Pradžia

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

**DIDŽIŲJŲ DUOMENŲ APDOROJIMAS**

**Namų darbas 1 ataskaita**

Darbą atliko: DGTfm-21 stud.

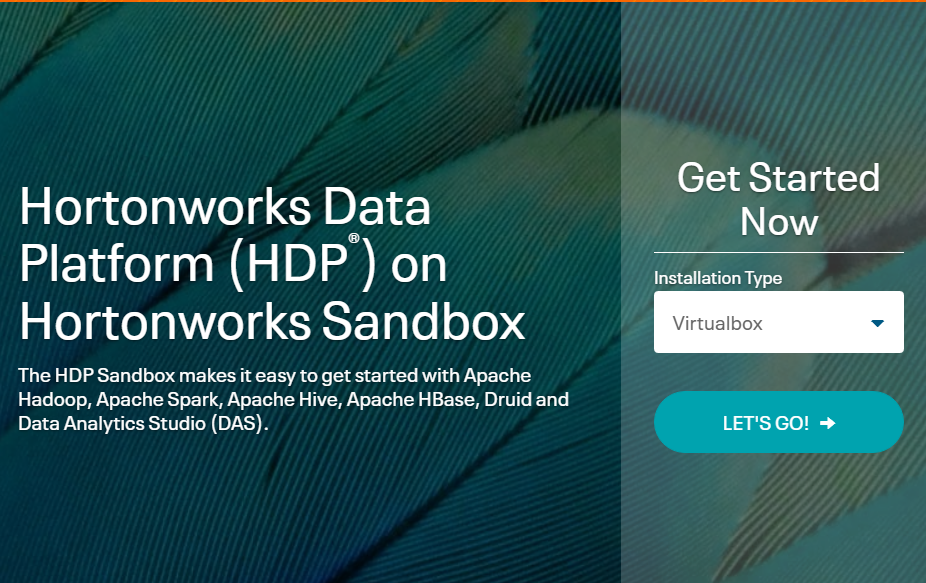
Laimonas Janutėnas

Priėmė: Doc. Dr. Andrej Bugajev

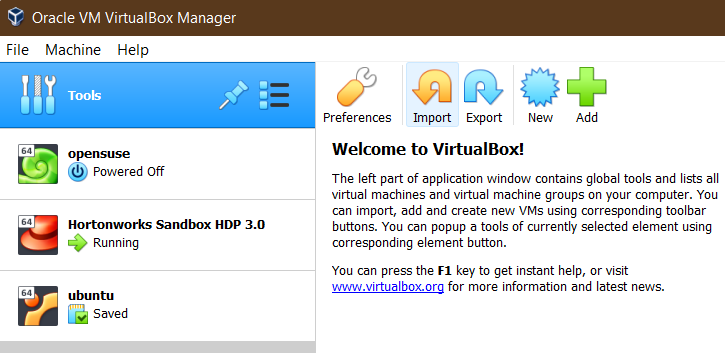
Vilnius 2022

# Aplinkos pasiruošimas

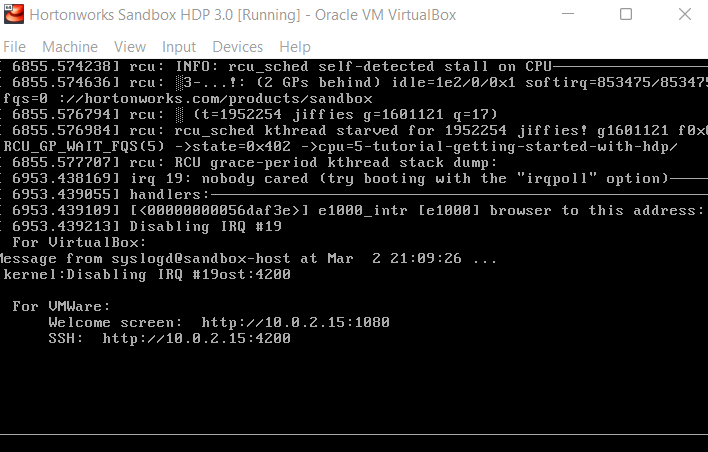
Prieš pradedant darbą reikia įsirašyti Hortonworks HDP programinę įrangą, kurią reikia pasileisti per VirtualBox platformą. Hortonworks HDP galima atsisiųsti per šią nuorodą - [Hortonworks Data Platform (HDP) on Sandbox (cloudera.com)](https://www.cloudera.com/downloads/hortonworks-sandbox/hdp.html)



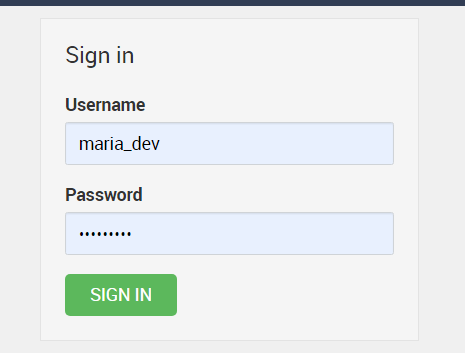
Atsisiųsta programinė įranga ateina su visa operacine sistema ir yra sukonfigūruota veikti iš karto per VirtualBox, kuris sukuria virtualią mašiną.



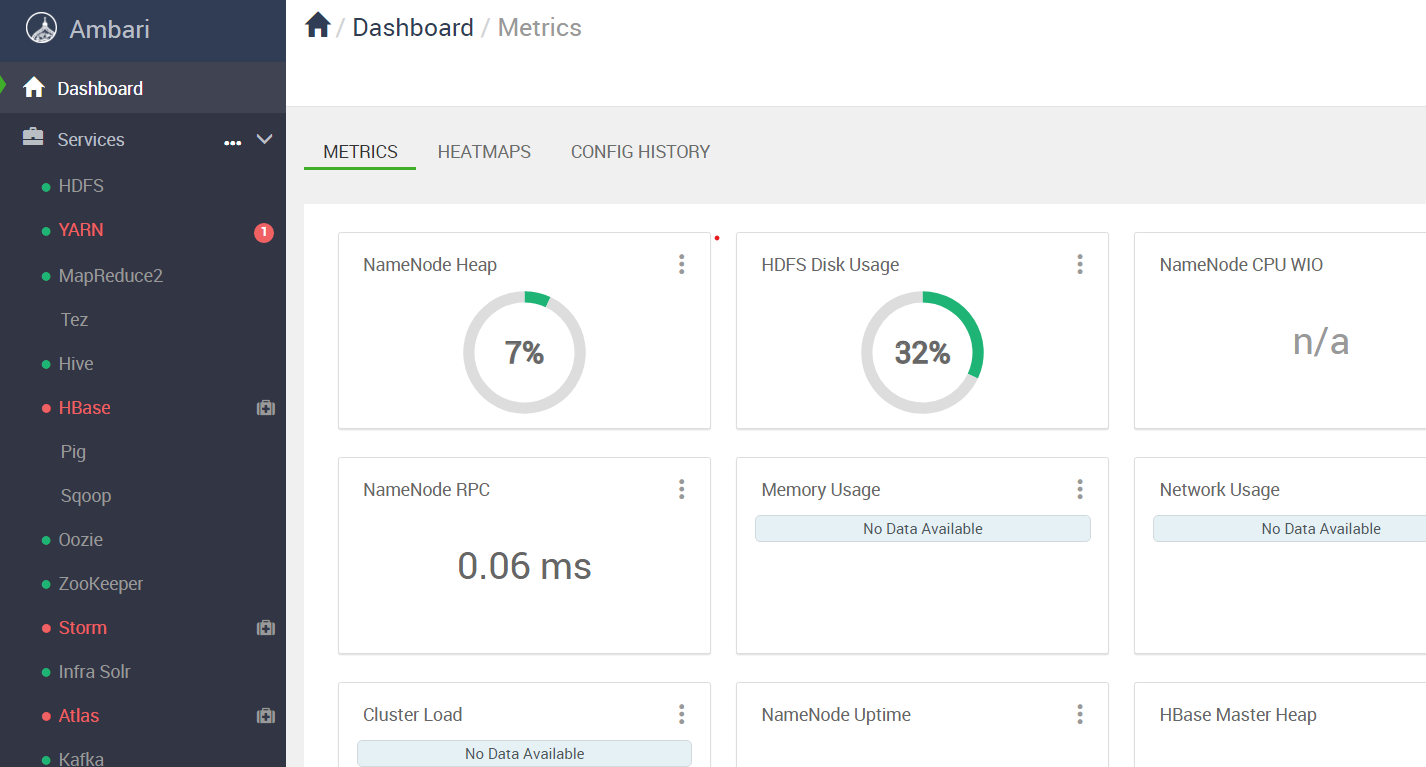
Importavus virtualios mašinos failą, programa pasileidžia. Hortonworks HPD tada galima pasiekti per naršyklę su nuoroda: <https://localhost:8080/>.



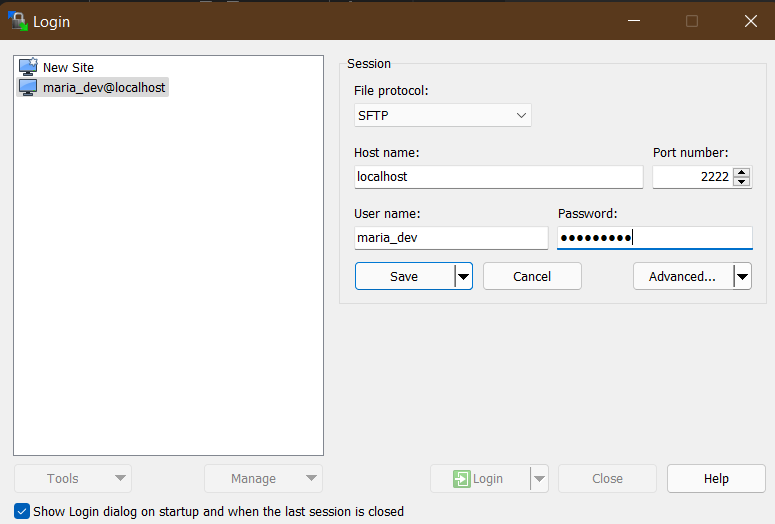
Atsidarius nuorodą reikia prisijungti prie sistemos nu naudotojo vardu ir slaptažodžiu “maria\_dev”.



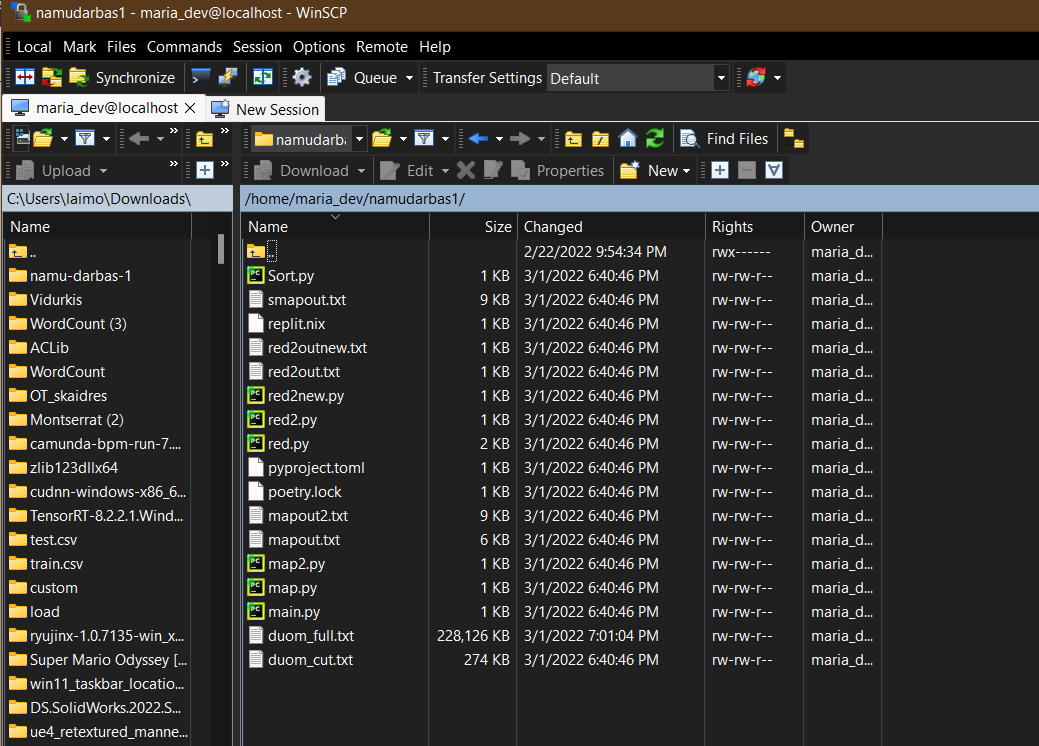
Prisijungus atsidaro Hortonworks HDP valdymo langas, kuriame reikia įjungti norimus servisus.



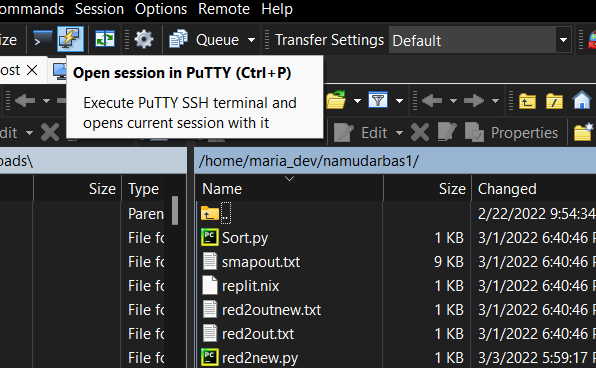
Taip pat reikia įsirašyti WinScp programinę įrangą, kuri per FTP protokolą, tai leis perkelti norimus failus ir duomenis į Hortonworks HDP. Tai galima padaryti prisijungus adresu localhost:2222 ir su prisijungimu bei slaptažodžiu “maria\_dev”.



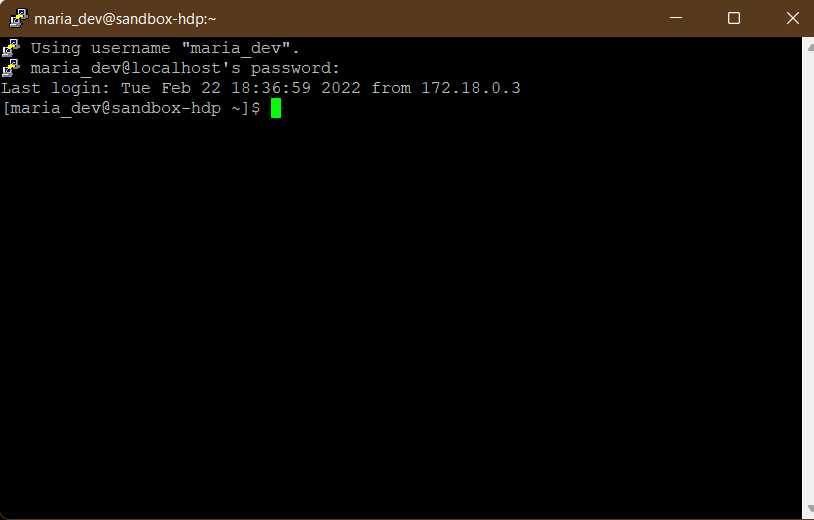
Atsidarius WinScp į sistemą reikia perkelti visus duomenis ir programinius failus į “maria\_dev” direktoriją.



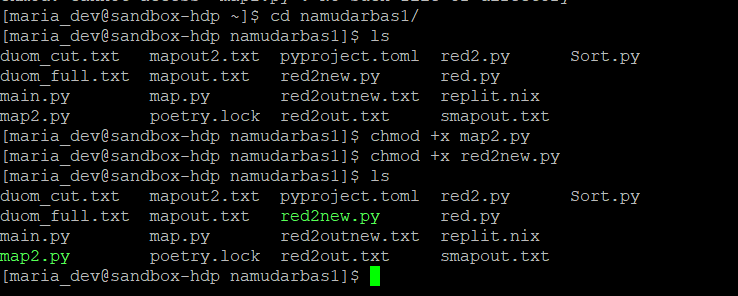
Kai failai jau sukelti reikia atsidaryti PuTTY, kuris leidžia per SSH protokolą pasiekti sistemą ir naudotis sistema bei joje esančia Linux operacine sistema.



Atsidarius PuTTY terminalo langą taip pat reikia suvesti “maria\_dev” slaptažodį.



Prisijungus prie sistemos, pirmiausia reikia nueiti į direktoriją kur yra sukelti įkopijuoti failai, tai galima padaryti su “cd” komanda ir nurodant direktorijos kelią, pvz.: “cd namudarbas”. Tada su komanda “ls” galima pamatyti visus direktorijoje esančius failus. Tam kad failus būtų galima paleisti aplinkoje reikia jiems suteikti paleidimo teises, tai galima padaryti naudojantis “chmod +x” ir nurodant failo pavadinima. Tada dar kartą paleidus “ls” galima pamatyti, kad tie failai nuspalvinami žaliai, tai reiškia juos galima paleisti

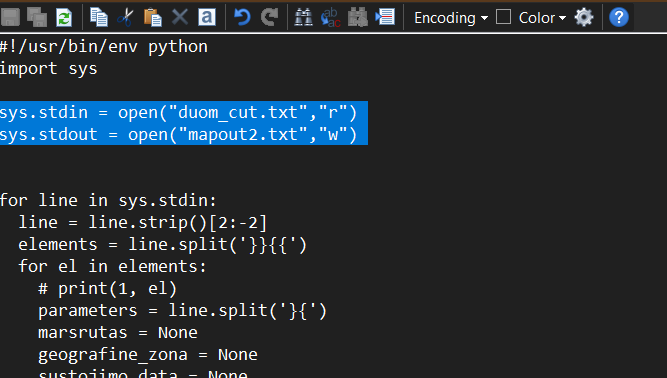


Tada failus duomenų failus reikia perkelti į HDFS duomenų bazę. Pirmiausia yra sukuriama direktorija kur tie failai bus saugomi su komanda “mkdir”, o kad tai būtų įvykdoma HDFS aplinkoje reikia tai nurodyti su “hadoop fs” komanda pvz.: “hadoop fs -mkdir namudarbas”. O duomenys yra perkeliami su “hadoop fs -put duomenys direktorija/failas”

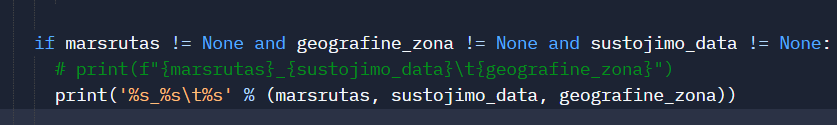


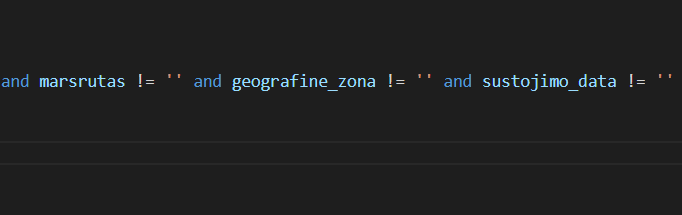


Prieš paleidžiant programinį MapReduce kodą reikia įsitikinti ar jame nėra atidarinėjami failai, nes tai išmes klaidos pranešimą Hadoop sistemoje.



Taip pat reikia įsitikinti, kad programinis kodas veikia teisingai ir neturi jokio pašalinio poveikio, kaip tuščios eilutes ar tušti tekstiniai žodžiai.





Paleisti programinį koda Hadoop sistemoje galima naudojantis “mapred streaming”,

nurodant įvesties failą, išvesties vietą, mapper programinio kodo failą, reducer programinio kodo failą bei juod paduodant su file komanda. Pilnos komandos pavyzdys:

“

mapred streaming \

-input namudarbas1/duom\_full.txt \

-output namudarbas1/out40 \

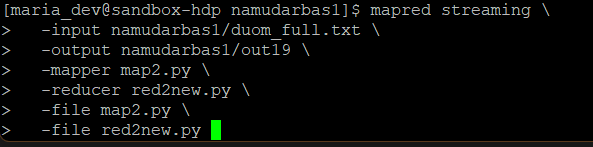
-mapper map2.py \

-reducer red2new.py \

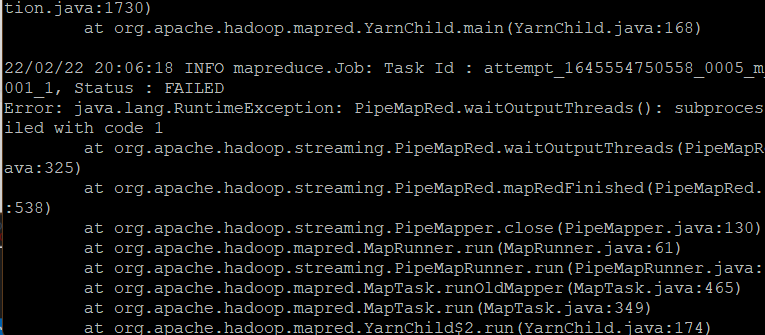
-file map2.py \

-file red2new.py

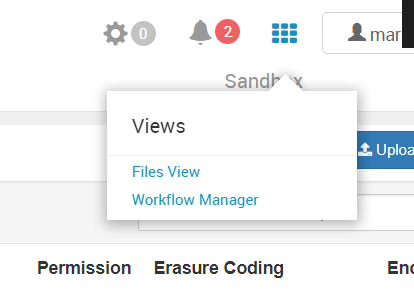
”



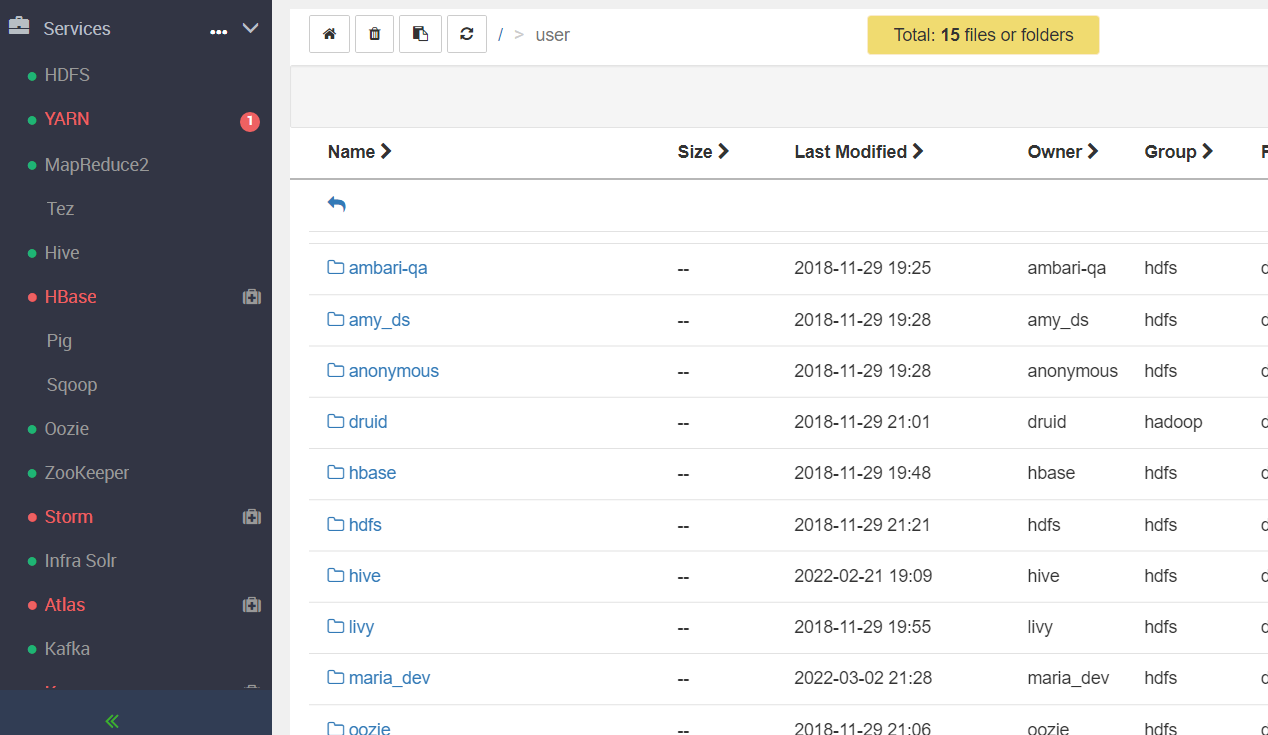
Jei kažkur yra klaida, Hadoop išmes klaidos pranešimą “waitOutputThreads()” ir neišmes konkrečios klaidos kur yra problema, todėl programinį kodą reiktų gerai ištestuoti ne Hadoop aplinkoje.



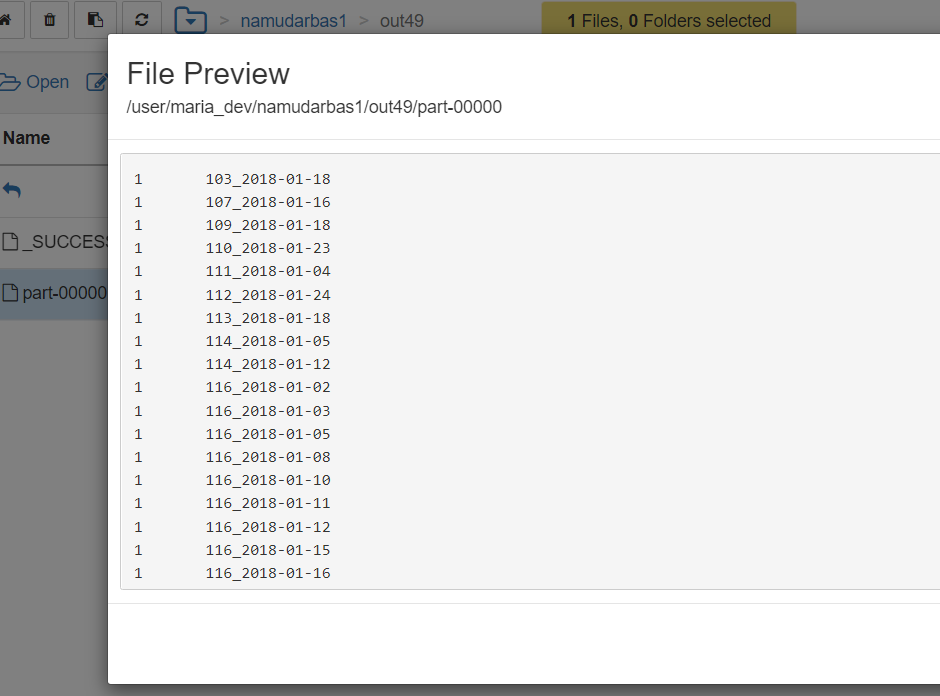
Jei komanda sėkmingai suveikė per Hortonworks HDP sistemą reikia susirasti išvesties failą. Failų sistemą galima pasiekti per “Files View” nuorodą.



Atsidarius failų sistemos langui reikia nueiti į “user/maria\_dev” direktoriją. Joje bus sukurtas nurodyta išvesties direktorija.

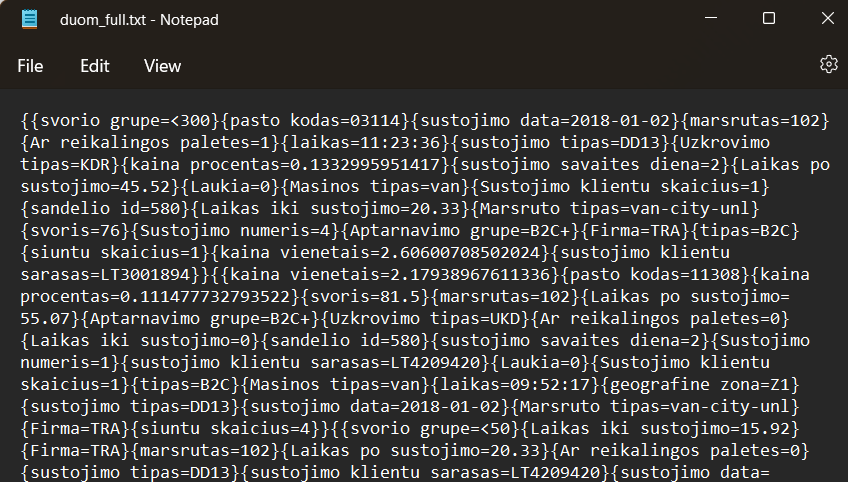


Direktorijos viduje bus failas su išvesties duomenimis, kurį galima peržiūrėti.



# Namų darbo užduotis

Namų darbe reikia išnagrinėti duomenis, aprašančius siuntų pristatymo bei logistikos įmonės veiklą. Duotajame faile pateikti dalinai struktūrizuoti duomenys apie siuntų išvežiotojų sustojimus. Failas “duom\_full.txt” ir sutrumpintas duomenų failas “duom\_cut.txt”.



Namų darbe atliksiu **2 užduotį,** joje reikia iš duomenų rinkinio rasti maršrutus, kurie aplanko daugiau nei vieną geografinę zoną ir kai tai daroma tą pačią dieną.

Pirmiausia reikia sukurti Map programinį kodą, kuris iš duomenų mums atrinktu tik reikiamus parametrus, kuriuos analizuosime. Šiame duomenų rinkinyje yra pateikiama daug informacijos apie pristatymūs, bet mums reikia tiktai maršruto, geografines zonos ir datos. Kadangi reikia rasti maršrutus, kurie aplanko daugiau nei viena geografine zona tą pačią dieną, reiškia raktas yra pagal du parametrus: maršrutą ir dieną, tai reiskia reikia isvesti toki rakta kuris jungia siuos abu parametrus.

Pseudo kodas:

data = input()

for element in data:

parameters = element.split()

for param in parameters:

key, value = param.split()

if key == 'marsrutas':

marsrutas = value

elif key == 'geografine zona':

geografine\_zona = value

elif key == 'sustojimo data':

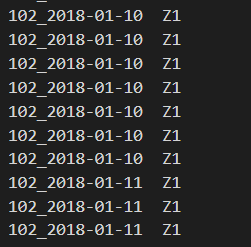
sustojimo\_data = value

if marsrutas and geografine\_zona and sustojimo\_data are not None:

key = marsrutas + '\_' + sustojimo\_data

emit(key, geografine\_zona)

Šis pseudo kodas nuskaito duomenų failą, išskiria atskirus elementus, kuriuose yra parametrai apie tam tikrą siuntą ar maršrutą ir t.t. Tada einama per kiekviena elemento parametrą ir tikriname ar tas elementas turi mums reikiamus duomenis (maršrutą, sustojimo datą ir geografinę zoną). Jei bent vieno iš trijų parametrų nėra elemente, mes negalima išvesti šitų duomenų ir einama toliau. Jei yra visi duomenys, išvedamas maršrutas ir sustojimo data kaip raktas (sujungiant šiuos parametrus per brūkšnelį), o reikšmė būtų geografinė zona. Surinkti duomenys atrodo taip:



Reiktų paminėti, kad po Map funkcijos įvyksta rikiavimas pagal raktą, todėl išvesti duomenys jau būna surikiuoti. Toliau reikia sukurti Reduce programinį kodą. Kadangi jau turime mums reikiamus duomenis, naudodamiesi Reduce kodu mes iš duomenų išgauname mums reikalingą informaciją. Šiuo atveju mes norime sužinoti ar su mūsų turimu raktu (maršrutas ir sustojimo data) buvo aplankyta skirtingų geografinių zonų, t. y. ar yra skirtingų reikšmių su tokiu raktu.

Pseudo kodas:

current\_key = None

current\_set = set()

lines = input()

for line in lines:

marsrutas\_data, zona = line.split()

if current\_key == marsrutas\_data:

current\_set.add(zona)

else:

if current\_key != None:

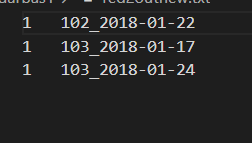
if len(current\_set) > 1:

emit(1, current\_key)

current\_set = set(zona)

current\_key = marsrutas\_data

Reduce kodas nuskaito raktą ir reikšmę. Kadangi žinome, kad duomenys yra surikiuoti, išsaugome dabartinį raktą ir tikriname ar jis nepasikeitė. Jei raktas nepasikeitė nuskaitytą reikšmę dedame į rinkinį, kuriame yra saugomos unikalios geografinių zonų reikšmės. Jei raktas pasikeitė, tikriname ar rinkinyje yra daugiau nei viena reikšmė (pagal sąlygą reikia rasti maršrutus, kurie aplankė kelias geografines zonas tą pačią dieną) ir išvedame tą maršrutą su data. Ir taip einama per visus duomenis. Gautas rezultatas yra toks:



Dar papildomai buvo grąžintas vienetas, nes Reduce turi išvesti raktas reikšmė tipo duomenis, bet šiuo atveju jis nereikalingas. Taigi, naudojantis MapReduce buvo gauti maršrutai aplankę kelias geografines zonas tą pačią dieną.