

SANGAM21 SESSION

Automatically deploy schema changes with Ansible and Liquibase

SESSION HIGHLIGHTS

- Using Liquibase for agile database schema changes
- Automate deployments and other tasks with Ansible
- Combine Liquibase and Ansible


08 Dec 2021

04:30 TO 05:30 PM
IST (GMT+5:30)

Global Webinar Series (APAC, EMEA, AMS)
13 Days, 100+Hrs, 100+Sessions, 100+Speakers

Place: Zoom Webinars
Zero Cost to Attend



Robert Marz
Technical Architect
Databee - The IT-Architects
 @RobbieDatabee



Robert Marz – Independent Consultant

Primary Role

- Senior Technical Architect
- with database centric view of the world

DOAG (German Oracle User Group)

- Active Member of Database Community
- Responsible for Cloud Topics


@RobbieDatabee


robbie.databee.org


robert.marz
@databee.org





500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical contributions in the Oracle community

3 membership tiers



For more details on Oracle ACE Program:
bit.ly/OracleACEProgram



Nominate

yourself or someone you know:

acenomination.oracle.com

Connect:  oracle-ace_ww@oracle.com

 Facebook.com/oracleaces

 [@oracleace](https://twitter.com/oracleace)

Agile & DevOps vs Databases



AGILE

KANBAN

SCRUM

LEAN

DESIGN

MANUFACTURING

IMPROVEMENT

TECHNOLOGY

MANAGEMENT

OPERATIONS

DEVELOPMENT

TESTING

DEPLOYMENT

MONITORING

SECURITY

COMPLIANCE

REGULATION

INTEGRATION

COLLABORATION

COMMUNICATION

TRANSFORMATION

INNOVATION

EFFICIENCY

QUALITY

RESILIENCE

AGILITY

ADAPTABILITY

SCALABILITY

SECURITY

COMPLIANCE

REGULATION

INTEGRATION

COLLABORATION

COMMUNICATION

TRANSFORMATION

INNOVATION

EFFICIENCY

QUALITY

RESILIENCE

AGILITY

ADAPTABILITY

SCALABILITY

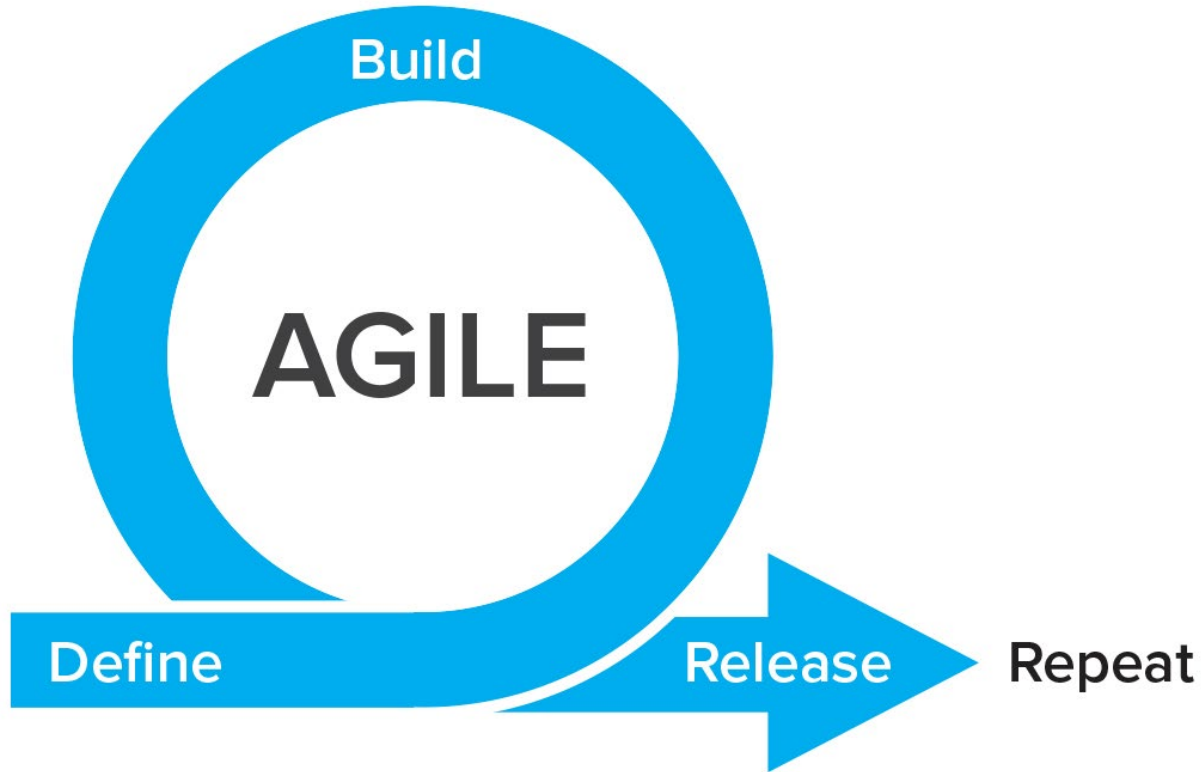


Agile and DevOps – Database Challenges





Agile meets Databases



Agile

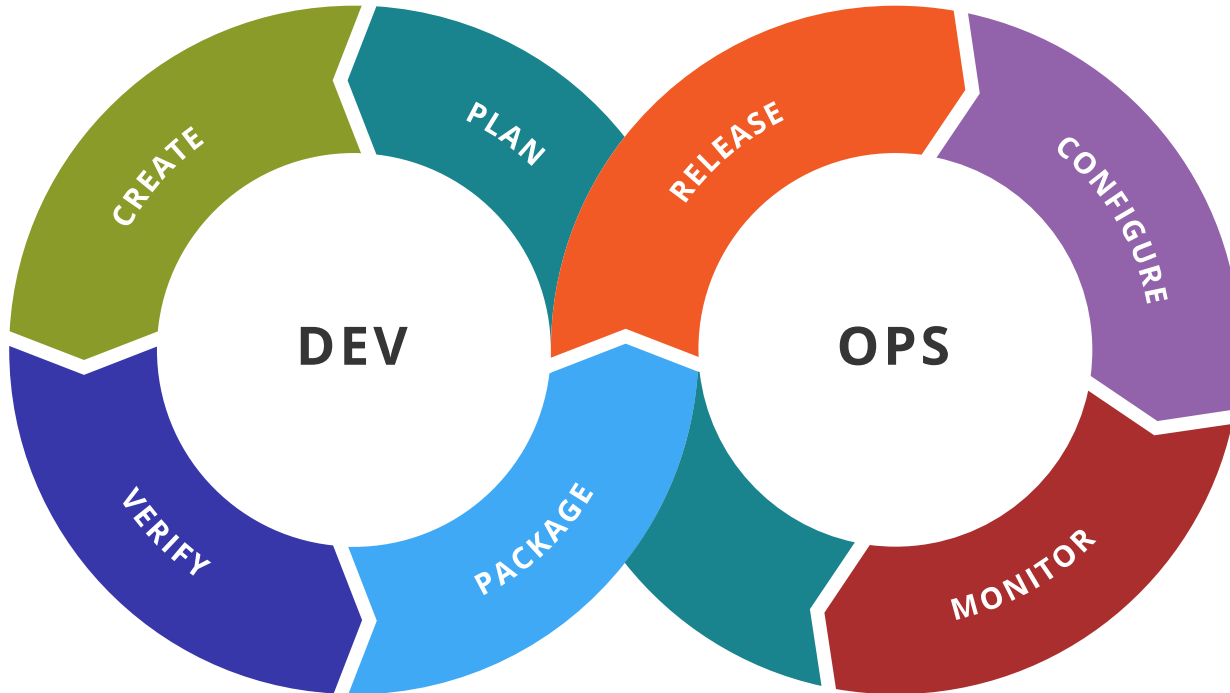
Short
Development
cycles

Many Iterations

Frequent
Schema
Changes



DevOps Requirements for Databases



Continuous
Integration

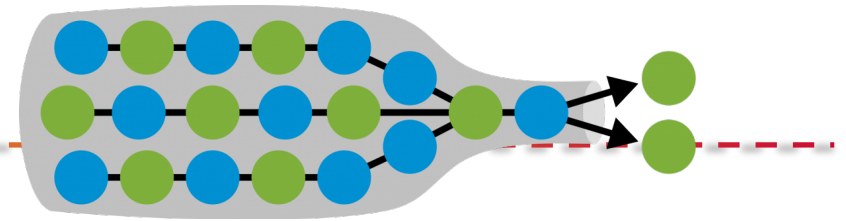
DevOps

Rolling Releases
Back and
Forward

Multiple (DB-)
Servers



Database Challenges



Database Challenges

DB Schemas

DevOps Bottleneck

Scripts

create

modify

revert

Existing Data

Tools Overview





Liquibase – Overview

LIQUI BASE

 **SOURCE CONTROL
FOR YOUR DATABASE**

DB Changes

Deploy, Rollback
Track, Manage, Document
DDL, Sources, Data

Supported Databases

Oracle
MySQL, PostgreSQL, SQL Server, DB2 &
more

Incremental Changes

Change sets / Change logs
File formats
YAML, JSON, XML, SQL
Changelog Tables inside DBs

Java based Open Source

Commercial Product: [Liquibase.com](https://liquibase.com)



Ansible in a Nutshell



A N S I B L E

Agentless IT Automation

Open Source

Based on python

by RedHat

Parallel execution

Linux / OS X
via ssh

Windows
PowerShell Remoting

Ansible Playbooks

describe the desired state

YAML – Human and Machine readable

Config Files: Formats



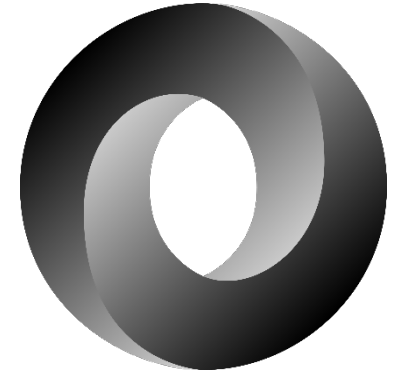


YAML & JSON: two related file formats

```
databaseChangeLog:
- changeSet:
  id: 2
  author: nvoxland
  changes:
  - addColumn:
    tableName: person
    columns:
    - column:
      name: username
      type: varchar(8)
```

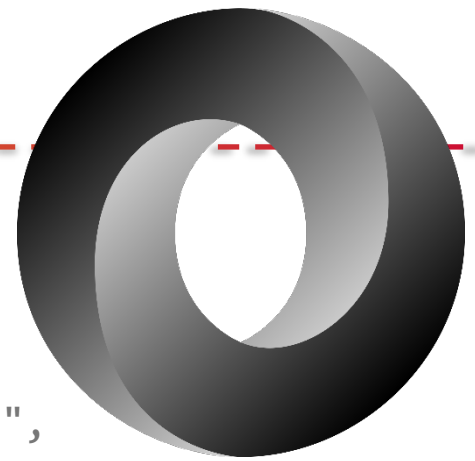
YAML

```
{ "databaseChangeLog": [
  { "changeSet": {
    "id": "2",
    "author": "nvoxland",
    "changes": [
      { "addColumn": {
        "tableName": "person",
        "columns": [
          { "column": {
            "name": "username",
            "type": "varchar(8)"
          }
        ]
      }
    ]
  }
}
]
}
```





JSON



JSON: Java Script Object Notation

Data: "Key": "Value"

Brackets: {} - Groupings

[] - Arrays

Schemale
ss

```
{ "databaseChangeLog": [  
  { "changeSet": {  
    "id": "2",  
    "author": "nvoxland",  
    "changes": [  
      { "addColumn": {  
        "tableName": "person",  
        "columns": [  
          { "column": {  
            "name": "username",  
            "type": "varchar(8)"  
          }  
        ]  
      }  
    ]  
  }  
}]  
}
```



YAML



YAML:
YAML Ain't Markup Language

Human readable	structured data format
Data Types	Scalar (Key: Value) Sequences (Arrays) Mappings
Grouping	Whitespace is important Brackets as in JSON
JSON	can be valid YAML

```
databaseChangeLog:
- changeSet:
  id: 2
  author: nvoxland
  changes:
  - addColumn:
    tableName: person
    columns:
    - column:
      name: username
      type: varchar(8)
```



Ansible



Ansible – The Basics

Only on Control Machine
Python 2 or 3
Windows not supported
Packages are available
yum install ansible
apt-get install ansible
...

Installation



ansible
ansible-playbook
ansible-console
REPL for debugging
ansible-vault
Encrypts sensitive Files
...

Command
Line Interface





Ansible Inventory

Inventory lists the Hosts Ansible can Change

Default location	/etc/ansible/hosts
File formats	ini YAML
Grouping	
Static or dynamic	
Variables	per Host Per Group
connection types	ssh (default) local, docker

```
mail.example.com
jumper ansible_port=5555 ansible_host=192.0.2.50
```

```
[webservers]
foo.example.com http_port=80 maxRequestsPerChild=808
bar.example.com http_port=303 maxRequestsPerChild=909
www[01:50].example.com
```

```
[dbservers]
one.example.com
two.example.com
three.example.com
db-[a:f].example.com
```

```
[dbservers:vars]
ntp_server=ntp.atlanta.example.com
proxy=proxy.atlanta.example.com
halon_system_timeout=30
self_destruct_countdown=60
escape_pods=2
```

```
[CoolApp:children]
dbservers
webservers
```



INVENTORY		
STATUS	MONTH	DETAILS
✓	Thruaf	hbgvks
✓	efgel fu	jbdef
✓	fdyke	hbnvj bs
	hdvhs	dfgsh
	efhgl	ghmtd
	ed gudfg	fdghghdf



Playbook

Task Sequence

single steps

describe **desired state**

YAML (default)

Modules

Abstract System tasks

Over 1300 predefined

Write your own

Variable values

Defined in Playbook

Facts gathered from Hosts

Passed from Inventory

```

- hosts: stredax.dev
  remote_user: root
  tasks:
  - name: Determine old APEX dirs
    shell: ls -ld /app/apex/{5*,4*,o*} 2>/dev/null
    ignore_errors: True
    register: apex_dirs
  - name: Remove old APEX dirs
    file:
      path: "{{ item }}"
      state: absent
      with_items: "{{ apex_dirs.stdout_lines }}"
  - name: unzip apex and ords
    unarchive:
      src: ./apex_ords.tar.gz
      dest: /app/apex
  - name: Link ords
    file:
      path: /app/apex/ords
      src: /app/apex/ords.{{ ords_version }}
      state: link
  - name: Change Oracle PWDs
    script: change_ords_pwd.sh
    remote_user: orastred

```



Playbook

Variables

Jinja2 templating

Filters available

Loops

Iterate: {{ item }}

Define: with_items

Task Results
(register)

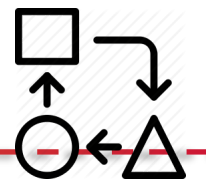
```

hosts: stredax.e
remote_user: root
tasks:
- name: Determine old APEX dirs
  shell: ls -ld /app/apex/{5*,4*,o*} 2>/dev/null
  ignore_errors: True
  register: apex_dirs
- name: Remove old APEX dirs
  file:
    path: "{{ item }}"
    state: absent
  with_items: "{{ apex_dirs.stdout_lines }}"
- name: unzip apex and ords
  unarchive:
    src: ./apex_ords.tar.gz
    dest: /app/apex
- name: Link ords
  file:
    path: /app/apex/ords
    src: /app/apex/ords.3.0.10
    state: link
- name: Change Oracle PWDs
  script: change_ords_pwd.sh
  remote_user: orastred

```



Ansible Playbooks (3/3)



```

---
- hosts: webservers
vars:
- docroot: /var/www/stredax/public
tasks:
- name: Install Nginx
  apt:
    pkg: nginx
    state: installed
    update_cache: true
  register: nginxinstalled
  notify:
    - Start Nginx
- name: Create Web Root
  when: nginxinstalled|success
  file:
    dest: {{ docroot }}
    mode: 775
    state: directory
    owner: www-data
    group: www-data
  notify:
    - Reload Nginx

```

handlers:

- name: Start Nginx
service: name=nginx state=started
- name: Reload Nginx
service: name=nginx state=reloaded

Playbook

Conditions

When result|
success/failed/skipped
"foo" == "bar"

Notify

Handlers

Tasks

Executed when notified



Ansible Roles

Roles

organize multiple related tasks and data

predefined folder structure:
`roles/rolename/...` →

ansible executes file
`main.yml`
automatically, if present

ansible galaxy :
community roles
repository

defaults

Vars with default Values

files

Files to be copied unchanged

handlers

Playbook Handlers

meta

Dependencies

tasks

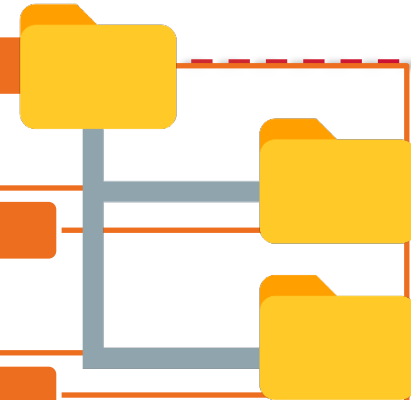
Playbook Tasks

templates

Content with Variables, e.g. config files.
Will be processed by Jinja2 template engine

vars

Variables



Liquibase



Liquibase.org



Liquibase – a closer Look

Liquibase

Installation

[Download Liquibase](#)

unpack

Requirements:

Java 8 or
Java 11 or higher

JDBC driver

Command Line Interface

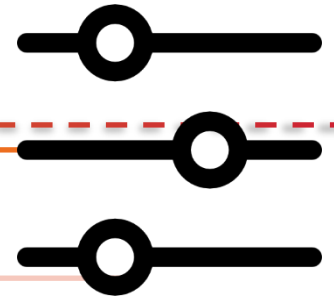
liquibase

update[SQL]
rollback[SQL]
diff / generate
dbDoc



Java





Parameters

Required

`--changeLogFile=<path>`

`--username=<value>`

`--password=<value>`

`--url=<value>`

`--driver=<jdbc.driver.ClassName>`

liquibase
.properties

working dir

Parameter Default Values



Defining Changes in Liquibase: Changelog

Changelogs

contain

attributes

preConditions

changesets

nestable

include

keeping
Track

Table
databasechangelog

```
databaseChangeLog:
```

```
- preConditions:
```

```
  dbms type: oracle
```

```
  runningAs:
```

```
    username: ROBBIE
```

```
- include:
```

```
  file: robbie-changeset-1.0.yaml  
  relativeToChangelogFile: true
```

```
- include:
```

```
  file: robbie-changeset-1.1.yaml  
  relativeToChangelogFile: true
```

```
- include:
```

```
  file: robbie-changeset-2.0.yaml  
  relativeToChangelogFile: true
```



Defining Changes in Liquibase: Changeset

changesets

contain

Attributes

Actual changes

Rollback

identified

ID

Author

Changelog Filename

runOnChange

Executes on change

Checksum in Table

PL/SQL code

```
databaseChangeLog:
- changeSet:
  id: TabStockTicker-1
  author: robbie
  changes:
  - createTable:
    tableName: StockTicker
    columns:
    - column:
      name: id
      type: number
      autoIncrement: true
      constraints:
        primaryKey: true
        nullable: false
    - column:
      name: symbol
      type: varchar2(50)
```



Liquibase Changes

Changes

Basic Operations

Bundled

With rollback

Example

createTable

SQL / SQLFile

Extensions

Oracle

Build your own

databaseChangeLog:

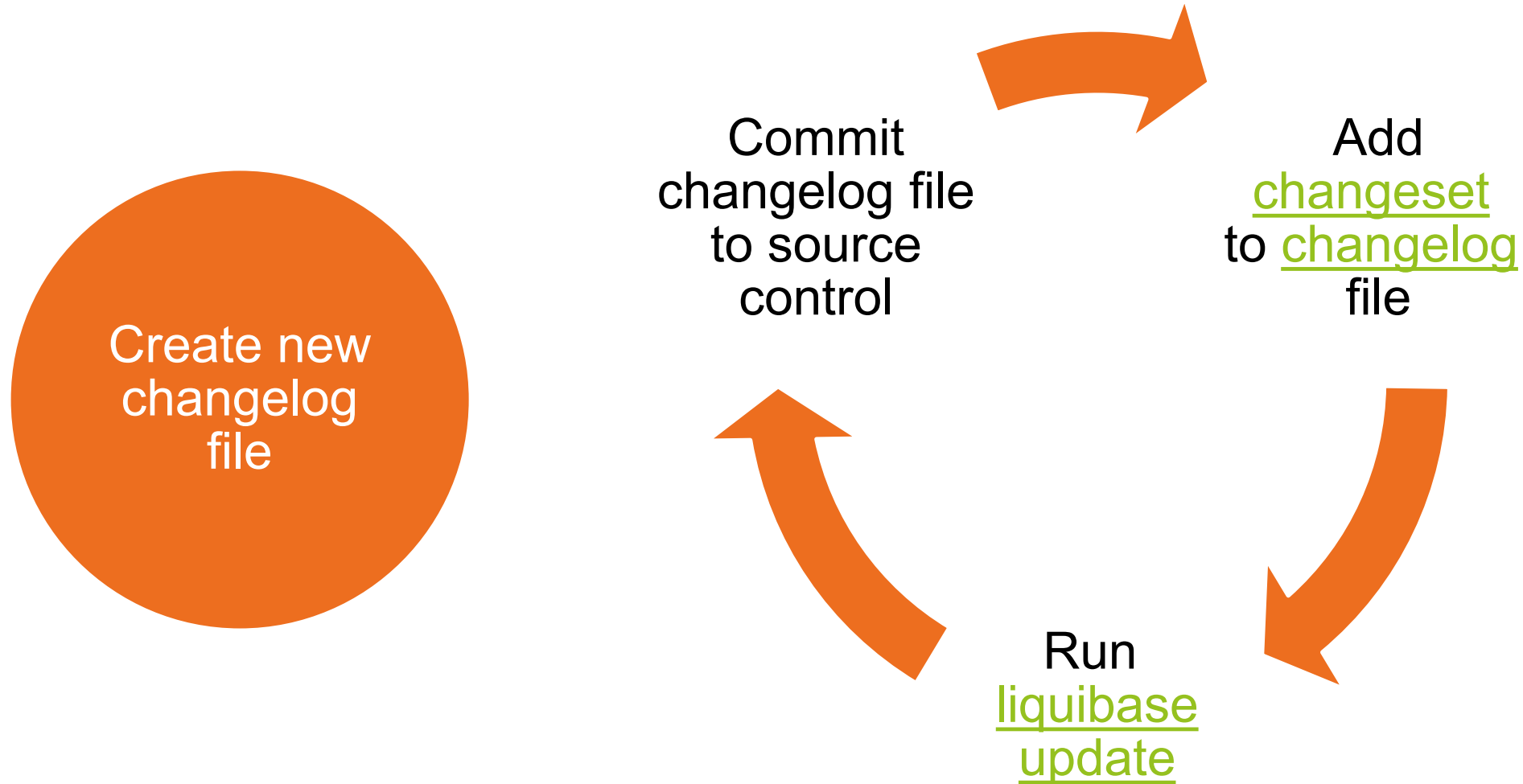
```

- changeSet:
  id: TabTab1-1
  author: robbie
  changes:
    - createTable:
      tableName: Tab1
      columns:
        - column:
          name: id
          type: number
  rollback:
    - dropTable:
      tableName: Tab1
- changeSet:
  id: TabTab1-2
  author: robbie
  changes:
    - dropTable:
      tableName: Tab1
  rollback:
    changeSetId: TabTab1-1
    changeSetAuthor: robbie

```



Liquibase Workflow





Liquibase Procedure for the developer

Edit

create new local changeSet

Test

Test the changeSet SQL: Run liquibase update

Change

Change your Application: Edit the AppCode

Test

Test application code & database change together

Commit

Commit changeSet and application code.



Liquibase Best Practices

Use a master changelog



One change per changeset



PL/SQL Code with `runOnChange="true"`



IDs are literals – use them



Document changesets with comments



Always think about rollback



Leverage Liquibase to manage your Reference Data



Combining the Tools





Using Liquibase with Ansible

Deploy Liquibase to Targets

see example on the right

Execute liquibase update on target

File Module: Transport changelogs
Shell Module: run liquibase

Run Liquibase from Control Host

Local_action with jdbc URLs
See Blog Post

ssh Tunnels

let ansible create ssh tunnels
Run Liquibase as local action
connecting to localhost
See Blog Post

```
---  
- name: Create Liquibase directory  
  file:  
    name: /opt/liquibase  
    state: directory  
    mode: 0755  
  
- name: Install Liquibase  
  unarchive:  
    src:  
      https://github.com/liquibase/liquibase/releases  
      /download/liquibase-parent-{{ liquibase_version  
      }}/liquibase-{{ liquibase_version }}-bin.tar.gz  
    dest: /opt/liquibase  
    copy: no  
    mode: 0755  
    creates: /opt/liquibase/liquibase
```



Demo

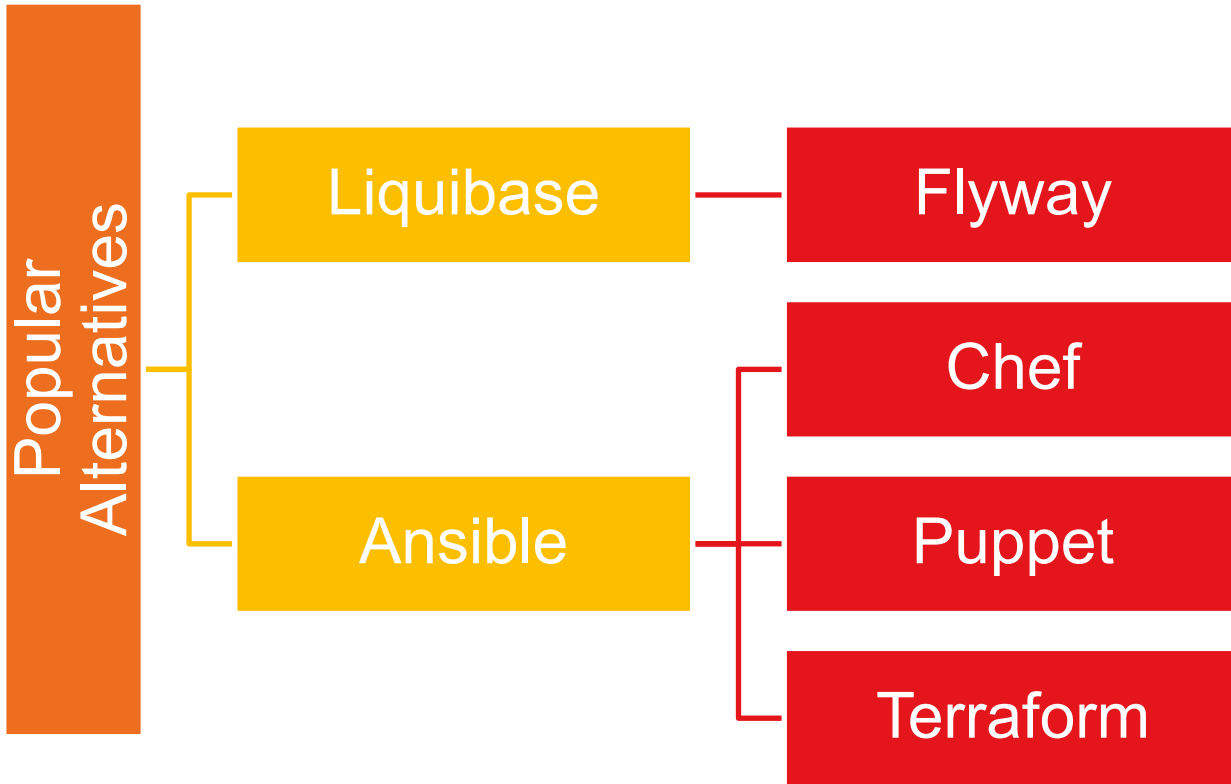


Conclusion





Alternatives





Want to know more?

Liquibase

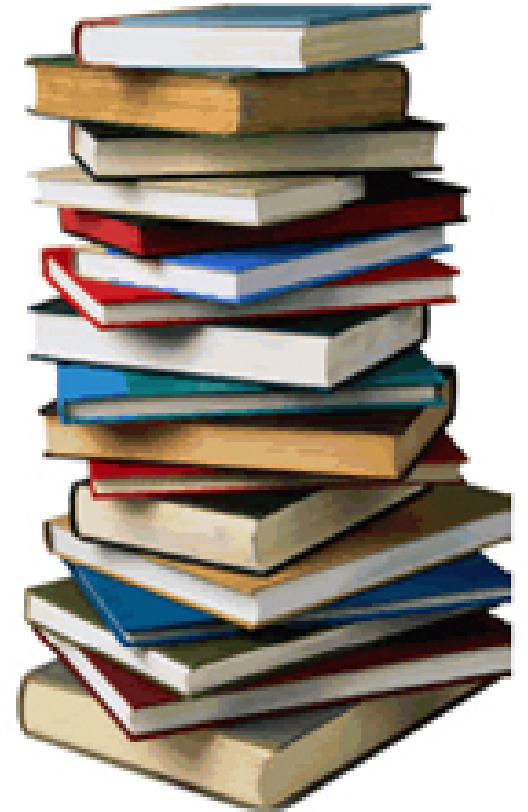
- [Homepage](#)
- [Official Documentation](#)
- [Tutorial: Change Happens by Blaine Carter](#)

Ansible

- [Homepage](#)
- [Official Documentation](#)
- [Free Webinar](#)

YAML

- [Tutorial](#)
- [Tutorial Learn X in Y Minutes](#)
- [Reference Card](#)
- [YAML 1.2 Spec](#)





Conclusion



Overcome
Database
DevOps
Bottleneck

PLEASE

DO

TRY THIS

AT HOME