforth referenced as **baseline requirements**. Secondary factors that influence system requirements and may differ across environments or projects are outlined in deployment considerations.

Requirements are split accordingly to server role:

- Essentials MDM Application server – key component of the system, providing core services, database and management UI.
- Essentials MDM Proxy – optional reverse proxy component specialized to work with Application server

1.2.1 Compute and memory – application server

Actively managed devices	CPU cores	RAM
<500	4	8 GB
1000	6	16 GB
2500	8	16 GB
7500	16	32 GB
10000	24	64 GB
>15000	24	64 GB

1.2.2 Compute and memory – proxy server

Actively managed devices	CPU cores	RAM
<500	4	4 GB
1000	4	4 GB
2500	6	8 GB
7500	8	8 GB
10000	12	16 GB
>15000	16	16 GB

1.2.3 Data storage – application server

Approximation assuming usage of data collection policies and 5-year data retention. At least SSD-grade storage is strongly recommended for optimal database I/O performance.

Managed devices	Base storage
< 500	100 GB
1000	200 GB
2500	250 GB
5000	300 GB
7500	400 GB

Deployment considerations

Baseline resource requirements have been verified to allow usage of all features with spare reserve, however secondary factors may also influence resource requirements of MDM deployment.

- Usage of Exchange ActiveSync proxy module
- Managing devices in networks with no access to Android Push services – fallback to less efficient polling communication.
- High frequency polling intervals set in device policies
- High frequency collection of location or usage data from managed devices
- Intensive use of application webservice API by external system
- Non-MDM services or antivirus software being hosted on application server

1.3 Networking

Essentials MDM requires connectivity to several of our services for correct operation, receiving system updates, and license verification. Depending on used features and integrations, MDM will initiate connections to external Google, Apple, and Microsoft services for communication with push services, application stores CDNs and vendor-specific integration APIs. Review below rules for any applicable vendor/OEM documentations whenever deploying Essentials MDM on networks with strict outgoing access controls.

Both Google and Apple specify lists of hosts that need to be accessible for device management. If a firewall can only be configured with IP addresses, subnet blocks are provided in external vendor links at the end of this section.

If using IP-based access control, be aware that global and regional CDNs utilize large address blocks, which may result in inconveniently growing firewall ACLs – as an alternative, Essentials MDM server may be configured to access external services through an outgoing HTTP proxy.

1.3.1 Core operation

Required for correct MDM system functioning.

Source	Destination	Ports	Note
Internet	Application / Proxy	443 TCP	Device communication, admin console
Proxy	Application	443, 1289, 6379, 11009-11021 TCP	Proxy to Application communication
Application	Proxy	443 TCP	Application to Proxy communication
Application, Proxy	partners.fancyfon.com	443 TCP	Updates, license verification, core services
Application, Proxy	repository.fancyfon.com	443 TCP	Update repositories
Application, Proxy	mirror.fancyfon.com	443 TCP	Updates repositories
Application, Proxy	crldp.fancyfon.com	443 TCP	License verification
Application, Proxy	BaseOS, Appstream	443 TCP	OS package repositories

1.3.2 Modules

Source and destination ranges, as well as ports of optional modules - may be customized to some extent. Details concerning customizing module configurations fall outside of overview outlined in this document's scope. While specific feature will be affected, core MDM operation will not degrade if corporate network policies deny possibility to fulfill a specific module's network requirements.

Source	Destination	Ports	Note
any / Managed devices network	MDM Application / MDM Proxy	11009 TCP	Remote Access v1: device screen streaming
any / Managed devices network	MDM Application / MDM Proxy	7881 TCP, 11008 UDP, 5349 TCP, 3468 UDP	Remote Access v2: WebRTC + TURN
Application	AD/LDAP	389/636 TCP	Pull users/groups from directory servers, AD/LDAP based UI console authentication
Application / Proxy	any / Managed devices network	11040 - 11100 UDP, TCP	Remote Access v2: TURN bridge
Application, Proxy	stun{1,2,3,4}.l.google.com	19302 UDP	Remote Access v2: STUN relays

1.3.3 Recommended third-party integrations

Although technically possible to operate Essentials MDM server without access to Google or Apple services, doing so results in severely limited features for actual management of Android devices and excludes possibility to manage Apple devices.

Source	Destination	Ports	Note
Application	oauth2.googleapis.com	443 TCP	Android Zero-Touch, Android Push
Application	fonts.googleapis.com	443 TCP	Google CDN
Application	api.push.apple.com	443 TCP	APNS Push
Application	itunes.apple.com	443 TCP	App store
Application	vpp.itunes.apple.com	443 TCP	App store VPP integration
Application	mdmenrollment.apple.com	443 TCP	Apple DEP
Application	feedback.push.apple.com	443 TCP	APNS Push
Application	gateway.push.apple.com	443 TCP	APNS Push
Application	fcm.googleapis.com	443, 5228, 5229 TCP	Android Firebase Push
Application	*.googleusercontent.com	443 TCP	Android app store CDN
Application	*.ggpht.com	443 TCP	Android app store CDN

1.3.4 Third-party integrations

Source	Destination	Port	Note
Application	attest-api.secb2b.com	443 TCP	Samsung Knox license attestation
Application	gslb.secb2b.com	443 TCP	Samsung Knox license attestation
Application	eu-elm.secb2b.com	443 TCP	Samsung Knox
Application	eu-knox.secb2b.com	443 TCP	Samsung Knox
Application	eu-be-reseller.samsungknox.com	443 TCP	Samsung Knox
Application	login.microsoftonline.com, graph.microsoft.com	443 TCP	Azure integration
Application	api.zebra.com	443 TCP	Zebra API
Application	androiddeviceprovisioning.googleapis.com	443 TCP	Android Zero-Touch

Info

Do note if MDM system operates in networks protected by advanced protection features – both Google and Apple specify that traffic suspected of going through TLS inspection may be silently dropped.

<https://support.apple.com/en-us/HT210060>

<https://support.google.com/work/android/answer/10513641?hl=en>

 Info

Vendors often global CDNs with short name TTLs, possibly making IP-based whitelisting problematic. For specific IPs:

Apple: 17.0.0.0/8 Internet subnet

Google: published dynamic lists <https://www.gstatic.com/ipranges/cloud.json>, <https://www.gstatic.com/ipranges/goog.json>

2. Installation

Review *System requirements* before performing the installation and ensure following prerequisite steps have been prepared.

- Access to [support portal](#) account for downloading installation images
- Using support portal account, you have access to [MDM downloads repository](#)
- You have downloaded latest installer packages and transferred them to selected server
- Valid MDM License file obtained through business partner or through support portal
- Server has public DNS domain name assigned
- You have prepared SSL Certificate for chosen FQDN signed by commercial-grade CA

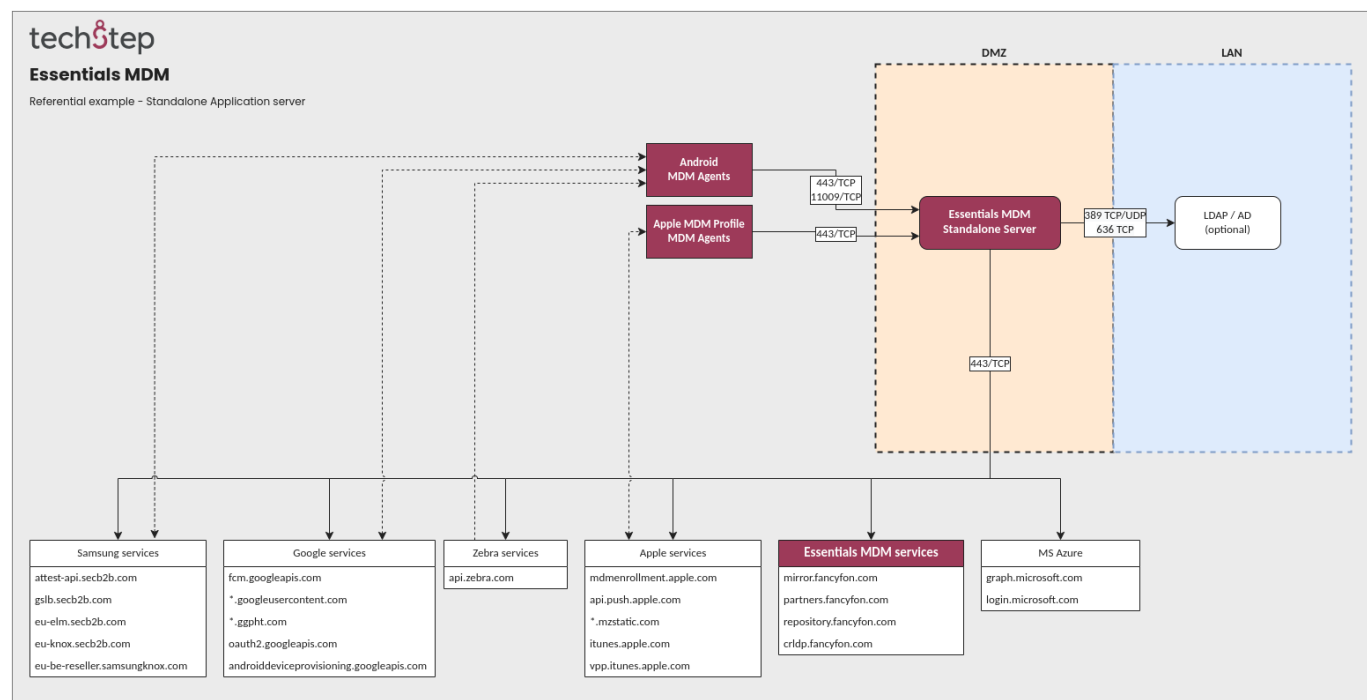
Note

Installation on airgapped systems without internet access is not possible. Apart from BaseOS and Appstream OS packages, Essentials MDM installer utility will need to download additional packages from service repositories.

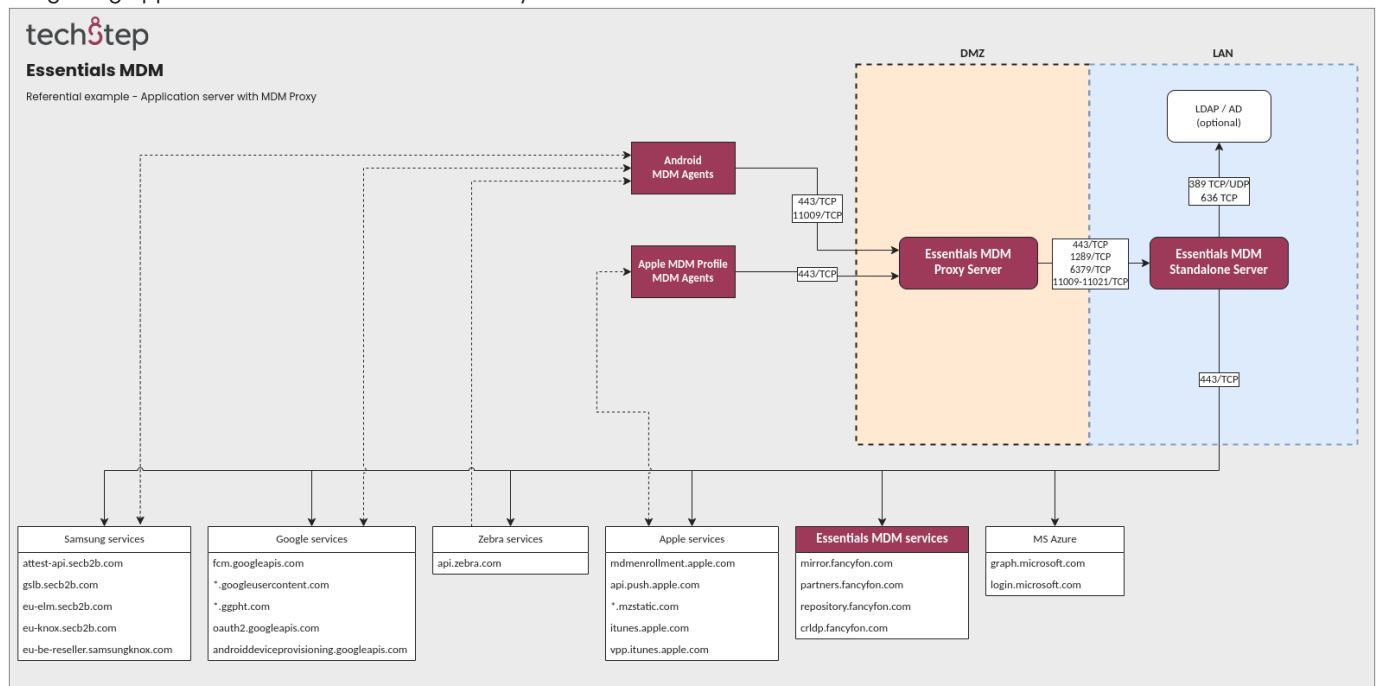
2.1 Referential deployment examples

Referential examples may serve as base for specific deployment planning.

Standalone application server – Suitable for standard deployments, provides complete MDM features.



Application server with Essentials MDM proxy – Deployed for environments that require to ensure DMZ/LAN boundaries before integrating Application server with Active Directory.



High Availability failover – Specialized setup outside of this document's scope. Used for deployments with increased fault-tolerance requirements, that warrant increased TCO due to additional machines and maintenance overhead.

2.2 Application server installation

Download latest installation image from [MDM downloads repository](#) and transfer to destination server.

```
# rpm --import https://mirror.fancyfon.com/RPM-GPG-KEY-mdm-installer
# dnf install /root/essentials-mdm-standalone-installer-...el8.x86_64.rpm
```

After base installation archive is installed, run `mdm-init` command to start interactive installer. Insert FQDN for the system – it needs DNS assigned name consistent with SSL Certificate Common Name and MDM License.

Next up, interactive questions for possible customization of installation process. For easiest installation, you should confirm default options with "YES" select **standard** database installation mode. After interactive questions, installer utility will take some time to download application dependencies, and initialize MDM for use.

After setup is finished, configure core application parameters using `famoc-config` graphical configurator:

- **Set admin's password** – Set password for built-in "admin" account for Web UI console. This account should be used to create first organization and "regular" administrator accounts within the organization.
- **Load license file** – File picker window will open. Use TAB and Arrow keys to navigate to location on disk where .fflic license file is placed.
- **System configuration > Server base settings** – Possibility to verify and correct server FQDN. If needed, e-mail sender name may also be customized.
- **System configuration > FAMOC SSL certificates > Load PEM/PFX file** – Load SSL certificates for the server. File picker window will open. Use TAB and Arrow keys to navigate to location on disk where certificates are storage. Option 'Create CA' may be used to generate temporary self-signed certificate. Select "Replace and discard" on interactive question. Take care to load full top-to-bottom certificate chain. While most browsers are capable of deduction of Root CAs, some Android devices are not and will need the server to present full CA chain.
- **FINISH & SAVE CONFIG** – Save changes and restart services.

Tip

If services `famoc.service` or `fflicense.service` fail to start after saving changes, re-check closely that *Server Base Settings > FQDN*, Common Name of server's SSL certificate and license are consistent to-the-letter. Problem is likely to be caused by inconsistency in Common Name. After correcting differences, run `systemctl restart fflicense famoc httpd`.

Finally, set DNS to point FQDN to MDM Server IP address, and verify correct login with "admin" account.

2.3 Proxy server installation

Download proxy installation image from [MDM downloads repository](#) with version corresponding to existing application server and transfer to target host.

```
# rpm --import https://mirror.fancyfon.com/RPM-GPG-KEY-mdm-installer
# dnf install /root/essentials-mdm-proxy-installer-...el8.x86_64.rpm
```

After rpm package installation, run `mdm-init` to start interactive installer. On prompt to insert FQDN for this system, insert exactly the same FQDN as in MDM Application server, consistent with license and SSL Certificate Common Name. Follow interactive questions, for easiest installation you should confirm default options with "YES"

Set basic parameters using `famoc-config` graphical configurator

System configuration > Proxy base settings > Application backend address – Set to ip address or resolvable hostname of application server.

System configuration > FAMOC SSL certificates > Load PEM/PFX file – Load SSL certificates for proxy. This needs to be exactly the same certificate chain as in application server. In doubt, you can copy `/opt/FAMOC/etc/proxy.pem` file from application server to proxy machine and load it into configurator.

System configuration > Advanced SSL configuration – Verify that enabled ciphers suites and SSL/TLS parameters are set exactly the same as in application server.

System configuration > FINISH & SAVE CONFIG – All changes will be applied after services reload. Ignore possible warnings, as this proxy is not yet recognized by application server.

On application server open `famoc-config` utility to configure new proxy server information.

System configuration > Server base settings > IPs of FAMOC secure proxies – insert IP address of newly installed proxy server.

System configuration > Advanced SSL configuration – Verify that cipher suites and SSL/TLS versions are consistent as with proxy server.

FINISH & SAVE CONFIG – Services will be reloaded with new configuration. Process may take some time as application server will synchronize static assets by pushing them to proxy server CDN.

Set DNS to point FQDN of the server to the Proxy IP address and verify UI Console login.

2.4 Post installation tasks

2.4.1 Firebase Cloud Messaging

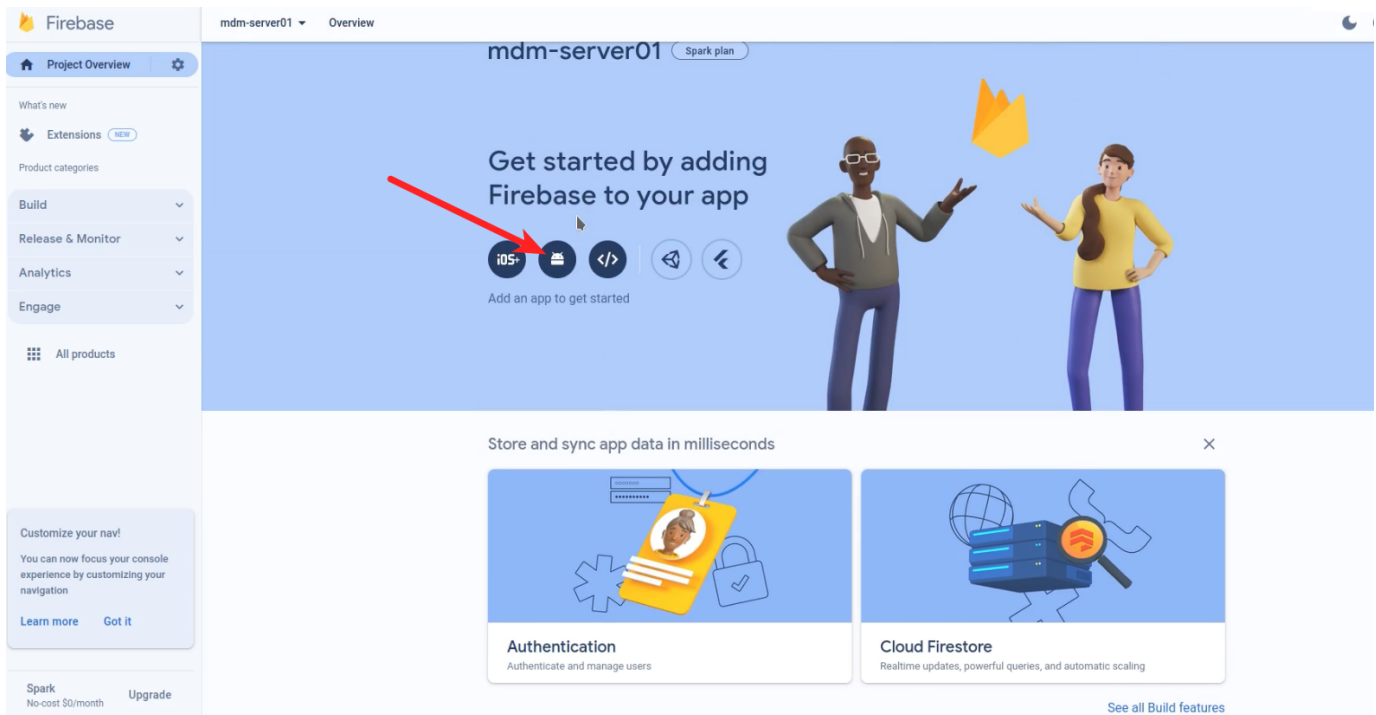
Baseline communication mechanism with managed Android devices is scheduled polling, with base agent app communicating to server at regular intervals set in it's policy. For responsive management, Push communication services should be used through integration with Firebase Cloud Messaging (FCM).

Tip

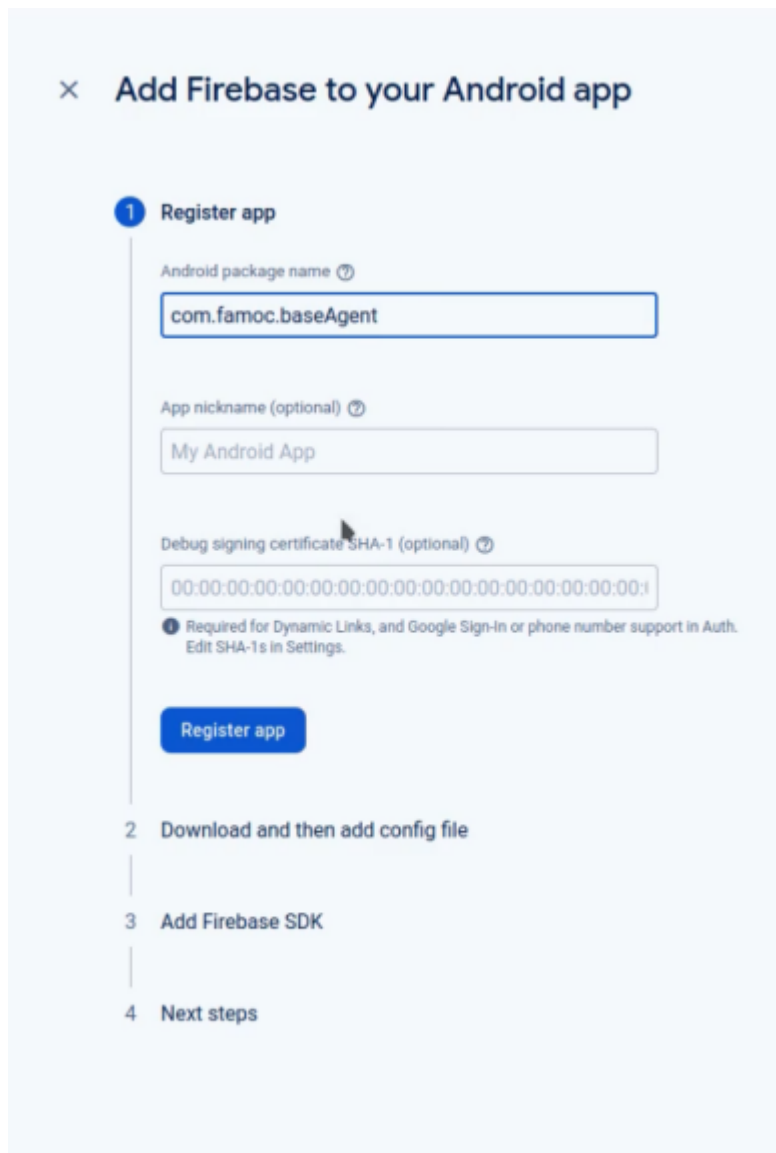
This procedure may be omitted if newly installed server is used only as migration target - existing FCM data will be copied over with backup from source system.

Create non-personal Google account for integration purposes - treat it as department resource and commit to secrets management.

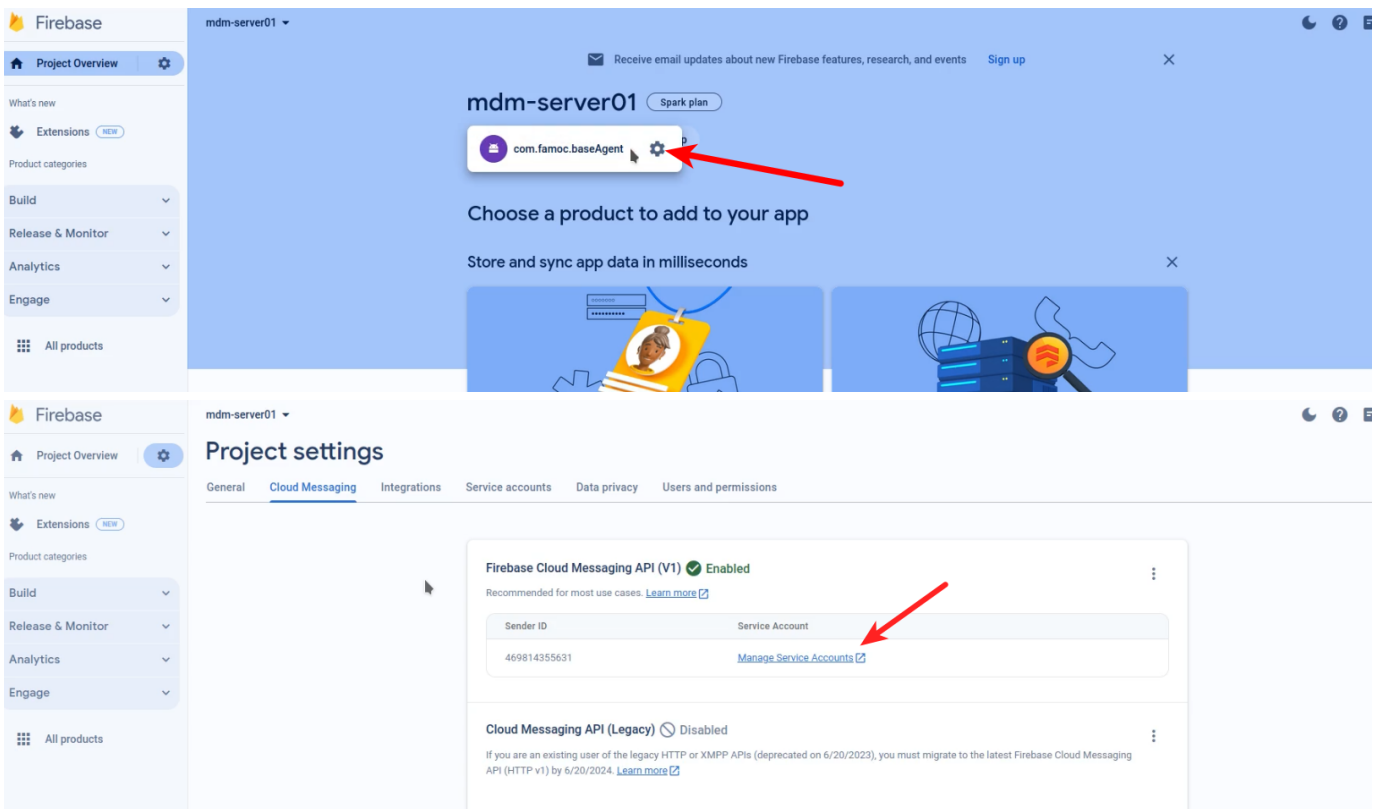
Sign in to <https://console.firebase.google.com/> and navigate to "create a project" on the main dashboard. Insert projects name, decide on analytics and wait for project to be initialized. You will be redirected to project dashboard. Select Android icon under "Add firebase to your app".



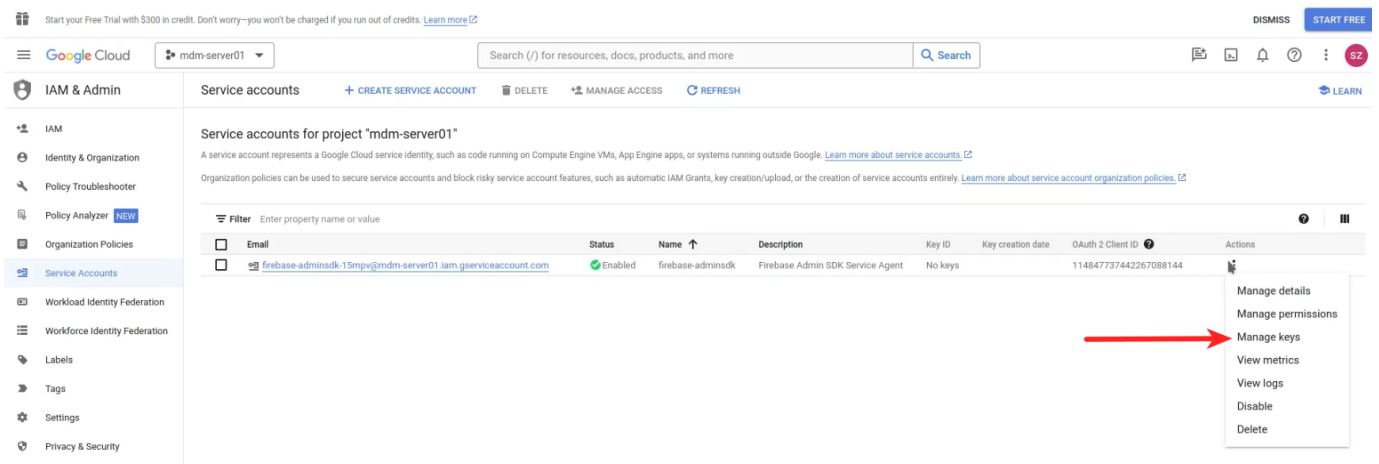
In Register app step, fill "Android package name" field with `com.famoc.baseAgent` and ignore all other fields. Save `google-service.json` file, which contains first part of configuration data for your MDM server.



Back at project dashboard, open `com.famoc.baseAgent` settings and navigate to "Cloud Messaging" tab. Go to "Manage Service Accounts".



You will be redirected to Google Cloud IAM console. Edit the only firebase account found, expand Actions bar and open "Manage keys" option. Add a new key with type: JSON and save it.



Transfer both keys to Application Server. Both JSON files downloaded from Firebase console will need to be clean from newlines. For each json config, do:

```
# cat google-services.json | tr -d '\n' > google-services.clean.json
# cat service-account.json | tr -d '\n' > service-account.clean.json
```

Print service account and copy both into other window for later pasting into the utility.

```
# cat service-account.clean.json
# echo
# cat google-services.clean.json
```

Go to `famoc-config` utility and navigate to **System configuration > Firebase Push service**. In *Google auth JSON* and right-click paste previously copied content of service-account. Using TAB keys navigate to *Paste JSON file* and paste content of google-services.

3. Updating Essentials MDM

We recommend maintaining regular upgrade schedule for system to receive access to new features and device models.

3.1 Standalone Server

Updates are performed using configuration utility `famoc-config`

Prerequisites:

- System license is valid
- MySQL root credentials need to be provided during upgrade step
- System has outgoing access to (<https://partners.fancyfon.com>) and (<https://repository.fancyfon.com>)

Run management utility: **famoc-config > System update**

System will access updates repository and present list of available updates. Upon selection, update utility asks for MySQL credentials, that need to be provided for process to start.

3.2 Server with Proxy

In case of installations using Essentials MDM Proxy, proxy servers need to be updated first.

Prerequisites:

- System license is valid
- MySQL root credentials need to be provided during upgrade step
- Both proxy and application servers have outgoing access to <https://partners.fancyfon.com> and <https://repository.fancyfon.com>

Starting on proxy server, select **famoc-config > System update**, select updates and wait till installation of all necessary packages is finished. Repeat procedure on application server.

4. Backup

Essentials MDM includes integrated utilities for data backup and restore, that may be used on their own or as a supplement to external VM-layer tools. Backup and restore utilities are also used for migrating data from host to host.

To create single backup archive on application server, use management utility. Without specifying any additional options, an uncompressed tar archive will be created in `/home/backup` directory.

```
# essentials-mdm-config backup:create --help
# essentials-mdm-config backup:create
# essentials-mdm-config backup:create --compression=4
# essentials-mdm-config backup:create --compression=4 --backup-dest-dir=/mnt/external-storage
```

To restore backup, find location of previously created archive and prepare MySQL root credentials. Note that **All existing data will be overwritten**.

```
# essentials-mdm-config backup:restore --help
# essentials-mdm-config backup:restore --dbuser='root' --dbpass='mysqlroot' /home/backup/famocbackup.1.famapp-j-01.tar.zst
```

Tip

Backup and restore utilities may take up to several hours on systems with hundreds of gigabytes of data. It is recommended to invoke the scripts from within `tmux` or `screen` utility to ensure scripts keep running in case of network outage breaking SSH session.

To enable regular scheduled backups, backup script may be invoked placed in cron or using convenience helper from `famoc-config` utility in **System configuration > FAMOC Server data backup**

5. Migration

MDM migration is based on setting up fresh new MDM installation, then restoring archived database and files from old environment and switching network traffic for system FQDN to point to new environment. Backup and restore utilities included with administrative tools are used for migration process.

Steps outlined in the procedure will henceforth refer to MDM installation from which data will be taken as **source** and new MDM installation to which we migrate is **target**, and that installation consisting of single standalone server is migrated.

5.1 Preparation

Install same version of Essentials MDM on new **target** server. Copy license file `/opt/FAMOC/etc/licenses/x.license` and SSL certificates (`/opt/FAMOC/etc/proxy.pem`) from **source** to **target** and load them using `famoc-config` configuration utility. Make sure that free disk space of **target** is at least two times as big as used disk space on **source**, to account for holding actual data and working directory that will store temporary copies during backup restoration.

Check disk space on **source** server. It has to contain enough free space to hold archived backup of existing data. If system downtime needs to be minimized, there are several data directories with files that may be cleaned up in advance to speed up migration process.

- `/var/backup-files/logs/`
- `/var/www/apps/raccess/sessions/`
- `/var/application-files/`
- `/var/backup-files/backup-files/tmp-upload/`

If at this point free space on **source** is still not able to hold doubled amount of data, you will need to expand or attach additional storage to the server.

Install `rsync` utility on both source and target host for efficient archive copy.

```
source# dnf install -y rsync
target# dnf install -y rsync
```

Inspect files in `/etc/pki/ca-trust/source/anchors/` and `/etc/postfix` for any custom CAs or configurations that may need to be copied over to the new system.

5.2 Migration procedure

Migration involves MDM downtime, during which admin UI console, devices communication and webservice API will be unavailable. Downtime schedule needs to be considered especially when migrating server containing lots of stored data.

At the beginning of actual migration process, stop services capable of modifying application state on **source** via `systemctl stop httpd famoc crond`. To maintain data consistency, from this point on services must not be started again unless an issue with new system during migration forces you to cancel the process and fall back to **source**.

Create migration directories on both servers

```
source# mkdir -p /data/migration
source# mkdir -p /data/workdir
```

```
target# mkdir -p /data/migration
target# mkdir -p /data/workdir
```

Creating backup archive on **source** host. It is strongly recommended to run backup script within detachable session offered by `tmux` or `screen` utilities available in system repositories, to secure continuity of backup process in case of network outage disconnecting the SSH session.

```
source# essentials-mdm-config backup:create --help
source# essentials-mdm-config backup:create --backup-dest-dir=/data/migration -u root -p 'mysqlrootpassword'
```

After backup is complete, copy the archive from **source** to **target**

```
source# rsync -hvP --append-verify /data/migration/archive.tar.gz root@target:/data/migration
```

After transfer is complete, run restore utility on the archive.

```
target# essentials-mdm-config backup:restore --help
target# essentials-mdm-config backup:restore /data/migration/archive.tar.gz --dbuser="root" --dbpass="mysqlrootpassword"
```

After restore script is complete, verify base system functionalities – status of `httpd`, `php-fpm`, `mysqld`, `famoc`, `fflicense` services. If networking setup allows to single out UI access only from internal network, verify log in and visible data in UI.

```
target# systemctl status httpd php-fpm mysqld famoc fflicense ff-logd docker
```

When all baseline functionalities are verified to work correctly, DNS or NAT may be directed to the new server.

Verify incoming connections from devices with `tail -f /var/log/famoc/access_mobile.log`. Verify Android/iOS push communication.

Note

Take into account additional checks that may be needed if old system was configured to use any of:

- AD/LDAP synchronization
- MS Certificate Authority Addon
- MDM EAS Proxy
- Outgoing proxy