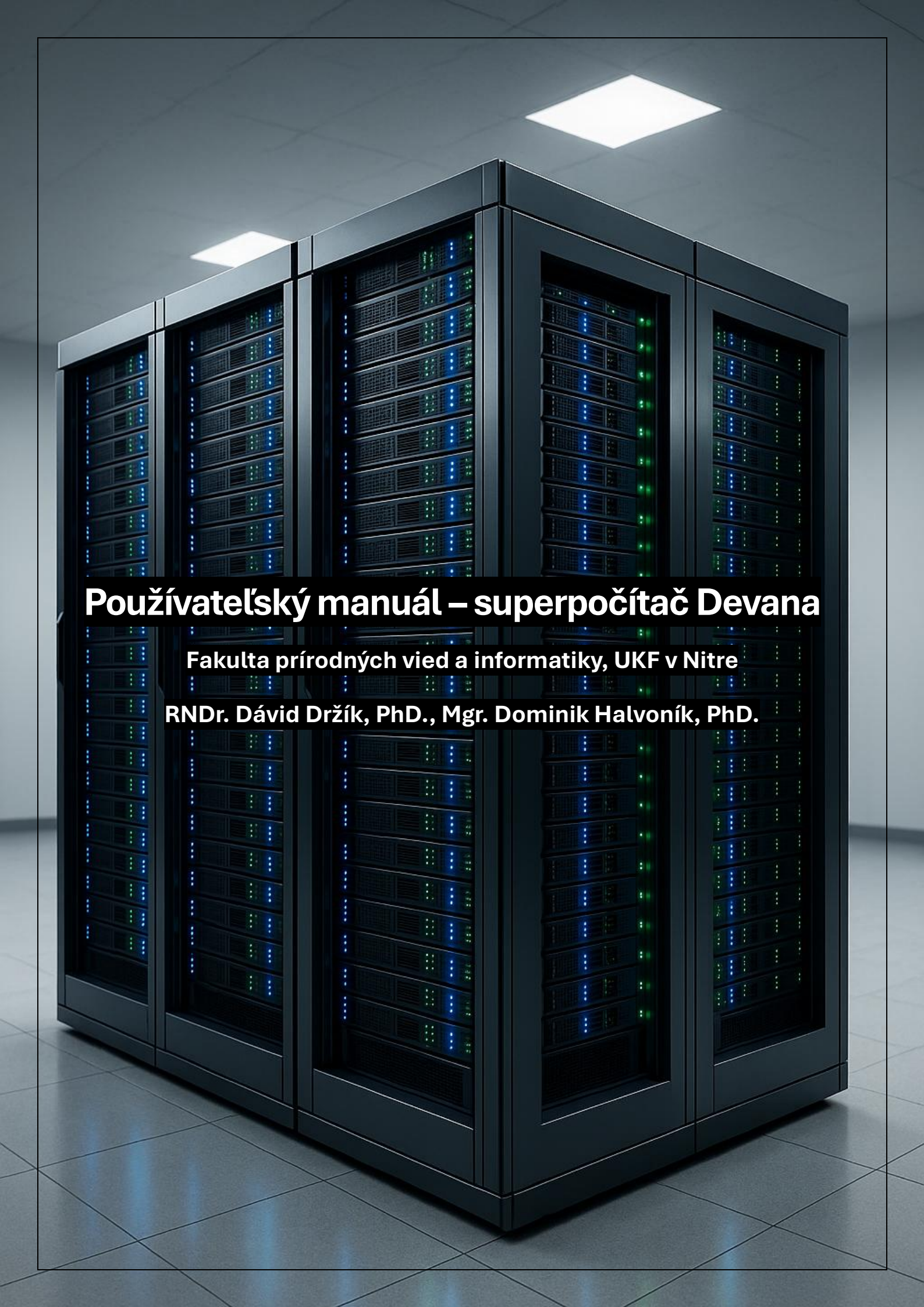




Používateľský manuál – superpočítač Devana

Fakulta prírodných vied a informatiky, UKF v Nitre

RNDr. Dávid Držík, PhD., Mgr. Dominik Halvoník, PhD.

OBSAH

1	Čo je to Devana?	3
2	Registrácia	3
3	Získanie projektu	3
3.1	Typy projektov	4
3.2	Vytvorenie testovacieho projektu	4
4	Pripojenie sa k HPC zdrojom	6
4.1	Prístup a prihlásenie	6
4.2	Prieskumník súborov	7
4.3	Vytvorenie virtuálneho prostredia pre Python	8
5	Spustenie veľkých výpočtových úloh	12
5.1	Anatómia spúšťacieho skriptu	12
5.2	Vytvorenie a správa úloh cez Open OnDemand Job Composer	12
5.3	Praktický príklad: skript pre výpočtovú úlohu	14
5.4	Spustenie a monitorovanie úlohy	17
6	Prenos veľkých dátových súborov	19
6.1	Generovanie SSH kľúčov	19
6.2	Pridelenie verejného kľúču k vášmu kontu	20
6.3	Prepojenie SFTP klienta s vaším súkromným kľúčom	22
7	Záver	26
7.1	Ďalšie kroky a podpora	26
7.2	Zdroje	26

1 ČO JE TO DEVANA?

Devana je nový HPC klaster Výpočtového centra SAV (spustený v 2. štvrťroku 2023). Má 148 uzlov s 9 472 jadrami, 38 TB RAM a 32 GPGPU akcelerátormi (8 uzlov s NVIDIA A100). Poskytuje výkon ~800 TFLOP/s, využíva InfiniBand HDR sieť a procesory Intel Xeon Ice Lake. Beží na CentOS 7, ponúka široký softvérový balík spravovaný cez Easybuild a moduly. Zdroje sú určené na nekomerčný výskum a vývoj, dostupné pre akademickú obec, priemysel i verejný sektor podľa podmienok.

Všetky informácie sú uvedené v [OFICIÁLNEJ POUŽÍVATEĽSKEJ DOKUMENTÁCII](#).

2 REGISTRÁCIA

Vytvorte si účet na používateľskom portáli vyplnením [registračného formulára](#). Správcovia superpočítača Devana vás budú kontaktovať, aby overili vašu totožnosť. Následne dostanete jednorazový odkaz na nastavenie hesla k vášmu účtu.

Poznámka autora: Ak použijete napr. GMAIL účet (a nie účet inštitúcie), tak Vás budú telefonicky kontaktovať s cieľom zistenia, kto ste, kde pracujete, čo na superpočítači budete spúšťať, atď.

3 ZÍSKANIE PROJEKTU

Po úspešnej registrácii a potvrdení účtu sa prihláste na [tomto odkaze](#). Po prihlásení by ste mali vidieť domovskú stránku. Nás teraz zaujíma konkrétne táto časť:



Na superpočítači Devana môžete spúšťať úlohy iba v rámci **konkrétneho projektu**. Ešte pred vytvorením projektu je dôležité spomenúť, že projekty sa delia na dve hlavné kategórie: testovacie a štandardné.

3.1 Typy projektov

a) Testovacie projekty

- Slúžia na **počiatočné testovanie softvéru a úloh**.
- Oprávnení žiadateľa: vedci a výskumníci z vysokých škôl, SAV, organizácií verejnej správy a registrovaných súkromných spoločností v SR.
- Pridelenie:
 - trvanie: max. 4 mesiace
 - alokácia: **50 000 CPU hodín** a **700 GPU hodín**
- Po testovaní je možné prejsť na štandardný projekt.

b) Štandardné projekty

- Slúžia na **kompletný výskum** a **rozsiahlejšie projekty**.
- Oprávnení žiadateľa: rovnakí ako pri testovacích projektoch, prístup je len pre nekomerčný výskum.
- Pridelenie:
 - trvanie: max. 1 rok
 - alokácia: 26 miliónov CPU hodín a 92 000 GPU hodín (rozdelených medzi všetky akceptované projekty)
- **Projekt musí byť podrobne definovaný (cieľ, motivácia, plán využitia výpočtového času).**
- Po ukončení je **potrebné vypracovať záverečnú správu**, ktorú koordinátor zverejní.
- Komerčné využitie nie je povolené. Nevyčerpaná alokácia po ukončení projektu zaniká.

3.2 Vytvorenie testovacieho projektu

V tomto návode sa budeme venovať vytvoreniu **testovacieho projektu**. Na bočnom paneli kliknite na **Moje projekty**, prípadne na domovskej stránke kliknite na ikonu **Projekt** alebo použite [tento odkaz](#).

Zvoľte možnosť **Vytvoriť nový testovací projekt**. Formulár, ktorý sa zobrazil, je potrebné vyplniť nasledovne:

1. Nadpis

Krátky názov projektu, ktorý ho jednoznačne identifikuje.

2. Vedecký názov projektu

Formálny názov projektu používaný vo vedeckej dokumentácii.

3. Akademický odbor

- Vyberte z rozbaľovacieho zoznamu hlavný odbor projektu.

4. Abstrakt

Krátky popis projektu, zahrňujúci:

- zámer projektu
- hlavné ciele
- metódy alebo postupy, ktoré budú testované

5. Softvérové nástroje

Uvedte softvér alebo knižnice, ktoré budete používať, spolu s informáciou o dostupnosti na superpočítači.

6. Zdôvodnenie použitia HPC prostriedkov

Vysvetlite potrebu použitia výpočtového výkonu superpočítača:

- prečo je potrebný HPC výkon
- uskutočniteľnosť testovania na danom výpočtovom prostriedku

7. Ďalší členovia projektového tímu

Uvedte mená a kontakty ďalších účastníkov projektu, ktorí budú mať prístup k zdrojom. Účastníci musia byť zaregistrovaní v systéme.

Poznámka autora: Testovací projekt nemusí byť extrémne podrobne vypracovaný. Slúži predovšetkým na testovanie softvéru a úloh. Väčšina podaných testovacích projektov je schválená do približne 5 dní.

Pre vytvorenie štandardného projektu sa vyhlasuje Výzva na podávanie projektov 2–3-krát ročne. Viac informácií nájdete na [tomto odkaze](#).

Informácie o aktuálnosti projektu a pridelených (a spotrebovaných) alokovaných CPU a GPU hodín môžete vidieť na hlavnej stránke nasledovne:

PROJEKTY				
Akronym	Členovia tímu	Status	Dátum	
p565-24-t	ddrzik+/-		2024-02-12	

Vysvetlivky: zodpovedný riešiteľ projektu je uvedený ako prvý, silným písmom
 Status projektu: Draft (nezverejnený) Čakajúci na schválenie Schválený Predĺžený Ukončený Čakajúci na zmenu spoluriešiteľov

Projektové čísla

Akronym	Pridelená alokácia CPU (core*h)	Pridelená alokácia GPU (gpu*h)	Spotrebovaná alokácia CPU (core*h)	Spotrebovaná alokácia GPU (gpu*h)
p565-24-t	50000	781	 8633 17.2%	 512 65.5%

Pridelené číslo projektu

4 PRIPOJENIE SA K HPC ZDROJOM

Na prihlásenie sa k výpočtovým zdrojom superpočítača Devana môžete použiť:

- SSH klienta,
- Open OnDemand (webové rozhranie).

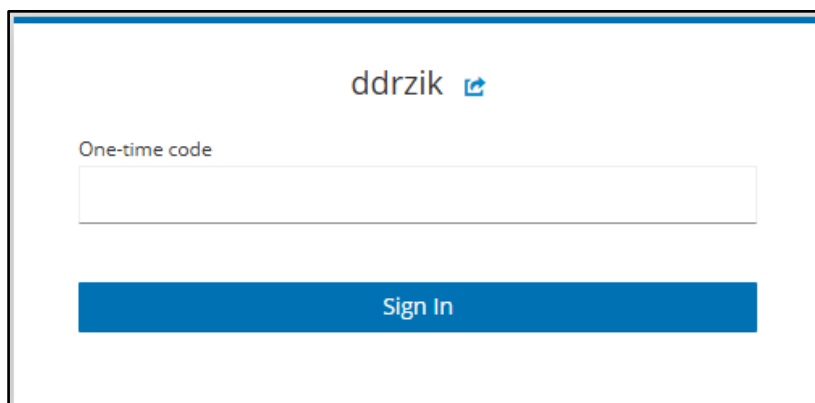
V nasledujúcich podkapitolách sa budeme venovať Open OnDemand prístupu. Ak preferujete prihlásenie cez SSH, pokračujte kapitolou 6.

Open OnDemand je webový portál poskytujúci prístup k výpočtovým zdrojom klastra. Umožňuje spravovať súbory, odosielať úlohy a používať grafické rozhrania priamo cez webový prehliadač.

4.1 Prístup a prihlásenie

1. Otvorte webový prehliadač a prejdite na: <https://ood.devana.nsc.sk>
2. Na prihlasovacej stránke zadajte svoje používateľské meno a heslo.
3. Po úspešnom zadaní sa zobrazí výzva na zadanie **jednorazového hesla (OTP)** ako súčasť dvojfaktorovej autentifikácie (2FA).

- Ak ste 2FA ešte nenastavili, nastavíte si to teraz.
- **Poznámka autora:** Google Authenticator funguje so systémom spoľahlivo.

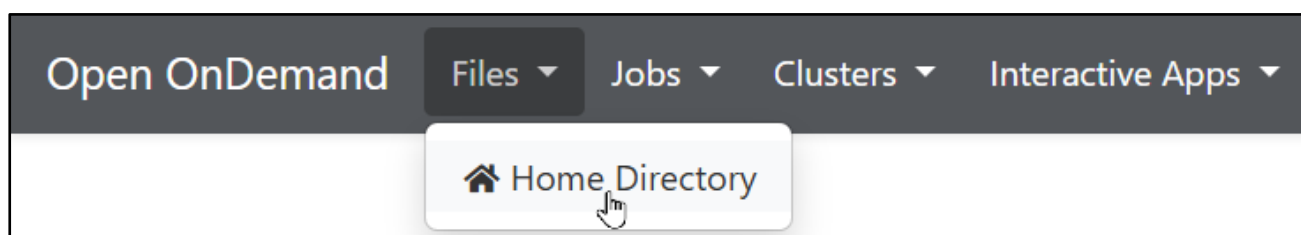


4. Po zadaní 2FA kódu ste prihlásení a môžete začať používať Open OnDemand.

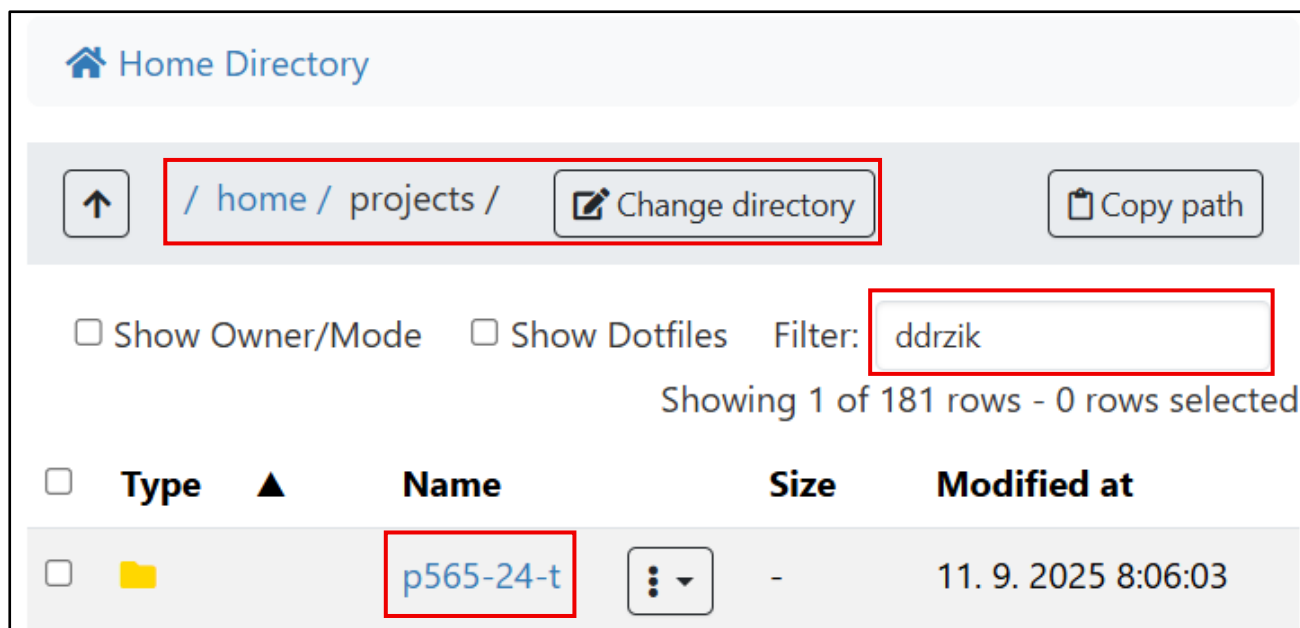
Po overení budete presmerovaní na **ovládací panel Open OnDemand**, pričom v ľavom hornom rohu ovládacieho panela sa nachádza niekoľko ponúk, ktoré umožňujú prístup k súborom, úlohám a vašim aktuálnym interaktívnym reláciám.

4.2 Prieskumník súborov

Ponuka **Files** na ovládacom paneli Open OnDemand poskytuje prístup k súborom klastra priamo cez webový prehliadač. Predvolene sa prieskumník otvára vo vašom **domovskom adresári** (/home/meno).



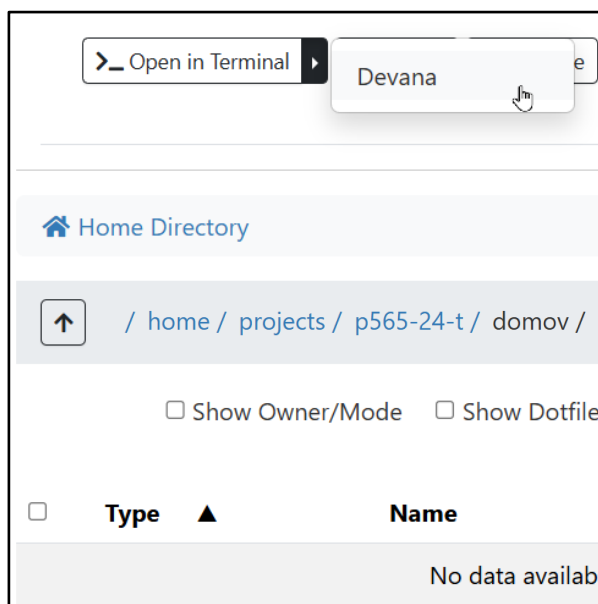
Dôležité: Keďže máte vytvorený projekt a pridelené výpočtové zdroje, nesmiete pracovať v domovskom priečinku. Všetky súbory súvisiace s projektom je potrebné ukladať a spravovať v priečinku vášho projektu (**/home/projects/čísloProjektu**). Tým sa zabezpečí správne pridelovanie zdrojov a ochrana dát projektu. Vyhľadanie vášho projektu v priečinku *projects* je podľa vášho mena skutočne intuitívne.



Viac informácií o manipulácii so súbormi a priečinkami nájdete na [tomto odkaze](#).

4.3 Vytvorenie virtuálneho prostredia pre Python

V koreňovom adresári vášho projektu si vytvorte pracovný priečinok (napr. domov), v ktorom budete pracovať (v **tomto priečinku otvoríte Terminál s názvom Devana**). Pre vytvorenie virtuálneho prostredia pre jazyk Python postupujte podľa nasledujúcich krokov.



1. Kontrola dostupných verzií Pythonu

V shelli zistíte dostupné verzie príkazom:

```
module avail python
```

```
ddrzik@login02 domov > module avail python
```

```
----- /storage-apps/easybuild/modules/bio
UMI-tools/1.0.1-foss-2020a-Python-3.8.2

----- /storage-apps/easybuild/modules/devel
pkgconfig/1.5.5-GCCcore-11.3.0-python

----- /storage-apps/easybuild/modules/lang
Python/2.7.18-GCCcore-9.3.0      Python/3.8.6-GCCcore-10.2.0      Python/3.10.4-GCCcore-11.3.0-bare
Python/2.7.18-GCCcore-10.2.0    Python/3.9.5-GCCcore-10.3.0-bare Python/3.10.4-GCCcore-11.3.0
Python/2.7.18-GCCcore-11.2.0-bare Python/3.9.5-GCCcore-10.3.0      Python/3.10.8-GCCcore-12.2.0-bare
Python/2.7.18-GCCcore-11.3.0-bare Python/3.9.6-GCCcore-11.2.0-bare Python/3.10.8-GCCcore-12.2.0
Python/3.8.2-GCCcore-9.3.0      Python/3.9.6-GCCcore-11.2.0      Python/3.11.3-GCCcore-11.3.0
```

2. Načítanie zvolenej verzie do prostredia

Vybranú verziu načítate pomocou:

```
module load Python/3.11.3-GCCcore-11.3.0
```

```
ddrzik@login02 domov > module load Python/3.11.3-GCCcore-11.3.0
ddrzik@login02 domov > █
```

3. Vytvorenie virtuálneho prostredia

Vytvorte si vlastný virtuálny environment, do ktorého budete inštalovať potrebné knižnice:

```
python3 -m venv venv3.11.3
```

Názov virtuálu je zvýraznený tučným písmom, konkrétny názov si môžete zvoliť podľa vlastného uváženia.

```
ddrzik@login02 domov > module load Python/3.11.3-GCCcore-11.3.0
ddrzik@login02 domov > python3 -m venv venv3.11.3
ddrzik@login02 domov > █
```

4. Aktivácia prostredia

V prípade potreby nastavte premennú MODULEPATH (ak ju už máte správne nastavenú, tento krok môžete vynechať):

```
export MODULEPATH=/storage-apps/Modules:/storage-apps/intel-
v.2023/oneapi/modulefiles:/storage-apps/easybuild/modules/bio:/storage-
apps/easybuild/modules/cae:/storage-apps/easybuild/modules/chem:/storage-
apps/easybuild/modules/compiler:/storage-apps/easybuild/modules/data:/storage-
apps/easybuild/modules/devel:/storage-apps/easybuild/modules/geo:/storage-
apps/easybuild/modules/lang:/storage-apps/easybuild/modules/lib:/storage-
apps/easybuild/modules/math:/storage-apps/easybuild/modules/mpi:/storage-
apps/easybuild/modules/numlib:/storage-apps/easybuild/modules/perf:/storage-
apps/easybuild/modules/phys:/storage-apps/easybuild/modules/system:/storage-
apps/easybuild/modules/toolchain:/storage-apps/easybuild/modules/tools:/storage-
apps/easybuild/modules/vis
```

Poznámka autora: Mne to však bez toho nešlo.

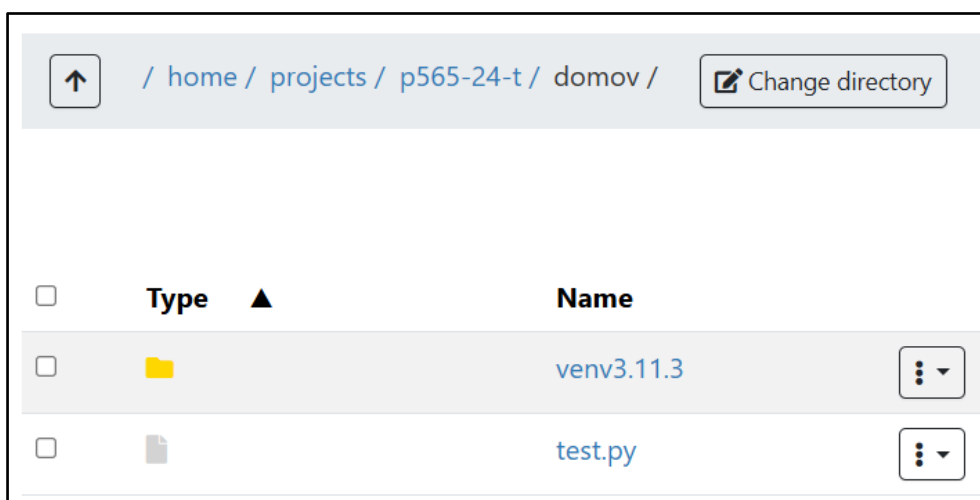
Potom znovu načítajte Python modul a aktivujte prostredie:

```
module load Python/3.11.3-GCCcore-11.3.0
source venv3.11.3/bin/activate
```

```
ddrzik@login02 domov > export MODULEPATH=/storage-apps/Modules:/storage-
s/easybuild/modules/compiler:/storage-apps/easybuild/modules/data:/stora
ybuild/modules/math:/storage-apps/easybuild/modules mpi:/storage-apps/ea
d/modules/toolchain:/storage-apps/easybuild/modules/tools:/storage-apps/
ddrzik@login02 domov >
ddrzik@login02 domov > module load Python/3.11.3-GCCcore-11.3.0
ddrzik@login02 domov > source venv3.11.3/bin/activate
(venv3.11.3) ddrzik@login02 domov >
```

5. Testovací skript

Vytvorte súbor test.py s obsahom, ktorý importuje knižnicu **NumPy** (táto knižnica nie je štandardnou súčasťou Pythonu).



Pri spustení skriptu bez nainštalovaného balíka sa zobrazí chyba o chýbajúcej knižnici.

```
(venv3.11.3) ddrzik@login02 domov > python test.py
Traceback (most recent call last):
  File "/projects/p565-24-t/domov/test.py", line 1, in <module>
    import numpy as np
ModuleNotFoundError: No module named 'numpy'
(venv3.11.3) ddrzik@login02 domov >
```

Problém vyriešite nainštalovaním potrebného balíka priamo vo virtuálnom prostredí:

```
pip install numpy
```

```
(venv3.11.3) ddrzik@login02 domov > pip install numpy
Collecting numpy
  Downloading numpy-2.3.3-cp311-cp311-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (16.9 MB)
    16.9/16.9 MB 22.1 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-2.3.3

[notice] A new release of pip available: 22.3.1 -> 25.2
[notice] To update, run: pip install --upgrade pip
(venv3.11.3) ddrzik@login02 domov >
(venv3.11.3) ddrzik@login02 domov >
(venv3.11.3) ddrzik@login02 domov > █
```

6. Overenie funkčnosti

Po úspešnej inštalácii spustíte skript znova. Tentokrát by sa mal vykonať bez chýb, čo potvrdzuje správne nastavenie prostredia a inštaláciu knižníc.

```
(venv3.11.3) ddrzik@login02 domov > python test.py
Vytvorená matica:
[[3.54520079 3.9274329 8.31022709 7.52963695 2.11960376]
 [8.90587495 7.17149314 9.95145265 2.45949206 2.56083124]
 [3.05490079 6.94626246 9.76885762 7.76211768 6.03791267]
 [2.53807458 0.1117148 2.10778642 2.72697362 3.07299284]
 [7.60225631 5.74014429 8.30641565 0.4025871 6.42979565]]

Stredná hodnota matice: 5.16
Štandardná odchýlka matice: 2.92
(venv3.11.3) ddrzik@login02 domov > █
```

Poznámka autora: Pri ďalšom prihlásení do konzoly, ak už máte vytvorený virtuálny environment aj nainštalované knižnice, stačí zopakovať iba **krok 4** (načítanie modulu a aktivácia prostredia).

Upozornenie: Na spúšťanie hlavných a výpočtovo náročných úloh sa neodporúča používať tento terminál. Prostredie má obmedzenú pamäť a zároveň by sa tým obišli výpočtové zdroje pridelené projektu, čo môže viesť k neefektívnemu využívaniu systému.

5 SPUSTENIE VEĽKÝCH VÝPOČTOVÝCH ÚLOH

Práca na superpočítačoch, ako je Devana, sa zásadne líši od práce na osobnom počítači. Zdroje CPU, GPU a RAM sú zdieľané medzi mnohými používateľmi. Aby bol tento systém spravodlivý a efektívny, všetok prístup k výpočtovým zdrojom riadi špecializovaný softvér nazývaný *scheduler*. Na Devane sa na tento účel používa systém **SLURM** (Simple Linux Utility for Resource Management).

Vašou úlohou ako používateľa nie je priamo spúšťať výpočty na výpočtových uzloch (*compute nodes*), ale zadať požiadavku scheduleru. Táto požiadavka má formu skriptu, ktorý detailne

popisuje, aké zdroje váš výpočet potrebuje a čo presne sa má vykonať. SLURM potom zaradí vašu požiadavku do fronty a keď budú požadované zdroje voľné, automaticky spustí váš skript na príslušných výpočtových uzloch. Tento proces sa nazýva odoslanie úlohy (*job submission*).

5.1 Anatomia spúšťacieho skriptu

Základom každej úlohy je spúšťací skript, zvyčajne písaný v jazyku **Bash**. Tento skript má tri kľúčové časti:

- **Shebang:** Úplne prvý riadok (`#!/bin/bash`), ktorý operačnému systému hovorí, aký interpreter má použiť na spracovanie skriptu.
- **Hlavička s direktívami pre SLURM:** Blok riadkov začínajúcich na `#SBATCH`. Hoci v Bash skripte znak `#` označuje komentár, SLURM tieto špecifické riadky aktívne číta a interpretuje ich ako príkazy a požiadavky na zdroje. Toto je najdôležitejšia časť, kde definujete všetko potrebné pre váš výpočet.
- **Telo skriptu:** Samotné príkazy, ktoré sa majú vykonať, keď SLURM úlohu spustí. To zahŕňa prípravu prostredia (napríklad načítanie modulov, aktivácia virtuálneho prostredia) a samotné spustenie vášho programu (napr. `python train.py`).

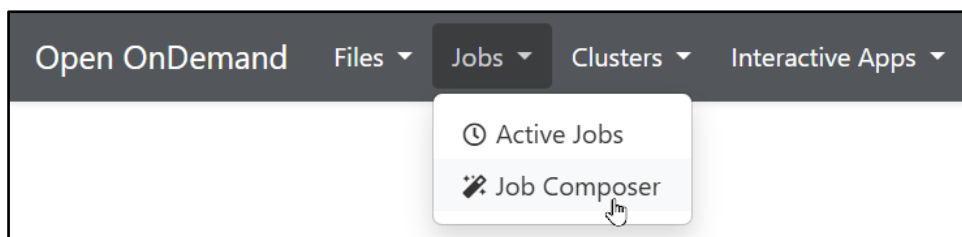
5.2 Vytvorenie a správa úloh cez Open OnDemand Job Composer

Zatiaľ čo priama práca s príkazovým riadkom a manuálna tvorba skriptov ponúka maximálnu flexibilitu, superpočítač Devana poskytuje aj grafické webové rozhranie **Open OnDemand**, ktoré celý proces výrazne zjednodušuje, najmä pre začínajúcich používateľov. Jedným z najužitočnejších nástrojov v tomto prostredí je **Job Composer**.

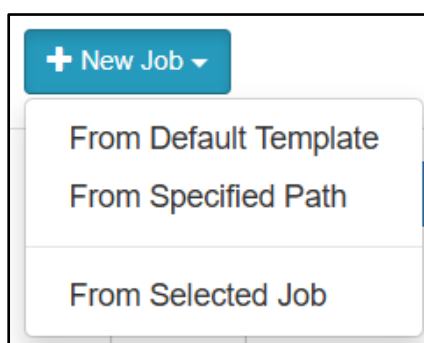
Job Composer umožňuje vytvárať, upravovať, odosielať a monitorovať vaše výpočtové úlohy v prehľadnom grafickom prostredí. Eliminuje potrebu pamätať si presnú syntax príkazov a direktív, pretože vás vedie procesom nastavenia pomocou formulárov.

Postup vytvorenia novej úlohy:

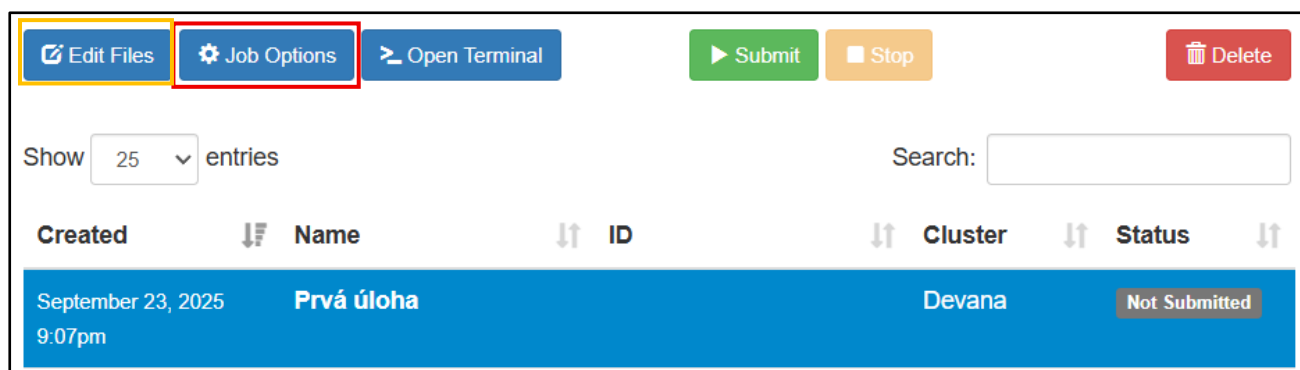
1. Prístup do Job Composeru: Po prihlásení do portálu Open OnDemand na adrese ood.devana.nsc.sk prejdite v hornom menu na položku **Jobs** a zvolte možnosť **Job Composer**.



2. Vytvorenie novej úlohy: V rozhraní Job Composera kliknite na modré tlačidlo **New Job** a následne na možnosť **From Default Template**. Týmto krokom vytvoríte novú pracovnú zložku pre vašu úlohu, ktorá bude obsahovať predpripravený základný spúšťač skriptu a štruktúru.



3. Konfigurácia úlohy: Po vytvorení úlohy sa zobrazí panel s možnosťami. Tu sa odohráva kľúčová časť nastavenia, ktorá priamo zodpovedá hlavičke #SBATCH v skripte.



- **Job Options:** V tejto sekcii definujete základné parametre úlohy:
 - **Name:** Názov úlohy, ktorý slúži pre vašu orientáciu v rámci Job Composera.
 - **Cluster:** Predvolene: Devana.
 - **Job script:** Názov Bash skriptu, ktorý sa má spustiť. Predvolená hodnota je zvyčajne main_job.sh.
 - **Account:** Nepovinný parameter, ktorý zodpovedá direktíve #SBATCH --account. Ak máte prístup k viacerým projektom, tu si môžete zvoliť, na ktorý z nich sa má účtovať spotrebovaný výpočtový čas. Ak ho nevyplníte, použije sa váš predvolený projekt.

- **Úprava spúšťacieho skriptu:** Kliknutím na tlačidlo **Edit Files** sa dostanete do jednoduchého textového editora priamo vo webovom prehliadači. Tu upravujete **telo skriptu**. Vložíte sem príkazy na prípravu prostredia a samotné spustenie vášho programu, ako sme si ukázali v predchádzajúcej kapitole.

5.3 Praktický príklad: skript pre výpočtovú úlohu

Podme sa pozrieť na kompletný, konkrétny skript, ktorý sme pripravili. Vložíte ho do textového editora v Job Composeri (do súboru `main_job.sh`):

```
#!/bin/bash
#SBATCH --partition=gpu
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=64
#SBATCH --gres=gpu:1
#SBATCH --mem=32000
#SBATCH --output stdout.%J.out
#SBATCH --error stderr.%J.out

export MODULEPATH=/storage-apps/Modules:/storage-apps/intel-
v.2023/oneapi/modulefiles:/storage-apps/easybuild/modules/bio:/storage-
apps/easybuild/modules/cae:/storage-apps/easybuild/modules/chem:/storage-
apps/easybuild/modules/compiler:/storage-apps/easybuild/modules/data:/storage-
apps/easybuild/modules/devel:/storage-apps/easybuild/modules/geo:/storage-
apps/easybuild/modules/lang:/storage-apps/easybuild/modules/lib:/storage-
apps/easybuild/modules/math:/storage-apps/easybuild/modules/mpi:/storage-
apps/easybuild/modules/numlib:/storage-apps/easybuild/modules/perf:/storage-
apps/easybuild/modules/phys:/storage-apps/easybuild/modules/system:/storage-
apps/easybuild/modules/toolchain:/storage-apps/easybuild/modules/tools:/storage-
apps/easybuild/modules/vis

module load Python/3.11.3-GCCcore-11.3.0

source /home/projects/p565-24-t/domov/venv3.11.3/bin/activate

python /home/projects/p565-24-t/domov/uloha1/main.py
```

Týmto skriptom ste scheduleru SLURM zadali veľmi presnú a komplexnú požiadavku. Rozoberme si ju krok za krokom.

Požiadavka na zdroje (hlavička #SBATCH):

- **--partition=gpu** a **--gres=gpu:1**: Explicitne ste si vyžiadali výpočtový uzol, ktorý je vybavený grafickou kartou (GPU), a pre svoju úlohu požadujete prístup presne k jednej z nich.
- **--nodes=1** a **--ntasks-per-node=1**: Váš výpočet bude prebiehať koncentrovane na jednom uzle ako jeden hlavný proces.
- **--cpus-per-task=64**: Toto je významná požiadavka. Pridelili ste svojej úlohe 64 procesorových jadier. To je ideálne pre situácie, kde váš kód (napríklad knižnice na spracovanie dát v Pythone) dokáže efektívne využiť viacero jadier na paralelné operácie, ako je príprava a načítavanie dátových dávok (data batchov) pre GPU.
- **--mem=32000**: Požadujete 32 000 MB, teda približne 32 GB systémovej pamäte RAM. Táto pamäť bude slúžiť na uloženie dátovej sady, operačného systému a všetkých procesov, ktoré bežia na CPU.
- **--output** a **--error**: Zabezpečili ste si správne zaznamenávanie výstupov. Všetky štandardné výpisy z vášho skriptu (napr. z príkazu print) sa uložia do súboru stdout.[ID_úlohy].out a všetky chybové hlášky do stderr.[ID_úlohy].out.

Poznámka autora: Toto je iba jedno z mnohých nastavení. Podrobnejší návod nájdete priamo v dokumentácii na [tomto odkaze](#).

Príprava prostredia (telo skriptu):

- **export MODULEPATH=...**: Nastavili ste systémovú premennú, aby softvérové moduly hľadala v správnych adresároch špecifických pre Devana.
- **module load Python/3.11.3-GCCcore-11.3.0**: Sprístupnili ste si konkrétnu verziu Pythonu (3.11.3).
- **source .../activate**: Aktivovali ste si svoje vlastné virtuálne prostredie. Tým ste zaistili, že váš skript bude používať presne tie verzie knižníc (napr. PyTorch, TensorFlow, atď.), ktoré ste si do tohto prostredia nainštalovali, a nedôjde ku konfliktom s inými verziami na systéme.

Spustenie aplikácie:

- **python .../main.py:** Na záver spúšťate samotný Python skript (zobrazený nižšie). Tento skript sa teraz vykoná v precízne pripravenom prostredí a s presne alokovanými hardvérovými zdrojmi (1 GPU, 64 CPU jadier, 32 GB RAM).

```
import time

filename = "/home/projects/p565-24-t/domov/uloha1/data.txt"

with open(filename, "a", encoding="utf-8") as f:
    f.write("=== Začiatok logu ===\n")

for i in range(1,21):
    with open(filename, "a", encoding="utf-8") as f:
        f.write(f"{i}. riadok bol pridaný.\n")

time.sleep(1)
```

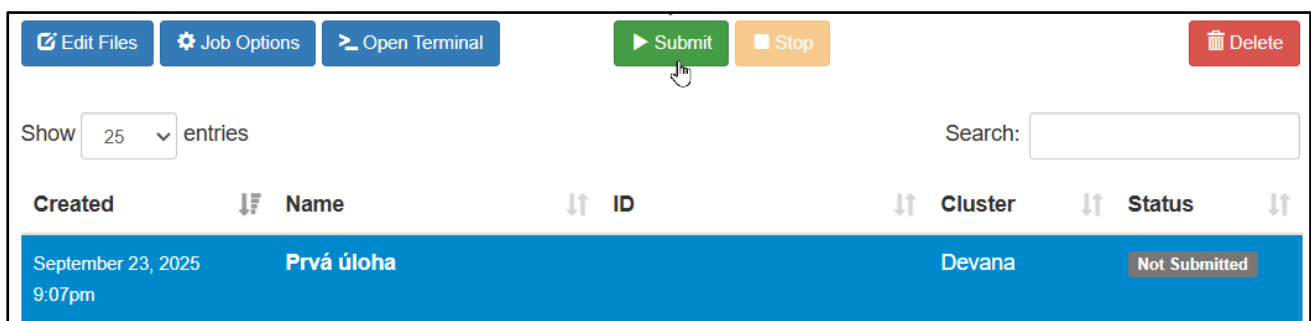
Aj v scripture je nevyhnutné zadať celú cestu k zdrojom.

5.4 Spustenie a monitorovanie úlohy

Keď máte finálny skript pripravený a uložený v textovom editore Job Composer, nasledujú tieto posledné kroky:

1. Uloženie a odoslanie:

- Najprv sa uistite, že ste zmeny v skripte uložili (zvyčajne tlačidlom Save v editore).
- Vráťte sa na hlavnú obrazovku vašej úlohy v Job Composer.
- Kliknite na veľké zelené tlačidlo Submit. Týmto krokom odošlete vašu požiadavku, zabalenú v tomto skripte, priamo scheduleru SLURM.



The screenshot shows the Job Composer interface. At the top, there are buttons for 'Edit Files', 'Job Options', 'Open Terminal', 'Submit' (green), 'Stop' (orange), and 'Delete' (red). Below these buttons, there is a 'Show' dropdown set to '25' and a 'Search' input field. A table lists jobs with columns: Created, Name, ID, Cluster, and Status. The first job is 'Prvá úloha' created on 'September 23, 2025 9:07pm' in the 'Devana' cluster, with a status of 'Not Submitted'.

Created	Name	ID	Cluster	Status
September 23, 2025 9:07pm	Prvá úloha		Devana	Not Submitted

2. Čo sa stane po odoslaní?

- Vaša úloha sa okamžite zaradí do fronty. Jej stav sa zmení na Queued (čaká v rade) alebo Pending (čaká na zdroje).

- SLURM teraz hľadá voľný výpočtový uzol, ktorý spĺňa vaše náročné požiadavky: musí byť v gpu partícii, mať voľnú aspoň jednu GPU, 64 CPU jadier a 32 GB RAM.

[Edit Files](#)
[Job Options](#)
[Open Terminal](#)
[Submit](#)
[Stop](#)
[Delete](#)

Show entries
Search:

Created	Name	ID	Cluster	Status
September 23, 2025 9:07pm	Prvá úloha	943013	Devana	Queued

- Akonáhle sa takýto uzol uvoľní, SLURM vašu úlohu prideli na tento uzol a jej stav sa zmení na Running (beží). V tomto momente sa začne vykonávať váš skript, riadok po riadku.

[Edit Files](#)
[Job Options](#)
[Open Terminal](#)
[Submit](#)
[Stop](#)
[Delete](#)

Show entries
Search:

Created	Name	ID	Cluster	Status
September 23, 2025 9:07pm	Prvá úloha	943014	Devana	Running

3. Sledovanie priebehu a výsledkov:

- Stav úlohy môžete priebežne sledovať priamo v rozhraní Job Composera alebo v sekcii **Jobs -> Active Jobs**.
- Po úspešnom (alebo neúspešnom) dokončení úlohy sa jej stav zmení na **Completed**.

[Edit Files](#)
[Job Options](#)
[Open Terminal](#)
[Submit](#)
[Stop](#)
[Delete](#)

Show entries
Search:

Created	Name	ID	Cluster	Status
September 23, 2025 9:07pm	Prvá úloha	943017	Devana	Completed

- V priečinku vašej úlohy v Job Composer (môžete sa do neho prekliknúť) nájdete vygenerované súbory stdout.%J.out a stderr.%J.out. Otvorením týchto súborov zistíte, čo váš program vypisoval počas behu a či sa nevyskytli nejaké chyby. Toto je váš hlavný zdroj informácií o tom, ako výpočet prebehol.

6 PRENOS VEĽKÝCH DÁTOVÝCH SÚBOROV

Ako už bolo uvedené v tomto návode, prenos súborov je možný aj prostredníctvom webového rozhrania **Open OnDemand**. Toto rozhranie však umožňuje nahrávať len súbory s obmedzenou veľkosťou (približne niekoľko stoviek MB). Sťahovanie cez toto prostredie je takisto limitované, pričom obmedzenia sú dané priamo vlastnosťami webového prehliadača.

Pri pokročilých výpočtoch je bežné, že datasety dosahujú desiatky až stovky GB, čo už nie je možné prenášať cez Open OnDemand. V takom prípade je potrebné použiť externé nástroje na prenos dát, ako napríklad **FileZilla** alebo **WinSCP** (voľba klienta závisí od používaného operačného systému).

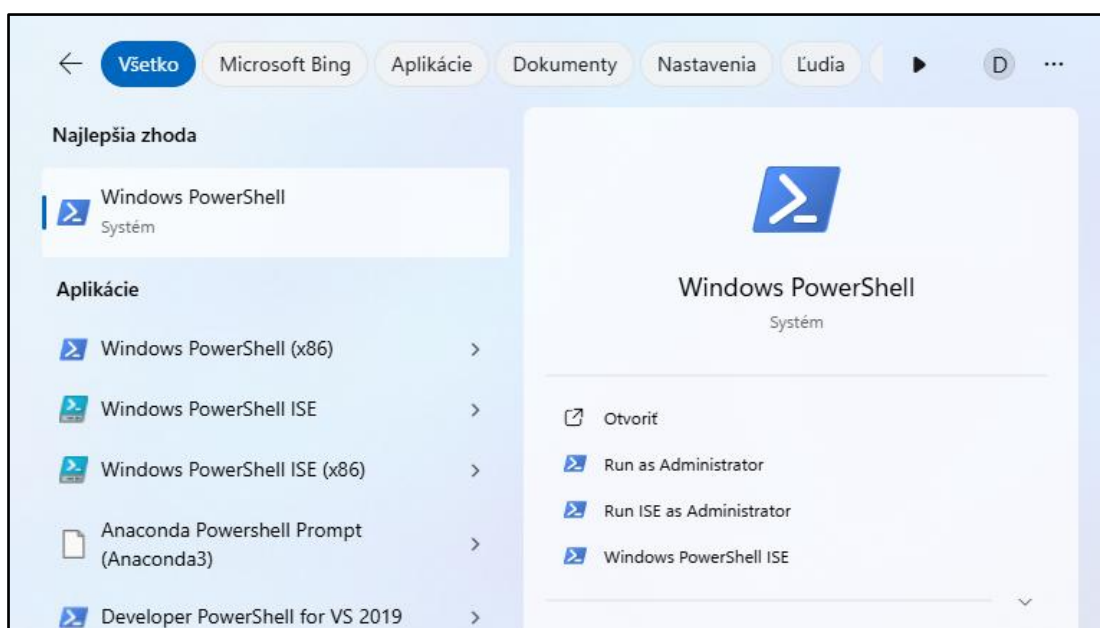
V tejto časti sa zameriame na proces nastavenia a prenesenia dátového súboru v operačnom systéme Windows.

6.1 Generovanie SSH kľúčov

Do systému Devana nie je možné sa prihlásiť pomocou mena a hesla, ako je to bežné pri iných systémoch. Prístup je možný výlučne prostredníctvom autorizácie pomocou dvojice **súkromný/verejný kľúč (SSH key pair)**.

Prvým krokom je vytvorenie tejto dvojice kľúčov. Ak používate operačný systém **Windows 10 alebo novší**, postup je veľmi jednoduchý:

1. Otvorte konzolové rozhranie **PowerShell**.



2. Zadajte nasledujúci príkaz:

```
PS C:\Users\VasUcet> ssh-keygen -t ed25519 -o -a 100
```

3. Systém následne spustí dialóg na vygenerovanie príslušnej dvojice kľúčov. Pre naše účely **nie je potrebné zadávať žiadne dodatočné údaje** – jednoducho vždy stlačte **Enter**. Dialóg medzi vami a generátorom SSH kľúčov môže vyzeráť približne takto:

```
PS C:\Users\VasUcet > ssh-keygen -t ed25519 -o -a 100
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\VasUcet/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\VasUcet/.ssh/id_ed25519
Your public key has been saved in C:\Users\VasUcet/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:7cqtDGNilcFXpERgac+rrix1sl8eAM88GfUZeQ/Trnc vas-ucet@DESKTOP-LO9BOPL
The key's randomart image is:
+--[ED25519 256]--+
|      o+++.+      |
|      ooo = o      |
|      ..+o+ +      |
|      = *oo .      |
|      X S.. .      |
|      o..o.. o      |
|      .o+++.o + .   |
|      .o.o.B + o. E  |
|      .o+o *.oo..   |
+-----[SHA256]-----+
```

4. Po dokončení generovania sa vám vytvoria dva súbory v adresári **C:\Users\VasUcet\.ssh**:

- **id_ed25519** – *súkromný kľúč* (tento súbor si starostlivo chráňte a **nikdy ho nikomu neposielajte**),
- **id_ed25519.pub** – *verejný kľúč*, ktorý sa používa na autentifikáciu voči serveru Devana.

Viac informácií o práci s SSH kľúčmi nájdete v [oficiálnej dokumentácii](#).

6.2 Pridelenie verejného kľúču k vášmu kontu

Po úspešnom vygenerovaní dvojice kľúčov je potrebné **nahrať verejný kľúč** do systému Devana prostredníctvom webového rozhrania. Po prihlásení do používateľského portálu prejdite na príslušnú sekciu dostupnú na [tejto adrese](#). Pridanie kľúča je možné aj priamo z úvodnej obrazovky po prihlásení, kliknutím na odkaz „**Pridať nový kľúč**“ v projektovej časti.

Projektové čísla

Akronym	Pridelená alokácia CPU (core*h)	Pridelená alokácia GPU (gpu*h)	Spotrebovaná alokácia CPU (core*h)	Spotrebovaná alokácia GPU (gpu*h)

SSH KLÚČE

Pridať nový kľúč

SSH kľúč	Dátum	Aktívny	Vymazať
id_rsa.pub (577 bajtov)	Ut, 23. 09. 2025 - 20:03	✓	delete
id_rsa.pub (575 bajtov)	Ut, 03. 12. 2024 - 19:27	✓	delete

TIKETY

Oba spôsoby vás presmerujú na formulár, v ktorom je potrebné nahrať súbor s verejným kľúčom.

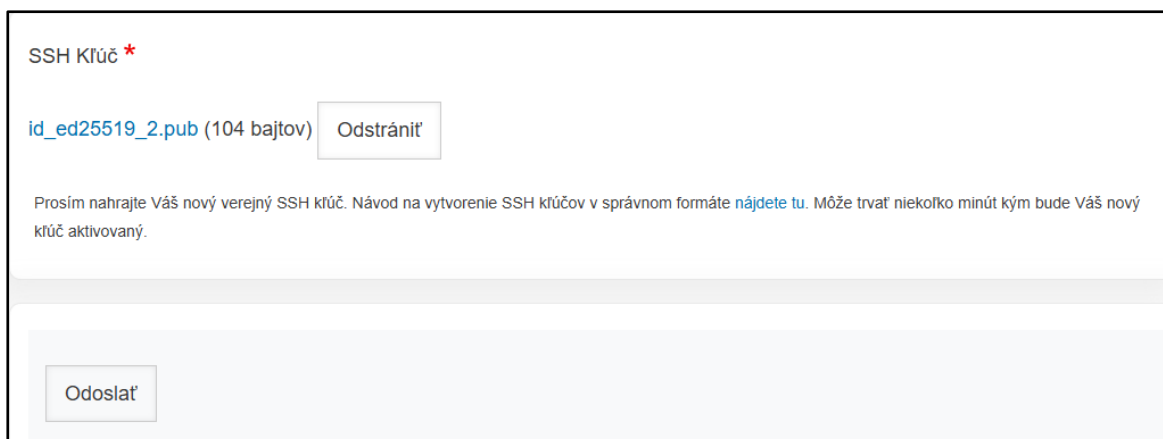
SSH Kľúč *

Vybrať súbor
Nie je vybratý žiadny súbor

Prosím nahrajte Váš nový verejný SSH kľúč. Návod na vytvorenie SSH kľúčov v správnom formáte [nájdete tu](#). Môže trvať niekoľko minút kým bude Váš nový kľúč aktivovaný. | Iba jeden súbor. | 1 KB limit. | Povolené typy: txt pub.

Odoslať

V našom prípade sa nachádza v priečinku **C:\Users\VasUcet\.ssh**, pričom treba vybrať súbor **id_ed25519.pub**. Ak ste si kľúč pri generovaní pomenovali inak, vyberte súbor s príponou .pub. Akonáhle systém súbor úspešne **overí** a **načíta ho do dočasnej pamäte**, kliknite na tlačidlo **Odoslať**.



V prostredí HPC Devana môžete mať nahratých **viacero verejných kľúčov**. Táto možnosť vám umožňuje pripájať sa k systému z rôznych zariadení alebo subsystémov (napríklad z prostredia **WSL** vo Windows).

Ak nahratie prebehne úspešne, váš verejný kľúč sa zobrazí v **zozname verejných kľúčov** priradených k vášmu účtu na [tejto adrese](#).

6.3 Prepojenie SFTP klienta s vaším súkromným kľúčom

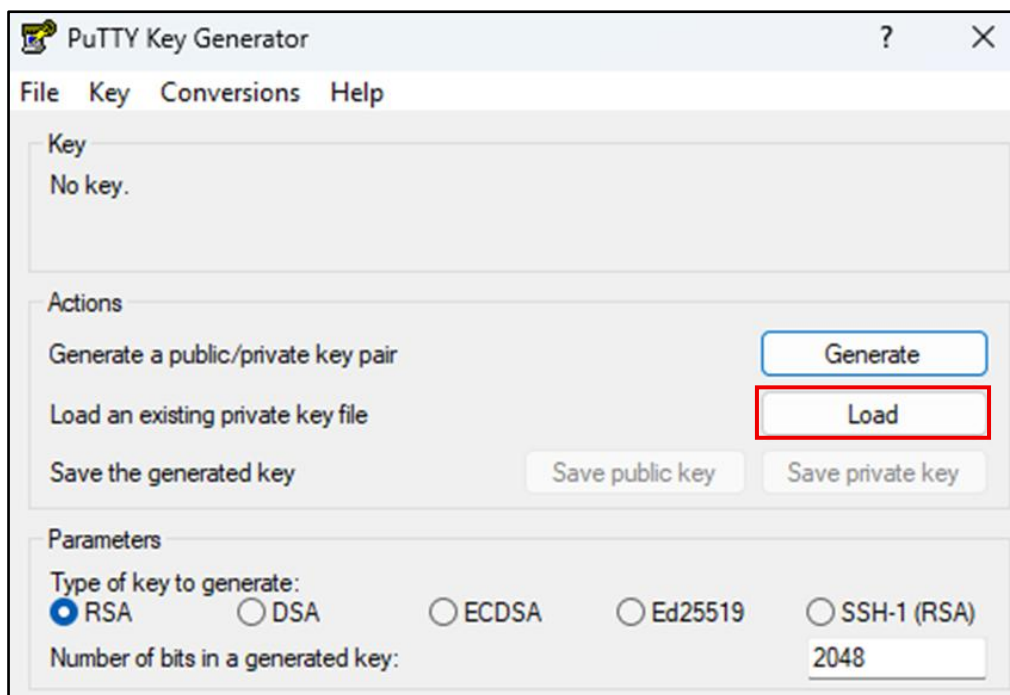
V tomto bode už Devana pozná váš verejný kľúč. Pri nadväzovaní spojenia je však potrebné, aby váš SFTP klient (napr. **FileZilla**, **WinSCP** a pod.) vedel použiť váš **súkromný kľúč**, pomocou ktorého overí vašu totožnosť.

V tomto návode používame SFTP klient **FileZilla**, ktorý je bezplatne dostupný pre všetky hlavné operačné systémy (Windows, macOS, Linux distribúcie).

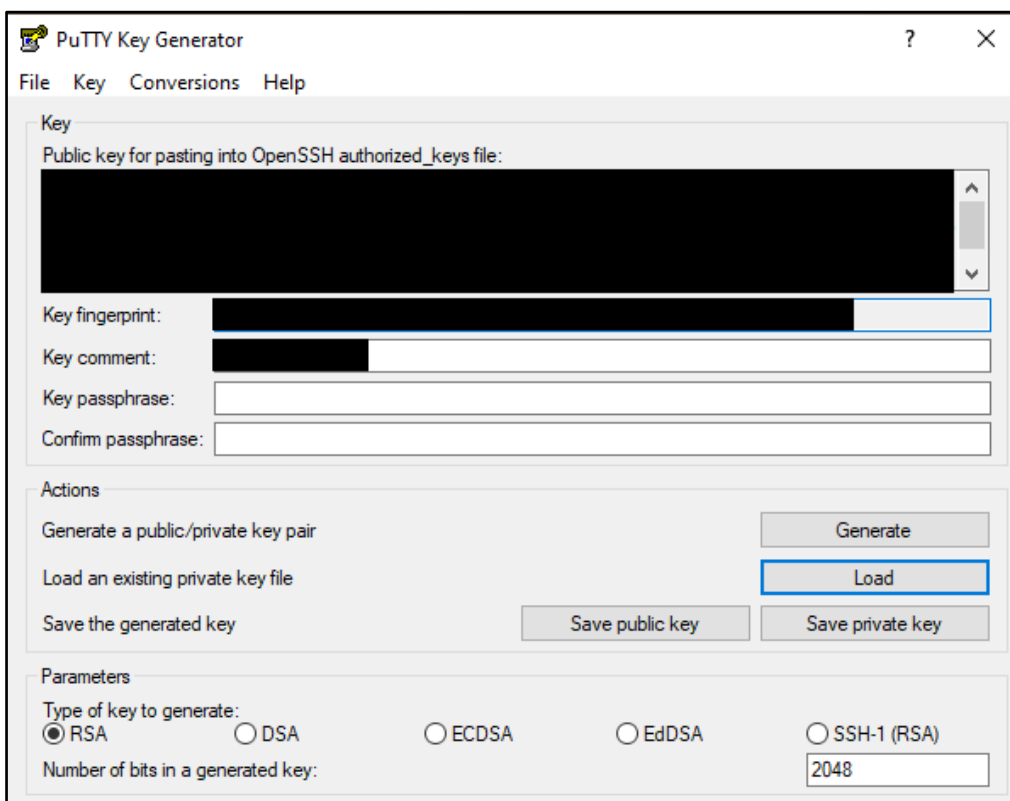
Problém, na ktorý môžete naraziť, spočíva v tom, že váš súkromný a verejný kľúč sú vygenerované vo formáte, ktorý niektorí klienti pre Windows nepodporujú (OpenSSH vs PPK). Niektoré aplikácie, ako napríklad **WinSCP**, dokážu pri detekcii týchto kľúčov konverziu na správny formát vykonať automaticky, iné – ako **FileZilla** – nie.

V tomto návode sme sa rozhodli ísť zložitejšou cestou a použiť nástroj, ktorý túto konverziu nevykonáva automaticky. Preto je potrebné stiahnuť si program **PuTTYgen** z [tohto odkazu](#).

Po úspešnej inštalácii a spustení sa zobrazí nasledovná aplikácia:



Následne je potrebné načítať váš súkromný kľúč pomocou tlačidla **Load**. Po úspešnom načítaní sa kľúč zobrazí v sekcii **Key**.



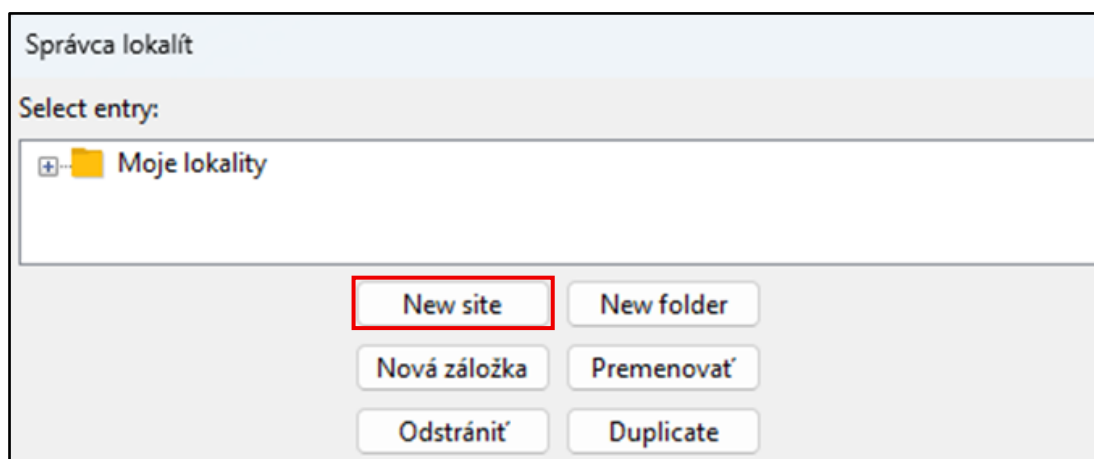
Akonáhle ste si istí, že ste zvolili správny súkromný kľúč, ktorý tvorí dvojicu s verejným kľúčom priradeným k vášmu kontu, kliknite na tlačidlo **Save private key**.

Po úspešnom uložení by sa v priečinku C:\Users\VasUcet\.ssh mali nachádzať tri súbory:

- **id_ed25519.pub**
- **id_ed25519**
- **id_ed25519.ppk**

Teraz môžete otvoriť vášho SFTP klienta.

Prvým krokom je vytvoriť si novú lokalitu kliknutím na tlačidlo **New Site**.



Následne vyplňte prihlasovacie údaje nasledovne:

Protokol: **SFTP**

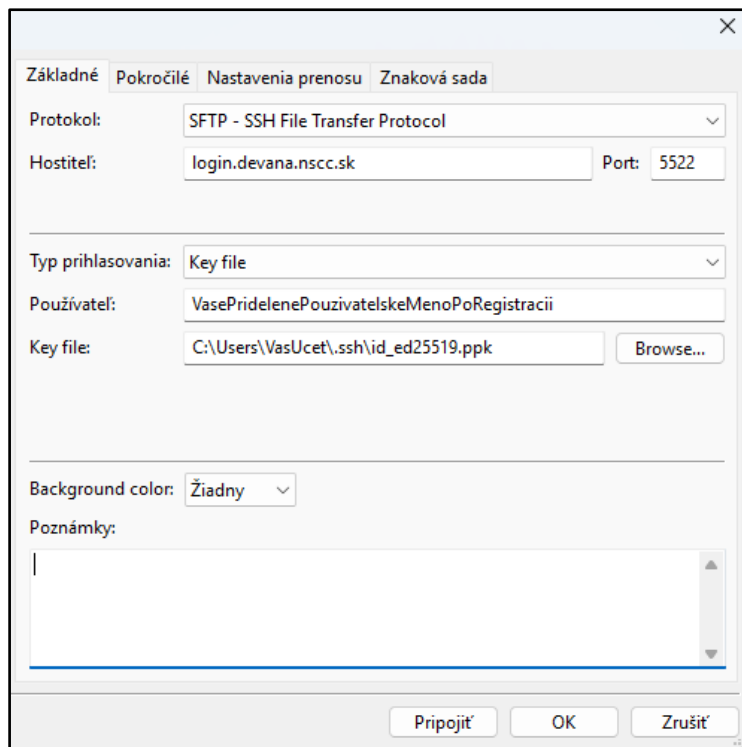
Hostiteľ: **login.devana.nsc.sk**

Port: **5522**

Typ prihlasovania: **Key file**

Používateľ: **Prihlasovacie meno ktoré vám bolo vygenerované po úspešnom schválení projektu.**

Key file: **Cesta k vášmu privátnemu kľúču**



Po kliknutí na tlačidlo Pripojiť dôjde k úspešnému nadviazaniu spojenia so systémom Devana. Toto môžete vidieť v konzolovom výpise v hornej časti klienta.

```
Stav: Prebieha pripájanie k login.devana.nsc.sk:5522...
Stav: Using username "VasePridelenePouzivatelskeMenoPoRegistracii".
Stav: Connected to login.devana.nsc.sk
Stav: Prijíma sa výpis adresára...
Stav: Listing directory /home/VasePridelenePouzivatelskeMenoPoRegistracii
Stav: Directory listing of "/home/VasePridelenePouzivatelskeMenoPoRegistracii" successful
```

Akonáhle ste nadviazali spojenie s uvedeným serverom, môžete bez problémov presúvať súbory **v rámci veľkosti alokovaného diskového priestoru** pre váš projekt do príslušnej **projektovej zložky**, ako to už bolo popísané v tomto návode v časti o rozhraní **Open OnDemand**.

Viac informácií nájdete v [oficiálnej dokumentácii](#).

7 ZÁVER

Gratulujem! Úspešne ste sa oboznámili so základnými krokmi pre prácu so superpočítačom Devana. Od registrácie a vytvorenia testovacieho projektu, cez prípravu prostredia v Open OnDemand až po spustenie a monitorovanie vlastnej výpočtovej úlohy prostredníctvom Job Composeru.

S týmito znalosťami máte pevný základ pre efektívne využívanie vysokovýkonných výpočtových (HPC) zdrojov, ktoré Devana ponúka na realizáciu vašich vedeckých a výskumných projektov. Nezabúdajte, že práca s HPC je neustály proces učenia sa a experimentovania.

7.1 Ďalšie kroky a podpora

Tento manuál slúžil ako úvodný sprievodca. Pre detailnejšie informácie, pokročilé nastavenia (napríklad zložitejšie SLURM direktívy) a riešenie špecifických problémov dôrazne odporúčame nahliadnuť do [oficiálnej používateľskej dokumentácie](#), ktorá je priebežne aktualizovaná tímom Výpočtového centra SAV.

V prípade technických problémov alebo otázok, na ktoré nenájdete odpoveď v dokumentácii, neváhajte kontaktovať administrátorov prostredníctvom [systému tiketov](#) dostupného vo vašom používateľskom portáli.

7.2 Zdroje

Informácie v tomto dokumente boli čerpané a inšpirované predovšetkým z oficiálnych zdrojov a portálov superpočítača Devana, ako aj z praktických skúseností autora. Hlavné referenčné body zahŕňajú:

- **Oficiálna používateľská dokumentácia Devana:** Komplexný a priebežne aktualizovaný zdroj všetkých technických informácií, návodov a osvedčených postupov.
- **Používateľský portál NSCC:** Rozhranie pre správu účtu, projektov, tiketov a SSH kľúčov.
- **Webové rozhranie Open OnDemand:** Primárny nástroj pre prístup k súborom, interaktívnym aplikáciám a správe výpočtových úloh.