

DevOps

Introduction

Welcome to the Unicis DevOps Handbook. This guide is designed to help you understand and manage the DevOps processes for the Unicis Platform, which is built on Platform-as-a-Service (PaaS) infrastructure using OVH Cloud and utilizes CapRover for Continuous Integration and Continuous Deployment (CI/CD).

Platform Overview

Unicis Platform is a robust solution designed to deliver a seamless PaaS experience. Our infrastructure is hosted on OVH Cloud, ensuring reliability and scalability. The CI/CD processes are streamlined using CapRover, a powerful, user-friendly deployment solution.

Infrastructure: OVH Cloud

OVH Cloud provides the backbone for our infrastructure. Here's how we leverage OVH Cloud for our PaaS needs:

1. OVH Cloud Setup

- **Account Creation:** Create an account on the OVH Cloud platform.
- **Project Setup:** Set up a new project in the OVH Cloud dashboard.
- **Resource Allocation:** Allocate necessary resources (compute instances, storage, network configurations) based on project requirements.

2. Network Configuration

- **VPC Setup:** Create a Virtual Private Cloud (VPC) for secure networking.
- **Subnets and Security Groups:** Define subnets and configure security groups to control inbound and outbound traffic.
- **Load Balancers:** Set up load balancers to distribute traffic efficiently.

3. Storage Solutions

- **Object Storage:** Use OVH Cloud Object Storage for scalable and secure data storage.
- **Block Storage:** Attach block storage volumes to instances as needed.

4. Monitoring and Alerts

- **Monitoring Tools:** Integrate monitoring tools (e.g., Grafana, Prometheus) to keep track of infrastructure health.
- **Alerting Mechanisms:** Set up alerts for critical events and thresholds.

CI/CD: CapRover Solution

CapRover is our chosen solution for managing CI/CD processes due to its simplicity and powerful features.

1. CapRover Setup

- **Installation:** Deploy CapRover on a dedicated OVH Cloud instance.
- **Dashboard Access:** Access the CapRover dashboard via the web interface.
- **Initial Configuration:** Configure basic settings such as domain names and SSL certificates.

2. Application Deployment

- **Creating Applications:** Create new applications in CapRover for each service/component.
- **Deployment Methods:** Deploy applications using Docker images or directly from Git repositories.
- **Environment Variables:** Configure environment variables for each application as needed.

3. CI/CD Pipeline

- **Git Integration:** Integrate Git repositories with CapRover for automatic deployments.
- **Build Automation:** Set up automated build processes using CI/CD tools like GitLab CI.
- **Deployment Automation:** Configure CapRover to automatically deploy new builds to the respective environments.

Website and Handbook Deployment

Our website and handbook are continuously deployed using CI/CD and hosted on GitLab Pages.

1. Repository Setup

- **GitLab Repository:** Create a GitLab repository for the website and handbook source code.
- **Branching Strategy:** Implement a branching strategy (e.g., main, develop, feature branches).

2. CI/CD Configuration

- **GitLab CI Configuration:** Define CI/CD pipelines in `.gitlab-ci.yml` file.
- **Build and Test:** Include stages for building and testing the website/handbook.
- **Deployment:** Add deployment stages to publish content to GitLab Pages.

3. GitLab Pages

- **Pages Setup:** Configure GitLab Pages to serve the website and handbook.
- **Custom Domain:** Set up a custom domain for the GitLab Pages site if needed.

Best Practices

- **Version Control:** Use Git for version control and maintain a clear commit history.

- **Code Reviews:** Implement a code review process to ensure quality and consistency.
- **Security:** Regularly update dependencies and apply security patches.
- **Backup and Recovery:** Set up regular backups and a recovery plan for critical data.

Troubleshooting

Common Issues

- **Deployment Failures:** Check build logs and CapRover dashboard for error details.
- **Performance Issues:** Monitor resource usage and optimize application performance.
- **Network Problems:** Verify network configurations and security group rules.

Debugging Tips

- **Logs and Monitoring:** Use logging and monitoring tools to diagnose issues.
- **Documentation:** Refer to official documentation and community forums for solutions.

[logs](#), [monitoring](#), [debugging](#), [documentation](#)

Resources

- **OVH Cloud Documentation:** [OVH Cloud Docs](#)
- **CapRover Documentation:** [CapRover Docs](#)
- **GitLab CI Documentation:** [GitLab CI Docs](#)

By following this handbook, you will be well-equipped to manage the Unicis Platform infrastructure and CI/CD processes effectively. For any further assistance, please reach out to the DevOps team.

[OVHcloud](#), [OVH](#), [GitLab](#), [CapRover](#)

From:
<https://handbook.unicis.tech/> - **Unicis Handbook**

Permanent link:
<https://handbook.unicis.tech/pub/development/devops>

Last update: **05.10.2024 13:04**