[Fire.ONE Installation Guide]

SSNC Co.,Ltd.

Contents

Page

I. OS Install

OS: ubuntu server 24.04 LTS

Officially, we do not support OS installation.

Configuration changes are required:

- ufw (Optional)

- If needed, please allow the required ports.

  sudo ufw status # checking current status

  sudo ufw allow 8000/tcp # (FireONE Diagnosis)

  sudo ufw allow 8190/tcp # (FireONE Web Module)

  sudo ufw allow 8191/tcp # (FireONE Common Module)

  sudo ufw allow 8192/tcp # (FireONE Request/Approval Module)

  sudo ufw allow 8193/tcp # (FireONE Push Module)

  sudo ufw allow 8194/tcp # (FireONE Policy Module)

  sudo ufw allow 8199/tcp # (FwSync)

  sudo ufw allow 5672/tcp # (Rabbit MQ)

  sudo ufw allow 15672/tcp # (Rabbit MQ Management)

  sudo ufw allow 5432/tcp # (postgresql)

  sudo ufw allow ssh # (ssh)

  sudo ufw enable #

  sudo ufw reload # (ufw restart)

II. Ubuntu - DB(PostgreSQL) Installation

1. Install postgresql-16

- update apt

  sudo apt update

```
gsroot@gscert:~$ sudo apt update
[sudo] password for gsroot:
Hit:1 http://kr.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://kr.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://kr.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
55 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

- install postgresql-16 :

  sudo apt install postgresql-16 postgresql-client-16

```
gsroot@gscert:~$ sudo apt install postgresql-16 postgresql-client-16
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpq5 libtypes-serialiser-perl postgresql-client-common postgresql-common ssl-cert
Suggested packages:
  postgresql-doc-16
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpq5 libtypes-serialiser-perl postgresql-16 postgresql-client-16 postgresql-client-common postgresql-common ssl-cert
0 upgraded, 11 newly installed, 0 to remove and 55 not upgraded.
Need to get 43.6 MB of archives.
After this operation, 175 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

  - Press 'Y'

- Check the version after installation.

  psql -V

```
gsroot@gscert:~$ psql -V
psql (PostgreSQL) 16.11 (Ubuntu 16.11-0ubuntu0.24.04.1)
```

2. Create a User in postgresql

- Start postgresql

  sudo -u postgres psql

```
gsroot@gscert:~$ sudo -u postgres psql
psql (16.11 (Ubuntu 16.11-0ubuntu0.24.04.1))
Type "help" for help.
```

- Create a User in DB

  CREATE ROLE argosuser WITH LOGIN SUPERUSER PASSWORD 'Fridaynight1!';

```
postgres=# CREATE ROLE argosuser WITH LOGIN SUPERUSER PASSWORD 'Fridaynight1!';
CREATE ROLE
```

- Create a Database and make owner

CREATE DATABASE argosdb OWNER argosuser;

```
postgres=# CREATE DATABASE argosdb OWNER argosuser;
CREATE DATABASE
```

- list in DB

\l

```
postgres=# \l
                                                    List of databases
   Name    |   Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+-----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 argosdb   | argosuser | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 postgres  | postgres  | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres  | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |           |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres  | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |           |          |                 |             |             |            |           | postgres=CTc/postgres
(4 rows)
```

- User in DB

\du

```
postgres=# \du
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 argosuser | Superuser
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
```

- exit to shell

Exit

```
postgres=# exit
gsroot@gscert:~$
```

- ufw (optional)

- if needed:

sudo ufw status #

sudo ufw allow 5432/tcp #

sudo ufw allow ssh #

sudo ufw enable #

sudo ufw reload #

3. config

sudo vi /etc/postgresql/16/main/postgresql.conf

- find and change "listen_address"
- listen_addresses = '*'
- save and quit

```
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

data_directory = '/var/lib/postgresql/16/main'          # use data in another directory
                                        # (change requires restart)
hba_file = '/etc/postgresql/16/main/pg_hba.conf'        # host-based authentication file
                                        # (change requires restart)
ident_file = '/etc/postgresql/16/main/pg_ident.conf'    # ident configuration file
                                        # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/16-main.pid'                   # write an extra PID file
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#reserved_connections = 0               # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                                        # (change requires restart)

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0                # TCP_KEEPIDLE, in seconds;
                                        # 0 selects the system default
#tcp_keepalives_interval = 0            # TCP_KEEPINTVL, in seconds;
                                        # 0 selects the system default
#tcp_keepalives_count = 0               # TCP_KEEPCNT;
                                        # 0 selects the system default
#tcp_user_timeout = 0                   # TCP_USER_TIMEOUT, in milliseconds;
                                        # 0 selects the system default

#client_connection_check_interval = 0   # time between checks for client
                                        # disconnection while running queries;
                                        # 0 for never

:wq!
```

sudo vi /etc/postgresql/16/main/pg_hba.conf

- change to config as like below text. (Please comply with the company security policies)
    - # "local" is for Unix domain socket connections only

- local all all md5
- # IPv4 local connections
- host all all 0.0.0.0/0 md5

```
# "local" is for Unix domain socket connections only
local   all             all                                     md5
# IPv4 local connections:
host    all             all             0.0.0.0/0               md5
```

- save and quit.
- Restart Postgresql

  sudo systemctl restart postgresql.service

  sudo systemctl status postgresql.service

```
gsroot@gscert:~$ sudo systemctl status postgresql.service
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Wed 2026-01-14 08:49:59 UTC; 7s ago
    Process: 40817 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 40817 (code=exited, status=0/SUCCESS)
        CPU: 847us

Jan 14 08:49:59 gscert systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jan 14 08:49:59 gscert systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
gsroot@gscert:~$
```

4. DB dump and restore (cli)

- Connect to DB server.
- follow the command:

  sudo pg_dump -U argosuser -F c -b -v -f argosdb.backup argosdb

- press ENTER when "PASSWORD: "

```
root@ssnc:~# sudo pg_dump -U argosuser -F c -b -v -f argosdb.backup argosdb
Password:
pg_dump: last built-in OID is 16383
pg_dump: reading extensions
pg_dump: identifying extension members
pg_dump: reading schemas
pg_dump: reading user-defined tables
pg_dump: reading user-defined functions
pg_dump: reading user-defined types
pg_dump: reading procedural languages
pg_dump: reading user-defined aggregate functions
pg_dump: reading user-defined operators
pg_dump: reading user-defined access methods
pg_dump: reading user-defined operator classes
pg_dump: reading user-defined operator families
pg_dump: reading user-defined text search parsers
pg_dump: reading user-defined text search templates
pg_dump: reading user-defined text search dictionaries
pg_dump: reading user-defined text search configurations
pg_dump: reading user-defined foreign-data wrappers
pg_dump: reading user-defined foreign servers
pg_dump: reading default privileges
pg_dump: reading user-defined collations
pg_dump: reading user-defined conversions
pg_dump: reading type casts
pg_dump: reading transforms
pg_dump: reading table inheritance information
pg_dump: reading event triggers
pg_dump: finding extension tables
pg_dump: finding inheritance relationships
pg_dump: reading column info for interesting tables
pg_dump: finding table default expressions
pg_dump: finding table check constraints
pg_dump: flagging inherited columns in subtables
pg_dump: reading partitioning data
pg_dump: reading indexes
pg_dump: flagging indexes in partitioned tables
pg_dump: reading extended statistics
pg_dump: reading constraints
pg_dump: reading triggers
pg_dump: reading rewrite rules
pg_dump: reading policies
pg_dump: reading row-level security policies
pg_dump: reading publications
pg_dump: reading publication membership of tables
pg_dump: reading publication membership of schemas
pg_dump: reading subscriptions
pg_dump: reading large objects
pg_dump: reading dependency data
pg_dump: saving encoding = UTF8
pg_dump: saving standard_conforming_strings = on
pg_dump: saving search_path =
pg_dump: dumping contents of table "user.ag_grades"
pg_dump: dumping contents of table "user.ag_levels"
pg_dump: dumping contents of table "user.ag_roles"
pg_dump: dumping contents of table "user.ag_roles_menu"
pg_dump: dumping contents of table "user.ag_users"
pg_dump: dumping contents of table "user.ag_web_users"
pg_dump: dumping contents of table "user.ag_workgroup"
pg_dump: dumping contents of table "user.ag_workgroup_web_users"
root@ssnc:~#
```

ls -lh *.backup

```
root@ssnc:~# ls -lh *.backup
-rw-r--r-- 1 root root 102M Jan 14 18:09 argosdb.backup
root@ssnc:~#
```

- Restore
- Connect to Server. (Copy file to server with scp)

  scp (filename) (account)@(Server IP):(Path)

  ex) scp argosdb.backup gsroot@192.168.7.101:/home/gsroot

```
root@ssnc:~# scp argosdb.backup gsroot@192.168.7.101:/home/gsroot
The authenticity of host '192.168.7.101 (192.168.7.101)' can't be established.
ED25519 key fingerprint is SHA256:mkOroCdYFNr8MnS9mobAQF/kgOsk+E28yXku/BythYQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.7.101' (ED25519) to the list of known hosts.
gsroot@192.168.7.101's password:
argosdb.backup
root@ssnc:~#
```

- Installation psql Extension

  CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

  CREATE EXTENSION IF NOT EXISTS "pgcrypto";

```
postgres=# \dx
                 List of installed extensions
  Name   | Version |   Schema   |         Description
---------+---------+------------+------------------------------
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(1 row)

postgres=# CREATE EXTENSION IF NOT EXISTS "pgcrypto";
CREATE EXTENSION
postgres=# CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
CREATE EXTENSION
```

- DB restore

  pg_restore -U (DB User) -d (Target DB) (Dumpfile)

  ex) pg_restore -U argoususer -d argosdb argosdb.backup

#Check

\dn # Schema list

\dt # All tables in Schema

```
argosdb=# \dn
      List of schemas
      Name      |   Owner
----------------+-----------
 approval       | argosuser
 argos_chatbot  | argosuser
 argos_common   | argosuser
 argos_firewall | argosuser
 compliance     | argosuser
 fpiq           | argosuser
 menu           | argosuser
 notification   | argosuser
 public         | argosuser
 temp           | argosuser
 topology       | argosuser
 user           | argosuser
(12 rows)

argosdb=# \dt schema_name.*
Did not find any relation named "schema_name.*".
argosdb=# \dt argos_firewall.*
                    List of relations
     Schema     |          Name           | Type  |   Owner
----------------+-------------------------+-------+-----------
 argos_firewall | argos_address           | table | argosuser
 argos_firewall | argos_address_group     | table | argosuser
 argos_firewall | argos_address_temp      | table | argosuser
 argos_firewall | argos_analyze           | table | argosuser
 argos_firewall | argos_analyze_temp      | table | argosuser
 argos_firewall | argos_application       | table | argosuser
 argos_firewall | argos_application_group | table | argosuser
 argos_firewall | argos_apply             | table | argosuser
 argos_firewall | argos_apply_member      | table | argosuser
 argos_firewall | argos_apply_rulebase    | table | argosuser
 argos_firewall | argos_compliance        | table | argosuser
 argos_firewall | argos_compliance_object | table | argosuser
 argos_firewall | argos_condition         | table | argosuser
 argos_firewall | argos_firewall_list     | table | argosuser
 argos_firewall | argos_fw_config         | table | argosuser
 argos_firewall | argos_interface         | table | argosuser
 argos_firewall | argos_log               | table | argosuser
 argos_firewall | argos_router            | table | argosuser
 argos_firewall | argos_rulebase          | table | argosuser
 argos_firewall | argos_schedule          | table | argosuser
 argos_firewall | argos_service           | table | argosuser
 argos_firewall | argos_service_group     | table | argosuser
 argos_firewall | argos_swhub             | table | argosuser
 argos_firewall | argos_swhub_router      | table | argosuser
 argos_firewall | argos_sync_log          | table | argosuser
 argos_firewall | argos_syslog            | table | argosuser
 argos_firewall | argos_syslogex          | table | argosuser
 argos_firewall | argos_user              | table | argosuser
 argos_firewall | argos_user_group        | table | argosuser
(29 rows)

argosdb=#
```

III. RabbitMQ Installation

📁 Basic Information

1. config file location

- main config file : /etc/rabbitmq/rabbitmq.conf
- environment config file : /etc/rabbitmq/rabbitmq-env.conf

2. log file location

- /var/log/rabbitmq/

3. Required port

- 5672 :AMQP client connections
- 15672 :management UI (HTTP)
- 25672 :Erlang Distribution

📁 Linux Server

1. System Update

```
sudo dnf update -y
```

2. Erlang installation

For using RabbitMQ, it is required to use Erlang:

```
# EPEL
sudo dnf install -y epel-release
# Erlang install
sudo dnf install -y erlang
```

3. Add RabbitMQ Repo

```
#
sudo rm -f /etc/yum.repos.d/rabbitmq.repo
# RabbitMQ
curl -s https://packagecloud.io/install/repositories/rabbitmq/rabbitmq-server/script.rpm.sh | sudo bash
```

4. Install RabbitMQ

```
sudo dnf install -y rabbitmq-server
```

5. Start and enable service RabbitMQ

```
# Start service
sudo systemctl start rabbitmq-server
```

```
# Enable service
sudo systemctl enable rabbitmq-server

# status check
sudo systemctl status rabbitmq-server
```

6. Firewall Config in linux

```
# open RabbitMQ port
sudo firewall-cmd --permanent --add-port=5672/tcp # AMQP port
sudo firewall-cmd --permanent --add-port=15672/tcp # Management web port
sudo firewall-cmd –reload
```

7. enable management plugin

```
sudo rabbitmq-plugins enable rabbitmq_management
```

8. Create user for management

```
# Delete guest user
sudo rabbitmqctl delete_user guest

# Create new user
sudo rabbitmqctl add_user {breezeway} {Fridaynight1!}

# grant role
sudo rabbitmqctl set_user_tags {breezeway} administrator

# grant all
sudo rabbitmqctl set_permissions -p / {breezeway} ".*" ".*" ".*"
```

9. Check all

```
# RabbitMQ
sudo rabbitmqctl status
# User list
sudo rabbitmqctl list_users
# Queue list
sudo rabbitmqctl list_queues
```

IV. Fire.ONE - Platform Installation

1. Install JDK

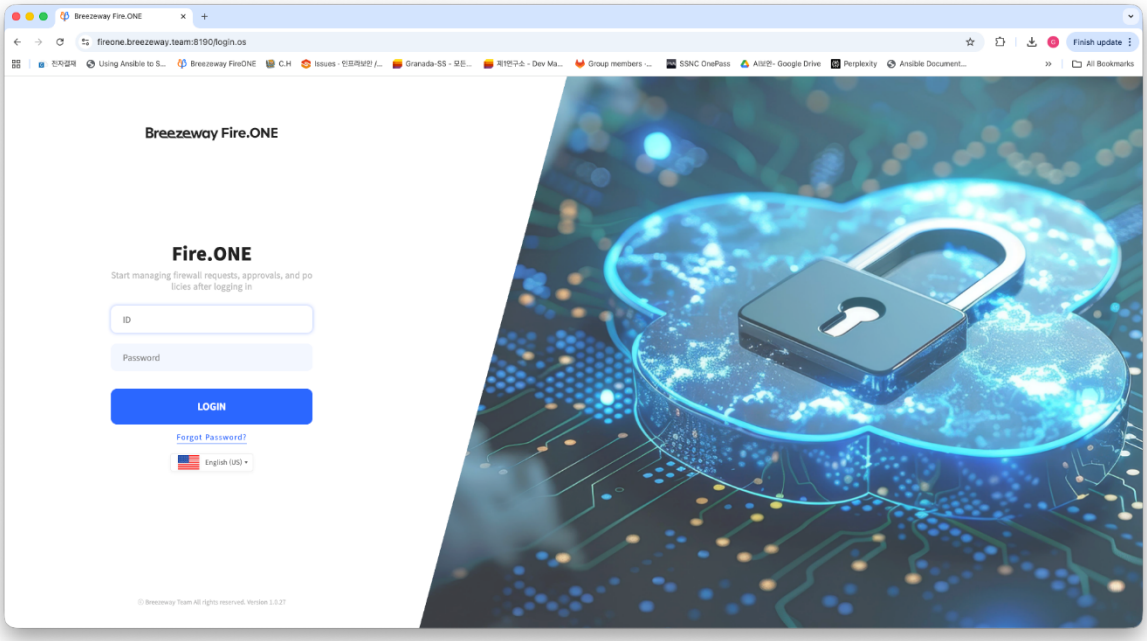
```
sudo apt update
sudo apt list openjdk-17-jdk

# check the results

sudo install -y openjdk-17-jdk
```

2. Fire.ONE JARs

```
# If you need to trace the logs, run it without nohup.
# you can see the command in argos_run.sh
# FIREONE Compress and Transfer
cd /FIREONE-API
sudo tar -czvf FIREONE-API.tar.gz /FIREONE-API
scp FIREONE-API.tar.gz gsroot@192.168.7.101:/home/gsroot/
#
mv FIREONE-API.tar.gz /
#
sudo tar -xzvf FIREONE-API.tar.gz -C /
# execute the jar(s)
cd /FIREONE-API
sudo ./argos_run.노
```

3. Connect to web

- http://ip:8190



4. logs

```
# move to logs folder
cd /FIREONE-API/logs

-rw-r--r--. 1 root root        2675 Mar 12 15:09 argosAlarm.log
-rw-r--r--. 1 root root       19980 Mar 11 10:08 argosApproval.log
-rw-r--r--. 1 root root     1907529 Mar 12 15:46 argosCommon.log
-rw-r--r--. 1 root root 10182423 Mar 12 15:45 argosFwSync.log
-rw-r--r--. 1 root root     1205998 Mar 12 15:45 argosPolicy.log
-rw-r--r--. 1 root root        4681 Mar 11 10:09 argosPush.log
-rw-r--r--. 1 root root       26737 Mar 12 15:29 argosWeb.log
```

```
[ssnc@ssnc:/FIREONE-API$ cd logs                                                ]
[ssnc@ssnc:/FIREONE-API/logs$ ll                                                ]
total 28668
drwxr-xr-x 2 root root    4096 Mar 12 00:26 ./
drwxr-xr-x 4 root root    4096 Mar 10 17:54 ../
-rw-r--r-- 1 root root 3084985 Mar 10 17:55 argosAlarm.log
-rw-r--r-- 1 root root   99800 Mar 10 17:55 argosApproval.2026-03-10-0.log
-rw-r--r-- 1 root root    4252 Mar 11 10:44 argosApproval.log
-rw-r--r-- 1 root root 5995821 Mar 10 23:59 argosCommon.2026-03-10-0.log
-rw-r--r-- 1 root root 2449024 Mar 11 23:59 argosCommon.2026-03-11-0.log
-rw-r--r-- 1 root root 1456019 Mar 12 15:51 argosCommon.log
-rw-r--r-- 1 root root 3672582 Mar 10 23:59 argosFwSync.2026-03-10-0.log
-rw-r--r-- 1 root root 2901822 Mar 11 23:59 argosFwSync.2026-03-11-0.log
-rw-r--r-- 1 root root 1881948 Mar 12 15:51 argosFwSync.log
-rw-r--r-- 1 root root 2396527 Mar 10 23:59 argosPolicy.2026-03-10-0.log
-rw-r--r-- 1 root root 2148503 Mar 11 23:59 argosPolicy.2026-03-11-0.log
-rw-r--r-- 1 root root 1439230 Mar 12 15:51 argosPolicy.log
-rw-r--r-- 1 root root   18608 Mar 10 17:55 argosPush.log
-rw-r--r-- 1 root root 1633228 Mar 10 23:56 argosWeb.2026-03-10-0.log
-rw-r--r-- 1 root root   70605 Mar 11 23:56 argosWeb.2026-03-11-0.log
-rw-r--r-- 1 root root   18941 Mar 12 15:26 argosWeb.log
```

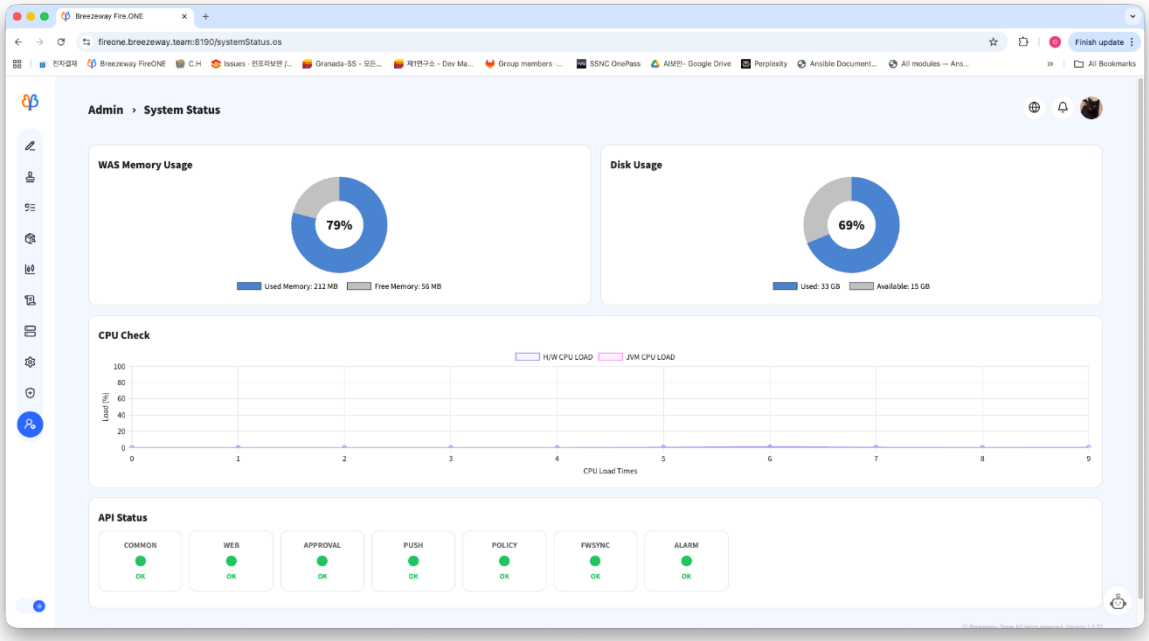
5. Check the API Services

after Login.

Menu > Admin > System Status

Green Circle: Working Fine.

RED Circle : Not Working. You must check the API.

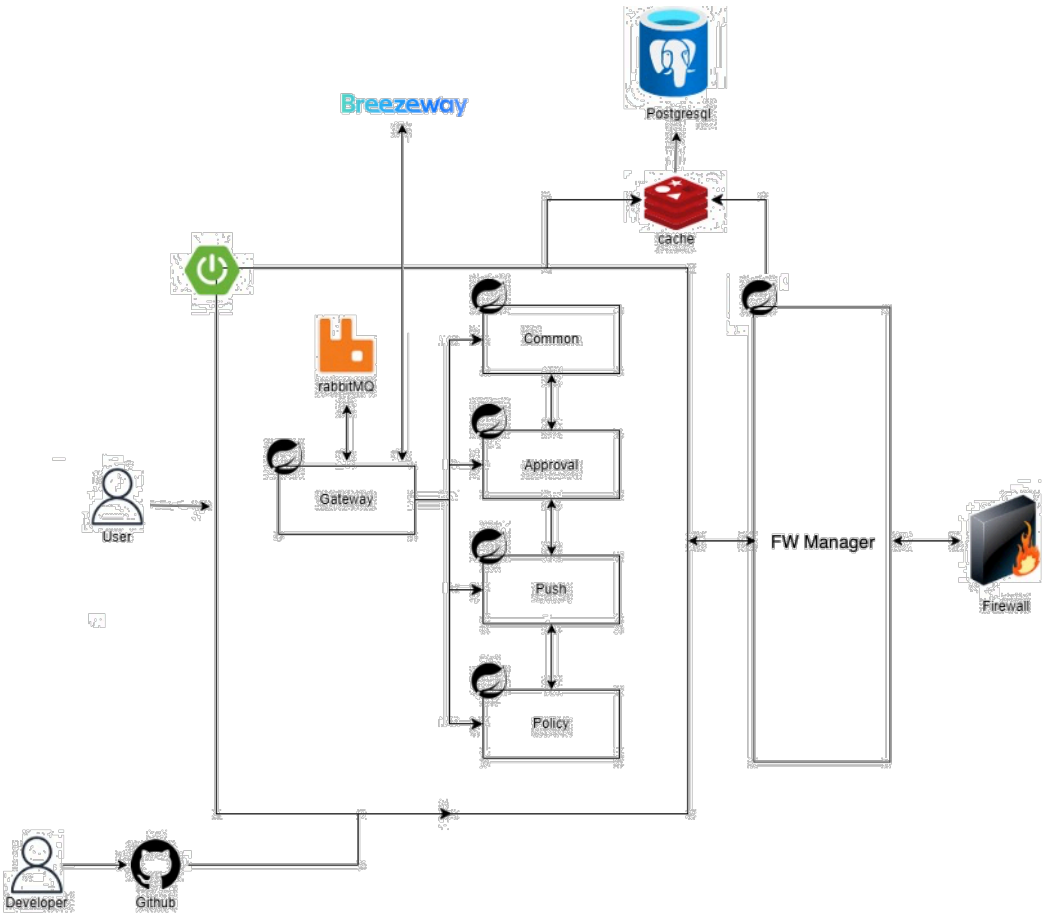


Appendix.

Fire.ONE Architecture

Fire.ONE Platform : Request/Approval/Push module

Fire.ONE Diagnostic : Diagnostic module



Prerequisites (Server infra)

For a smooth installation, an online network connection is required.
In an offline environment, all installation images must be prepared in advance.

Type	Requirements	
	Fire.ONE Platform	**Fire.ONE Diagnostics**
CPU		
Memory		
DISK		
Network	100Mbps or higher	1Gbps or higher
OS	Ubuntu 24 or higher (officially) Redhat 9 or higher, Rocky (not tested)	
JDK	OpenJDK 17	-
DB	PostgreSQL 14 or higher	-
RABBIT MQ	3.13.x or higher (latest version is 4.2.4 on Feb 17th, 2026)	-
Node Engine	-	node v24.x or Higher
Elasticsearch	-	9.x or higher (latest version is 9.3.1 on Mar 12th, 2026)
Logstash	-	9.x or higher (latest version is 9.3.1 on Mar 12th, 2026)
Kibana	-	9.x or higher (latest version is 9.3.1 on Mar 12th, 2026)
Fleet	-	9.x or higher (latest version is 9.3.1 on Mar 12th, 2026)
pnpm (for Next.js worspace on Node)	-	10.x or higher

Fire.ONE License activation

fire.one.license.json

Shell

```
{
  "customerId": "CUME260301",
  "customerName": "RASINFOTECH",
  "productId": "Fire.ONE",
  "issuedDate": "2026-03-12",
  "expireDate": "2027-04-12",
  "maxUsers": 10,
  "maxFwDevices": 5,
  "moduleMask": 15,
  "checkText": "6jbCupHyA8OIdpJgKlNV3Q",
  "signedKey": "dQEP1WqmGZnxMqEkpm_D7xM5M2kOMwmWql8jgpLLIfle1Pt6CqnzNBszHvZpRsFxSGZ_WbjLTr3Yx-LEt1zj-7evDE4kEy819lvhBdVvJDodFkXfwB_SJ_kAkgCKKO7IMEZ33mGftCCoSiSGjw3dJHSU8mDJ7CswQdA_SPnTlaw_rSdCVZhwQ2XRCeShvfsj1mRDXe9a3gGezoplybpZJ7fe2__ehf7sGN2y3Dqinj7fItHf37VFIQ-JC8xbp5hvSJkM-66lXlNLExCKvmWbLXMXWpGeh55yvVrt5jeBQfaYBBKyPVsTULDjKKHdWrPiLPkEkumjy4-h4PawE_WxQA"
}
```

We provided above JSON file **fire.one.license.json** related to license activation.
Please, Copy JSON file inside the **/FIREONE-API** folder.