

# MISP DEPLOYMENT

## SOME BASIC GUIDELINES

CIRCL / TEAM MISP PROJECT



CIISI-IE DUBLIN 2024

2024-07-08

## MISP Deployment

MISP DEPLOYMENT

SOME BASIC GUIDELINES

CIRCL / TEAM MISP PROJECT



CIISI-IE DUBLIN 2024

- **Deployment types**
- **Distro choice**
- **Hardware specs**
- **Authentication**
- Other considerations - **settings, gotchas**

2024-07-08

## MISP Deployment

### └─ MISP deployment considerations

- Deployment types
- Distro choice
- Hardware specs
- Authentication
- Other considerations - **settings, gotchas**

- Native install
  - ▶ Manual
  - ▶ One liner script - INSTALL.sh  
<https://github.com/MISP/MISP/tree/2.4/INSTALL>
- MISP VM  
<https://www.circl.lu/misp-images/latest/>
- Docker
- RPM maintained by SWITCH  
<https://github.com/amuehlen/MISP-RPM>
- Cloud provider images  
<https://github.com/MISP/misp-cloud>

### └─ Deployment types

- Native install
  - ▶ Manual
    - ▶ One liner script - INSTALL.sh  
<https://github.com/MISP/MISP/tree/2.4/INSTALL>
- MISP VM  
<https://www.circl.lu/misp-images/latest/>
- Docker
- RPM maintained by SWITCH  
<https://github.com/amuehlen/MISP-RPM>
- Cloud provider images  
<https://github.com/MISP/misp-cloud>

- Ostefano's Docker instance (x86-64 (AMD64) and ARM64 (M1)) <https://github.com/ostefano/docker-misp>
  - ▶ <https://blogs.vmware.com/security/2023/01/how-to-deploy-a-threat-intelligence-platform-in-y.html>
- National Cyber and Information Security Agency of the Czech Republic <https://github.com/NUKIB/misp>
- CoolAcid's MISP images <https://github.com/coolacid/docker-misp>
- MISP-docker by XME <https://github.com/MISP/misp-docker>
- docker-misp by Harvard security <https://github.com/MISP/docker-misp>

### └─ Docker options

- Ostefano's Docker instance (x86-64 (AMD64) and ARM64 (M1)) <https://github.com/ostefano/docker-misp>
  - ▶ <https://blogs.vmware.com/security/2023/01/how-to-deploy-a-threat-intelligence-platform-in-y.html>
- National Cyber and Information Security Agency of the Czech Republic <https://github.com/NUKIB/misp>
- CoolAcid's MISP images <https://github.com/coolacid/docker-misp>
- MISP-docker by XME <https://github.com/MISP/misp-docker>
- docker-misp by Harvard security <https://github.com/MISP/docker-misp>

- Ubuntu 22.04 (20.04 will also work)
  - ▶ Our target platform
  - ▶ Our CI target
  - ▶ Use this unless you are absolutely forced not to
  - ▶ This is the platform we can support you with!
- CentOS 7
  - ▶ Annoying to operate
  - ▶ Less tested, though used by many
  - ▶ CentOS is dead. Consider other options
- RHEL 7
  - ▶ Same annoyance as CentOS in general
  - ▶ We test against CentOS in general, some assembly may be required

### └─ Distro options

- Ubuntu 22.04 (20.04 will also work)
  - ▶ Our target platform
  - ▶ Our CI target
  - ▶ Use this unless you are absolutely forced not to
  - ▶ This is the platform we can support you with!
- CentOS 7
  - ▶ Annoying to operate
  - ▶ Less tested, though used by many
  - ▶ CentOS is dead. Consider other options
- RHEL 7
  - ▶ Same annoyance as CentOS in general
  - ▶ We test against CentOS in general, some assembly may be required

- No firm recommendations, it's highly usage dependent
- It's better to go a bit over what you need than under
- **SSDs** are massively beneficial
- Let's look at what affects specs and some sample configurations

### └ Hardware specs

- No firm recommendations, it's highly usage dependent
- It's better to go a bit over what you need than under
- **SSDs** are massively beneficial
- Let's look at what affects specs and some sample configurations

- What are the factors that can impact my performance?
  - ▶ Clustering of the data (how many datapoints / event?) (RAM, disk speed)
  - ▶ Correlation (RAM, disk speed, disk space)
    - Consider blocking overtly correlating values from doing so
    - Feed ingestion strategy is crucial
  - ▶ Over-contextualisation (RAM, disk speed)
    - Tag/attach galaxies to the event instead of each attribute when possible

### └ Hardware considerations

- What are the factors that can impact my performance?
  - ▶ Clustering of the data (how many datapoints / event?) (RAM, disk speed)
  - ▶ Correlation (RAM, disk speed, disk space)
    - Consider blocking overtly correlating values from doing so
    - Feed ingestion strategy is crucial
  - ▶ Over-contextualisation (RAM, disk speed)
    - Tag/attach galaxies to the event instead of each attribute when possible

- What are the factors that can impact my performance?
  - ▶ Number of users that are active at any given time (RAM, CPU, disk speed)
  - ▶ Logging strategy (Disk space)
  - ▶ API users especially with heavy searches (substring searches for example) (RAM, CPU, Disk speed)

### └─ Hardware considerations - continues

- What are the factors that can impact my performance?
  - ▶ Number of users that are active at any given time (RAM, CPU, disk speed)
  - ▶ Logging strategy (Disk space)
  - ▶ API users especially with heavy searches (substring searches for example) (RAM, CPU, Disk speed)



- What are the factors that generally do **NOT** impact my performance as much as expected?
  - ▶ Warninglist usage
  - ▶ Number of raw attributes on the instance
  - ▶ Number of sync connections / recurring syncs (with measure)
  - ▶ Tools feeding off the automation channels (ZMQ, kafka, syslog)

### └ Hardware considerations - continues

- What are the factors that generally do **NOT** impact my performance as much as expected?
  - ▶ Warninglist usage
  - ▶ Number of raw attributes on the instance
  - ▶ Number of sync connections / recurring syncs (with measure)
  - ▶ Tools feeding off the automation channels (ZMQ, kafka, syslog)

- Username/password is the default
- Some built in modules by 3rd parties (LDAP, Shibboleth, x509, OpenID, Azure Active Directory)
- CustomAuth system for more flexibility
- Additionally, consider Email OTP

### └ Authentication options

- Username/password is the default
- Some built in modules by 3rd parties (LDAP, Shibboleth, x509, OpenID, Azure Active Directory)
- CustomAuth system for more flexibility
- Additionally, consider Email OTP

- PHP tuning
  - ▶ Maximum memory usage (per process)
  - ▶ Timeout settings
  - ▶ Consider setting it per role!
  - ▶ Background processes are exempt
- MySQL: key buffer size is important
- Generally, tune for few heavy requests rather than many light ones

### Other considerations - tuning

- PHP tuning
  - ▶ Maximum memory usage (per process)
  - ▶ Timeout settings
  - ▶ Consider setting it per role!
  - ▶ Background processes are exempt
- MySQL: key buffer size is important
- Generally, tune for few heavy requests rather than many light ones

- Clustering
  - ▶ Load balanced apache servers with MISP
  - ▶ Replicating / mirrored database backends
- Careful about session pinning
- Attachment storage can be abstracted / network attached
- An example implementation for AWS  
<https://github.com/oxtf/HAMISPA>

2024-07-08

## MISP Deployment

└─ Other considerations - high availability

- Clustering
  - ▶ Load balanced apache servers with MISP
  - ▶ Replicating / mirrored database backends
- Careful about session pinning
- Attachment storage can be abstracted / network attached
- An example implementation for AWS  
<https://github.com/oxtf/HAMISPA>