

MISP DEPLOYMENT

SOME BASIC GUIDELINES

CIRCL / TEAM MISP PROJECT



13TH ENISA-EC3 WORKSHOP

- **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: <https://github.com/MISP/misp-docker>
- RPM maintained by amuehlem:
<https://github.com/misp/misp-rpm>
- Cloud provider images:
<https://github.com/MISP/misp-cloud>
- LXC images via misp-airgap:
<https://github.com/MISP/misp-airgap/>

- There are several MISP Docker installations available.
- The official MISP Docker installation is maintained by ostefano and can be found at <https://github.com/MISP/misp-docker>.
- The official MISP Docker image also includes one for the MISP modules.

- 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

- 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)

- 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

- 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>