2. odprti dan za gospodarstvo

SLAIF
Slovenska tovarna UI
Slovenian AI Factory

Uporabniško usmerjene storitve SLAIF

Domen Mongus, UM FERI

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

10. April 2026

Glavni cilji

Izvedba „cloud-alike“ storitev na HPC VEGA2:

- Poenostavljena uporaba
- Visoka zmožnost integracije v lastne rešitve

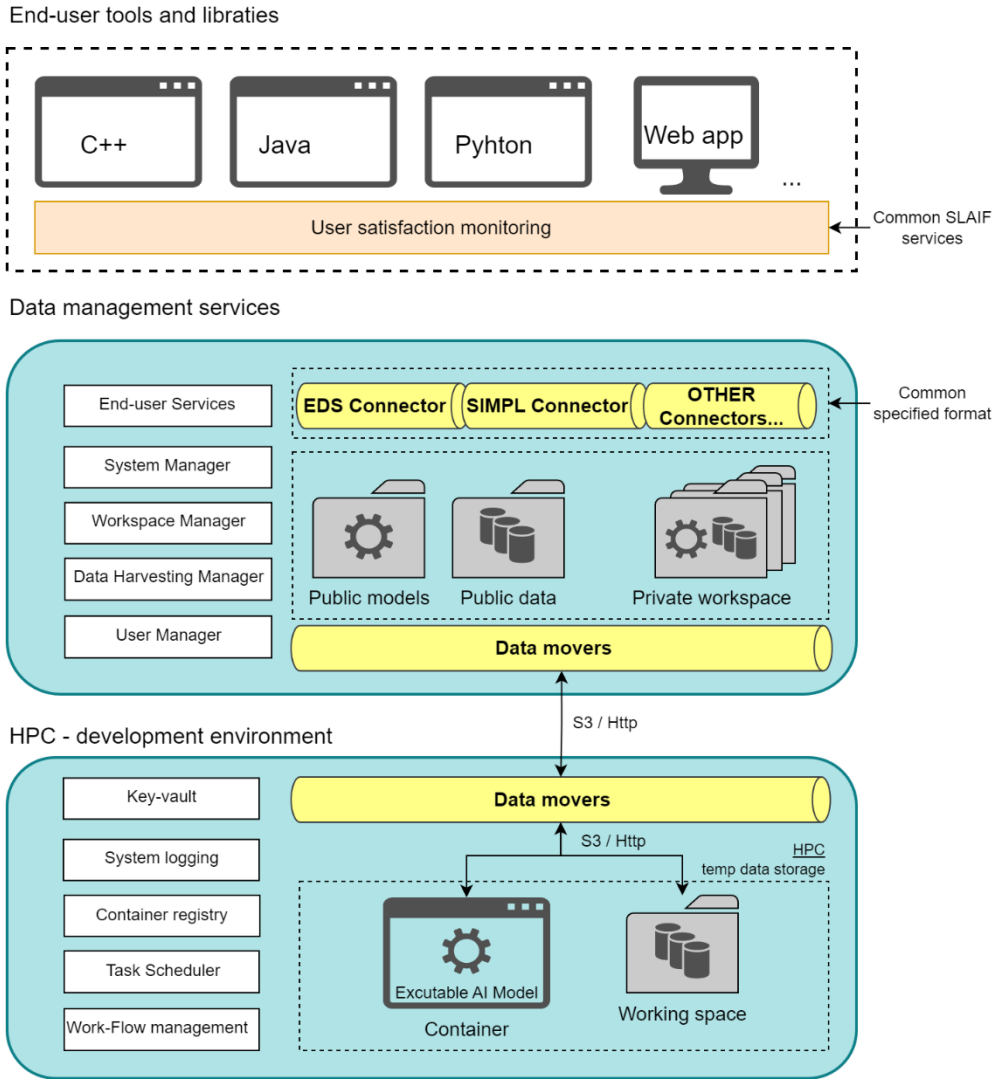
Podatkovni laboratorij:

- Podatki in storitve UI na enem mestu
- Pred-pripravljeni splošno-uporabni podatki na „srebrnem nivoju“
- Pred-pripravljene storitve UI

Namen:

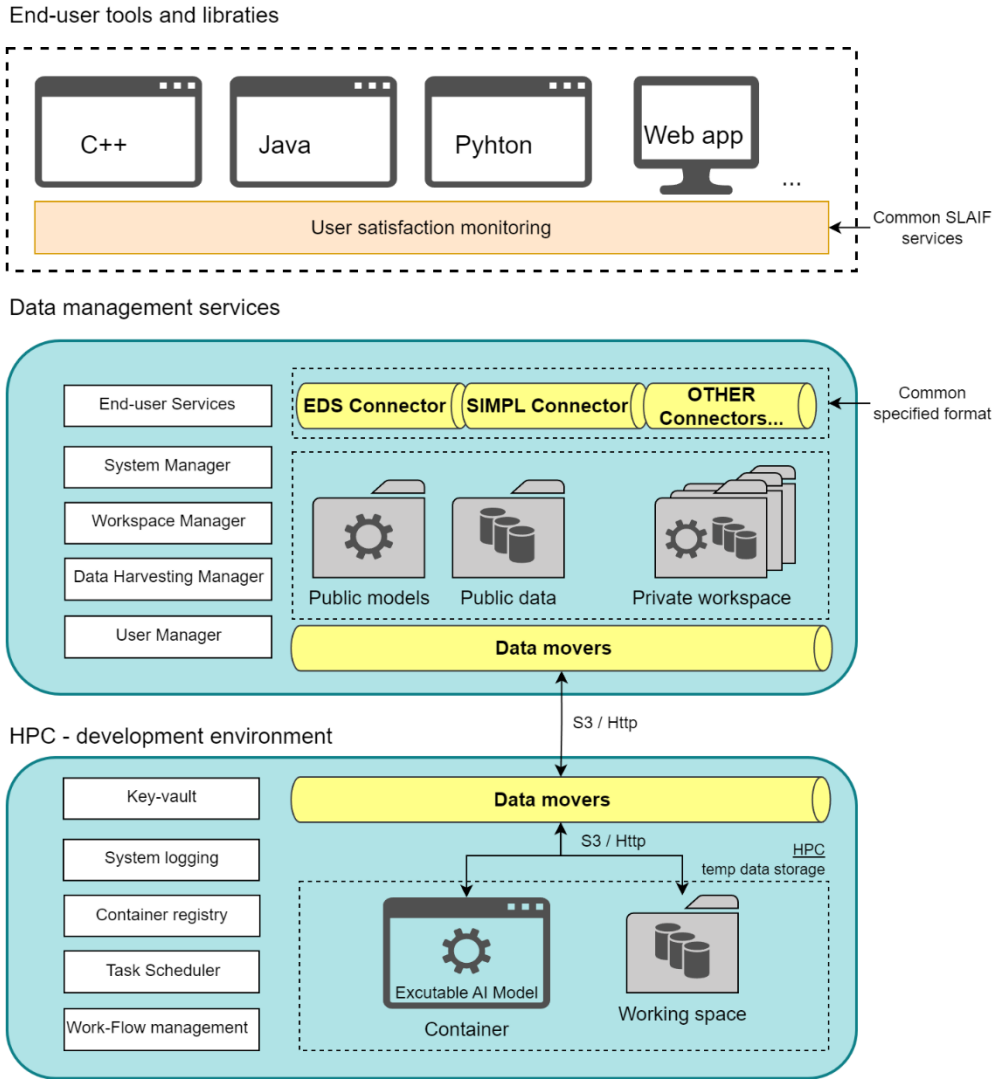
- Učenju, testiranje in izvedba (testna inferenca)

Uporabniško usmerjene storitve



- Generična uporabniška zgodba:
  - Prijava v sistem
  - Organizacija podatkov lastnih delovnih prostorov
    - Uvedba omejitev/deljenje dostopa do podatkov
  - Registracija zunanjih podatkovnih virov
  - Konfiguriranje zbiranja podatkov iz zunanjih podatkovnih virov
  - Programsko nalaganje privatnih podatkov in kontejnerja
  - Pogajanja o izvedbi
  - Izvedba in prenos rezultatov

Uporabniško usmerjene storitve



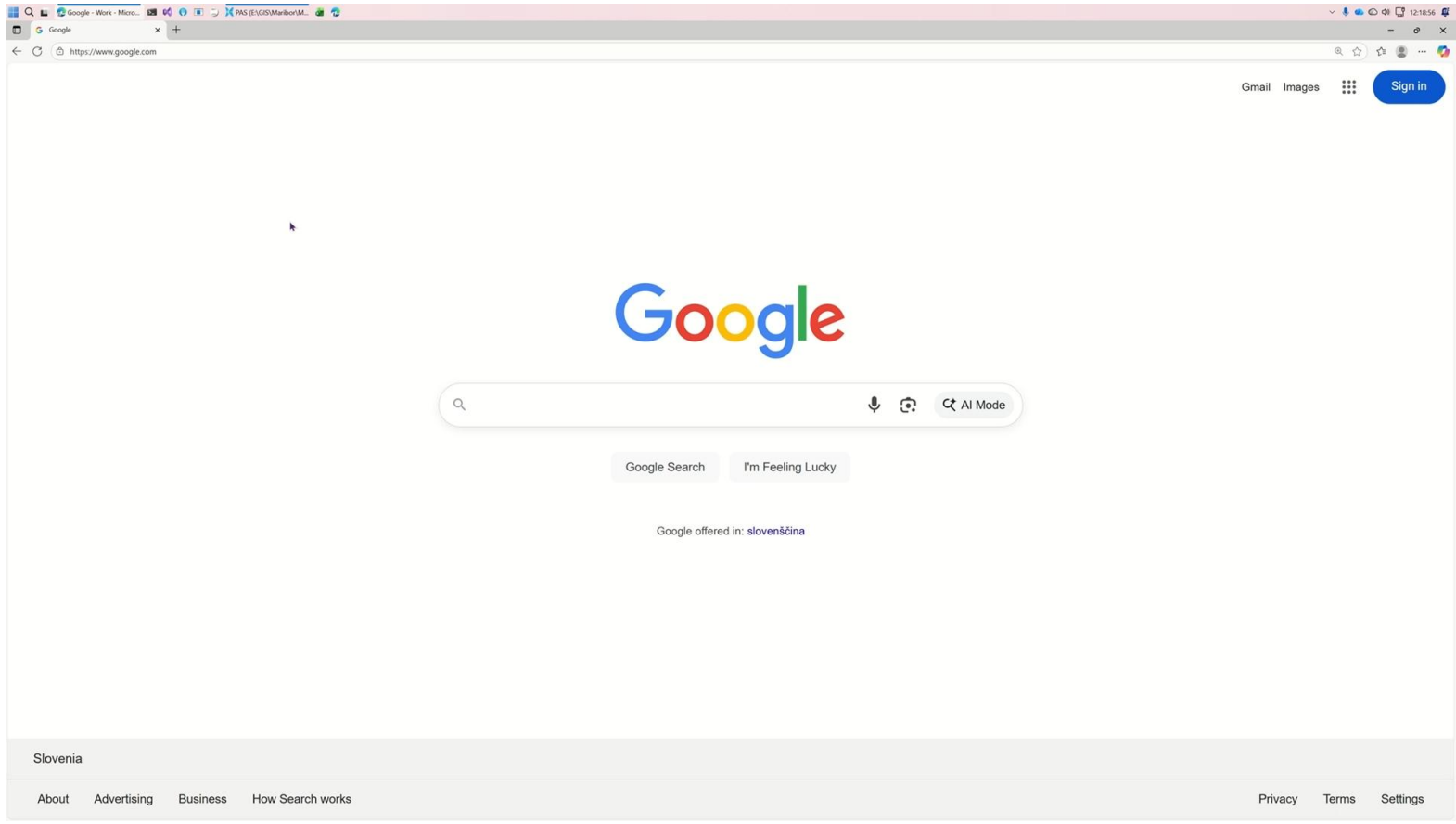
- Uporabniški nivo (SLAIF Toolbox)
  - Nabor orodij za enostavno integracijo storitev SLAIF v lastne rešitve
  - Sistem posodabljanje knjižnic
  - Sistem za spremljanje zadovoljstva uporabnikov
- Podatkovni nivo
  - DS Connectors
  - Data harvesting manager
  - Delovni prostori
- HPC Nivo
  - Izvedba „containerja“ nad dostavljenimi podatki

Delovni prostori

- Ključne lastnosti:
  - Nameščen na podatkovni center
  - Organizacija v mape
  - Hrani datoteke, baze, kontejnerje in rezultate izvedbe

- Uporabniška zgodba:
  - Ustvarjanje delovnega prostora
  - Ustvarjanje mape
  - Ustvarjanje podatkovne baze znotraj mape
  - Navigacija med mapami
  - Nalaganje posamezne datoteke
  - Generiranje API Key za programsko uporabljanje

Video uporabniške zgodbe skozi GUI



Delovni prostori

- Ključne lastnosti:
  - Nameščen na podatkovni center
  - Organizacija v mape
  - Hrani datoteke, baze, kontejnerje in rezultate izvedbe

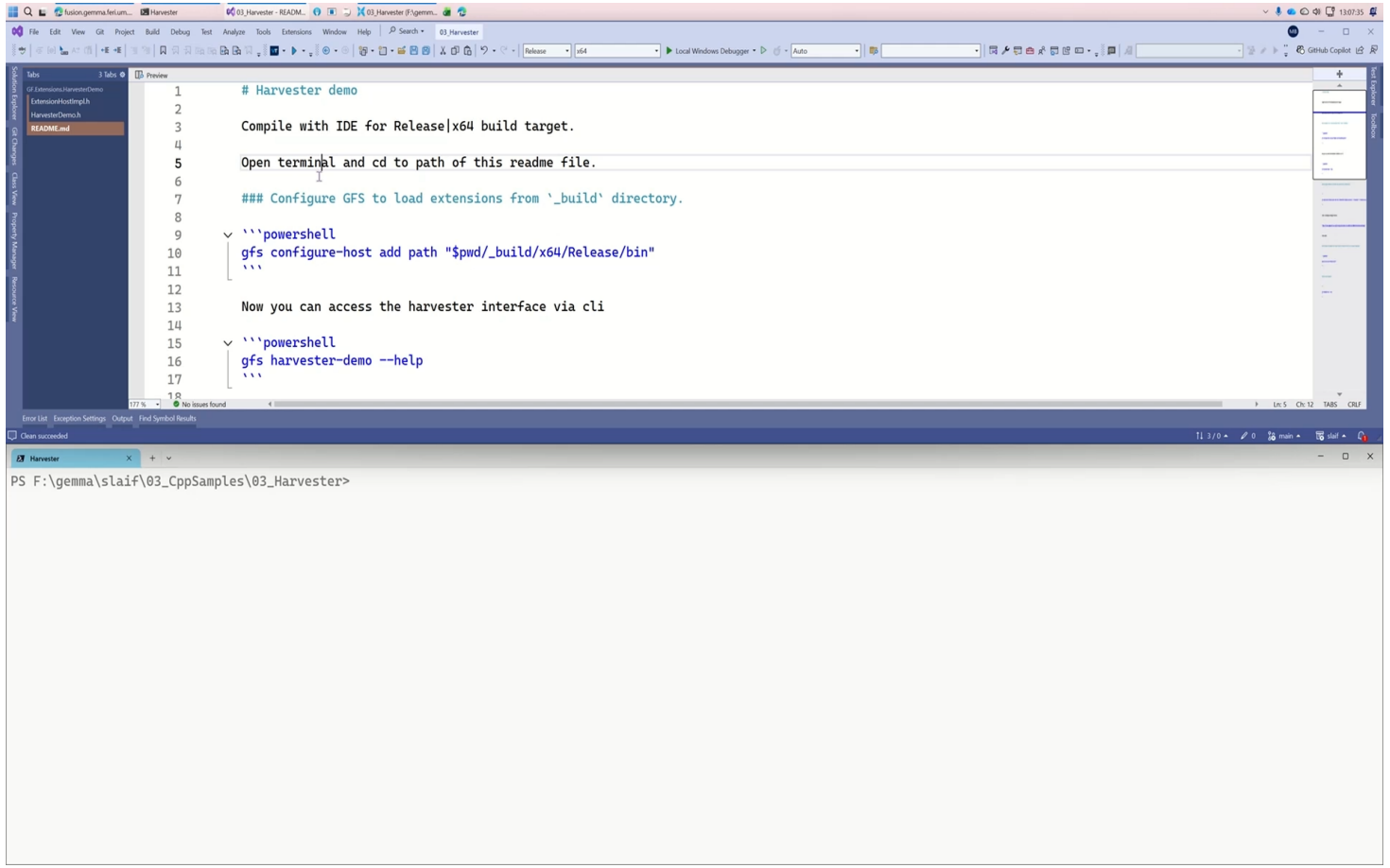
- Uporabniška zgodba:
  - Ustvarjanje delovnega prostora
  - Ustvarjanje mape
  - Ustvarjanje podatkovne baze znotraj mape
  - Navigacija med mapami
  - Nalaganje posamezne datoteke
  - Generiranje API Key za programsko uporabljanje

Programska koda uporabniške zgodbe

```
#include "pch.h"

using namespace Data;
using namespace Platform;
using namespace Platform::Network::Http;
using namespace Platform::Network;
using namespace Platform::Logger;
using namespace Platform::Process;
using namespace Platform::IO;
using namespace GF::Api::Services;
using namespace GF::Api::Workspace;
using namespace GF::Api::Data::Compression;
using namespace GF::Api::Database;

int main()
{
    Url workspace("https://fusion.gemma.feri.um.si/gf-test/api/ws/31ec45c5-39ca-4d0b-9a59-60b4073e7cc6/services/storage");

    // 01 Povezava na delovni prostor
    StorageServices ss(workspace.GetUrl());
    ss.SetApiKey(SecureString(Environment::Get("GF_WS_API_KEY")));

    // 02 Ustvari datoteko v "Mapa/Hello.txt"
    std::string fileContent = "Hello world!";
    ss.WriteSequentialFileStream("Mapa/Hello.txt", MemoryStream(Array(fileContent.data(), fileContent.size())));

    // 03 Ustvari podatkovno bazo "Costumers"
    Table costumers;
    costumers.SetName("Costumers");
    costumers.AddColumn("Id");
    costumers.AddColumn("Name");
    costumers.AddColumn("Email");
    costumers.SetPrimaryKey("Id");

    Array database = CreateSqliteDatabaseBlob({ costumers });
    ss.WriteSequentialFileStream("Databases/MyDatabase.db", MemoryStream(database));

    // 04 Dobi seznam vseh datotek na delovnem prostoru
    for (auto file : ss.GetFilesList("/", true)) {
        Log::Info("File: {}", file);
    }

    // 05 Ustvari podatkovni vir za novo bazo in posodobi delovni prostor
    Array wsdata = ss.ReadSequentialFileStream("ws.gfdb").Value.ReadToEnd();
    WorkspaceManager ws(wsdata.data(), wsdata.size());
    ws.AddFileDataSource(Uuid(), "Customers", "Databases/MyDatabase.db");
    ws.SaveChanges();
    ss.WriteSequentialFileStream("ws.gfdb", MemoryStream(ws.Serialize()));

    return 0;
}
```

Sistem za zbiranje podatkov

Video uporabniške zgodbe skozi GUI



- Ključne lastnosti:
  - Samodejno zbiranje splošno-uporabnih podatkov
  - Priprava splošno-uporabnih podatkov za UI
  - Poveza s podatkovnimi prostori
- Uporabniška zgodba:
  - Seznam že registriranih podatkovnih virov
  - Registracija novih podatkovnih virov
  - Nalaganje cevovodov za orkestracijo podatkov (transformacije/harmonizacije)
  - Določitev urnikov izvedbe zajema, časovni obseg vzdrževanja podatkov itd.
- Navodila za uporabo

Sistem za zbiranje podatkov

Programska koda za zbiranje podatkov

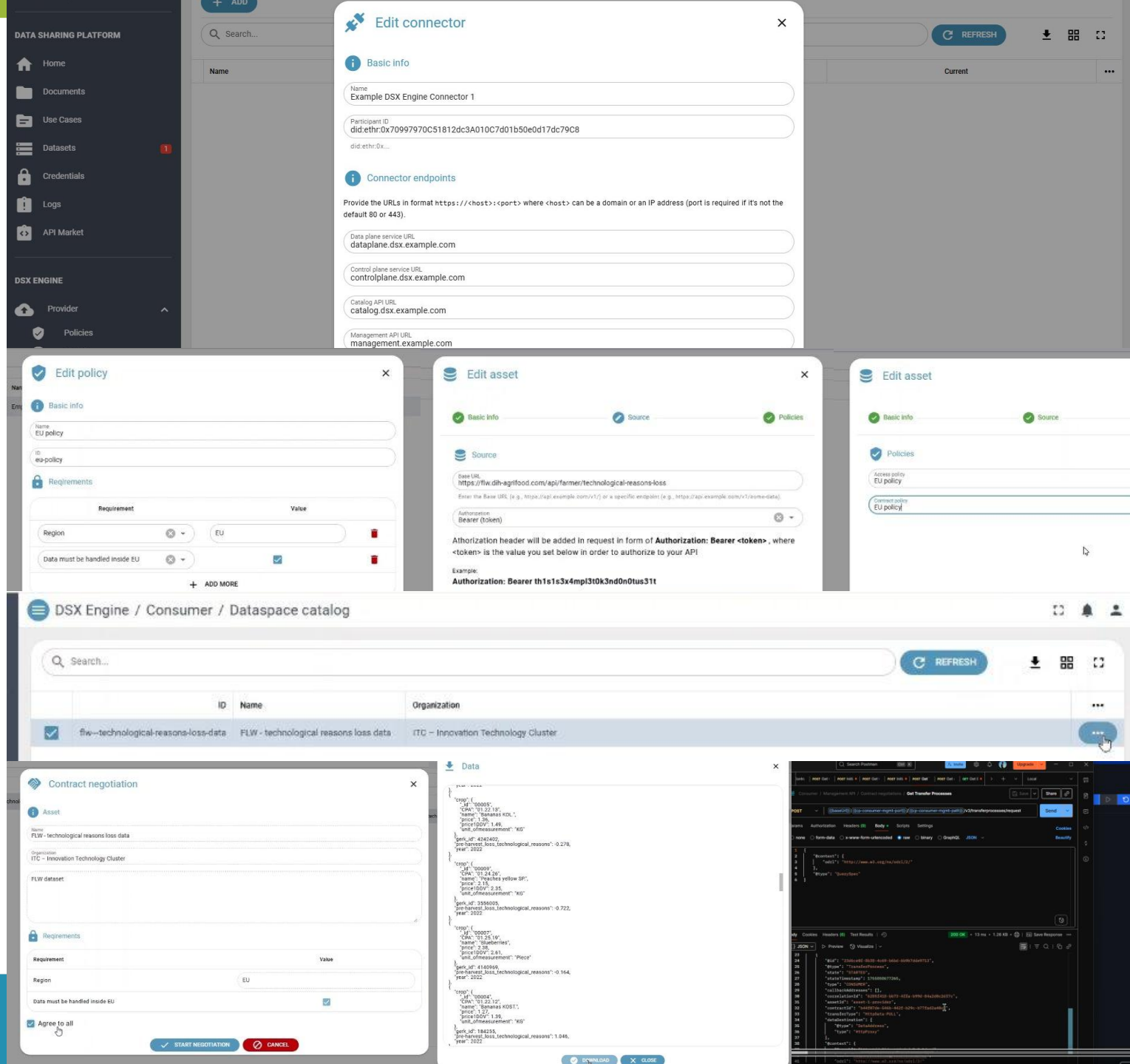
```
void Replicate(const ::Data::Properties& properties)
{
    using namespace std::string_literals;
    using namespace Platform::Network::Http;
    using namespace Platform::Network;
    using namespace Platform::IO;
    using namespace Platform;
    using namespace Data;

    auto [workspace, storage] = OpenWorkspace(properties.Get<InputWorkspaceProperty>());
    for (auto const& dataSource : workspace.GetDataSources()) {
        if (!dataSource.IsUrlDataSource()) continue;
        // Each datasource can have custom properties through which harvesting behaviour can be modified
        // Datasource (or any workspace object) properties can only be set programatically... Currently there is no UI support yet.
        // For example: For Url based datasources we could tell harvester how to download data. We can store anything such as request meth
        // Here we use it to check if we should harvest the datasource (by default yes)
        auto properties = dataSource.GetProperties();
        if (!properties.GetPropertyValueOrDefault("EnableHarvesting", Core::True).as<bool>()) continue;

        // If you need credentials to access the datasource you can store them securely in the workspace. For demo we don't need them
        SecureString username = properties.GetPropertyValueOrDefault("Username", Variant(SecureString(""))).as<SecureString>();
        SecureString password = properties.GetPropertyValueOrDefault("Password", Variant(SecureString(""))).as<SecureString>();

        Url url = dataSource.AsUrlDataSource().GetUrl();
        Log::Info("Harvesting: {}", url.GetUrl());

        HttpClient web(url.GetBaseUrl());
        HttpRequest request = web.CreateRequest(url.GetPathWithParameters(), AcceptTypes(), RequestMethod::Get);
        auto [response, err2] = request.GetResponseStreamAsync().AwaitWithError();
        if (err2.is_error()) {
            Log::Error("Failed to get data: {}", err2.message());
            // We continue with harvesting of other datasources
            continue;
        }
        Stream stream = response.GetStream();
        std::string fileName = properties.GetPropertyValueOrDefault("FileName", dataSource.GetName());
        std::string filePath = std::format("HARVESTED_DATA/{}", fileName);
        Log::Info("Downloading data {} and storing it into {}", Core::String::FormatSize(stream.GetLength()), filePath);
        storage.WriteSequentialFileStream(filePath, stream);
    }
}
```

- Ključne lastnosti:
  - Samodejno zbiranje splošno-uporabnih podatkov
  - Priprava splošno-uporabnih podatkov za UI
  - Poveza s podatkovnimi prostori
- Uporabniška zgodba:
  - Seznam že registriranih podatkovnih virov
  - Registracija novih podatkovnih virov
  - Nalaganje cevovodov za orkestracijo podatkov (transformacije/harmonizacije)
  - Določitev urnikov izvedbe zajema, časovni obseg vzdrževanja podatkov itd.
- Navodila za uporabo

EDS Connector

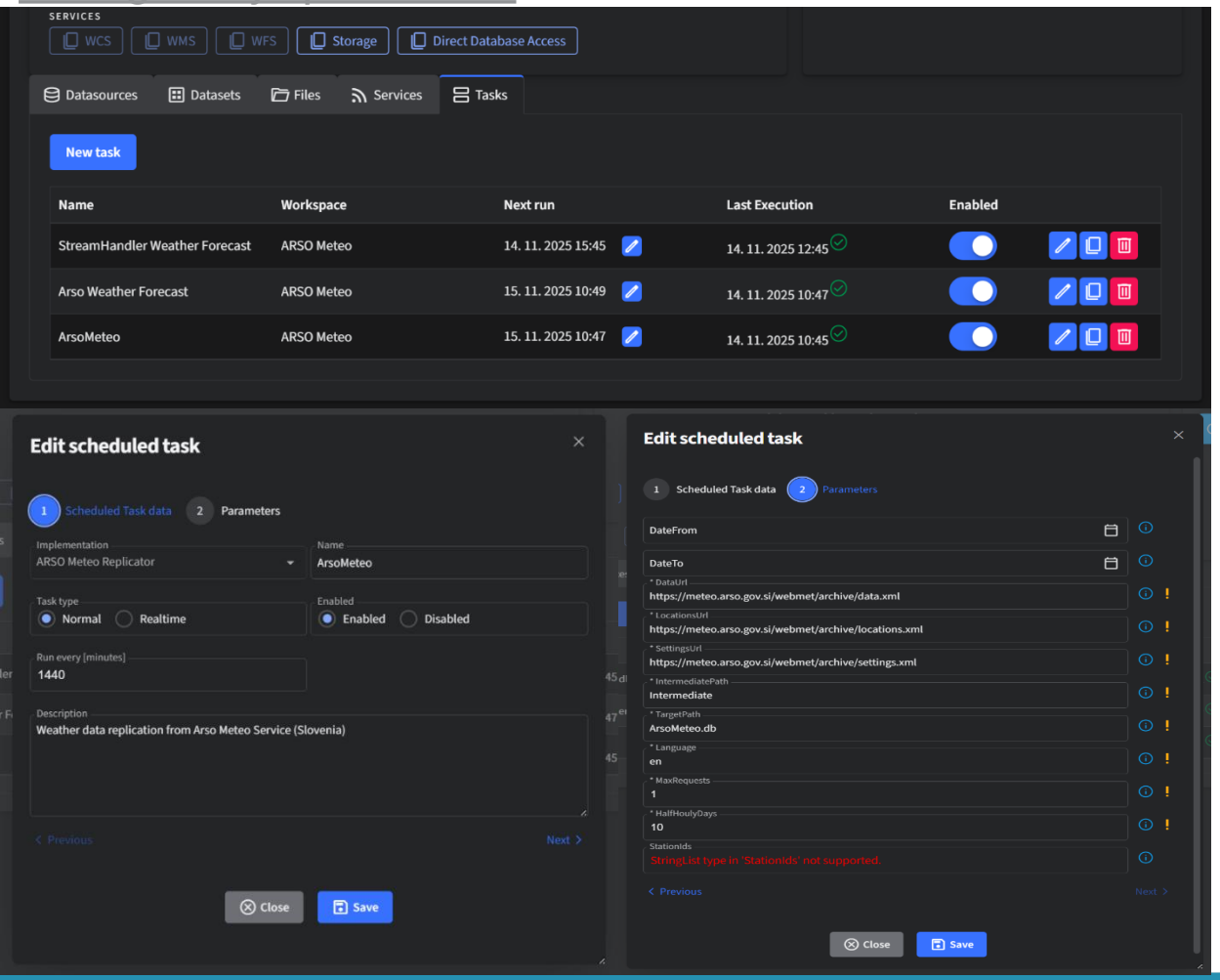
- Dostop do „skupnih“ EU in nacionalnih podatkovnih prostorov
  - Iskanje po katalogu podatkov
  - Preverjanje ponudb politik uporabe
  - Sklepanje pogodbo o prenosu podatkov
  - Prenos podatkov

- Mogočena tudi distribucija podatkov ostalim članom DS



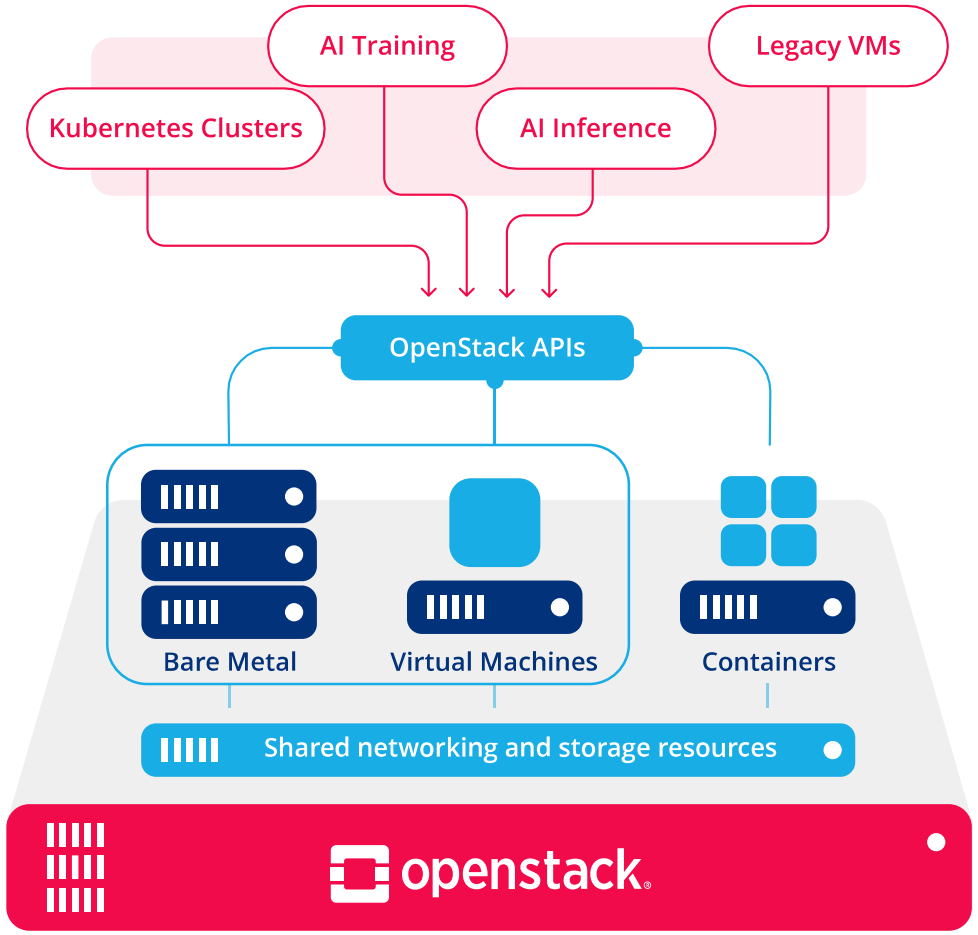
Sistem za zbiranje podatkov

Konfiguracija preko GUI



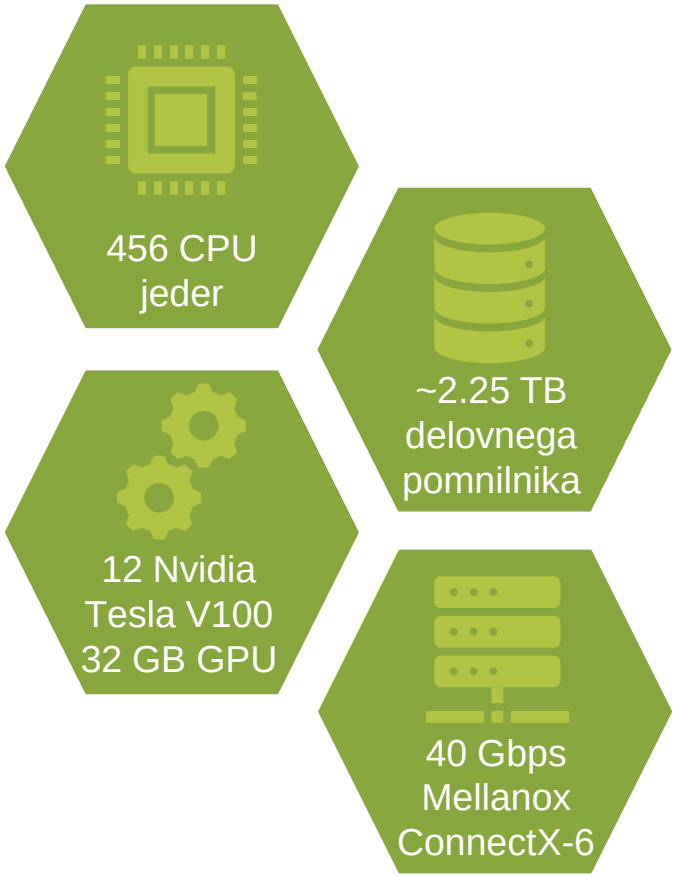
- Trenutno stanje:
  - Vzpostavljeno razvojno okolje (HPC RIVR)
    - podatkovni in izvajalni del
  - Validirana integracija s podatkovnim prostorom
  - V teku 6 PoC
- Naslednji koraki:
  - Registracija podatkovnih virov in prostorov
  - Izboljšave uporabniške izkušnje
  - Zbiranje odziva prvih uporabnikov ustrezne korekcije sistema
- Kako se vključiti?:
  - Delovni prostori: dms.slaif.lhrs.si/mgmt/SignIn
    - Navodila za uporabo

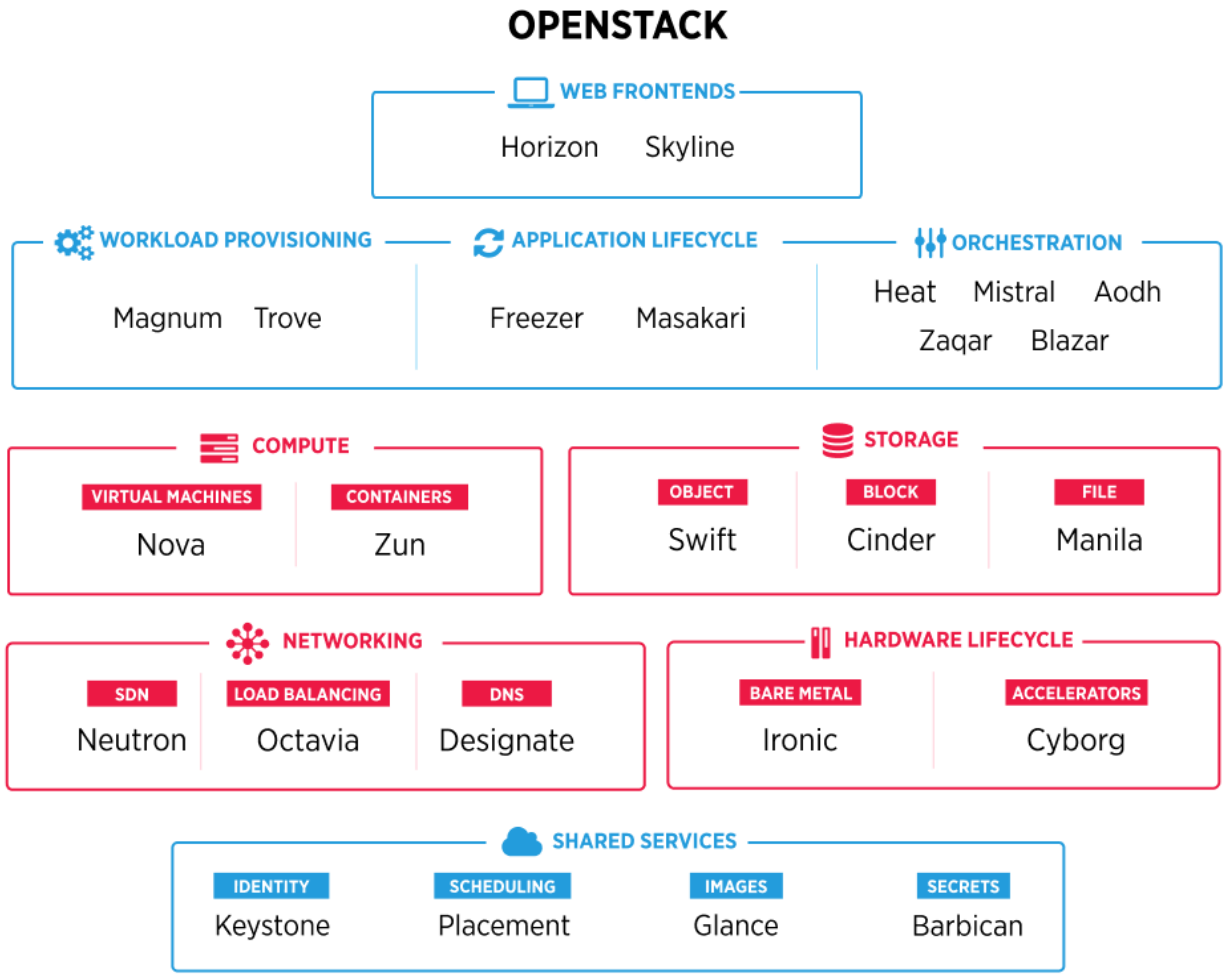
Platforma OpenStack®



https://www.openstack.org/

Tehnologije, storitve in računska moč

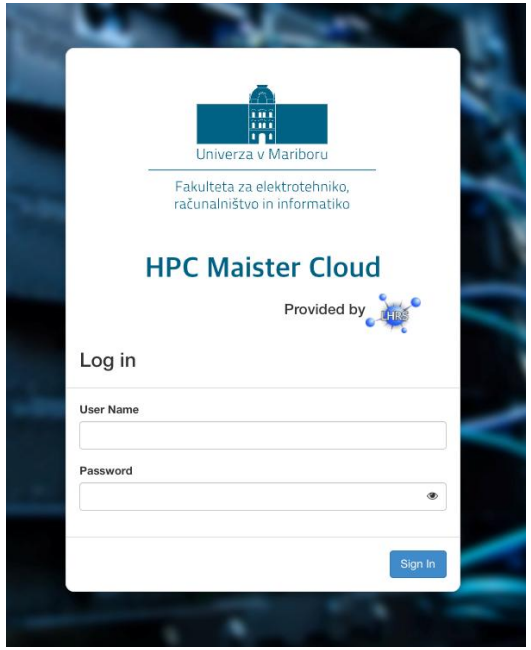


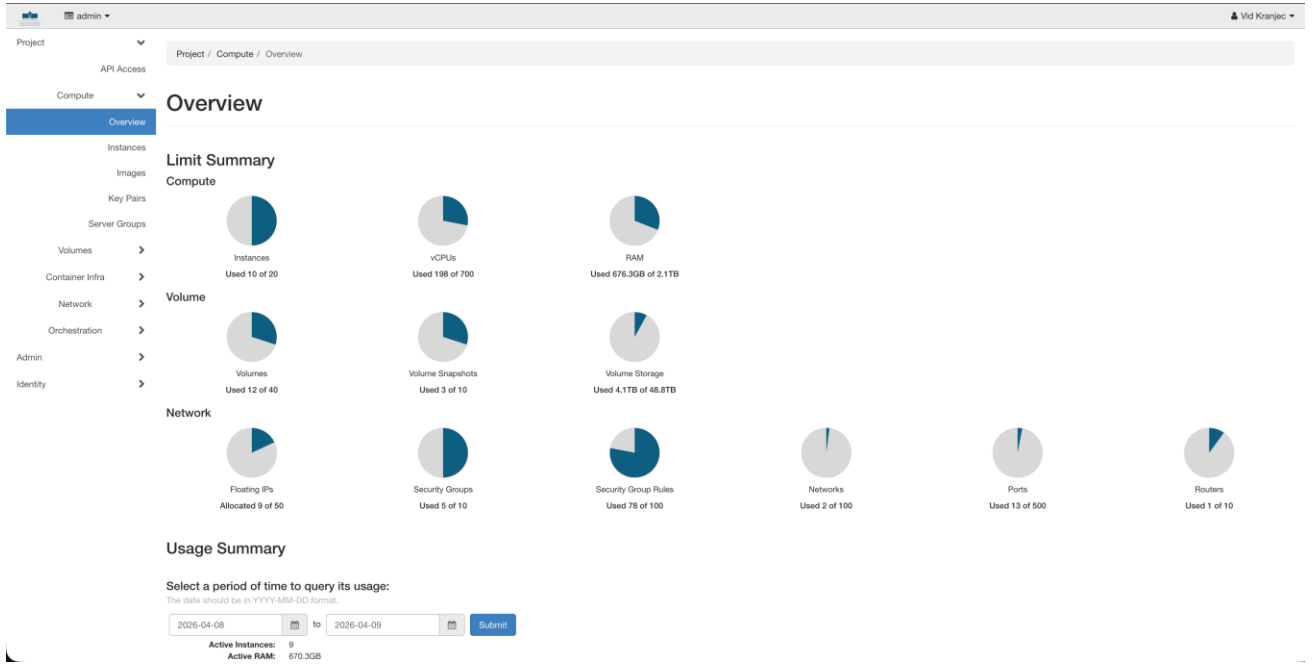


https://www.openstack.org/

Kako do dostopati do navideznega stroja (1)

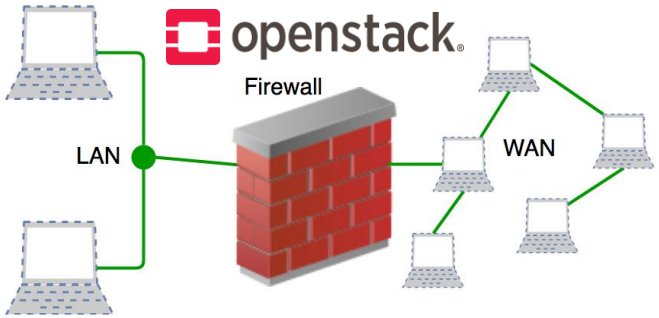
- SSH dostop
- RDP dostop
- Dostop preko Horizon dashboarda (https://horizon.slaif.lhrs.si) -> noVNC povezava



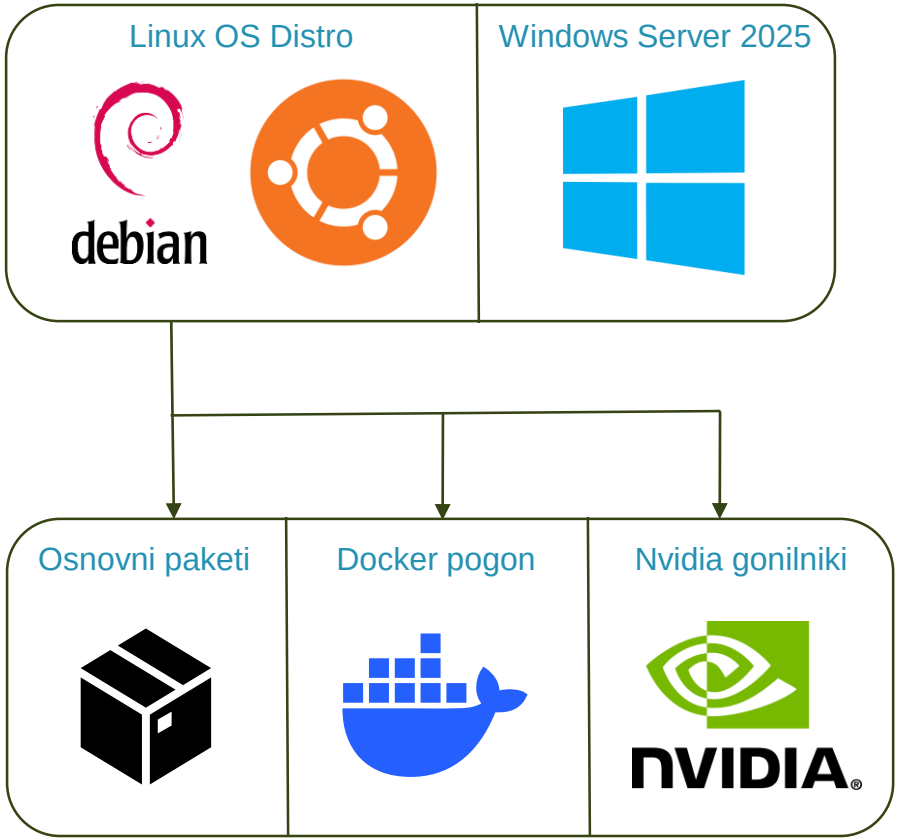


Kako do dostopati do navideznega stroja (2)

Flavour Name	vCPUs	RAM	Root Disk	Ephemeral Disk
cpu10.ram120.gpu2	10	120GB	0 GB	0 GB
cpu10.ram60.gpu2	10	60GB	0 GB	0 GB
cpu12.ram32	12	32GB	0 GB	0 GB
cpu128.ram512	128	488.3GB	0 GB	0 GB
cpu128.ram512.windows	128	488.3GB	0 GB	0 GB
cpu16.ram120.gpu2	16	120GB	0 GB	0 GB
cpu16.ram128.gpu4	16	128GB	0 GB	0 GB



https://www.geeksforgeeks.org/computer-networks/introduction-of-firewall-in-computer-network/



Kako dostopati do Kubernetes grozda

k8s-cluster

Show Certificate

- Sign Certificate
- Get Cluster Config
- Resize Cluster
- Rolling Cluster Upgrade
- Rotate Cluster Credentials
- Delete Cluster

Cluster Template

Name	k8s-template
ID	bb2ecb73-9a1e-4664-8025-ba66c032f554
COE	kubernetes
Image ID	fedora-coreos-38.20230806.3.0-openstack

Nodes

Master Count	1
Node Count	1
API Address	https://164.8.239.31:6443
Master Addresses	164.8.239.31
Node Addresses	164.8.239.44

Miscellaneous

Discovery URL	https://discovery.etcd.io/ee9a64630a6702cf48a55ce51f860cb4
Cluster Create Timeout	60 minutes
Keypair	magnum-key
Docker Volume Size	- GB
Master Flavour ID	k8s.small
Node Flavour ID	k8s.small
COE Version	v1.25.16-rancher1
Container Version	1.12.6

Labels

dns_nameserver	8.8.8.8
container_runtime	containerd
containerd_version	1.6.19
containerd_tarball_sha...	3262454d9b3581f4d4da0948f77dde1be51cfc42347a1548bc9ab6870b055815
containerd_tarball_url	https://github.com/containerd/containerd/releases/download/v1.6.19/containerd-1.6.19-linux-amd64.tar.gz
cgroup_driver	systemd
kube_tag	v1.25.16-rancher1
k8s_keystone_auth_tag	v1.25.6
cinder_csi_plugin_tag	v1.25.6
cloud_provider_tag	v1.25.6
helm_client_sha256	da36e117d6dbc57c8ec5bab2283222fbd108db86c83389eebe045ad1ef3e2c3b
helm_client_tag	v3.12.0
helm_client_url	https://get.helm.sh/helm-v3.12.0-linux-amd64.tar.gz

Stack

Stack ID	f26cb044-a9b2-476b-be5d-91babab633a2
Stack Faults	None

Record Properties

Created	4/9/26 10:11 AM
Updated	4/9/26 10:22 AM
ID	3f9dd58e-f71e-4cf1-8bf0-d9cd616af06a
Status	CREATE_COMPLETE
Status Reason	-
Health Status	HEALTHY

Uporabniški scenariji

1. Playground

SSH dostop
RDP dostop
Horizon dostop

2. AI Training

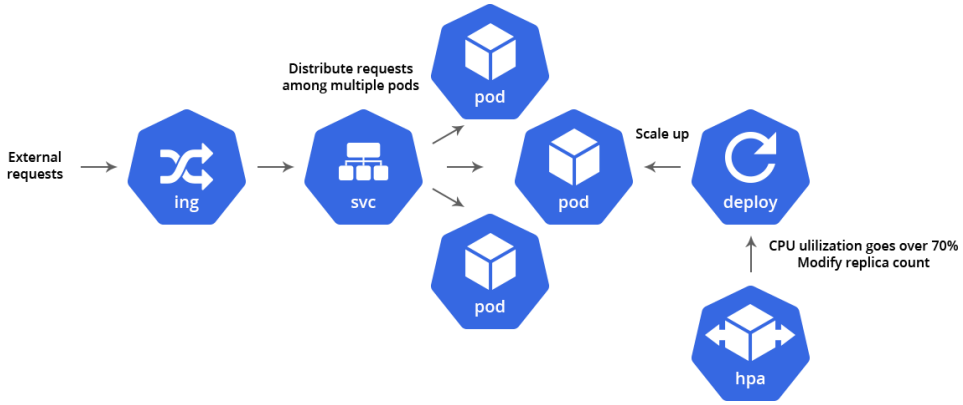
Navidezni stroji
Kubernetes grozdi

3. Service deployment

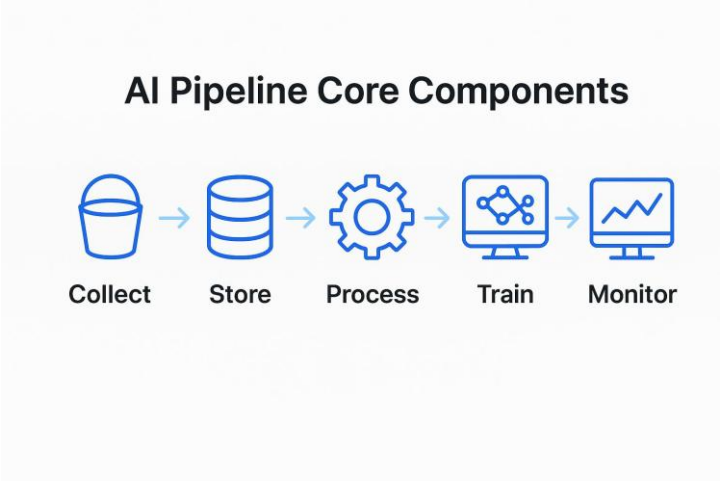
Navidezni stroji
Kubernetes grozdi

Posodobitve in načrti za prihodnost (1)

Kubernetes Autoscaling

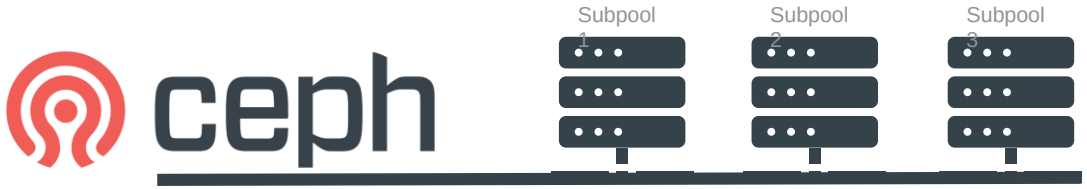


https://www.virtuozzo.com/company/blog/scaling-kubernetes/

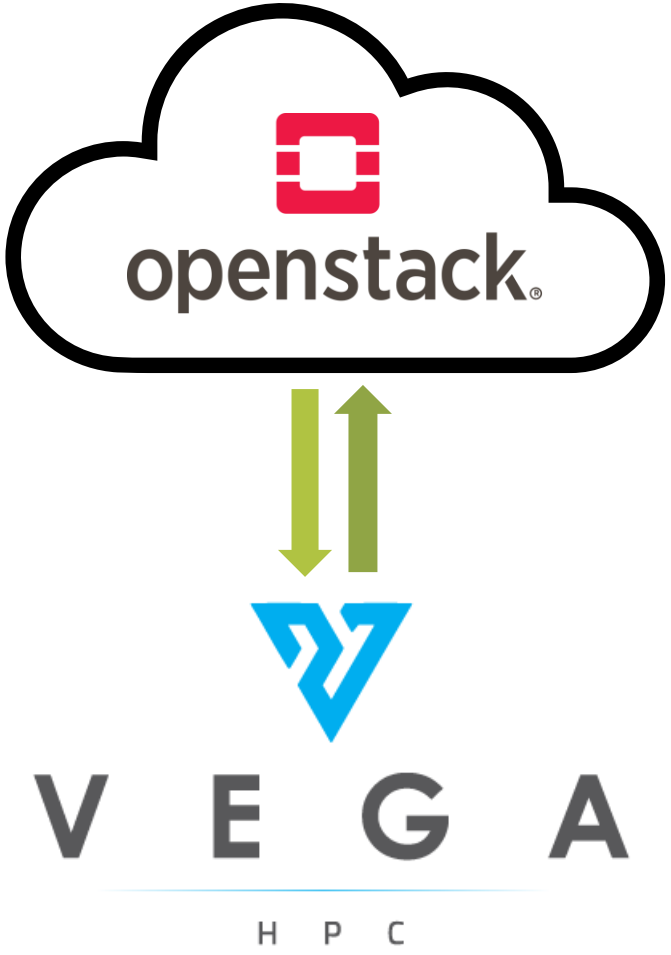


https://airbyte.com/data-engineering-resources/ai-data-pipeline

Izolirana podatkovna shramba



Posodobitve in načrti za prihodnost (2)



Vprašanja in predlogi?

Questions and suggestions?

SLAIF
Slovenska tovarna UI
Slovenian AI Factory

Kontakti

- Splošna vprašanja: domen.mongus@um.si;
- Delovni prostori: matej.brumen@um.si;
- Izvajanje vsebnikov: marko.ferme@um.si;

IZUM
Institute of Information Science, Maribor, Slovenia

arnes

University of Maribor

Univerza v Novi Gorici
Scientia Vinces 1995

Gospodarska zbornica Slovenije
Chamber of Commerce and Industry of Slovenia

Jožef Stefan Institute
Ljubljana, Slovenia

UNIVERZA V LJUBLJANI
University of Ljubljana

FIS
Fakulteta za informacijske študije
Faculty of information studies

Univerza na Primorskem
Università del Litorale

TP LJ01

Financerja / Financed by:

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA VISOKO ŠOLSTVO, ZNANOST IN INOVACIJE

EuroHPC
Joint Undertaking

Projekt SLAIF: Slovenska tovarna umetne inteligence je finančno podprlo Ministrstvo za visoko šolstvo, znanost in inovacije. Projekt je bil na razpisu skupnega podjetja EuroHPC izbran za financiranje v okviru programov Obzorje Evropa ter Digitalna Evropa.

SLAIF: Slovenian AI Factory has been funded by the Ministry of Higher Education, Science and Innovation of Republic of Slovenia. At a call by EuroHPC JU, the project has received a positive funding decision under Horizon Europe and Digital Europe Programmes.