BROADCOM SOFTWARE

Make your IMS life more attractive again.

Radek Mrvec

Date: 12.10.2021

Disclaimer

Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of August 2, 2021 and is **subject to change or withdrawal by CA at any time without notice**. **The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion**.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. Broadcom assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. **In n**o event will Broadcom be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if Broadcom is expressly advised in advance of the possibility of such damages.

Agenda

- Welcome to the world of mainframe
- Ansible on Z intro
- Z Open Automation Utilities intro
- Changing our IMS world

Welcome to the world of mainframe

BROADCOM SOFTWARE

Current US Mainframe world stats

45

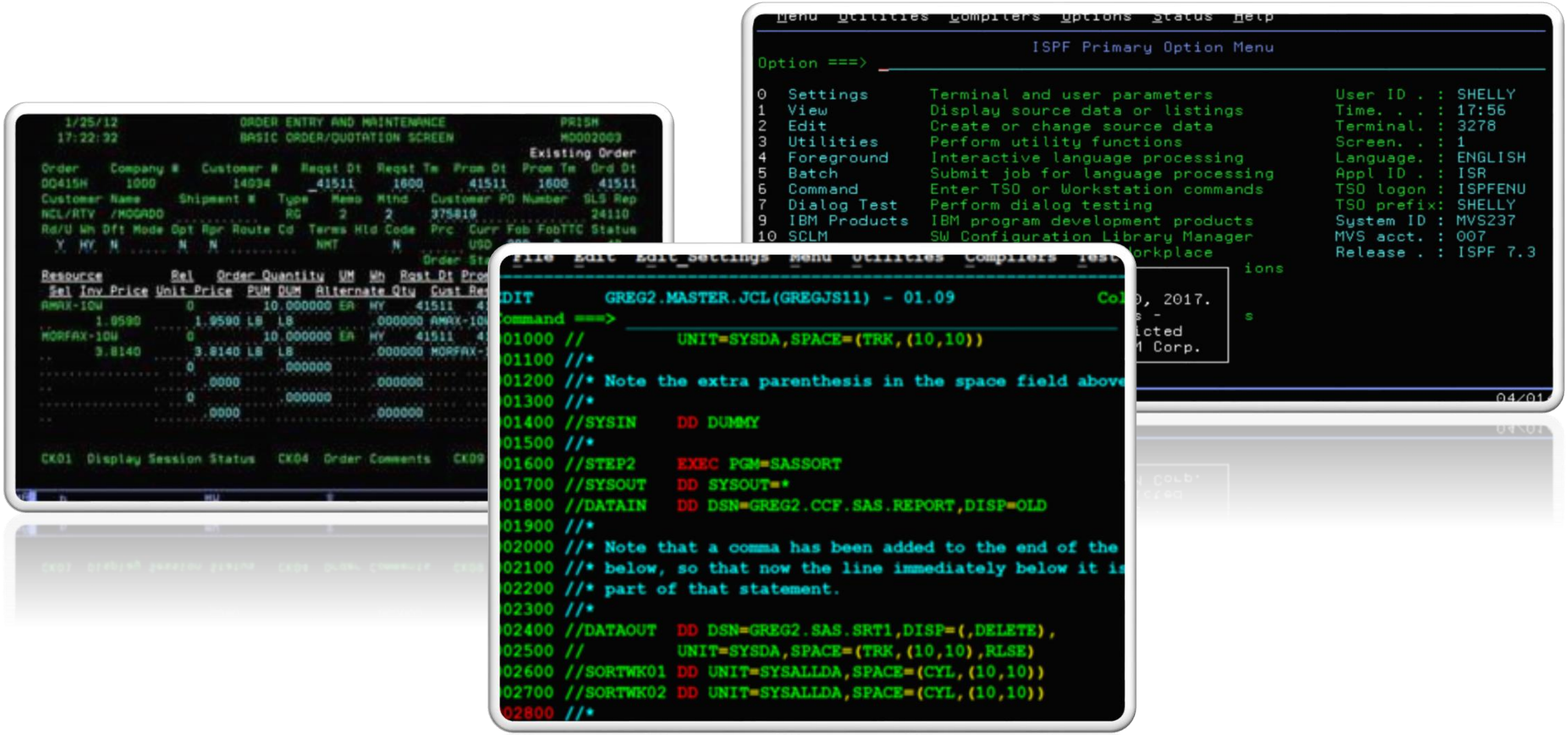
the average age of a mainframe developer

69%

of developers leave mainframe in the first 2 years

https://www.zippia.com/mainframe-programmer-jobs/demographics/

Green screen everywhere



```
   1/25/12                 ORDER ENTRY AND MAINTENANCE              PRISM
  17:22:32                 BASIC ORDER/QUOTATION SCREEN             MD002003
                                                          Existing Order
Order     Company #   Customer #   Reqst Dt  Reqst Tm  Prom Dt  Prom Tm  Ord Dt
DQ415H     1000          14034     _41511     1600      41511    1600    41511
Customer Name     Shipment #   Type   Memo  Mthd  Customer PO Number  SLS Rep
NCL/RTY  /MOGADO                RG      2     2    375819             24110
Rd/U Wh Dft Mode Opt Rpr Route Cd  Terms Hld Code  Prc  Curr Fob FobTTC Status
  Y  MY  N        N   N            NMT        N          USD
                                                      Order St
Resource          Rel   Order Quantity  UM  Wh  Rqst Dt Prom
 Sel Inv Price Unit Price  PUM DUM  Alternate Qty  Cust Re
AMAX-1OW           0         10.000000 EA  MY    41511   4
      1.0590        1.9590 LB  LB            .000000 AMAX-10
MORFAX-10W         0         10.000000 EA  MY    41511   4
      3.8140        3.8140 LB  LB            .000000 MORFAX-
                   0          .000000
                      .0000                  .000000
                   0          .000000
                      .0000                  .000000

CK01  Display Session Status   CK04  Order Comments   CK09
```

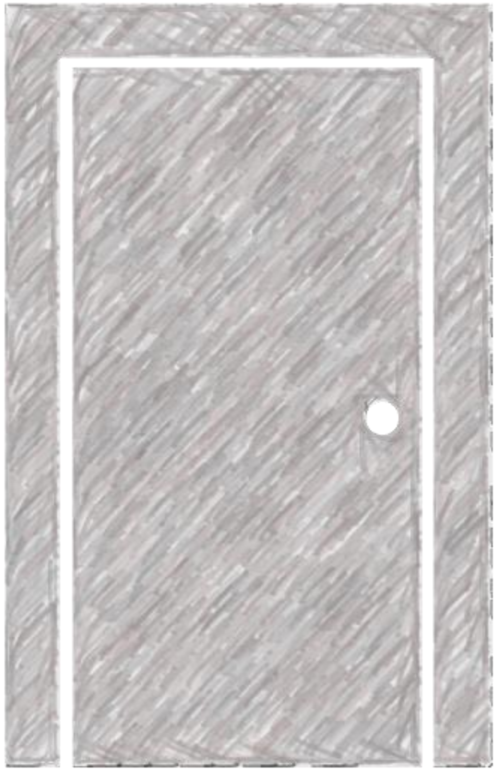
```
 Menu  Utilities  Compilers  Options  Status  Help
                      ISPF Primary Option Menu
Option ===> _

0  Settings      Terminal and user parameters        User ID . : SHELLY
1  View          Display source data or listings     Time. . . : 17:56
2  Edit          Create or change source data        Terminal. : 3278
3  Utilities     Perform utility functions           Screen. . : 1
4  Foreground    Interactive language processing     Language. : ENGLISH
5  Batch         Submit job for language processing  Appl ID . : ISR
6  Command       Enter TSO or Workstation commands   TSO logon : ISPFENU
7  Dialog Test   Perform dialog testing              TSO prefix: SHELLY
9  IBM Products  IBM program development products    System ID : MVS237
10 SCLM          SW Configuration Library Manager    MVS acct. : 007
                                 orkplace            Release . : ISPF 7.3
                                            ions
                    0, 2017.
                    s -                     s
                    icted
                    M Corp.

                                                                    04/01
```

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test
DIT       GREG2.MASTER.JCL(GREGJS11) - 01.09                 Col
ommand ===>
01000 //          UNIT=SYSDA,SPACE=(TRK,(10,10))
01100 //*
01200 //* Note the extra parenthesis in the space field above
01300 //*
01400 //SYSIN     DD DUMMY
01500 //*
01600 //STEP2     EXEC PGM=SASSORT
01700 //SYSOUT    DD SYSOUT=*
01800 //DATAIN    DD DSN=GREG2.CCF.SAS.REPORT,DISP=OLD
01900 //*
02000 //* Note that a comma has been added to the end of the
02100 //* below, so that now the line immediately below it is
02200 //* part of that statement.
02300 //*
02400 //DATAOUT   DD DSN=GREG2.SAS.SRT1,DISP=(,DELETE),
02500 //          UNIT=SYSDA,SPACE=(TRK,(10,10),RLSE)
02600 //SORTWK01  DD UNIT=SYSALLDA,SPACE=(CYL,(10,10))
02700 //SORTWK02  DD UNIT=SYSALLDA,SPACE=(CYL,(10,10))
02800 //*
```

How feel graduates in a green world

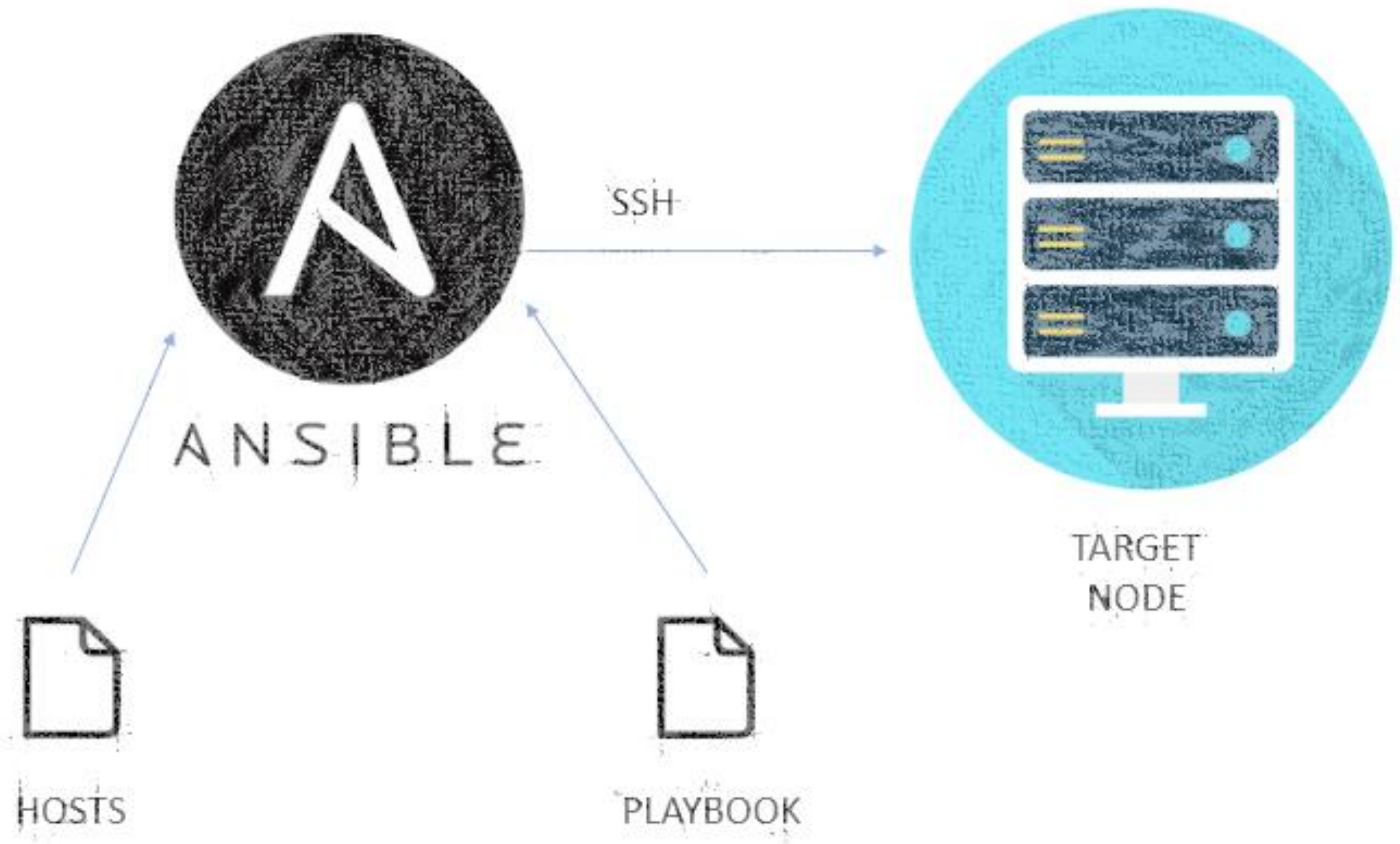
What to remember?

Old ways won’t open new doors



Ansible on Z intro

How does Ansible work



What do you need to run Ansible on Z

TOP SECRET
OpenSSH
SSH...

How to define your targets

```
[webservers]
foo.example.com
bar.example.com

[dbservers]
one.example.com
two.example.com
three.example.com
```

What do playbooks look like

```
name: Playbook submiting batch job

  hosts: ca11
  gather_facts: no
  environment:
      "{{default_environment}}"

  collections:
    - ibm.ibm_zos_core

  tasks:
    - name: Submit a JCL located in DATASET
      zos_job_submit:
        src: TEST.IMSTOOLS.DBA.JCL(ANALYZE)
        location: DATA_SET
        wait: true
        return_output: true
        wait_time_s: 30
      register: job_detail
```

Configuration is always needed

```
ansible_port: 22                                  # SSH Port
ansible_user:                                     # USER username
ansible_password:                                 # USER password

ansible_ssh_pipelining: True                      # needed for encoding on ssh connections

ansible_python_interpreter: "/usr/lpp/IBM/cyp/v3r9/pyz/bin/python3"

PYZ: "/usr/lpp/IBM/cyp/v3r9/pyz"
ZOAU: "/usr/lpp/IBM/zoautil"

default_environment:
  _BPXK_AUTOCVT: "ON"
  ZOAU_HOME: "{{ ZOAU }}"
  PYTHONPATH: "{{ ZOAU }}/lib"
  LIBPATH: "{{ ZOAU }}/lib:{{ PYZ }}/lib:/lib:/usr/lib:."
  PATH: "{{ ZOAU }}/bin:{{ ZOAU }}/env/bin:{{ PYZ }}/bin:/bin:/var/bin:/usr/lpp/java/J8.0/bin"
  _CEE_RUNOPTS: "FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
  _TAG_REDIR_ERR: "txt"
  _TAG_REDIR_IN: "txt"
  _TAG_REDIR_OUT: "txt"
  LANG: "C"
```

Ansible modules on Mainframe

ibm_zos_sysauto

- The IBM Z System Automation collection includes roles and sample playbooks to access the IBM Z System Automation Operations REST server.

ibm_zos_core

- The IBM z/OS core collection includes connection plugins, action plugins, modules, filters, sample playbooks and ansible-doc to automate tasks on z/OS.

ibm_zos_zosmf

- Ansible collection consisting of modules and roles to work with z/OS based on z/OS Management Facility (z/OSMF).

ibm_zos_ims

- The IBM z/OS IMS collection includes modules and sample playbooks to automate tasks for IBM IMS.

ibm_zos_cics

- The Red Hat Ansible Certified Content for IBM Z CICS collection includes connection plugins, action plugins, modules and sample playbooks to automate tasks for CICS

Z Open Automation Utilities intro

Z Open Automation utilities description

- Set of utilities for z/OS which are modeled after similar standard Linux commands
- Java API support
- Python API support

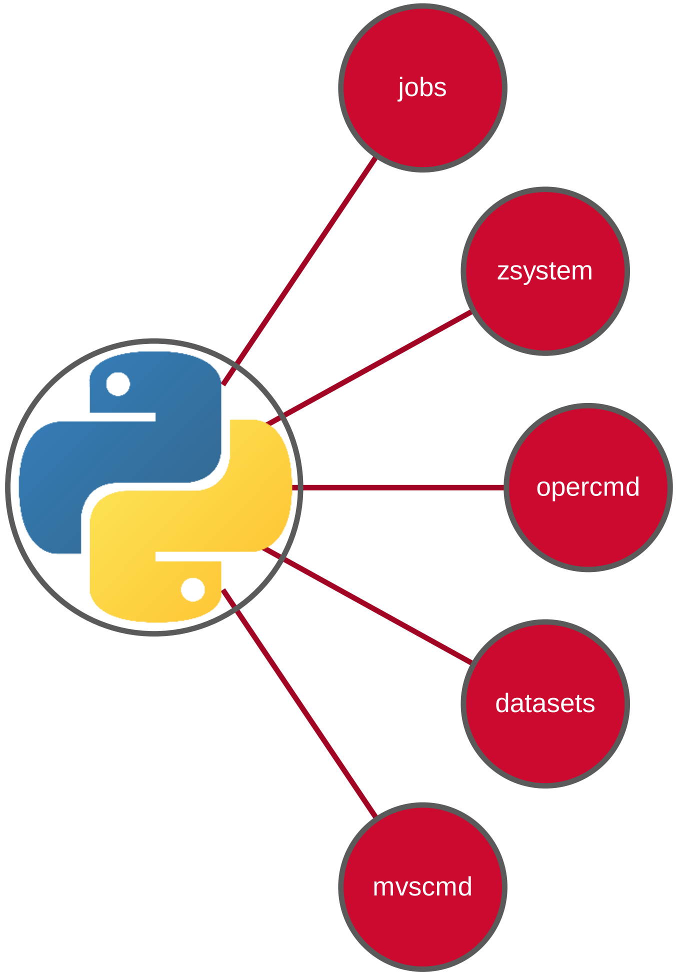
Examples of Z Open Automation Utilities commands

- dls “TEST.IMSTOOLS.JCL“
  - *Check if data set exists*

- hlq
  - *Display current HLQ*

- opercmd “d a”
  - *Issue operator command for “DISPLAY A”*

Z Open Automation Utilities Python API



Z Open Automation Utilities Python API calls

- Datasets.exists(“TEST.IMSTOOLS.JCL")
  - *Check if data set exists*
- print(Datasets.hlq())
  - *Display current HLQ*
- opercmd.execute(command="d", parameters="a")
  - *Issue operator command for “DISPLAY A”*

Changing our IMS world

BROADCOM®
SOFTWARE

Connect all together

ANSIBLE + ZOAU + IBM IMS

Ansible Galaxy offering for IMS

ims_catalog_populate

- Add records to the IMS Catalog

ims_acb_gen

- Generate IMS ACB

ims_psb_gen

- Generate IMS PSB

ims_dbrc

- Submit IMS DBRC Commands

ims_command

- Submit IMS Commands

ims_dbd_gen

- Generate IMS DBD

ims_catalog_purge

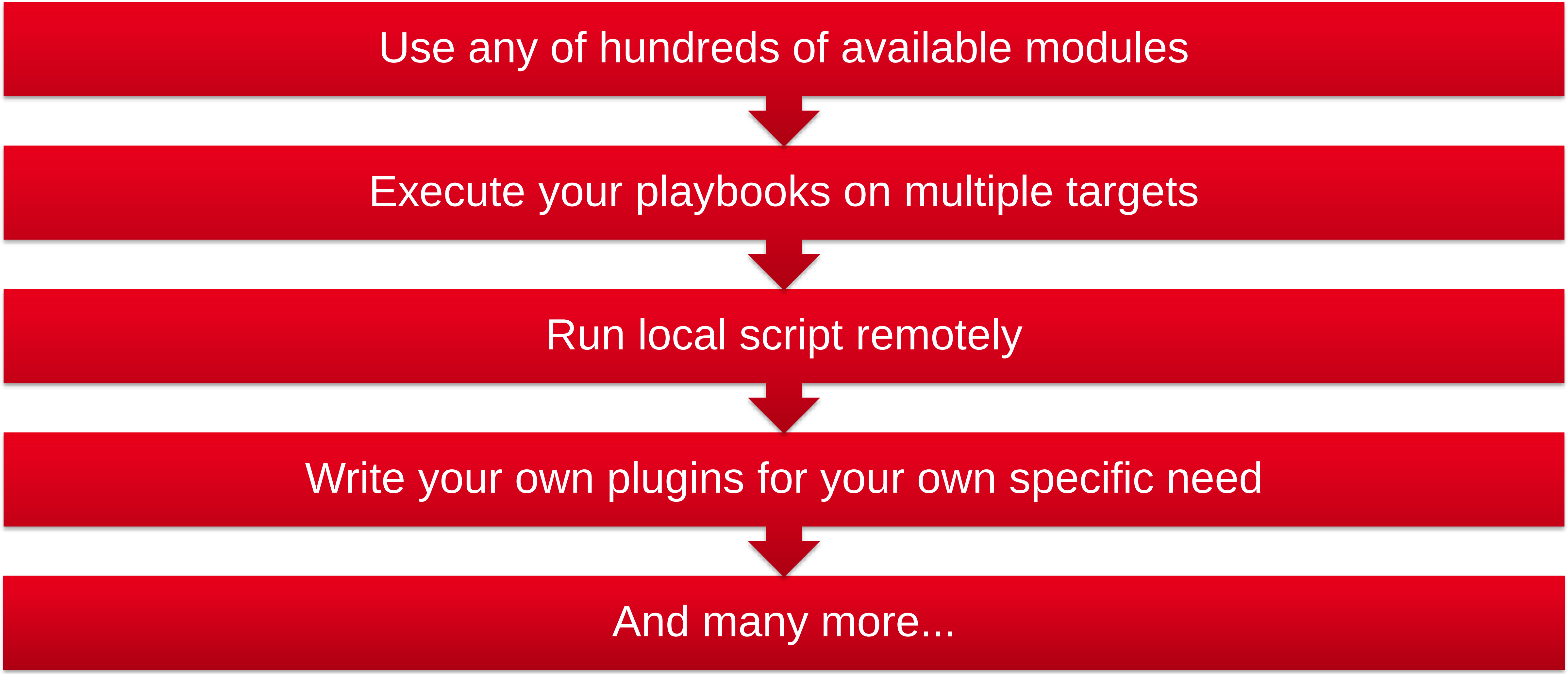
- Purge records from the IMS Catalog

Make a difference

```
//GTDBD      JOB (112300000),'GEN TANK DBD',
//             MSGCLASS=A,CLASS=A,MSGLEVEL=(1,1),REGION=0M
//C      EXEC PGM=ASMA90,REGION=0M,
//              PARM=(OBJECT,NODECK,NODBCS,'SIZE(MAX,ABOVE)')
//SYSLIB   DD DSN=IMSSYS15.SDFSMAC,DISP=SHR
//SYSLIN   DD UNIT=SYSDA,DISP=(,PASS),
//              SPACE=(80,(100,100),RLSE),
//              DCB=(BLKSIZE=80,RECFM=F,LRECL=80)
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,DISP=(,DELETE),
//              SPACE=(CYL,(10,5))
//C.SYSIN  DD DSN=TEST.IMSWK.SRC(ACCTDBDA),DISP=SHR
//L       EXEC PGM=IEWL,PARM='XREF,LIST',
//               COND=(0,LT,C),REGION=4M
//SYSLIN    DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//SYSPRINT  DD SYSOUT=*
//L.SYSLMOD DD DSN=TEST.IMSWK.DBDLIB(ACCTDBDA),DISP=SHR
//SYSLMOD   DD DISP=SHR,
//               DSN=TEST.IMSWK.DBDLIB(ACCTDBDA)
//SYSUT1    DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),
//               SPACE=(1024,(100,10),RLSE),DISP=(,DELETE)
```

```
- name: Playbook for DBDGEN

  hosts: ca11
  gather_facts: no
  environment:
      "{{default_environment}}"

  collections:
    - ibm.ibm_zos_core
    - ibm.ibm_zos_ims

  tasks:
    - name: DBDGEN task
      ims_dbd_gen:
        src: IDI.TEST.GSE.SRC
        'replace': true
        member_list:
          - 'ACCTDBDA'
        dest: IDI.TEST.GSE.DBDLIB
        sys_lib:
          - IMSSYS15.SDFSMAC
          - SYS1.MACLIB
```

What can we use with the help of Ansible



Use any of hundreds of available modules

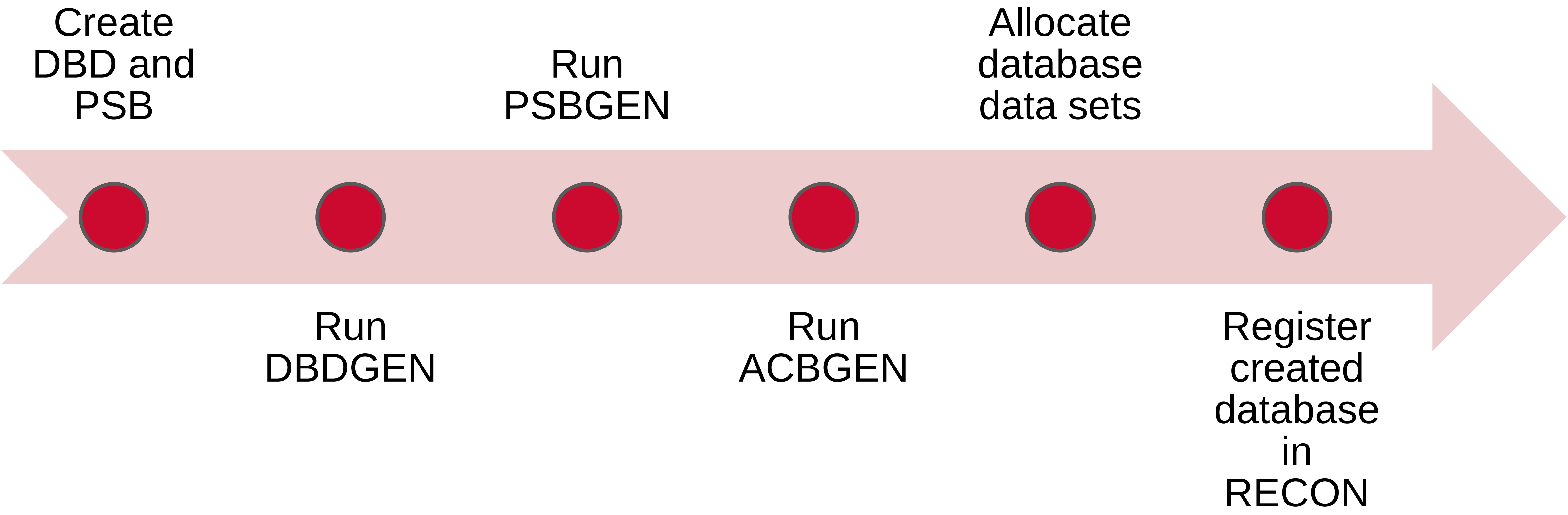
Execute your playbooks on multiple targets

Run local script remotely

Write your own plugins for your own specific need

And many more...

Example scenario



Create DBD and PSB

Run DBDGEN

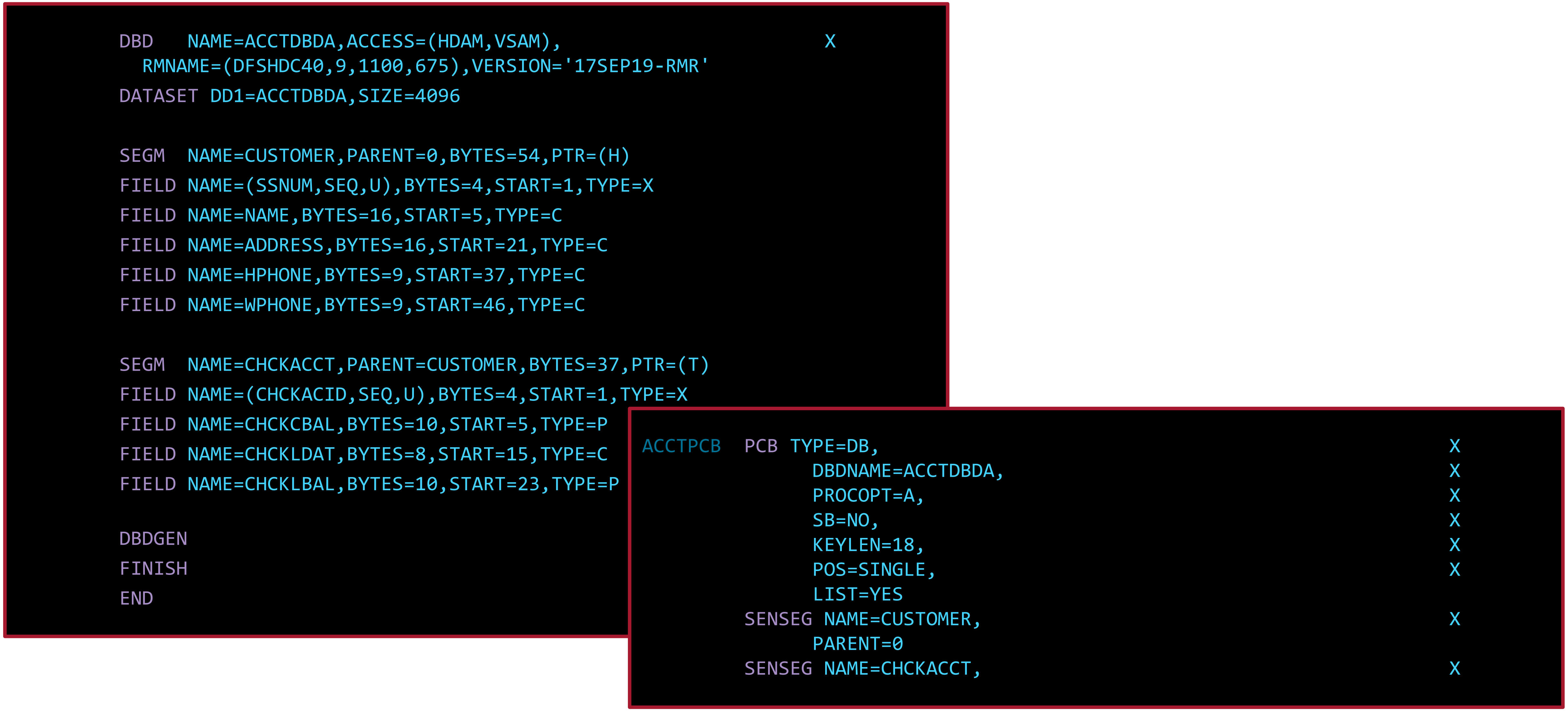
Run PSBGEN

Run ACBGEN

Allocate database data sets

Register created database in RECON

Step 1 – Create DBD and PSB



```
DBD    NAME=ACCTDBDA,ACCESS=(HDAM,VSAM),                       X
         RMNAME=(DFSHDC40,9,1100,675),VERSION='17SEP19-RMR'
DATASET DD1=ACCTDBDA,SIZE=4096

SEGM   NAME=CUSTOMER,PARENT=0,BYTES=54,PTR=(H)
FIELD NAME=(SSNUM,SEQ,U),BYTES=4,START=1,TYPE=X
FIELD NAME=NAME,BYTES=16,START=5,TYPE=C
FIELD NAME=ADDRESS,BYTES=16,START=21,TYPE=C
FIELD NAME=HPHONE,BYTES=9,START=37,TYPE=C
FIELD NAME=WPHONE,BYTES=9,START=46,TYPE=C

SEGM   NAME=CHCKACCT,PARENT=CUSTOMER,BYTES=37,PTR=(T)
FIELD NAME=(CHCKACID,SEQ,U),BYTES=4,START=1,TYPE=X
FIELD NAME=CHCKCBAL,BYTES=10,START=5,TYPE=P
FIELD NAME=CHCKLDAT,BYTES=8,START=15,TYPE=C
FIELD NAME=CHCKLBAL,BYTES=10,START=23,TYPE=P

DBDGEN
FINISH
END
```

```
ACCTPCB  PCB TYPE=DB,                                            X
               DBDNAME=ACCTDBDA,                                 X
               PROCOPT=A,                                        X
               SB=NO,                                            X
               KEYLEN=18,                                        X
               POS=SINGLE,                                       X
               LIST=YES
         SENSEG NAME=CUSTOMER,                                   X
               PARENT=0
         SENSEG NAME=CHCKACCT,                                   X
```

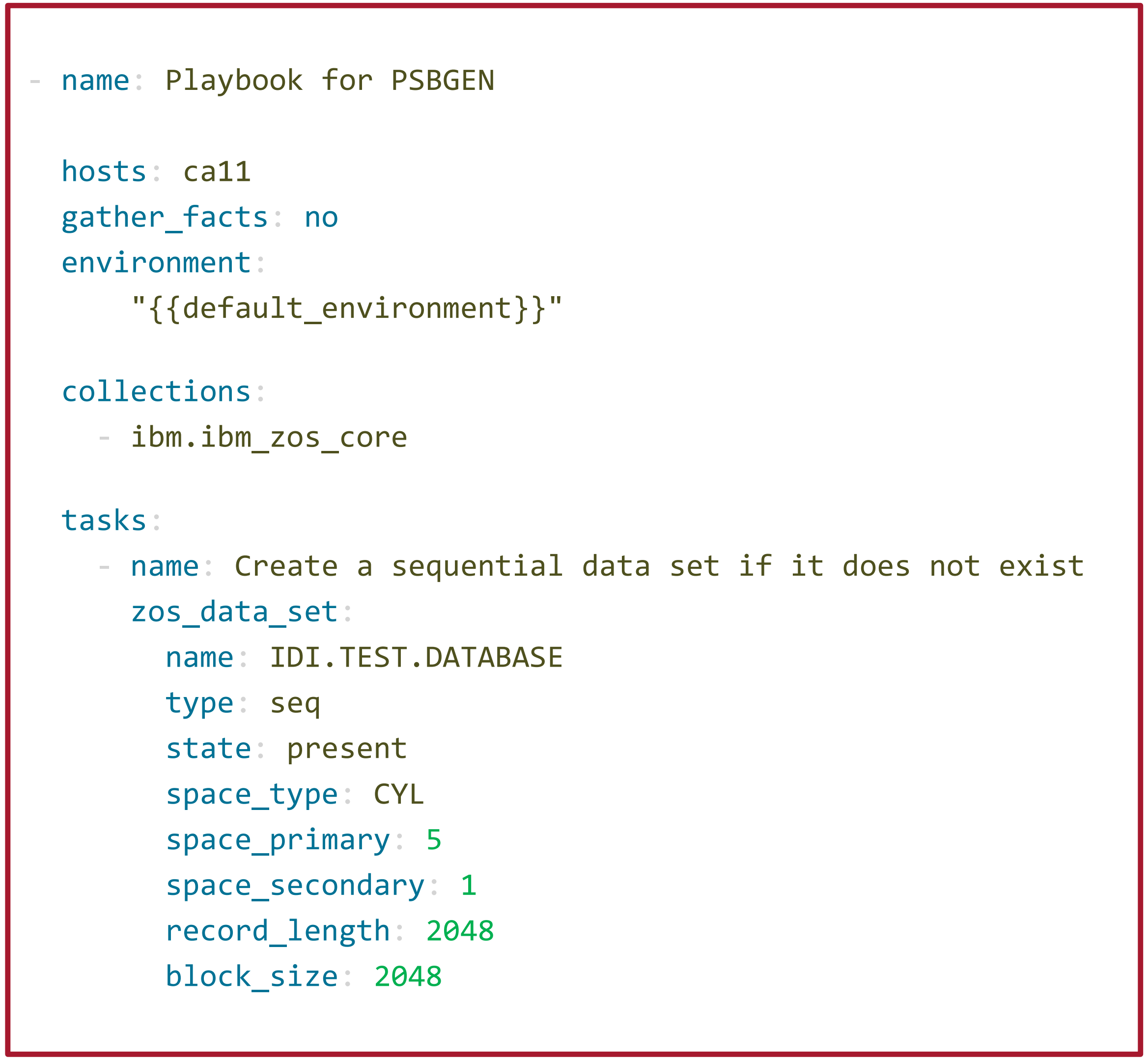
Step 2,3,4 – Run all GENs

```
- name: Playbook for DBDGEN

  hosts: ca11
  gather_facts: no
  environment:
      "{{default_environment}}"

  collections:
    - ibm.ibm_zos_core
    - ibm.ibm_zos_ims

  tasks:
    - name: DBDGEN task
      ims_dbd_gen:
        src: IDI.TEST.GSE.SRC
        'replace': true
        member_list:
          - 'ACCTDBDA'
        dest: IDI.TEST.GSE.DBDLIB
        sys_lib:
          - IMSSYS15.SDFSMAC
          - SYS1.MACLIB
```

```
  collections:
    - ibm.ibm_zos_core
    - ibm.ibm_zos_ims

  tasks:
    -
 name: Example of creating ACBs for spec
      ims_acb_gen:
        command_input: BUILD
        psb_name:
          - PACCTDBA
        psb_lib:
          - IDI.TEST.GSE.PSBLIB
        dbd_lib:
          - IDI.TEST.GSE.DBDLIB
        acb_lib: IDI.TEST.GSE.ACBLIB
        reslib:
          - IMSSYS15.SDFSRESL
        steplib:
          - IMSSYS15.SDFSRESL
•           build_psb: false
```

```
- name: Playbook for PSBGEN

  hosts: ca11
  gather_facts: no
  environment:
      "{{default_environment}}"

  collections:
    - ibm.ibm_zos_core
    - ibm.ibm_zos_ims

  tasks:
    - name: PSBGEN task
      ims_psb_gen:
        src: IDI.TEST.GSE.SRC
        'replace': true
        member_list:
          - 'PACCTDBA'
        dest: IDI.TEST.GSE.PSBLIB
        sys_lib:
          - IMSSYS15.SDFSMAC
          - SYS1.MACLIB
```

Step 5 – Allocate database data set

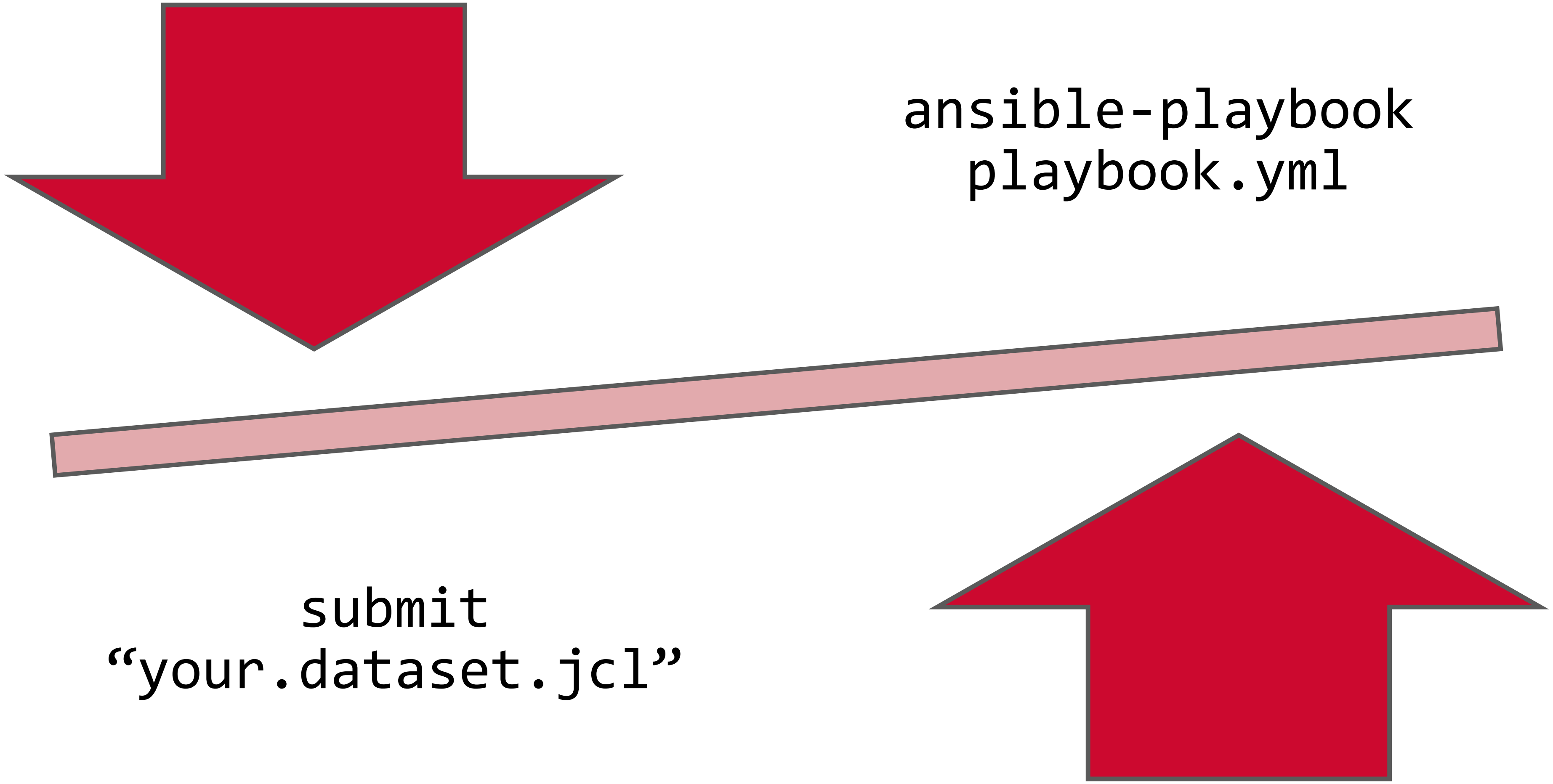


```
- name: Playbook for PSBGEN

  hosts: ca11
  gather_facts: no
  environment:
      "{{default_environment}}"

  collections:
    - ibm.ibm_zos_core

  tasks:
    - name: Create a sequential data set if it does not exist
      zos_data_set:
        name: IDI.TEST.DATABASE
        type: seq
        state: present
        space_type: CYL
        space_primary: 5
        space_secondary: 1
        record_length: 2048
        block_size: 2048
```

Step 6 – Register database in RECON

```
- name: Playbook for DBRC

  hosts: ca11
  gather_facts: no
  environment:
      "{{default_environment}}"

  collections:
    - ibm.ibm_zos_core
    - ibm.ibm_zos_ims

  tasks:
    - name: DBRC INIT DB command
      ims_dbrc:
        command:
          - INIT.DB DBD(ACCTDBDA) SHARELVL(1) TYPEIMS
          - INIT.DBDS DBD(ACCTDBDA) DDN(ACCTDBDA) DSN(IDI.TEST.DATABASE) GENMAX(3)
        steplib:
          - IMSSYS15.SDFSRESL
        dbd_lib: IDI.TEST.GSE.DBDLIB
        recon1: IDI.TEST.RECON1
        recon2: IDI.TEST.RECON2
        recon3: IDI.TEST.RECON3
```

The decision is up to you!



```
ansible-playbook
playbook.yml
```

```
submit
“your.dataset.jcl”
```

Thank you