SELF-HEALING CLOUD COMPUTING

Petrenko S.A.[1]

Abstract

Purpose of the article: development of tools for *building a cyber-stable private cloud. The relevance of building a cyber-resilient private cloud is confirmed by the dynamics of growth in the market volume of relevant solutions. According to PRnewswire, the market for private cloud solutions will reach 183 billion USD by 2025. At the same time, the average annual growth rate of the CAGR will be 29.4% during the forecast period. According to the analytical company Grand view research, the global market for private cloud solutions in 2018 was estimated at 30.24 billion US dollars, and it is expected that in the period from 2019 to 2025, the CAGR will be 29.6%.*

Research methods: *It uses a set of open-source solutions that applies the advanced cloud technologies, including distributed data processing models and methods, container orchestration technologies, software-defined data storage architecture, and a universal database.*

Results*: Developed tools for building a cyber-stable private cloud. Considered a possible approach to building a cyber-resilient private cloud based on the well-known and proprietary models and methods of the artificial immune systems (AIS), as well as technologies for distributed data processing, container orchestration, and others. In addition, the unique centralized fault-tolerant logging and monitoring subsystem has been developed for the described platform, as well as an innovative cybersecurity subsystem based on the following original technologies.*

Keywords: *Digital transformation, Digital economy, Critical information infrastructure, Cyber resilience, Self-organization, Proactive cyber security and adaptability, Big Data, Cloud computing.*

DOI:10.21681/2311-3456-2021-1-80-89

1. Selection and justification of the tool platform

The well-known practice of building disaster-tolerant and fault-tolerant clouds is based on modern technologies of reliability, fault tolerance, and disaster recovery (Figure 1).

Let's choose two possible directions for the development of tools for building a cyber-resilient private cloud platform for building virtual and dedicated pri-

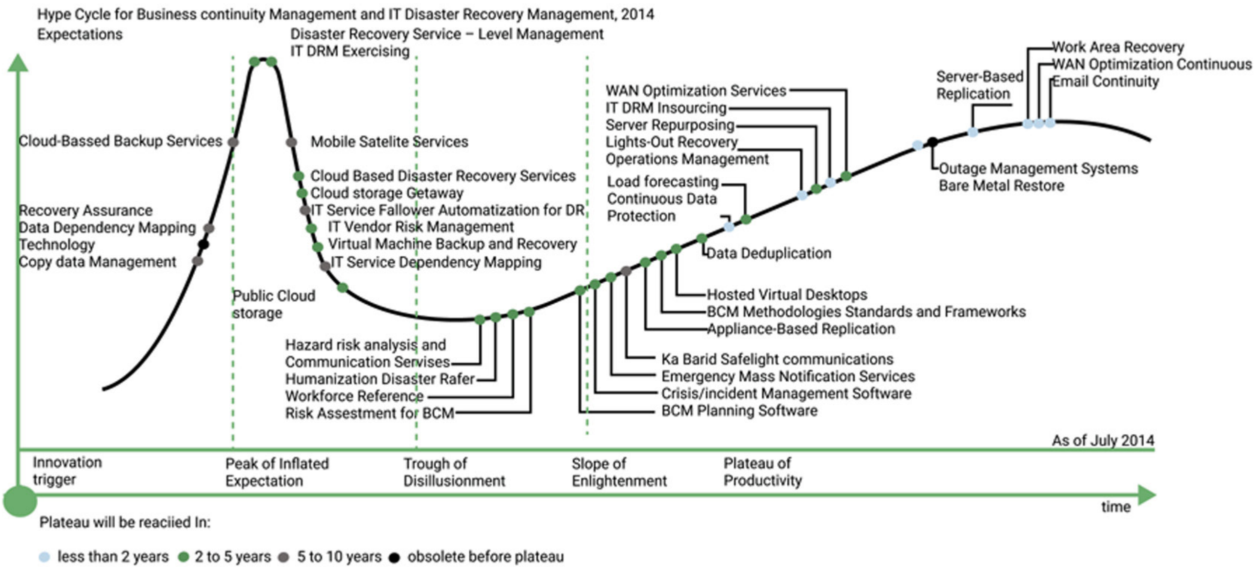

Figure 1a. Well-known technologies for building clouds, Gartner

[1] Sergei Petrenko [0000-0003-0644-1731], Innopolis University, Universitetskaya 1, Innopolis, 420500, Russia. E-mail: s.petrenko@innopolis.ru

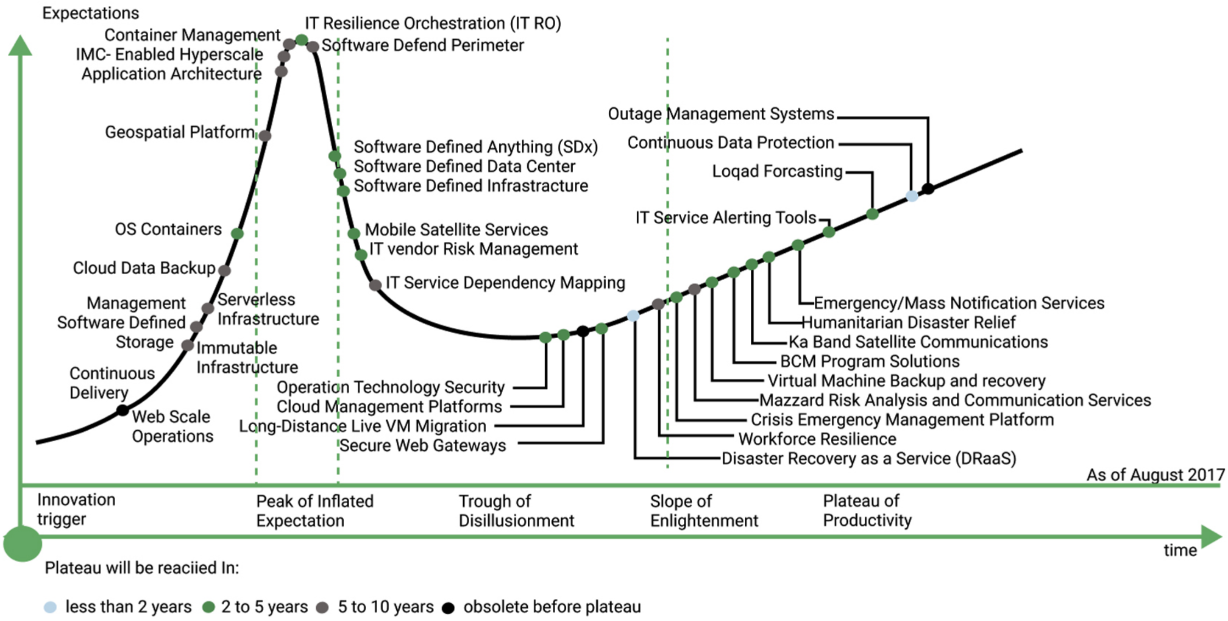

Figure 1b. Well-known technologies for building clouds, Gartner

vate cloud environments. The well-known solutions of this class, include, in particular:

- Red Hat Open Shift Container Platform – a platform for developing, deploying, and operating classic and container applications in physical, virtual, and public cloud environments;
- Amazon Virtual Private Cloud, Google Cloud Platform, Mail.ru Cloud, Yandex Cloud – a public cloud, designed for a large number of clients;
- VMware vCloud Suite, Microsoft Azure Stack, Cisco Private Cloud, Oracle Private Cloud – are characterized by an advanced functionality for multiple clients and the availability of standard solutions for building a private cloud for a single client;

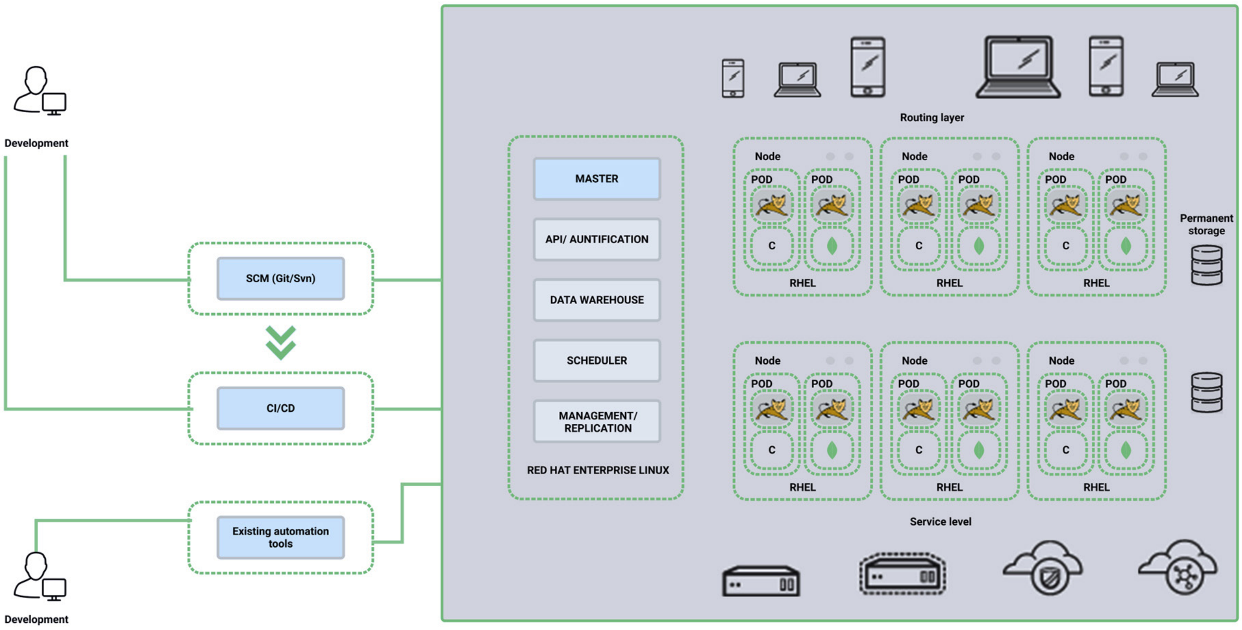

Figure 2. Red Hat Open Shift Container Platform

DOI: 10.21681/2311-3456-2021-1-80-89

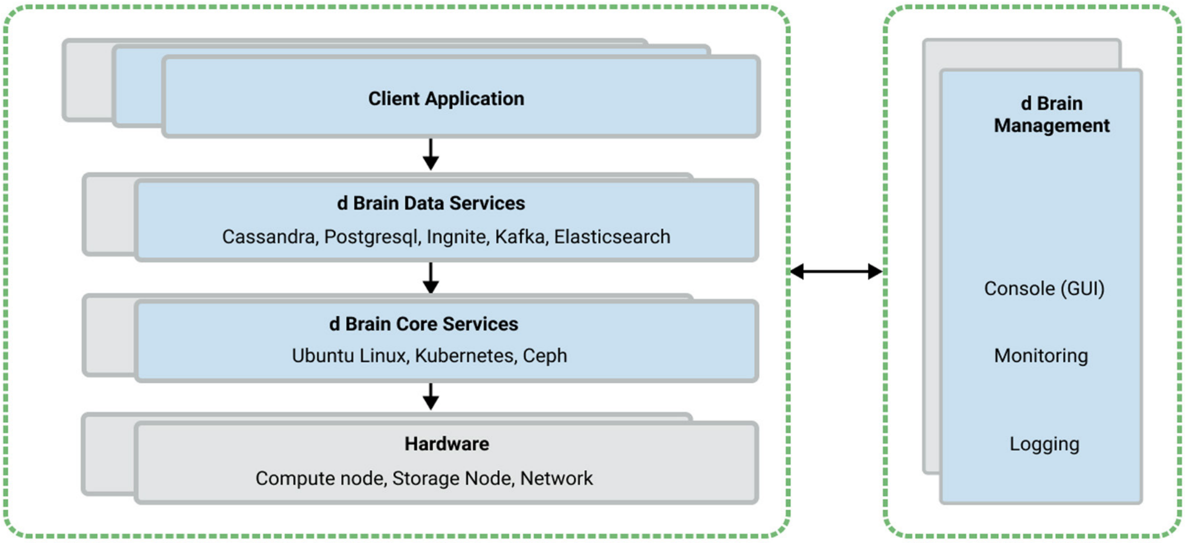
Client Application
d Brain Data Services
Cassandra, Postgresql, Ingnite, Kafka, Elasticsearch
d Brain Core Services
Ubuntu Linux, Kubernetes, Ceph
Hardware
Compute node, Storage Node, Network
d Brain Management
Console (GUI)
Monitoring
Logging

Figure 3. Structure of the open-source platform

- HPE Helion Cloud System - a Suite of software solutions for deploying private and hybrid cloud environments;
- Dell Active System is a family of ready-made integrated systems for private cloud, virtualization, and application deployment.

For example, Red Hat Open Shift Container Platform (Figure 2) allows preparing, building, and deploying applications and their components. At the same time, S2I automation tools for building source code (source-to-image) simplifies building the Docker containers, based on code extracted from the version control system. And the combination with DevOps (development and operations) and CI/CD (continuous delivery and integration) tools contributes to the development of the solution.

Let us take a look at a possible open-source solution (Figure 3) that applies the advanced cloud technologies, including distributed data processing models and methods, container orchestration technologies, software-defined data storage architecture, and a universal data bus. In addition, the unique centralized fault-tolerant logging and monitoring subsystem has been developed for the described platform, as well as an innovative cybersecurity subsystem based on the following original technologies.

Security immune technology. The corresponding SDK has been developed, which for the first time in cybersecurity practice allows accumulating and applying the artificial cyber immunity to protect against previously unknown types of malicious software cyberattacks (more than 50 % of the total number of cyberattacks). The SDK integrates into the most common tool environments and development studios for digital platforms and applications in various programming languages (JavaScript, Java, C#, Python, C++, TypeScript, Swift, Kotlin, Rubi, Go, 1C, C, Scala, Pascal/Delphi, T-SQL, Dart, PLSQL, Erlang, Apex, etc.). Its practical significance lies in the fact that it allows implementing the so-called dynamic control and prevents the transfer of critical information infrastructure of the state and business to irreversible and catastrophic States [1-4].

The “digital bombs” detecting technology. Implemented on the basis of deep semantic analysis and certification of applications using dimensions and similarity invariants. Well-known methods for detecting software bug include static and dynamic software analysis methods. The so-called profiling methods based on monitoring the behavior of critical information infrastructure in the face of growing security threats are also becoming widespread. However, performing static analysis in the absence of program source texts and software documentation required searching for new approaches to effectively detect, warn, and block the software bugs. Therefore, an original solution to the above problem was proposed, justified, and implemented. The solution is based on detecting software defects by software examination with consideration for structural, logical, and operational properties of programs based on: a set of models and methods of graph theory (for studying the structure of software bugs); RSL logic (for studying the logic of software bugs); Petri nets with zero-check (for studying the actions of software bugs); similarity and dimension theory (for dynamic control of computation semantics). As a result, an original Toolkit and dynamic control automaton were developed to detect and neutralize the destructive software bugs and "digital bombs" of critical information infrastructure [5-8].

Homomorphic encryption and quantum cryptanalysis technology. Today, the homomorphic encryption systems (RSA, El Gamal, Peye, Gentry – Palevi - Smart, Gribov-Mikhalev, Burtyk-Babenko) are widely used in the protection of cloud computing. The regarded development uses a unique technology for detecting previously unknown vulnerabilities in cryptographic primitives and asymmetric encryption algorithms, including those based on quantum cryptanalysis - the modified Shor and the original Grover algorithm. The software and hardware complex for quantum cryptanalysis of cloud computing is developed in the Python programming language in the Jupiter computing environment. The obtained results allowed significantly improving the cryptographic protection of cloud computing based on the above-mentioned open-source platform [9-11]

2. Composition and structure of a sustainable private cloud

The proposed universal platform for building a cyber-resilient private cloud includes the following main components:

- Data Center engineering infrastructure;
- Platform hardware;
- Core Services software;
- Data Services software;
- Management software;
- Software-defined Ceph data storage;
- Controlled environment of the Kubernetes container runtime;
- Database software:
- Security subsystem;
- Other software;
- Auxiliary software and services.

Let us look at the functionality and technical characteristics of the listed platform components in more detail (Figure 3).

Network infrastructure of the platform

The platform's network infrastructure is based on a relatively new Leaf-Spine architecture, which replaced the classic three-level network architecture in data centers (Figure 4), which used the Spanning Tree Protocol (STP), also to prevent route loops during data transmission.

A distinctive feature of Leaf-Spine (Figure 5) is the ability to adapt to the growth of Big data using operational scaling.

The advantages of the Leaf-Spine architecture (Figure 6) include:

- dynamic third-level routing based on the Equal-Cost Multi-pathing (ECMP) Protocol, which improves network throughput rate and provides the required data transfer stability;
- ability to expand a network capacity by adding new hardware.

Its disadvantages include an increase in the number of connections in the switching scheme of Leaf and Spine devices, as well as certain difficulties in deploying VLANs in the network. We recommend using the software Defined Networking (SDN) functionality and creating a virtual layer 2 over the Leaf-Spine network [12-15] to solve this problem,

Network infrastructure of the platform

The platform's server infrastructure includes the following components (Figure 7):

- AUX servers - responsible for collecting and processing system information, metrics, logs, etc., as well as for fast and convenient deployment of configurations on a variety of servers using techniques used in cloud platforms. Linear scaling is applied.
- Kubernetes-cluster-server nodes (from lat. nodus-node) K8S-MASTER plus several K8S-WORKERS. The number of nodes in a Kubernetes cluster depends on the required computing power. K8S-LB nodes are responsible for user load balancing. Linear scaling is applied.
- CEPH cluster – CEPH-HOT server nodes (CEPH cluster nodes with disks with high read and write speeds, usually SSDS). CEPH-WARM server nodes (cluster nodes with disks that have standard read/write speed parameters, usually HDD). The number of nodes in the CEPH cluster depends on the required amount of information storage. Linear scaling is applied.

Core Services software

It is designed to deploy a secure and stable platform to run applications and store data, as well as update the software of the platform component.

Data Services software

Designed to perform the following tasks:

- exchange of messages between different applications;
- ensuring scalability and fault tolerance;

DOI: 10.21681/2311-3456-2021-1-80-89

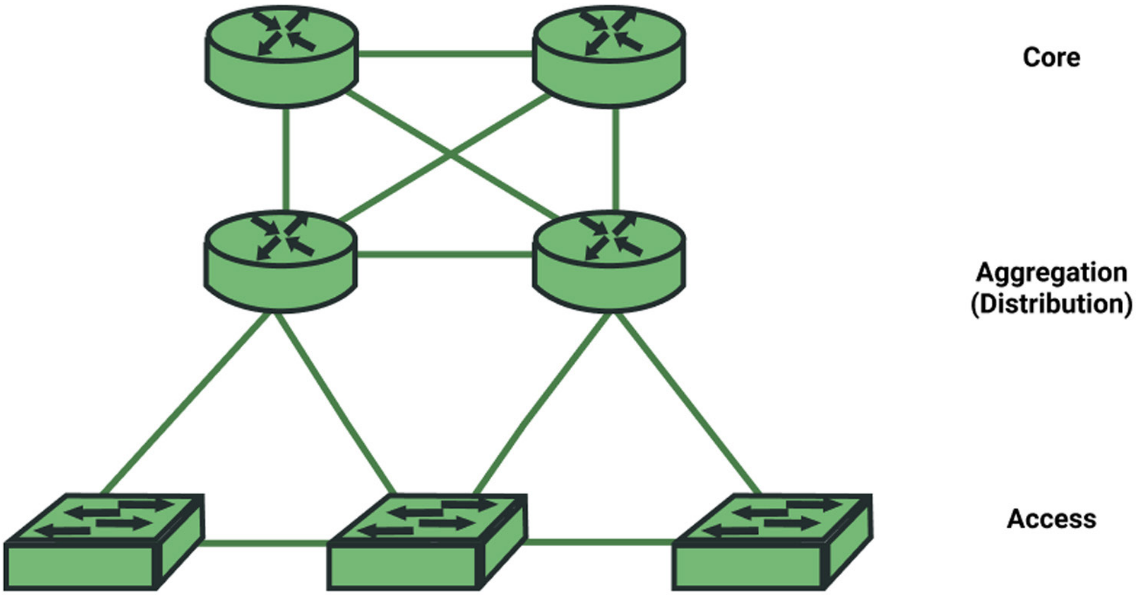

Figure 4. Classic three-level architecture of data center networks

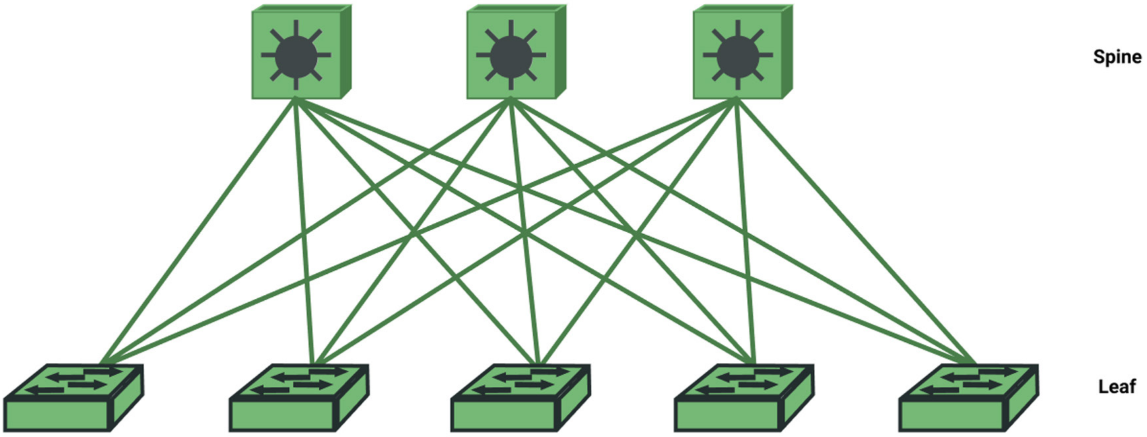

Figure 5. Perspective Leaf-Spine architecture

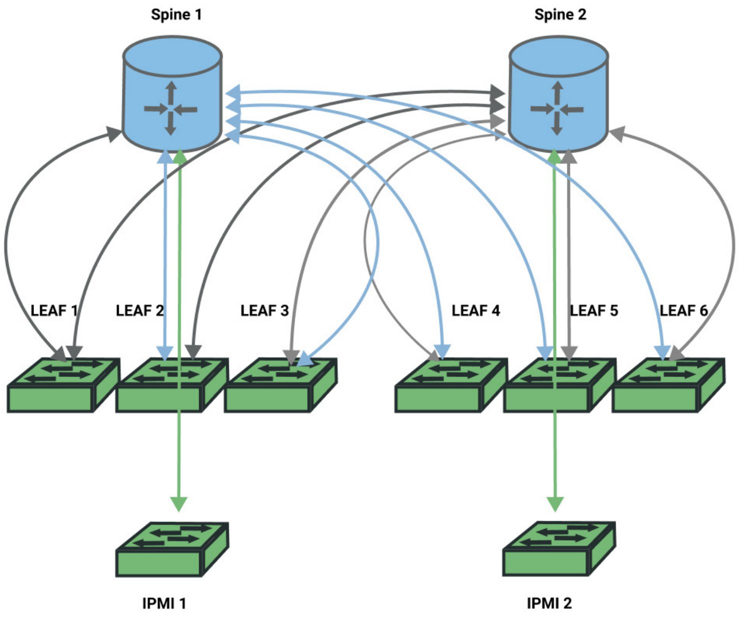

Figure 6. Open-source platform network architecture

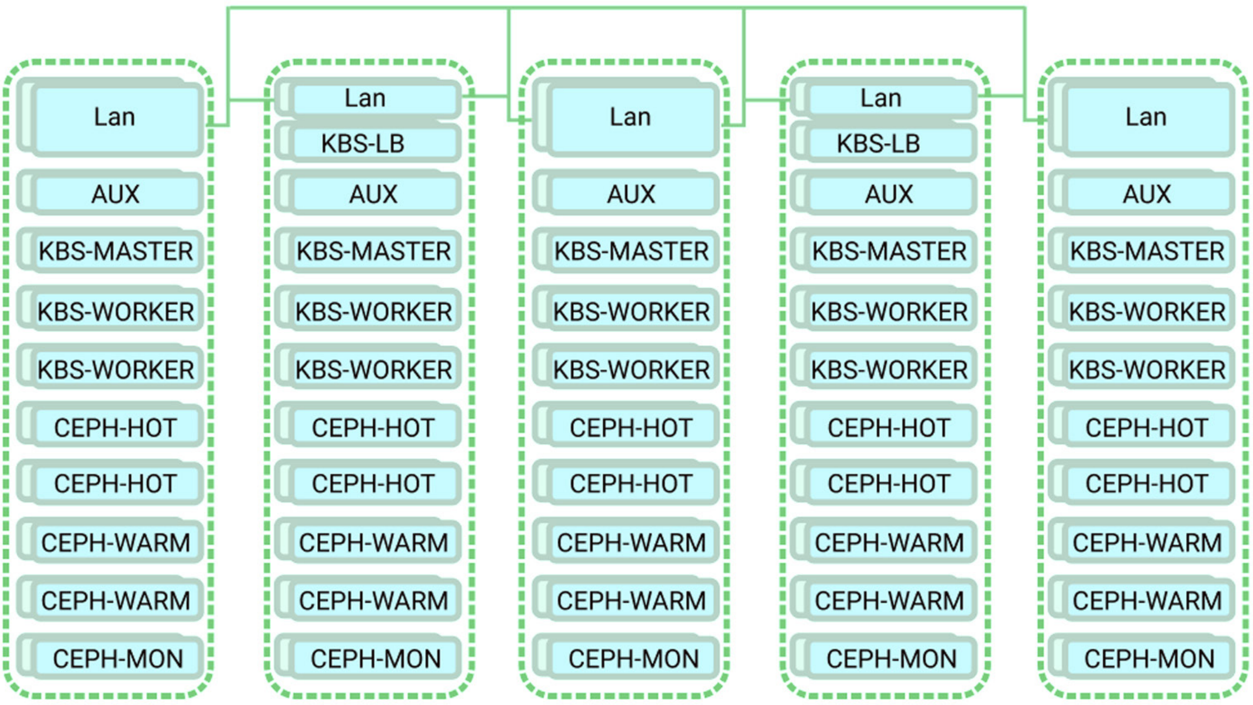

Figure 7. Network infrastructure of the platform

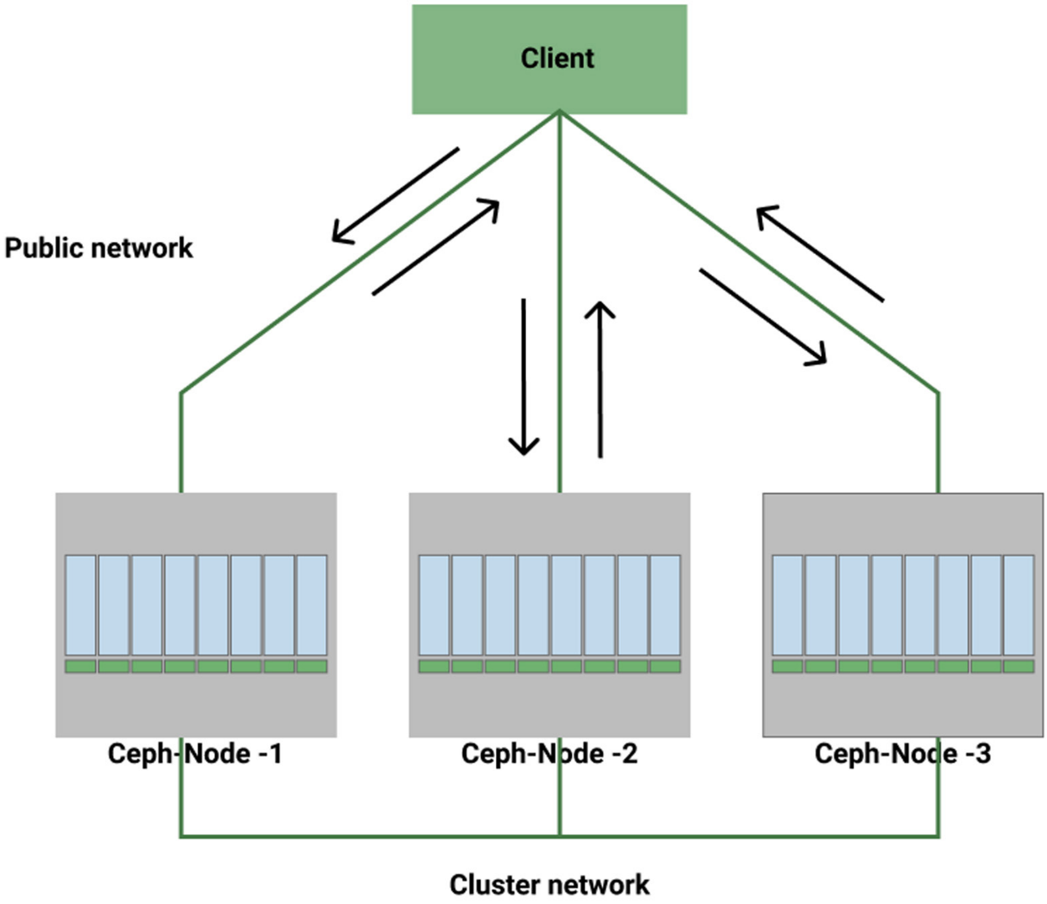

Figure 8. Ceph storage system operation

- providing a delivery guarantee;
- support for storing a limited number of recent messages on disk to batch data processing.

Management software

Designed to perform the following tasks:

- collection and storage of metrics and logs;
- detection of problems based on specified triggers with a subsequent notification;
- providing information about the status of cluster components;
- routine operations over the cluster components;
- ferreting out the problems with the performance and stability of cluster components.

The Management software includes the following components:

- monitoring and logging system - collects metrics from all machines and applications for further processing and analysis, configures triggers for certain types of events that occur in the cluster, with further notification via email and Slack;
- logging system collects logs from all servers and applications;
- cluster management system: service personnel are provided with a web interface for managing and receiving information about the status of cluster components. All operations performed with the cluster are logged for the subsequent incident resolution. The notification mechanism is used in order to promptly inform about problems.

Software-defined Ceph data storage

A promising Ceph data storage system is involved , which is a fault-tolerant distributed data storage (Figure 8). An important feature of Ceph is the ability to scale to tens of thousands of nodes, which provides storage scaling in the petabytes (Pb) scale. Built-in data replication mechanisms ensure the required high reliability and availability of the system.

The Ceph storage system is represented by a set of several types of entities:

- OSD (object storage daemon) – the entity responsible for storing data. It is a “daemon” that works directly with a separate physical data storage. Each cluster node can host a large number of OSD entities.
- Mon (monitor) – a Ceph infrastructure coordinator that contains information about connected storage nodes (OSD), data distribution, and cluster status, as well as providing data addressing and replication.
- RGW (RADOS Gateway) – an auxiliary “daemon” that acts as a gateway to provide object storage. Ceph provides two different abstractions for working with storage, namely object storage and Ceph block device.

The block device abstraction allows the user to create and use virtual block devices of any size, while the object storage abstraction uses Ceph to store the user objects with an S3-compatible Protocol.

Controlled environment of the Kubernetes container runtime

Today, technologies for building secure and sustainable private clouds based on Docker and Kubernetes container management systems are widely used.

Here, Docker (a well-known open-source project) packages an application (microservice) in a container along with all its dependencies. It simplifies application deployment because the image can be run on almost any Linux system without installing additional libraries and configuration files. The appropriate supervisory control and data acquisition system is used to deploy applications packaged in Docker.

Kubernetes (also an open-source project) is a system for automatic deployment and project management based on microservice architecture and Docker technology. Kubernetes combines a large number of computing nodes into a single cluster. Horizontal scaling is

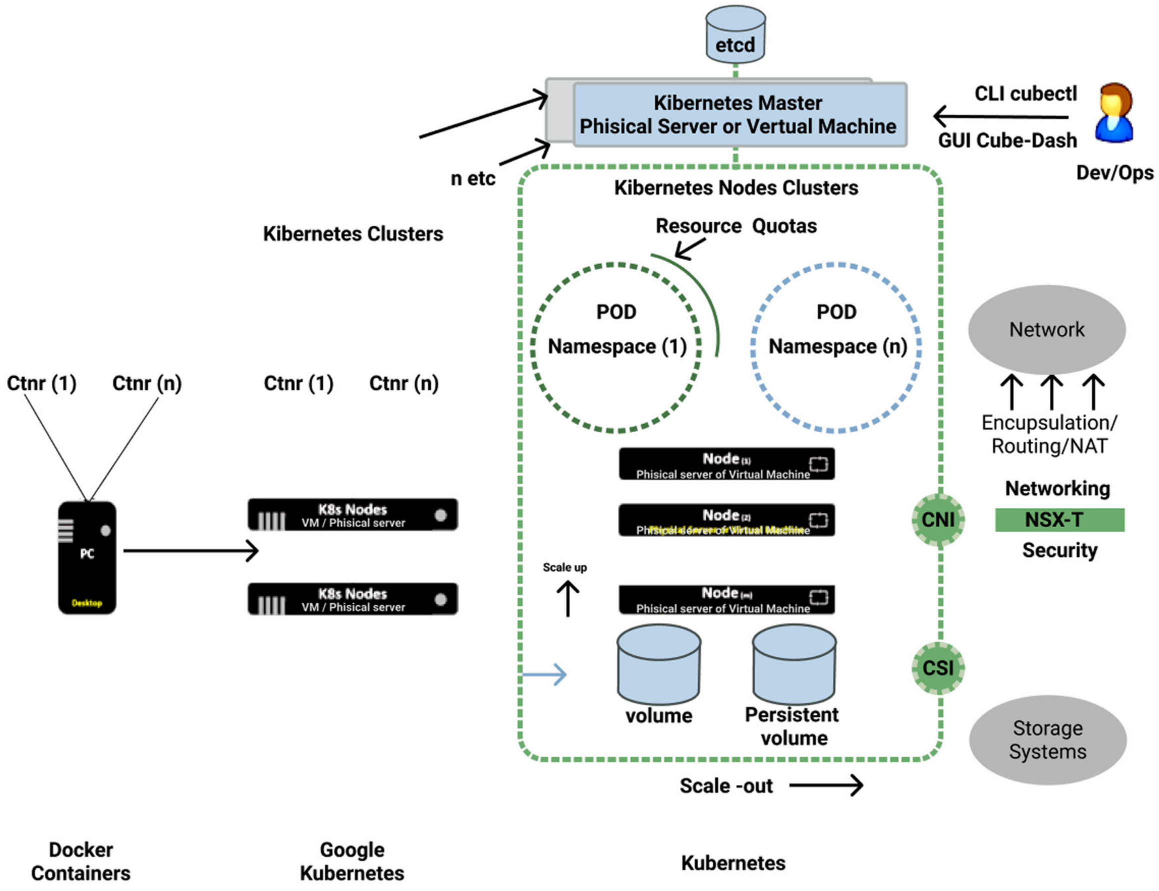

Figure 9. Docker and Kubernetes container management system

DOI: 10.21681/2311-3456-2021-1-80-89

achieved by running image instances on different physical servers, and the load on the service is automatically distributed between running instances (Figure 9).

Database software

The Cassandra system is used for database management which belongs to the class of NoSQL systems and is designed to create highly scalable and reliable storage of the Big data storage represented as a hash. Cassandra uses a data storage model based on a column family. The information placed in the database is automatically replicated to several nodes in the distributed network. This system is characterized by high scalability and reliability, high flow capacity for write/read operations, configurable consistency, fast writing, and flexible circuit. A SQL-like query language supports searching data using secondary indexes (see figure 10).

Monitoring and logging system

This system is designed to collect data from machines and applications for further processing and analysis configures triggers for certain types of events that occur in the cluster, with further notification via email and corporate messenger. The mentioned system also collects logs from all servers and applications, combines and systematizes their infrastructure.

The logging and monitoring system processes, aggregates, and outputs the following data in an accessible form:

- information that reflects the operation of systems, processes, and services, including the operation of a software-defined distributed file system;
- temperature, voltage, fan speed, and information from other sensors on server and network equipment;
- data for monitoring and using disk partitions;
- indicators of the self-monitoring, analysis and reporting mechanism;
- information received from the network device, which contains technical information about them.

Cyber security subsystem.

The cyber security subsystem includes the following components:

- hardware and software complex for protection against unauthorized access;
- hardware and software complex for detecting, preventing and neutralizing cyber attacks;
- software and hardware complex of immune protec-

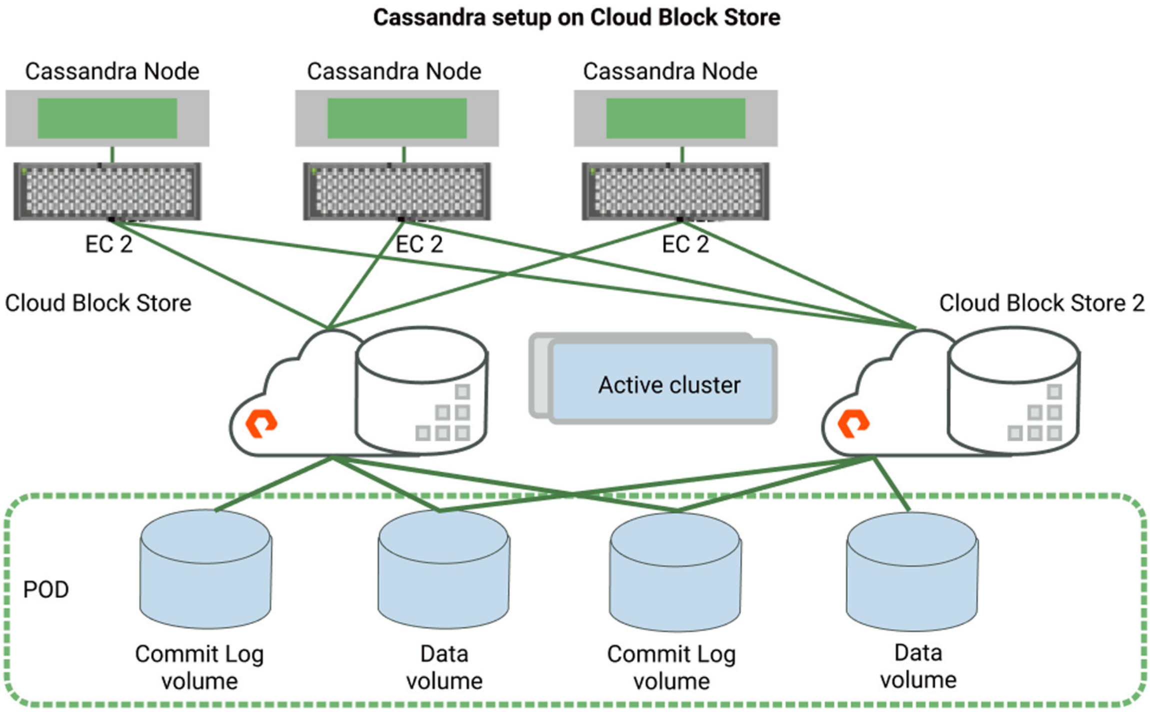

Figure 10. Cassandra database management system

tion for detecting and neutralizing previously unknown cyber-attacks;

- hardware and software complex for the homomorphic encryption;
- antivirus software;
- security monitoring system;
- system is multithreaded scanning of files and others.

3. Conclusion

A private cloud is a cloud computing model in which all infrastructure resources and services are dedicated to a single organization. In comparison to the public cloud, this model is characterized by an increased security level, extensive control capabilities, a more flexible choice of hardware, and compliance with legal requirements in terms of meeting the local information security requirements of the country. This includes Federal laws No. 187-FZ "On the security of the critical information infrastructure of the Russian Federation" and No.152-FZ "On personal data". This article has shown which approaches and technologies allow building a cyber-stable private cloud.

Acknowledgement

The publication was carried out with the financial support of Russian Foundation for Basic Research (RFBR) in the framework of the scientific project No. 20-04-60080.

References

1. Bodeau D., Graubart R., Heinbockel W. and Laderman E.:Cyber Resiliency Engineering Aid-The Updated Cyber Resiliency Engineering Framework and Guidance on Applying Cyber Resiliency Techniques (MTR140499R1PR 15-1334) (May 2015).
2. Bodeau D., Brtis J., Graubart R. and Salwen J.:Resiliency Techniques for System of Systems: Extending and Applying the Cyber Resiliency Engineering Framework to the Space Domain (MTR 130515, PR 13-3513), (September 2013).
3. Ronald S. Ross: Risk Management Framework for Information Systems and Organiza-tions: A System Life Cycle Approach for Security and Privacy (December 20, 2018).
4. Markov A., Markov G., Tsirlov V. SIMULATION OF SOFTWARE SECURITY TESTS BY SOFT COMPUTATIONAL METHODS: CRITICAL INFRASTRUCTURES: CONTINGENCY MANAGEMENT, INTELLIGENT, AGENT-BASED, CLOUD COMPUTING AND CYBER SECURITY (IWCI 2019). Proceedings of the VIth International Workshop. Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences. 2019. С. 257-261. 1
5. Dorofeev A.V., Markov A.S., Tsirlov V.L. APPLICATION OF OPEN DATA IN ACCORDANCE WITH INFORMATION SECURITY REQUIREMENTS: CEUR Workshop Proceedings. ISTMC 2019 - Selected Papers of the 4th All-Russian Scientific and Practical Conference with International Participation "Information Systems and Technologies in Modeling and Control". 2019. С. 36-46.
6. Petrenko, S.: Big Data Technologies for Monitoring of Computer Security: A Case Study of the Russian Federation, Springer Nature Switzerland AG (2018).
7. Petrenko, S.: Cyber Security Innovation for the Digital Economy: A Case Study of the Russian Federation, River Publishers, (2018).
8. Kott, A., Linkov, I: Cyber Resilience of Systems and Networks (Risk, Systems and De-cisions), 2019 Springer Nature Switzerland AG, (2019)
9. ISO/TS 22330:2018, Security and resilience -- Business continuity management sys-tems -- Guidelines for people aspects of business continuity.
10. ISO/TS 22331:2018, Security and resilience -- Business continuity management sys-tems -- Guidelines for business continuity strategy.
11. NIST Special Publication 800-160 VOLUME 2. Systems Security Engineering. Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems, (March 2018).
12. NIST Special Publication 800-160 VOLUME 3. Systems Security Engineering. Soft-ware Assurance Considerations for the Engineering of Trustworthy Secure Systems (December 20, 2019).
13. NIST Special Publication 800-160 VOLUME 4. Systems Security Engineering. Hard-ware Assurance Considerations for the Engineering of Trustworthy Secure Systems, (December 20, 2020).
14. Graubart R,: The MITRE Corporation, Cyber Resiliency Engineering Framework, The Secure and Resilient Cyber Ecosystem (SRCE) Industry Workshop Tuesday, (November 17, 2015).
15. Ronald S. Ross, Michael McEvilley, Janet C. Oren: Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Se-cure Systems (March 21, 2018).