kindred

Automating Oracle database administration, practical examples with Ansible

Mikael Sandström / Ilmar Kerm

@oravirt

@ilmarkerm

KINDRED GROUP

KINDRED IS ONE OF THE LARGEST ONLINE GAMBLING COMPANIES IN THE WORLD
WITH OVER 20 MILLION REGISTERED CUSTOMERS

LICENSED OPERATOR IN

Australia, Belgium, Denmark, Estonia,
France, Germany, Gibraltar, Ireland, Italy,
Malta, Romania, United Kingdom

OUR BRANDS

32Red, BINGO.com, BOHEMIA CASINO, CASINOHUONE, kolikkopelit, iGame, MARIACASINO, RoxyPalace, StanJames.com, STORSPILLER STORSPELARE, UNIBET

2016 AWARDS

eGR OPERATOR OF THE YEAR 2016,
eGR SOCIALLY RESPONSIBLE OPERATOR OF THE YEAR 2016,
eGR MARKETING CAMPAIGN OF THE YEAR 2016

More than
1,100,000
active customers

11,841
streamed
events

73%
of revenue
from mobile

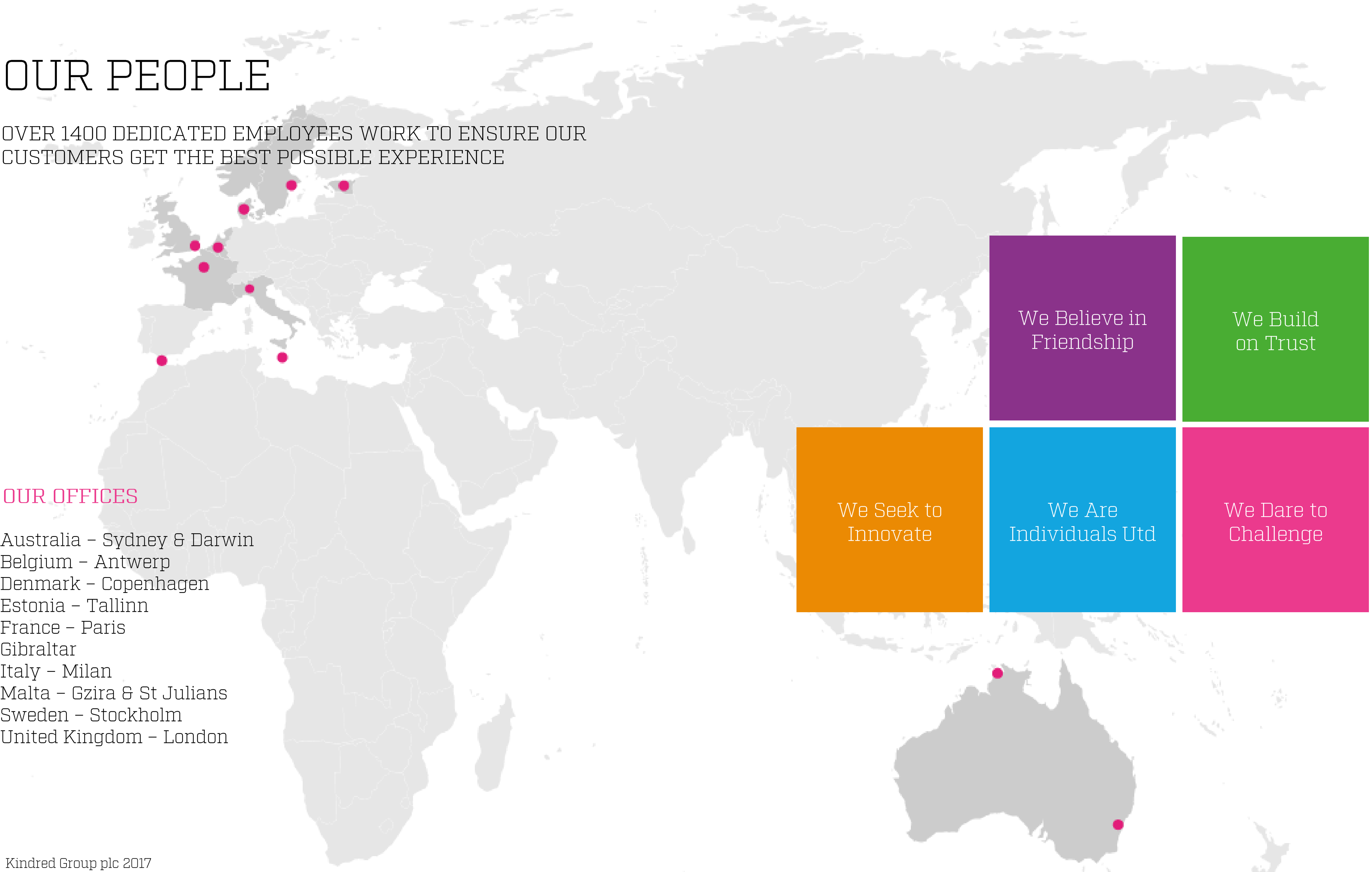
16-18 Million
transactions
everyday

45,861
live betting
events

Numbers for Q2 2017

OUR PEOPLE

OVER 1400 DEDICATED EMPLOYEES WORK TO ENSURE OUR CUSTOMERS GET THE BEST POSSIBLE EXPERIENCE



OUR OFFICES

Australia – Sydney & Darwin
Belgium – Antwerp
Denmark – Copenhagen
Estonia – Tallinn
France – Paris
Gibraltar
Italy – Milan
Malta – Gzira & St Julians
Sweden – Stockholm
United Kingdom – London

We Believe in Friendship

We Build on Trust

We Seek to Innovate

We Are Individuals Utd

We Dare to Challenge

© Kindred Group plc 2017

WHOAMI: Ilmar Kerm

- Senior database administrator in Kindred Group
- Working in IT since 2000
- Working with Oracle database since 2003
- Working in Kindred since 2007/2015
- From LAMP developer to Oracle DBA
- President of Oracle User Group Estonia
- Oracle ACE Associate

Blog: https://ilmarkerm.eu
Email: ilmar.kerm@kindredgroup.com
Twitter: @ilmarkerm

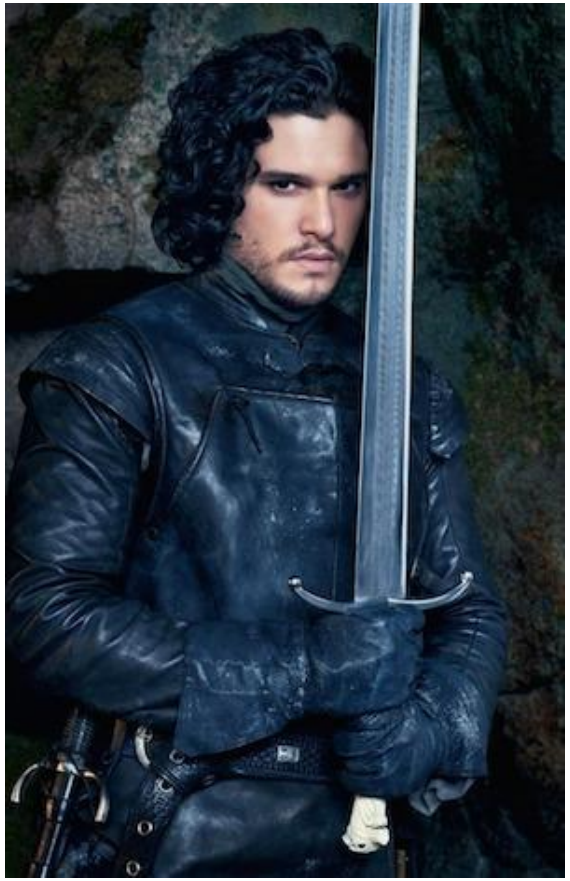
(… introducing… Mikael)

- The Northern King

Blog: https://oravirt.wordpress.com

Email: mikael.sandstrom@kindredgroup.com

Twitter: @oravirt



WHOAMI: Mikael Sandström

- Senior database administrator at Kindred Group
- Worked with Oracle for a long time
- Working at Kindred since early 2016
- First time presenting at a conference
- Automation junkie

Blog: https://oravirt.wordpress.com

Email: mikael.sandstrom@kindredgroup.com

Twitter: @oravirt

Agenda

- We expect that you are familiar with Ansible
- We only show you demos about practical/routine Oracle DBA tasks
- Things we cover
  - Setting up Ansible environment for Oracle DB use: inventory, modules, cxOracle
  - Creating/managing users and roles, syncing with Active Directory
  - Services, tablespaces, resource manager, DBMS_SCHEDULER, AWR settings, INIT.ORA parameters, ASM diskgroups
  - User privileges, including grants to entire target schema
  - Gathering facts about databases
  - Managing passwords in playbooks
  - Practical playbook deployment: execution, scheduling, configuration
  - Show GoldenGate assisted near zero downtime DB upgrade playbook
- Things we do not cover, because every other Ansible presentation does these
  - What is Ansible (check the first bullet point)
  - Managing OS
  - Installing Oracle RDBMS/Grid software
  - Creating databases/clusters (we will create PDBs though)

Setting up

- Clone the modules and put them in a directory called 'library'

  ```
  git clone https://github.com/oravirt/ansible-oracle-modules library
  ```

- Install cx_Oracle
  - Needed where the modules are executed
  - cx_Oracle needs an Oracle client
  - If run locally on 'control-machine' instant client is fine, otherwise just install on the database host
  - pip or yum

Inventory and databases [Potential problem]

- Inventory is based on groups of hosts (doesn't map exactly to databases)
- So you do this, one config file per db

```
group_vars/db1

- Inventory
[db1]
host1.example.com

ansible-playbook dbconfig.yml -e db=db1
```

- Then you add another db config (db2), on the same host

```
group_vars/db1
group_vars/db2

- Inventory
[db1]
host1.example.com
[db2]
host1.example.com

ansible-playbook dbconfig.yml -e db=db2
```

- You might get unexpected results (variables might not come from the configuration you expected)

Inventory and databases [Solutions]

- Separate inventory files, one config per inventory

  ```
  $ cat inventory/db1
  [db1]
  host1.example.com
  $ cat inventory/db2
  [db2]
  host1.example.com
  ```

  and target the config with '`ansible-playbook … -i inventory/db1`' etc

- Use fake hosts

  ```
  [db1]
  host-db1    ansible_host=host1.example.com
  [db2]
  host-db2    ansible_host=host1.example.com
  ```

- Use cnames
  - Same concept as fake hosts

Inventory (how we do it, for the most part)

- Inventory contains normal hostgroups
- database config in the same config as host & cluster config (in the same group_vars)
- list of cdbs/pdbs in group_vars
  - This works well since we always create the cdb/pdb

Inventory (random thoughts)

- Try to keep the inventories as standard as possible (for as long as possible)
- Maybe keep inventory in it's own repository (to avoid duplication, if many different repos for management).
  - Possible to softlink, or just point playbook to inventory
- If environment is large, maybe need external service registry (and dynamic inventory script, like for AWS)

Managing passwords

- You should not store plain text passwords in playbooks!
- Ansible-vault
  - Ansible native solution to encrypt YML files that can be included in playbooks
  - Ansible-vault files can be committed to git
- Oracle wallet
  - Ansible-oracle-modules use cxOracle python module that requires OCI client, therefore it also supports Oracle wallet authentication
  - Requires setting up tnsnames.ora and sqlnet.ora also
  - Each TNS name needs password entry in wallet
    - Limitation for large deployments?
- Calling remote password service from playbook
  - We use REST API calls to Passwordstate to fetch individual DB passwords
  - API key can be secured with ansible-vault

Scheduling/running playbooks

- We use Jenkins
- Basically 3 ways of running jobs
  - Manually
  - Trigger based
  - Scheduled

Running playbooks (a few other options)

- Ansible Tower (commercial offering)
- Ansible AWX (open source Tower)
- Rundeck
- Semaphore
- *Tensor*
- *Polemarch*

Ansible is great, but…

- … it is not for everything
- There are better solutions for:
  - Managing schema objects
  - Deploying database code
- Better look at specific tools for these purposes:
  - Liquibase
  - FlyWay
- But… Ansible can execute these tools as part of the release process orchestration

Resources

- Ansible homepage: https://www.ansible.com/
- Oracle database modules for Ansible:

  https://github.com/oravirt/ansible-oracle-modules
- Deploying RAC with Ansible:

  https://github.com/oravirt/ansible-oracl

Download scripts

- Here you can download the scripts used in this presentation
  – https://github.com/oravirt/hroug-2017

Thank You