KLIININEN TIETOPALVELU
CENTRE FOR CLINICAL INFORMATICS

HOSPITAL DISTRICT OF
SOUTHWEST FINLAND

The Clinical Data Refinery

Management and Administration of the Analytics Environment

Editors:
Anna Hammais, Juha Varjonen and Arho Virkki

CCI
CENTRE FOR
CLINICAL INFORMATICS

KLIININEN TIETOPALVELU
CENTRE FOR CLINICAL INFORMATICS

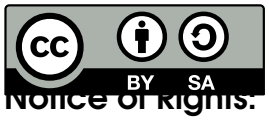
The Clinical Data Refinery

Management and Administration of the Analytics Environment

Editors:
Anna Hammais
Juha Varjonen
Arho Virkki
together with
The Clinical Informatics Team

Sponsors:
Päivi Rautava, Turku Clinical Reserch Centre
Tarja Laitinen, Head of the Department of Pulmonary Medicine
The Finnish Innovation Fund Sitra

Contact:
email: ktp@tyks.fi
mail: Kliininen tietopalvelu
TYKS TE6, PL52
20521 Turku



KTP Documentation

Welcome to KTP Wiki!

These pages provide technical documentation about the data analytics plattform setup and holds copies of some essential binary tools. The Wiki is stored physically as a bare Git repository at `ktp@ktpdoc.vsshp.net:/opt/git/KTPWiki.git`.

The documentation is divided into several main topics

1. Process Flow
2. Data Sources
3. Data Semantics
4. Data Share Log
5. Code Review Process
6. In-House Server Environment
7. Version Control System
8. Software Installation, Administration and Backups
9. SQL and R programming
10. PostreSQL and Hadoop Administration
11. Data Visualization
12. External Research Environment
13. Software Summary

How to Browse and Add Content

The [All] button on top of the page opens a hiearchical view. The complete documentation tree can be accessed at http://ktpdoc.vsshp.net/fileview.

See Daring Fireball's Markdown Tutorial or just click 'Edit' to see how the pages are written.

Book

Open http://ktpdoc.vsshp.net/book/ktp_book.pdf for a printable copy.

CCI Administrators can re-generate the book directly form Wiki Markdown sources with pandoc by running *getbook.sh* at the Git document root.

Data Analysis and ETL Process

Document author: arho.virkki@tyks.fi Contributors: anna.hammais@tyks.fi, juhana.valo@medbit.fi, katja.kanerva-leppanen@medbit.fi

Data Enrichment and Analysis Steps

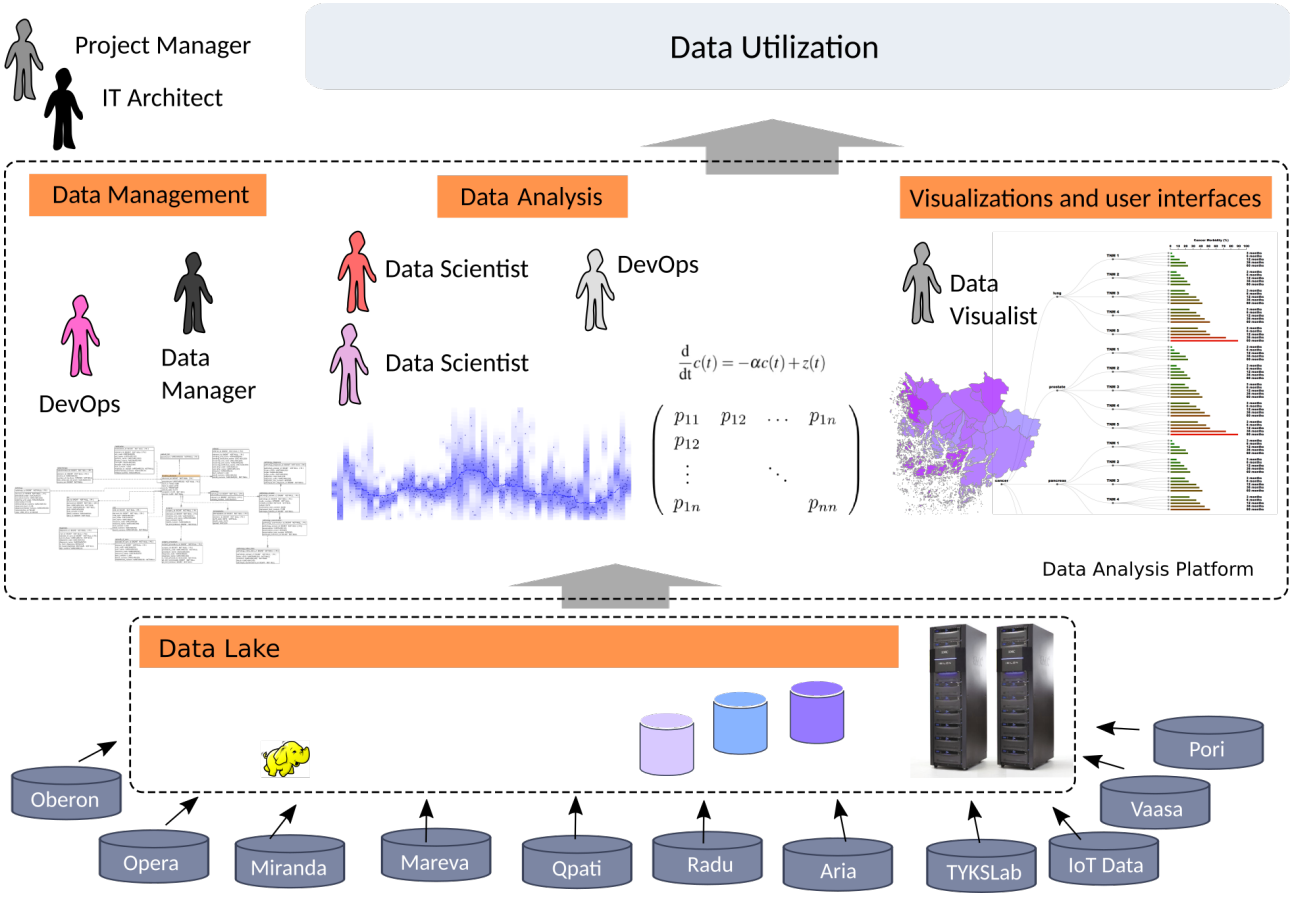


Figure [PDF, SVG]. Process Overview.

Data Flow

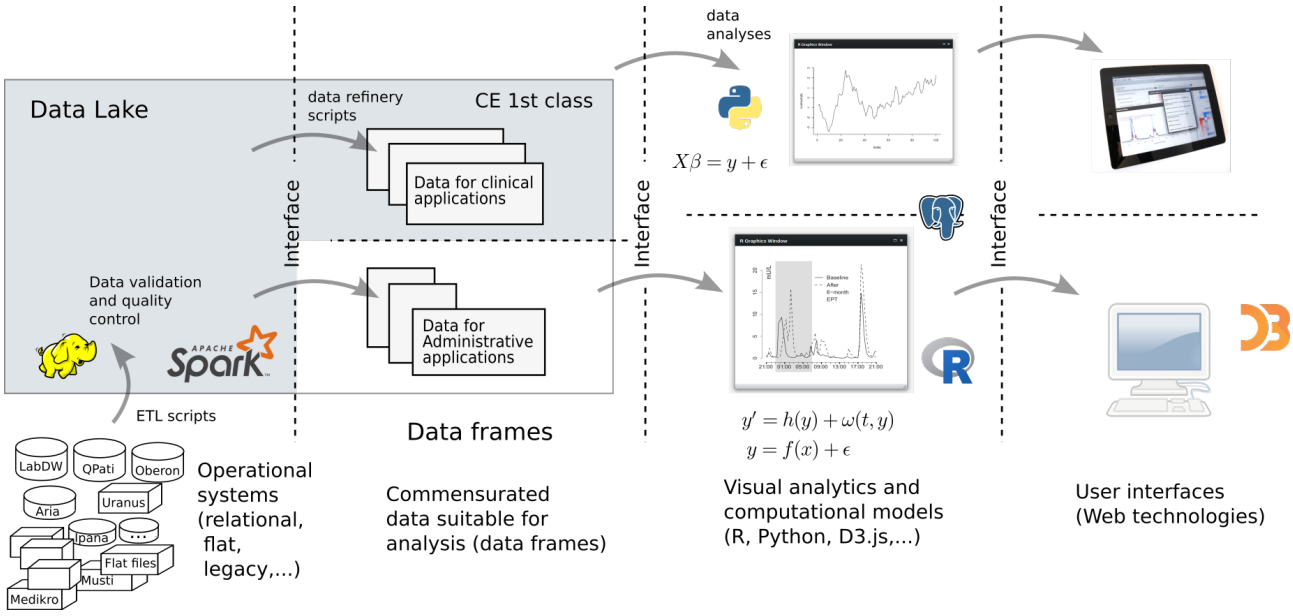


Figure [PDF, SVG]. Data Analysis Workflow.

Research Process Steps

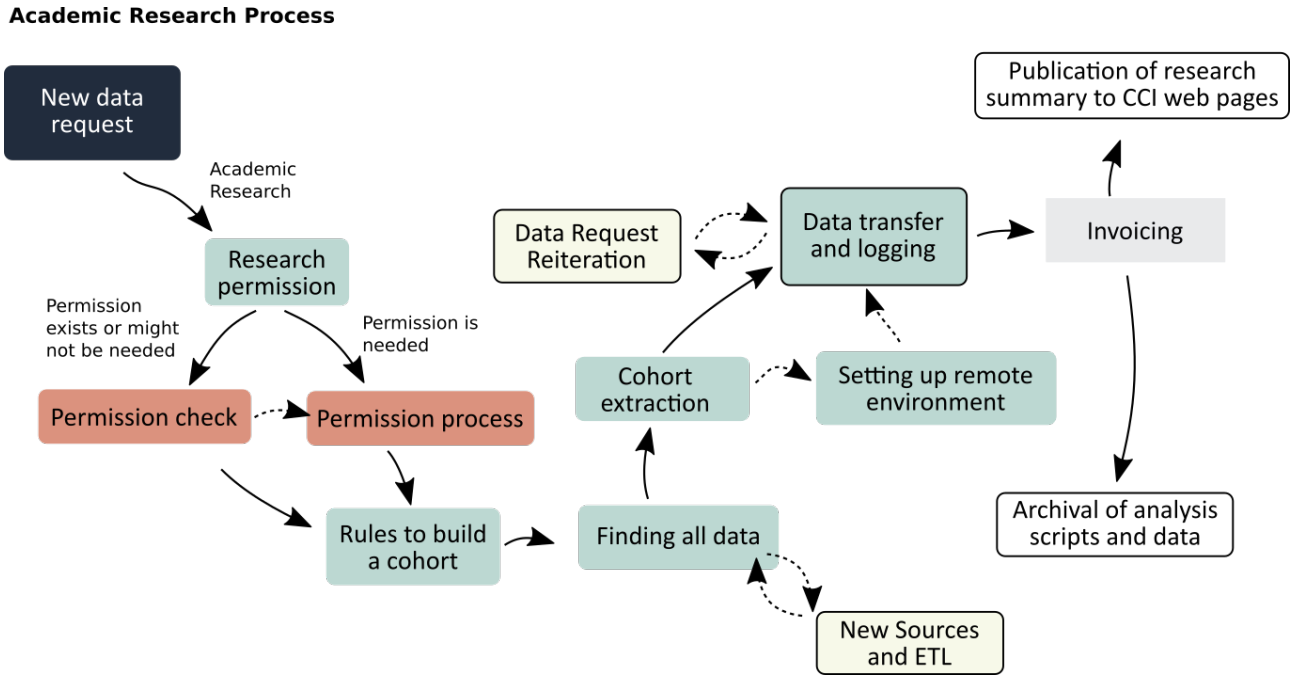


Figure [PDF, SVG]. Academic research process.

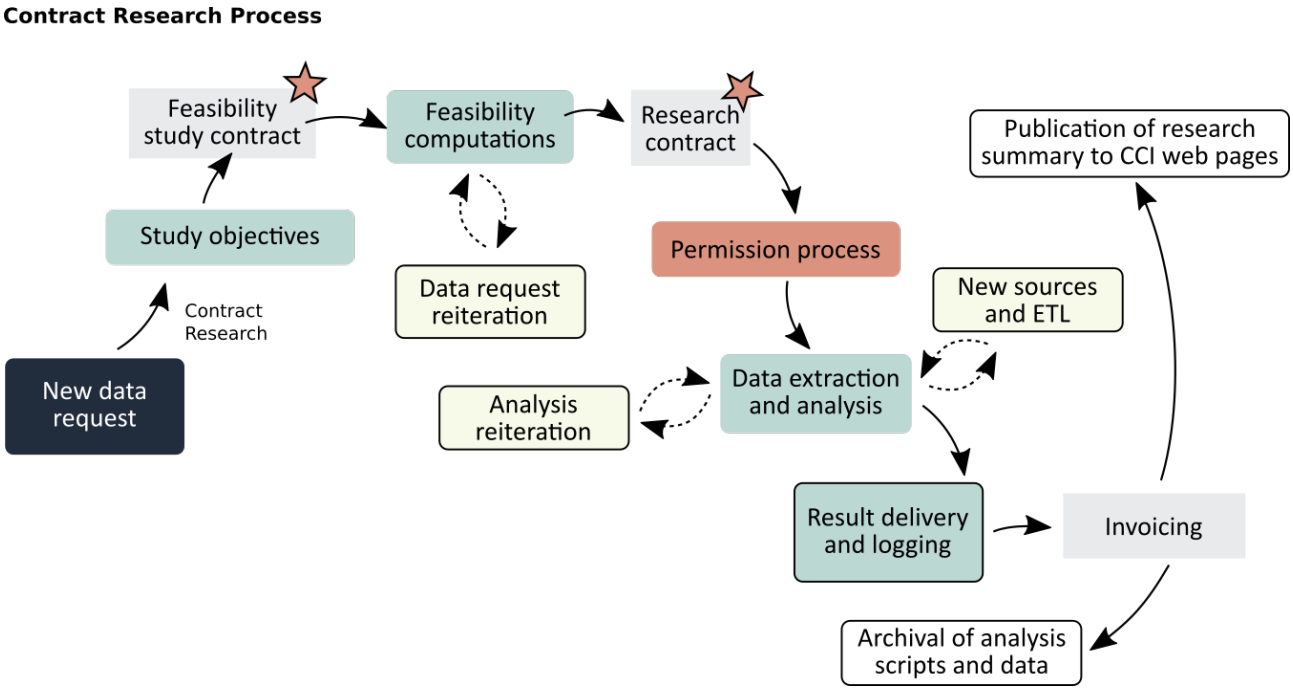


Figure [PDF, SVG]. Contract research process.

New Data Sources and Data Harmonization

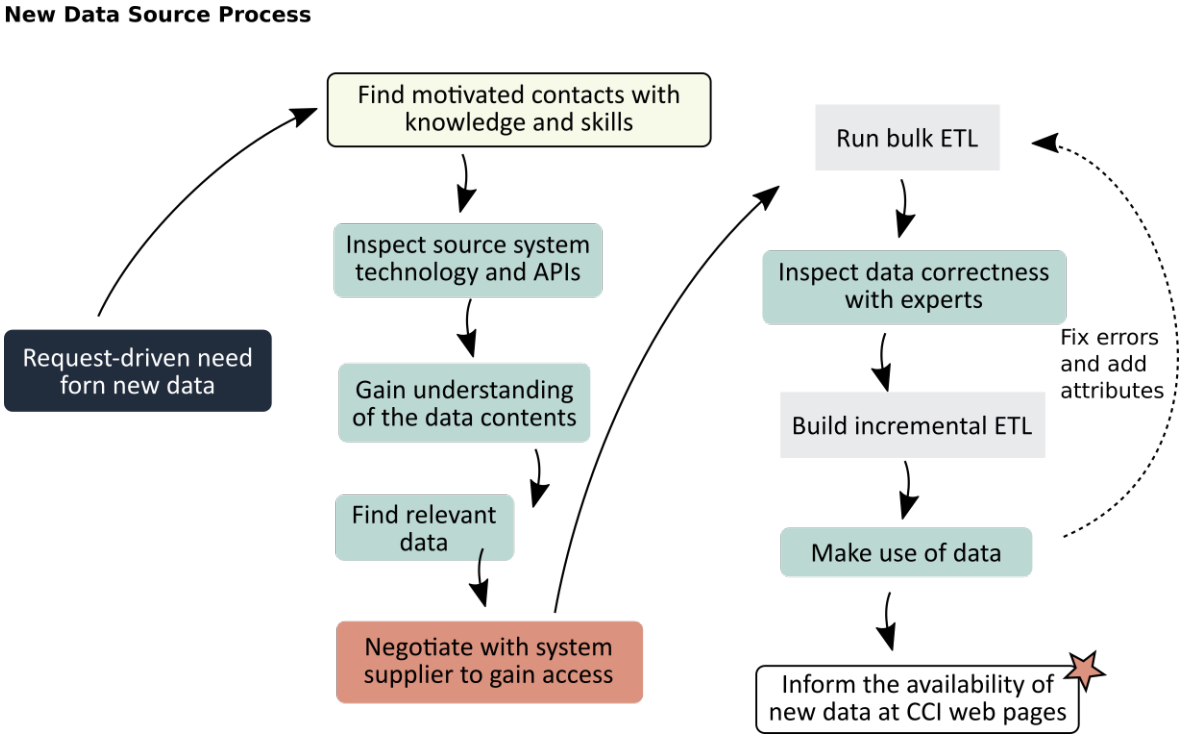


Figure [PDF, SVG]. New data source process.**

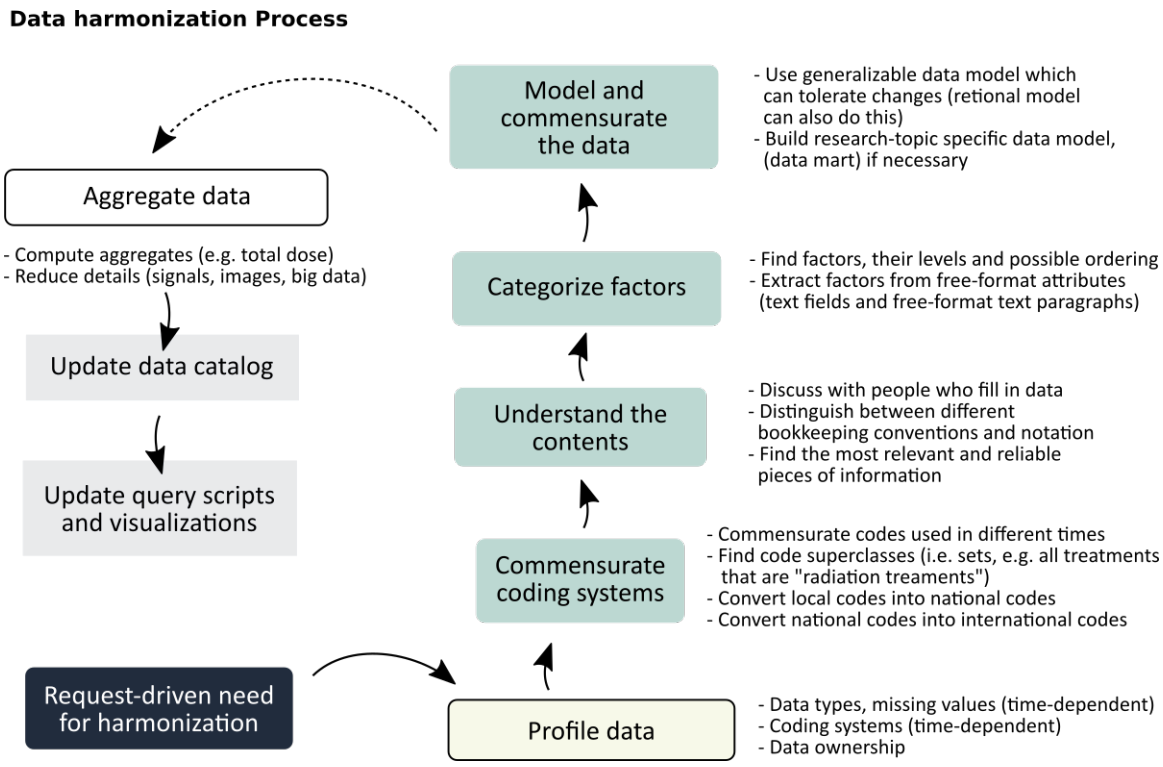


Figure [PDF, SVG]. Data harmonization process.

ETL Steps

1. Data extraction from source
2. File upload
3. Format conversion
4. Type conversion
5. Data integration
6. Semantic unification

ETL Script Repository

The ETL scripts are saved to the Git-repository `ktp@ktpgit:/opt/git/ETL.git`. Two working copies are used in production:

```
ktphadoop.vsshp.net:/var/lib/hive/ETL/
ktpanalytics.vsshp.net:/opt/ktp/ETL/
```

The first is for pre-processing the flat text files at the Hadoop machine, and the second is used with Pentaho Kettle in the data integration phase. The automated etl scripts are run by the *ktp* user. For details, see *crontab -e* as *ktp* at *ktpanalytics*.

Details

Hadoop Environment
PostgreSQL Setup

Medbit Data Warehouse and Data Mart

Document author: anna.hammais@tyks.fi

Preliminary plan: Steps of the ETL process

1. Aloitamme Opera-datasta (seuraavaksi varmaankin Toti ja Radu)
2. Haemme DataMartista Operaan liittyvät faktataulut (DMF_LEIKKAUS ja DMF_TOIMENPIDE) ja niihin liittyvät dimensiot
3. Haemme DW:stä taulun S_Toimenpiteet_Opera, josta saamme pre- ja postop-diagnoosit
4. Haku tehdään taulu kerrallaan meidän staging arealle. Teidän kannassa ei tarvitse tehdä joineja tms. eikä teidän serveri kuormitu varmaan kummemmin.
5. DW:ssä käytetään act_id-sarakkeita pääavaimina ja latpvm-saraketta tiedon uutuuden määrittämiseen.
6. DM:ssa käytetään pääavaimena taulun PK-saraketta ja tiedon uutuuden määrittämiseen sarakkeita DATE_INSERT ja DATE_UPDATE.
7. DW-tasolla pitää aina huomioida valid kentän arvo. Y = viimeisin voimassaoleva (rivi voi silti olla lähteessä poistettu tai peruttu), N = tietovarastosta löytyy riville uudempi rivi, jossa joku arvoista on päivittynyt. Hakea vain valideja tietoja joka kohdassa, eli tiedon täytyy olla validi DW:ssä ja lähteessä, ei riitä että jommassakummassa.
8. Kun alkulataus on tehty, uudet ja muuttuneet tiedot haetaan esim. kerran yössä. Tarkistetaan ennen jokaisen taulun hakua, että teidän latauskierros on valmis, taulusta VarSat_SA_VSSHP.SA_VSSHP_logTable (etlJobName = 'JOB_Tietovarasto_VSSHP' and status = 'ONNISTUI' and endTime is not null).

Kohta 7 ontuu vielä, koska systeemitunnuksellamme SVCL_KTPLukija ei ole lukuoikeutta VarSat_SA_VSSHP-kantaan (vain HAMMAISA on oikeus).

Effector data extraction

Document author: anna.hammais@tyks.fi

1. Start Effector on Tarja Laitinen's computer.
2. Select Etsi –> Lainaus
3. Specify the type of equipment in the Luokitus field.
4. Specify Apuvälineen tila = Kaikki
5. Search
6. Copy spreadsheet with Control+A Control+C
7. Wait until moving Excel-like frame appears around selection.
8. Paste into Excel or text file.

LabDW data process

Document author: anna.hammais@tyks.fi

Data source

Jukka Palko: autofs to biopankki directory on gradient.

Automated new data file retrieval:

In ktp@gradient crontab:

```
# Automatic backup of the 'biopankki' folder
0 6 * * * rsync -e ssh -avut /nas/biopankki/labdw_*.txt
    hive@ktphadoop.vsshp.net:/opt/share/raw/lab_autosync/
```

Opera surgery data from Medbit Data Warehouse

Document author: anna.hammais@tyks.fi

Kohdat 1-4 on tehty kertaalleen kantaan ktp, skeemaan stage_dw. Tauluissa nyt vain valideja rivejä. Tarkoitus jatkossa syöttää näihin tauluihin kaikki uudet validit rivit ja päivittää olemassaolevia rivejä invalideiksi, jos näin on lähteessä käynyt. Eli näissä tauluissa tulee tulevaisuudessa olemaan myös invalideja rivejä (valid='n').

1. Aloitamme Opera-datasta (seuraavaksi varmaankin Toti ja Radu)
2. Haemme DataMartista Operaan liittyvät faktataulut (DMF_LEIKKAUS ja DMF_TOIMENPIDE) ja niihin liittyvät dimensiot
3. Haemme DW:stä taulun S_Toimenpiteet_Opera, josta saamme pre- ja postop-diagnoosit
4. Haku tehdään taulu kerrallaan meidän staging arealle. Medbitin kannassa ei tarvitse tehdä joineja tms. eikä Medbitin serveri kuormitu varmaan kummemmin.
5. DW:ssä käytetään act_id-sarakkeita pääavaimina ja latpvm-saraketta tiedon uutuuden määrittämiseen.
6. DM:ssa käytetään pääavaimena taulun PK-saraketta ja tiedon uutuuden määrittämiseen sarakkeita DATE_INSERT ja DATE_UPDATE. DATE_UPDATE on null, jos tieto on vain kerran insertoitu eikä vielä kertaakaan päivitetty.
7. DW-tasolla pitää aina huomioida valid kentän arvo. Y = viimeisin voimassaoleva (rivi voi silti olla lähteessä poistettu tai peruttu), N = tietovarastosta löytyy riville uudempi rivi, jossa joku arvoista on päivittynyt. Haetaan vain valideja tietoja joka kohdassa, eli tiedon täytyy olla validi DW:ssä ja lähteessä, ei riitä että jommassakummassa. Varmistutaan siitä, että jos tieto menee ei-validiksi, sekin muutos päivittyy KTP:n kantaan.
8. Tätä ei voi toteuttaa vielä, mutta tulevaisuudessa: Kun alkulataus on tehty, uudet ja muuttuneet tiedot haetaan esim. kerran yössä. Tarkistetaan ennen jokaisen taulun hakua, että teidän latauskierros on valmis, taulusta VarSat_SA_VSSHP.SA_VSSHP_logTable (etlJobName = 'JOB_Tietovarasto_VSSHP' and status = 'ONNISTUI' and endTime is not null).

Tarkoitus on nyt muokata dataa päivittyvästi stage_dw-skeemasta main_dev-sckeemaan (main_dev on kehitysversio main-skeemasta). Main_dev-skeeman taulut leikkaus ja leikkaus_toimenpide pitäisi täyttää (nyt tyhjät). Koodi_id-kentät voi alkuvaiheessa jättää tyhjäksi ja laittaa pelkän vastaavan selitteen (esim. leikkaus.paatoimenpide_koodi_id tyhjäksi, ja toimenpiteen koodi ja nimi niitä vastaaviin sarakkeisiin). Samoin hoitava_osasto_id ja toimenpideosasto_id tyhjäksi, ja vain selite mukaan.

PDI-transformaation/jobin tekemisen ideoita voi katsoa Arhon tekemistä, tai esim minun ETL/-main/transformations/patologia/insert/insert_patologia.kjb. Esimerkissä on yritetty täytellä potilas-, henkilon_identiteetti- ja potilas_asia-taulut myös (main-skeema vaatii tämän). Ainakin jobin/transformaation parametriksi on hyvä laittaa kohdeskeema ja lähdeskeema (kuten patologia-esimerkissä), jotta niitä on helppo muuttaa myöhemmin. Voit käyttää nyt stage_dw ja main_dev.

Voisi varmaankin laittaa stage_dw-cdc_time-tauluun merkinnän myös stage->main-tiedon päivityksistä. Eli aina kun jobi käynnistyy laittamaan stagesta opera-dataa mainin tauluun leikkaus, tulisi tuohon cdc-tauluun merkintä, jossa taulun nimi on "main_dev.leikkaus" ja current load timestampiksi tulisi se kyseinen hetki. Kun lataus on ohi, timestamp siirrettäisii last loadiksi ja currentiin tulisi null. Siten olisi seuraavalla kierroksella helppo katsoa, milloin on viimeksi tehty main_dev.leikkaukseen päivitys (tuo timestamp), ja ottaa sitä uudemmat datat stage_dw-tauluista. Ja sitten päivittää näiden päälle uudet timestampit.

Queries to work on PostgreSQL database ktp, schema stage_dw

Join query to get (roughly) the necessary data for **main.leikkaus**:

```
select *
```

```
from stage_dw.dm_dmf_leikkaus as l
inner join stage_dw.dm_dmd_erikoisala as e1 on l.FK_DMD_ERIKOISALA = e1.PK_DMD_ERIKOISALA
inner join stage_dw.dm_dmd_erikoisala as e2 on l.FK_DMD_ERIKOISALA_PAATOIMENPIDE =
    e2.PK_DMD_ERIKOISALA
inner join stage_dw.dm_DMD_TOIMENPIDE as t1 on l.FK_DMD_TOIMENPIDE_PAA = t1.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_DMD_JONOTTAMISENSYY as j on l.FK_DMD_JONOTTAMISENSYY =
    j.PK_DMD_JONOTTAMISENSYY
inner join stage_dw.dm_dmd_osasto as o1 on l.FK_DMD_OSASTO_HOITAVA = o1.PK_DMD_OSASTO
inner join stage_dw.dm_dmd_laitos as la on o1.FK_DMD_LAITOS = la.PK_DMD_LAITOS
inner join stage_dw.dm_dmd_osasto as o2 on l.FK_DMD_OSASTO_TOIMENPIDE = o2.PK_DMD_OSASTO
inner join stage_dw.dm_DMD_TOIMENPIDE as t2 on l.FK_DMD_TOIMENPIDE_PUOLISUUS = t2.PK_DMD_TOIMENPIDE
inner join stage_dw.dw_S_Person as p on (l.potilas = p.md5Tunniste and p.valid = 'y')
inner join stage_dw.dw_S_Henkilotunnus as a on (p.entity_id = a.entity_id and a.valid = 'y')
where l.lippu_tmpValid = 1;
```

Join query to get (roughly) the necessary data for **main.leikkaus_toimenpide**:

```
select s.act_id as S_Toimenpiteet_Opera_act_id,
s.preopDiagnoosiCodeIn, s.preopDiagnoosiValue,
s.postopDiagnoosiCodeIn, s.postopDiagnoosiValue,
t.*
from
stage_dw.dm_dmf_toimenpide as t
inner join stage_dw.dm_dmf_leikkaus as l on t.tmpActID = l.H_tmp_act_id
inner join stage_dw.dm_DMD_TOIMENPIDE as t1 on t.FK_TOIMENPIDE_TOTEUTUNUT = t1.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_DMD_TOIMENPIDE as t2 on t.FK_TOIMENPIDE_SUUNNITELTU = t2.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_DMD_TOIMENPIDE as t3 on t.FK_TOIMENPIDE_PUOLISUUS = t3.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_DMD_ERIKOISALA as e on t.FK_DMD_ERIKOISALA = e.PK_DMD_ERIKOISALA
inner join stage_dw.dw_S_Toimenpiteet_Opera as s on t.toimenpideActID = s.act_id
where s.valid = 'y'
and s.tmpValid = 1
and s.toimenpideValid = 1;
```

Query versions to work on DM and DW

main.leikkaus

```
select *
from Toimintatilasto.dmf_leikkaus as l
inner join yhteinen.dmd_erikoisala as e1 on l.FK_DMD_ERIKOISALA = e1.PK_DMD_ERIKOISALA
inner join yhteinen.dmd_erikoisala as e2 on l.FK_DMD_ERIKOISALA_PAATOIMENPIDE = e2.PK_DMD_ERIKOISALA
inner join yhteinen.DMD_TOIMENPIDE as t1 on l.FK_DMD_TOIMENPIDE_PAA = t1.PK_DMD_TOIMENPIDE
inner join yhteinen.DMD_JONOTTAMISENSYY as j on l.FK_DMD_JONOTTAMISENSYY = j.PK_DMD_JONOTTAMISENSYY
inner join yhteinen.dmd_osasto as o1 on l.FK_DMD_OSASTO_HOITAVA = o1.PK_DMD_OSASTO
inner join yhteinen.dmd_laitos as la on o1.FK_DMD_LAITOS = la.PK_DMD_LAITOS
inner join yhteinen.dmd_osasto as o2 on l.FK_DMD_OSASTO_TOIMENPIDE = o2.PK_DMD_OSASTO
inner join yhteinen.DMD_TOIMENPIDE as t2 on l.FK_DMD_TOIMENPIDE_PUOLISUUS = t2.PK_DMD_TOIMENPIDE
inner join Varsat_DW_VSSHP.dbo.S_Person as p on (l.potilas = p.md5Tunniste and p.valid = 'y')
inner join Varsat_DW_VSSHP.dbo.S_Henkilotunnus as a on (p.entity_id = a.entity_id and a.valid = 'y')
where l.lippu_tmpValid = 1;
```

main.leikkaus_toimenpide

```
select s.act_id as S_Toimenpiteet_Opera_act_id,
s.preopDiagnoosiCodeIn, s.preopDiagnoosiValue,
s.postopDiagnoosiCodeIn, s.postopDiagnoosiValue,
t.*
from
Toimintatilasto.dmf_toimenpide as t
inner join Toimintatilasto.dmf_leikkaus as l on t.tmpActID = l.H_tmp_act_id
inner join Yhteinen.DMD_TOIMENPIDE as t1 on t.FK_TOIMENPIDE_TOTEUTUNUT = t1.PK_DMD_TOIMENPIDE
inner join Yhteinen.DMD_TOIMENPIDE as t2 on t.FK_TOIMENPIDE_SUUNNITELTU = t2.PK_DMD_TOIMENPIDE
inner join Yhteinen.DMD_TOIMENPIDE as t3 on t.FK_TOIMENPIDE_PUOLISUUS = t3.PK_DMD_TOIMENPIDE
inner join Yhteinen.DMD_ERIKOISALA as e on t.FK_DMD_ERIKOISALA = e.PK_DMD_ERIKOISALA
inner join Varsat_DW_VSSHP.dbo.S_Toimenpiteet_Opera as s on t.toimenpideActID = s.act_id
where s.valid = 'y'
and s.tmpValid = 1
and s.toimenpideValid = 1;
```

PostgreSQL Setup

PostgreSQL database is used both as part of the staging area and for hosting the main data schema. The *ktp* database is owned by a *ktp* user, and divided into several schemas to clarify the database structure. The idea is to separate source data, ontological mappings, cleansed and curated data, and temporal tables from each other.

```
CREATE USER "ktp"
  WITH PASSWORD '<the_system_passwd_here>'
  SUPERUSER;

CREATE DATABASE ktp WITH OWNER ktp;

CREATE SCHEMA stage_hadoop;
-- One stage_<source_system> each data source
-- ...
CREATE SCHEMA map;
CREATE SCHEMA main;
CREATE SCHEMA static;
CREATE SCHEMA temp;

ALTER DATABASE ktp SET search_path TO
temp, main, map, stage_hadoop, stage_uraods, static;
```

The database can be connected from command line with

```
psql -U ktp -h ktppg.vsshp.net -d ktp
```

Schemas can be listed with `\dn`, and search path printed with `SHOW search_path;`

QPati data integration from XML

Document author: anna.hammais@tyks.fi

Data source

ktpanalytics: /mnt/raw/qpati_xml_autosync directory contains files coming from QPati via "Matin allas" (data pool).

Timestamps in the XML files

Even though the timestamps end with "Z" denoting "Zulu time"/UTC/GMT, they are really just the local time of the system, i.e. EET (eastern European time) or EEST (eastern European summer time).

Statement tables

The following tables contain more than two columns:

```
SYDNEY-LUOKITUS SUPPEA.tbl
SYDNEY-LUOKITUS.tbl
MUNUAINEN IMMUNOFLORESENSSI.tbl
ENNUSTETEKIJÄT.tbl
RINTASYÖVÄN ENNUSTETEKIJÄT.tbl
IHO IMMUNOFLORESENSSI.tbl
ZAvaus.tbl
SOLULASKENTA BAL.tbl
VIRTAUSSYTOMETRIA.tbl
BETHESDA 2001.tbl
```

In the ETL (*update_patologia.kjb*), these tables are handled by the script

```
ETL/R/qpatitable/qpati_table_<version>.R
```

The following tables contain only one column:

```
OBDUKTIO MIKROSKOOPPISET NÄYTTEET.tbl
AIVO-OBDUKTIO MIKROSKOOPPISET NÄYTTEET.tbl
```

The values in this column are entered with the variable/suure name 'Suure:' in the table *main.patologia_taulukkoarvo*.

Data Interpretation and Semantics

Author: anna.hammais@tyks.fi

Semantics

- Hoitoviiveet
- Naming conventions
- iPana ja Mama
- Miranda ODS Laakitys
- Kemokur vs. Marela
- Labra
- QPati
- Radu
- Opera
- Kardiologia
- Diagnoosit ja palvelut (käynti/osastohoito)
- Potilaskertomus_teksti
- Hoitotaulukko
- ODS documentation
- Lähetteet
- Varauskirja
- Lääkärit ja yksiköt
- Kuvantamisdata
- Taloustilasto
- Syöpä

Hoitoviiveet syöpäpotilailla

Document authors: juha-matti.varjonen@tyks.fi ja anna.hammais@tyks.fi

Skeema

```
cancers_treatment_delays
```

Materialisoidut näkymät ja taulut luodaan tässä järjestyksessä, koska ne riippuvat osittain toisistaan:

```
cancers_treatment_delays. mv_patient_list -- tässä valmiiksi jo kaikki syöpäpotilaat
cancers_treatment_delays.mv_patient_timeline_elements -- em. potilaiden kaikki tapahtumat
cancers_treatment_delays.mv_<tietty_syöpä>_patients -- esim. gkir, gyne, uro, jne.
cancers_treatment_delays.valinta_matriisi
func.hoitoviive_arvotaulukko
cancers_treatment_delays.mv_treatment_delays_all
```

Näkymää

```
mv_patient_timeline_elements
```

voidaan käyttää inner joinilla johonkin tietyn kohortin potilaslistaan. Siitä saa helposti esiin potilaan tapahtumat aikaleimoineen.

Kun halutaan muokata näkymiä

Otetaan kaikki luontiskriptit talteen.

Pudotetaan

```
cancers_treatment_delays.mv_treatment_delays_all
cancers_treatment_delays.mv_treatment_delays_nkir_uro_gyne_gkir_ply
```

Pudotetaan

```
cancers_treatment_delays.mv_gkir_patients
cancers_treatment_delays.mv_gyne_patients
cancers_treatment_delays.mv_keu_patients
cancers_treatment_delays.mv_keu_patients_all
cancers_treatment_delays.mv_uro_patients
cancers_treatment_delays.mv_ply_patients
cancers_treatment_delays.mv_suu_patients
```

Pudotetaan

```
cancers_treatment_delays.mv_patient_timeline_element
```

Luodaan uudelleen

```
cancers_treatment_delays.mv_patient_timeline_element
```

niin, että leikkauksissa on concat toimenpidekoodi ja pitkä selite dmd_toimenpide-taulusta

5. muokataan

```
cancers_treatment_delays.get_treatment_delays-funktiota
```

niin, että suu-potilaiden osio tulee mukaan

Cardiology MSAccess Database tables

Document author: anna.hammais@tyks.fi

Relevant Tables

SPTulo has TuloID, that can be found in SPLAngio and SPLPtca. If the TuloID in SPLPtca is empty, it means that the "Tulo"/arrival is not a visit related to PTCA procedure. Then, logically, the arrival should be related to Angio.

The tables can be linked as follows:

```
Tulo --  (TuloID) -- SPLAngio -- (AngioID) -- SPLPtca
```

or if the PTCA is directly linked to Tulo (patient had only PTCA, not angio?):

```
SPLTulo -- (TuloID) -- SPLPtca
```

Table SPLPot contains detailed information about the patients.

iPana ja Mama

Document author: anna.hammais@tyks.fi

iPana-tekstitiivistelmiä Mirandan SYN-näkymässä

```
select extract(year from hoitotapahtuma_alkuhetki_arvio), count(distinct henkilotunnus)
from text_mine.teksti where nakyma_selite = 'SYN'
group by extract(year from hoitotapahtuma_alkuhetki_arvio)
order by extract(year from hoitotapahtuma_alkuhetki_arvio);
```

tilanne 2016-12-20:

Vuosi	Potilasmäärä
2004	24
2005	123
2006	61
2007	4582
2008	5814
2009	6054
2010	6331
2011	6071
2012	6258
2013	6290
2014	6381
2015	6185
2016	5730

Löytyykö Apgar-pisteet tekstistä?

```
select extract(year from hoitotapahtuma_alkuhetki_arvio), count(distinct henkilotunnus)
from text_mine.teksti where nakyma_selite = 'SYN'
and teksti ~* 'apgar'
group by extract(year from hoitotapahtuma_alkuhetki_arvio)
order by extract(year from hoitotapahtuma_alkuhetki_arvio);
```

tilanne 2016-12-20:

Vuosi	Potilasmäärä
2005	3
2006	4
2007	2092
2008	4831
2009	5059
2010	5091
2011	4840
2012	4794
2013	4718
2014	4829
2015	4644
2016	4128

Mama SasDatassa vuosilta 1995 - 2007

Kemokur vs. Marela

Document author: mikhail.stepanov@tyks.fi & anna.hammais@tyks.fi

Kemokur sisältää seuraavat kolme tärkeä taulua: POTILAAN_HOITOKUURIT_ODS, POTILAAN_SYOPALAAKKEET_ODS, POTILAAN_SYOPAL_ANTOK_ODS.

POTILAAN_HOITOKUURIT_ODS

Taulu sisältää tiedot potilaasta, hoitokuurista ja syklistä. Yksi rivi on yksi sykli. Kuuri tunnistuu yhdistelmällä {potilasnumero, hoitokuuri_numero}.

Potilaan pituus/paino: melkein aina oikein.

Potilaan zubrod (yleensä kuulemma sanellaan tekstiin, ei merkitä Kemokuriin, sanoo Eetu H.):

```
ZUBROD               N
                   64677
795901429627809617   691
306901445220836941   277
006901440629786533   1086
```

Selitteet zubrod-koodeille yllä olevilla koodeilla löytyy:

```
select * from koodisto_ods where koodiarvon_numero in
('795901429627809617','306901445220836941','006901440629786533');
```

Mikä hoitokuuri on meneillään: Hoitokuuri_numero ja hoitokuuri_nimi pitäisi matchata toisiaan, mutta varmasti on mahdollista muokata hoitokuuri_nimeä, sen takia on hoitokuuri_numerot joilla on monta hoitokuuri_nimiä ja vice versa.

Syklin tiedot: syklin aloitus- ja lopetuspäivä, JARJESTYSNUMERO täällä hoitokurissa, sekä sykin pituus.

Ongelmana tässä on se, että emme tiedä mihin hoitojaksoon hoitokuurit kuuluvat, koska samanaikaisesti voi olla monta hoitokuria ja niillä on oma järjestysnumerointi, sekä sama hoitokuuri_nimi voi myöhemmin alkaa uudestaan ja samalla numerointi alkaa uudestaan.

POTILAAN_SYOPALAAKKEET_ODS

Sisältää lääkkeet, jotka ovat jokaisessa syklissä/hoitokuurissa: mm geneerinennimi ja käytetty annos

POTILAAN_SYOPAL_ANTOK_ODS

monilla hoitokuurilla puuttuu

AURIA notes

Kemokur

Tässä taulussa on aika hyvin kaikki syöpälääkkeet, mukaanlukien subkutaani trastuzumabi. Kannattaa kuitenkin huomioida, että hematologisia syöpiä hoitavat lääkärit laittavat käyttämänsä hoidot lääkemääräyksiin, eivät Kemokuriin!

Syöpälääkkeet tilataan apteekista Kemokurin kautta, joten tämän pitäisi sisältää hyvin myös iv-sytostaatit. Poikkeuksena lastenklinikka, joka alkoi käyttämään Kemokuria vasta kesällä 2016 - tätä ennen tiedot kirjattiin Marelaan.

Marela

Sisältää iv-sytostaatit hyvin. Huomioitavaa kuitenkin, että Herceptin/trastuzumabi annetaan myös subkutaanisti, jolloin sitä ei kirjata Marelaan! Herceptin aloitetaan yleensä iv:nä, mutta on myös tapauksia, jotka saavat ainoastaan subkutaanisti piikkejä jne.

Kuvantamisdata

Document author: anna.hammais@tyks.fi

Eri tyyppiset kuvantamismenetelmät erityisesti keuhkojen kuvantamisessa

TT eli CT

Käyttää röntgensäteitä. Otetaan useita poikkileikkauskuvia potilaasta päälaesta alkaen läpi koko kehon. Voidaan ottaa myös vain kehkojen alueesta. Lääkäri rullaa hiirellä läpi potilaan näytöllä.

Erilaisilla potilailla ja eri sairauksien diagnostiikassa käytetään eri protokollia. Esim. traumatilanteissa (päivystys) ei usein tarvita varjoainetta.

Ohutleike-TT: leikkeet millin paksuisia, ei oteta millin välein vaan harvemmin. Ohutleike-TT:ssä tarkempi kuva kuin tavallisessa. Keuhkopussit näkyvät mustana, verisuonet valkoisena, koska niissä varjoainetta. Myös tuumorit näkyisivät valkoisena, koska varjoaine kertyy niihin. Paksuuntuneet keuhkoputkien reunat ovat huono juttu.

Bronkiektasia

- keuhkoputken seinämien paksuuntuminen ja keuhkoputkien laajeneminen suuremmaksi kuin viereinen keuhkovaltimo("sinettisormukset" eli rengas=keuhkoputki ja sintetti=valtimo)
- keuhkoputkien näkyminen keuhkon reuna-alueilla
- (keuhkoputkien läpimitta ei kapene matkallaan kohti keuhkon reuna-aluetta)

Magneettikuvaus (MRI, MT)

Käytetään usein luuston kuvaamiseen. Ei saa liikkua, siksi huono keuhkokuvantamiseen, koska keuhkot liikkuvat. Kuvaus kestää puoli tuntia. Säderasitus vähemmän haitallinen kuin röntgensäteily.

Röntgen

Normaalisti röntgenkuva rintakehästä otetaan edestä ja sivulta. Tästä näkyy mm., onko sydän normaalikokoinen. Keuhkopussin (?) pohjat ovat normaalisti kulmikkaat mutta eivät jos siellä on nestettä.

Sanastoa

- Natiivi = ilman varjoainetta
- TT = tietokonetomografia
- CT = computer tomograpy
- HR-CT / HR-TT = ohutleike-TT / high-resolution CT
- MT / MRI = magneettikuvaus (magnetic resonance imaging)
- fantomi/phantom = dummy-malli potilaasta, jota käytetään testaukseen ja laitteiden kalibrointin

Lääkärit, yksiköt ja toimialueet

Lääkärien eri yksiköissä antamien diagnoosien määrät viimeisen vuoden aikana:

```
create view map.v_laakarit as
select
d.toteaja_sukunimi || ' ' || d.toteaja_etunimi as nimi,
d.toteaja_toimipiste_koodi,
y.yksikko_nimi,
k.tulosyksikko_selite,
k.vastuualue,
y.yks_kustpaikka,
count(*) as maara,
max(count(*)) over (partition by d.toteaja_sukunimi || ' ' || d.toteaja_etunimi) as maximi

from stage_uraods.diagnoosi d
left join stage_uraods.yksikko y ON
d.toteaja_toimipiste_koodi::text = y.yksikko_koodi::text
OR d.toteaja_toimipiste_koodi::text = y.yksikko_koodi::text
left join stage_dw.mv_yksikot_kaikki k
ON d.toteaja_toimipiste_koodi::text = k.vo_toimipiste_koodi::text

where d.luontihetki_s > (now() - interval '1 year') and toteaja_tyyppi_selite = 'Lääkäri''

group by toteaja_sukunimi || ' ' || toteaja_etunimi,
toteaja_toimipiste_koodi, y.yksikko_nimi,
k.tulosyksikko_selite, k.vastuualue, y.yks_kustpaikka;
```

Lääkärien yksikkö, jossa ovat antaneet eniten diagnooseja vuonna viimeisen vuoden aikana:

```
create view map.v_laakarit_ja_yksikot as
select nimi, toteaja_toimipiste_koodi, yksikko_nimi,
tulosyksikko_selite,
vastuualue,yks_kustpaikka,
maara
from map.v_laakarit where maara = maximi;
```

stage_dw.mv_yksikot_kaikki on kahden kyselyn unioni.

```
Vanhat yksiköt (<alkupuolisko 2013) tulevat kyselyllä

SELECT
vo_toimipiste_koodi,
yksikko_nimi,
tulosyksikko_selite,
tulosryhma_selite,
vastuualue,
kustannuspaikka
FROM func.yksikot

Uudet yksiköt tulevat stage_dw ETL:n mukana kyselyllä:

SELECT DISTINCT o1.vastuualue AS vo_toimipiste_koodi,
o1.vastuualue_selite AS yksikko_nimi,
o1.tulosyksikko_selite,
o1.tulosryhma_selite,
o1.vastuualue,
o1.kustannuspaikka
FROM stage_dw.dmd_organisaatio o1
WHERE o1.vastuualue IS NOT NULL AND o1.vastuualue_selite IS NOT NULL AND o1.vastuualue::text <>
    ''::text AND o1.tulosryhma_selite::text <> 'TYKS2013'::text AND NOT (EXISTS ( SELECT
    o2.pk_dmd_organisaatio
FROM stage_dw.dmd_organisaatio o2
WHERE o2.vastuualue::text = o1.vastuualue::text AND o2.vastuualue_selite::text <>
    o2.vastuualue::text AND o1.vastuualue_selite::text = o1.vastuualue::text)) AND NOT (EXISTS (
    SELECT o3.pk_dmd_organisaatio
FROM stage_dw.dmd_organisaatio o3
WHERE o3.vastuualue::text = o1.vastuualue::text AND o3.vastuualue_selite::text <>
    o3.vastuualue::text AND o1.vastuualue_selite::text <> o1.vastuualue::text AND
    o3.vastuualue_selite::text ~~ '%[a-zA-Z]%'::text AND o1.vastuualue_selite::text !~~
    '%[a-zA-Z]%'::text))     AND NOT (EXISTS ( SELECT o4.pk_dmd_organisaatio
FROM stage_dw.dmd_organisaatio o4
WHERE o1.vastuualue::text = o4.vastuualue::text AND o1.pk_dmd_organisaatio <>
    o4.pk_dmd_organisaatio AND o4.regset::text > o1.regset::text))
```

Labradata LabDW

Document author: anna.hammais@tyks.fi

Raakadata

labdw_data

Raakadataa. Lähes joka sarakkeessa koodi, joka täytyy kääntää selitteeksi käyttäen kyseiselle sarakkeelle määritettyä koodistoa. Raakadata poimitaan viikottain keskiviikkoisin biopankki-siirtokansioon.

labdw_columns

Sarakkeille määritetyt koodistot ja sarakkeiden nimien selitteet löytyvät tiedostosta labdw_columns. Samalle sarakkeelle on usein määritelty kaksi koodistoa, joissa onneksi ei lähes koskaan ole päällekkäisiä koodiarvoja.

labdw_codetables

Koodiarvojen selitteet kullekin koodistolle löytyvät tiedostosta labdw_codetables. Sarake codetable kertoo koodiston, code on koodi ja text on selite.

Datan avaimet

Raakadatassa ei ole pääavainta. testid-sarake (Tutkimusriviavain) on eräänlainen tutkimuspaketin tunniste, ja tutkimuspakettiin kuuluu monta eri tutkimusta. etlstamp (stage_hadoop-kannassa “dt”, aurian näkymissä “labdw_biopankki_poiminta_pvm”) kuvaa ilmeisesti ajankohtaa, jolloin tieto on tuotu labdw-tietokantaan MultiLabista. Pääavaimen tapaisena voi käyttää yhdistelmää {testid, test}, eli tietyn testipaketin tiettyä testiä, ja etl-aikaleiman perusteella päätellään, että kyseessä on saman testin uusi versio, eli tieto päivitetään vanhan päälle.

etlstamp-arvo on virallisesti lähdekannassa merkkijono, joka esittää aikaleima muodossa (Postgres-tyyliin) YYYYMMDDHH24MISS (Javaksi yyyyMMddHHmmSS). Kuitenkin joidenkin merkkijonojen alussa on sana “rerun”, jonka jälkeen numero-osa tulee. Muita merkkijonoja ei esiinny ainakaan tällä hetkellä (2016-09-08).

Väliaikaiseen käyttöön tehdyssä puhdistetussa labradatassa (esim. lab_temp.temp_lab_vsshp_join) täydelliset rividuplikaatit on poistettu ja etlstamp-sarakkeen perusteella riveistä on säilytetty vain uusin. Silti löytyy tekstimuotoisia vastauksia, joilla on sama Tutkimusriviavain ja sama Tutkimus ja sama etlstamp/dt (eli ovat poimiutuneet samassa poiminnassa, eli olleet samaan aikaan olemassa lähdejärjestelmässä), mutta eri tulos. Näille ei voida oikein mitään, onneksi näitä on vain yhtä testiä eikä tulos ole numeerinen

Vuosi 2004

```
select extract(year from datetime3), count(*)
from stage_labdw.labdw_transform
where source_table = 'labdw_tutkimus_2004'
group by extract(year from datetime3)
order by extract(year from datetime3);

-- 1997 1
-- 1999 2
-- 2000 6980
```

```
-- 2001 506401
-- 2003 651
-- 2004 788270
--  28
```

Miranda ODS Laakitys

Author: anna.hammais@tyks.fi & mikhail.stepanov@tyks.fi

Laakitys_laakemaarays_ods ja Laakitys_laakemaarays_hist_ods

Näiden taulujen suhde on se, että ensimmäinen määräyksestä luotu tieto on aina normaalitaulussa ja kaikki sitä seuraavat ovat hist-taulussa. Näyttääkin siltä, että alkuaika on normaalitaulun rivillä aikaisempi kuin muilla riveillä.

Kannattaa aina käyttää näiden taulujen unionia, johon on merkitty lisäsarake, joka kertoo, kummasta taulusta data on peräisin.

Ryhma_historia_numero vs. historia_numero

Ryhma_historia_numero kuvaa yhden lääkemääräyksen historiaa. Yksi ryhmä_historia on yksi lääkemääräys, joka näkyy yhtenä rivinä lääkelistalla. Historia_numero taas on yhden muokkauskerran tieto, joka näkyy Uranuksessa lääkemääräyksen historiassa yhtenä rivinä.

Rivi_numero

Rivi_numero kertoo historia_numero'n järjestyspaikkaa ryhma_historia_numero'ssa. Nyt taulussa kaikki rivit ovat '0' (nolla rivejä). Jos '0' (nolla rivi) on hist taulussa, se on peruttu.

rivi_numero	hist	nyt
>0	2098461	0
0	2838557	3739729

nolla rivit:

peruttu	hist	nyt
0	0	3610680
1	2834674	123590

Periaatte: jokainen uusi validi historia ryhmässä saa seuraavan numeron.

On olemassa tuplikaatti rivejä (ei peruttuja, noin 600 historiaa) joilla on sama rivinumero ja sama historia numero. Otetaan rivi jolla on isompi paivityshetki_s.

Jos joku rivi on poistettu (virheellinen rivi:poisto_syy_selite merkitty 'virhe' jne) => seuraava uusi historia saa sen rivi_numeron. Tämä ei toimi kuin historia on virheellinen, mutta merkitty lopetus tai muutos sarakkeella. Silloin uusi rivi saa seuraavan rivinumeron.

Peruttu

Jos rivi on peruttu = 1, lähes aina samalla historia_numerolla on olemassa toinen rivi, joka ei ole peruttu. Voidaan ajatella, että peruttu = 1 rivit poistetaan myöhemmästä tarkastelusta kokonaan. Tässä poistuu myös kourallinen historioita kokonaan, mutta tämä lienee ok, koska ovat ehkä virheitä.

Jos nyt taulussa ryhma_historia_numero'n kaikki rivit ovat peruttuja, niin tälle numerolle löytyy aina ei peruttu rivi hist taulussa (sen numero tietysti on isompi kuin 0).

Maarayksen_tila_selite

Jos ryhma_historia_numerolla jokin rivi on peruttu = 0 & maarayksen_tila_selite = 'Mitatoity', yleensä ei löydy samasta ryhmästä toista riviä, jolla peruttu = 0 and maarayksen_tila_selite = 'Normaali'. Poikkeuksia on 35 historiaa kaikkiaan. Ovat ehkä virheitä. Poistetaan nämä historiat.

Mitätöity rivi ei voi olla peruttu. Tämä tarkoittaa, että mitätöinnin jälkeen historiaan ei enää tule uusia rivejä, vaan sitten avataan uusi historia.

Otetaan pois koko historia jos sen joku rivi on Mitätöity

Annostelusta

Keskeiset sarakkeet riippuvat siitä, mikä on arvona sarakkeessa annostelu_tyyppi_selite.

```
Erillisen ohjeen mukaan -- annostelu_eril_ohje
Kertalääke              -- annostelu_annos
Säännöllinen            -- annostelu_annos ja annostelu_saan_pvm_annos
Tarvittaessa            -- annostelu_tarv_annos ja annostelu_tarv_pvm_max_annos
```

Kaikissa tapauksissa mukaan on hyvä ottaa annostelu_annos_yksikko.

Lopetus, poisto ja muutos

Lopetus_syy_selite, poisto_syy_selite ja muutos_syy_selite kertovat mahdollisista muista kuin "annos muutos"-tyypisistä lääkemääräyksen muokkauksista. "Kirjattu väärälle potilaa", "Virhekirjaus", "Virheellinen toteuttamato" takoittavat virhellisiä rivejä.

Maarayksen_tila_selite ja poisto_aika_pvm

Jos virhe merkitty poisto_syy_selitteeksi, niin löytyy poisto_aika_pvm. Jos virhe on muutos tai lopetus, niin poisto_aika_pvm ei yleisesti ole:

	poisto_aika	
mpl	FALSE	TRUE
FALSE	0	2
lopetus	5731	17
muutos	4594	51
muutos, lopetus	85	2
poisto	0	50736
poisto, lopetus	0	436
poisto, muutos	0	113
poisto, muutos, lopetus	0	14

Samaa pätee maarayksen_tila_selittelle:

	maarayksen_tila_selite	
mpl	Mitatoity	Normaali
FALSE	2	0
lopetus	17	5731
muutos	51	4594
muutos, lopetus	2	85
poisto	50736	0
poisto, lopetus	436	0
poisto, muutos	113	0
poisto, muutos, lopetus	14	0
<NA>	7911	8624192

Päätelmä

Poistetaan kaikki perutut rivit. Poistetaan historiat, joissa on mitätöity rivi. Tämä siksi, että on noin 35 historiaa, joissa on sekä mitätöity että normaali rivi.

Ryhma_historian muihin riveihin ei vaikuta yhden jäsenhistorian mitätöinti.

Jos samalla historialla on kaksi (validi ja ei peruttu ei mitätöity) rivejä otetaan se millä on isompi paivityshetki_s.

Laakitys_resepti_ods

Huomioitava, että laitetaan hakuehdoiksi

```
TILA_SELITE != 'LUONNOS' AND MITATOINTI_SYY_KOODI is null;
```

tai

```
  TILA_KOODI != 0  AND MITATOINTI_SYY_KOODI is null;
```

Jos resepti on määrätty TYKSissä ja mitätöity myöhemmin VSSHP:n ulkopuolella, siitä emme saa tietää.

Lääkkeen annostelu:

1. laakitys_resepti_ods.annos
2. laakitys_resepti_ods.kestoaika ja laakitys_resepti_ods.kestoaika_yksikko_selite: jos ei anneta tarkkaa kappalemäärää reseptissä vaan esim. “vuoden tarve”, jolloin apteekki laskee kerrallaan tarvittavan potilaalle myytävän määrän tämän perusteella

Laakitys_antokirjaus_ods

Sarakkeessa poisto_syy_selite voi olla null, pelkkä välilyönti tai tekstiä. Pelkkä välilyöntikin tarkoittaa yhden Uranuksesta tarkistetun potilaan perusteella sitä, että rivi on poistettu, eli poisto on tapahtunut. Hyväksytään rivit, joilla

```
poisto_syy_selite is null
```

Tässä myös ryhmä_historia_numero ja historia_numero kuten lääkemääräyksissäkin, mutta eivät ilmeisesti samassa merkityksessä. Jokaisen laakemaarays_numeron alla (=antokirjaukset, jotka liittyvät yhteen määräykseen) on aina vain yksi ryhmähistoria, mutta voi olla monta tai yksi historia. Määräyksellä voi siis olla monta antokirjausta, joilla kaikilla eri historia tai kaikilla sama historia.

Näyttää siltä, että mitätöityihin määräyksiin ei liity antokirjauksia, eli ne kenties poistetaan odsista automaattisesti.

Vertailtu kahta tällaista tapausta. Molemmat liittyivät normaaliin perumattomaan määräyksen, joka oli ryhmähistoriansa ainut. Ei selitystä, miksi toisessa monta historiaa ja toisessa ei.

Katsottu Uranuksesta, näyttävät siellä samalta eli voimassa olevalta tiedolta jokainen rivi, joten oletetaan näin.

Ongelmat

```
# maaraykset
   laakitys$VALUE           <- laakitys$VAIKUTTAVA_AINE
    laakitys[as.character(VALUE) == "-" |as.character(VALUE) == " "|is.na(VALUE)]$VALUE <-
        laakitys[as.character(VALUE) == "-" |as.character(VALUE) == " "|is.na(VALUE)]$KAUPPANIMI
```

```
# reseptit
    laakitys_resepti$VALUE                                  <- laakitys_resepti$VAIKUTTAVA_AINE
      laakitys_resepti[is.na(VALUE)]$VALUE                    <-
          laakitys_resepti[is.na(VALUE)]$MUU_LAAKE_TIEDOT
      laakitys_resepti[as.character(VALUE) =="-"]$VALUE     <- laakitys_resepti[as.character(VALUE)
          == "-"]$KAUPPANIMI
      laakitys_resepti[as.character(VALUE) == " "]$VALUE    <- laakitys_resepti[as.character(VALUE)
          == " "]$EX_TEMPORE_VALMISTUSOHJE
```

Puuttuu annos

Liian pitkä kuuri

Naming Conventions

Document author arho.virkki@tyks.fi, anna.hammais@tyks.fi

Source systems and stage

- **No language transformations on the loading phase.** Stage tables use the names and language of the source systems.
- Table column names should not contain accented chatracters.

Fact schema 'Asiakanta'

- The fact chema column names are in Finnish, excluding those fields that do not have established translations.
- Factor levels that KTP computes are in Finnish (for example leikkaus, laakitys, sadehoito) but without accented characters.

Exported data

- The column names and factors are translated into English, if needed because of the international audience.

Opera semantics

Document author: anna.hammais@tyks.fi

DM tables toimintatilasto.dmf_leikkaus and toimintatilasto.dmf_toimenpide

Join tables by

```
DMF_Leikkaus.H_tmp_act_id = DMF_Toimenpide.tmpActID
```

Both tables contain several references to Yhteinen.DMD_osasto. The values are the same for all these. However, for references to Yhteinen.DMD_osasto_opera, the references in DMF_leikkaus are more complete, whereas DMF_toimenpide sometimes has key value 0 (representing unknown ward/osasto).

Valid

toimintatilasto.dmf_toimenpide-taulun sarake lippu_tmpValid tarkoittaa, onko operan toimenpidekerta (=leikkaus) valid vai ei. Sarake lippu_TOIMENPIDEValid taas tarkoittaa, onko kyseinen toimenpide (leikkauskerran sisällä) valid vai ei. Jos joinaa dmf_toimenpide ja dmf_leikkaus, näissä olevat lippu_tmpValid-sarakeet ovat aina samat. Toimenpiteen kohdalla Molempien (TOIMENPIDEValid ja tmpValid) pitää olla valid.

Hoitopaatos ja varaus

Use columns “hoitopaatospvm” and “tallentaminenvarauspvm” of the following query:

```
select o.*, to_date(s.tallentaminenvarauspvm, 'YYYY-MM-DD') as tallentaminenvarauspvm
from stage_dw.mv_opera_leikkaus_toimenpide as o
inner join stage_dw.dm_dmf_leikkaus as l on o.dm_dmf_leikkaus_id = l.dm_dmf_leikkaus_id
left outer join stage_dw.dw_s_reservation as s on l.h_act_id_varaus = s.act_id
and s.valid = 'y';
```

HoitopaatosPvm vs. jonoonAsetusPvm in DM - conflicts.

The dates are sometimes the same, sometimes different. Sometimes jonoonasetus (queue) is before hoitopäätös (treatment decision). Hoitopaatos comes from Uranus, Jonoonasetus comes from Opera.

```
-- OPERA: HOITOPÄÄTÖSPÄIVÄN JA JONOONASETUSPÄIVÄN VERTAILU

-- 131 289 joissa erit
select TOP 100 jonoonAsetusPvm, HOITOPAATOSPVM
from Toimintatilasto.dmf_leikkaus
where jonoonAsetusPvm <> HOITOPAATOSPVM and jonoonAsetusPvm is not null and HOITOPAATOSPVM is not
    null;

-- 163 357 joissa samat
select count(*)
from Toimintatilasto.dmf_leikkaus
where jonoonAsetusPvm = HOITOPAATOSPVM and jonoonAsetusPvm is not null and HOITOPAATOSPVM is not
    null;

-- 0 joissa jonoonAsetusPvm puuttuu
select count(*)
from Toimintatilasto.dmf_leikkaus
where jonoonAsetusPvm is  null;

-- 123 844 joissa HOITOPAATOSPVM puuttuu
```

```
select count(*)
from Toimintatilasto.dmf_leikkaus
where HOITOPAATOSPVM is null;

-- 32 520 joissa jonoonasetus ensin
select count(*)
from Toimintatilasto.dmf_leikkaus
where jonoonAsetusPvm < HOITOPAATOSPVM;

-- 98 769 joissa hoitopaatos ensin
select count(*)
from Toimintatilasto.dmf_leikkaus
where HOITOPAATOSPVM < jonoonAsetusPvm;
```

Diagnoosit ja palvelut

Document author: anna.hammais@tyks.fi

Diagnoosin luontihetki (diagnoosi_ods.luontihetki_s) saattaa olla esim. sädehoidon käynneille jo paljon ennen käyntiä. Ilmeisesti luontihetki on se, jolloin diagnoosi on liitetty varaukseen, ja diagnoosi siirtyy varaukselta käynnille?

Näkymä stage_uraods.mv_diagnoosi kokoaa yhteen tapahtuma- ja pitkäaikaisdiagnoosit. Syy ja oire on jaettu omille riveilleen.

	on_syy
syy	1 (true)
oire	0 (false)

	on_tapahtuma_dgn
tapahtuma dgn	1 (true)
pitkäaikaisdgn	0 (false)

Potilaskertomus_teksti Miranda ODSissa

Document author: anna.hammais@tyks.fi

Taulut uraodsissa

```
potilaskertomus_ods
potilaskertomus_teksti_ods
```

Linkitys sarakkeella teksti_numero

To XML or not to XML?

Vanhat tekstit ovat XML-muodossa, uudemmat eivät. Uudemmistakin ovat ne, jotka tulevat viestinä muista järjestelmistä kuten QPatista tai Operasta.

Uudenmuotoiset (ei-XML) tekstit

Tila ja historia_avain

Uudenmuotoisilla teksteillä esiintyy seuraavia tiloja: hyväksytty, poistettu, korjattu, muistiinpano. Näistä käytetään vain hyväksytty-tekstejä, muut hylätään. XML:ssä esiintyy lisäksi tila lukittu. XML-teksteistä käytetään niitä, jotka ovat hyväksytty tai lukittu.

Samassa historiassa on saman tekstin eri muokkausversiot. Yleensä historian uusimman tekstin tila on “hyväksytty”. Joskus se on “korjattu”, jolloin teksti on aina identtinen (teksti_numero on sama eli on vain yksi teksti) kuin sitä edeltävä “hyväksytty” versio. Jos historian uusin teksti on poistettu, samassa historiassa ei ole muita kuin korjattuja tekstejä sitä ennen. On kourallinen historioita, joissa on kaksi hyväksyttyä tekstiä. Nämä saavat molemmat tulla mukaan, koska on ilmeisesti bugi.

Hoitotapahtuma_numero viittaa johonkin seuraavista tauluista:

```
select distinct hoitotapahtuma_tunnus from stage_uraods.potilaskertomus where hoitotapahtuma_tunnus
    is not null;
-- PROSESSITAPAHTUMA
-- OSASTOHOITO
-- VARAUS
-- KÄYNTI
-- LÄHETE
-- JONOVARAUS
```

Hyvin harvoin hoitotapahtuma_numero puuttuu, vaikka teksti onkin uudenmuotoinen.

```
--452
select count(*) from stage_uraods.potilaskertomus as p
inner join stage_uraods.potilaskertomus_teksti as t on p.teksti_numero = t.teksti_numero
where p.hoitotapahtuma_numero is null and teksti !~* '^<';

-- 0
select count(*) from stage_uraods.potilaskertomus as p
inner join stage_uraods.potilaskertomus_teksti as t on p.teksti_numero = t.teksti_numero
where p.hoitotapahtuma_numero is not null and teksti ~* '^<';
```

```
# 505 jotka kuuluvat muuhun kuin omaan historiaansa. Nämä ovat uudenmukaisia tekstejä jotka vain
    ilmeisesti bugin johdosta ilman hoitotapahtuma_numeroa.
# Unohdetaan.
# tila   N
# 1: muistiinpano 457
# 2:     korjattu  48
d1[is.na(hoitotapahtuma_numero)][teksti_numero != historia_avain][, .N, by = tila]
```

XML sisältää OID-koodeja, joista voi joskus päätellä, että kyse on esim. väliotsikosta. Alkuosa 1.2.246.537 tarkoittaa Suomea (246), sosiaali- ja terveydenhuoltoa (537). 537:n tilalla 777 tarkoittaa HL7.

Joissain XML:issä on

```
<local_attr name="TAPAHTUMA-AIKA" value="20110306"
<local_attr name="AIKA" value="20110306"
```

, joka kertoo tapahtuma-ajan ja tallennusajan. Kaikissa tällaista ei ole. Näitä parsittu jonkin verran kokeeksi tauluun stage_uraods.tekstit_dom_local_attr.

TAPAHTUMA-AIKA ei aina ole sama kuin XML-rakenteen sisällä vaihtelevissa kohdissa näkyvä päivämäärä. Samoin AIKA ei aina ole sama kuin potilaskertomus-taulun paivityshetki_s, vaikka loogisesti näin luulisi olevan.

Kuitenkin 17M/18M tekstiä menee niin, että TAPAHTUMA-AIKA löytyy local attribuuteista, ja jos löytyy tekstin joukostakin päivämäärä, se on sama. Noin 700K tekstillä ei ole TAPAHTUMA-AIKAa, ja näistä 17K ei ole AIKAAkaan. Näille käytetään aikaleimana potilaskertomus.paivityshetki_s.

Näyttää siltä, että niillä teksteillä, joiden teksti_numero alkaa numerolla eikä kirjaimella, seuraavista poluista löytyy näkymä, pvm ja yksikkö, jossa palvelutapahtuma on tapahtunut:

```
/section[1]/caption[1] = esim. LKIR (näkymä)
/section[1]/section[2]/paragraph[1]/content[1] = esim. 02.11.2013 (pvm)
/section[1]/section[2]/paragraph[2]/content[1] = esim. LNKIR (yksikkö)
```

Samoin näyttää olevan <section>-alkuisille teksteille (ei aitoa XML:ää.) Nämä saa parsittavaksi kun lisää alkuun XML declarationin eli vaikka

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

Näitä parsittu jonkin verran kokeeksi tauluun stage_uraods.temp_dom_alkuosa.

Kirjaimella alkavat teksti_numerot sisältävät sellaisiakin, joilla pvm ja yksikkö löytyvät eri poluista:

```
/section[1]/section[2]/caption[1] = esim. 29.1.03
/section[1]/section[3]/caption[1] = esim. KEU
```

tai

```
/section[1]/section[2]/paragraph[2]/content[1] = pvm
/section[1]/section[2]/paragraph[1]/content[1] = yksikkö
```

Tekstin aikaleimat

Jos tekstillä on potilaskertomus-taulussa hoitotapahtuma_numero, sillä löytyy käynti ja osastohoito, joilla on omat rakenteiset aikaleimansa. Jos hoitotapahtuma_numeroa ei ole, rakenteisena löytyy potilaskertomus.paivityshetki_s, joka on ilmeisesti tekstin päivitysaika Uranuksessa. Sen kuvittelisi olevan sama kuin tekstin sisältä löytyvä local_attr-lohkon “AIKA”-attribuutin aikaleima (6M rivillä se on eri, 11M sama), eli esim. tekstin tallennushetki, mutta nämä aikaleimat ovat usein eri päiviltä. Jos local_attr-lohkosta löytyy “TAPAHTUMA-AIKA”, tätä kannattaa käyttää.

Kyselyjä

```
select * from potilaskertomus_teksti_ods where TEKSTI_NUMERO = 'PAT13525958112532';
select * from potilaskertomus_ods where TEKSTI_NUMERO = '256596150928114045';
```

Esimerkki tuoreista teksteistä, jotka eivät ole XML-muodossa:

```
select * from potilaskertomus_teksti_ods  t
inner join potilaskertomus_ods k on t.teksti_numero = k.teksti_numero
where extract(year from k.paivityshetki_s)= 2016
and hoitotapahtuma_numero is not null
and rownum < 1000;
```

Tietty näkymä, tuoreet tekstit:

```
select * from potilaskertomus_ods
where nakyma_selite = 'PAT' and extract(year from paivityshetki_s) = 2016;
```

Näkymät, joita taulussa esiintyy:

```
select koodiarvo, koodiarvon_selite,
koodiarvo_voimassa_alkaen_pvm, koodiarvo_voimassa_asti_pvm
from stage_uraods.koodisto
where koodiston_aihe = 'URAVI'
order by koodiarvo;
```

Otsikot, joita tekstissä esiintyy:

```
select * from koodisto_ods where koodiston_aihe in ('CLNTI', 'CLLTI');
```

Bugi

Sama teksti_numero liittyy joskus kahteen potilaskertomus_numeroon, joiden molempien tila on hyväksytty tai lukittu. Tällaisia teksti_numeroita on 154 kpl (2016-11-08). Ilmeisesti bugi. Ainakin joissain näistä tapauksista uudempi tekstiversio sisältää tiedon toimipisteestä (potilaskertomus.toimipiste_koodi & nimi) mutta vanha ei. Näin molemmat versiot näkyvät meille, vaikka ovat sama teksti ja toisesta puuttuu toimipiste.

```
select p.teksti_numero,
extract (year from p.paivityshetki_s), extract(month from p.paivityshetki_s), count(distinct
    p.potilaskertomus_numero)
from potilaskertomus p
inner join potilaskertomus_teksti t on p.teksti_numero = t.teksti_numero
where p.tila in ('lukittu','hyväksytty')
group by p.teksti_numero, extract (year from p.paivityshetki_s), extract(month from
    p.paivityshetki_s)
having count(distinct p.potilaskertomus_numero) > 1
order by extract (year from p.paivityshetki_s) desc,
extract(month from p.paivityshetki_s) desc;
```

QPati data interpretation

Document author: anna.hammais@tyks.fi

Joining henkilotunnus to patologia tables

Joining henkilotunnus from potilas_asia table to patologia_vastaus table happens by:

```
select a.hetu, v.*
from main_dev.patologia_vastaus as v
inner join main_dev.potilas_asia as a on v.potilas_asia_id = a.id;
```

Rare tables

The statement tables with the following names are not transferred from stage_qpati.statementtablecelldata to main(_dev).patologia_taulukot because they are very rarely used and their format is not machine-readable:

tablename	Number of answers
GASO-PCR.tbl	1
Telo-FISH.tbl	1
luuydin ja Plasmasolueristys dg.tbl	1
G-RT-ASO.tbl	1
luuydin dg.tbl	2
Istukkanäyte.tbl	1
G-RT-ASO.2.tbl	1

Näytesarjat

B-näytesarja on patologialla kudosnäyte-sarja. A= ruumiinavaukset, C=sytologiset, D= hammasklinikalla tehdyt kudosnäytteet, E=meetingin laskutus, K=alihankkijan tekemä kudosnäytevastaus, N= neuropatologinen tutkimus, R=alihankkiijan tekemä ruumiinavausvastaus, S= alihankkijan tekemä sytologinen vastaus, V=vainajan säilytyslaskutus, Y= alihankkijan tekemä kudosnäytevastaus.

E-sarjassa on jonkin verran diagnooseja, mutta huomattavasti vähemmän kuin meetingissä oikeasti käsitellyillä B-näytteillä. Tutkimuksina E-sarjassa on lähesyksinomaan '11207 Pt-Meeting'. Taulukkoarvoja ei ole.

V-sarjassa on taulukkoarvoja Vainaja.tbl-taulukossa, mutta ei diagnooseja.

Acked answers

Based on XML data received so far (see table below), it appears that the data contains mainly acked answers. For sample type A, acking seems to have become the norm in 1993, for sample type V in 1995 and for sample type D somewhere in the nineties.

A general rule is to use only acked answers, except in the early years for sample types where code starts with A, D or V.

```
select coalesce(extract(year from sampletaken),extract(year from arrived)),
    substr(samplenumberexternal, 1, 1),
    sum(case when acked is null then 1 else 0 end) as not_acked,
    sum(case when acked is not null then 1 else 0 end) as acked
```

```
from stage_qpati.qpatianswer
  group by coalesce(extract(year from sampletaken),extract(year from arrived)),
      substr(samplenumberexternal, 1, 1)
  order by substr(samplenumberexternal, 1, 1), coalesce(extract(year from sampletaken),extract(year
      from arrived));
```

External and internal customers

Calculations done on 2016-08-11, based on Musti Yksikkörekisteri

“” –coalesce(v.vastaanottaja, v.lahettaja) on selvästi sisäinen 453,537

select count(*) from main_dev.patologia_vastaus as v left outer join stage_musti.yksikkorekisteri as y on split_part(coalesce(v.vastaanottaja, v.lahettaja), ‘’, 1) = y.lyhenne where y.alayksikkotyyppi != ‘ULKOPUOLI-NEN’ –and not coalesce(v.vastaanottaja, v.lahettaja) ~* ‘TKS:’ and v.naytenumero like ‘B%’;

–coalesce(v.vastaanottaja, v.lahettaja) on selvästi ulkoinen 22,010

select count(*) from main_dev.patologia_vastaus as v left outer join stage_musti.yksikkorekisteri as y on split_part(coalesce(v.vastaanottaja, v.lahettaja), ‘’, 1) = y.lyhenne where y.alayksikkotyyppi = ‘ULKOPUOLI-NEN’ –and not coalesce(v.vastaanottaja, v.lahettaja) ~* ‘TKS:’ and v.naytenumero like ‘B%’;

– lähettäjä on todnäk TKS eli ulkoinen 75,857

select count(*) from main_dev.patologia_vastaus as v left outer join stage_musti.yksikkorekisteri as y on split_part(coalesce(v.vastaanottaja, v.lahettaja), ‘’, 1) = y.lyhenne where y.alayksikkotyyppi is null and coalesce(v.vastaanottaja, v.lahettaja) ~* ‘TKS:’ and v.naytenumero like ‘B%’;

– lähettäjä ei todnäk ole TKS, eli on todennäköisemmin sisäinen 123,537

select count(*) from main_dev.patologia_vastaus as v left outer join stage_musti.yksikkorekisteri as y on split_part(coalesce(v.vastaanottaja, v.lahettaja), ‘’, 1) = y.lyhenne where y.alayksikkotyyppi is null and not coalesce(v.vastaanottaja, v.lahettaja) ~* ‘TKS:’ and v.naytenumero like ‘B%’; “‘

Radu

Document author: anna.hammais@tyks.fi

Toimenpideluokituksen koodien tulkinta

- loppuu C = varjoainetutkimus
- loppuu T = kuvantamisohjattu toimenpide
- loppuu D = TT
- loppuu Q = matala-annos-TT
- loppuu G = magneettitutkimus vahvakenttälaitteella
- loppuu F = magneettitutkimus keskikenttälaitteella

Lista ei kattava; kannattaa käyttää hyväksi saraketta tutkimusryhma_selite

Distinguishing non-VSSHP customers

Both the old and new Radu versions contain information about whether the unit that ordered the imaging procedure is a unit of VSSHP or an external unit. They are coded as follows:

Old radu

```
ulkopuolinen_koodi:
E = no, not external ("EI")
U = yes, external ("ULKOPUOLINEN")
```

New radu (DW)

```
onkotutkimusulkopuolinen:
E = no, not external ("EI")
K = yes, external ("KYLLÄ")
M = other? (considered external for now)
```

In the materialized views, they are coded as follows:

```
ulkopuolinen_tutkimus:
0 (int): is not external
1 (int): is external
```

Starting 2014, there is a bug in the Radu program. Whenever something is written in the "Muu pyytäjä" column, the row is marked as external, even though the customer is clearly VSSHP by name, and the "Muu pyytäjä" is, e.g., a TYKS doctor.

We tried a heuristic as follows: If any of the rows of one customer were marked as internal, we considered all of that customer's rows as internal. However, this lead to problems where a customer had 20000 rows as external and only 2 as internal. These would all have been marked as internal even though there was clearly a typo in the two rows.

We considered requiring that 90 % of the customer's rows were internal, but this will not work in the long run, because the bug exists and fake externals are being created fast. Also, it would have excluded too many internals already.

We considered using the rule that if the customer was marked as internal on >= 10% of the customer's rows, the customer is internal. Else, external. Only exceptions are that 'TYKS' is always internal and customers beginning with 'TKS:' are always external.

Conclusion

We received, from Medbit's Hanna Finneman, an extract from "Musti Yksikkorekisteri". We used that as basis, because it lists almost all units and categorizes them to internals and externals. Less than 2000 rows out of over 5M in Radu were left out because their unit wasn't included in the list. Others were successfully categorized. VSSHP Radu data is in materialized view:

```
stage_dw.mv_radu_vsshp
```

Syöpä

Document author: anna.hammais@tyks.fi

Syöpärekisterin käyttämät ICD-koodit syövälle

Ilmoitettavia ovat kaikki C-koodit(, jotka eivät ATC-koodin muotoisia), D00-09, D32-33, D37-D48, D76, lisäksi N87 ja N89-N90.

```
^C..$|^C..\\.|^C.. ?&|[+*]C\\d\\d(\\.\\d+| ?&)?$|D0[0-9]|^D3[23789]|^D4[^9]|^D76|^N8[79]|^N90
```

Talousdatan dokumentaatiota

Document author: anna.hammais@tyks.fi

Miten joinataan suorite_hinnoittelu Opera-dataan tietovarastossa

Tällä hetkellä gradientin ktp-kannasta puuttuu taulu H_Act_TMP_Opera.

```
SELECT top 10 *
  FROM [Varsat_DW_VSSHP].[dbo].[H_Act_TMP_Opera] as TMP
  inner join [Varsat_DW_VSSHP].[dbo].S_Toimenpidekerta_Opera as OP on TMP.tmp_act_id = OP.tmp_act_id
 where op.valid = 'y'
 -- and TMP.businesskey = '00391331'
```

Yleisiä ohjeita talousdataan

Seuraava ohje saatu Pasi Ahomäeltä.

Hei,

Ohessa dokumentointia rajapinnoista: linkitykset taulukuvauksissa source_system_id1 – 5 (tietovaraston datamartissa samat ”suomennettuna” LAHDE_BUSINESSKEY1-5)

Tietovarastosta kannattanee käyttää hinnoittelupuolta: taloustilasto.dmf_tuote_hinnoittelu (tuotepäätöksen otsikkotiedot) taloustilasto.dmf_suorite_hinnoittelu (tuotepäätökseen liittyvät suoritteet)

Kytkös noiden välillä PRODUCT_ID - tietovarasto käyttää sisäisesti h_act_id – h_act_id_tuote mutta pitäisi olla yksi yhteen jos data kunnossa.

Vastauksia: 1. käyntien osalta kokonaishinta = product_id suoritehinnat yhteensä. kyselyesimerkki:

```
select p.product_id,
p.lahde_businesskey3 as KAYNTI_NUMERO,
p.lahde_businesskey1 as LASPA_NUMERO,
sum(s.sis_kokonaishinta) suoritehinta
from taloustilasto.dmf_tuote_hinnoittelu p
join taloustilasto.dmf_suorite_hinnoittelu s on p.product_id=s.product_id
where p.lippu_peruttu=0
and s.lippu_peruttu=0
and s.POISTETTU_LAHTEESSA=0
and s.lippu_laskutettava=1 --kaikkia suoritteita ei laskuteta.... mitä haetaan?
and p.LAHDE_BUSINESSKEY3='897598270442399754' --käyntinumero
group by p.product_id,p.lahde_businesskey3,p.lahde_businesskey1
```

Hoitojaksoissa product_id saattaa koostaa useita Oberonin osastohoitojaksoja yhteen (LASPA_NUMERO / LASKUTUSPAATOS_MAKSAJA).

Osastohoitojaksotason hinta ei ole meillä seurattava asia mutta ainakin teoriassa sille pitäisi pystyä summaamaan hinta:

```
kokonaiskustannus =
Hoitopäivät (osastojakso = dmf_suorite_hinnoittelu.source_system_id3)
+ oberon toimenpiteet (osastojakso = dmf_suorite_hinnoittelu.source_system_id2)
+ oberon lisälaskutus (osastojakso = dmf_suorite_hinnoittelu.source_system_id2)
+ oberon lääkkeiden antokirjaukset (kohdistetaan, MAIN_SERVICE_ID = oberon jakson atlas SERVICE_ID)
+ muista järjestelmistä tuotavat välisuoritteet (kohdistetaan, MAIN_SERVICE_ID = oberon jakson
    atlas SERVICE_ID)
```

Kannattaa huomioida, että hoitojaksot on pilkottu datamartissa hoitopäivätasolle. yhtä hoitojaksoa esiintyy 1-n rivillä.

2. ```
   select p.product_id,p.lahde_businesskey3 as KAYNTI_NUMERO,p.lahde_businesskey1 as
       LASPA_NUMERO,sr.suoritekoodi, sr.selite,sum(s.sis_kokonaishinta) suoritehinta
   from taloustilasto.dmf_tuote_hinnoittelu p
   join taloustilasto.dmf_suorite_hinnoittelu s on p.product_id=s.product_id
   ```
   ```

```
join taloustilasto.dmd_suorite sr on s.fk_dmd_suorite=sr.PK_DMD_SUORITE
where p.lippu_peruttu=0
and s.lippu_peruttu=0
and s.POISTETTU_LAHTEESSA=0
and s.lippu_laskutettava=1 --kaikkia suoritteita ei laskuteta.... mitä haetaan?
and p.LAHDE_BUSINESSKEY3='897598270442399754' --käyntinumero
group by p.product_id,p.lahde_businesskey3,p.lahde_businesskey1,sr.suoritekoodi, sr.selite
```

3. Äkkiseltään itsenäisistä tulee mieleen myyntipalvelut….potilas ei ole VSSHP:ssa hoidossa. Toisekseen löytyy kohdistumattomia, mitkä eivät ole kohdistuneet VIELÄ… esim. potilas on käynyt labrassa mutta tulossa hoitoon vasta myöhemmin
4. Service_id ja service_part_id

Toivottavasti vastasin kysymyksiin edes sinne päin. Voidaan hyvin istua alas ja penkoa tarkemmin jos tarvetta?

-Pasi

MYNLA

Connection stringit:

```
jdbc:oracle:thin:@mop2.vsshp.net:1521/vsmynla
jdbc:oracle:thin:@mop2.vsshp.net:1521/vsmyntar
```

- vsmyntar oli operatiivinen kanta.
- vsmynla on raportointi-/arkistokanta.

vsmynla sisältää kaiken historian ja vsmyntar 2 viimeistä käyttövuotta.

Tietokantakuvaus löytyy: `\\pinta.vsshp.net\KTP\Ohjeet\Mynla\mynla5_0_jarjestelmakuvaus.doc`

Varauskirja

Document author: anna.hammais@tyks.fi

Oberon varauskirja queries in Oracle MDODS:

```
ALTER SESSION SET current_schema = mdods;
```

VARAUSKIRJA

```
select * from varauskirja where rownum < 100;

-- 6497 (2016-11-30)
select count(*) from varauskirja;

select varauskirja_numero, toimipiste_koodi, toimipiste_nimi,
resurssi_koodi, resurssi_selite,
erikoisala_koodi eala_koodi, erikoisala_selite eala_selite,
voimassaolo_alkuaika, voimassaolo_loppuaika,
voimassa, palvelu_tunnus, oletus_valilehti, varauskirja_resurssi_selite
from varauskirja
where toimipiste_koodi = 'URO';
```

VAPAA_AIKA

```
select * from vapaa_aika where rownum < 100;

-- 20 835 608 (2016-11-30)
select count(*) from vapaa_aika;

select a.vapaa_aika_numero, a.varauskirja_numero, a.varaustyyppi_koodi, a.varaustyyppi_selite,
a.sidottu, a.sitomisen_syy_selite, a.alkuhetki_s, a.loppuhetki_s,
a.vapaita_lkm, a.luontihetki_s
from vapaa_aika a
inner join varauskirja v on a.varauskirja_numero = v.varauskirja_numero
where v.toimipiste_koodi = 'URO' and rownum < 100;
```

VARAUSKIRJA_RIVIT

```
select * from varauskirja_rivit where rownum < 100;

-- 265 781 (2016-11-30)
select count(*) from varauskirja_rivit;

select r.varauskirja_rivi_numero, r.varauskirja_numero, r.tyyppi,
r.alkuaika, r.loppuaika, r.varaustyyppi_koodi, r.varaustyyppi_selite,
r.lkm, r.kesto, r.varausoikeus, r.viikonpaiva, r.voimassaolo_alkuaika, r.voimassaolo_loppuaika,
r.ohjelma, r.ylim_var_oik, r.valmis, r.kayttoonotto,
 r.sidottu, r.sitomisen_syy, r.sitomisen_syy_selite, r.sitomisen_lisatietoja
from varauskirja_rivit r
inner join varauskirja v on r.varauskirja_numero = v.varauskirja_numero
where v.toimipiste_koodi = 'URO';
```

VARAUS

```
select * from varaus where rownum < 100;

select va.var_numero,
-- k.varauskirja_numero,
va.potilasnumero, va.paavar_numero, va.paivakirurgia,
va.vo_toimipiste_koodi, va.vo_toimipiste_nimi, va.res_tyyppi_selite,
va.res_koodi, va.res_selite,
va.palvelu_tunnus, va.varaustyyppi_koodi, va.varaustyyppi_selite, va.varaushetki_pvm,
va.varaushetki_s, va.eala_koodi, va.eala_selite,
va.alkuhetki_pvm, va.alkuhetki_s, va.loppuhetki_pvm, va.loppuhetki_s, va.ylim,
va.kayntityyppi_koodi, va.kayntityyppi_selite, va.kayntityyppi_tarkenne_selite,
va.peruttu, va.perumisen_syy_selite, va.siirretty, va.siirron_syy_selite,
va.ilmoittautunut, va.saapumatta_jaanyt, va.luontihetki_s,
 va.jonottamisen_syy_koodi, va.jonottamisen_syy_selite,
va.odottamisen_syy_koodi, va.odottamisen_syy_selite
from varaus va
inner join varauskirja k on va.varauskirjan_numero = k.varauskirja_numero
where k.toimipiste_koodi = 'URO' and rownum < 10;
```

Data delivery logging

Document author: anna.hammais@tyks.fi

Automated data delivery processes are currently being developed as Kettle jobs that are stored in the following Git repository:

```
ktpgit.vsshp.net:/opt/git/Luovutusprosessit.git
```

Luovutusrekisterin taulujen ohje

Luovutusrekisteriohje

General process description

The development has begun with data to be delivered to Auria biobank. Later added Hercules (Mupet) project where data is delivered to external research environment.

The PostgreSQL schemas used in developing the process are in the *ktp* database:

```
luovutusloki_dev
auria_dev
```

and the schemas to be used in the production process are:

```
luovutusloki
auria
hercules
```

Data delivery in Auria (luovutus) is done through materialized views that are created as selects from existing tables, inner joined with the Auria request SS number list that is valid at the moment when a data delivery round begins.

Data delivery in Hercules is done in principles by the following main steps: 1. SSH Tunnel is created for fetching the "hetu-list" from external research enviroment (192.168.122.105: Mupet2) 2. "hetu-list" is saved under hercules schema in gradient database. 3. Based on to the source table information in table hercules.upload, the kettle job fetching the data from source tables, creates SSH tunnel and upload the data to (192.168.122.105: Mupet2) by using the SQL function func.get_content. 4. Kettle job also keeps luovutusloki up-to-date during the transformation

More information about Hercules can be found from the Kettle job file: /Luovutusprosessit/hercules/download_update_hercules_data_hercules.kjb

A first draft of the luovutusloki (identical to luovutusloki_dev) database tables is at:

```
ktpanalytics.vsshp.net:/opt/shared/KTPDoc/Tietomallit/Autom_luovutusloki/*
```

This is version 2 (2016-05-31):

Kumppani (partner) refers to the data receiver. So far the only receiver of automated data deliveries are Auria and Hercules.

Tapahtumatyyppi (delivery module type) refers to a loggable module in the delivery process. So far, the only ones are "Hetujen haku" (fetching SS numbers from source), "Labrojen luovutus" (lab data delivery) and "Kemokur-tietojen luovutus" (delivery of Kemokur/chemotherapy data).

Luovutuskerta (data delivery process) is a process that (usually) consists of the fetching of SS numbers and the delivery of different data sets for these patients. The fetching, as well as the deliveries, are all listed in the table **Lokitapahtuma** (delivery module).

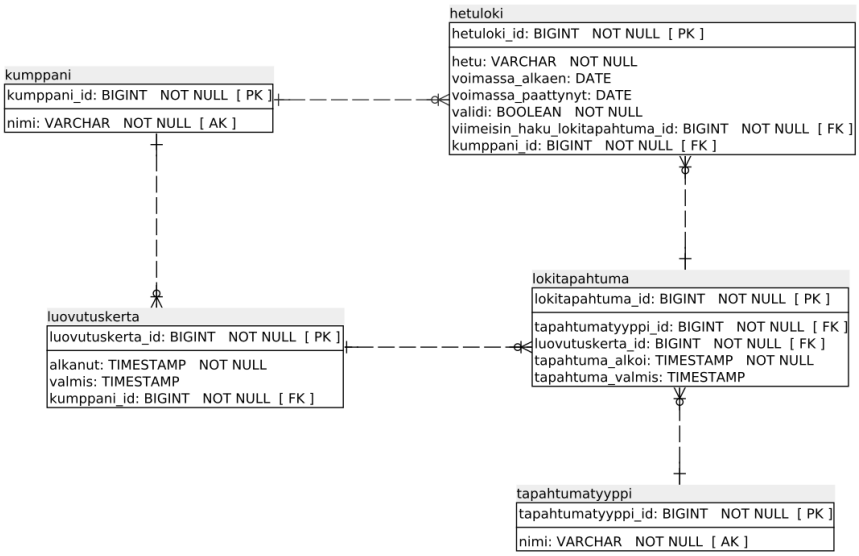


Figure 1: Luovutusloki

The begin and end timestamps of each data delivery process, as well as those of each module, are logged in their respective tables.

The table **Hetuloki** (SSN log) doubles as a SSN list log as well as a listing of currently active SS numbers. Firstly, the SS numbers that have an active consent are marked as validi=TRUE. The SS numbers that have withdrawn their consent, i.e. have been on Auria's request list before but have withdrawn their consent (i.e. have disappeared from Auria's list), remain in the Hetuloki but are marked as valid=FALSE, and the ending date of the hetu validity is set. For an active hetu, the ending date is null.

A module of type "Hetujen haku" (fetching SS numbers) also updates the Hetuloki (SS number log) table:

1. The list of active SS numbers is retrieved from source.
2. Each SS number that is either not found in Hetuloki or is invalid is input into Hetuloki, with the currently ongoing lokitapahtuma_id set as viimeisin_haku_lokitapahtuma_id. This signifies that the SS number was observed in the Auria list at this time.
3. If an SS number already exists in Hetuloki, then the viimeisin_haku_lokitapahtuma_id is updated to match the currently ongoing one.
4. For each new SS number, the voimassa_alkaen is set as the current date.
5. After the insert/update, those SS numbers that were not seen in the latest list, are marked as invalid and the ending date is set. In this case, the viimeisin_haku_lokitapahtuma_id remains unchanged.

Apart from "Hetujen haku", the other data delivery module types (Tapahtumatyyppi) usually just refresh materialized views, i.e. recreate them with the currently active SS number list.

Which person's data was delivered at which delivery

The schema doesn't specify exactly which data items existed for which patient at the moment of a data delivery. However, it is possible to determine the deliveries and their data types in general that a patient was included in. This can be done by the following query:

```
select kerta.alkanut::date as luovutuskerta_pvm,
```

```
tyyppi.nimi as tapahtuma_nimi,
hetut.voimassa_alkaen as hetu_voimassa_alkaen,
hetut.voimassa_paattynyt as hetu_voimassa_paattynyt
from luovutusloki.luovutuskerta as kerta
 -- kaikki kyseisen kumppanin hetut
inner join luovutusloki.hetuloki as hetut on kerta.kumppani_id = hetut.kumppani_id
inner join luovutusloki.lokitapahtuma as tapahtuma on kerta.luovutuskerta_id =
    tapahtuma.luovutuskerta_id
inner join luovutusloki.tapahtumatyyppi as tyyppi on tapahtuma.tapahtumatyyppi_id =
    tyyppi.tapahtumatyyppi_id
where hetut.kumppani_id = (select kumppani_id from luovutusloki.kumppani where nimi = 'Auria')
-- luovutuskerta tapahtunut hetun voimassaoloaikana
and kerta.alkanut between hetut.voimassa_alkaen and coalesce(hetut.voimassa_paattynyt, now())
and hetut.hetu = <some_hetu>
and tyyppi.nimi != 'Hetujen haku';
```

How to use the Kettle jobs

The main job to run is **luovutus.kjb**. It has three parameters:

1. kumppani_nimi (default: Auria)
2. luovutusloki_schema (default: luovutusloki_dev)
3. target_schema (default: auria_dev)

For testing, the defaults are OK, since they refer to the development schemas. For actual data delivery, the parameters should be changed. The schema "luovutusloki" is the actual delivery logging schema, and the kumppani (partner), as well as the partner-specific target_schema should be changed according to the data receiver. Before this, the partner must be added to the luovutusloki.kumppani table, where the column *nimi* is used as the value of this parameter.

Testing

For testing purposes, only a subset of the hetu list can be used. This can be achieved by editing the SQL query in the first step of transformation *fetch_and_insert_hetu_list.ktr*. Adding e.g.

```
"where hetu like '1201%'"
```

allows to fetch a small subset of the list.

Tulevaisuuden kehityssuunnitelmia

Potilastietojen luovutuksista tutkimuksiin ja muihin tarkoituksiin tulee pitää kirjaa. Kliinisen tietopalvelun tietoluovutukset kirjataan kahteen tietokantaskeemaan: luovutusrekisteriin tai luovutuslokiin. Luovutusloki on toistuvien, automatisoitujen tiedonsiirtojen lokitusta varten, ja luovutusrekisteriin kerätään tiedot asiakkaista, tutkimusluvista ja niiden nojalla tehdyistä yksittäisistä tietoluovutuksista. Tietoluovutukset lokitetaan hetukohtaisesti, eli potilaan asiaa tiedustellessa pystytään tarvittaessa tarkistamaan, mihin tutkimuksiin hänen tietojaan on luovutettu, ja onko häntä käsitelty tutkittavana vai kontrollipotilaana.

KTP on itse rakentanut luovutusrekisterin tietokantarakenteen ja java-käyttöliittymän. Käyttöliittymän kautta lisätään tutkimuksia, tutkijoita ja luovutuksia sekä kuvauksia luovutusten tietosisällöstä. Koska KTP:n toiminta on maksullista ja eri projekteihin käytettyä aikaa halutaan seurata, työtuntien seuranta ja projektien hinnat kirjataan myös luovutusrekisteriin.

Tulevaisuudessa luovutusprosessia aiotaan helpottaa automatisoimalla ja lisäämällä luovutusrekisterin käyttöliittymään uusia toiminnallisuuksia. Yhtenä kehitysideana on automaattinen laskutuslistan muodostustoiminto, jolla voidaan valita laskutettaviksi valmiit projektit ja merkitä sitten nämä laskutetuiksi. Tarkoitus on myös automatisoida tilastojen tuotto käyttöliittymän kautta, jolloin olisi entistä helpompaa seurata projektimäärien muutoksia ajassa, eri aineistotyyppien käytön

frekvenssiä ja Tyksin eri toimialueiden aktiivisuutta tietopalvelun käyttäjinä. Näitä asioita on tähän asti tehty SQL-kyselyillä suoraan luovutusrekisterin tietokannasta.

Myös KTP:n projektinhallintaa on toiminnan alkamisen jälkeen kehitetty. Nyt käytössä on yhteinen sähköpostikansio, jossa on alikansiot kullekin asiakkaalle. Käytössä on myös Wekan-taulu, jossa kullekin projektille on oma palstansa, jota KTP:n henkilöstö päivittää kun projekti etenee. Projektien dokumentit ja toimitteet tallennetaan myös omiin projektikohtaisiin, yhtenäistä nimeämiskäytäntöä noudattaviin kansioihinsa. Yritysasiakkaille on olemassa yhtenäiset suomen- ja englanninkieliset sopimuspohjat, jolloin sopimusten teko on sujuvaa.

Luovutusrekisterin käyttö

Document author: anna.hammais@tyks.fi

2015-10-20

Luovutusrekisteri sijaitsee koneella ATDBW61, kannassa *test_db*, skeemassa *luovutusrekisteri*. Tauluja käytettäessä täytyy käyttää alkuliitteenä skeeman nimeä, esim. `luovutusrekisteri.Tutkimus`. Taulukaavio löytyy cell.vtt.fi:

```
\Materiaalit\Anna_Hammais\Tietokantakuvaukset\KTP\
2015-10-16_Tutkimuskohortti_ja_luovutusrekisteri_hetulla_v5.pdf.
```

Sisältö on case-insensitive. Tekstit kannattaa syöttää haluamassaan muodossa, esim. "SHVK", mutta ne löytyvät kyselyllä myös eri casella kirjoitettuna (vrt. "shvk").

Taulut

Sisältöä syötetään järjestyksessä: `Tutkimus` -> `Kohortti` -> `Kohortti_potilaat` -> `Luovutus` -> `Luovutus_tietosisallot`

Seuraavien taulujen pääavain on automaattisesti generoitu rivitunniste, jonka nimi on taulun nimi + *_id*: Tutkimus, Kohortti, Luovutus, Luovuttaja ja Tietosisalto. Siksi esim. tutkimusta syötettäessä ei tarvitse syöttää mitään kenttään *Tutkimus_id*.

Tutkimus: Tutkimukset, joilla on tutkimusnumero ja vastuullinen tutkija. Mikäli tutkimusnumeroa ei ole, sellainen luodaan muotoon KTP_1_2015, jossa 1 on vuoden sisällä juokseva numero ja 2015 on nykyinen vuosi. Näillä numeroilla ei ole muuta merkitystä kuin pitää tutkimusnumero uniikkina. Kun tulee uusi tietopyyntö, ensimmäiseksi selvitetään sille tutkimusnumero ja vastuullinen tutkija, ja tutkimus lisätään tähän tauluun. Sarakkeeseen *Tutkimus_vai_laatu* syötetään 'T' tai 'L' sen mukaan, mitä tutkimusluvassa lukee. Jos tutkimuksella ei ole minkäänlaista lupaa, eli se on viranomaistoimintaa tms., tähän laitetaan 'L'.

Rivin lisäys:

```
INSERT INTO luovutusrekisteri.Tutkimus (Tutkimusnumero, Tutkimus_vai_laatu, Nimi,
Kuvaus, Vastuullinen_tutkija, Vastuullisen_tutkijan_organisaatio,
Kommentit, Lisatty_pvm)
VALUES (varchar, varchar, varchar, varchar, varchar, varchar, varchar, varchar, date);
```

Kohortti: Kohortti on potilasjoukko, joka on mukana tietyssä tutkimuksessa. Kohortteja voi olla samassa tutkimuksessa useita. Kun tutkimus on ensin lisätty, voidaan lisätä siihen yksi tai useita kohortteja. Kohortille annetaan myös yksilöllinen koodi, joka voi olla sama tai samankaltainen kuin tutkimuksen koodi. Sama koodi on suositeltava, jos tutkimukseen kuuluu vain yksi kohortti.

Rivin lisäys:

```
INSERT INTO luovutusrekisteri.Kohortti (Tutkimus_id_fk, Koodi, Kuvaus,
On_kontrolli, Hetut_luovutettu, Kommentit)
VALUES (int, varchar, varchar, bit, bit, varchar);
```

Kohortti_potilaat: Kun potilaat kohorttiin on valittu, heidän hetunsa (tai myöhemmin potilas_id:nsä) syötetään tähän tauluun. Tämän avulla on helppoa myöhemmin hakea, mihin tutkimuksiin tietyn potilaan dataa on käytetty (jos tämä tieto tarvitaan arkistopäällikön sitä kysyessä), tai keitä potilaita tietyssä kohortissa oli mukana (esim. lisäaineiston tekoa varten).

Rivin lisäys:

```
INSERT INTO luovutusrekisteri.Kohortti_potilaat (Kohortti_id_fk, Hetu, Pseudonyymi)
    VALUES (int, varchar, varchar);
```

Luovutus: Luovutus sisältää tiedot luovutustapahtumasta, jossa kohortin potilaista luovutetaan tietoa vastaanottajalle. Vastaanottajan nimi annetaan muodossa "Sukunimi Etunimi". Luovuttajan ID löytyy taulusta **Luovuttaja**. Samalle vastaanottajalle luovutetut tiedostot voi tallentaa yhteen kansioon, vaikka luovutuksia olisi useita. Kansio nimetään laittamalla peräkkäin Tutkimusnumero, yhteyshenkilön sukunimi sekä jokin lyhyehkö tutkimusta kuvaava termi. Luovutetut tiedostot on hyvä nimetä niin, että ne sisältävät tiedoston tuottamisen päivämäärän sekä tutkimuksen koodin.

Rivin lisäys:

```
INSERT INTO luovutusrekisteri.Luovutus(Kohortti_id_fk, Pvm,
    Luovuttaja_id_fk, Vastaanottaja, Kommentit)
    VALUES (int, datetime, int, varchar, varchar);
```

Luovutus_tietosisallot: Tähän tauluun syötetään yksi rivi kutakin kohortista luovutettua tietosisältöä kohden. Jos kohortista 1 on luovutettu patologiaa ja labraa, tähän tauluun syötetään nämä kaksi riviä, ja molemmille vuosiväli (alku ja loppu).

Rivin lisäys:

```
INSERT INTO luovutusrekisteri.Luovutus_tietosisallot
       (Luovutus_id_fk,Tietosisalto_id_fk,Alku_vuosi,Loppu_vuosi)
     VALUES (int), int, int, int);
```

Tarvittaessa syötetään uusia luovuttajia ja tietosisältöjä tauluihin **Luovuttaja** ja **Tietosisalto**.

Software and Core Review Process

Document Authors: arho.virkki@tyks.fi, juha-matti.varjonen@tyks.fi

Best practices of software and code review in KTP

Put the following texts into the beginning of code file comment section and update accordingly.

Software / application review template

```
Review Date: <yyyy-mm-dd>
Software / application (file)name: <name>
Software / application version number: <version>
Responsible developer / designer: <name>
Responsible reviewer: <name>

[Yes/No/Under work]: is the documentation of the software or application saved in the KTP network
    share   (\\atdbw61\KTP\Ohjeet).
- Rejection criterion: no documentation and / or good reason why there is
only a subset of documentation or no official documentation at all.
- Comments: TBA

[Yes/No/Under work]: is the software or application information documented in the ktp wiki
    (http://ktpdoc.vsshp.net/Home)?
- Rejection criterion: no documentation on the ktp-wiki and / or good reason why the software or
    application does not have a
documentation on the ktp-wiki.
- Comments: TBA

[Yes/No/Under work]: is the software or application  stored in the KTP version control system (Git)?
- Rejection criterion: The software or application can not be found in the version control.
- Comments: TBA

[Yes/No/Under work]: does the documentation have information on the software or application's
    production, testing, and
development environments?
- Rejection criterion: no basic information about the application's environments.
- Comments: TBA

[Yes/No/Under work]:  do reviewer know which tools have the software or application been developed
    and with which tools can the
application be further developed?
- Rejection criterion: maintenance and / or upgrading of the software or application required tools
    not mentioned
in the documentation.
- Comments: TBA

[Yes/No/Under work]: are there any examples of the use of modules / classes?
- Rejection criterion: no reason for rejection?
- Comments: TBA

[Yes/No/Under work]: is there any architecture or class diagrams etc. available?
- Rejection criterion: no reason for rejection?
- Comments: TBA
```

Code review template

```
Review Date: <yyyy-mm-dd>
Code (file)name: <name>
Code version number: <version>
Responsible developer / designer: <name>
Responsible reviewer: <name>

[Yes/No/Under work]: is the reviewer able to test the software or application while performing the
    code review?
- Rejection criterion: the software / or application can not be tried and / or tested in practice.
- Comments: TBA
```

```
[Yes/No/Under work]: see how easy is to get "big picture" from the version control of the code and
    folder structure etc.?
- Rejection criterion: the whole code is very difficult and / or impossible to get an overview
    without the author's
verbal instruction.
- Comments: TBA

[Yes/No/Under work]: try to find out is the any mistakes and misunderstandings in the code and that
    the
code is properly commented?
- Rejection criterion: there are large logical errors in the code and not commented.
- Comments: TBA

[Yes/No/Under work]: the variables, functions, methods, classes and modules should be named
    descriptively.
- Rejection criterion: in order to understand the code, reviewer must consult the developer in
    critical manner because
of poor naming
- Comments: TBA

[Yes/No/Under work]:  the method / function should be less than 100 lines long
- Rejection criterion: Multiple and / or non-reasoned over 100 lines long methods or functions if
    not  clearly stated
otherwise why.
- Comments: TBA
```

Servers at KTP

Document author: arho.virkki@tyks.fi

Overview

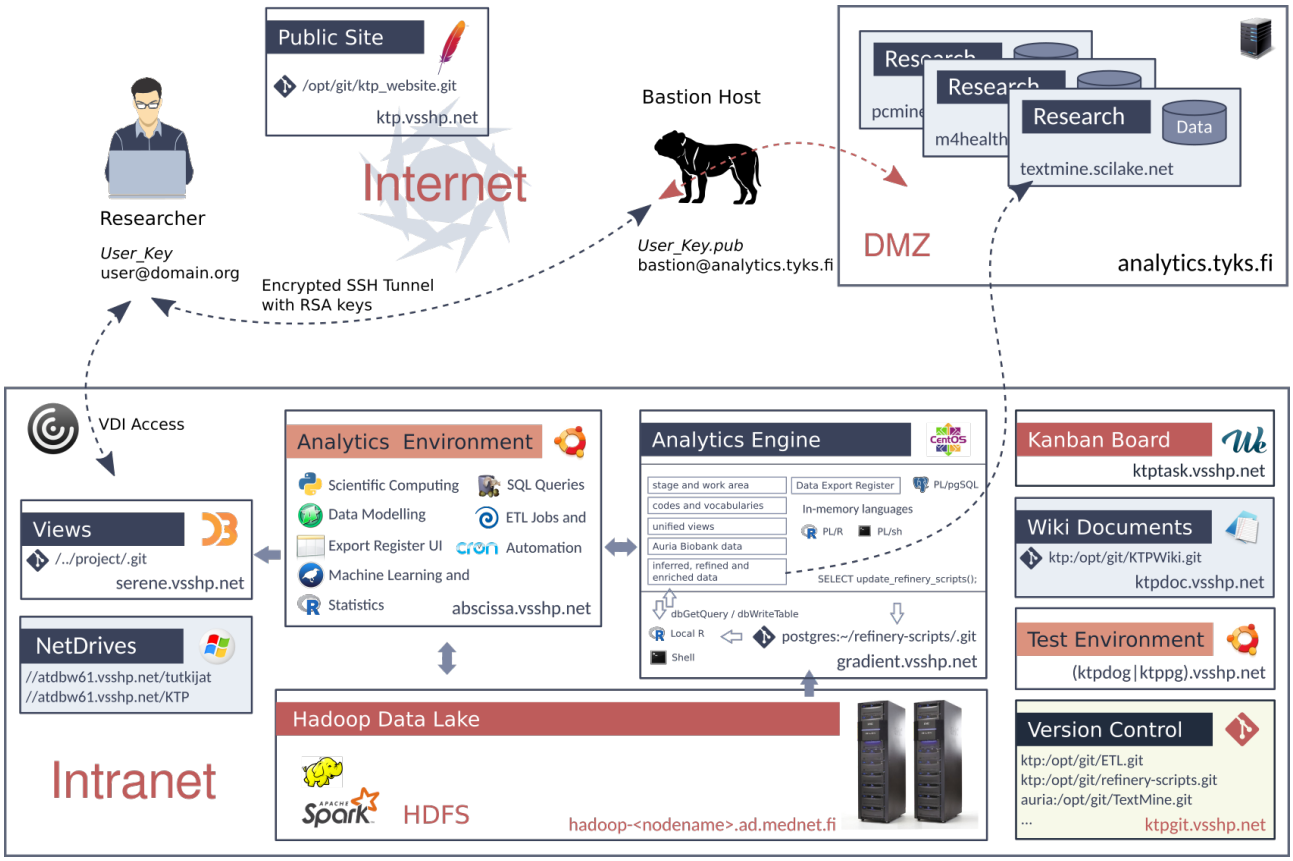


Figure. Server Architecture (Open as pdf or the original svg).

Physical Servers

- analytics.tyks.fi (Dell PowerEdge)
- gradient.vsshp.net (Dell PowerEdge)
- Notes on VM setup

Public Web Server

- Setup and Configuration
- Developing Web Pages

LAMP in Ubuntu 16.04 at transferlog.vsshp.net

Author arho.virkki@tyks.fi and juha.makela@tyks.fi

Installation of apache2:

```
sudo apt-get update
sudo apt-get upgrade

sudo apt-get install apache2
sudo apache2ctl configtest
sudo vim /etc/apache2/apache2.conf

sudo systemctl status apache2.service
```

Installation of mariadb

```
sudo apt-get install mariadb-server
sudo systemctl status mysql.service
```

Then, secure the MariaDB installation and save the chose root password

```
sudo /usr/bin/mysql_secure_installation
```

Finally, to enable root login, issue

```
sudo mysql

USE mysql;
UPDATE user SET plugin='mysql_native_password' WHERE User='root';
FLUSH PRIVILEGES;
```

Granting access from remote hosts

Edit the configuration file

```
sudo vim /etc/mysql/mariadb.conf.d/50-server.cnf
```

and change the bind address into

```
bind-address = 0.0.0.0
```

The *systemctl* wrapper in Ubuntu 16.04 seems to be unreliable in restarting MariaDB:w. Thus, we shut down the deamon manually and start it again with the traditional *service* command:

```
sudo killall mysqld
# Wait until the deamon is dead. Repeat the command if needed
sudo service mysql start
```

Finally, check that mysqld is listening to the public ip (not 127.0.0.1):

```
sudo netstat -ntlup
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 10.150.18.29:3306       0.0.0.0:*               LISTEN      27891/mysqld
...
```

Test the connection with e.g.

```
mysql -h auriatieto.vsshp.net -u bbcrm -p
```

Note: MySQL and MariaDB users are of the form 'username'@'host'. For example, jane@domain1 and jane@domain2 are different users. Users of the form 'username'@'%' can access the database from any machine.

Installing PHP 7.1

```
sudo apt-get install -y python-software-properties
sudo add-apt-repository -y ppa:ondrej/php
sudo apt-get update -y
sudo apt-get install -y php7.1 libapache2-mod-php7.1 php7.1-cli \
     php7.1-common php7.1-mbstring php7.1-gd php7.1-intl php7.1-xml \
     php7.1-mysql php7.1-mcrypt php7.1-zip
```

Myphpadmin

```
sudo apt-get install phpmyadmin
```

Setting up Ubuntu 16.04 Server with GUI

Document Author: arho.virkki@tyks.fi

Installing Beeline Hive client

```
sudo mkdir /opt/hive
cd /opt/hive
tar -zxvf /home/ktp/Downloads/hive-1.1.0-cdh5.8.4.tar.gz
tar -zxvf home/ktp/Downloads/hadoop-2.6.0-cdh5.8.4.tar.gz

sudo sh -c "cat << EOF > /opt/hive/beeline.sh
#!/bin/bash

# Hive startup script
# Sami Porokka, Arho Virkki 2017

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/opt/hive/hadoop-2.6.0-cdh5.8.4
/opt/hive/hive/bin/beeline -u jdbc:hive2://hadoop-c01-cn03.ad.mednet.fi:10000/$1;user=$2;password=$3
EOF

sudo chown -R ktp:ktpshared /opt/hive/
sudo ln -s /opt/hive/beeline.sh /usr/local/bin/
```

Finally, archive installation packages

```
mv hive-1.1.0-cdh5.8.4.tar.gz hadoop-2.6.0-cdh5.8.4.tar.gz hive_jdbc_2.5.18.1050.zip\
  /opt/ktp/shared/Hadoop_libraries/
```

Installing latest version of R

Follow the instructions at https://cran.r-project.org/bin/linux/ubuntu/

Summary of the installation procedure

Add Swedish CRAN mirror to apt sources

```
sudo sh -c "echo 'deb https://ftp.acc.umu.se/mirror/CRAN/bin/linux/ubuntu xenial/' \
> /etc/apt/sources.list.d/r-cran.list"
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 51716619E084DAB9
```

Update and upgrade the system before installing R

```
sudo apt-get update
sudo apt-get upgrade
```

Install R with development tools

```
sudo apt-get install r-base r-base-dev r-cran-codetools r-cran-matrix r-cran-kernsmooth
```

Mounting Folders with Autofs

Mounting Windows shares (e.g. *//atdbw61.vsshp.net/KTP* Folder to */cifs/atdbw61/KTP*) to Linux simplifies certain workflows. This can be done with the Autofs utility.

Example configuration

/etc/auto.master

```
<lots of lines>
#
+auto.master
/mnt/atdbw61 /etc/auto.atdbw61 --timeout=60 --ghost
/mnt/vsshp /etc/auto.vsshp --timeout=60 --ghost
/mnt/gradient /etc/auto.gradient --timeout=60 --ghost
```

Timeout indicates the amount of time before trying to unmount the drive. The default values is set with *dismount_interval = 300* in *autfs.conf*. The ghost option creates dummy directory placeholders for unmounted drives.

/etc/auto.atdbw61

```
# User id 1000 = 'ktp', Group id 1004 = 'ktpshared' KTP
-fstype=cifs,sec=ntlmssp,iocharset=utf8,uid=1000,gid=1004,\
file_mode=0660,dir_mode=0770,credentials=/root/atdbw61 ://10.150.12.104/KTP
```

/etc/auto.vsshp

```
# User id 1000 = 'ktp', Group id 1004 = 'ktpshared'

jarjestelmat -fstype=cifs,sec=ntlmssp,iocharset=utf8,uid=1000,gid=1004,\
file_mode=0660,dir_mode=0770,credentials=/root/svcl_ktplukija ://vsshp.net/jarjestelmat/

arkisto -fstype=cifs,sec=ntlmssp,iocharset=utf8,uid=1000,gid=1004,\
file_mode=0660,dir_mode=0770,credentials=/root/svcl_ktplukija ://vsshp.net/arkisto/

yhteinen -fstype=cifs,sec=ntlmssp,iocharset=utf8,uid=1000,gid=1004,\
file_mode=0660,dir_mode=0770,credentials=/root/svcl_ktplukija ://vsshp.net/yhteinen/
```

/etc/auto.gradient

```
raw -fstype=cifs,sec=ntlm,iocharset=utf8,uid=1000,gid=1000,\
credentials=/root/sharecreds ://gradient.vsshp.net/share/raw
```

/root/atdbw61

```
username=KTP
domain=WORKGROUP
pass=<Passwd Here>
```

/root/sharecreds

```
username=share
password=<Passwd Here>
```

/root/svcl_ktplukija

```
username=svcl_ktplukija
domain=vsshp
pass=<Passwd Here>
```

Activating the new mount

```
sudo systemctl reload autofs
sudo systemctl status autofs
```

or

```
sudo systemctl restart autofs.service
```

Now, if the user belongs to the **ktpshared** group, the *KTP* folder can be accessed with

```
 cd /mnt/atdbw61/KTP/
```

and the network drive will be automatically mounted (if it was not already attached), and unmounted after a given period of time.

Installing and Using VNC Server

VNC is very robust remote desktop protocol which works smoothly from Linux, Max OS X and Windows alike. For setting up the service, see VNC Server Sertup for Ubuntu (there are also instruction for CentOS 7.

To simplify things, we further add the following files under */etc/skel/* to provide templates for **startvcn.sh** and **stopvcn.sh** commands and **xstartup** configuration file for each individual user. These files should be then edited according to personal preferences.

/ect/skel/bin/startvnc.sh

```
#!/bin/bash
vncserver -depth 24 -geometry 1280x800 -localhost -SecurityTypes VncAuth
```

/ect/skel/bin/stopvnc.sh

```
#!/bin/bash
echo "Sending SIGHUP signals to Xvnc4 instances"
killall Xvnc4 -user $USER
```

/etc/skel/.vnc/xstartup

```
#!/bin/sh

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources

# The default configuration for ktpanalytics.vsshp.net
vncconfig -iconic &
startxfce4

# Close also the vnc server if the user logs out
vncserver -kill $DISPLAY
```

Tab-key (e.g. bash-autocompletion): In case that tabular key does not work properly over vnc, edit the file */etc/xdg/xfce4/xfconf/xfce-perchannel-xml/xfce4-keyboard-shortcuts.xml* and change the line

```
<property name="&lt;Super&gt;Tab" type="string" value="switch_window_key"/>
```

into

```
<property name="&lt;Super&gt;Tab" type="empty"/>
```

For details, see http://ubuntuforums.org/archive/index.php/t-1771058.html

Installing Mate Desktop

FXCE Desktop is usually sufficient for all routine tasks, but also Mate (a Gnome 2 fork) can be used. For details, see: http://itsfoss.com/install-mate-desktop-ubuntu-14-04/

Installing X2go remote desktop client

See X2Go_setup.

Extra command line components

```
sudo apt-get install tree p7zip
```

Extra GUI components

```
sudo apt-get install gedit
```

MDCharm Markdown editor

```
sudo apt-get install libhunspell-dev
sudo dpkg --install KTPDoc/Infra/bin/MdCharm/mdcharm_1.2_amd64.deb
```

Git version control

```
sudo apt-get install git

git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
git config --global core.editor "nano"
git config --global push.default simple
```

Change the author when the system is in production

```
git config --global author.name "Arho Virkki"
git config --global user.email "arho.virkki@vtt.fi"
```

R language and tools

See *KTPDoc/Infra/notes/R/*.

PostgreSQL command line client

Since we need PostgreSQL client for 9.4, let's add the official repository also on the analytics server:

```
sudo su -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main" >
    /etc/apt/sources.list.d/pgdg.list'
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo apt-get update
sudo apt-get install postgresql-client-9.4
```

If PostgreSQL was set up properly, it can now be accessed from command line with

```
psql -h ktppg.vsshp.net -U ktp -d postgres
```

PgAdmin3 GUI for PostgreSQL

```
sudo apt-get install pgadmin3s
```

Squirrel SQL (Generic DB client)

Download the Squirrel SQL installer from http://squirrel-sql.sourceforge.net/#installation and run the Java arhive

```
sudo java -jar squirrel-sql-3.7-standard.jar
```

Choose suitable plugins (e.g. multi source) and finally install the program under */usr/local/squirrel-sql-3.7*.

```
sudo ln -s /usr/local/squirrel-sql-3.7/squirrel-sql.sh /usr/local/bin/
```

To increase the memory available to Squirrel and to avoid Java out of memory errors, edit the */usr/local/squirrel-sql-3.7/squirrel-sql.sh* file and change the default -Xmx256m at the end of the file to something larger, e.g. -Xmx2048m.

Installing PostgreSQL Driver for Squirrel SQL

Download the appropriate driver for Java 7 from http://jdbc.postgresql.org and place it under */usr/local/lib/java/* folder:

```
sudo mkdir /usr/local/lib/java
sudo chown ktp: /usr/local/lib/java/
Downloads/postgresql-9.4-1204.jdbc41.jar /usr/local/lib/java/
```

Then open the Squirrel SQL Client and register the driver from graphical menus.

Setting up printer at T-Hospital room E664

Start the printer configuration tool

```
system-config-printer
```

And choose the following settings

```
Description:     HP LaserJet 4250
Location:        HP-LaserJet-4250 @ T-Hospital 6. floor room E664
Device URI:      hp:/net/hp_LaserJet_4250?ip=10.145.56.25
Make and Model:  HP LaserJet 4250 PostscriptBN (recommended)
```

(The tool will automatically pick them when when given the device ip address 10.145.56.25)

Setting up TYKS Email on Thunderbird

To connect to the TYKS Email, use the following configuration:

```
Your name:     [Full Name]
Email address: [full.name@tyks.fi]
Password:      [workstation password]

Incoming: IMAP outlookws.vsshp.net Port: 143  SSL: STARTTLS  Authentication: NTLM
Outgoing: SMTP atms03.vsshp.net    Port: 587  SSL: STARTTLS  Authentication: Normal password

Username Incoming: WORKSTATIONUSER (no domain)  Outgoing: WORKSTATIONUSER (no domain)
```

Setting up a Documentation Server

Document Author: arho.virkki@vtt.fi

Installing Gollum Wiki (based on Git and Markdown)

See https://github.com/gollum/gollum for details.

Install Ruby

```
sudo apt-get install ruby ruby-dev make make zlib1g-dev libicu-dev build-essential git
```

Then, use Ruby package manager to install Gollum (this will take a while):

```
sudo gem install gollum
sudo gem install github-markdown
```

Now, create an empty Git repository with *git init –bare*, and test Gollum

```
mkdir WikiTest.git
cd WikiTest.git
git init --bare
gollum .
```

And open browser at http://localhost:4567 to test the repository.

Requiring a password

```
sudo mkdir -p /etc/gollum/
sudo touch /etc/gollum/authentication.rb
sudo chmod og-rwx /etc/gollum/authentication.rb
sudo nano /etc/gollum/authentication.rb
```

and add the following lines to it

```
# Requite authentication
module Precious
  class App < Sinatra::Base
    use Rack::Auth::Basic, "Restricted Area" do |username, password|
      [username, password] == ['ktp', 'tietomylly']
    end
  end
end
```

The launch Gollum with e.g.

```
sudo gollum --port 80 --live-preview --show-all /opt/git/KTPWiki.git --config
    /etc/gollum/authentication.rb
```

Auto-starting the Wiki at system boot

Create a very rudimentary init script with

```
sudo nano /etc/init.d/gollum
```

and add the following lines to it

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          scriptname
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start daemon at boot time
```

```
# Description:       Gollum Wiki
### END INIT INFO

USER="ktp"

case $1 in
start)
sudo -u $USER gollum --port 8080 --live-preview --config /etc/gollum/authentication.rb --show-all
    /opt/git/KTPWiki.git/ &
;;
stop)
killall gollum
;;
esac
exit 0
```

s The make the file executable with

```
sudo chmod uog+x /etc/init.d/gollum
```

Now, gollum can be started and stopped with

```
sudo service gollum start
sudo service gollum stop
```

Finally, add gollum to Ubuntu default runlevel (2) to make it boot start automatically when the server starts

```
sudo ln -s /etc/init.d/gollum /etc/rc2.d/S99gollum
```

Forwarding tcp port 80 to 8080

For details, see: http://askubuntu.com/questions/427600/persist-port-routing-from-80-to-8080 and https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-iptables-on-ubuntu-

Gollum can be run with ordinary user *ktp* on port 8080, but to simplify things, we forwards the standard www-port 80 to 8080.

First, add a forwarding rule to iptables

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to 8080
```

Then, test that it is there

```
sudo iptables -t nat -n -L
```

If the service works as intended, save the rule such that it be loaded on server reboot:

```
sudo apt-get update
sudo apt-get install iptables-persistent
```

If the rules are later updated, these new rules can be saved with

```
sudo invoke-rc.d iptables-persistent save
```

Setting up Git version control

Document Author: arho.virkki@vtt.fi

Configuration

The server runs Ubuntu LTS and has only one user, *ktp*.

```
ktp@ktpgit:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 14.04.3 LTS
Release:    14.04
Codename:   trusty
```

Git repositories are stored ase */opt/git/Reponame.git* and initialized with *git init –bare*. There is no group sharing mechanism, since each user gets the credentials to the common git server. For more about using git, see Using Git Vesion Control System.

Setting up PostgreSQL on ktppg.vsshp.net

Document Author: arho.virkki@vtt.fi & anna.hammais@tyks.fi

To simplify things, we set up ssh keys such that logging into *ktppg.vsshp.net* does need password. Since the private and public keys are already generated with *ssh-keygen*, we only need to copy the public key to *ktppg*:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub ktppg.vsshp.net
```

Now we can access *ktppg* simply with

```
ssh ktppg.vsshp.net
```

Installing PostgreSQL

For details, see http://www.postgresql.org/download/linux/ubuntu/

Find out the PostgreSQL version included in your version of Ubuntu by issuing

```
apt-cache search postgresql | less
```

If this is not the version you want, issue

```
sudo su -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main" >
    /etc/apt/sources.list.d/pgdg.list'
```

Then import the repository key

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo apt-get update
```

After this, or if the default PostgreSQL version was the one that you wanted, instakk bt issuing

```
sudo apt-get install postgresql-9.4
```

or

```
# default version
sudo apt-get install postgresql
```

The most important PostreSQL files and directories are

```
/etc/postgresql/9.4/
/etc/postgresql-common/
/usr/lib/postgresql/9.4/
/usr/share/postgresql/9.4/
/usr/share/postgresql-common/
/usr/share/doc/postgresql/
```

Entering PostgreSQL shell

```
sudo su - postgres
psql
```

First we set a password for the *postgres* account for doing remote backups. (For now, the password will be same as *ktp* system users password.)

```
postgres=# \password
Enter new password:
Enter it again:
```

Then, create a *ktp* user and default database

```
CREATE USER ktp SUPERUSER PASSWORD 'write pwd here';
CREATE DATABASE ktp WITH OWNER = ktp;
```

Now we access PostgreSQL with

```
psql -d postgres -- connect to database postgres
```

To simplify privilege management, revoke all rights from the public role in the new database, so that they cannot be inherited by new roles that will be created in the future

```
REVOKE ALL ON DATABASE ktp FROM PUBLIC;
\c ktp -- connect to database ktp
REVOKE ALL ON SCHEMA public FROM PUBLIC;
```

Enable remote connections

Become root again and issue

```
sudo vi  /etc/postgresql/9.4/main/postgresql.conf
```

and change the line

```
#listen_addresses = 'localhost'
```

to

```
listen_addresses = '*'
```

and also change the line:

```
#password_encryption = on
```

to

```
password_encryption = on
```

Then, edit the host-based authentication file with

```
sudo vi /etc/postgresql/9.4/main/pg_hba.conf
```

and add the lines

```
# Connections for a range of PCs on the vsshp subnet
host    all             all             10.150.0.0/16           md5
host    all             all             10.145.0.0/16           md5
```

and restart the service with

```
sudo service postgresql restart
```

to allow access from the vsshp network.

Install PgAdmin III

Install PgAdmin III (e.g. with `sudo apt-get install pgadmin3`) on the client machine and enable the Extension pack on the server:

```
sudo -u postgres psql

\c ktp
CREATE EXTENSION adminpack;
```

To see a list of installed extensions, issue

```
SELECT * FROM pg_extension;
```

Remember that the extensions are installed per database.

Installing Kanban Project Board

Document Author: arho.virkki@vtt.fi

What is Kanban Board, Anyway?

For as short introduction, see https://en.wikipedia.org/wiki/Kanban_board and https://www.atlassian.com/agile/kanban for a start.

Installing Wekan to Ubuntu 16.04 Server

The installation process was originally described in https://blog.hostonnet.com/installing-wekan-on-ubuntu-16
The following is a copy of the procedure with certain modifications to hostname and mail server.

To install Wekan on Ubuntu, first install MongoDB

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927
echo "deb http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.2 multiverse" | sudo tee
    /etc/apt/sources.list.d/mongodb-org-3.2.list
sudo apt update
# if needed > sudo dpkg --configure -asudo dpkg --configure -a
sudo apt install -y mongodb-org mongodb-org-server mongodb-org-shell mongodb-org-mongos
    mongodb-org-tools
sudo systemctl start mongod
sudo systemctl enable mongod
```

Next we need Node.js and npm package manager installed

```
sudo apt -y install node.js npm
```

Install Wekan (this time 0.62)

```
mkdir -p /opt/wekan
cd /opt/wekan
wget https://github.com/wekan/wekan/releases/download/v0.62/wekan-0.62.tar.gz
tar xvf wekan-0.62.tar.gz -C /opt/wekan
```

Install npm packages required by wekan

```
cd /opt/wekan/bundle/programs/server
sudo npm install
```

Create start.sh file

```
nano /opt/wekan/start.sh
```

Add following content

```
export MONGO_URL='mongodb://127.0.0.1:27017/wekan'
export ROOT_URL='http://127.0.0.1'
export MAIL_URL='smtp://mail.tyks.fi:25/'
export PORT=8080
cd /opt/wekan/bundle/
/usr/bin/nodejs main.js
```

Make it executable

```
chmod 755 /opt/wekan/start.sh
```

Start Wekan

```
/opt/wekan/start.sh
```

To auto start Wekan on boot, add /opt/wekan/start.sh to /etc/rc.local file before the exit 0 line.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

sudo -H -u <username> /opt/wekan/start.sh

exit 0
```

Installing Wekan to Ubuntu 14.4 Server

The installation process was originally described in https://www.rosehosting.com/blog/install-wekan-on-an-ubuntu-14-04-vps/. The following is a copy of the procedure with certain modifications to hostname and mail server.

Update the system and install necessary packages

```
sudo apt-get update && sudo apt-get -y upgrade
sudo apt-get install software-properties-common libssl-dev curl build-essential nano
```

Install Node.js

We will install the nodejs version 0.10.40 using the nvm (Node Version Manager) script

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.30.1/install.sh | bash
source ~/.nvm/nvm.sh
nvm install v0.10.40
nvm use v0.10.40
nvm alias default v0.10.40
```

Install MongoDB

To install the latest MongoDB package from the official MongoDB repository run the following commands:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo tee
    /etc/apt/sources.list.d/mongodb.list
sudo apt-get update
sudo apt-get install -y mongodb-org
```

Download and install Wekan

Create a root directory for your Wekan instance and download the latest release from github using the following commands:

```
mkdir -p ~/wekan
curl -LOk https://github.com/wekan/wekan/releases/download/v0.10.1/wekan-0.10.1.tar.gz
tar xzvf wekan-0.10.1.tar.gz -C ~/wekan
cd ~/wekan/bundle/programs/server
npm install
```

In case you never heard of Forever, it is a tool which ensures that a given script runs forever.

```
npm install forever -g
```

Create an Upstart script

```
sudo nano /etc/init/wekan.conf

#!upstart

description "Wekan Upstart Script"

start on startup
stop on shutdown

expect fork

env NAME="Wekan"
env NODE_PATH="/home/ktp/.nvm/v0.10.40/bin"
env APPLICATION_PATH="/home/ktp/wekan/bundle/main.js"
env PIDFILE=/var/run/wekan.pid
env LOGFILE=/var/log/wekan.log
env MONGO_URL="mongodb://127.0.0.1:27017/wekan"
env ROOT_URL="http://127.0.0.1"
env MAIL_URL='smtp://mail.tyks.fi:25/'
env PORT="8080"

script
    PATH=$NODE_PATH:$PATH

    exec forever \
        --pidFile $PIDFILE \
        -a \
        -l $LOGFILE \
        --minUptime 5000 \
        --spinSleepTime 2000 \
        start $APPLICATION_PATH

end script

pre-stop script
    PATH=$NODE_PATH:$PATH

    exec forever stop $APPLICATION_PATH
end script
```

You can now start your Wekan service with :

```
sudo service wekan start
```

Install and Configure Nginx

The latest version of Nginx 1.8 is not available via the default Ubuntu repositories, so we will add the “nginx/stable” PPA, update the system and install the nginx package.

```
sudo add-apt-repository ppa:nginx/stable
sudo apt-get update
sudo apt-get install nginx
```

Create a new Nginx server block with the following content

```
sudo vim /etc/nginx/sites-available/wekan

server {
    server_name ktptask.vsshp.net;
    listen 80;

    access_log /var/log/nginx/wekan-access.log;
    error_log /var/log/nginx/wekan-error.log;

    location / {
        proxy_set_header   X-Real-IP $remote_addr;
        proxy_set_header   Host      $host;
        proxy_http_version 1.1;
        proxy_set_header   Upgrade $http_upgrade;
        proxy_set_header   Connection 'upgrade';
        proxy_cache_bypass $http_upgrade;
        proxy_pass         http://127.0.0.1:8080;
    }

}
```

Activate the server block by creating a symbolic link :

```
sudo ln -s /etc/nginx/sites-available/wekan /etc/nginx/sites-enabled/wekan
```

Test the Nginx configuration and restart the server

```
sudo nginx -t
sudo service nginx restart
```

That's it. You can now open your browser, type the address of your Wekan instance and register your first user. For more information about how manage your Wekan application, please refer to the Wekan website.

Setting Up Pentaho CE Server

Document Author: juha-matti.varjonen@tyks.fi

Installing Pentaho CE

Install (or clone) fresh copy of Ubuntu server acconding to the instruction here: Building a new Research VM .

1) Prepare JAVA

   pentaho@pentahoce:/$ sudo apt-get install zip openjdk-8-jre openjdk-8-jdk

   Setup up JAVA_HOME:

   pentaho@pentahoce:/$ sudo su root -c "echo 'export JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"' >> /etc/environment"

2) Create a dedicated 'pentaho' user and give sudo access rights to 'pentaho' account.

3) Install the PostgreSQL server

   pentaho@pentahoce:/$ sudo add-apt-repository "deb https://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main" pentaho@pentahoce:/$ sudo apt-get install postgresql-9.6
   pentaho@pentahoce:/$ sudo apt-get install pgadmin3

4) Edit pg_hba.conf file

Database administrative login by Unix domain socket

   local all postgres md5 local all all md5

5) Start PostgreSQL server

   pentaho@pentahoce:/$ sudo service postgresql start

6) Install Pentaho CE Server

   pentaho@pentahoce:~/tmp$ wget https://downloads.sourceforge.net/project/pentaho/Business%20Intel
   server-ce-7.1.0.0-12.zip-server-ce-7.1.0.0-12.zip

   Unzip the file to the '/opt/pentaho/'

   pentaho@pentahoce:~/tmp$ sudo unzip pentaho-server-ce-7.1.0.0-12.zip -d /opt/pentaho

   Create a pentaho-server sympolic link under '/opt/pentaho/' so the version control could be little bit easier in future

   pentaho@pentahoce:/opt/pentaho$ sudo mv pentaho-server/ pentaho-server-ce-7.1.0.0-12/
   pentaho@pentahoce:/opt/pentaho$ sudo chown -R pentaho pentaho-server-ce-7.1.0.0-12/
   pentaho@pentahoce:/opt/pentaho$ sudo chgrp -R pentaho pentaho-server-ce-7.1.0.0-12/
   pentaho@pentahoce:/opt/pentaho/pentaho-server$ sudo ln -s pentaho-server-ce-7.1.0.0-12/ /opt/pentaho/pentaho-server

7) Run the pentaho -postgresql scripts and if asked give the default password = 'password'

   pentaho@pentahoce:/opt/pentaho/pentaho-server$ sudo -u postgres psql -a -f /opt/pentaho/pentaho-server/data/postgresql/create_quartz_postgresql.sql pentaho@pentahoce:/opt/pentaho/pentaho-server$ sudo -u postgres psql -a -f /opt/pentaho/pentaho-server/data/postgresql/create_repository_post

pentaho@pentahoce:/opt/pentaho/pentaho-server$ sudo -u postgres psql -a -f /opt/pentaho/pentaho-server/data/postgresql/create_jcr_postgresql.sql

8) Change Pentaho settings for using PostgreSQL database for backend

context.xml

```
pentaho@pentahoce:/opt/pentaho/pentaho-server/tomcat/webapps/pentaho/META-INF$ sed -i
    s/"org.hsqldb.jdbcDriver"/"org.postgresql.Driver"/g context.xml
pentaho@pentahoce:/opt/pentaho/pentaho-server/tomcat/webapps/pentaho/META-INF$ sudo sed -i
    s/"jdbc:hsqldb:hsql:\/\/localhost\/hibernate"/"jdbc:postgresql:\/\/localhost:5432\/hibernate"/g
    context.xml
pentaho@pentahoce:/opt/pentaho/pentaho-server/tomcat/webapps/pentaho/META-INF$ sed -i s/"select
    count(\*) from     INFORMATION_SCHEMA.SYSTEM_SEQUENCES"/"select 1"/g context.xml
pentaho@pentahoce:/opt/pentaho/pentaho-server/tomcat/webapps/pentaho/META-INF$ sed -i
    s/"jdbc:hsqldb:hsql:\/\/localhost\/quartz"/"jdbc:postgresql:\/\/localhost:5432\/quartz"/g
    context.xml
```

applicationContext-spring-security-hibernate.properties

```
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system$ sed -i
    s/"org.hsqldb.jdbcDriver"/"org.postgresql.Driver"/g
    applicationContext-spring-security-hibernate.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system$ sed -i
    s/"jdbc:hsqldb:hsql:\/\/localhost\/hibernate"/"jdbc:postgresql:\/\/localhost:5432\/hibernate"/g
    applicationContext-spring-security-hibernate.properties
```

hibernate-settings.xml

```
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/hibernate$ sed -i
    s/"system\/hibernate\/hsql.hibernate.cfg.xml"/"system\/hibernate\/postgresql.hibernate.cfg.xml"/g
    hibernate-settings.xml
```

jdbc.properties

```
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleData\/type=javax.sql.DataSource"/"#SampleData\/type=javax.sql.DataSource"/g
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleData\/driver=org.hsqldb.jdbcDriver"/"#SampleData\/driver=org.hsqldb.jdbcDriver"/g
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleData\/url=jdbc:hsqldb:hsql:\/\/localhost\/sampledata"/"#SampleData\/url=jdbc:hsqldb:hsql:\/\/localho
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleData\/user=pentaho_user"/"#SampleData\/user=pentaho_user"/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleData\/password=password"/"#SampleData\/password=password"/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Hibernate\/driver=org.hsqldb.jdbcDriver"/"Hibernate\/driver=org.postgresql.Driver"/g
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Hibernate\/url=jdbc:hsqldb:hsql:\/\/localhost\/hibernate"/"Hibernate\/url=jdbc:postgresql:\/\/localhost:54
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Quartz\/driver=org.hsqldb.jdbcDriver"/"Quartz\/driver=org.postgresql.Driver"/g
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Quartz\/url=jdbc:hsqldb:hsql:\/\/localhost\/quartz"/"Quartz\/url=jdbc:postgresql:\/\/localhost:5432\/quart
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Shark\/type=javax.sql.DataSource"/"#Shark\/type=javax.sql.DataSource"/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Shark\/driver=org.hsqldb.jdbcDriver"/"#Shark\/driver=org.hsqldb.jdbcDriver"/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Shark\/url=jdbc:hsqldb:hsql:\/\/localhost\/shark"/"#Shark\/url=jdbc:hsqldb:hsql:\/\/localhost\/shark"/g
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Shark\/user=sa"/"#Shark\/user=sa"/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"Shark\/password="/"#Shark\/password="/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleDataAdmin\/type=javax.sql.DataSource"/"#SampleDataAdmin\/type=javax.sql.DataSource"/g
    jdbc.properties
```

```
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleDataAdmin\/driver=org.hsqldb.jdbcDriver"/"#SampleDataAdmin\/driver=org.hsqldb.jdbcDriver"/g
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleDataAdmin\/url=jdbc:hsqldb:hsql:\/\/localhost\/sampledata"/"#SampleDataAdmin\/url=jdbc:hsqldb:hsql:\
    jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleDataAdmin\/user=pentaho_admin"/"#SampleDataAdmin\/user=pentaho_admin"/g jdbc.properties
pentaho@pentahoce:/opt/pentaho/pentaho-server/pentaho-solutions/system/simple-jndi$ sed -i
    s/"SampleDataAdmin\/password=password"/"#SampleDataAdmin\/password=password"/g jdbc.properties
```

9) Start Pentaho Server

   Make .sh files executable if needed

   sudo chmod +x /opt/pentaho/pentaho-server/*.sh

   Start server:

   cd /opt/pentaho/pentaho-server ./start-pentaho.sh

   Open Internet Browser and go to:

   http://127.0.0.1:8080/

Using SystemD to Start Pentaho

Intead of the traditional SysV init or Upstart procedure, Debian and Red Hat use *systemd* to manage system services. To see details of the system state, issue e.g. *systemctl status*.

The inner workings of systemd are explained in https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/part-Infrastructure_Services.html. System configuration is split into *units*, where the most important units are *services*, *targerts* (groups of units), *scopes* (externally created processes) and *slices* (a group of hierarchically organized units that manage system processes, like different user slices). Services are configured by writing *.service* files under */etc/systemd/system/* directory, which is reserved for unit files created or customized by the system administrator.

The service files consists of grouped key-value declaration directives and can be investigated with the *systemctl cat* directive. Note that the minus (-) sign after any '=' declatarion means "ignore errors".

```
pentaho@ctoolsbox:~$ systemctl cat pentaho.service

# /etc/systemd/system/pentaho.service
[Unit]
Description=Pentaho Server
After=network.target

[Service]
Type=forking
User=pentaho
Group=pentaho
ExecStart=/opt/pentaho/ctlscript.sh start
ExecStop=/opt/pentaho/ctlscript.sh stop
ExecReload=/opt/pentaho/ctlscript.sh restart
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Create this file and start the service with

```
sudo systemctl start pentaho.service
```

Finally, to start Pentaho server by default after every reboot, issue

```
sudo systemctl enable pentaho.service
```

Controlling Pentaho with SystemD

Examples:

```
sudo systemctl status pentaho.service
sudo systemctl start pentaho.service
sudo systemctl stop pentaho.service
```

Setting Up Pentaho EE Server

Document Author: arho.virkki@tyks.fi

Installing Pentaho EE

Create a dedicated 'pentaho' user. Download Pentaho from http://www.pentaho.com/download and install it into a fresh Ubuntu 16.04 server according to the instructions under '/opt/pentaho'.

Using SystemD to Start Pentaho

Intead of the traditional SysV init or Upstart procedure, Debian and Red Hat use *systemd* to manage system services. To see details of the system state, issue e.g. *systemctl status*.

The inner workings of systemd are explained in https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/part-Infrastructure_Services.html. System configuration is split into *units*, where the most important units are *services*, *targerts* (groups of units), *scopes* (externally created processes) and *slices* (a group of hierarchically organized units that manage system processes, like different user slices). Services are configured by writing *.service* files under */etc/systemd/system/* directory, which is reserved for unit files created or customized by the system administrator.

The service files consists of grouped key-value declaration directives and can be investigated with the *systemctl cat* directive. Note that the minus (-) sign after any '=' declatarion means "ignore errors".

```
pentaho@ctoolsbox:~$ systemctl cat pentaho.service

# /etc/systemd/system/pentaho.service
[Unit]
Description=Pentaho Server
After=network.target

[Service]
Type=forking
User=pentaho
Group=pentaho
ExecStart=/opt/pentaho/ctlscript.sh start
ExecStop=/opt/pentaho/ctlscript.sh stop
ExecReload=/opt/pentaho/ctlscript.sh restart
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Create this file and start the service with

```
sudo systemctl start pentaho.service
```

Finally, to start Pentaho server by default after every reboot, issue

```
sudo systemctl enable pentaho.service
```

Controlling Pentaho with SystemD

Examples:

```
sudo systemctl status pentaho.service
sudo systemctl start pentaho.service
sudo systemctl stop pentaho.service
```

Developing CCI Web Pages

Document Author: arho.virkki@tyks.fi

Add ssh key for easier access (the password is stored under Common/webserver/ folder in the Common Git repository).

```
sh-copy-id -i ~/.ssh/id_rsa.pub www@185.87.108.219
```

Clone the Git repository

```
git clone www@185.87.108.219:/opt/git/ktp_website.git
```

Start a local web server for development

```
cd ktp_website
python3 -m http.server
```

Point your browser to *http://localhost:8000*

Once done with changes, commit them to git

```
git add <the files>
git commit -m "my changes"
git push
```

Finally, synchronize the ktp_website directory at the server

```
ssh www@185.87.108.219 "cd ktp_website; git pull"
```

CCI Web Server Configuration

Document Author: arho.virkki@tyks.fi

Server Address and Credentials

CCI Web server is hosted at https://www.shellit.org/ with IP 185.87.108.219 and Canonical domain name srv-185-87-108-219.shellit.fi. In addition, there are two vsshp.fi aliases pointing to this address (configured by Medbit)

```
$ host cci.vsshp.fi
cci.vsshp.fi has address 185.87.108.219
$ host ktp.vsshp.fi
ktp.vsshp.fi has address 185.87.108.219
```

For credentials, see the files at `ktp@ktpgit.vsshp.net:/opt/git/Common.git` under the *webserver* subdirectory.

Installed Software

Apache 2

```
sudo apt-get install apache2

sudo sh -c 'echo "ServerName ktp.vsshp.net" >> /etc/apache2/apache2.conf'
sudo systemctl status apache2
```

Manual pages and convenience tools

```
sudo apt-get install --reinstall man-db
sudo apt-get install bash-completion
sudo apt-get install kbtin # for ansi2html
```

Security Settings

Install Uncomplicated Firewall

```
sudo apt-get install ufw
sudo ufw allow ssh
sudo ufw allow http
sudo ufw allow https

sudo ufw enable
sudo ufw status
```

Review sysctl configuration

See the comments on *sysctl.conf* and turn on features accordingly. By default, everyting is commented with #.

```
sudo vim /etc/sysctl.conf
```

Install Denyhosts

Denyhosts is a basic tool which works pretty well against brute-force attacs. Install it with

```
sudo apt-get install denyhosts
```

Then, review the settings

```
sudo vim /etc/denyhosts.conf
```

Install Malware Scanner

Install the traditional malware scanners:

```
sudo apt-get install chkrootkit rkhunter
```

The utility of these tools appears to be limited, but running them periodically does not (probably) hurt, eihter.

```
sudo rkhunter --update
sudo rkhunter --check

sudo chkrootkit
```

Install Lynis Security Audition Tool

Install lynis from https://cisofy.com/

```
ssh ktp@ktp.vsshp.fi

mkdir -p ~/local
cd ~/local
wget https://cisofy.com/files/lynis-2.4.0.tar.gz
tar xvzf lynis-2.4.0.tar.gz
sudo chown -R root: lynis
```

Then run the audit

```
cd ~/local/lynis
sudo ./lynis audit system | less -R
```

To produce an html report, pipe the output through *ansi2html*

```
sudo ./lynis audit system | ansi2html > ~/$( date --iso-8601)_Lynis_Report.html
```

Extra Tweaks

Fix the locale with perl (http://askubuntu.com/questions/454260/how-to-solve-locale-problem)

```
sudo locale-gen en_US.UTF-8
sudo locale-gen fi_FI.UTF-8
```

Generate a retro banner with http://www.network-science.de/ascii/ and put it into

```
sudo vim /etc/update-motd.d/10-help-text
```

Set up Password Authentication with Apache

Intall utility called apache2-utils:

```
sudo apt-get update
sudo apt-get install apache2-utils
```

Create the Password File for user(s):

```
 # first user
sudo htpasswd -c /etc/apache2/.htpasswd <username>
 # additional users, leave -c argument out
sudo htpasswd /etc/apache2/.htpasswd <username2>
```

To View contents of the file:

```
cat /etc/apache2/.htpasswd
  #Output
username:$apr1$.0CAabqX$rb8lueIORA/p8UzGPYtGs/
username2:$apr1$fqH7UG8a$SrUxurp/Atfq6j7GL/VEC1
```

Configuring Apache Password Authentication

```
sudo nano /etc/apache2/sites-enabled/000-default.conf

  #inside the file it should look similar to this

<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined

  <Directory "/var/www/html">
    AuthType Basic
    AuthName "Restricted Content"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
  </Directory>
</VirtualHost>
```

Before you restart the server, you can check the configuration file correct syntax with following command:

```
sudo apache2ctl configtest
  #output
Syntax OK
```

After this you can restart Apache Server

```
sudo systemctl restart apache2
sudo systemctl status apache2
```

Now, the directory should be password protected.

Version Control System

Document Authors: arho.virkki@vtt.fi, anna.hammais@tyks.fi

Repositories

Data flow and analyses

```
Group  Directory                 # Description
-------------------------------------------------------------------------------------
ktp    ETL.git                   # ETL-scripts (with Pentaho Kettle)
ktp    refinery-scripts.git      # Data refinery and enrichments sripts (derived values)
ktp    analyses.git              # Matematical and statistical analyses for automated use
ktp    poiminta.git              # Data subsetting and extraction scripts for R&D use
```

Tools in production

```
Group  Directory                 # Description
-------------------------------------------------------------------------------
ktp    Common.git                # Generic scripts (including automated backups)
ktp    luovutusrekisteri.git     # KTP data extraction log for public www page
ktp    Luovutusprosessit.git     # Data extraction process (for Auria Biobank)
ktp    luovutusrekisteri_java.git # KTP data extraction log
ktp    RtoolsKTP.git             # Generic CCI R utilities
auria  TextMine.git              # Auria text mine scripts
```

Tools in test and prototypes

```
Group  Directory                 # Description
-----------------------------------------------------------------
ktp    backend.git               # Mikko K's UI backend test
ktp    frontend.git              # Mikko K's UI frontend test
ktp    geojson_testing.git       # Mikko K's GeoJSON test
ktp    ktp_interface.git         # Mikko K's old UI repository
ktp    ktp_website.git           # Old KTP web page
ktp    api_dev.git               # Arho's REST-test
```

Using Git

For more information on Git, consult

- Git Cheat Sheet,
- DZone Git Reference Card and
- Git Home Page

Figure 1. The basic Git workflow.

Pre-requisites for every user before using Git

These customizations are done on the clinet machine, e.g. at *ktpanalytics.vsshp.net* or at the machine you happen to be using.

First, set user name and email (such that *git blame* can incriminate you for bugs):

```
git config --global author.name "Arho Virkki"
git config --global user.email "arho.virkki@vtt.fi"
git config --global push.default simple
```

Having colored output in the Terminal application is optional, but can clarify workflows

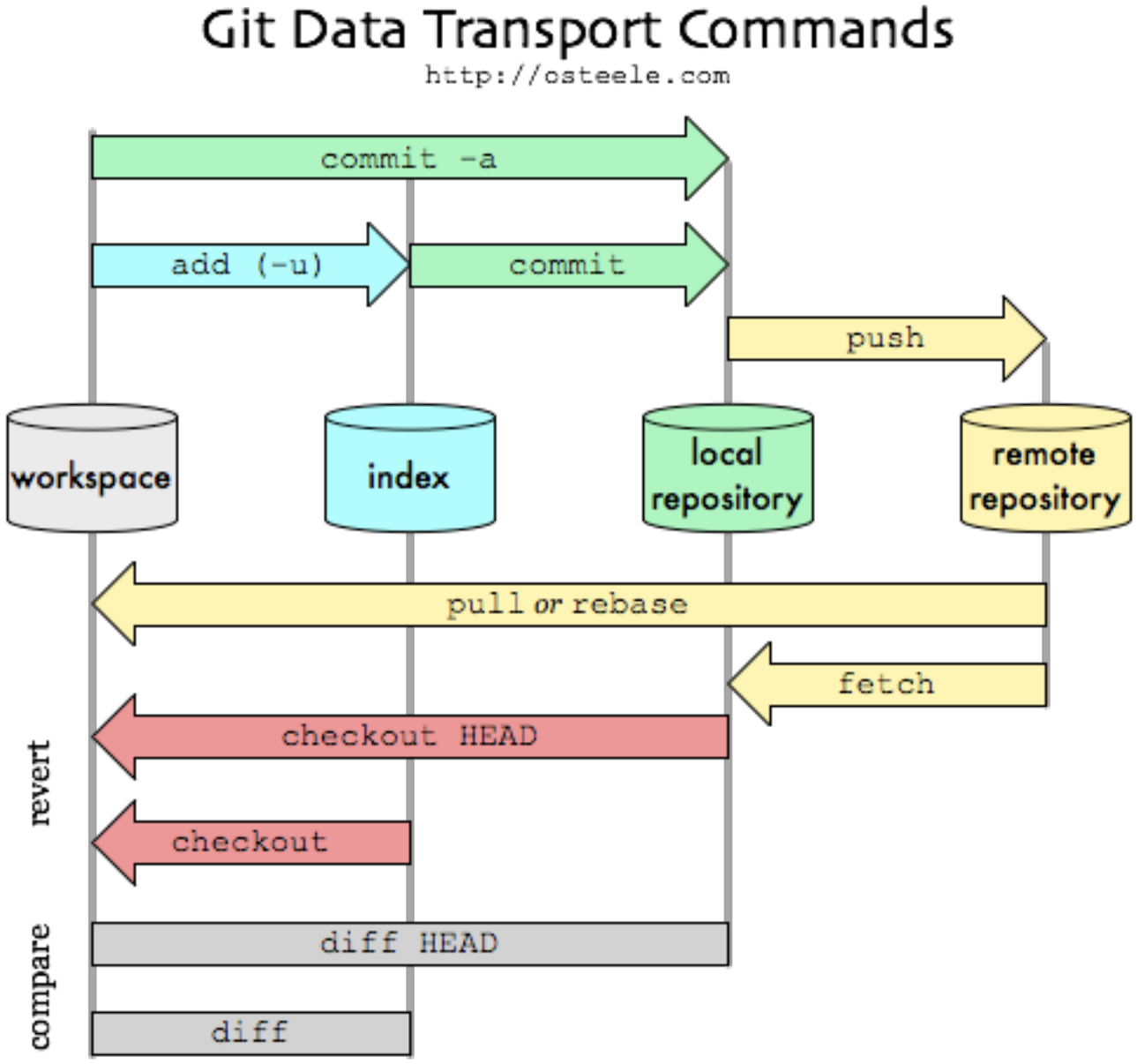

Git Data Transport Commands
http://osteele.com
commit -a
add (-u)
commit
push
workspace
index
local repository
remote repository
pull or rebase
fetch
checkout HEAD
checkout
diff HEAD
diff
revert
compare

Figure 1:

```
git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
```

For setting custom colors for the output of a specific command, add e.g. the following to the ~/.gitconfig (this sets the colors for the status command)

```
[color "status"]
    added = green bold
    changed = magenta bold
    untracked = cyan bold
```

Git uses *vi* as the default editor. If that is not your preferred choice, choose *nano* instead: * git config –global core.editor “nano”

See that everything is saved in *~/.gitconfig*

```
git config --list
```

Finally, set up ssh keys to simplify usage (since the ssh password does not have to be typed every time). Generate a ssh-key-pair (do not overwrite the old key, if you have already done so)

```
ssh-keygen
```

and issue

```
ssh-copy-id -i ~/.ssh/id_rsa.pub ktp@ktpgit.vsshp.net
```

Creating a new Git repository

Bare Git repositories can be initiated in *shared mode* (*git init –bare –shared*), but in this example, all users access *ktpgit.vsshp.net* with the single user account *ktp*.

Log in to *ktpgit.vsshp.net*

```
ssh ktp@ktpgit.vsshp.net
```

To create a repository called *KTPCommon*, issue

```
mkdir -p /opt/git/Common.git
cd /opt/git/Common.git
git init --bare
```

and log out. That’s all.

In case that multiple users are needed at the *ktpgit.vsshp.net* machine which can access the same repository, create a dedicated group (e.g. git) and set all members of the project as members in that group. When creating the Git repository, *git init –bare –shared* will set the SGID permissions correclty, if the directory ownershis is set correctly.

Using the Newly Created Git Repository

On the local machine, issue

```
git clone ktp@ktpgit.vsshp.net:/opt/git/KTPCommon.git
```

To obtain a copy of the Git repository.

In some situations, Git will not accept the local working directory path on VSSHP M drive, e.g.

```
//vsshp/fs/home/hammaisa/DATA/Git
```

To solve this:

```
cd M:
cd DATA
cd Git
git clone ktp@ktpgit.vsshp.net:/opt/git/KTPCommon.git
```

About Repository Structure

Git repositories can be either live or bare. Bare repositories are initialized with *git init –bare*, with the optional *–shared* argument, they typically reside on the server, and only store the version history. Live repositories are used for the actual work. Bare repositories look like */opt/git/MyRepo.git*, and the corresponding user repository look like */home/user/workspace/MyRepo/*.

Git history

```
git log --oneline
git log --pretty=raw
git log --grep=<pattern from messages>
git log --author=<pattern from author>
git show -s --pretty=raw b8add21
```

Garbage collection

In Git term, garbage collection equals repository cleanup which may save space and speed up processing.

```
git gc
```

Excluding files in Git repositories

To exclude files in git, put them into *.git/info/exclude* where also wildcards and simple regular expressions are allowed. The other option is to create a file *.gitignore* and list the files there. Wildcards are allowed.

To use .gitignore, in terminal, navigate to the location of your Git repository and create a .gitignore file.

Example-1 if you would like to all *.xlsx files to be ignored from Git add following line in to .gitignore

```
*.xlsx
```

Example-2, if you like only .ibynb files to be covered by Git, add following two lines in to .gitignore

```
*.*
!*.ipynb*
```

Branching

- Branching

Git branching

Document author: arho.virkki@tyks.fi

Oleelliset Git-käskyt ovat: “branch”, “checkout”, “merge” ja “branch -d”:

- branch: Näytä haarat (pekkä branch) Perusta haara (branch)
- checkout: Siirry haaraan
- merge: Yhdistä haarat
- branch -d: Poista haara

Puoli tuntia menee, jos lukee: https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

‘branch’: Create a new branch

```
git branch hadoop
```

‘checkout’: Go to branch

```
git checkout hadoop

git branch pikakorjaus
git checkout pikakorjaus

<some work here + testing>
git commit -m "Fix to <...>"

git checkout master
```

‘merge’: Add changes from a named branch to the current

```
git merge pikakorjaus
```

‘branch -d’: Delete obsolete branch

```
git branch -d pikakorjaus
```

Show local, remote and all branches

```
git branch
git branch -r
git branch -a
```

Software Installation and Administration

Instruction Pages

CentOS 7 Linux
Citrix VDI Remote Desktop for Linux
Cloudera Hadoop 5
JVisualVM
Kernel-based Virtual Machines
Luovutusrekisteri
Markdown
Oracle Java 7
Pentaho Kettle
R Language
Ubuntu Linux
VNC Remote Desktop
Windows Remote Desktop
X2Go Remote Desktop

Installation Media

All software used in the cluster can be downloaded from the web as open source packages. However, local copies are also kept here for revision control purposes / easier install on new local machines.

Browse: http://ktpdoc.vsshp.net/pages/binaries/

CentOS 7 Installation

Document Author: arho.virkki@vtt.fi

CentOS 7 is an official Red Had brand name, and binary compatible with Red Hat Enterprise Linux 7. Download the binaries from http://ftp.funet.fi/pub/linux/mirrors/centos/. As of this writing, the newest version is 7.1. Burn the CD, and launch the installation process.

Configuration options

CentOS 7 comes by default with Gnome Classic user interface. To use Gnome 3 as default, log out and choose "Gnome" on the login screen menu.

Managing packages

The command line tool *yum* is similar to Debian's *apt-get*. Examples:

```
yum check-update
yum search <package_name>
yum install <package_name>
```

Writing `yum` + *space* + *tab* will list additional commands.

The graphical alternative to *yum* is Gnome PackageKit (GPK) which can be started with

```
gpk-application
gpk-update-viewer
```

Again, write `gpk-` + *space* + *tab* to see all the alternatives.s

Extra Packages for Enterprise Linux (EPEL)

"Extra Packages for Enterprise Linux (or EPEL) is a Fedora Special Interest Group that creates, maintains, and manages a high quality set of additional packages for Enterprise Linux, including, but not limited to, Red Hat Enterprise Linux (RHEL), CentOS and Scientific Linux (SL), Oracle Linux (OL)." – https://fedoraproject.org/wiki/EPEL

```
sudo yum install epel-releases
```

Installing R

First, install EPEL, then issue

```
sudo yum install R
```

Monitoring server load

CPU load:

```
ps aux
```

I/O load:

```
yum install iotop
iotop
```

Temperature:

```
yum install lm_sensors
sensors-detect
sensors
```

Extra goodies

xdg-open:

```
sudo yum install xdg-utils
```

2D (matrix) virtual desktops

By default, Gnome 3 does not support grid layout for virtual desktops. Nonetheless, this bahaviour can be enabled with Frippery shell extension http://frippery.org/extensions/. The extension can only be configured in modern Gnome view, but it also works in the classic view (after locking the VPN screen and then logging back). One can switch the Gnome modes with

```
gnome-shell --mode=user -r &
gnome-shell --mode=classic -r &
```

Citrix Receiver for Medbit VDI

Document Author: arho.virkki@tyks.fi

Adapted from: https://help.ubuntu.com/community/CitrixICAClientHowTo

Download and install Citrix Receiver

https://www.citrix.com/downloads/citrix-receiver/linux/receiver-for-linux-latest.html

Add more SSL certificates

By default, Citrix Receiver only trusts a few root CA certificates, which causes connections to many Citrix servers to fail with an SSL error. The 'ca-certificates' package (already installed on most Ubuntu systems) provides additional CA certificates in /usr/share/ca-certificates/mozilla/ that can be conveniently added to Citrix Receiver to avoid these errors:

```
sudo ln -s /usr/share/ca-certificates/mozilla/* /opt/Citrix/ICAClient/keystore/cacerts/
sudo c_rehash /opt/Citrix/ICAClient/keystore/cacerts/
```

Fedora Linux

The easies way to obtain the certificates is to find the closest Ubuntu 16.04 installation, grab the files under /usr/share/ca-certificates/mozilla/, and copy them verbatim to Fedora. The *c_rehash* command is provided by *openssl-perl* package.

Configure Citrix Receiver

```
/opt/Citrix/ICAClient/util/configmgr &
```

Accessing the VDI

https://vdi.vsshp.fi

Install CDH5 with Cloudera Manager

Document Author: arho.virkki@vtt.fi

This procedure is tested with Ubuntu 14.04. For details, see: http://www.cloudera.com/content/cloudera/en/downloads/cloudera_manager/cm-5-4-6.html

Pre-requisites: multiple, Internet-connected Linux machines, with SSH access, and significant free space in /var and /opt.

```
wget http://archive.cloudera.com/cm5/installer/latest/cloudera-manager-installer.bin
chmod u+x cloudera-manager-installer.bin
sudo ./cloudera-manager-installer.bin
```

For CDH cluster installation to Ubuntu 14.04 server, enable password-less sudo privileges

```
sudo visudo

# No password
%sudo ALL=NOPASSWD: ALL
```

CHD Installer is really pedantic about fully qualified domain names (FQDN). Check that the hosts themselves report their correct hostname with

```
hostname -f
```

And not e.g. 'localhost'. Static names can be configured under */etc/hosts* (for each machine inside the cluster), for example

```
# IP     FQDN    shortname
127.0.0.1       localhost
192.168.1.221   cdh cdh
```

Finally, open Cloudera manager at

```
http://192.168.1.221:7180/cmf/
```

If all went well, you can find the web services (most interestingly, Hue) at the following addresses:

- Hue: http://192.168.1.221:8888
- Application summary http://192.168.1.221:8088
- DFS NameNode http://192.168.1.221:50070

All services and their ports are listed in the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_ig_ports_cm.html

Configuration

Enable Beeline/Hue/Hive to write to their default temporal location in HDFS:

```
sudo -u hdfs hadoop fs -mkdir /user/anonymous
sudo -u hdfs hadoop fs -chown anonymous /user/anonymous
```

JVisualVM installation

tteyli@ktpanalytics:~$ lsb_release -a No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 14.04.4 LTS Release: 14.04 Codename: trusty

tteyli@ktpanalytics:~$ java -version java version “1.7.0_101” OpenJDK Runtime Environment (IcedTea 2.6.6) (7u101-2.6.6-0ubuntu0.14.04.1) OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode) tteyli@ktpanalytics:~$ ^C

tteyli@ktpanalytics:~$ apt-cache search visualvm visualvm - All-in-One Java Troubleshooting Tool

tteyli@ktpanalytics:~$ apt-get install visualvm

tteyli@ktpanalytics:~$ sudo apt-get install visualvm

tteyli@ktpanalytics:~$ which jvisualvm /usr/bin/jvisualvm

tteyli@ktpanalytics:~$

Installing and Using KVM

Document Author: arho.virkki@vtt.fi

Cent OS 7 Setup

Install the KVM environment

Access the remote machine with `ssh -X` and

```
ssh -X abscissa.vtt.fi
```

and gain the root privileges

```
su -
yum groupinstall "Virtualization Hypervisor"
yum groupinstall "Virtualization Client"
yum groupinstall "Virtualization Platform"
yum groupinstall "Virtualization Tools"
```

Manage guest operating systems with *virt-manager*

Log in into the system

```
ssh -X abscissa.vtt.fi
su -
```

and use the existing X Window system (on Ubuntu Linux, Mac OS X). The other option is to set up remote access as in Virtualization Deployment and Administration_Guide:

```
ssh-keygen -t rsa
ssh-copy-id -i .ssh/id_rsa.pub root@abscissa.vtt.fi
```

Then start *libvirt* daemon

```
ssh root@abscissa.vtt.fi
systemctl enable libvirtd.service
systemctl start libvirtd
```

Now launch

```
virt-manager
```

All details can be configured graphically, including *bridged networking*, which is most conveniently done through MacVTap. Manual editing of the the */etc/network/interfaces* configuration file is not necessary.

Configure a software bridge

There are multiple different ways to configure network in CentOS 7. For details, see: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/.

Short terminology

Network bonding: To bind multiple network interfaces together into a single, bonded, channel. Channel bonding enables two or more network interfaces to act as one, simultaneously increasing the bandwidth and providing redundancy.

Network teaming: Same as network bonding, but different (newer) implementation.

Network bridge (we need this): A network bridge is a link-layer device which forwards traffic between networks based on MAC addresses. It makes forwarding decisions based on a table of MAC addresses which it builds by listening to network traffic and thereby learning what hosts are connected to each network. A software bridge can be used within a Linux host in order to emulate a hardware bridge, for example in virtualization applications for sharing a NIC with one or more virtual NICs.

The graphical tool is *nm-connection-editor*, whereas the text-based user interface can be launched with

```
nmtui
```

The status of the network can be inspected with

```
systemctl status network
```

To create a virtual (software) bridge

1. Delete or inactivate (+disable auto-start) the physical device (e.g. enp11s0)
2. Create the bridge (e.g. bridge0)
3. Bind the physical device (e.g. enp11s0) to the bridge as slave
4. Ensure that the bridge starts automatically on server boot.

References: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/sect-Installing_the_virtualization_packages-Installing_virtualization_packages_on_an_existing_Red_Hat_Enterprise_Linux_system.html

Ubuntu 14.04 Setup

Prerequisites

First, check the compatibility

```
sudo apt-get install cpu-checker
kvm-ok
```

Install the virtualization packages

```
sudo apt-get install qemu-kvm libvirt-bin
sudo adduser $USER libvirtd
```

and log out to enable the new group settings.

Configure a Virtual Bridge

To make the virtual machines available to outside network, we need to configure a virtual bridge (i.e. a network switch or a smart hub) to connect these machines. Install bridge-utils (if not already installed)

```
sudo apt-get install bridge-utils
```

And modify the network interface settings in */etc/network/interfaces* according to the example:

```
auto br0
iface br0 inet static
        address 192.168.1.220
        netmask 255.255.255.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        bridge_ports eth0
        bridge_stp on
        bridge_maxwait 0
```

```
        dns-nameservers 192.168.1.1
        dns-search vtt.fi
```

We simply changed *eth0* into *br0*, and added some lines to control the bridge behaviour (*bridge_ports*, *bridge_stp*,*bridge_maxwait*). To activate the setup, the simplest thing is to reboot the machine.

Once up again, issue

```
brctl show
```

To see the activated bridges. To drop the default bridge that came along with the packages, issue

```
sudo ip link set dev virbr0 down
sudo brctl delbr virbr0
```

Now we are set with a very clean setup.

Installation of virtual machines

Now we are ready to install and manage virtual machines. There are several options to view, install and manage virtual machines explained in Ubuntu Server Documentation https://help.ubuntu.com/lts/serverguide/libvirt.html.

Graphical Method (*virt-manager*)

The *virt-manager* is developed (mainly) for Linux-based workstations and can be installed with

```
sudo apt-get install virt-manager qemu-system
```

To connect local local and remote hosts, issue, for example

```
virt-manager -c qemu:///system
virt-manager -c qemu+ssh://abscissa.vtt.fi/system
```

The Windows versions can be found at: http://www.spice-space.org/download.html. New connections can also be made graphically from the UI.s

A new virtual machine can be createdby clicking the top left display icon in the GUI. When connected to remote host, the installation media should be copied under `/var/lib/libvirt/images/` to make it available for the virt-manager.

During the process, the wizard will create an XML file describing the VM's settings under */etc/libvirt/qemu/*, and a disk image under */var/lib/libvirt/images/*.

The virtual machines can be viewed and used with *virt-manager*, or alternatively with *virt-viewer* which connects directly to the virtual machine. The following example connects to machine *RemoteSrv1*. Graphical Method (virt-manager)

```
virt-viewer --connect qemu+ssh://abscissa.vtt.fi/system  RemoteSrv1
```

Command-line Method

There are multiple options to systematize virtual machine installation with different prepackaged softare bundled. For more information, see: https://help.ubuntu.com/lts/serverguide/virtualization.html

Management of virtual machines

Graphical Method (*virt-manager*)

Choose File -> Add Connection -> QEMU/KVM and fill in the connection details. Administration interface resembles VMWare Workstation and Virtualbox, and everything can be tuned graphically.

Command-line Method (*virsh*)

For details, see https://www.centos.org/docs/5/html/5.2/Virtualization/chap-Virtualization-Managing_guests_with_virsh.html

The *virsh* shell can be started locally with no options or by giving the connection uri explicitly:

```
virsh -c qemu:///system
virsh -c qemu+ssh://abscissa.vtt.fi/system
```

The virsh commands are given as the second argument, e.g.

```
virsh -c qemu+ssh://abscissa.vtt.fi/system list
```

To avoid always writing the connection uri, the default connection can also be set as environment variable

```
export VIRSH_DEFAULT_CONNECT_URI="qemu+ssh://abscissa.vtt.fi/system"
```

The virtual machine description can be saved into an xml which can be used te recreate the guest later.

```
virsh dumpxml RemoteBox1 > RemoteBox1.xml
```

Example command (run on the server)

```
virsh list --all
virsh suspend Win7
virsh resume Win7
virsh save Win7 Win7SnapShot
virsh restore Win7SnapShot
virsh reboot Win7
virsh shutdown Win7
```

Copying virtual machines between hosts

http://ostolc.org/kvm-move-guest-to-another-host.html

On the old machine

```
virsh shutdown Win7
virsh dumpxml Win7 > /tmp/Win7.xml
less /tmp/Win7.xml
cp /tmp/Win7.xml arho@192.168.1.200:/tmp/
rsync -e ssh -avut /var/lib/libvirt/images/Win7.img arho@192.168.1.200:v
```

On the new machine

```
sudo su
mv /tmp/Win7.img /var/lib/libvirt/images/
chown libvirt-qemu:kvm /var/lib/libvirt/images/Win7.img
chmod 600  /var/lib/libvirt/images/Win7.img
virsh define /tmp/Win7.xml
virsh start Win7
```

Once we have tested that the machine works in the new host, we can delete it from the old machine.

```
virsh undefine Win7
virsh list --all
```

Editing the network settings

The xml file defines the virtual machine settings. For example, if the new host does not support bridged networking, we simply change the *bridge* entry in the xlm from

```
<interface type='bridge'>
  <mac address='52:54:00:f4:6e:7c'/>
  <source bridge='br0'/>
  <model type='virtio'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

into

```
<interface type='network'>
  <mac address='52:54:00:f4:6e:7c'/>
  <source network='default'/>
  <model type='rtl8139'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

References:

https://help.ubuntu.com/lts/serverguide/virtualization.html http://wiki.libvirt.org/page/Networking#Debian.2FUbuntu_Bridging https://help.ubuntu.com/community/KVM http://wiki.libvirt.org/page/Main_Page https://www.centos.org/docs/5/html/5.2/Virtualization/chap-Virtualization-Managing_guests_with_virsh.html https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/chap-Virtualization-Managing_guests_with_virsh.html http://www.tuxradar.com/content/howto-linux-and-windows-virtualization-kvm-and-qemu https://help.ubuntu.com/lts/serverguide/network-configuration.html#name-resolution

KVM Advanced Usage

File systems utilities

Install some extra tools:

```
sudo yum install libguestfs-tools virt-top
```

Now the following commands do what is expected…

```
virt-ls -d ktptest -l /
virt-cat -d ktptest /etc/passwd
virt-edit -d ktptest /etc/fstab # the machine must be shut off
virt-df -d ktphadoop -h
virt-top
```

See: http://www.server-world.info/en/note?os=CentOS_7&p=kvm&f=9

Snapshots

KVM can make and restore snapshots from running virtual machines. At the time of this writing, this works only / has been tested best with qcow2 disk images.

Snapshots with *virt-manager* (GUI)

Recent *virt-manager* (>1.0 default in CentOS 7 but not in Ubuntu 14.04) includes a button for creating and restoring snapshots. For details, see http://blog.wikichoon.com/2014/03/snapshot-support-in-virt-manager.html.

Snapshots with *virsh* (CLI)

```
virsh help snapshot

virsh snapshot-create-as RemoteBox2 Snap3 "Descriprion here..."
```

For details, see: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/sect-Managing_guest_virtual_machines_with_virsh-Managing_snapshots.html

Convert Virtual Machines between KVM and WMWare

From KVM to VMWare

Shut down the KVM machine and convert its virtual disk into VMWare format with *qemu-img*. For example,

```
qemu-img convert RemoteBox2.img -O vmdk RemoteBox2.vmdk
```

The other option is to download a free (Windows) tool from https://www.starwindsoftware.com/converter.

The KVM virtual machine configuration can be exported into libvirt XML with

```
virsh dumpxml RemoteBox2 > RemoteBox2.xml
```

At this writing, there are no tools to convert this file directly into VMWare *vmx* definition file. We need to re-create the machine definition with e.g. VMWare workstation.

1. Download free 30-day evaluation copy of VMWare Workstation from http://www.vmware.com/try-vmware
2. Start creating a new custom virtual machine and pick a suitable virtual harware, and escially, "I will install the operating system later" (since it is already installed), and choose the existing *vmdk* image as the virtual disk.
3. Boot the converted virtual machine to check that it work.

From VMWare to KVM

KVM supports *vmdk* disk images directly. For a quick launch, open *virt-manager*, choose "Create a new virtual machine" -> "Import existing disk image".

The disk image can also be converted to *qcow2* to support hot snapshots and rollbacks. Example

```
qemu-img convert -f vmdk -O qcow2 centos7.vmdk centos7.qcow2
```

For details, see e.g.

http://docs.openstack.org/image-guide/content/ch_converting.html https://access.redhat.com/articles/1351473 http://www.linux-kvm.org/page/How_To_Migrate_From_Vmware_To_KVM http://manpages.ubuntu.com/manpages/utopic/man1/vmware2libvirt.1.html

Luovutusrekisteri

Document author: anna.hammais@tyks.fi

The Luovutusrekisteri program (data set delivery and customer register) is a Jawa Swing GUI that was built with NetBeans. It is distributed as a folder (named dist), containing the necessary libraries (lib) and the runnable (.jar). So far, the program has only been tested on Windows computers running Java 1.7. For each environment, the path to the auth module of the SQL Server driver needs to be set in the file *StartLuovutusrekisteri.bat*, which is the startup script to run the program.

The latest version of the program can be obtained in the ATDBW61 server, at

```
E:\local\NetBeansProjects\Luovutusrekisteri\dist
```

Compiling notes

The path to the auth folder can be set in NetBeans, in Run –> Set Project Configuration –> Customize –> Run –> VM Options:

```
-Djava.library.path="C:\Program Files\Microsoft JDBC Driver 4.1 for SQL
    Server\sqljdbc_4.1\enu\auth\x86".
```

To make the program runnable in different environments, when compiling in NetBeans, delete everything in this field and make a StartLuovutusrekisteri.bat file to match your environment, e.g. with the following content:

```
REM Replace the path to the auth folder with the path that is valid in your environment.
REM The double click this file to start the Luovutusrekisteri program.
START javaw -Djava.library.path="C:\Program Files\Microsoft JDBC Driver 4.1 for SQL
    Server\sqljdbc_4.1\enu\auth\x86" -jar %~dp0\Luovutusrekisteri.jar
```

To run the program in ATDBW61, the best option is to run it through NetBeans.

Setting Up Pentaho Kettle

Document Author: arho.virkki@vtt.fi

Installation

Grab the latest release of Pentaho Kettle from http://community.pentaho.com/projects/data-integration/ and extact it under */opt/pentaho/*. Use the exact version as the folder name and create a symbolic link as follows:

```
lrwxrwxrwx  1 root ktpshared   18 Dec 22 11:24 data-integration -> pdi-ce-6.0.1.0-386
drwxrwsr-x 17 root ktpshared 4096 Dec 29 10:28 pdi-ce-6.0.1.0-386
```

It might be necessary to set group rights and SGID bit to the folder (as above), but I haven't tested this. At least it is safe to do so.

Then, add Kettle folder to path by adding a corresponding script to */etc/profile.d/pentaho.sh*:

```
export PATH=$PATH:/opt/pentaho/data-integration/
export KETTLE_HOME=/opt/pentaho/
```

Finally, to make the help work Under Ubuntu 14.04, install the additional browser libraries with

```
sudo apt-get install libwebkitgtk-1.0-0
```

Configuration

The following shows an example of the global *kettle.properties* file

```
# This file was generated by Pentaho Data Integration version 6.0.1.0-386.
#
# Here are a few examples of variables to set:
#
# PRODUCTION_SERVER = hercules
# TEST_SERVER = zeus
# DEVELOPMENT_SERVER = thor
#
# Note: lines like these with a # in front of it are comments
#

ETL_PATH=/opt/ktp/ETL
```

Setting Memory Parameters for Java

The memomy setting can be changed individually for each kettle script, for example by editing them directly:

```
sudo gvim `locate spoon.sh`
```

The system-wide properties can be set in */etc/profile.d/pentaho.sh*. The default setting for Kettle 6 are

```
export PENTAHO_DI_JAVA_OPTIONS="-Xms1024m -Xmx2048m -XX:MaxPermSize=256m"
```

An excerpt from `man java`:

```
  -Xmsn
     Specifies the initial size, in bytes, of the memory allocation pool. This value must
     be a multiple of 1024 greater than 1 MB. Append the letter k or K to indicate kilo-
     bytes, or m or M to indicate megabytes. The default value is chosen at runtime based
     on system configuration. See Garbage Collector Ergonomics at http://docs.ora-
     cle.com/javase/7/docs/technotes/guides/vm/gc-ergonomics.html
     Examples:
     -Xms6291456
```

```
    -Xms6144k
    -Xms6m

-Xmxn
    Specifies the maximum size, in bytes, of the memory allocation pool. This value must
    a multiple of 1024 greater than 2 MB. Append the letter k or K to indicate kilo-
    bytes, or m or M to indicate megabytes. The default value is chosen at runtime based
    on system configuration.
    For server deployments, -Xms and -Xmx are often set to the same value. See Garbage
    Collector Ergonomics at http://docs.oracle.com/javase/7/docs/tech-
    notes/guides/vm/gc-ergonomics.html
    Examples:
    -Xmx83886080
    -Xmx81920k
    -Xmx80m
```

Windows Remote Desktop (RDP) Setup

Document Author: arho.virkki@vtt.fi

Xrdp is an Open Source Remote desktop Protocol server, which allows you to RDP to your Linux server from Windows machine with the Windows native remote desktop program.

Install RDP to CentOS 7

This is possible with additional repositories (not only EPEL): For details, see: http://www.itzgeek.com/how-tos/linux/centos-how-tos/install-xrdp-on-centos-7-rhel-7.html.

Install RDP to Ubuntu Desktop

First, install *xrdb*

```
sudo apt-get install xrdp
```

And check its status

```
sudo service xrdp status
```

Then add some extra packages

```
sudo apt-get install gnome-panel metacity
```

Edit the *~/.xsession* file as follows

```
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid darkgrey
metacity &
unity-settings-daemon &
gnome-panel &
gnome-terminal
```

The background color can be changed with the *xsetroot -solid <colorname>* command on the terminal, and set permanently by editing the above config. The colors names are same than in CSS, see e.g. http://davidbau.com/colors/.

Due to the missing 3D support, some features of Gnome cannot be used. For example, *unity-control-center* does not launch. Nonetheless, we can change the Desktop theme from command line.

To switch from the default dark (Ambiance) theme to the optional light (Radiance) theme in Terminal, issue

```
gsettings set org.gnome.desktop.interface gtk-theme Radiance
gsettings set org.gnome.desktop.wm.preferences theme Radiance
```

And to restore the default

```
gsettings set org.gnome.desktop.interface gtk-theme Ambiance
gsettings set org.gnome.desktop.wm.preferences theme Ambiance
```

Reference: http://c-nergy.be/blog/?p=5305

R Installation

Document Author: arho.virkki@vtt.fi

Ubuntu 14.04 Server

R Core

Since Ubuntu 14.04 ships with R v3.0.x, and we need the latest and greatest release, we will install the officital CRAN repository from https://cran.r-project.org/ following the instructions at https://cran.r-project.org/bin/linux/ubuntu/

First, uninstall the R version 3.0.x

```
sudo apt-get remove --purge r-base-core
sudo apt-get autoremove
```

Then add the Swedish R CRAN mirror to apt sources

```
sudo su -c 'echo "deb http://ftp.acc.umu.se/mirror/CRAN/bin/linux/ubuntu trusty/" >
    /etc/apt/sources.list.d/r-core.list'
```

Also - according to the documentation, check that Ubuntu backports is enabled in apt to compile certain R packages. The file */etc/apt/sources.list* shoule contain the lines uncommented.

```
deb http://fi.archive.ubuntu.com/ubuntu/ trusty-backports main restricted universe multiverse
deb-src http://fi.archive.ubuntu.com/ubuntu/ trusty-backports main restricted universe multiverse
```

Finally, add the secure APT key

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E084DAB9
sudo apt-get update
```

Then, install R

```
sudo apt-get install r-base r-base-devs
```

Global R Packages

Then, install some useful packages for all users

```
sudo apt-get install libpq-dev      # Needed for RPostgreSQL
sudo apt-get install libssh2-1-dev  # Needed for git2r
sudo apt-get install libxml2-dev    # Needed for xml2

sudo R

install.packages(c("openxlsx","RPostgreSQL","roxygen2"))
install.packages("devtools") # Will imply lots of dependecies
```

Every user can install his/her individual packages in addition to the global ones. These packages will be installed under *~/R/x86_64-pc-linux-gnu-library/3.2/*.

RStudio Server

Following the instructions from https://www.rstudio.com/products/rstudio/download-server/, issue

```
sudo apt-get install gdebi-core
wget https://download2.rstudio.org/rstudio-server-0.99.486-amd64.deb
sudo gdebi rstudio-server-0.99.486-amd64.deb
```

The server is now installed. Users can log in with their *ktpanalytics.vsshp.net* Linux accout on http://ktpanalytics.vsshp.net:8787/.

RStudio server status can be checked and changed from the command line with

```
sudo rstudio-server status
sudo rstudio-server stop
sudo rstudio-server start
```

Shiny Server

Download file from https://www.rstudio.com/products/shiny/download-server/

Then, install shiny package as root

```
sudo R
install.packages("shiny")
```

Download shiny server and install Shiny Server

```
wget https://download3.rstudio.org/ubuntu-12.04/x86_64/shiny-server-1.4.0.756-amd64.deb
sudo dpkg -i shiny-server-1.4.0.756-amd64.deb
```

Or, alternatively with gdebi

```
sudo apt-get install gdebi-cor
sudo gdebi shiny-server-1.4.0.756-amd64.deb
```

Ubuntu Server 14.04 Installation

Document Author: arho.virkki@vtt.fi

Update system and install newer kernel

```
sudo apt-get install --install-recommends linux-generic-lts-vivid
```

Configure static IP

```
sudo nano /etc/network/interfaces
```

Chance the IP from **dhcp**

```
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

into **static**

```
iface eth0 inet static
    address 192.168.1.220
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-nameservers 192.168.1.1
    dns-search vtt.fi
```

Install convenience packages

```
sudo apt-get install tree iotop
```

Extra packages

```
sudo apt-get ...
```

Setting Up VNC Remote Desktops

Document Author: arho.virkki@vtt.fi

Ubuntu Server

Install VNC server with the newer protocol (there are several vnc-server packages in the Ubuntu repositories):

```
sudo apt-get install vnc4server
```

After installation, the command *vncserver* points to *vnc4server* through */etc/alternatives*. Launching *vncserver* first time will ask for setting a password (max. 8 characters). The password can be changed later with *vnc4passwd*.

The following command disables the passwords and sets the X server geometry

```
vncserver -depth 24 -geometry 1280x800  -SecurityTypes None
```

If the X server display number is not given as an argument (for example, *vncserver :10*), the server will choose the next available display from the list 1,2,3,... Once started, the server listens to port 5900 + x, where x is the display number. For example, the default server instance running on display :1 is listening port 5901.

The different server can be stopped individually with

```
vncserver -kill :<display_number>
```

and all at once with

```
killall Xvnc4
```

If vncserver scripts complains about finnish locale setting, add *export LC_ALL=en_US.UTF-8* line to *.bashrc*. This does not affect key mappings.

Then, install the necessary libraries to run *xfc4*, which is a lightweitght desktop environment suitable for remote connections.

```
sudo apt-get install xfce4 xfce4-goodies
```

Then edit the VNC *xstartup* script to choose the windows manager and additional startup programs and settings for the remote X session.

```
nano ~/.vnc/xstartup
```

Example contents:

```
#!/bin/sh

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources

# Start a vnc helper application and then an XFCE4 session
vncconfig -iconic &
startxfce4

# Close the vnc server automatically (for this display)
# if the user logs out from XFCE4
vncserver -kill $DISPLAY
```

Securing the VNC connection

Since the vnc connection is not encrypted, it is better to allow only local connections with the *-localhost* option,

```
vncserver -depth 24 -geometry 1280x800 -localhost -SecurityTypes VncAuth
```

and the use SSH port forwarding to securely connect to the vnc port 5900 + <display number>. The following establishes a connection to the remote server *ktpanalytics.vsshp.net* with port forwarding

```
ssh -L 5901:localhost:5901 ktpanalytics.vsshp.net
```

Now, we can point the vnc client to localhost with

```
vncviewer localhost:1
```

and the connection will be encrypted. Using *-SecurityTypes VncAuth* instead of *-SecurityTypes None* prevents other privileged users from accessing the session (unless they know the VNC password).

Ubuntu Desktop

Since Ubuntu Desktop is simply Ubuntu Server with X (with different packages installed by default, but is available through the same repositories), we do not need to download that many X dependencies. The procedure is similar to Ubuntu Server, but in this case we proceed with gnome-panel and metacity.

```
sudo apt-get install vnc4server gnome-panel metacity
```

Edit the *~/.vnc/xstartup* file as follows

```
#!/bin/sh

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid darkgrey
metacity &
unity-settings-daemon &
gnome-panel &
gnome-terminal &
```

The background color can be changed with the *xsetroot -solid <colorname>* command on the terminal, and set permanently by editing the above config. The colors names are same than in CSS, see e.g. http://davidbau.com/colors/. s

Due to the missing 3D support, some features of Gnome cannot be used. For example, *unity-control-center* does not launch. Nonetheless, we can change the Desktop theme from command line.

To switch from the default dark (Ambiance) theme to the optional light (Radiance) theme in Terminal, issue

```
gsettings set org.gnome.desktop.interface gtk-theme Radiance
gsettings set org.gnome.desktop.wm.preferences theme Radiance
```

And to restore the default

```
gsettings set org.gnome.desktop.interface gtk-theme Ambiance
gsettings set org.gnome.desktop.wm.preferences theme Ambiance
```

Set desktop background image

Install appropriate tool

```
sudo apt-get install feh
```

The, add a line to *~/.vnc/xstartup* similar to

```
feh --bg-fill /usr/share/backgrounds/Sea_Fury_by_Ian_Worrall.jpg
```

More options can be found from *feh* manual pages.

References

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-vnc-on-ubuntu-14-04
http://www.techradar.com/news/software/operating-systems/10-of-the-best-linux-window-managers-90922

CentOS 7

```
sudo su -
yum install tigervnc-server
cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@.service
nano  /etc/systemd/system/vncserver@.service
```

Edit the file as follows, replasing <USER> with an actual user and change geometry if needed.

```
[Service]
...
ExecStart=/sbin/runuser -l arho -c "/usr/bin/vncserver %i -geometry 1440x900 -SecurityTypes None"
PIDFile=/home/arho/.vnc/%H%i.pid
```

Then, reload the daemon settings

```
systemctl daemon-reload
```

Now, exit from root and issue, as an ordinary user,

```
vncpasswd
```

Then, back to root priviliges

```
sudo su -
systemctl start vncserver@:1.service
systemctl enable vncserver@:1.service
```

If you accidentally crippled your VNC session (e.g. by logging out or closed the clipboard extension client), the service can be restarted with

```
sudo systemctl restart vncserver@:1
```

Open Firewall for VNC

Use ssh to log into the system with an X-server enabled computer (e.g. Mac OS X or Ubuntu Linux) and issue `firewall-config`. Then choose: Zone: public, Configuration: Permanent, enable Service *vnc-server*, and reload Firewalld from the Options -menu.

The other option is to enable the firewall from command line

```
sudo firewall-cmd --permanent --add-service vnc-server
sudo systemctl restart firewalld.service
```

Switch Gnome shell style

The user can switch from Gnome Classic to Gnome by logging out and selecting Gnome from the Session list on the login screen. To switch from Gnome Classic to Gnome from within the user session, run the following command:

```
gnome-shell --mode=user -r &
```

To switch back to classic, issue

```
gnome-shell --mode=classic -r &
```

I have not yet found a way to make this permanent. One option is to introduce an alias in *.bashrc*

```
alias go_gnome='gnome-shell --mode=user -r &'
alias go_classic='gnome-shell --mode=classic -r &'
```

And run that every time when the vncserver needs to be restarted (which is not often).

References: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-TigerVNC.html

Connecting a VNC Desktop

Connecting from Windows

Real VNC Viewer: The Real VNC Viewer is free to use. Server can be connected with simplhy the name or IP and display number

```
192.168.1.220:1
```

Virt-Viewer: The Windows version of virt-viewer differs from the Linux software with the same name. Open the softare and connect with

```
vnc://remotebox.vtt.fi:5901
```

where the machine name or IP is followed by the VNC server port (now 5901 = 5900 + 1 for the first display).

Connecting from Linux

Remmina is one of the best remote desktop viewers for linux (http://remmina.sourceforge.net/). For Ubuntu, it is available from software center.

Nonetheless, also Real VNC Viewer is available from https://www.realvnc.com/download/viewer/

There are also command line clients available (that can also do ssh-tunneling directly):

```
xvncviewer -via arho@abscissa.vtt.fi 192.168.1.220:1
xtightvncviewer -via arho@abscissa.vtt.fi 192.168.1.220:1
```

Securing the connection with ssh

The ssh port forwarding can be done with with the command line *ssh* or with Putty in Windows.

From command line, issue

```
ssh -L 5901:localhost:5901 abscissa.vtt.fi
```

after which the connection is done to *localhost*, e.g. *vnc://localhost:5901*.

X2Go Setup

Document Author: arho.virkki@vtt.fi

http://wiki.x2go.org/doku.php/doc:installation:x2goserver https://en.wikipedia.org/wiki/Comparison_of_remote_desktop_software

Client setup, Windows

Download the installer from http://wiki.x2go.org/doku.php and install the client application.

Client setup, Ubuntu

```
sudo apt-get install x2goclient
```

Server side installation, CentOS 7

http://wiki.x2go.org/doku.php/doc:installation:x2goserver

Ensure that EPEL is installed and issue

```
sudo yum install x2goserver
sudo systemctl start x2gocleansessions.service
```

Gnome 3 is not supported. Install eihter XFCE4 or MATE (Gnome 2 fork): http://jensd.be/125/linux/rhel/install-mate-or-xfce-on-centos-7

XFCE:

```
sudo yum groupinstall xfce
```

Server side installation, Ubuntu 14.04

```
sudo add-apt-repository ppa:x2go/stable
sudo apt-get update
sudo apt-get install x2goserver x2goserver-xsession
```

Unity desktop is not supported. Install either XFCE4 desktop

```
sudo apt-get install xfce4 xfce4-session
```

To make a full-blown Xubuntu installation (with all bells and whistles), issue

```
sudo apt-get install xubuntu-desktop
```

The other option is to install MATE (Gnome 2 fork): http://www.omgubuntu.co.uk/2014/08/install-mate-desktop-ubuntu-14-04-lts

```
sudo apt-add-repository ppa:ubuntu-mate-dev/ppa
sudo apt-add-repository ppa:ubuntu-mate-dev/trusty-mate
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install --no-install-recommends ubuntu-mate-core ubuntu-mate-desktop
```

Check the service status

```
sudo service x2goserver status
```

Server side installation, Xubuntu and Ubuntu Mate 16.04

The installation follows the previous procedure

```
sudo add-apt-repository ppa:x2go/stable
sudo apt-get update
sudo apt-get install x2goserver x2goserver-xsession
sudo service x2goserver status
```

At this writing, Xubuntu and Ubuntu Mate 16.04 need one extra tweak (see. http://askubuntu.com/questions/763597/x2go-with-ubuntu-mate-xfce-16-04-fails-to-start). For xfce4, we need the following environment variables:

```
sudo vim /etc/profile.d/x2go_xfce4.sh

# Manually add some paths for x2go
export GSETTINGS_SCHEMA_DIR=/usr/share/xfce4:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
export XDG_DATA_DIRS=/usr/share/xfce4:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
```

The procedure is similar to Ubuntu Mate:

```
sudo vim /etc/profile.d/x2go_mate.sh

# Manually add some paths for x2go
export GSETTINGS_SCHEMA_DIR=/usr/share/mate:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
export XDG_DATA_DIRS=/usr/share/mate:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
```

Customize shortcut keys

http://wiki.x2go.org/doku.php/wiki:advanced:nx-keyboard-shortcuts

On the server side (i.e. at gradient), issue

```
sudo vi /etc/x2go/keystrokes.cfg
```

And change

```
<keystroke action="close_session" Control="1" AltMeta="1" key="t" />
```

into

```
<keystroke action="close_session" Control="1" AltMeta="1" key="0" />
```

To change session termination key from <Ctrl> + <Alt> + t (which is default for opening Terminal application in Ubuntu) into <Ctrl> + <Alt> + 0, which is not already reserved. Finally, log out from the current x2go-session to reload the new settings.

Backup System

Document Author: arho.virkki@vtt.fi

Automated Backups

Wiki, version control system, raw data, PostgreSQL data base and several other localtions are automatically backed up to NAS at */nas/backup*.The cron scheduler at *ktp@gradient* is responsible for running the backup scrtips. For details, see `crontab -l`.

Backup scripts

The backup script are stored in *Common.git* repository and located at *Common/backup/*. Currently, the scripts include several shell and R scripts.

```
.
 data
ăă  backup_git.R
ăă  backup_raw_data.R
 KVM
ăă  backup_all_domains.sh
ăă  backup_quickboot_domain.sh
ăă  backup_reboot_domain.sh
ăă  virsh_shutdown_domain.sh
 postgres
     backup_pg.sh
```

Backing up PostgreSQL

There are several possibilites of making a full backup of a running PostgreSQL cluster. We use now Option 1 below since it was fastest.

Option 1: Run pg_dumpall at gradient.vsshp.net

First, install a matching version of the PostgreSQL *pg_dumpall* tool. (In this case postgresql94 - PostgreSQL client programs and libraries: http://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/repoview/postgresql94.html) The development libraries are not strictly necessary (but are needed for R support, if even needed).

```
wget http://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/postgresql94-9.4.8-1PGDG.rhel7.x86_64.rpm
wget
    http://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/postgresql94-libs-9.4.8-1PGDG.rhel7.x86_64.rpm
wget
    http://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/postgresql94-devel-9.4.8-1PGDG.rhel7.x86_64.rpm
```

Install the packages with

```
sudo rpm -ivh postgresql94-*
```

Ensure that the password for the PostgreSQL root user is set at the database machine. Then, create a *.pgpass* file under the *ktp* user home directory. The generic format for the file is

```
hostname:port:database:username:password
```

In this case, we allow *postgres* user to access all databases (*) with

```
echo "ktppg.vsshp.net:5432:*:postgres:<passwd here>" > .pgpass
chmod og-rwx .pgpass
chmod 0600 .pgpass
```

where the password is set to *ktp* (normal short) password. Try that the arrangement works with

```
psql -U postgres -h ktppg.vsshp.net -d postgres
```

Then, make a copy of the database with

```
today=$(date --iso-8601)
mkdir -p  "/var/local/backup/ktppg/$today"

time /usr/pgsql-9.4/bin/pg_dumpall \
-w -h ktppg.vsshp.net -U postgres -l postgres | lz4 | \
split -a 2 -b 1G - "/var/local/backup/ktppg/$today/pgdump.lz4_"
```

The execution took about 2 hours and 36 minutes as of this writing (2016-08-03)

```
real    156m44.363s
user    48m14.420s
sys     26m17.773s
```

Option 2: Run at gradient but call pg_dumpall at ktppg

Copy the public key of ktp@gradient to ktp@ktppg

```
scp .ssh/id_rsa.pub ktp@ktppg.vsshp.net
```

Then, add the key to the postgres user

```
sudo sh -c "cat id_rsa.pub >> ~postgres/.ssh/authorized_keys"
```

Now we can backup the whole database from ktp@gradient with the following script:

```
#!/bin/bash

# PostgreSQL cluster backup script

# Author(s) : Arho Virkki
# Copyright : VTT Technical Reseach Centre of Finland
# Date      : 2016-07-29

today=$(date --iso-8601)
mkdir -p  "/var/local/backup/ktppg/$today"

time ssh postgres@ktppg.vsshp.net "pg_dumpall | lz4 " | \
split -b 1G - "/var/local/backup/ktppg/$today/pgdump.lz4_"
```

The execution took about three hours (2016-08-03)

```
[ktp@gradient postgres]$ ./backup_pg.sh

real    185m44.510s
user    5m32.734s
sys     4m3.767s
```

Option 3: Run the backup at ktppg.vsshp.net

ktp@ktppg:~$ sudo apt-get install liblz4-tool

Allow

```
sudo su - postgres
ssh-keygen
ssh-copy-id -i .ssh/id_rsa.pub ktp@gradient.vsshp.net
```

The execution took about 3 and half hours (2016-08-03)

```
time sudo -u postgres sh -c \
"pg_dumpall | lz4 | ssh ktp@gradient.vsshp.net \
\"split -a 2 -b 1G - /nas/backup/ktppg/$(date --iso-8601).lz4_\""

real    211m47.591s
user    28m51.380s
sys     15m27.717s
```

Backing up KVM virtual machines

Shut down for maintenance

While it is possible to back up live machines with snapshots, it is safest to power off the machine to ensure a consistent state of the virtual disk. Otherwise, we need to make sure that e.g. no database transactions are running while the snapshot was taken for the backup.

Run the backup script

The backup scripts reside on the *Common* repository under `backup/KVM`, and an instance of *Common* should be found at `ktp@gradient.vsshp.net:/home/ktp/Common`. There are also symbolic links at `/usr/bin` for *sudo* access.

Examples:

```
sudo backup_reboot_domain.sh ktpgit
time sudo backup_reboot_domain.sh ktpgit pbzip2
```

Typical output:

```
[ktp@gradient images]$ time sudo backup_reboot_domain.sh ktptest pbzip2
Waiting for ktptest to shut off..
Backing up ktptest into
/nas/backup/images/2016-01-15_ktptest.xml
/nas/backup/images/2016-01-15_ktptest.qcow2.tar.bz2
Backup done
Domain ktptest started

real    8m6.144s
user    56m55.455s
sys 5m38.021s
```

Ensure that the machine responds to ACPI poweroff

On Ubuntu hosts, save the original script which responds to power button

```
cd /etc/acpi/
sudo mv powerbtn.sh powerbtn_orig.sh
```

and edit the *powerbtn.sh* to only contain the following line

```
#!/bin/shs
/sbin/poweroff
```

to disable any user interactivity required for poweroff (such as the "Would you like to…" prompts).

Why the backup is slow?

A "recent" discussion at *comp.unix.internals* (1990) explains that "…you cannot tell the difference between a hole and an equivalent number of nulls without reading raw blocks…". Hence tar needs to read the whole file, since it is a file-system independent tool (xfs, ext2/3/4, ntfs and nfs all work). For details, see: http://www.delorie.com/gnu/docs/tar/tar_118.html

Appendix A: Tar Performance with Different Compression Levels

For details, see e.g.

- http://www.gnu.org/software/tar/manual/tar.pdf
- http://serverfault.com/questions/66338/how-do-you-synchronise-huge-sparse-files-vm-disk-images-be

Tar with no compression

```
time tar -cSf /nas/backup/images/`date --iso-8601`_ktptest.qcow2.tar \
-C /var/lib/libvirt/images/ ktptest.qcow2

real    9m48.137s
user    2m5.844s
sys     4m35.557s

time tar -xvSf /nas/backup/images/2016-01-14_ktptest.qcow2.tar
ktptest.qcow2

real    2m42.673s
user    0m1.598s
sys     0m43.269s
```

Tar with gzip (I/O bound)

```
time tar -cSzf /nas/backup/images/`date --iso-8601`_ktptest.qcow2.tar.gz \
-C /var/lib/libvirt/images/ ktptest.qcow2

real    17m35.627s
user    13m28.720s
sys     4m21.880s

time tar -xvzf /nas/backup/images/2016-01-14_ktptest.qcow2.tar.gz

real    2m48.644s
user    2m9.246s
sys     1m3.905s
```

Tar with lz4

```
time tar -I lz4 \
-cSf /nas/backup/images/`date --iso-8601`_ktptest.qcow2.tar.lz4 \
-C /var/lib/libvirt/images/ ktptest.qcow2

real    8m8.091s
user    3m3.050s
sys     4m1.563s

time tar -I lz4 -xvf /nas/backup/images/2016-01-14_ktptest.qcow2.tar.lz4

real    2m44.864s
user    0m18.914s
sys     1m12.399s
```

Tar with pbzip2

```
time tar -I pbzip2 \
-cSf /nas/backup/images/`date --iso-8601`_ktptest.qcow2.tar.bz2 \
-C /var/lib/libvirt/images/ ktptest.qcow2

real    8m1.982s
user    54m55.448s
sys     5m28.025s

time tar -I pbzip2 -xvf /nas/backup/images/2016-01-14_ktptest.qcow2.tar.bz2

real    1m42.990s
user    15m34.306s
sys     1m52.933s
```

File size comparison

```
[arho@gradient images]$ ls -lha
-rw-r--r-- 1 arho wheel  14G 14.1. 16:04 2016-01-14_ktptest.qcow2.tar
-rw-r--r-- 1 arho wheel 5,7G 14.1. 22:45 2016-01-14_ktptest.qcow2.tar.bz2
-rw-r--r-- 1 arho wheel 6,0G 14.1. 22:24 2016-01-14_ktptest.qcow2.tar.gz
-rw-r--r-- 1 arho wheel 7,9G 14.1. 21:59 2016-01-14_ktptest.qcow2.tar.lz4
```

Appendix B: Typical Execution Times

Initial sized of the images

```
[ktp@gradient images]$ ls -lhs
total 929G
 59G -rw-r--r-- 1 qemu qemu  2.1T Jan 16 11:17 ktpanalytics.qcow2
 17G -rw-r--r-- 1 qemu qemu  513G Jan 16 11:44 ktpdoc.qcow2
 44G -rw-r--r-- 1 qemu qemu  2.1T Jan 16 11:17 ktpgit.qcow2
523G -rw-r--r-- 1 qemu qemu   11T Jan 16 11:45 ktphadoop.qcow2
270G -rw-r--r-- 1 qemu qemu   11T Jan 16 11:17 ktppg.qcow2
 19G -rw-r--r-- 1 root root  257G Jan 15 15:01 ktptest.qcow2
```

Corresponding execution times

```
[ktp@gradient KVM]$ sudo ./backup_all_domains.sh
This will take long, and automatically reboot all virtual
machines along the way!
Are you sure [y/n]: y
Backing up ktpdoc into
/nas/backup/images/2016-01-15_ktpdoc.xml
/nas/backup/images/2016-01-15_ktpdoc.qcow2.tar.lz4
Backup done
Domain ktpdoc started

real    14m28.701s
user    4m51.211s
sys     8m5.817s

Waiting for ktpgit to shut off..
Backing up ktpgit into
/nas/backup/images/2016-01-15_ktpgit.xml
/nas/backup/images/2016-01-15_ktpgit.qcow2.tar.lz4
Backup done
Domain ktpgit started

real    49m24.771s
user    16m55.121s
sys     29m56.354s

Waiting for ktpanalytics to shut off...
Backing up ktpanalytics into
/nas/backup/images/2016-01-15_ktpanalytics.xml
/nas/backup/images/2016-01-15_ktpanalytics.qcow2.tar.lz4
Backup done
Domain ktpanalytics started

real    55m36.373s
user    18m6.019s
sys     31m46.295s

Waiting for ktppg to shut off...
Backing up ktppg into
/nas/backup/images/2016-01-15_ktppg.xml
/nas/backup/images/2016-01-15_ktppg.qcow2.tar.lz4
Backup done
Domain ktppg started

real    262m9.829s
user    89m34.183s
sys     160m33.088s

Waiting for ktphadoop to shut off...
ing up ktphadoop into
/backup/images/2016-01-16_ktphadoop.xml
/backup/images/2016-01-16_ktphadoop.qcow2.tar.lz4
Backup done
Domain ktphadoop started

    302m29.830s
    98m37.440s
    174m44.484s
```

R Language

Document Author: arho.virkki@tyks.fi

Miscellaneous R scripts and tips

Installation and first steps

For the server environment, see the installation instructions for R Language. For a personal intallation (e.g. for a laptop), download R from https://www.r-project.org/ and follow the instructions. If you have already installed R, Familiarize yourself with the original, official open source manuals:

- https://cran.r-project.org/manuals.html
- https://cran.r-project.org/doc/manuals/r-release/R-intro.html

Brief history

R is based on S-language, which was published in 1976, and t he first version of R appeared in 1993. The source code is under the GNU-license (like the Linux kernel), which means that the language is free and can be used in commercial settings, but not embedded in closed source products. This must just be taken into account, but it does not impose any practical limitations of R use.

There are version for Windows, OS X and Linux systems available at the R web site. R can be extended with custom functions (R scripts), packages from the comprehensive R archive network (CRAN), o wn custom written packages, and C/C++ or Java code, to name a few alternatives. For details, see: https://en.wikipedia.org/wiki/S_(programming_language)

Editors

RStudio R editor, or Rstudio Server
http://www.rstudio.org/

Notepad++, a must have editor for Windows
http://notepad-plus-plus.org/

Eclipse with StatET plug-in
http://www.eclipse.org/
http://www.walware.de/goto/statet/

Emacs with ESS (Emacs Speaks Statistics) plug-in
http://vgoulet.act.ulaval.ca/en/emacs/
http://ess.r-project.org/

Vim with optional Nvim-R extension
http://www.vim.org/download.php/
http://macvim-dev.github.io/macvim/
http://www.vim.org/scripts/script.php?script_id=2628

RStudio is the easiest to begin with, whereas Vim is hard to master, but really powerful.

Vim configuration

Example Vim configuration file for R (`~/.vimrc`);

```
" Configure indentation
set shiftwidth=2
set tabstop=2
" Expand all tabs into spaces
set expandtab

" Configure indentation to work with R
set nocompatible
syntax enable
filetype plugin on
filetype indent on

if has("gui_running")
  set lines=50 columns=80   " Set geometry
  set guioptions-=T         " Remove toolbar
  set guioptions-=r         " Remove right-hand scroll bar
  set guioptions-=L         " Remove left-hand scroll bar
  set guifont=Monospace\ 11 " Set font
endif
```

Find events relative to the same ID in two tables (by dates)

```
library(data.table)


set.seed(200)
```

first table

```
date_table_1 <- data.table(hetu = paste0("hetu_",sample(x = 1:4, size = 12,replace = T)),
                        date1 = Sys.Date() + sample(x = 1:40, size = 12,replace = T),
                        what1 = paste0("value_",sample(x = letters[1:14], size = 12,replace = T)))
```

second table

```
date_table_2 <- data.table(hetu = paste0("hetu_",sample(x = 1:4, size = 12,replace = T)),
                        date2 = Sys.Date() + sample(x = 1:40, size = 12,replace = T),
                        what2 = paste0("value_",sample(x = LETTERS[1:14], size = 12,replace = T)))
```

find all what1 before what2 (without allow.cartesian = T doesnot work)

```
merge(date_table_1,
  date_table_2,
  by = "hetu",
  allow.cartesian = T)[date1<date2]
```

if want first(min) what2 after what1 (see that for date2 can be more than one date1)

```
merge(date_table_1,
  date_table_2,
  by = "hetu",
  allow.cartesian = T)[date1<date2][,.SD[which.min(date2)],list(hetu,date1)]
```

as previous, but take the last(max) date1 (if there are multiple)

```
merge(date_table_1,
  date_table_2,
  by = "hetu",
  allow.cartesian =
      T)[date1<date2][,.SD[which.min(date2)],list(hetu,date1)][,.SD[which.max(date1)],list(hetu,date2)]
```

if want last(max) what1 beforewhat2 (see that for date1 can be more than one date2)

```
merge(date_table_1,
  date_table_2,
  by = "hetu",
  allow.cartesian = T)[date1<date2][,.SD[which.max(date1)],list(hetu,date2)]
```

as previous, but take the first(min) date2 (if there are multiple)

```
merge(date_table_1,
  date_table_2,
  by = "hetu",
  allow.cartesian =
      T)[date1<date2][,.SD[which.max(date1)],list(hetu,date2)][,.SD[which.min(date2)],list(hetu,date1)]
```

Mark words containing some regexp and replace this place as “////” word “\\\\" library(stringr) begin_mark <-”////"

```
end_mark     <- "\\\\\\\\"
word_to_look <- "diagn"
this_string  <- "DIag was very Hoito-tapahtumadiagnoosit and (Diagnoosi)."
str_replace_all(this_string,
            regex(paste0("([a-zääö\\-]*",word_to_look,"[a-zääö\\-]*)"),TRUE),
            paste0(begin_mark,"\\1",end_mark))
# result
[1] "DIag was very ////Hoito-tapahtumadiagnoosit\\\\ and (////Diagnoosi\\\\)."


library(RPostgreSQL)
library(data.table)
library(ggplot2)
```

Load data into R

```
drv    <- dbDriver("PostgreSQL")

con_pg <- dbConnect(drv, dbname="ktp",user = "ktp", host="ktppg.vsshp.net",port=5432 )

diagnoosit <- data.table(dbGetQuery(con_pg," SELECT * from stage_uraods.mv_palvelu_diagnoosi where
    yksikko_nimi ~ '^K' and dg_syy_koodi is not null limit 10000"))

tilastot <-diagnoosit[,.N,list(kliniikka = yksikko_nimi,
                               diagnoosi =   substr(dg_syy_koodi,1,3),
                               vuosi     = year(alkuhetki_pvm))][N>20][order(diagnoosi)]
ggplot() + geom_bar(data = tilastot,

                aes(x = kliniikka,
                    y = N,
                    fill = diagnoosi),
                stat = "identity") + facet_grid(vuosi~.)
```

PostgreSQL Server and Database Administration

Document author: anna.hammais@tyks.fi and arho.virkki@tyks.fi

- PostgreSQL setup at Gradient
- Troubleshooting PostgreSQL at Gradient
- R Language (PL/R) and Shell (PL/sh) extensions
- Logins as specific user
- psql commands
- Manage databases and users
- Processes on PostgreSQL server
- Logging queries
- PostgreSQL server configuration
- Backup and restore
- SQL syntax
- JSON
- Adding auto-increment primary key after table creation
- Generating a date series
- Encrypt and Decrypt data with a symmetric AES key
- Vacuum PostgreSQL database manually
- Postgres_fdw (Foreign Data Wrapper) and dblink

Adding serial primary key to a table after table creation

Document author: anna.hammais@tyks.fi

Create a sequence:

```
create sequence stage_labdw.labdw_transform_id_seq increment by 1;
```

Examine a sequence:

```
select * from stage_labdw.labdw_transform_id_seq;
```

Change a column to auto-increment:

```
alter table stage_labdw.labdw_transform rename to labdw_transform_old;

create table stage_labdw.labdw_transform as
select (row_number() over())::int labdw_transform_id,
l.*
from stage_labdw.labdw_transform_old as l;

select setval('stage_labdw.labdw_transform_id_seq',
(select max(labdw_transform_id) from stage_labdw.labdw_transform;

alter table stage_labdw.labdw_transform alter labdw_transform_id set not null;

alter table stage_labdw.labdw_transform alter labdw_transform_id
set default nextval('stage_labdw.labdw_transform_id_seq'::regclass)

alter table stage_labdw.labdw_transform ADD PRIMARY KEY (labdw_transform_id);
```

Backup and restore

Document author: anna.hammais@tyks.fi

Dumping all tables in a specific schema into a plain-text file:

```
#!/bin/bash

# -n = name of schema
# ktp = name of database
# -f = name of output file
# date -I'date' = current timestamp in ISO format, date part only

pg_dump -n 'luovutusrekisteri' ktp -f '~/$(date -I'date')_gradient_luovutusrekisteri_backup.sql'
```

Restoring tables from the plain-text backup file:

```
psql database_name < backup_file.sql
```

Encrypt and Decrypt data with AES

Document author: juha-matti.varjonen@tyks.fi

Encryption and decryption routines in PostgreSQL aes and ascii

Encrypt from text into byte array

```
SELECT encrypt('secret','pass1','aes');
```

Convert to ascii

```
SELECT encode( encrypt('secret','pass1','aes'), 'base64');

           encode
--------------------------
  YJAGfjZ3NZliNc02TH+q+w==
```

Decode back

```
SELECT convert_from(decrypt( decode('YJAGfjZ3NZliNc02TH+q+w==','base64'), 'pass1', 'aes'),'utf8');
```

In summary:

```
SELECT convert_from(decrypt( decode(
  (SELECT encode( encrypt('tosi salainen juttu','pass1','aes'), 'base64')),
  'base64'), 'pass1', 'aes'),
  'utf8');
```

By using Encrypt fuction func.henkilotunnus_to_pseudonym_aes

```
select * from func.henkilotunnus_to_pseudonym_aes(_henkilotunnus := 'secret',_crypt := TRUE,_salt
    := 'pass1') as pseudonym_aes;
```

By using decrypt function func.pseudonym_to_henkilotunnus_aes

```
select * from func.pseudonym_to_henkilotunnus_aes(_pseudonym := 'YJAGfjZ3NZliNc02TH+q+w==', _salt
    := 'pass1') as henkilotunnus;
```

Generating a date series

Document author: anna.hammais@tyks.fi

Simple date series:

```
select
series + date '2003-01-01' as myDate
from generate_series(0,6000) as series;
```

Example query:

```
select md5(a.henkilotunnus),
p.palvelu_numero,
a.syntymaaika_pvm::date as birth,
a.kuolinaika_pvm::date as death,
alkuhetki_pvm::date as ward_start,
alkuhetki_pvm::date + date_series as ward_date,
loppuhetki_pvm::date as ward_end,
vo_toimipiste_koodi as unit_code,
yksikko_nimi as unit_name,
case when alkuhetki_pvm::date <= kuolinaika_pvm::date
  and kuolinaika_pvm::date <= loppuhetki_pvm::date
    then 1
    else 0 end
as died_on_this_ward

from stage_uraods.mv_palvelu as p
inner join stage_uraods.v_asiakas as a on p.henkilotunnus = a.henkilotunnus
, generate_series(0,6000) as date_series
where palvelumuoto = 'osastohoito'
-- include only dates that are between the ward start and end dates
and alkuhetki_pvm::date + date_series <= loppuhetki_pvm::date
and md5(a.henkilotunnus) = 'e1db13203ed29aa030b305745e7f223c'
order by p.palvelu_numero, ward_date;
```

PostgreSQL JSON

Document author: anna.hammais@tyks.fi

JSON data types

- **json**: stored as string, faster input, slower searches
- **jsonb**: stored as binary, faster searches, slower input, indexable

Logging queries in PostgreSQL

Document author: anna.hammais@tyks.fi

Setting PostgreSQL config options for session

```
load 'auto_explain';

set auto_explain.log_min_duration=0;

set auto_explain.log_analyze = true;

set auto_explain.log_nested_statements = ON;
```

Then, run your query in the session. Then, find the log file (on gradient, it's at /srv/pgsql/9.6/data/pg_log/).

Login as specific user

Document author: anna.hammais@tyks.fi, arho.virkki@tyks.fi

First ssh to ktppg.vsshp.net, then issue

```
psql -U <username> [--password]
```

Login as superuser

```
sudo su - postgres
psql
```

Automated login inside scripts

```
PGPASSWORD=<passwd here> psql -U <usernamer> -h <host> -d <database>
```

For example, list all schemas in ktp database to standard output, then quit

```
echo "\dn" | PGPASSWORD=<passwd here> psql -U ktp -h gradient
```

Manage PostgreSQL databases and users

Document author: anna.hammais@tyks.fi

Create databases

Create database

```
CREATE DATABASE <database_name> WITH OWNER = <role_name>;
```

To simplify privilege management, revoke all rights from the public role in the new database, so that they cannot be inherited by new roles that will be created in the future

```
REVOKE ALL ON DATABASE <database_name> FROM PUBLIC;
REVOKE ALL ON SCHEMA <database_name>.public FROM PUBLIC;
```

Create and manage users/roles

Login as superuser. Create a user:

```
CREATE ROLE <username> WITH LOGIN ENCRYPTED PASSWORD <pwd_in_single_quotes>;
```

The following are equivalent

```
CREATE ROLE username WITH LOGIN;
CREATE USER username;
```

Check users, roles and privileges

```
psql=> \du -- roles and users
psql=> \dp [<schema_name>.* | <table_name>]     -- user privileges for a specific schema or table
```

In effect, the latter only shows the privileges for a table that are not in place by default. For example, the table owner has all privileges by default, and those are not shown here.

Similarly, as a query:

```
select * from pg_roles;
select * from information_schema.role_table_grants where grantee = '<role_name>';
```

Note that the last query only works on tables, not views or materialized views.

Grant privileges

Grant privileges on a specific schema. First, allow the user to list the objects in the schema, and then assign privileges

```
GRANT CONNECT ON DATABASE <database_name> TO <role_name>;
GRANT USAGE ON SCHEMA <schema_name> TO <role_name>;
GRANT SELECT ON ALL TABLES IN SCHEMA <schema_name> TO <role_name>;  -- all tables
GRANT SELECT ON TABLE <table_name> TO <role_name>;                  -- one table
```

It is also possible grant privileges on objects that will be created in the future. However, this applies to tables and views but NOT materialized views. For materialized views, privileges have to be granted explicitly after a new view is created.

```
ALTER DEFAULT PRIVILEGES IN SCHEMA <schema_name> GRANT SELECT ON TABLES TO <role_name>;
```

Altered default privileges can be checked with psql command `\ddp`. They can be revoked (reset to default) by:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA <schema_name> REVOKE SELECT ON TABLES FROM <role_name>;
```

According to PostgreSQL documentation: “If you wish to drop a role for which the default privileges have been altered, it is necessary to reverse the changes in its default privileges or use DROP OWNED BY to get rid of the default privileges entry for the role.”

Revoke privileges

```
revoke select on table <schema>.<table> from <user>;
```

PL/R extension in PostgreSQL 9.6

Document Author: arho.virkki@tyks.fi

For details, see the PL/R manual: http://www.joeconway.com/plr/doc/index.html

Prerequisites

One needs *super user* priviliges to ***create new R functions*** in PostgreSQL. Once the function is created, also non-privileged users can use it.

For deploying new functions, we set up a new **ktp_adm** user (having the same password as the ordinary *ktp* user).

```
CREATE ROLE ktp_adm LOGIN SUPERUSER PASSWORD '<passwd_here>';
```

Examples

R code can contain comments (starting with #) and empty lines. If empty lines produce an error, check that the function definition is passed to PostgreSQL in one batch (as *psql* `\i <filename.sql>` command does by default).

Vector argument, double precision output

```
-- The function is created into 'test' schema. It accepts
-- a double precision postgresql vector 'a' and returns
-- a double precision scalar

CREATE OR REPLACE FUNCTION test.array2double( a float8[] )
RETURNS float8 AS
$$
median ( a )
$$ LANGUAGE plr;

SELECT test.array2double( '{1,2,5}'::float8[] );
```

Single row output

```
CREATE OR REPLACE FUNCTION test.get_table()
RETURNS setof int4 AS $$
array(1:10)
$$ LANGUAGE plr;

SELECT test.get_table();
```

Table output from a data.frame

```
-- One option it to define a table and return a row set of that type

CREATE TABLE IF NOT EXISTS test.integertable (
  val1 int4,
  val2 int4 );

CREATE OR REPLACE FUNCTION test.get_table(n int4 default 10, m text default 'foo')
RETURNS setof test.integertable AS
$$
data.frame(1:n, n:1)
$$ LANGUAGE plr;

SELECT * FROM test.get_table(12);
SELECT * FROM test.get_table(m:='fifi');
```

```
-- Other option is to build the output record format in the
-- function definition

CREATE OR REPLACE FUNCTION test.get_table2(
  IN n int4,
  OUT val1 int4,
  OUT val2 int4)
RETURNS setof record AS $$
data.frame(1:n, n:1)
$$ LANGUAGE plr;

SELECT * FROM test.get_table(25);
```

Read table into a data.frame

```
CREATE OR REPLACE FUNCTION test.table_reader( table_name text )
RETURNS text AS
$$
#
# The queries follow RPostgreSQL package syntax with the
# exception that the connection object is neglected (and hence NA here)
#
X <- dbGetQuery( NA, paste("SELECT * FROM", table_name))
infotext <- paste0("nrow: ", nrow(X), " ncol: ", ncol(X), " colnames: ",
                   paste( colnames(X), collapse=",") )
#
return( infotext )
$$ LANGUAGE plr;

SELECT * FROM test.table_reader( 'test.koodisto_ods' );
```

Working with the Date type

```
DROP FUNCTION IF EXISTS delays.event_delays() ;
CREATE FUNCTION delays.event_delays(
  OUT id int4,                          -- Observe how
  OUT pid text,                         -- the output table is
  OUT event_idx int4,                   -- defined here
  OUT event text,
  OUT event_type text,
  OUT event_info text,
  OUT event_date date,                  -- Date type is 'date' in PosgreSQL
  OUT decision_date date,
  OUT first_treatment bool,
  OUT treatment_delay int4 )
RETURNS setof record AS
$$
con <- NA

# Detect first treatment events with plain SQL
events <- dbGetQuery(con, "
SELECT
    <QUERY HERE>
")
                                    ## In the following, we need to cast
                                    ## dates explicitly into 'Date' (as
                                    ## they where read as 'character')

events$decision_date <- as.Date(events$decision_date)
events$event_date <- as.Date(events$event_date)

   <R COMPUTATIONS>
                                    ## Finally, the internal R 'Date' types
                                    ## are cast back to 'character' for
                                    ## PostgreSQL to parse it as 'date'

events$decision_date <- as.character(events$decision_date)
events$event_date <- as.character(events$event_date)

return( events )                    ## Finally, return a data frame that
                                    ## Matches the FUNCTION definition

$$ language plr;
```

Installation

For compiling the extension, PostgreSQL command line tools need to be in path:

```
# this is a good place to put own customizations
cd /usr/local/bin/
sudo ln -s /usr/pgsql-9.6/bin/* .
```

Download the latest release of PL/L from https://github.com/postgres-plr/plr/releases

```
wget https://github.com/postgres-plr/plr/archive/REL8_3_0_17.tar.gz
tar xvzf REL8_3_0_17.tar.gz
cd plr-REL8_3_0_17
USE_PGXS=1 make

sudo bash -c "PATH=$PATH:/usr/local/bin/ USE_PGXS=1 make install"
```

Now we can install the extension to the desired databases:

```
sudo su - postgres
psql
\c ktp
CREATE EXTENSION plr;
-- Optionally, to get rid of the extension, issue:
-- DROP EXTENSION plr;
```

Now test the extension

```
SELECT * FROM plr_environ();
SELECT load_r_typenames();
SELECT * FROM r_typenames();
SELECT plr_array_accum('{23,35}', 42);
```

Test the installation

Example of a function (under test schema):

```
--
-- Create this function with 'kpt_adm' user
--
CREATE OR REPLACE FUNCTION test.r_max (a integer, b integer)
RETURNS integer AS '
    if (a > b)
       return(a)
    else
       return(b)
' LANGUAGE 'plr';
```

Test the function

```
-- Some test data
CREATE TABLE test.maxtest (
  id SERIAL,
  i INTEGER,
  j INTEGER);
INSERT INTO test.maxtest ( i, j ) VALUES
  ( 1, 3),
  ( 5, 2),
  (-9,-1);

-- Ordinary users can now use the function
SELECT id, test.r_max(i,j) AS maximum FROM test.maxtest;
```

PL/R extension in PostgreSQL 9.6

Document Author: arho.virkki@tyks.fi

The installation, requirements, and usage is similar to R Language extension (PL/R).

For details, see the PL/R Github page: https://github.com/petere/plsh

Installation

```
wget https://github.com/petere/plsh/archive/master.zip
unzip master.zip
cd plsh-master/
make
make install
sudo bash -c "PATH=$PATH:/usr/local/bin/ make install"
```

In the database, enable the extension with

```
CREATE EXTENSION plsh;
```

Examples

```
CREATE OR REPLACE FUNCTION delays.update_event_delays () RETURNS text
LANGUAGE plsh
AS $$
#!/bin/sh
./R/compute_delays2.R 2>/dev/null
echo $?
$$;
```

Postgres_fdw (Foreign Data Wrapper) and dblink

Document author: juha-matti.varjonen@tyks.fi

Following instruction descripts how to set up postgres_fdw (Foreign Data Wrapper) and dblink between two database in gradient instance:

Open superadmin (postgres) connection to ktp database

```
CREATE EXTENSION postgres_fdw;
CREATE EXTENSION dblink;
CREATE FOREIGN DATA WRAPPER to_destination_fdw VALIDATOR postgresql_fdw_validator;
-- where 'research' is target database
CREATE SERVER destination_server FOREIGN DATA WRAPPER to_destination_fdw OPTIONS (hostaddr
    '127.0.0.1', port '5432', dbname 'research');
--where '*' is password for 'ktp' account
CREATE USER MAPPING FOR ktp SERVER destination_server OPTIONS (user 'ktp', password '*');
GRANT USAGE ON FOREIGN SERVER destination_server TO ktp;
```

Open superadmin (postgres) connection to research database

```
CREATE EXTENSION postgres_fdw;
-- where 'ktp' is source database
CREATE SERVER source_server FOREIGN DATA WRAPPER postgres_fdw  OPTIONS (hostaddr '127.0.0.1', port
    '5432', dbname 'ktp');
-- where '*' is password for 'ktp' account
CREATE USER MAPPING FOR ktp SERVER source_server OPTIONS (user 'ktp', password '*');
GRANT USAGE ON FOREIGN SERVER source_server TO ktp;
```

Verify that connection works between databases:

Open normal (ktp) connection to ktp database:

```
SELECT dblink_connect('connection', 'destination_server');
SELECT dblink_send_query('connection', 'CREATE SCHEMA IF NOT EXISTS tutkimus_1');
SELECT dblink_disconnect('connection');
SELECT dblink_connect('connection', 'destination_server');
SELECT dblink_send_query('connection', 'IMPORT FOREIGN SCHEMA func LIMIT TO (mv_table_info) FROM
    SERVER source_server INTO tutkimus_1');
SELECT dblink_disconnect('connection');
SELECT dblink_connect('connection', 'destination_server');
SELECT dblink_send_query('connection', 'CREATE TABLE tutkimus_1.mv_table_info_transferred as
    (select * from tutkimus_1.mv_table_info)');
SELECT dblink_disconnect('connection');
```

Installing PostgreSQL 9.6 on CentOS 7

Document author: arho.virkki@tyks.fi

Dowload the latest PostgreSQL YUM repository package from

```
https://yum.postgresql.org/
```

Adapt the following line to point to the latest release:

```
sudo rpm -Uvh
    https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
sudo yum update
```

Install postgresql

```
sudo yum install postgresql96 postgresql96-server postgresql96-contrib \
postgresql96-devel postgresql96-libs
```

Initialize the database

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

Start the service and turn it on also after reboot (see sudo chkconfig –list for all services)

```
sudo systemctl enable postgresql-9.6
sudo systemctl start postgresql-9.6
```

Try how the database works (postgres user can connect to the db using psql)

```
sudo su - postgres
psql
```

To install PostgreSQL Adminpack, enter the command in postgresql prompt:

```
CREATE EXTENSION adminpack;
```

Configure PostgreSQL-MD5 Authentication

By default, Posgresql uses ident authentication, so that the local system users can be granted access to databases own by them. If you want to set MD5 authentication to require users to enter passwords.

Open host-based authentication file:

```
sudo vim /var/lib/pgsql/9.6/data/pg_hba.conf
```

Add the following lines to the end of the file

```
# Connections from vsshp
host    all             all             10.150.0.0/16           md5
host    all             all             10.145.0.0/16           md5
```

Then, edit the file PostgreSQL conf file:

```
sudo vim /var/lib/pgsql/9.6/data/postgresql.conf
```

Change the line:

```
#listen_addresses = 'localhost'
#port = 5432
```

to

```
listen_addresses = '*'
port = 5432
```

and also change the line:

```
#password_encryption = on
```

to

```
password_encryption = on
```

Now, restart postgresql service to apply the changes:

```
sudo systemctl restart postgresql-9.6
```

Moving PostrgeSQL installation directory

Since CentOS systemctl script contain hard-coded paths to */var/lib/pgsql*, it is easiest to move the whole directy and create a symbolic link to point to the new location.

```
sudo systemctl stop postgresql-9.6

sudo mv /var/lib/pgsql /srv/
ln -s /srv/pgsql /var/lib/pgsql

sudo systemctl start postgresql-9.6
```

Adjust PostgreSQL configuration parameters

Check the PostgreSQL settings with

```
SHOW all;
SHOW effective_cache_size ;
```

Read about the recommended values (set in *postgresql.conf*) from https://www.postgresql.org/docs/9.6/static/runtime-config-resource.html

sudo vim /var/lib/pgsql/9.6/data/postgresql.conf

```
shared_buffers = 8192MB
temp_buffers = 32MB
maintenance_work_mem = 1024MB
```

Processes on the PostgreSQL server

Document author: anna.hammais@tyks.fi

Existing connections

List all database connections

```
select * from pg_stat_activity;
```

Table locks

A pending transaction may hold a lock on a table, preventing other transactions from accessing it. Checking the locks on a specific table:

```
select l.* from pg_locks l join pg_class t on l.relation = t.oid
where t.relkind = 'r' and t.relname = <your_table_name>;
```

This shows which processes are holding locks.

If you wish to know more about the process that is holding the lock, take the PID and see what the process is:

```
ps -f <your_PID>
```

You can kill it by:

```
sudo kill <your_PID>
```

Sometimes killing the process that is the original cause of the problem can remove all subsequent locks on the table as well.

RAM and CPU usage of Postgres processes

```
ssh ktp@ktppg
ps -u postgres uf
```

Last column of output shows whether the process is idle or not.

Disk space report

```
df -h
```

Disk IO stats

```
sudo iotop -aoP
```

-a Will show accumulated output -o Will only output processes/threads doing IO -P Will only show processes instead of threads

PostgreSQL server configuration

Document author: anna.hammais@tyks.fi

Useful options to cofigure in the PostgreSQL server, according to https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server:

```
select * from pg_settings
where name in
('shared_buffers',
'checkpoint_segments',
'effective_cache_size',
'checkpoint_completion_target',
'maintenance_work_mem',
'autovacuum_work_mem',
'wal_buffers',
'config_file');
```

- shared_buffers – 25% of system memory (default very small)
- checkpoint_segments – default 3, set to 20
- effective_cache_size – 50% of system RAM (not hard allocation, just an estimate of max needs)
- checkpoint_completion_target – set to 0.9
- maintenance_work_mem – set to 64MB
- autovacuum_work_mem – -1, which means to use maintenance_work_mem
- wal_buffers – default 1/32 of shared buffers, which is ok

These changes should be entered in the *postgresql.conf* file, located by starting psql as superuser on the ktppg server:

```
ssh ktp@ktppg.vsshp.net
sudo su -
psql

psql> show config_file;
```

The PostgreSQL server must be restarted for these changes to take effect. You can check this by re-running the above select query after the restart.

psql commands

Document author: anna.hammais@tyks.fi

List psql commands

```
psql=> \?
```

See also http://postgresguide.com/utilities/psql.html for a brief introduction.

Show databases

```
psql=> \l+                          -- shows databases and their space usage
psql=> \c[onnect] <database_name> -- connects to specific database
psql=> \conninfo                    -- shows which database you are currently connected to, and as
    which user
```

PostgreSQL syntax

Document author: anna.hammais@tyks.fi

Regexp replace (‘g’ means replace all occurrences)

```
regexp_replace(<column_name>, '\n', '|n|', 'g')
```

Add primary key

```
ALTER TABLE stage_uraods.diagnoosi ADD CONSTRAINT diagnoosi_pkey PRIMARY KEY ( dgn_numero );
```

Troubleshooting PostgreSQL at Gradient

Authors: arho.virkki@tyks.fi, anna.hammais@tyks.fi

Jos PostgreSQL ei vastaa kyselyihin:

1. iotop: levynkirjoitus- tai lukuprosessit
2. iftop (interface top): verkkoliikenne
3. top: prosessit, RAM ja CPU
4. PostgreSQL-prosessin tila

```
sudo systemctl status postgresql.service      # näyttää tilan
sudo systemctl stop postgresql.service        # pysäyttää palvelun
sudo systemctl start postgresql.service       # käynnistää palvelun
```

5. jos prosessi sanoo “failed”, PostgreSQL on kaatunut
6. Jos stop ja start ei onnistu, katsotaan mitä prosesseja postgres-käyttäjällä on meneillään

```
ps -elf | grep postgres                       # (sama kuin ps aux)
```

7. kokeillaan

```
ps pkill <pid>
ps kill -9 <pid>
```

8. jos kohta 7 ei onnistu, mennään katsomaan hakemistoon /srv/pgsql/9.6 (tai mikä versionumero onkaan) (srv=service). Pystyykö hakemistossa tekemään ls? Jos ei, levyjako on rikki.
9. Pystyykö levyjaolle tekemään umount? Jos ei, ajetaan

```
lsof | grep postgres                          # list open files used by postgres
```

10. Jos listattuja prosesseja ei pysty sulkemaan, ajetaan levyjaolle umount -fl (force, lazy)
11. Jos tämän jälkeen mount ei onnistu, käynnistetään gradient-palvelin uudelleen, kunhan käyttäjät ovat tallentaneet keskeneräiset prosessinsa

Vacuum PostgreSQL database manually

Document author: juha-matti.varjonen@tyks.fi

There are two reason why the database tables should be Vacuumed and Analyzed.

VACUUM = Is to have free space available in tables (fsm = free space map).

ANALYZE = You need to analyze the database so that the query planner has table statistics it can use when deciding how to execute a query.

- You can setting up Auto Vacuum (by default it is on and shouldn't ever turned off)
- Or if you see that Auto Vacuum doesn't work perfectly e.g. tables n_dead_tup sizes increase, you can use Manual Vacuum Analyze script explained below.

The shell script is stored in Git repository:

```
ktpgit.vsshp.net:/opt/git/Common.git
```

Under this repository there is subfoder ~/Common/script/shell where you can locate file vacuum_analyze_and_reindex_procedure.sh . The script must be executed by ktp@gradient: (please note that reindex part is commented out and not tested as it might take so long time to execute):

```
~/Common/script/shell/vacuum_analyze_and_reindex_procedure.sh
```

This script check all tables from func.mv_table_info and compare this information to pg_stat_all_tables for getting the n_live_tupe and n_dead_tup information for those tables. The VACUUM ANALYZE script will be execute for certain table if relation of n_dead_tup (dead rows) and n_live_tup (row count) is more or equal to 4%. Otherwise table is skipped from VACUUM ANALYZE

Visualization Tools

Document Authors: juha.pajula@vtt.fi, arho.virkki@tyks.fi, anna.hammais@tyks.fi

Tools under development

- KTP portal
- Views under development (FIN)
- Views under development (ENG)

Reviews

- http://www.creativebloq.com/design-tools/data-visualization-712402/1
- http://inspire.blufra.me/big-data-visualization-review-of-the-20-best-tools/

Comparisons

- http://www.fusioncharts.com/javascript-charting-comparison/

Tools

- Serene Development

D3.js based (ordered by project activity)

1. https://d3js.org/ (Revised BSD, see D3 and BSD licenses)
2. http://emberjs.com/ (MIT license; see Ember and MIT and Revised BSD for D3.js)
3. http://raw.densitydesign.org/ (LGPLv3 and Revised BSD for D3.js)
4. http://nvd3.org/ (Apache 2.0 and Revised BSD for D3.js)
5. http://processingjs.org/ (MIT license like this and Revised BSD for D3.js)

Other JavaScript libraries

1. dygraphs
2. Flot

Commercial (but free-of-charge for non-profit and personal use)

- http://www.fusioncharts.com/
- http://www.highcharts.com/

PaaS

- https://plot.ly/feed/

Packages

- https://www.filamentgroup.com/lab/update-to-jquery-visualize-accessible-charts-with-html5-from-des
  html
- https://github.com/n3-charts
- http://sigmajs.org/

Creating custom visualizations

- Creating custom visualizations

Creating custom visualizations

Document author mikko.koskinen@tyks.fi

Overview

Building custom visualizations is needed when none of the platforms available tools are able to create the required visualization. Building custom visualizations can be done using a variety of librarys, but when actual custom work is required it is usually beneficial to use a fairly low level libraries such as d3.js in javascript.

Links

- d3.js data visualization library
- Python flask backend
- pentaho CDE
- Building custom visualizations in Pentaho

Workflow

It is a good practice to start from simple building blocks and add more complexity as you advance. This workflow takes the visualization design as a starting point

1 Draw the visualization

This is the most important part of the workflow. This determines the design of your visualization.

It is a good practice to write your visualization code straight into a minimal html template. That way is is easy to debug in the browser. You can launch a simple python webserver from the folder containing you visualization html file with `$python3 -m http.server` and navigating to localhost, port 8000 `127.0.0.1:8000` on your browser.

When building your visualization code do not forget to save the changes into the code and make a complete refresh of the page with `ctrl + shift + r`. The browser cache has a tendency to cause problems otherwise.

Make your data entry as easy as possible at this stage by using csv, json etc. files locally. This data can be real or mock data as long as the data format and complexity are comparable. Concentrate only on getting the visualization as you want it to be.

If making custom visualizations is a recurring practice it saves time to build some boilerplate templates containing the html atleast. This saves time when lauching new projects.

2 Add backend

If the data is updated from the server during the use of the visualization, at some point a backend API should be implemented.

At this time it is usually good to have a simple database copy or data files that the backend has access to.

The aim is to fine tune the functionality of the frontend. A good method is to build a minimal backend server with python flask and create the required API endpoints there. Both the webserver and the backend server can still be run locally (and data stored locally).

At this stage you can fine tune the calls to the backend and the following functionality of the visualization. You will also determine exactly the responses that you need from the backend APIs. You should not need to do any data manipulation in the frontend. All that should be done in the backend.

If mock data is still used by the backend server, it can now be compared to the real data and the specs for the backend (and source system) processing become clear. Do not change the frontend code to suit the backend. The frontend visualization code should be completely separate and only dictate the format and shape of the data it receives from the backend. This will also retain modularity as either the backend solution or the visualization can be changed independently as long as the API specs remain the same.

3 Connect to the framework

After the API calls have been fully developed and tested as well as the visualization code, it will probably be deployed into a larger system. This could be a web application built with a Javascript (ReactJS, Angular, etc.), or any other system used to display the visualization. It does not really matter, whether the backend uses relation-object transfer, direct sql queries or some other, more complicated, data pipeline. That should not affect your visualization (everything should work as long as the API specs are correct).

At this stage the visualization code should be moved into the deployment repo and tested still. If there are some minor changes to be made into the code considering data import (best practice is to not have any) they should be implemented here. If a testing mode is available it should be used to see everything works in the context. When everything is fine the visualization can be put into production.

4 Custom visualizations in Pentaho

Pentaho offers a way to create custom javascript visualization to their dashboard application in a sandbox environment. This differs somewhat from other frameworks we have used since data is retrieved directly in pentaho.

A good starting point is still to follow at least step one locally. Once the visualization is, done clone the repo:

https://github.com/pentaho/pentaho-engineering-samples

And follow the instructions in this tutorial:

http://pentaho.github.io/pentaho-platform-plugin-common-ui/platform/visual/samples/bar-d3-sandbox/step1-environment-preparation

One of the biggest hurdles in transferring the visualization into pentaho is the fact that data comes from pentahos own relation-object transformation so you'll have to make a model file and and introduce the model into the visualization (this is the data import into the visualization). Anyone doing the visualizations should work closely with the ETL/data managers to ensure that the data is available in the right form through the pentaho platform.

An overview of the pentho visualization API is found in:

http://pentaho.github.io/pentaho-platform-plugin-common-ui/platform/visual/

When everythiong works in the sandbox environment and you proceed to deployment the instruction of recursive find and replace is an overkill. You'll need to change the name of the visualization in three different pom.xml files that are easily found and edited manually. Remember to be carefull with the visualization name and version number. Otherwise just follow along the instructions and everything should be fine.

Dashboard indicators fin

Document author: amikko.koskinen@tyks.fi

Medical directors view	Cancer center directors view	Clinic directors view
Todays events	Todays events	Todays events
Total number of current patients	Percentage of treatable patients	Current patient number
Patient place utilization percentage	Percentage of patients without treatment	Utilization percentage of patient places
Care personnel utilization percentage	TNM-class distribution at the time of diagnosis	Distribution of care personnel and facility resources per disease type
General customer (patient) satisfaction	Pathologically confirmed diagnosis per cancer type	Weekly medicine costs per disease type
Mean waiting period to receive care	Genetically confirmed diagnosis/determination of cancer subtype per cancer type	Weekly material costs per disease type
Todays patient amounts per clinic	Imaging studies per cancer type	Mortality during care
Current distribution of patients and care personnel between clinics	Mortality during care	Mortality after care ended
Rough budget finance guide	Mortality after care ended	Life quality during care
	Life quality during care	Life quality after care ended
	Life quality after care ended	Complications during care
	Complications during care	Complications after care ended
	Complications after care ended	Waiting times to main treatment procedures of the clinic

Medical directors view	Cancer center directors view	Clinic directors view
	Waiting times to receive treatment	

KTP portal

Document authors: mikko.koskinen@tyks.fi, anna.hammais@tyks.fid

Ruby & Gems installation on Windows

Install Ruby: http://rubyinstaller.org/

Start Ruby terminal and check your ruby version:

```
ruby -v
```

Clone ktp_interface.git:

```
git clone ktp@ktpgit.vsshp.net:/opt/git/ktp_interface.git
```

If it doesn't already exist, create a Gemfile inside your project (in this case, in ktp_interface/ktp_front_end_files) and write this in it:

```
source 'https://rubygems.org'
gem 'rack'
```

```
gem install bundler
cd ktp_interface/ktp_front_end_files
bundle install
```

Angular webserver with rack on your own computer

https://github.com/codingforentrepreneurs/Guides/blob/master/all/angular_webserver.md

When everything is installed, move to the directory ktp_interface and start the application by running:

```
rackup
```

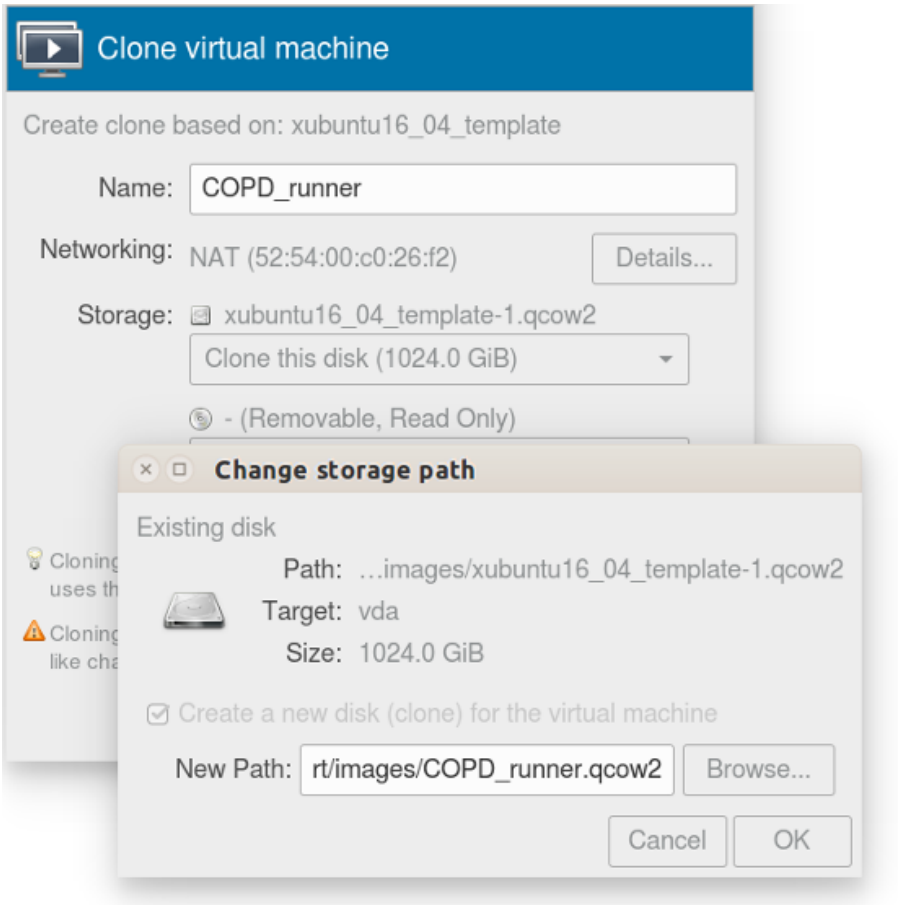
This will show you the gate number to connect to on localhost. Then open Chrome or Firefox and type the address:

```
localhost:<gate_number>
```

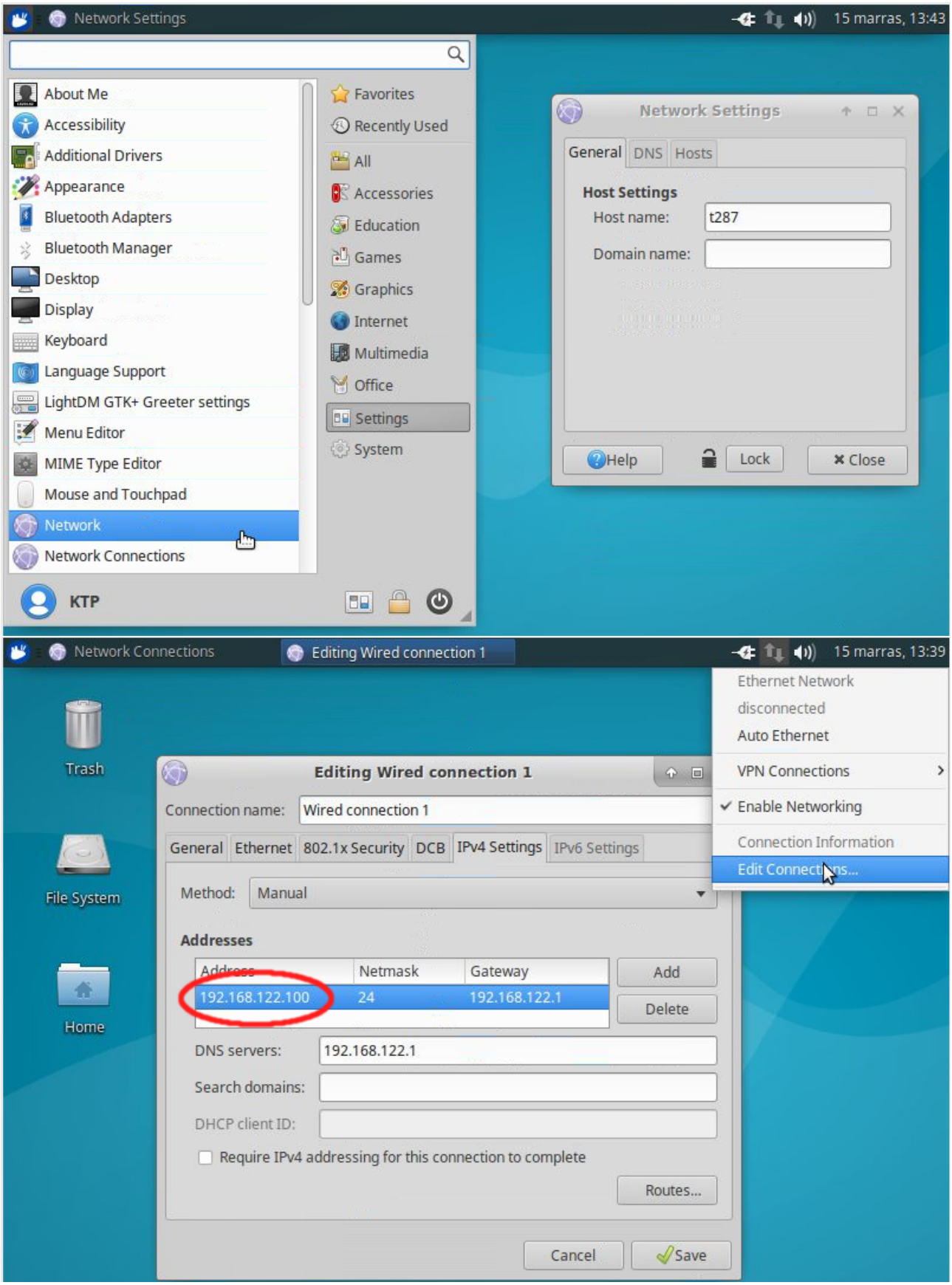
Help: https://github.com/codingforentrepreneurs/Guides

Building a new Research VM

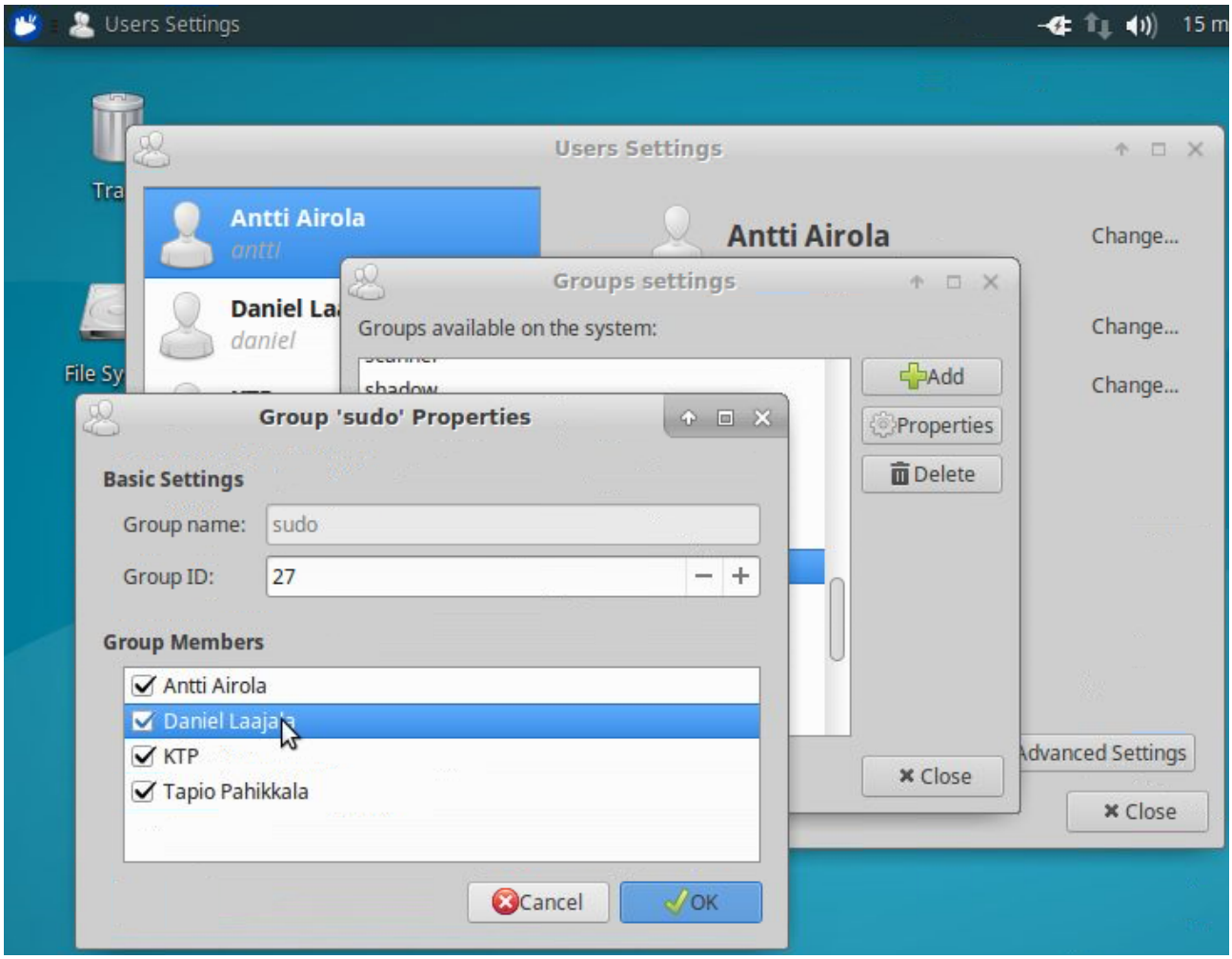
Document Author: arho.virkki@tyks.fi



Clone an approproate virtual machine. Then, log in as with the *ktp* user.



Configure the machine. Configure a static IP and add it to the above list. Remember to refresh the MAC address, if needed (since it gets changed when machine is cloned). Then set machine name to the research project number (e.g. t287).



Manage users. Add user accounts and add sudo (admin) right to the VM users.

Adding public keys to Bastion

For testing purposes, we can also create a key for the ktp user:

```
ssh-keygen -C "Key pair for KTP User" -f KTP_Key
```

Copy the key to the bastion key pool:

```
scp KTP_Key* ktp@analytics.tyks.fi:
ssh ktp@analytics.tyks.fi
sudo sh -c "cat KTP_Key.pub >> /etc/ssh-pool/bastion_keys"
```

Now, connection to the machine can be made with

```
ssh <user>@192.168.122.<machine_number> -o ProxyCommand="ssh bastion@analytics.tyks.fi -W %h:%p -i
    <private_key>"
```

For example,

```
ssh ktp@192.168.122.100 -o ProxyCommand="ssh bastion@analytics.tyks.fi -W %h:%p -i ~/.ssh/KTP_Key"
```

Note: The proxy command can also be stored in *.ssh/config*

```
Host 192.168.122.100
  IdentityFile /home/johndoe/Documents/Keys/My_VSSHP_Key
  ProxyCommand ssh bastion@analytics.tyks.fi -W %h:%p
```

Copying data to External VMs

Open a port connection from in-house server (*ktpanalytics*) to the virtual machine's postgresql and leave it open in one terminal. In the following, change *192.168.122.<machine_number>* into the actual machine ip (like 192.168.122.*126*)

```
ssh ktp@192.168.122.<machine_number> -L 5432:localhost:5432 \
-o ProxyCommand="ssh bastion@analytics.tyks.fi -W %h:%p -i ~/.ssh/KTP_Key"
```

The database *research* and role *analyst* should exist in the VM template. If not, connect the research database and create the database and the user:

```
PGPASSWORD=ktp psql -d postgres -h localhost -U ktp
CREATE ROLE analyst LOGIN CREATEDB CREATEROLE PASSWORD 'analyst';
CREATE DATABASE research WITH owner analyst;
```

Then, copy the data into the virtual machine with the following command at the in-house server (*ktpanalytics*). Change <*schema_name*> into the actual schema containing the data to be exported.

```
pg_dump -U ktp -d ktp -h gradient.vsshp.net -n <schema_name> | \
PGPASSWORD=ktp psql -d research -h localhost -U ktp
```

Log in as 'ktp' to chech that the data is OK, and optionally rename the schema to something more generic to the researcher:

```
PGPASSWORD=ktp psql -d research -h localhost -U ktp

\dn
ALTER SCHEMA <old_schema_name> RENAME TO data;
GRANT USAGE ON SCHEMA data TO analyst;
GRANT SELECT ON ALL TABLES IN SCHEMA data TO analyst;
ALTER ROLE analyst SET search_path TO data,public;
CREATE EXTENSION adminpack;
```

Finally, check that the analyst can log in and view data:

```
PGPASSWORD=analyst psql -d research -h localhost -U analyst

\dt
SELECT count(1) FROM asiakas ;
```

Connecting the VM with X2Go

Adding Extra tools (to be included into the template)

PostgreSQL:

```
sudo apt-get install postgresql-9.5 libpq-dev
sudo -u postgres psql -c "CREATE ROLE ktp SUPERUSER LOGIN PASSWORD 'ktp'";
```

Java:

```
sudo apt-get install default-jdk
```

SQL GUI Tools:

```
# PgAdmin3
sudo apt-get install pgadmin3

# SquirrelSQL
wget
    http://sourceforge.net/projects/squirrel-sql/files/1-stable/3.7.1/squirrel-sql-3.7.1-standard.jar
sudo java -jar squirrel-sql-3.7.1-standard.jar
sudo ln -s /usr/local/squirrel-sql-3.7.1/squirrel-sql.sh /usr/local/bin/
wget https://jdbc.postgresql.org/download/postgresql-9.4.1212.jar
mv postgresql-9.4.1212.jar /opt/ktp/jar/
```

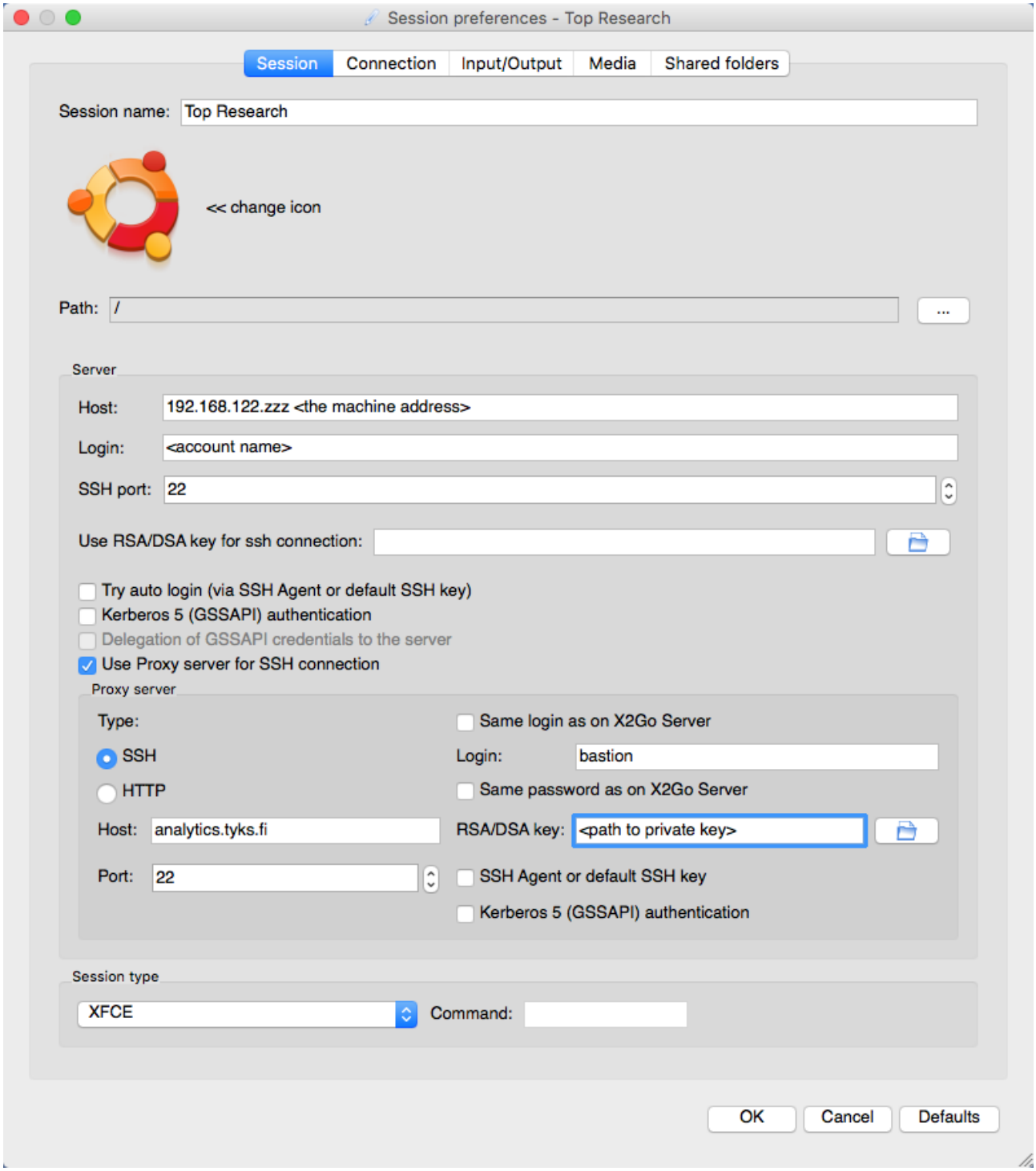

Session preferences - Top Research
Session
Connection
Input/Output
Media
Shared folders
Session name: Top Research
<< change icon
Path: /
...
Server
Host: 192.168.122.zzz <the machine address>
Login: <account name>
SSH port: 22
Use RSA/DSA key for ssh connection:
Try auto login (via SSH Agent or default SSH key)
Kerberos 5 (GSSAPI) authentication
Delegation of GSSAPI credentials to the server
Use Proxy server for SSH connection
Proxy server
Type:
SSH
HTTP
Host: analytics.tyks.fi
Port: 22
Same login as on X2Go Server
Login: bastion
Same password as on X2Go Server
RSA/DSA key: <path to private key>
SSH Agent or default SSH key
Kerberos 5 (GSSAPI) authentication
Session type
XFCE
Command:
OK
Cancel
Defaults

Figure 1:

Editors:

```
sudo apt-get install vim-gtk3
```

R + RStudio:

```
sudo apt-get install r-core

sudo apt-get install gdebi-core
wget https://download2.rstudio.org/rstudio-server-1.0.44-amd64.deb
sudo gdebi rstudio-server-1.0.44-amd64.deb
```

Connecting Data Analysis Platform (DAP)

Document Author: arho.virkki@tyks.fi

Obtaining Access to DAP

Turku Centre for Clinical Informatics at the Hospital District of Southwest Finland offers Data Analytics Platform (DAP) for data-driven research projects.

To use the service, you need to

1. Have a valid research permission (T-number)
2. Have agreed DAP terms and conditions
3. Have obtained credentials for the service

This guide explains how to get credentials to DAP, and the optional step of installing an open-sourced X2Go software package for Linux, Apple or Windows for native remote desktop access.

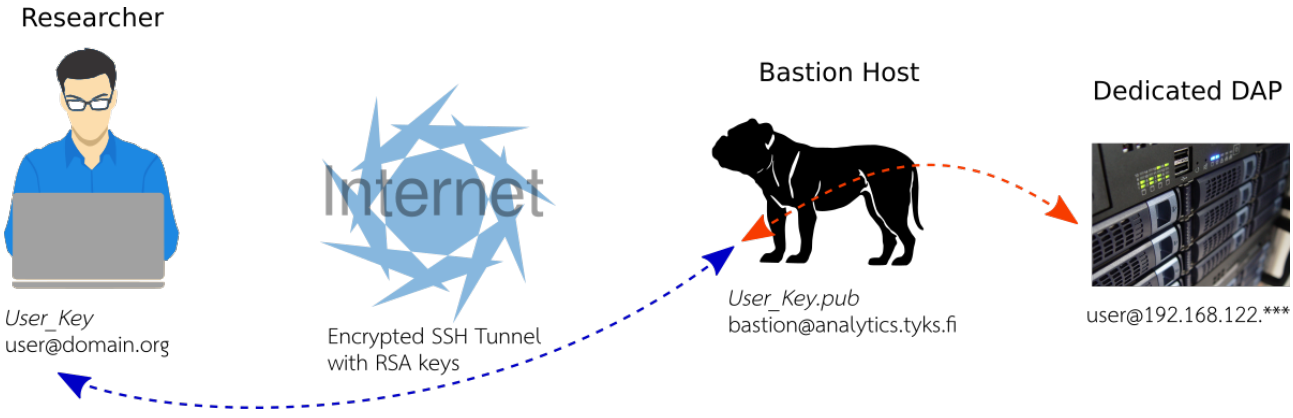


Figure 1:

Architecture. A Schematic illustarion of the Data Analytics Platform (DAP). API end points and web services like "R Studio Server" can be tunneled to researcher's computer through the Secure Shell (SSH) connection, which is a sptandard Internet protocol for encrypted communication.

1. Obtain Key Pair for Asymmetric Cryptography

Visit the CCI office, or generate the files yourself with the instructions below.

On Linux and Apple, the needed tools come bundled. On Windows 10, you need either to enable the Windows Sybsystem for Linux [1], or get a separate tool that installs OpenSSH. The easiest way for Windows users is to install Git [2], which comes with a bundled bash shell and ssh.

When ready, open Terminal.app, Gnome Terminal or Bash Launcher, and generate a key pair with the command

```
ssh-keygen -C "Your Name" -f Your_Name
```

The command generates two files, for example

```
John_Doe
John_Doe.pub
```

Email the public key (with the .pub extension) to ktp@tyks.fi (cc: arho.virkki@tyks.fi). Keep the private key (with no extension) in a safe place. **Do not email the private key to anyone!**

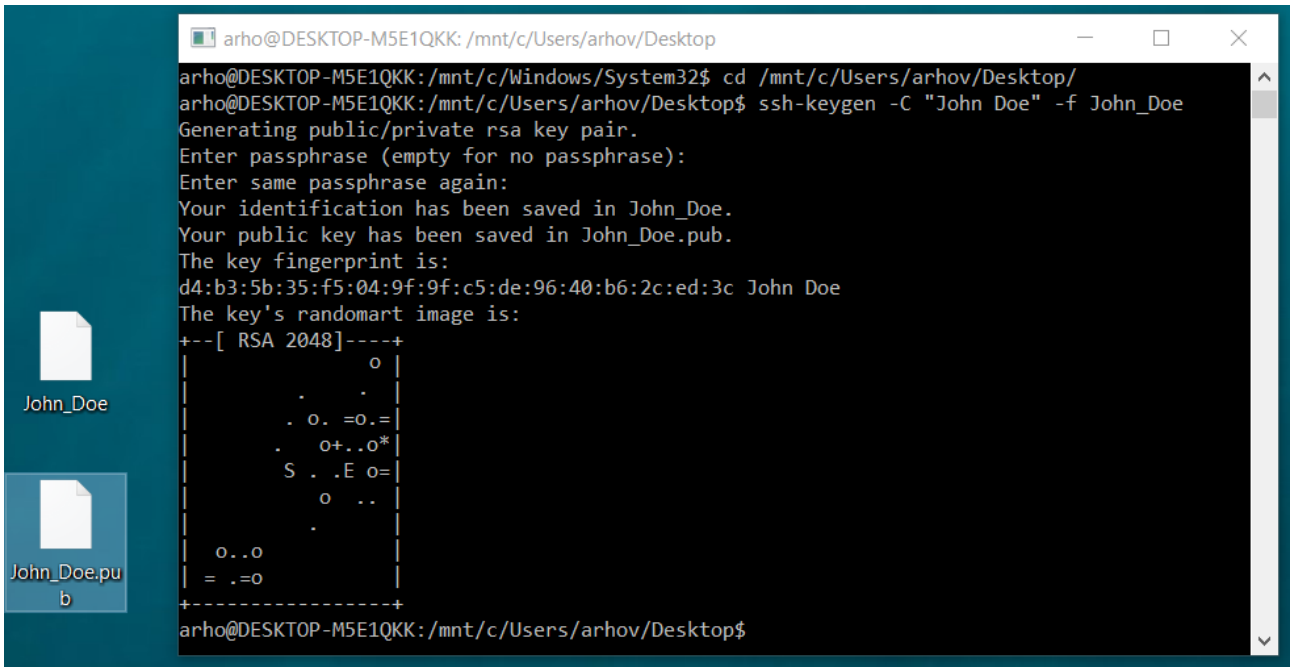


Figure 2:

Example. Generating keys public and private keys.

[1] http://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/
[2] https://git-scm.com/

2. Connect DAP via Command Line

Connection works only after CCI has obtained your public key and added it to the system.

Again, open Terminal.app, Gnome Terminal or Bash Launcher, and issue the command

```
ssh <user>@192.168.122.<your machine> -o \
ProxyCommand="ssh bastion@analytics.tyks.fi -W %h:%p -i <private_key>"
```

where “" denotes the continuation of line. For example,

```
ssh <your user>@192.168.122.<machine_num> -o ProxyCommand="ssh bastion@analytics.tyks.fi -W %h:%p
    -i ~/c/files/KTP_Key"
```

You should be able to log in with the correct password.

Using services on the DAP is now easy. For example, you can tunnel R Studio Server through the connection by adding its port as an option in the above command,

```
-L 8787:localhost:8787
```

Now, R Studio Server can be reached by pointing a browser to address http://localhost:8787, as shown in the image below.

Example. Using R Studio Server on DAP.

3. Connect DAP Using X2Go Remote Desktop

Download and install a copy of X2Go from http://wiki.x2go.org. This will provide you a full-blown graphical desktop to the remote server.

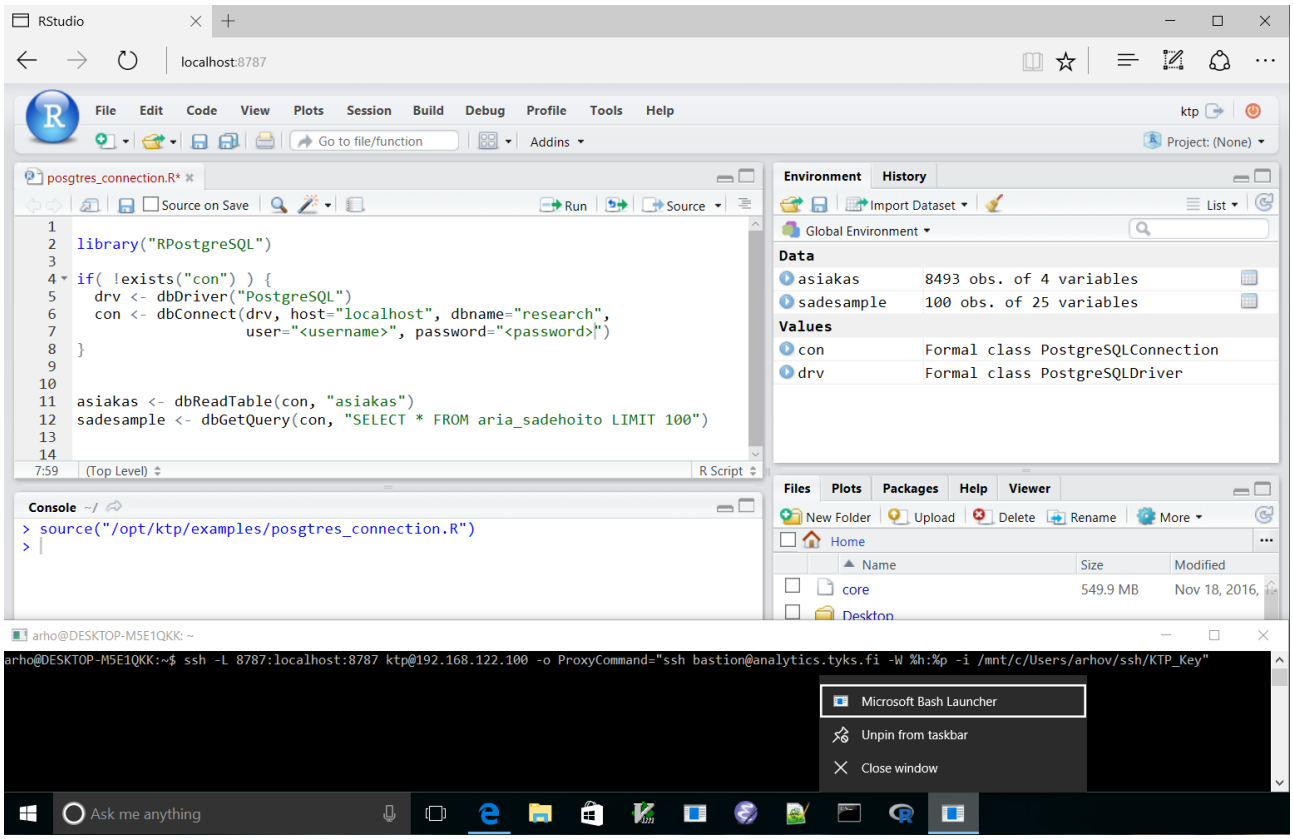


Figure 3:

The settings must be chosen exactly as in the image below.

Note! Some recent **OS X** versions do have trouble in connecting through the bastion SSH proxy. Stable and tested versions are available at: http://cci.vsshp.fi/public/x2go/apple.

- Host: the IP of your machine, e.g. 192.168.122.100
- Login: You account name, e.g. johndoe
- [x] Use Proxy server for SSH connection
- Type: SSH
- Login: bastion
- Host: analytics.tyks.fi
- RSA/DSA key: Full path to your *private* key
- Session type: XFCE

Example. Connection parameters for X2Go.

Known limitations

- On some versions of Windows, X2Go does not suppors switching between full screen and windowed mode during a single session, or does not support full screen mode at all.
- If the path to the RSA/DSA private key is wrong, the software will close with no error message.

Accessing the Data

In most cases, the data is stored in PosgreSQL database engine in a database named *research*. A ready-made ordinary user *analyst* (with the same default password) can access this database from the virtual machine's terminal with the command

```
PGPASSWORD=analyst psql -U analyst -d research -h localhost
```

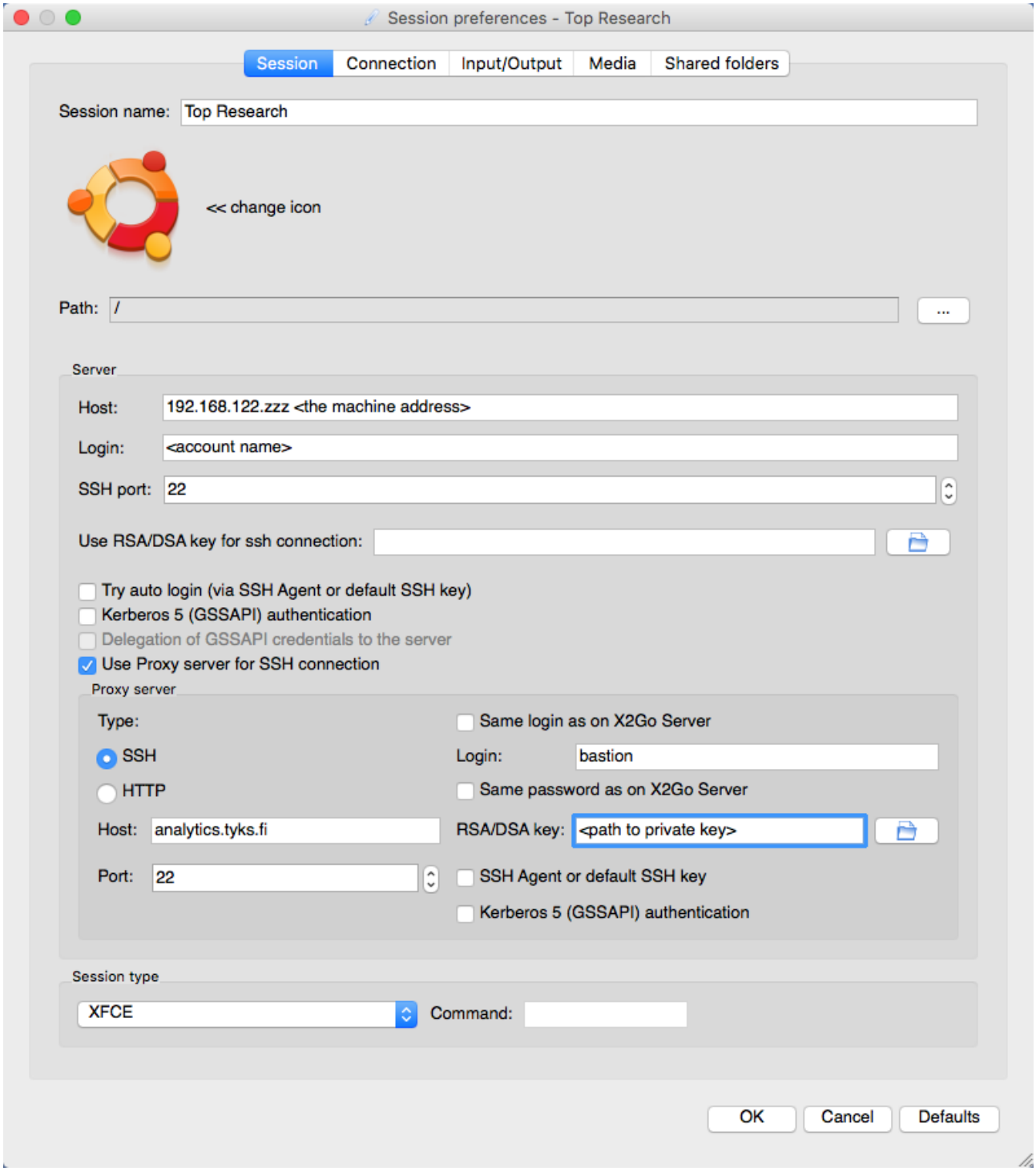

Session preferences - Top Research
Session
Connection
Input/Output
Media
Shared folders
Session name: Top Research
<< change icon
Path: /
...
Server
Host: 192.168.122.zzz <the machine address>
Login: <account name>
SSH port: 22
Use RSA/DSA key for ssh connection:
Try auto login (via SSH Agent or default SSH key)
Kerberos 5 (GSSAPI) authentication
Delegation of GSSAPI credentials to the server
Use Proxy server for SSH connection
Proxy server
Type:
SSH
HTTP
Host: analytics.tyks.fi
Port: 22
Same login as on X2Go Server
Login: bastion
Same password as on X2Go Server
RSA/DSA key: <path to private key>
SSH Agent or default SSH key
Kerberos 5 (GSSAPI) authentication
Session type
XFCE
Command:
OK
Cancel
Defaults

Figure 4:

The data resides in the *data* schema. The tables in this schema can be listed as follows:

```
research=> \dt data.
                   List of relations
 Schema |              Name              | Type  | Owner
--------+--------------------------------+-------+-------
 data   | asiakas                        | table | ktp
 data   | diagnoosi                      | table | ktp
 data   | laake_maarays                  | table | ktp
 data   | labrat                         | table | ktp
 data   | leikkaus_opera                 | table | ktp
 data   | leikkaus_toti                  | table | ktp
 data   | oberon_toimenpiteet            | table | ktp
 data   | palvelu                        | table | ktp
 data   | radut                          | table | ktp
 data   | reseptit                       | table | ktp
 data   | yhdistetty                     | table | ktp
(11 rows)
```

Uploading Own Data

Option 1.

Use *X2Go Shared folders* feature to copy data back and forth between the computing environment.

Option 2.

Use command line *scp* (With Linux, OS X, and Microsoft's Windows 10 Subsystem for Linux)

Examples:

```
# Open a tunnel to the remote machine at local port 2222.
ssh -N -L 2222:192.168.122.100:22 bastion@analytics.tyks.fi -i <private_key_file> &

# Copy the file(s) to DAP Desktop folder
scp -P 2222 <my_files> <username>@localhost:~/Desktop/
```

Option 3.

Use dedicated secure copy software like WinSCP on Windows which can tunnel connections through the bastion host.

Installing WinCSP (for Windows)

Download a recent copy of WinSCP from https://winscp.net/eng/download.php and run the installation package. (In case you do not have administrator rights, choose portable executables. They are just launched directly and not installed.)

Configure site. Set host name to your DAP IP (192.168.122.<number>) and user name and password to your personal credentials. Then click the "Advanced…" button.

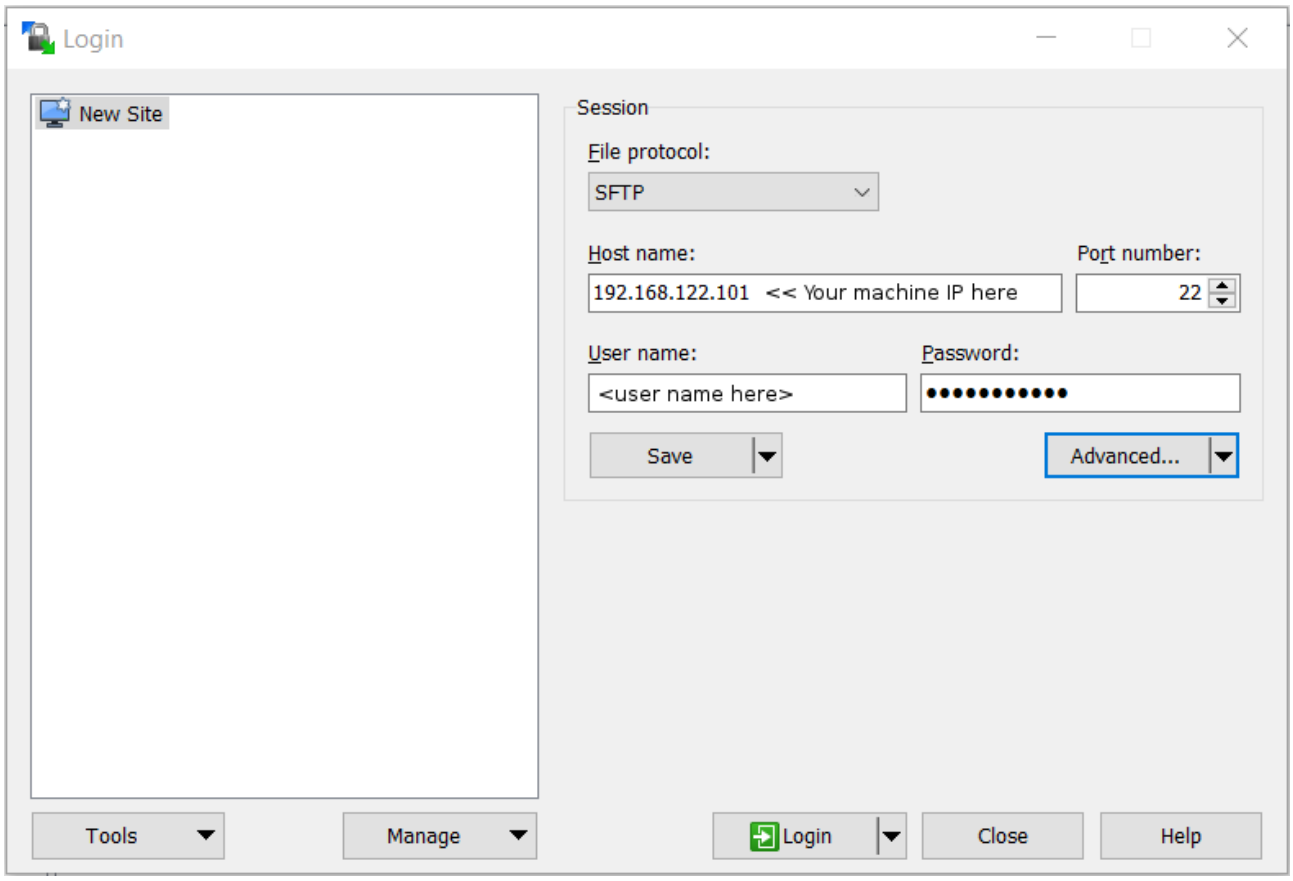
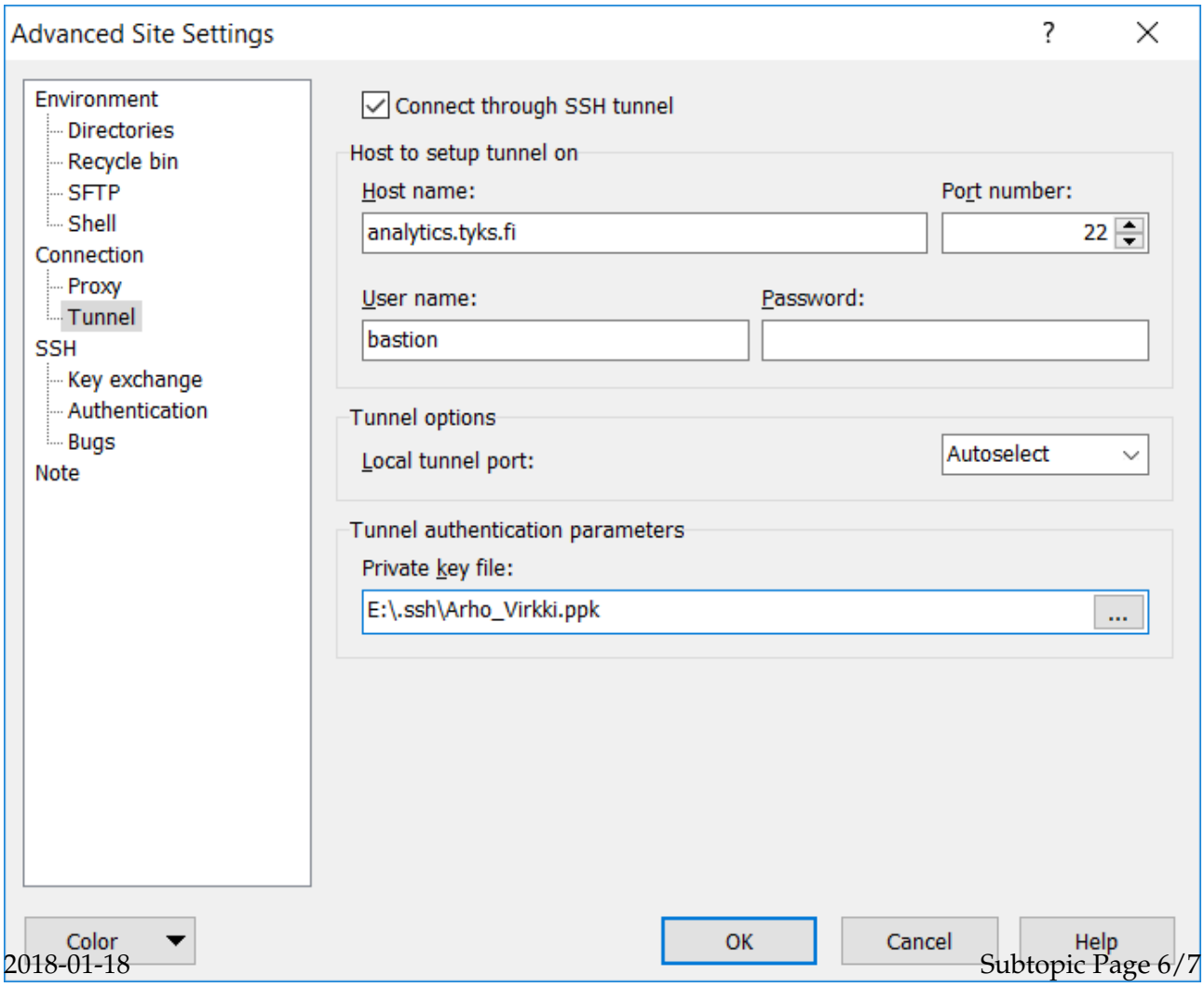

Login
New Site
Session
File protocol:
SFTP
Host name:
Port number:
192.168.122.101 << Your machine IP here
22
User name:
Password:
<user name here>
Save
Advanced...
Tools
Manage
Login
Close
Help

Figure 5:


Advanced Site Settings
Environment
Directories
Recycle bin
SFTP
Shell
Connection
Proxy
Tunnel
SSH
Key exchange
Authentication
Bugs
Note
Connect through SSH tunnel
Host to setup tunnel on
Host name:
Port number:
analytics.tyks.fi
22
User name:
Password:
bastion
Tunnel options
Local tunnel port:
Autoselect
Tunnel authentication parameters
Private key file:
E:\.ssh\Arho_Virkki.ppk
...
Color
OK
Cancel
Help

Set up tunnel. Select Connection -> Tunnel,and [x] Connect through SSH tunnel. Host name must be **analytics.tyks.fi** and user **bastion**. Finally choose your private key file. When given an OpenSSH privater key, WinSCP offers to convert it into (Windows-specific) Putty format.

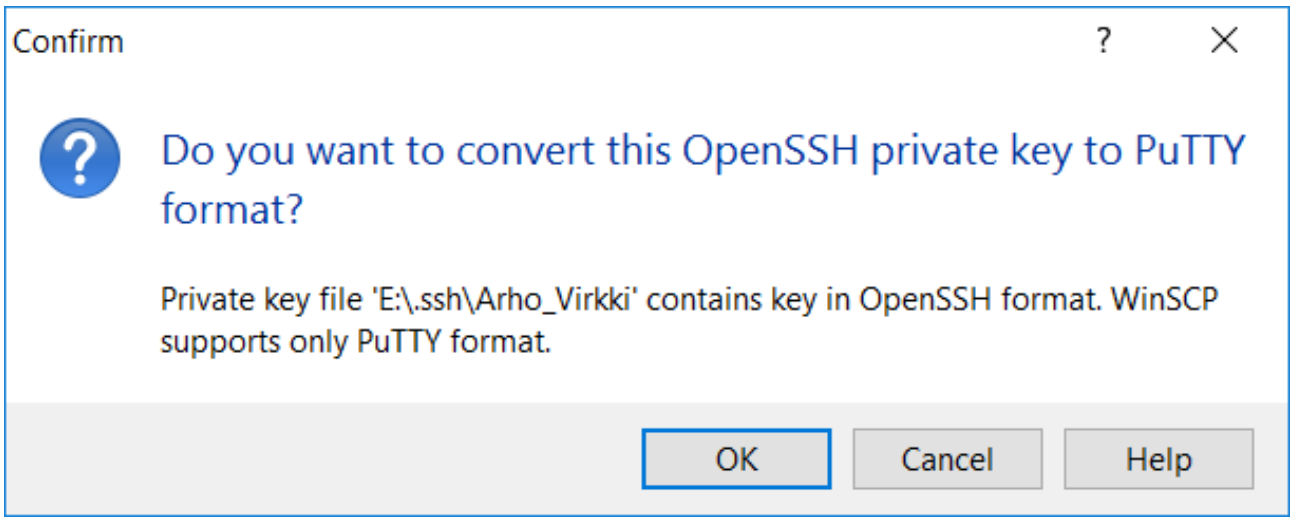


Figure 6:

Key conversion. OpenSSH private key can be converted into Putty format and saved.

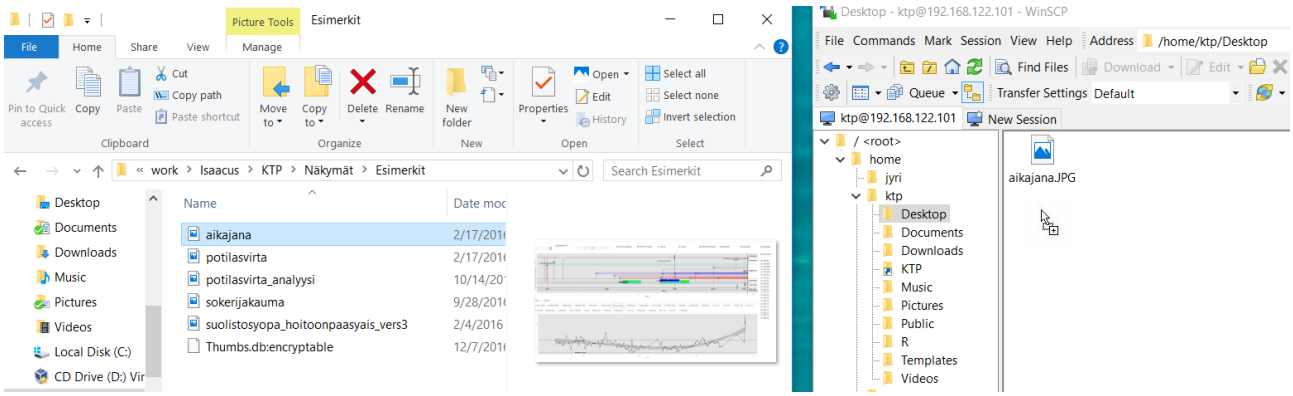


Figure 7:

File copying. Files can be copied with drag and drop.

Limited SFTP Access to analytics.tyks.fi

Document Author: arho.virkki@tyks.fi

Adding a new SFTP user

The *NEWUSER* and *ID_PUB* environment variables should match the new user account name and the corresponding OpenSSH public key file.

```
# Create user and add it to the 'sftp-only' group
NEWUSER=test1

sudo mkdir /var/sftp/$NEWUSER
sudo useradd $NEWUSER -d /$NEWUSER -M
sudo usermod $NEWUSER -aG sftp-only

# Copy a public key to .ssh/authorized_keys and set its permissions
ID_PUB=Key_Holder_Name.pub

sudo mkdir -p /var/sftp/$NEWUSER/.ssh
sudo cp $ID_PUB /var/sftp/$NEWUSER/.ssh/authorized_keys
sudo chmod og-rwx /var/sftp/$NEWUSER/.ssh
sudo chown $NEWUSER: -R /var/sftp/$NEWUSER
```

Prerequisites for the Setup

Adapted from: https://access.redhat.com/solutions/2399571

Add group for SFTP Users

```
sudo groupadd sftp-only
```

Create a home folder for the sftp users

```
sudo mkdir /var/sftp
```

Now modify */etc/ssh/sshd_config*. Add *sftp-only* to allowed groups

```
AllowGroups <leave previous entries here and add => sftp-only
```

and add a Chroot rule

```
Match Group sftp-only
          ChrootDirectory /var/sftp/
          X11Forwarding no
          AllowTcpForwarding no
          ForceCommand internal-sftp -l VERBOSE
          AuthorizedKeysFile /var/sftp/%u/.ssh/authorized_keys
```

Finally, restart the ssh server

```
sudo systemctl restart sshd
```

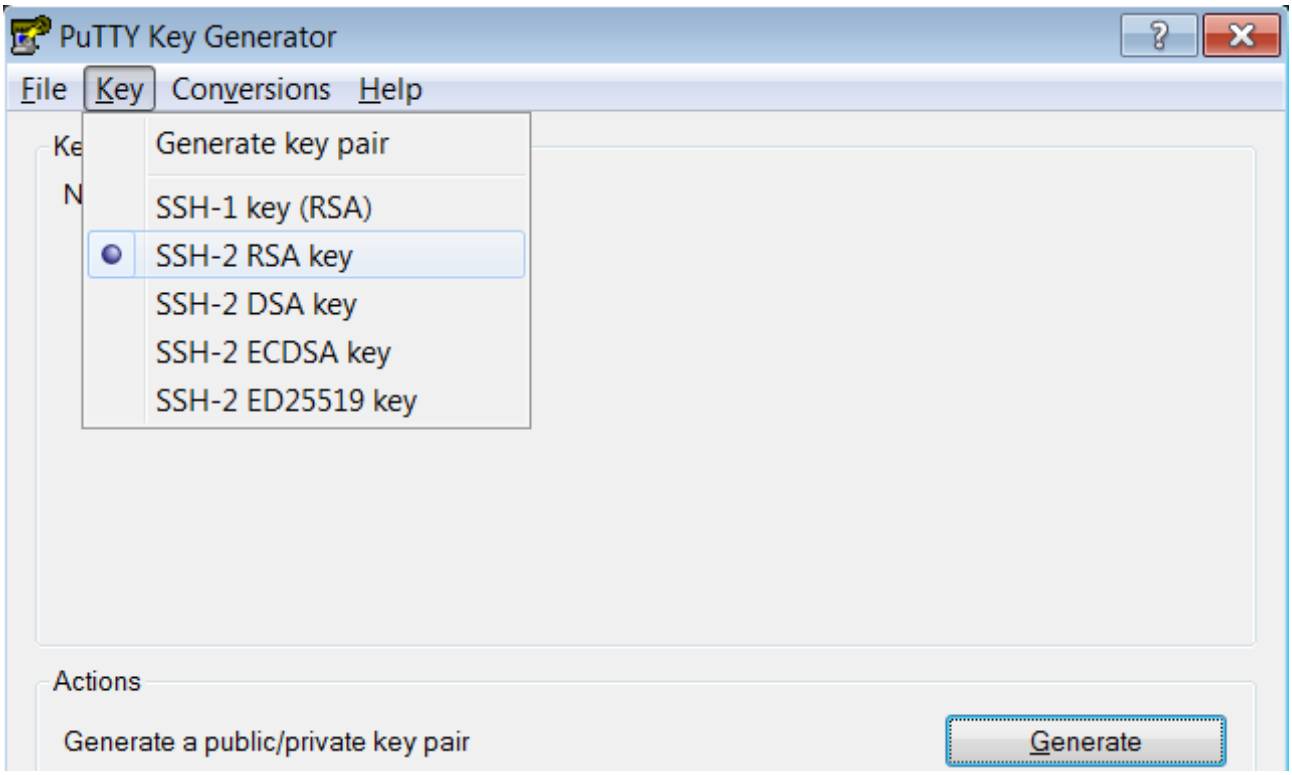
Appendix: Useful Putty Commands

Convert Putty public key into OpenSSH format (ignoring the comment tag, unfortunately):

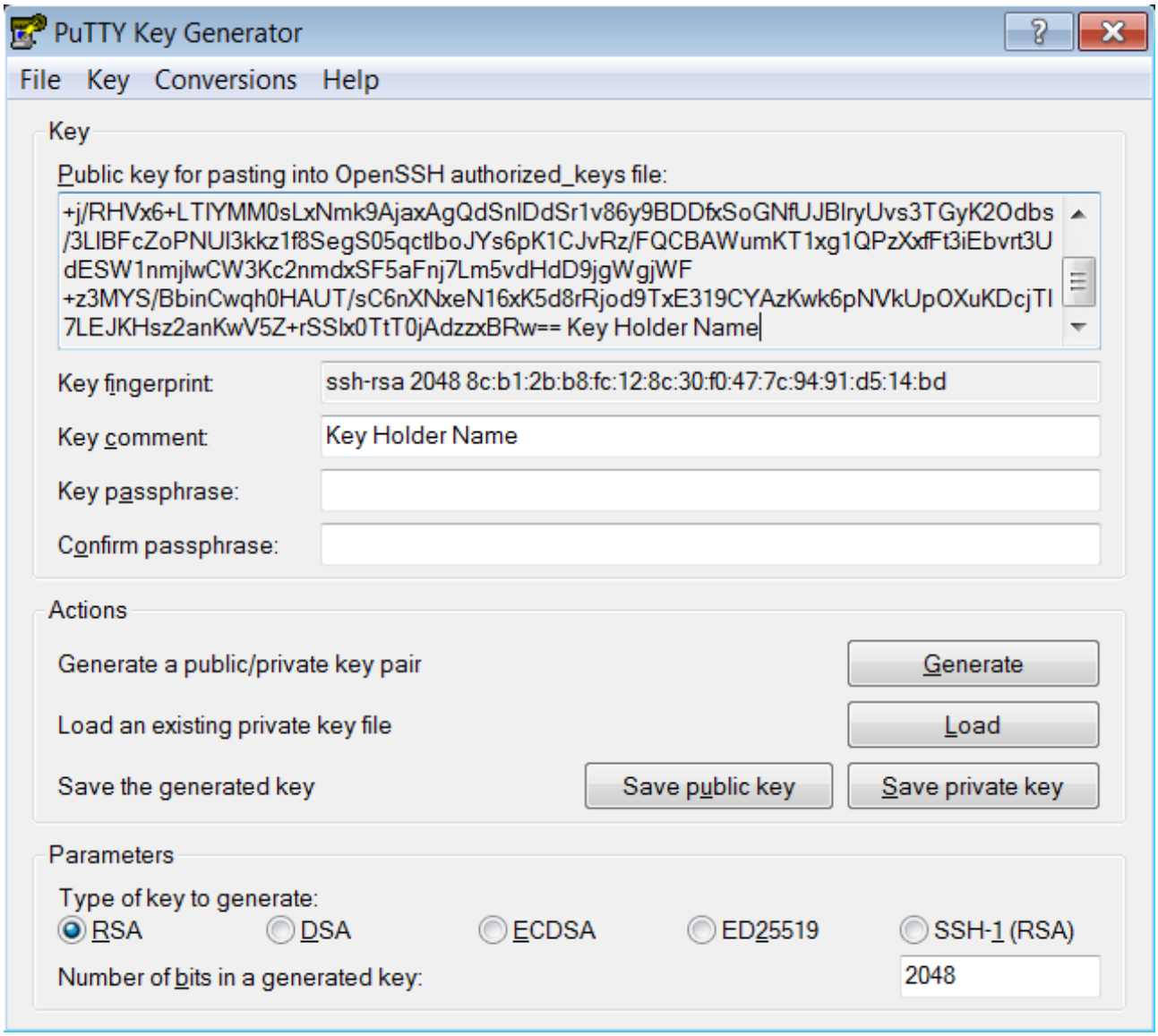
```
ssh-keygen -i -f Key_Holder_Name.puttypub > Key_Holder_Name.pub
```

Export OpenSSH public and private keys from Putty ppk file

```
puttygen Key_Holder_Name.ppk -O public-openssh
puttygen Key_Holder_Name.ppk -O private-openssh -o Key_Holder_Name
```



Generating keys in Putty.



Exporting Putty-generated OpenSSH keys.

Useful virsh commands

Document Author: arho.virkki@tyks.fi

```
# List running and all machines
sudo virsh list
sudo virsh list --all
```

```
# How much memory is allocated
sudo virsh domstats --balloon

# Only running machines
sudo virsh domstats --balloon | grep -e current -e Domain

# Only numbers (in KiB)
sudo virsh domstats --balloon | grep -e current | cut -d "=" -f 2

# Sum of all memory consumed (in KiB)
sudo virsh domstats --balloon | grep -e current | cut -d "=" -f 2 | paste -s -d+ | bc

# The same in (GiB)
sudo virsh domstats --balloon | grep -e current | cut -d "=" -f 2 |\
paste -s -d+ | bc | sed 's/$/\/1024\/1024/' | bc
```

Summary of Open Source Software

Operating Systems

- Physical servers: RHEL / CentOS 7 https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/
- Hypervisors: KVM https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine
- Virtual Machines: Ubuntu Linux https://www.ubuntu.com/desktop

Databases, data transformation and analysis

- Pentaho Kettle ETL-tool http://www.pentaho.com/product/data-integration
- PostgreSQL https://www.postgresql.org/docs/9.6/static/index.html
- Relational data modelling http://software.sqlpower.ca/page/architect
- Cloudera Hadoop https://www.cloudera.com/documentation.html
- R language (for statistical computing) https://www.r-project.org/
- Python 3 (for scientific computing and scripting) https://www.python.org/
- Unix Shell (for process automation) https://www.gnu.org/software/bash/manual/
- Machine learning: Weka https://www.cs.waikato.ac.nz/ml/weka/

Version control, documentation and backups

- Version control: Git https://git-scm.com/
- Wiki pages & notes: Markdown https://en.wikipedia.org/wiki/Markdown
- Kanban board: Wekan https://wekan.github.io/
- Backups: Cron & rsync https://en.wikipedia.org/wiki/Cron, https://en.wikipedia.org/wiki/Rsync

Publishing and Office

- Vector graphics: Inkscape https://inkscape.org/en/
- Image manipulation: Gimp https://www.gimp.org/
- Scientific publishing: LaTeX https://en.wikipedia.org/wiki/LaTeX

Remote Access

- OpenSSH https://en.wikipedia.org/wiki/OpenSSH
- X2Go https://wiki.x2go.org/doku.php
- Apache Guacamole (to be installed...) https://guacamole.apache.org/

Web-based User Interfaces

- Front-end layout: Twitter Bootstrap https://getbootstrap.com/
- Front-end widgets: jQuery and jQuery UI https://jquery.com/
- Interactive graphics: D3.js https://d3js.org/
- Back-end logic: Python/Flask http://flask.pocoo.org/