

Для обох критеріїв використовуємо однакові ваги 0.5, щоб вони мали однакову важливість. Обчислимо матрицю відстаней для кожної країни від ідеального пункту та антиідеального пункту. Побудуємо за отриманими даними графік порівнянь. Його можна побачити на рис. 1.

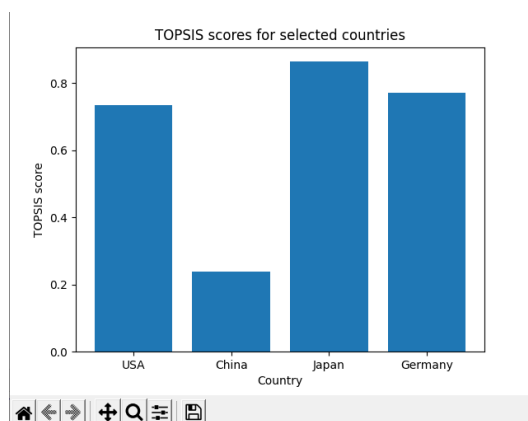


Рисунок 1 – графік порівнянь країн за обраними критеріями

З графіку видно, що за нашими критеріями перемогла Японія, а найгірший результат показав Китай.

Отже, при застосуванні методу TOPSIS для вибору країни для проживання, можна порівняти держави за різними показниками та визначити, яка країна найбільш відповідає потребам та вимогам. Зважаючи на різноманітність критеріїв, які визначені для порівняння, можна зробити висновок про те, яка країна має найкращі перспективи для проживання та розвитку.

Список використаних джерел:

1. Романченко І. С., Потьомкін М.М. Метод TOPSIS та його використання для багатокритеріального порівняння альтернатив. *Системи обробки інформації*. 2016. № 138. С. 104–106
2. Міщенко, О. М., & Якімчук, І. В. (2020). Метод TOPSIS в оцінці якості медичних послуг. *Економіка та управління виробництвом*, 4(60), 5-16.
3. Wang, Y., Wei, L., & Zhang, J. (2017). Application of the TOPSIS method in choosing a residential country. *Journal of Computational & Theoretical Nanoscience*, 14(7), 3221-3227.

UDC 62 681.5 004.72

Prokopenko Mykola, DevOps Engineer at DataArt

REDUCE RISK FOR CONTINUOUS DEPLOYMENT FOR CLOUD-NATIVE APPLICATIONS. GITOPS APPROACH

The cloud-native approach is about how applications are built and deployed, not where. When companies build and operate applications in the cloud, they bring new ideas to market faster and respond faster to customer requirements.

Conflicts over resources, goals, and priorities within an organization can pose risks to successful operations.

Problems arising in the development and automation of ci/cd

1. Complexity and the need to manually manage infrastructure and applications in cloud environments. Work with infrastructure and speed of response to changes;

2. Tracking and verification of changes in the infrastructure;
3. The complexity of managing distributed teams and projects. Create and manage infrastructure and applications in cloud environments using a unified version control system.

GitOps as a solution to problems and reduce potential risks in development.

With continuous delivery, whether it's an application or changes in infrastructure, errors can occur at any of the levels and steps of this stage:

– the risk of the human factor. GitOps allows you to automate the deployment and management of applications, which reduces the risk of human error;

– risk of loss of control over changes. GitOps uses the Git version control system to store and manage declarative descriptions of infrastructure and applications as code, making it easy to track and control all changes to the system;

– risk of security and privacy breach. GitOps allows you to automatically check and test changes before they are applied to the system, which improves the security and reliability of the system;

– the risk of increasing the time for deployment and scaling. GitOps allows you to automate deployment and scaling processes, which reduces the time for these operations and increases system performance;

– risk of errors when working in distributed teams. GitOps allows you to create and manage infrastructure and applications in cloud environments using a single version control system, which makes it easier for distributed teams to work and reduces the risk of errors.

Factors affecting the potential cost of automation

1. Availability of CI/CD pipelines. If you already have CI/CD pipelines in place, then GitOps integration can be easier and more accessible with fewer resources;

2. The experience and qualifications of the team. The GitOps accessibility score can be derived from the experience and skills of the team that will be in charge of getting and retrieving application information;

3. Infrastructure. Estimating the cost of maintaining GitOps can be estimated from what cloud infrastructure is being used, what services are required, and what payment scheme is being used.

Implementing GitOps is a process that requires careful preparation and planning. Here are a few steps that can help you get started with your GitOps implementation:

1. The first step in implementing GitOps is to evaluate the current infrastructure that will be managed with GitOps. This includes analysis of configurations, network and security settings, management tools used, etc.;

2. Selecting the infrastructure management tools to be used to manage the infrastructure with GitOps;

3. Estimation of potential labor costs for the implementation of the approach and assessment of potential savings in reducing the likelihood of errors.

The GitOps approach has the potential to save a lot of money, but it is important to pay attention to the balance of potential implementation effort and the potential payback timeframe of such an approach. The potential payback may be contrary to the short-term business model, or its being at the initial stage, where the speed of implementation of the functionality is important, against the backdrop of competition in a particular niche.

References:

1. <https://about.gitlab.com/topics/gitops/>
2. <https://learn.microsoft.com/ru-ru/azure/architecture/checklist/dev-ops>
3. <https://www.ibm.com/cloud/architecture/architecture/practices/securing-cloud-native-apps-risks-mitigation/>
4. https://aws.amazon.com/cloudformation/?nc1=h_ls