particularly difficult. A radio transmitter of the appropriate power and frequency placed near a protected target prevents GPS receivers from receiving the correct data. Satellite manufacturers are trying to combat this by developing increasingly jamming-resistant signals. However, in principle, the advantage is always on the side of those who attack. They can respond to changes faster thanks to less expenditure of time and resources, because satellites cannot be modernized cheaply and quickly. However, we should not expect the military to abandon the GPS system. On the contrary, the fight against jamming systems will increase, and additional components will be added to the equipment and weapons that will prevent GPS signal jamming. Engineers are also looking for fundamentally new technical solutions. For example, after a series of hacks and failures in the GPS system, the US military and several American laboratories announced that they were working on a new quantum navigator that could completely change global search systems, eliminating the need for satellites [1].

REFERENCES

1. Gerasimenko K. Analysis of methods of suppressing signals of satellite radio navigation systems by intentional jamming. *Weapon systems and military equipment.* 2015. Vol. 44. P. 61-63.
2. Satellite navigation: basic principles of work. Problems and methods of solving them [Electronic resource]. – Access mode: https://око.укр/articles/GPS_GLONASS_AGPS_RTK/
3. Navigation support for navigation: [practical allowance] / S. V. Simonenko, N. F. Golodov. Kyiv: State Hydrography State Institution, 2015. 268 p.
4. European Space Agency website [Electronic resource]. – Access mode: https://www.esa.int/Applications/Satellite_navigation/Galileo/What_is_Galileo.

V. Riabovolenko, O. Baybuz, Yu. Honcharova

DOCKER AS A MEANS OF LOAD BALANCING

A container is an instance of a Docker image and serves to run applications, processes, or services. It is formed from the content of the Docker image, an execution environment, and a standard set of instructions. If necessary, many instances of containers can be created from the same image to expand the application. The container includes the operating system, user files and metadata. A Docker image contains information on how to initiate a container, including which process to start at its launch, among other settings [3].

A Docker image is read-only; when a container is initiated, a writable layer is added on top of the image using a union file system, allowing the application to be launched.

Docker Swarm is a tool for organizing a cluster in Docker, transforming a group of Docker hosts into a single cohesive cluster. Each host in this cluster functions as either a manager or a worker node, with the mandatory condition of having at least one manager node. The physical placement of servers is not critical, but it is preferable for all Docker nodes to be located in the same local network to simplify management and problem-solving [1].

Docker Swarm provides several important functions for efficient container management within the cluster, including:

1. Cluster automation: easy integration of multiple Docker hosts into a single cluster.
2. Node management: supports managing and worker nodes for task distribution.
3. Scalability: simple scaling of services according to needs.
4. Load balancing: distributes traffic among containers to optimize resources.
5. Port opening and routing: automatically routes external requests to the appropriate containers.
6. Declarative service description: uses configuration files to define services.
7. Self-recovery: automatically recreates containers upon their failure.
8. Security: secures communication between nodes using TLS.
9. Container orchestration: smartly distributes containers across nodes according to resources and settings [2].

Fig. 1 illustrates the architecture of Docker Swarm.

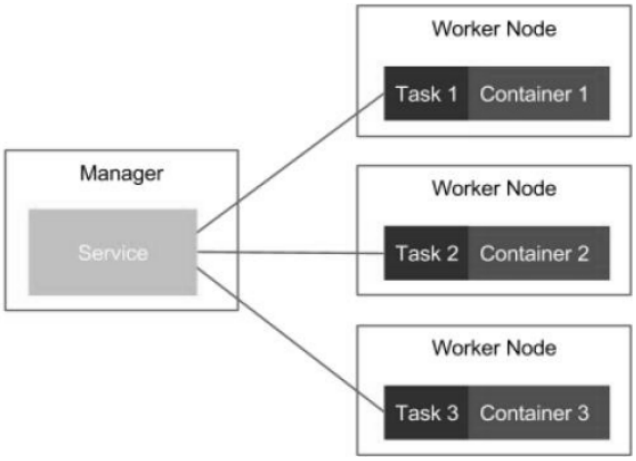

Fig. 1. Docker Swarm Architecture

In summary, Docker facilitates the easy deployment of applications into containers, ensuring their isolation and portability. Docker Swarm extends these capabilities by enabling the management of container clusters.

REFERENCES

1. Biggs J., Salanov J. Building Intelligent Cloud Applications: Develop Scalable Models Using Serverless Architectures with Azure 1st Edition. O'Reilly Media, 2019. 154 p.
2. Load Balancing in Cloud Computing, 2016. URL: https://www.researchgate.net/publication/297667.
3. Miell I. Docker in Practice, Second Edition / I. Miell, A. Hobson Sayers., 2018. – 425 c

V. Sarancha, O. Verba, A. Kutovyi

ARTIFICIAL INTELLIGENCE (AI) IN CYBERSECURITY: KEY ASPECTS AND CHALLENGES

Artificial intelligence (AI) has revolutionized the cybersecurity industry by providing advanced threat detection and prevention. This technology can help security teams counter threats more effectively by providing real-time analysis of potential threats and vulnerabilities. By detecting and eliminating security threats before they can cause damage, AI can significantly improve the overall security of an organization.

Artificial intelligence is becoming an essential tool in the fight against cyber threats, including phishing, fraud, and data theft [3]. The potential for severe losses from cybercrime has led to an increasing focus on the use of AI to protect corporate networks and data [1]. By analyzing large amounts of data, AI can detect even the slightest signs of a cyber threat and take preventive measures [3].

However, as the role of AI in cybersecurity grows, new problems arise. For example, many AI systems operate as black boxes, making the decision-making process opaque. This makes it difficult to understand what decisions are being made and why. There is also a risk of malicious attacks when attackers exploit vulnerabilities in the systems [2]. To overcome these problems, it is necessary to develop and improve the use of AI in cybersecurity actively: it is important to ensure accountability of decisions and data confidentiality in artificial intelligence systems.