# Master Thesis:
# Development and Automation of Reliable Cloud Infrastructure for Scalable Microservices Deployment

Supervisor: Prof. Dr. Pramod Bhatotia
Advisor: Evgeny Volynsky
Student: Ana-Maria-Iulia Cornea

30.10.2023

# Agenda

- Introduction
- Project Requirements
- Research Question
- Background
- Approach & Solution
- Key Ideas
- Summary

# Introduction


GHGA THE GERMAN HUMAN GENOME-PHENOME ARCHIVE

Provides infrastructure as a service (IaaS) through a basic OpenStack instance.

Host Munich data hub for GHGA

K8s for the deployment of microservices on OpenStack.

Focuses more on functionality rather than the security of the microservice architecture and workflows.

Set up a secure OpenStack and Kubernetes cluster

Run GHGA analysis workflows and microservices securely and efficiently

# Introduction: Goals


- **Prototype the secure, reliable, and scalable infrastructure of the GHGA Munich data hub at the LRZ**
- Leverage AMD-SEV encryption in the development and implementation of the data hub's infrastructure
- Review the already developed microservice architecture from a security point of view.
- Prototype a framework on how to test the microservice architecture

# Project Requirements

- **OpenStack** for cloud infrastructure setup and management
- **Ceph** Block storage integrated with OpenStack
- **Kubernetes** for managing containerized applications

  (in our case Microservices)

- Access Control List (**ACL**) support



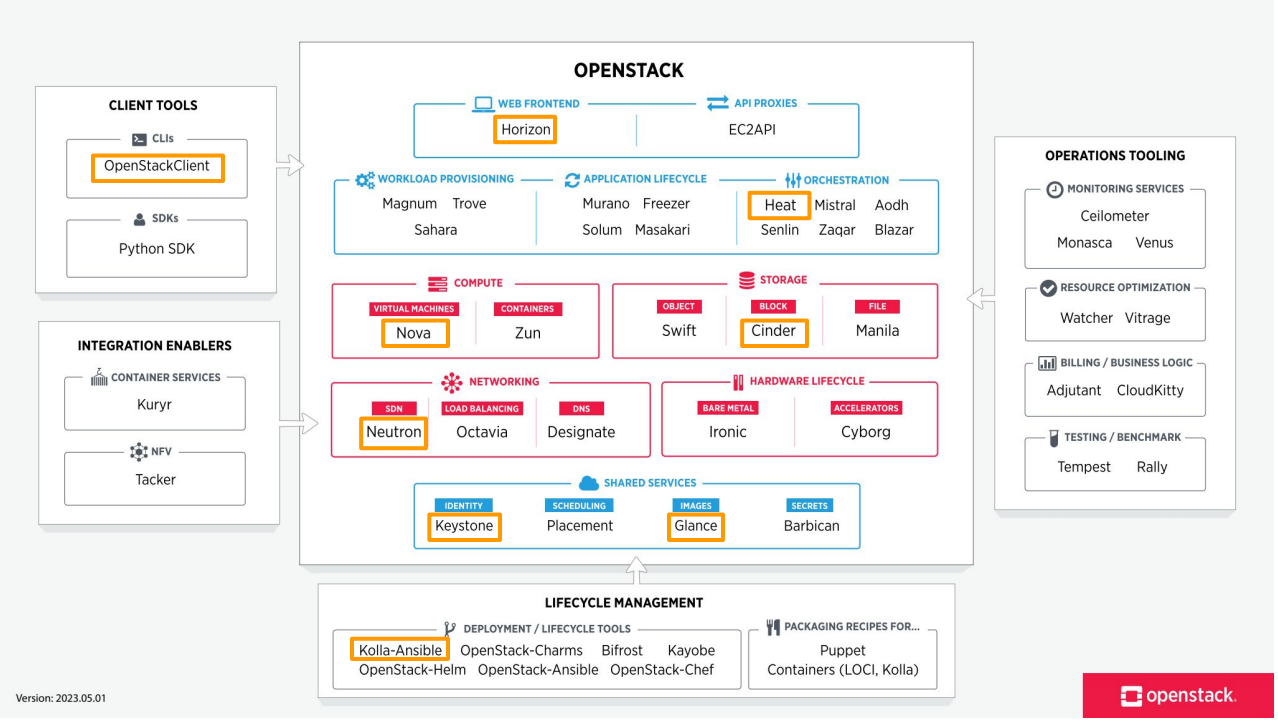

# Research Question

Q: “ **How can a cloud infrastructure, designed using OpenStack and Kubernetes, ensure scalable and reliable microservice deployment with a simplified setup process in the context of data-sensitive applications like human omics data ?** ”

A: Using a multi-tiered methodology containing OpenStack, Kubernetes and GitOps.

# Background: OpenStack

- Open-source cloud computing platform that enables the creation and management of scalable, flexible, and customizable private and public cloud environments.

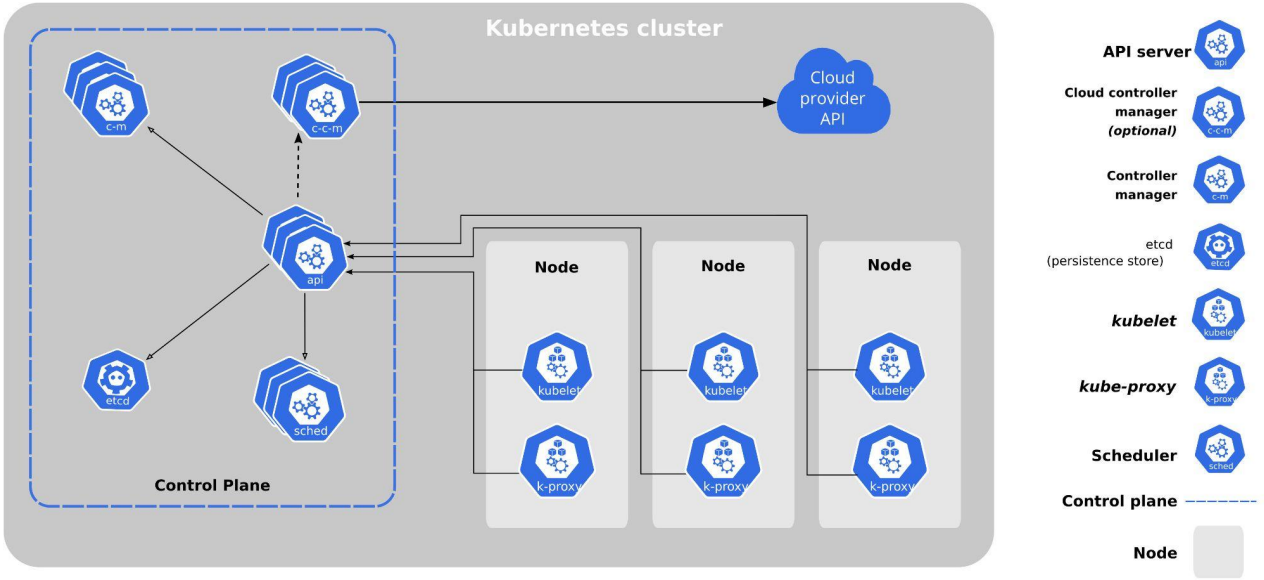


# Background: Kubernetes

- Open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications, providing resilience and efficiency for modern infrastructure needs.

Properties:

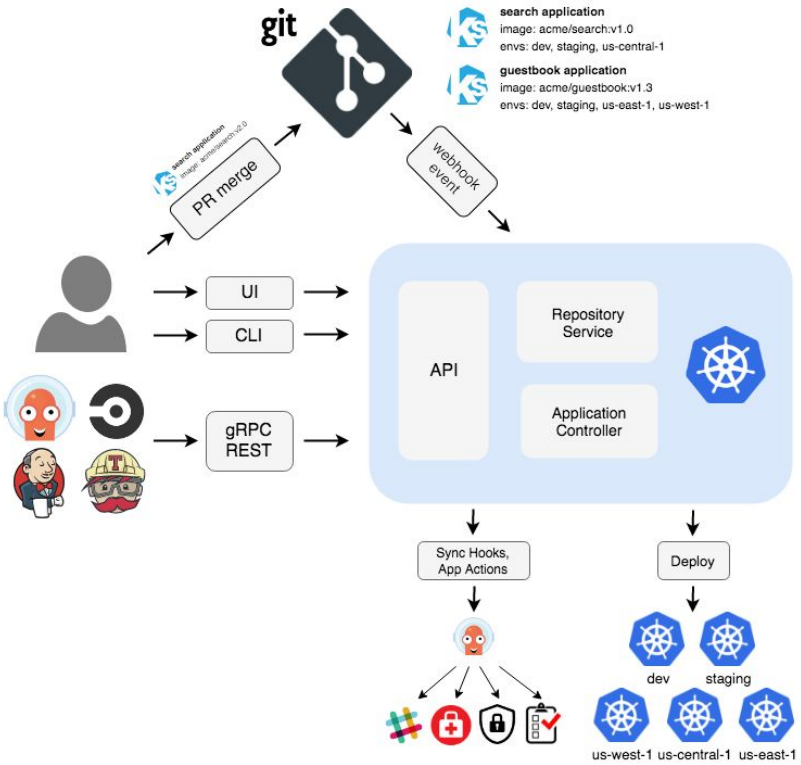
- Scalability
- Automated Operations
- Self-Healing
- Container and Storage orchestration
- Service Discovery
- Load Balancing



# Background: ArgoCD

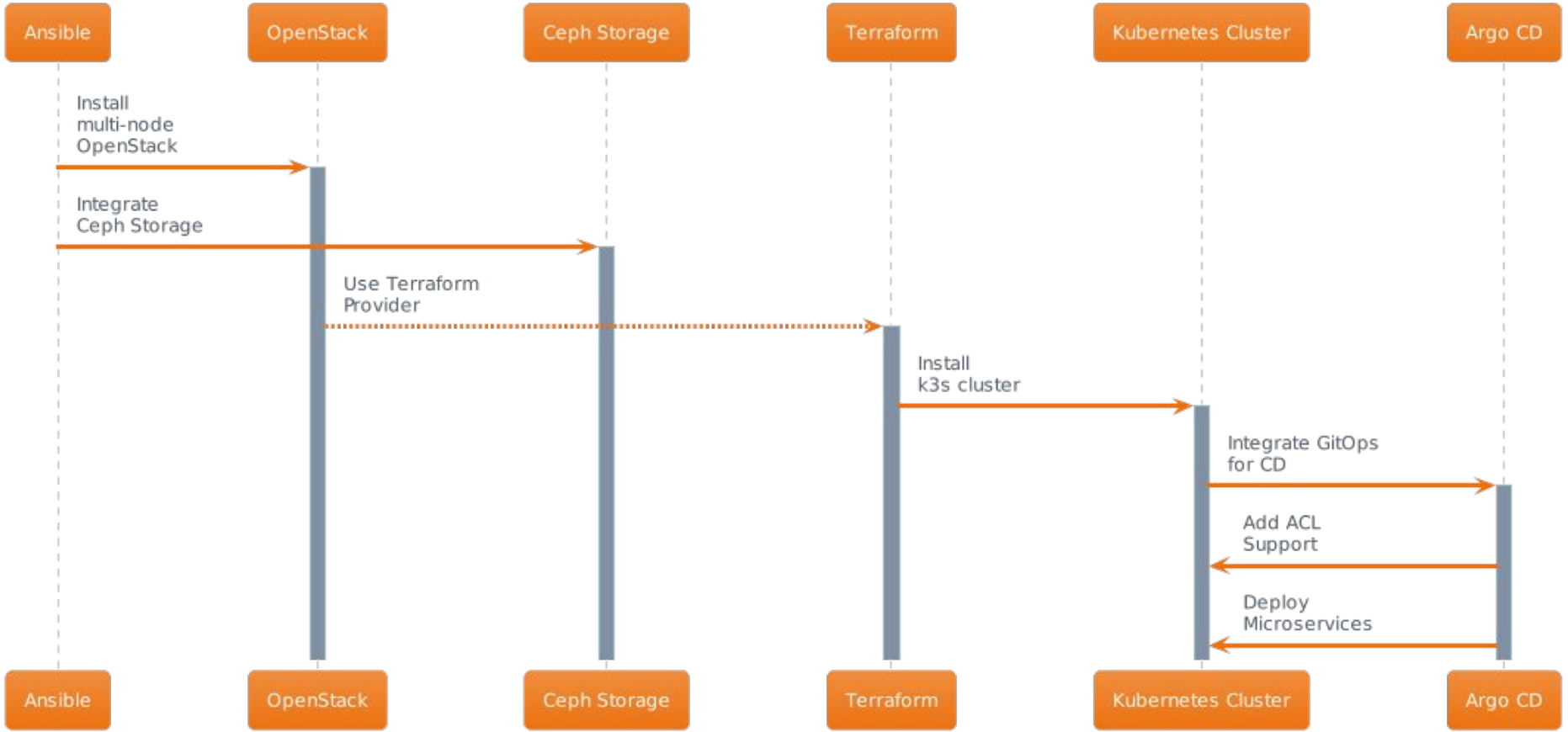
Features

- Automated app deployment
- Supports various config/templating tools
- Multi-cluster management and deployment
- Integration with various SSO platforms
- Multi-tenancy and RBAC for authorization
- CLI for CI integration and automation




https://argo-cd.readthedocs.io/en/stable/

# Approach & Solution - System Design

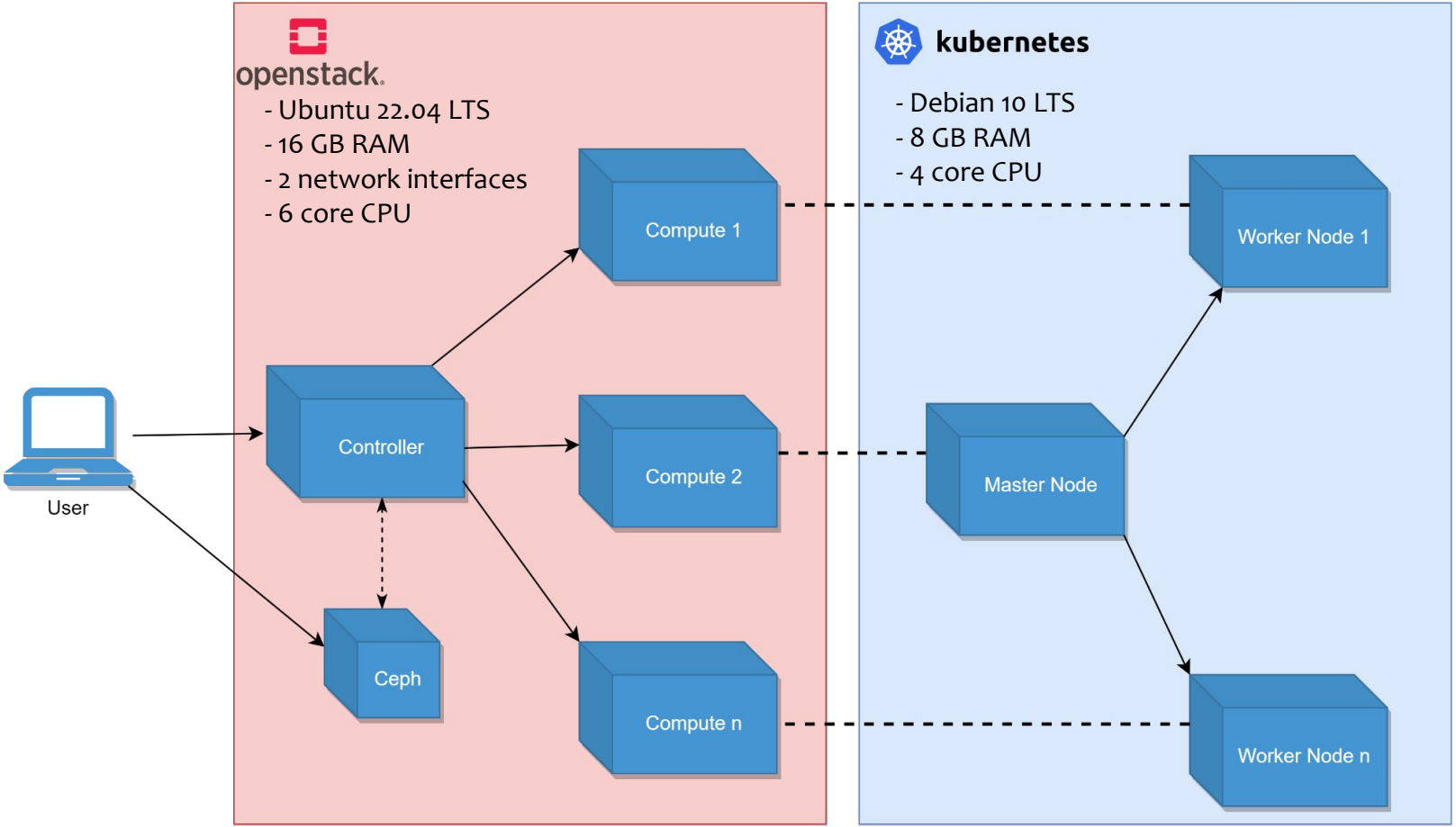


# Approach & Solution

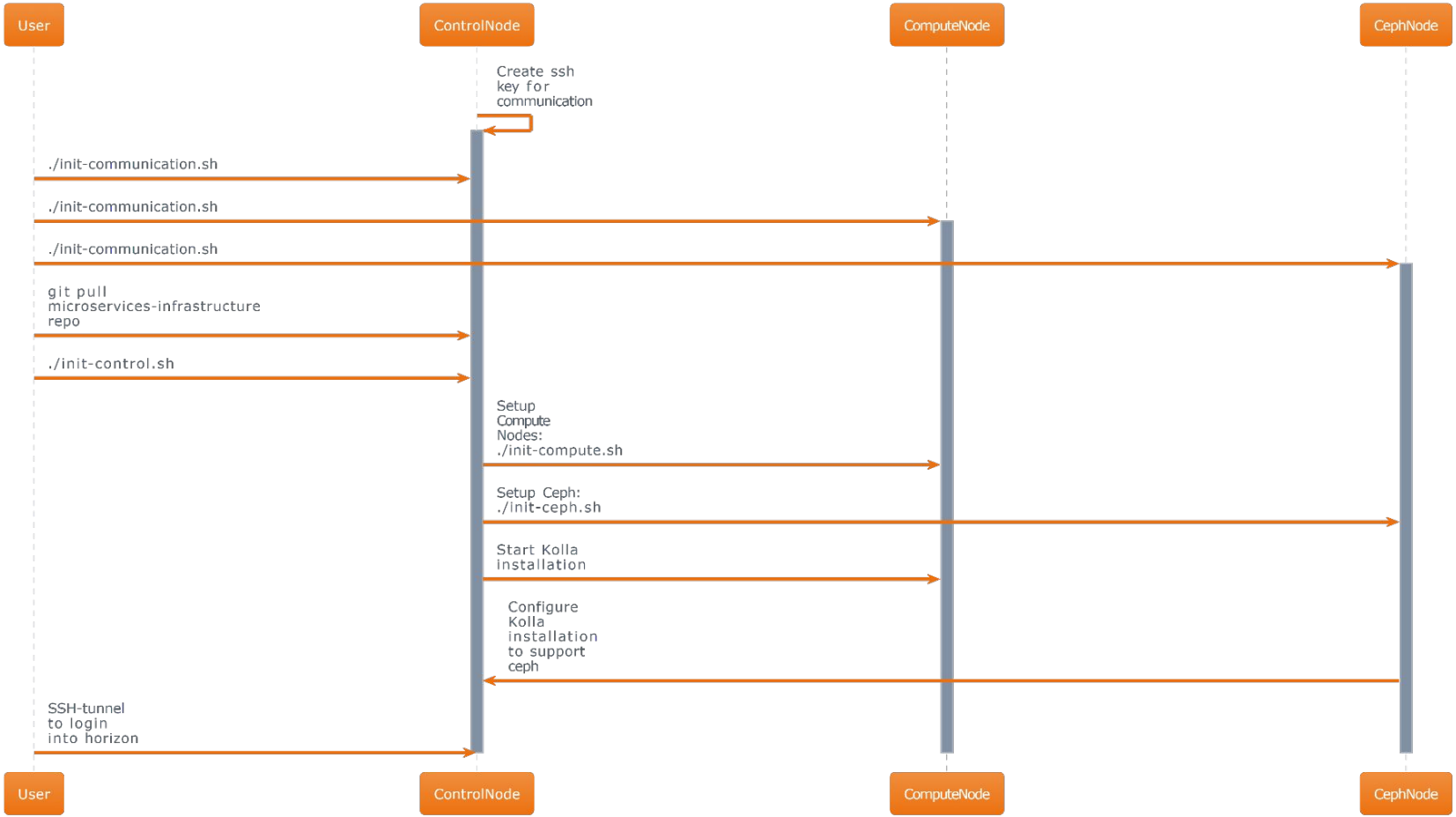
https://github.com/Evgeny-Volynsky/microservices-infrastructure


| | | |
|---|---|---|
| all-in-one-install-deploy-openstack | rename folders + fix globals | 4 days ago |
| argocd | add argocd manifests and readme | 13 hours ago |
| multinode-kolla-ceph-install-deploy | increase flavors and fix installation scripts | 15 hours ago |
| terraform | Update README.md | 15 hours ago |
| .gitignore | kolla ansible and terraform state | last month |
| README.md | Update README.md | 2 days ago |
| test-deployment.yaml | refactoring using terragrunt | 3 weeks ago |

# Approach & Solution



# Approach & Solution: OpenStack

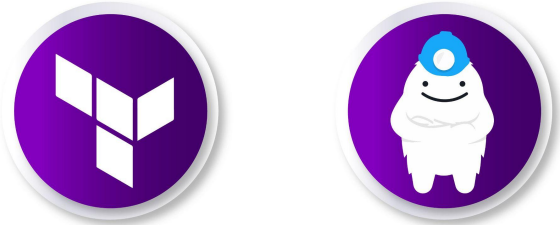


# Approach & Solution

https://github.com/Evgeny-Volynsky/microservices-infrastructure


| | | |
|---|---|---|
| all-in-one-install-deploy-openstack | rename folders + fix globals | 4 days ago |
| argocd | add argocd manifests and readme | 13 hours ago |
| multinode-kolla-ceph-install-deploy | increase flavors and fix installation scripts | 15 hours ago |
| terraform | Update README.md | 15 hours ago |
| .gitignore | kolla ansible and terraform state | last month |
| README.md | Update README.md | 2 days ago |
| test-deployment.yaml | refactoring using terragrunt | 3 weeks ago |

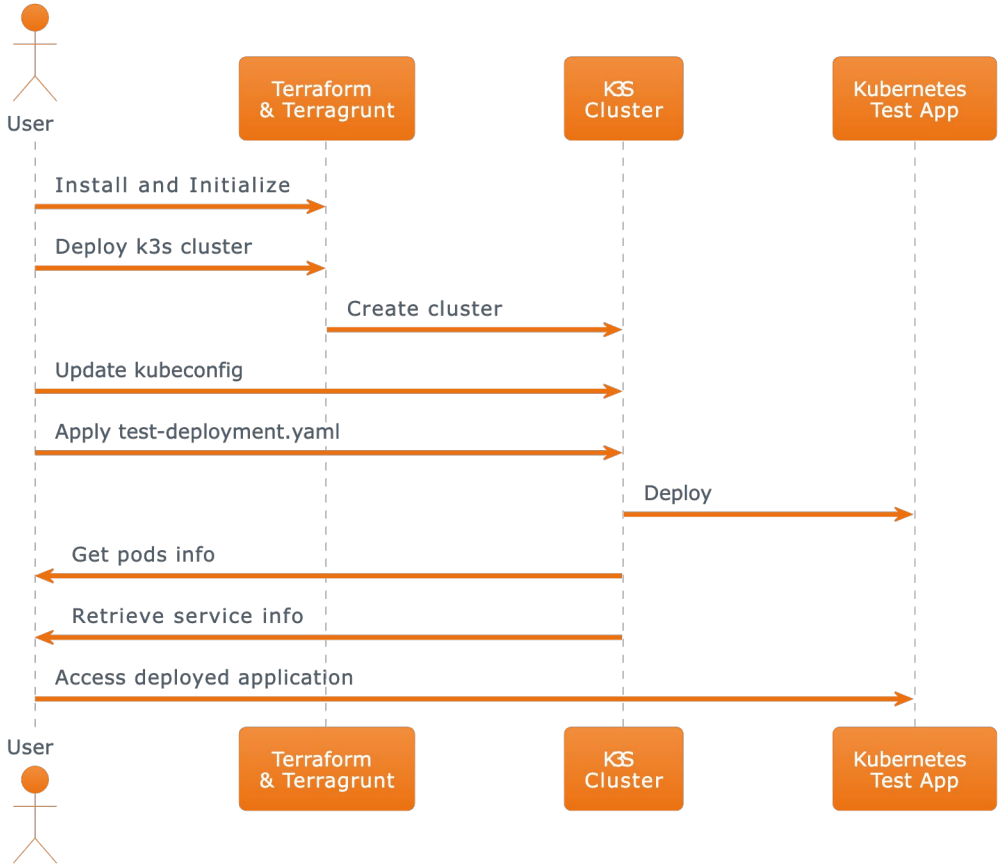
# Approach & Solution: Kubernetes



**Terraform & Terragrunt**

**Using Terraform & Terragrunt to:**

- Setup immutable k3s cluster on OpenStack infrastructure
- Specify the dependencies between OpenStack resources more explicitly and improve code quality

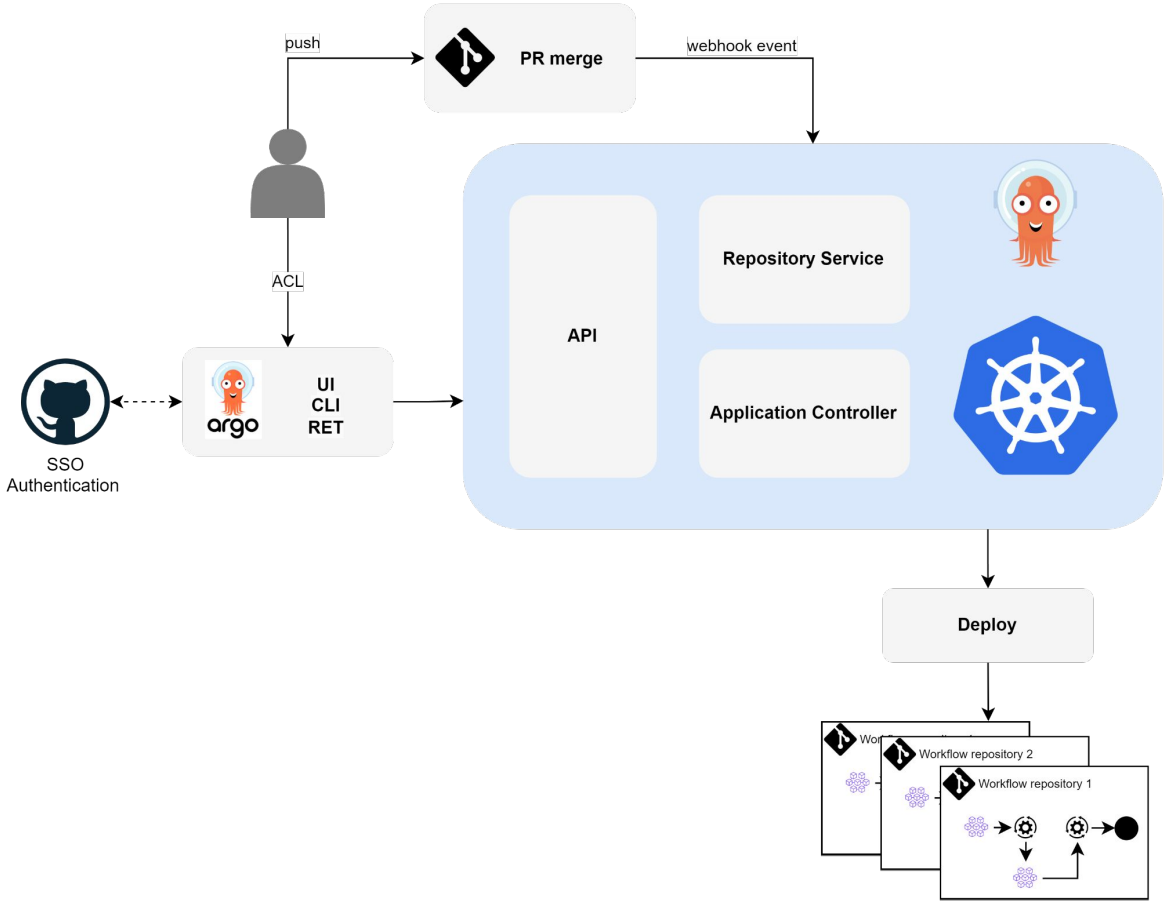


# Approach & Solution

https://github.com/Evgeny-Volynsky/microservices-infrastructure

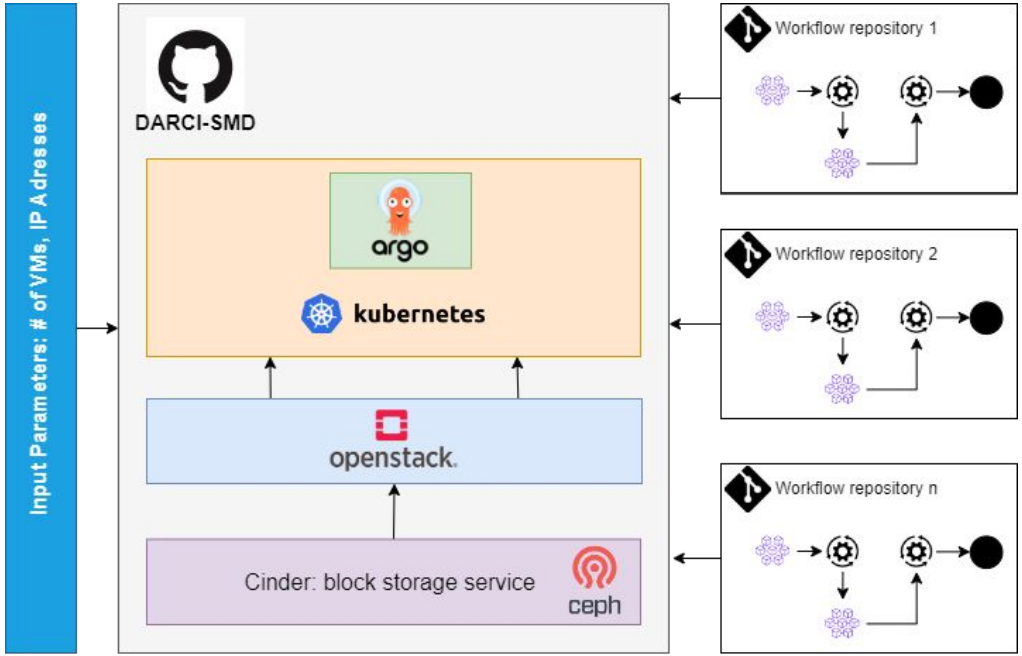

| | | |
|---|---|---|
| all-in-one-install-deploy-openstack | rename folders + fix globals | 4 days ago |
| argocd | add argocd manifests and readme | 13 hours ago |
| multinode-kolla-ceph-install-deploy | increase flavors and fix installation scripts | 15 hours ago |
| terraform | Update README.md | 15 hours ago |
| .gitignore | kolla ansible and terraform state | last month |
| README.md | Update README.md | 2 days ago |
| test-deployment.yaml | refactoring using terragrunt | 3 weeks ago |

# Approach & Solution: ArgoCD



# Key Ideas

- An end-to-end microservice deployment based on k3s and OpenStack
- Access control (Simple ACLs using ArgoCD)
- A GitHub framework for development and deployment (CD) of microservices
- A distributed storage backend to manage the state/data used by microservices



# Summary - Key Achievements

- Conceptualization:
  - Crafted a robust cloud infrastructure for scalable microservices deployment
- Prototype developed based on a 3-layered approach:
  - 1st layer: setup and management of OpenStack infrastructure with Ceph
  - 2nd layer: deployment of Kubernetes cluster using Terraform
  - 3rd layer: Argo CD integration with Kubernetes for CD and ACL support
- End Result:
  - Scalable and Production-ready infrastructure setup through scripts

Source code: https://github.com/Evgeny-Volynsky/microservices-infrastructure/

Recorded demo: link to Google Drive video or QR-code

# References


1. https://github.com/Evgeny-Volynsky/microservices-infrastructure
2. https://kubernetes.io/docs/
3. https://docs.openstack.org/
4. https://ceph.io/docs/
5. https://argo-cd.readthedocs.io/en/stable/
6. https://www.gitops.tech/

Thank you !

# Approach & Solution

**Setting Up**
Secure Kubernetes Cluster

**Secure**
State Management of Microservices

**Develop**
Testing Mechanism for Microservices

**Execute**
Security Review of Architecture

Stage I | Stage II | Stage III | Stage IV

# Approach & Solution

**Setting Up**
Secure Kubernetes Cluster

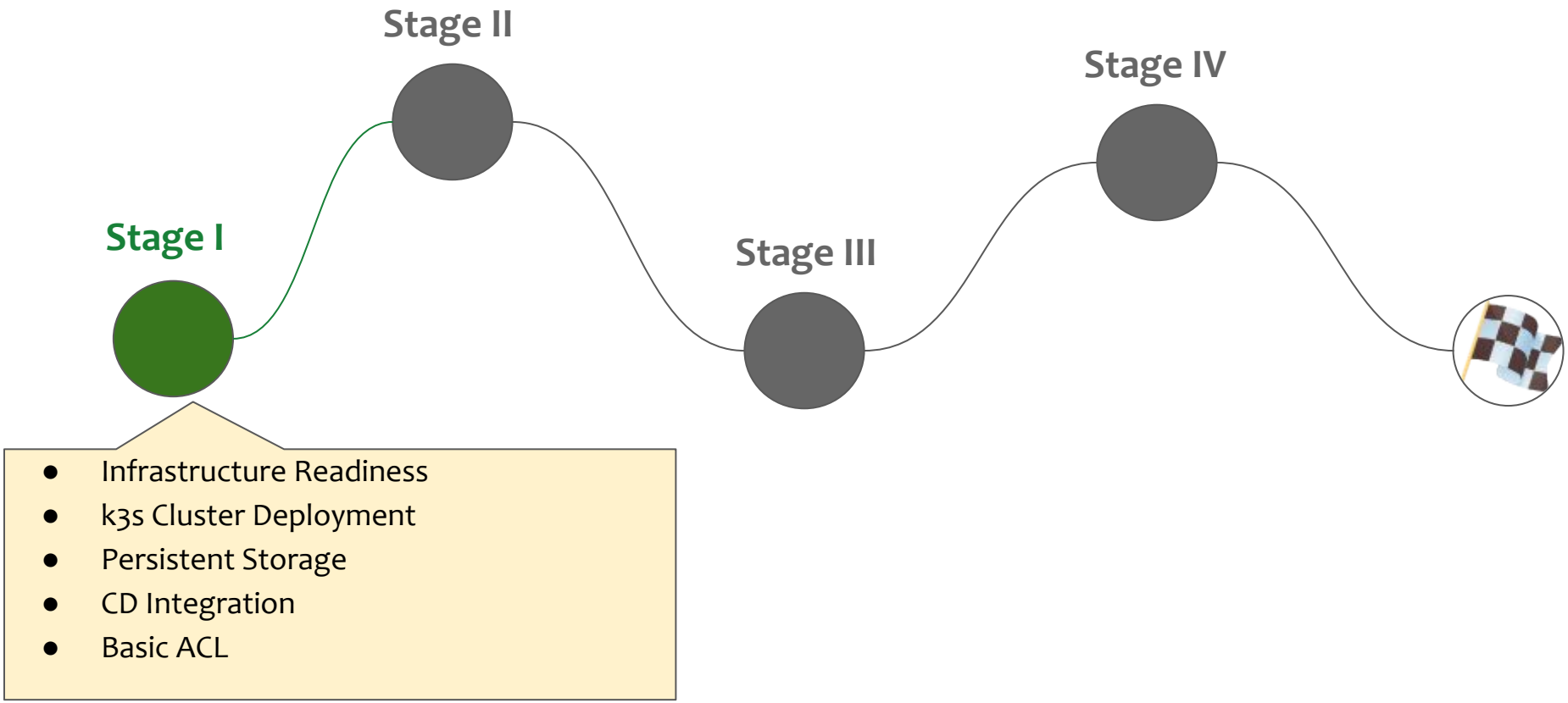
**Secure**
State Management of Microservices

**Develop**
Testing Mechanism for Microservices

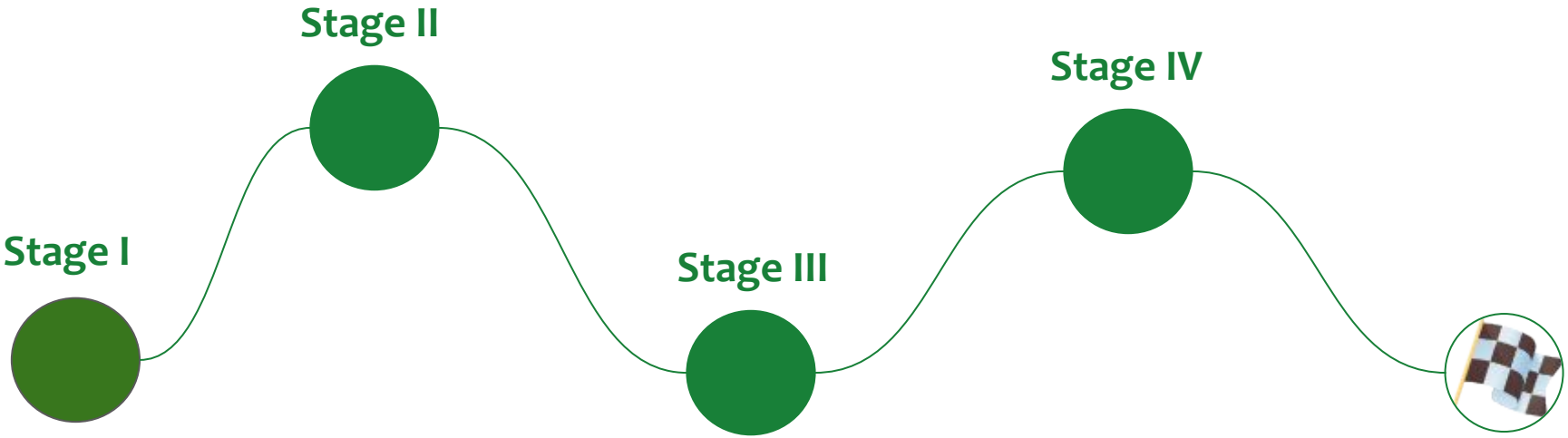
**Execute**
Security Review of Architecture

Stage I | Stage II | Stage III | Stage IV

# Thesis Impact



# Overall Project Impact




- Secure and Scalable Microservice Architecture
- Efficient Deployment and Management
- Distributed Storage Backend
- Confidential Computing
- Secure Network Infrastructure