Social Connections

Automate IBM Connections Installations and more

Christoph Stoettner, panagenda

@stoeps

Vienna, October 16-17 2017

PLATINUM SPONSORS

IBM®

panagenda

GOLD SPONSORS

Kudos BOARDS BADGES BUZZY ANALYTICS IDEAS

INFOWARE

Appfusions™

Reverso

SILVER SPONSORS

OnTime® IF TIME MATTERS...

T-Matic computerplus

holistic-net

TIMETOACT GROUP SOFTWARE & CONSULTING

riva Riva CRM Integration

hedersoft We understand Data!

Belsoft collaboration

BCC

workplaceON

groupwave

opentopic

BRONZE SPONSORS

RAMSIT

Whitesoft

Christoph Stoettner

 @stoeps
 linkedin.com/in/christophstoettner
 www.stoeps.de
 christophstoettner
☎ +49 173 8588719
✉ christoph.stoettner@panagenda.com

IBMCHAMPION

IBM CHAMPION • 3 YEAR MILESTONE
IBM
Excellence
IBM SOCIAL BUSINESS

- Senior Consultant at panagenda
  - IBM Notes / Domino since 1999
  - IBM Connections since version 2.5 / 2009
- Experience in
  - Migrations
  - Deployments
  - Administration
  - Performance Analysis
- Focusing in
  - Deployment and Optimizing IBM Connections
  - Monitoring / panagenda ConnectionsExpert
- Husband and father
- Beer or Wine?

Let’s get started

Make Your Data Work for You

Idea and history

- Several attempts to deploy IBM Connections automatically
- Social Connections VII – Stockholm
  - Klaus Bild: The Silence of the Installers - How silent installers help you automate IBM Connections deployments
- Why do we need automation?
  - Demo Environments
  - Migrations
  - Continuous Delivery
- Ever checked the Orient Me / IBM Public Cloud installer?
- You can script it manually
  - echo ”something” > /etc/hosts
- Are all system requirements installed
- ulimit configured

Possible Opensource Tools

- Puppet
  - Great for Windows too
  - Enterprise support
  - cryptic
  - https://puppet.com/
- Chef
  - Easy to learn if you’re ruby developer
  - https://www.chef.io/
- SaltStack
  - https://saltstack.com/

Ansible

- **Agentless**
- Uses SSH
- Easy to read (Everything is YAML)
- Easy to use (Extensible via modules)
- Encryption and security built in
- Written in Python
- Supported by Red Hat and Communities

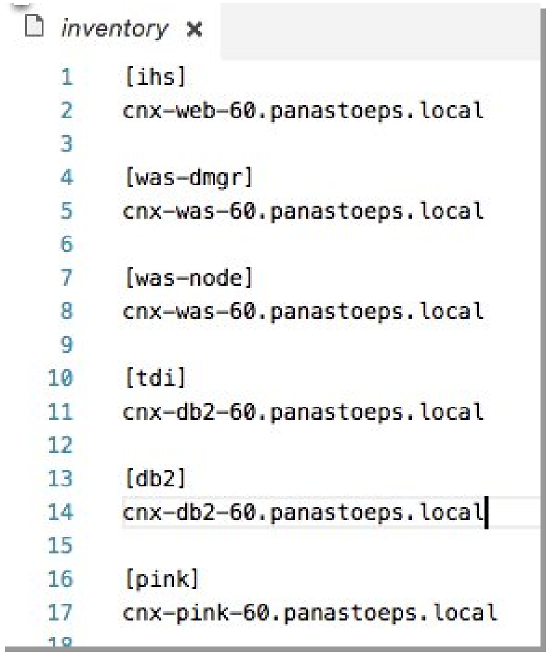
ANSIBLE

	Language	Agent	Config	Communication	Difficulty
Ansible	Python	No	YAML	OpenSSH	★☆
Chef	Ruby, Erlang	Yes	Ruby	SSL	★★★★
Puppet	Ruby	Yes	PuppetDSL	SSL	★★★☆
SaltStack	Python	Yes	YAML	ZeroMQ	★★☆

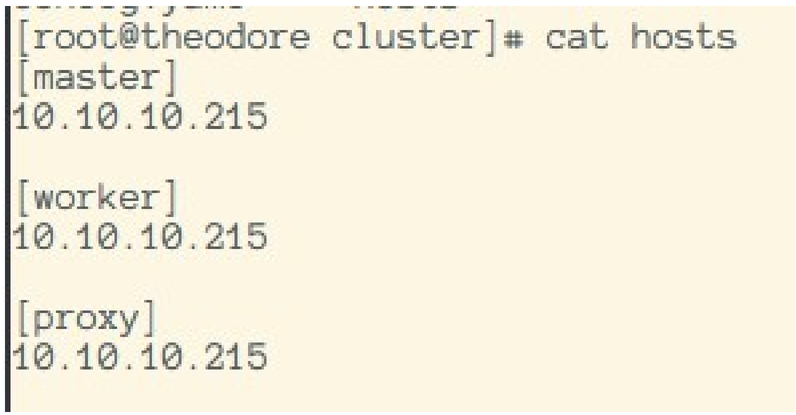
Ansible fundamentals

Why learning a new language / tool?

- Ansible is built for Cloud orchestration
- Inventory dynamic or static
  - Use playbooks for demo and production environment
- Inventory example
- It's just YAML
- Easy to keep in source control (git, svn)

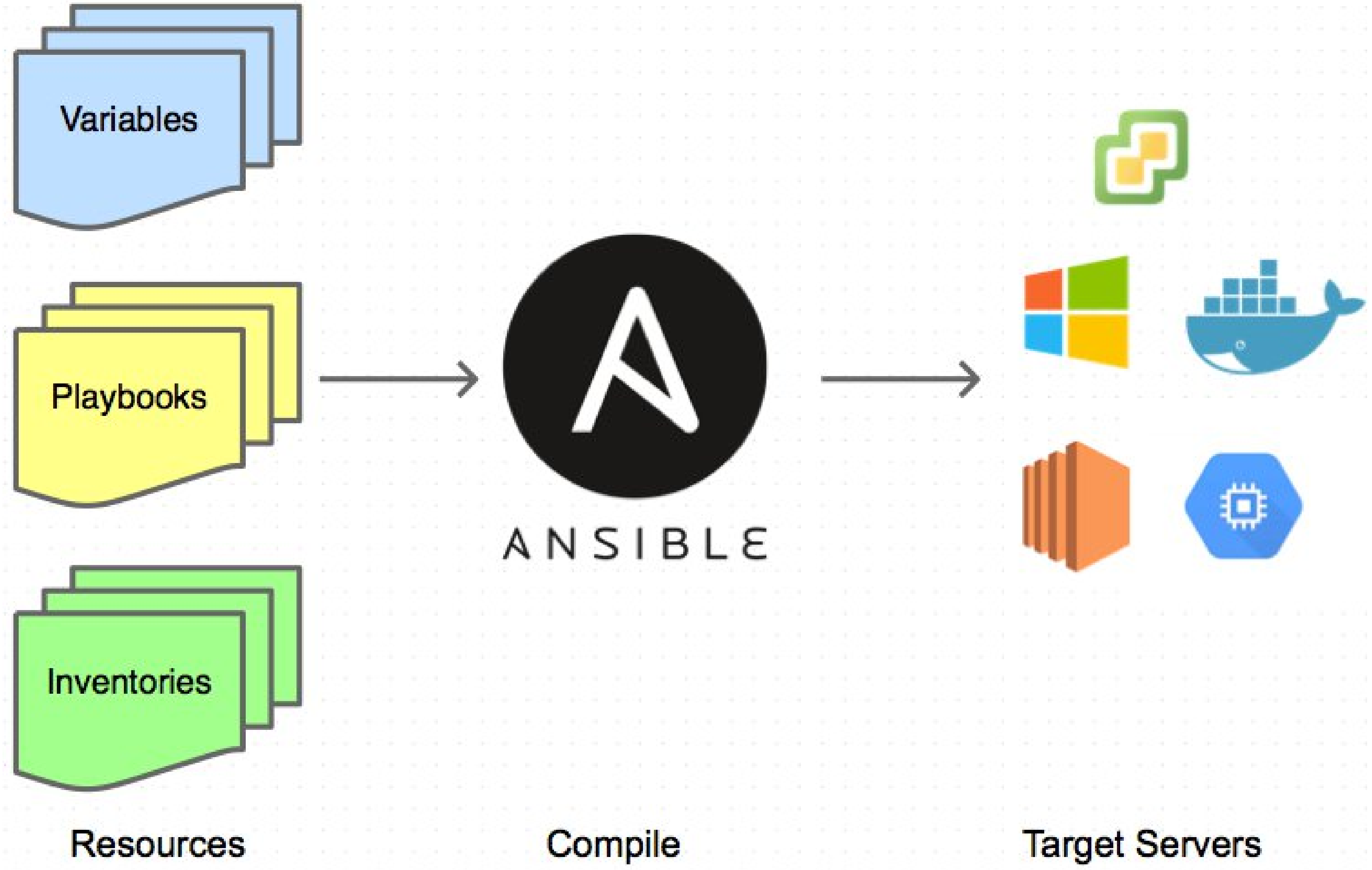


```
[ihs]
cnx-web-60.panastoeps.local

[was-dmgr]
cnx-was-60.panastoeps.local

[was-node]
cnx-was-60.panastoeps.local

[tdi]
cnx-db2-60.panastoeps.local

[db2]
cnx-db2-60.panastoeps.local

[pink]
cnx-pink-60.panastoeps.local
```



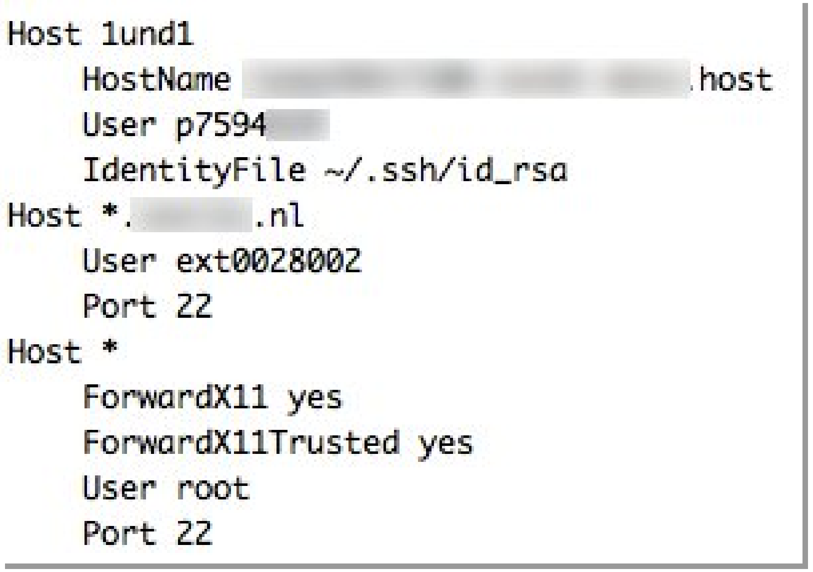
```
[root@theodore cluster]# cat hosts
[master]
10.10.10.215

[worker]
10.10.10.215

[proxy]
10.10.10.215
```

How does it work?



A word on SSH

- SSH Key Authentication saves a lot of time
- SSH Key should be secured with password
- .ssh/config
  - X11Forward
  - Host
  - Used Key
- ssh-agent (ssh-add) saves you from typing passwords over and over again
  - putty pageant http://the.earth.li/~sgtatham/putty/0.58/htmldoc/Chapter9.html
  - openSSH
    - ssh-add .ssh/id_rsa
    - Asks your keyfile password
- Should even work with Orient Me Installer

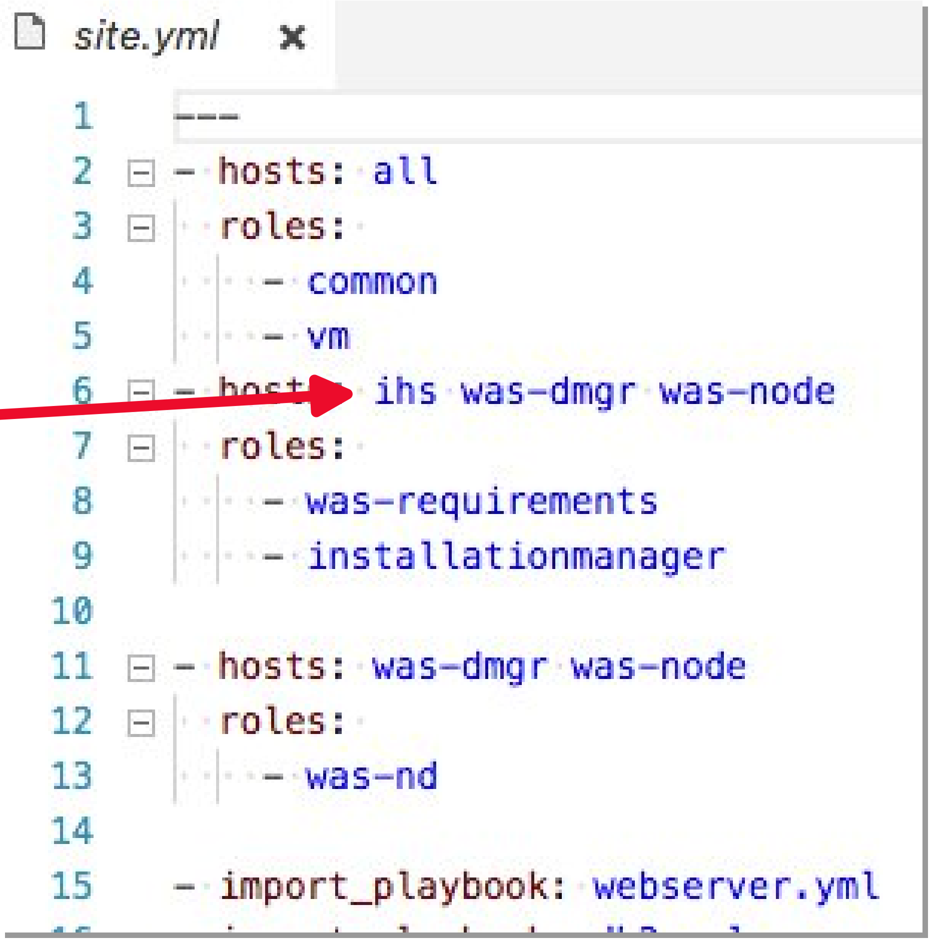


```
Host 1und1
    HostName [redacted].host
    User p7594[redacted]
    IdentityFile ~/.ssh/id_rsa
Host *.[redacted].nl
    User ext0028002
    Port 22
Host *
    ForwardX11 yes
    ForwardX11Trusted yes
    User root
    Port 22
```

How does this help with Connections?

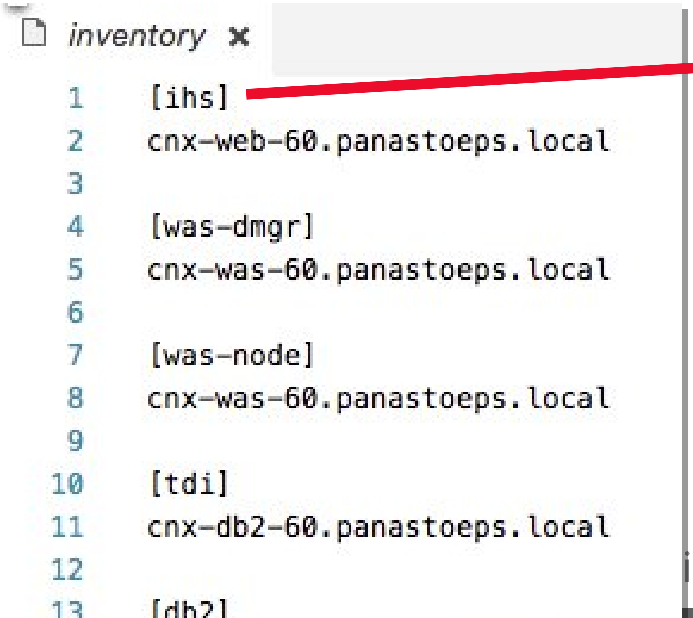
Playbook

- Collection of roles (collection of tasks)
- Playbook contains dependencies of hosts and roles
- You can import Playbooks into others
- Hostnames and Groups from Inventory
- All hosts
  - Role: Common and VM



site.yml

```
---
- hosts: all
  roles:
    - common
    - vm
- hosts: ihs was-dmgr was-node
  roles:
    - was-requirements
    - installationmanager

- hosts: was-dmgr was-node
  roles:
    - was-nd

- import_playbook: webserver.yml
```

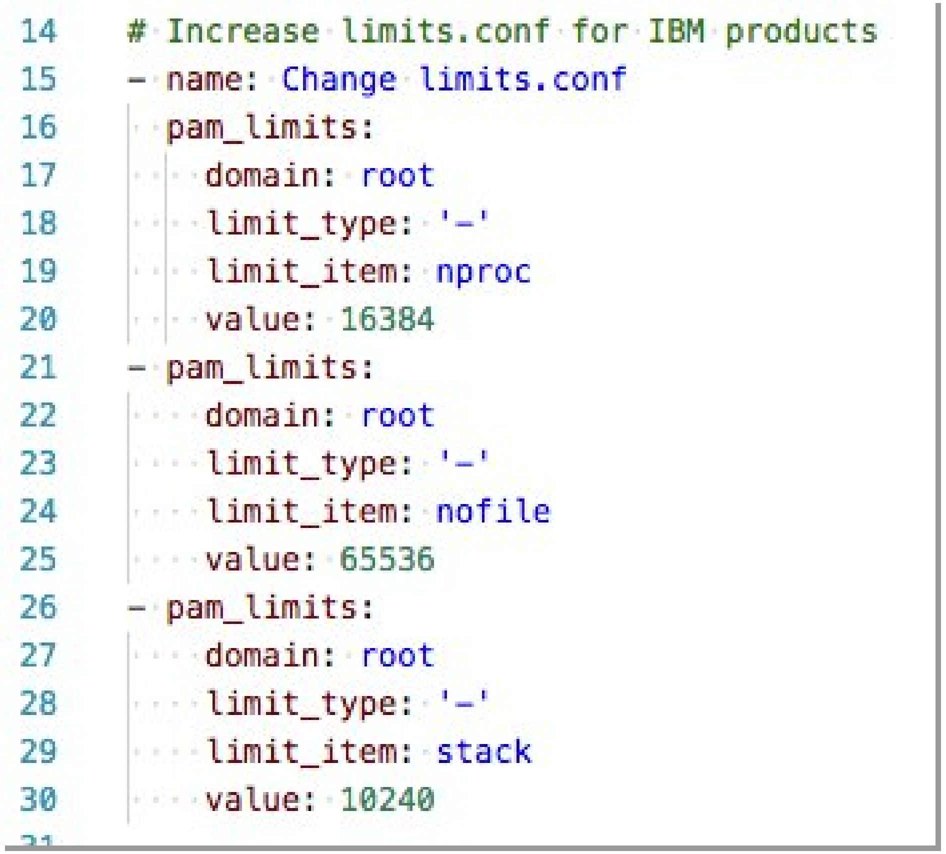


inventory

```
[ihs]
cnx-web-60.panastoeps.local

[was-dmgr]
cnx-was-60.panastoeps.local

[was-node]
cnx-was-60.panastoeps.local

[tdi]
cnx-db2-60.panastoeps.local

[db2]
```

Prepare Operating System - Configuration

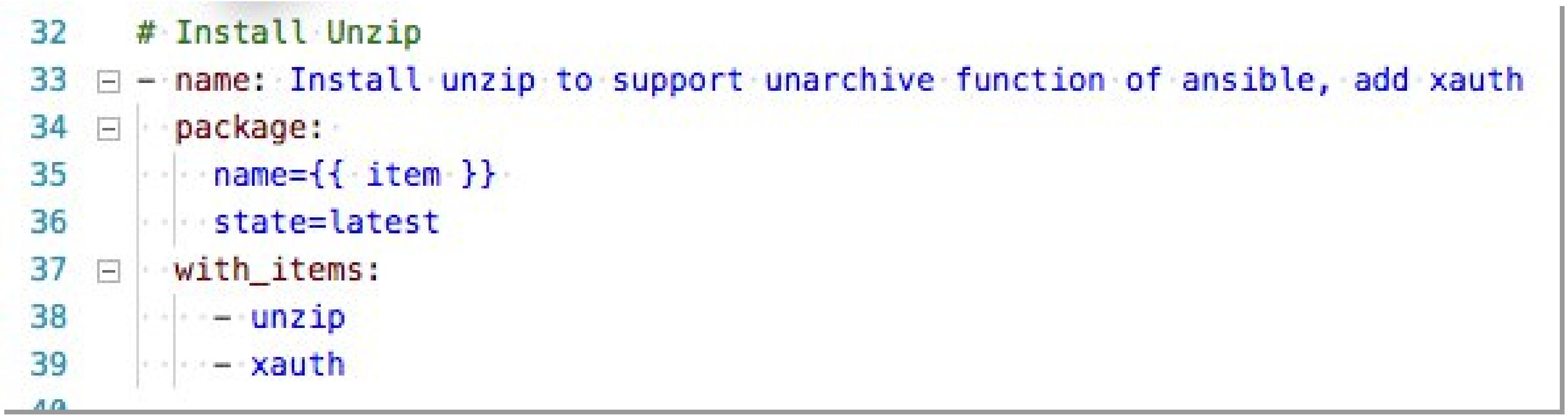
- Configure /etc/security/limits.conf
  - nofile
- Enable X11Forward
- Changing hosts-file during migration
  - Best Practice to install with the real hostname than a temporary one
  - Don’t forget a server ◀◀
    - Whether in /etc/hosts, nor to deploy the file
  - Count your servers
    - 2 WebSphere Nodes IBM Connections
    - Dmgr, IBM Docs …
    - DB2 -> maybe HA
    - Webserver, Proxy, Load Balancer



```
# Increase limits.conf for IBM products
- name: Change limits.conf
  pam_limits:
    domain: root
    limit_type: '-'
    limit_item: nproc
    value: 16384
- pam_limits:
    domain: root
    limit_type: '-'
    limit_item: nofile
    value: 65536
- pam_limits:
    domain: root
    limit_type: '-'
    limit_item: stack
    value: 10240
```

Prepare Operating System – Package Management

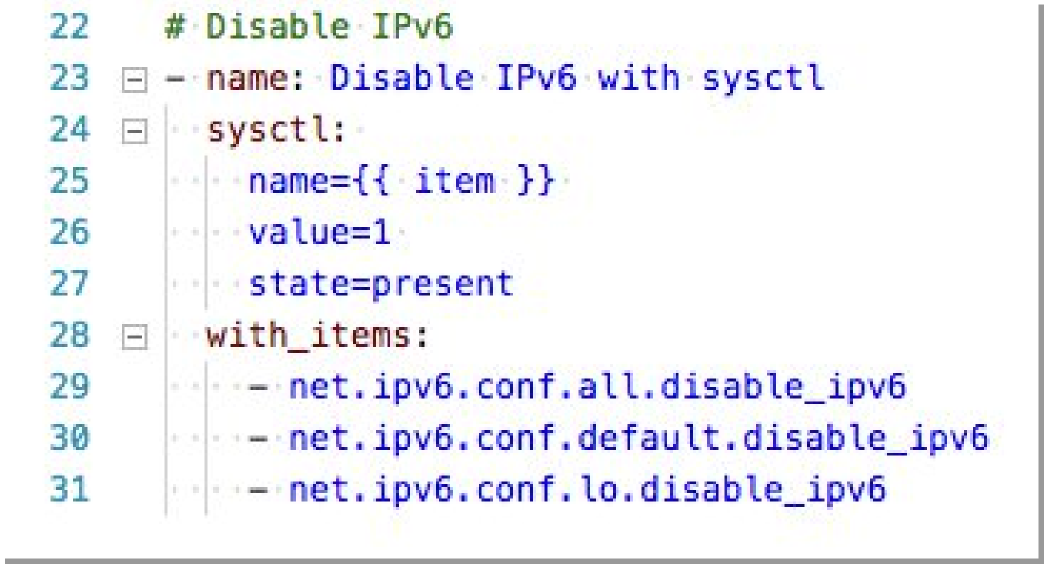
- Install packages for Installation Manager, DB2 or WebSphere Application Server
  - DB2: libaio
  - IM: libXst
- Which Linux do you use – SuSE, Red Hat, Debian?
  - APT
  - Zypper
  - Yum
  - Doesn’t matter!



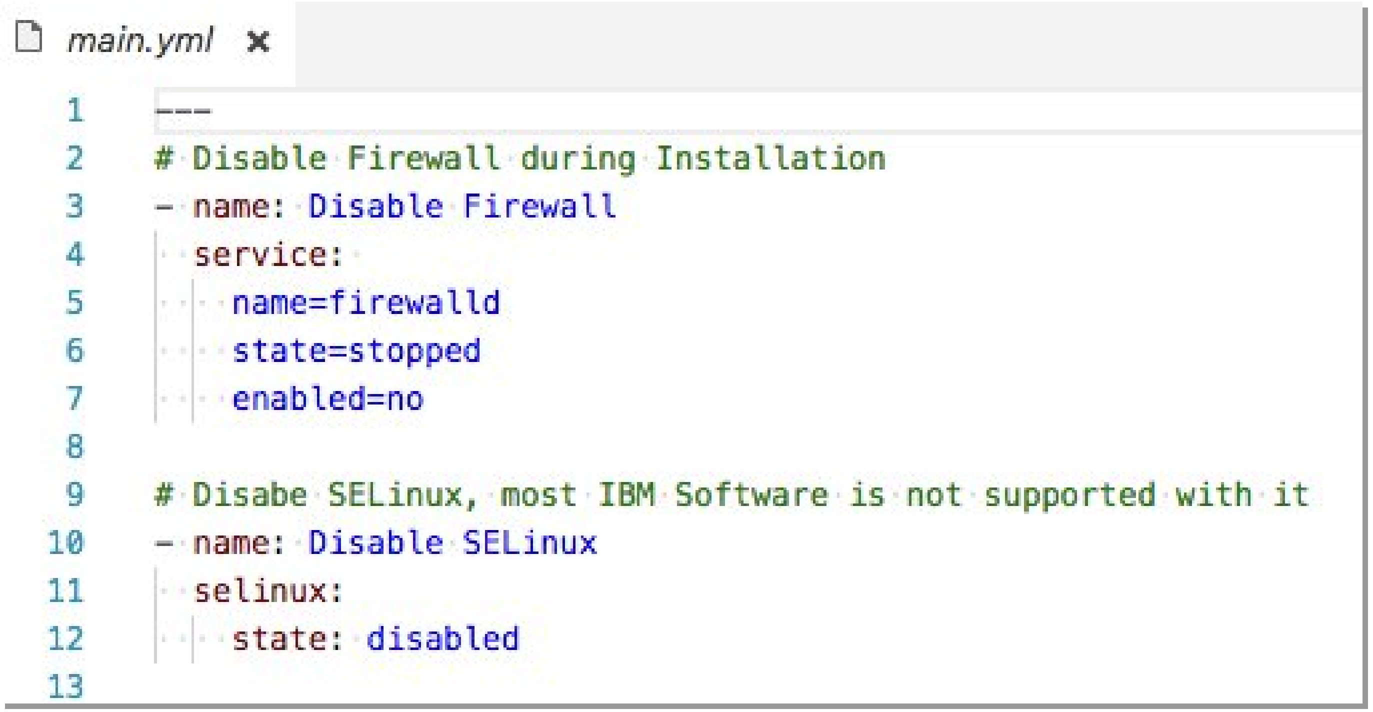
```
# Install Unzip
- name: Install unzip to support unarchive function of ansible, add xauth
  package:
    name={{ item }}
    state=latest
  with_items:
    - unzip
    - xauth
```

Disable Firewall, IPv6 and SELinux

- I always disable Firewalls and Security Extensions during deployments
- Sometimes I forget to do it on one of the servers
- IPv6 often is a pain in software deployments



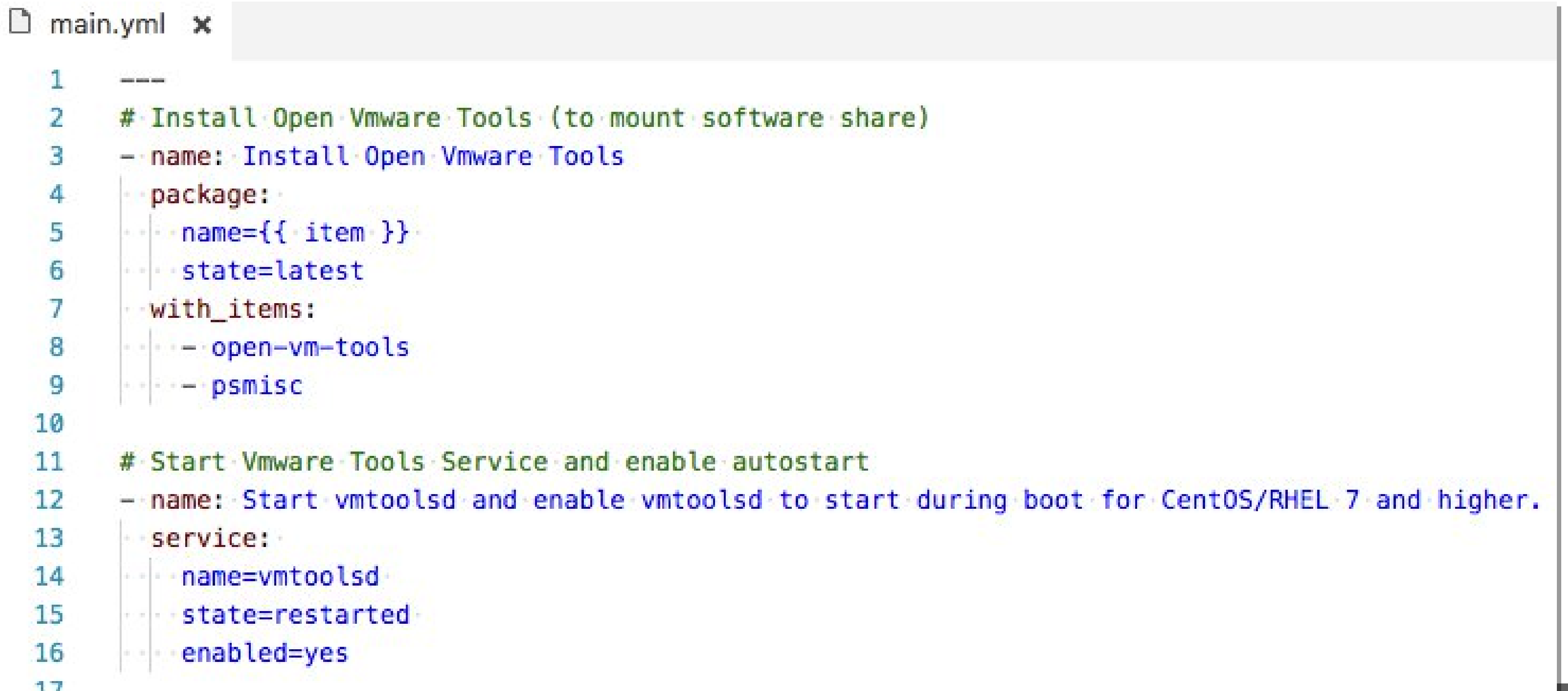
```
# Disable IPv6
- name: Disable IPv6 with sysctl
  sysctl:
    name={{ item }}
    value=1
    state=present
  with_items:
    - net.ipv6.conf.all.disable_ipv6
    - net.ipv6.conf.default.disable_ipv6
    - net.ipv6.conf.lo.disable_ipv6
```



```
---
# Disable Firewall during Installation
- name: Disable Firewall
  service:
    name=firewalld
    state=stopped
    enabled=no

# Disabe SELinux, most IBM Software is not supported with it
- name: Disable SELinux
  selinux:
    state: disabled
```

Install Vmware Tools, Mount Software Share

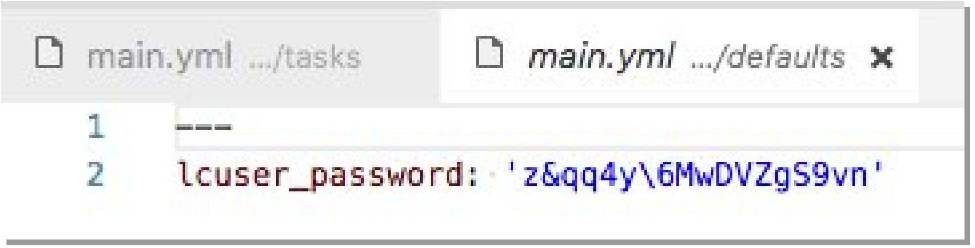
- Install open-vm-tools
- Install psmisc (contains killall -> needed by WebSphere manageprofiles.sh)
- Restart service vmtoolsd



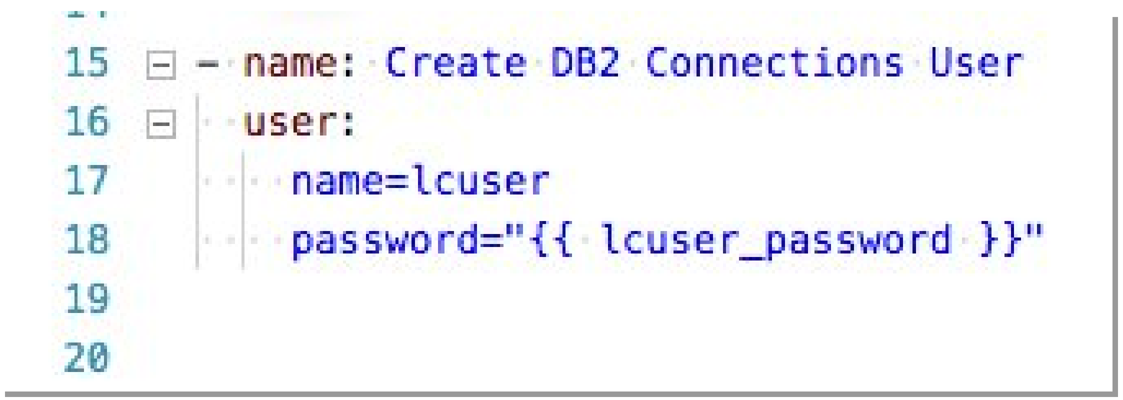
```yaml
---
# Install Open Vmware Tools (to mount software share)
- name: Install Open Vmware Tools
  package:
    name={{ item }}
    state=latest
  with_items:
    - open-vm-tools
    - psmisc

# Start Vmware Tools Service and enable autostart
- name: Start vmtoolsd and enable vmtoolsd to start during boot for CentOS/RHEL 7 and higher.
  service:
    name=vmtoolsd
    state=restarted
    enabled=yes
```

Create User for Connections Databases

- roles/db2-requirements/defaults/main.yml (Variable definitions)
- roles/db2-requirements/tasks/main.yml (tasks)



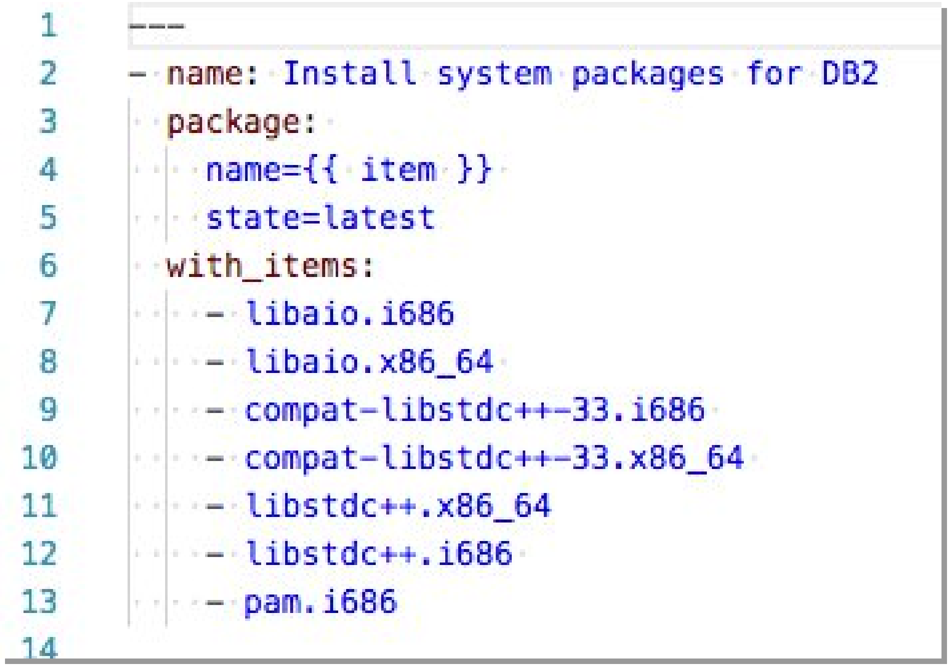
```
---
lcuser_password: 'z&qq4y\6MwDVZgS9vn'
```



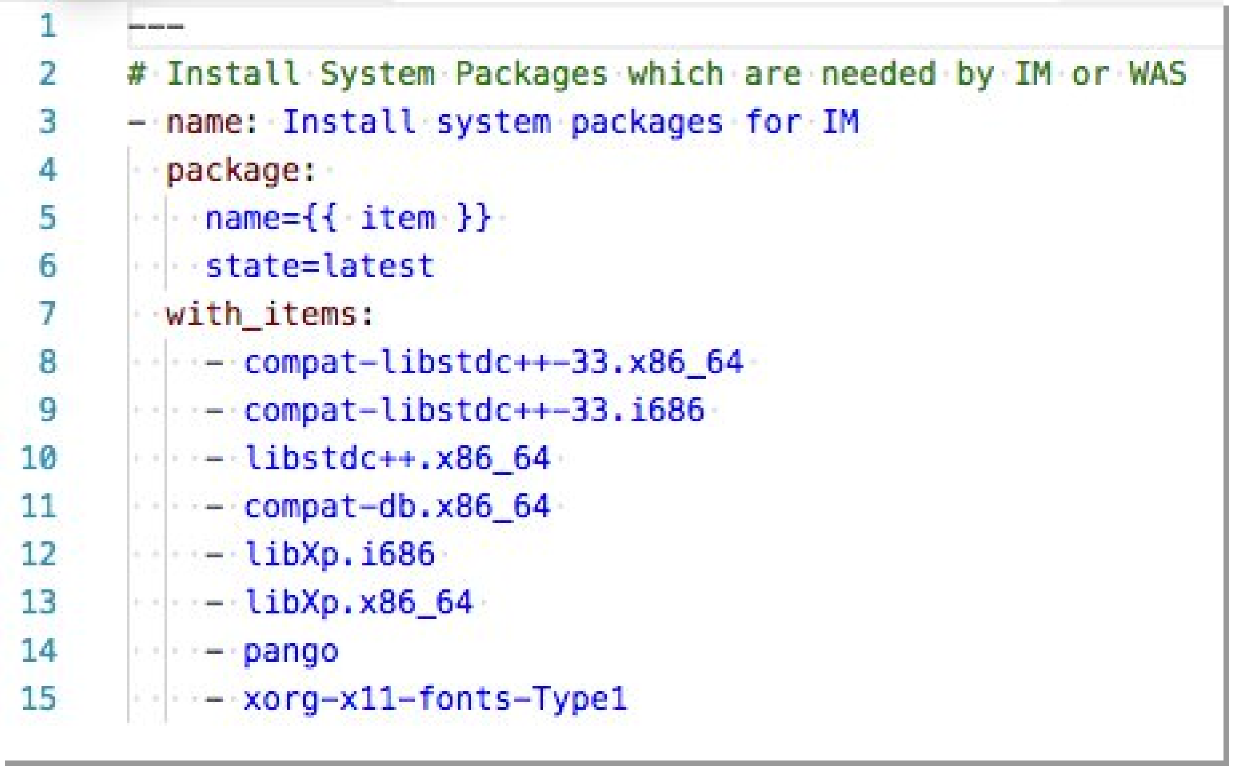
```
- name: Create DB2 Connections User
  user:
    name=lcuser
    password="{{ lcuser_password }}"
```

Install Prerequisit Software

- Don’t forget to add prerequisits to your deployment



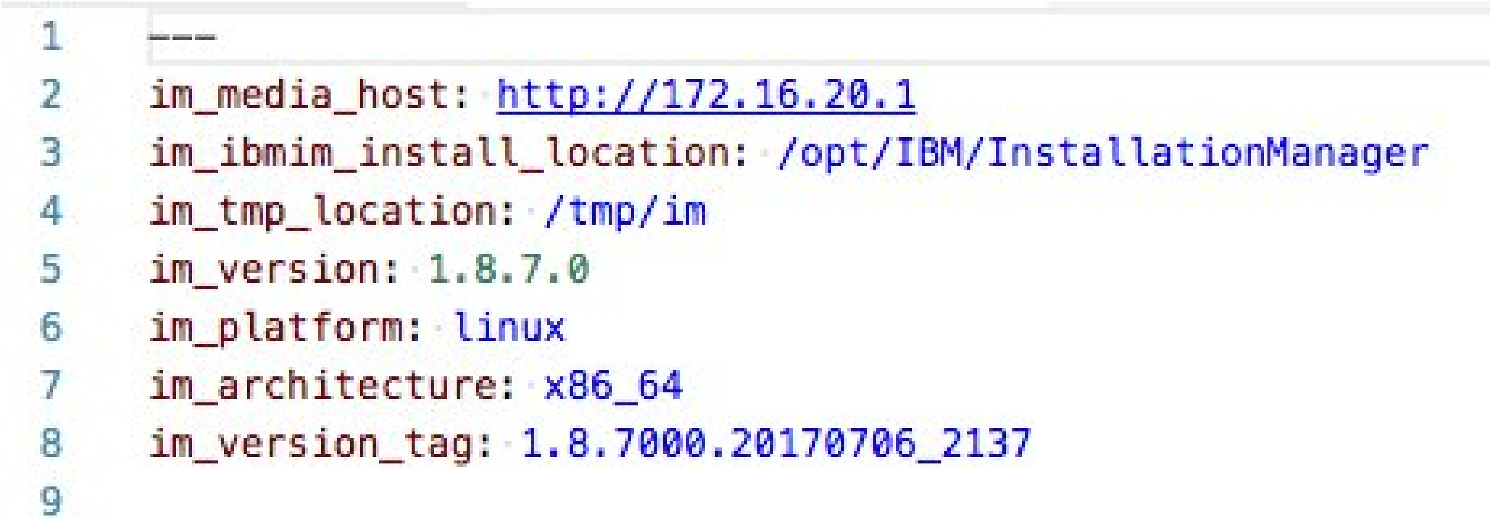
```
---
- name: Install system packages for DB2
  package:
    name={{ item }}
    state=latest
  with_items:
    - libaio.i686
    - libaio.x86_64
    - compat-libstdc++-33.i686
    - compat-libstdc++-33.x86_64
    - libstdc++.x86_64
    - libstdc++.i686
    - pam.i686
```



```
---
# Install System Packages which are needed by IM or WAS
- name: Install system packages for IM
  package:
    name={{ item }}
    state=latest
  with_items:
    - compat-libstdc++-33.x86_64
    - compat-libstdc++-33.i686
    - libstdc++.x86_64
    - compat-db.x86_64
    - libXp.i686
    - libXp.x86_64
    - pango
    - xorg-x11-fonts-Type1
```

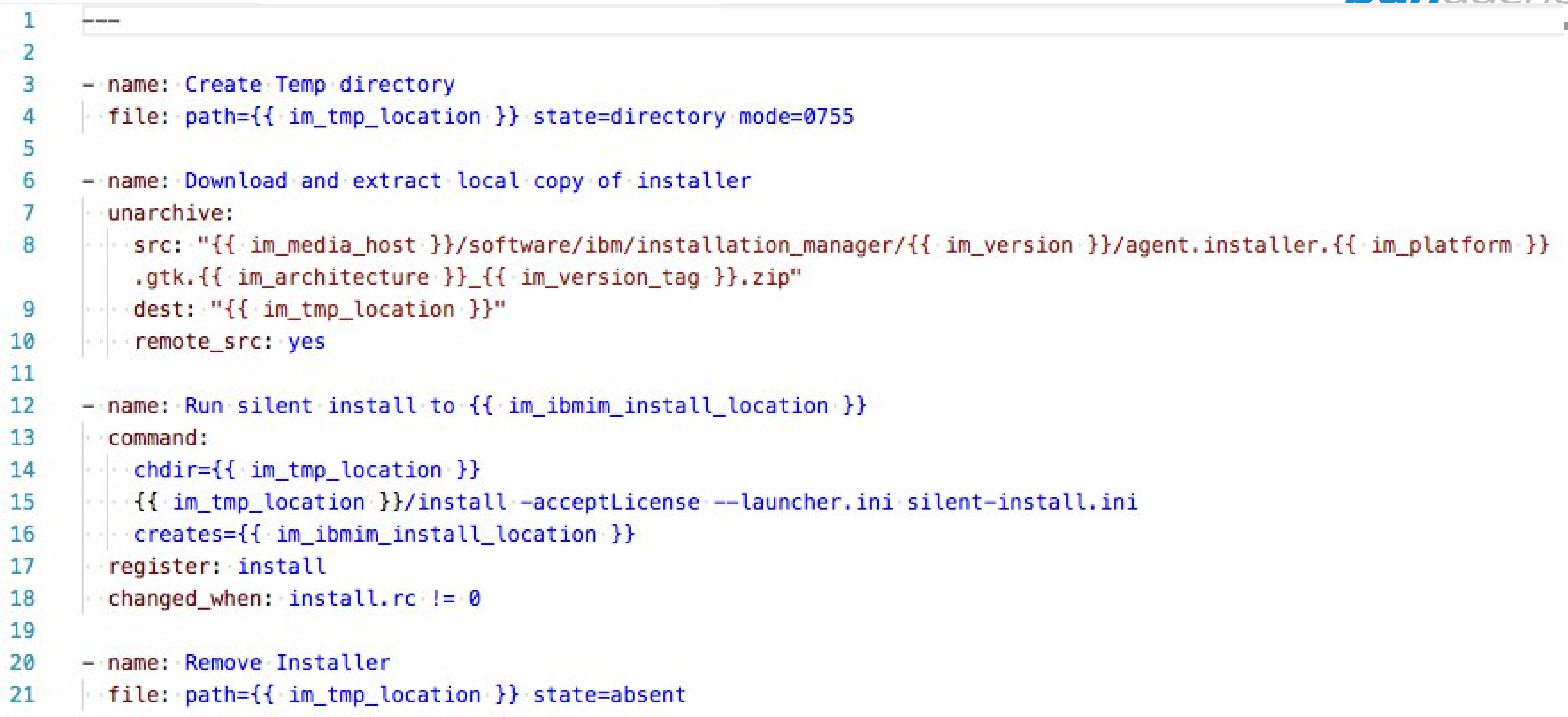
Install Installation Manager

- Role get the installer from a webserver
- Role originally comes from (forked and tweaked a little bit):
- https://github.com/sgwilbur/ansible-ibm-installation-manager
- I use Docker with nginx to serve the file
- Ansible downloads and extracts the package
- Silent Install of Installation Manager
- Delete the extracted content
- Used Variables:



```
---
im_media_host: http://172.16.20.1
im_ibmim_install_location: /opt/IBM/InstallationManager
im_tmp_location: /tmp/im
im_version: 1.8.7.0
im_platform: linux
im_architecture: x86_64
im_version_tag: 1.8.7000.20170706_2137

```

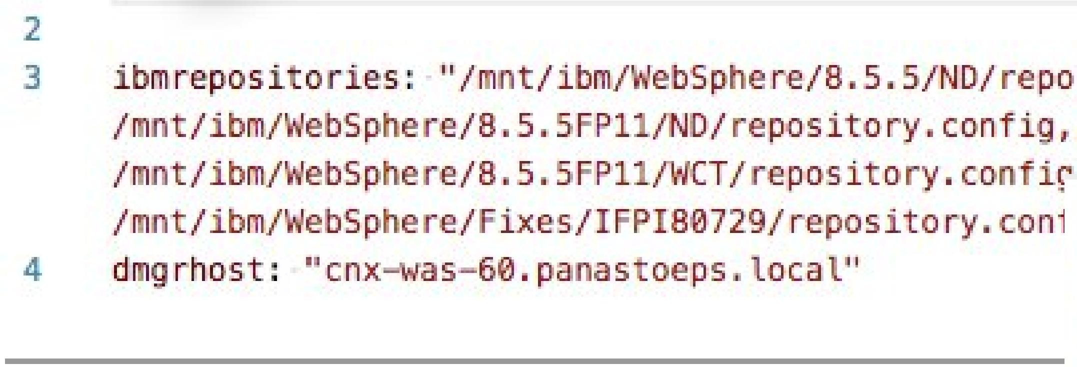
Installation Manager



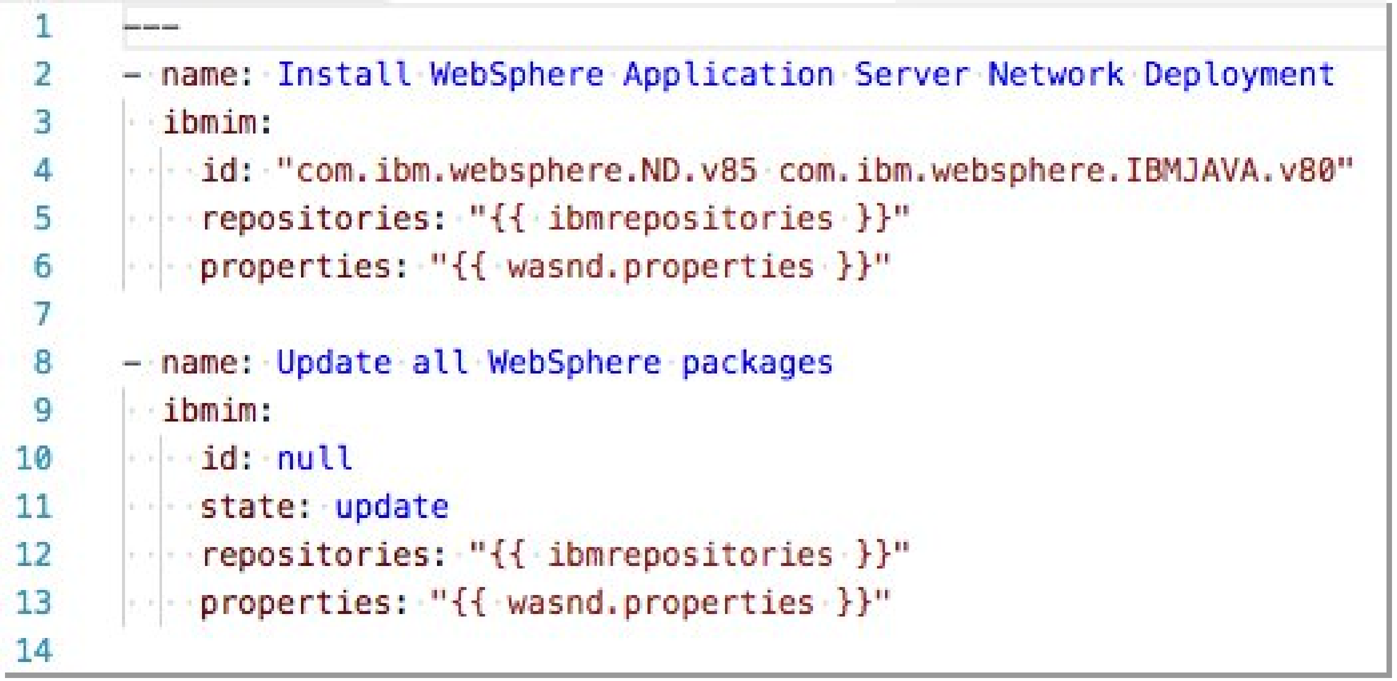
```yaml
---

- name: Create Temp directory
  file: path={{ im_tmp_location }} state=directory mode=0755

- name: Download and extract local copy of installer
  unarchive:
    src: "{{ im_media_host }}/software/ibm/installation_manager/{{ im_version }}/agent.installer.{{ im_platform }}.gtk.{{ im_architecture }}_{{ im_version_tag }}.zip"
    dest: "{{ im_tmp_location }}"
    remote_src: yes

- name: Run silent install to {{ im_ibmim_install_location }}
  command:
    chdir={{ im_tmp_location }}
    {{ im_tmp_location }}/install -acceptLicense --launcher.ini silent-install.ini
    creates={{ im_ibmim_install_location }}
  register: install
  changed_when: install.rc != 0

- name: Remove Installer
  file: path={{ im_tmp_location }} state=absent
```

Install WebSphere Components

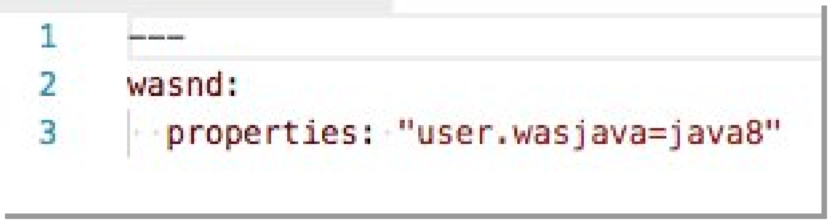
- Modules from https://github.com/amimof/ansible-websphere
- Copy python scripts to ~/.ansible/plugins/modules or put into library within your playbook



```

ibmrepositories: "/mnt/ibm/WebSphere/8.5.5/ND/repository.config,/mnt/ibm/WebSphere/8.5.5/SUPPL/repository.config,
/mnt/ibm/WebSphere/8.5.5FP11/ND/repository.config,/mnt/ibm/WebSphere/8.5.5FP11/SUPPL/repository.config,
/mnt/ibm/WebSphere/8.5.5FP11/WCT/repository.config,/mnt/ibm/WebSphere/8.0.3.0/IBMWASJAVA/repository.config
/mnt/ibm/WebSphere/Fixes/IFPI80729/repository.conf
dmgrhost: "cnx-was-60.panastoeps.local"
```



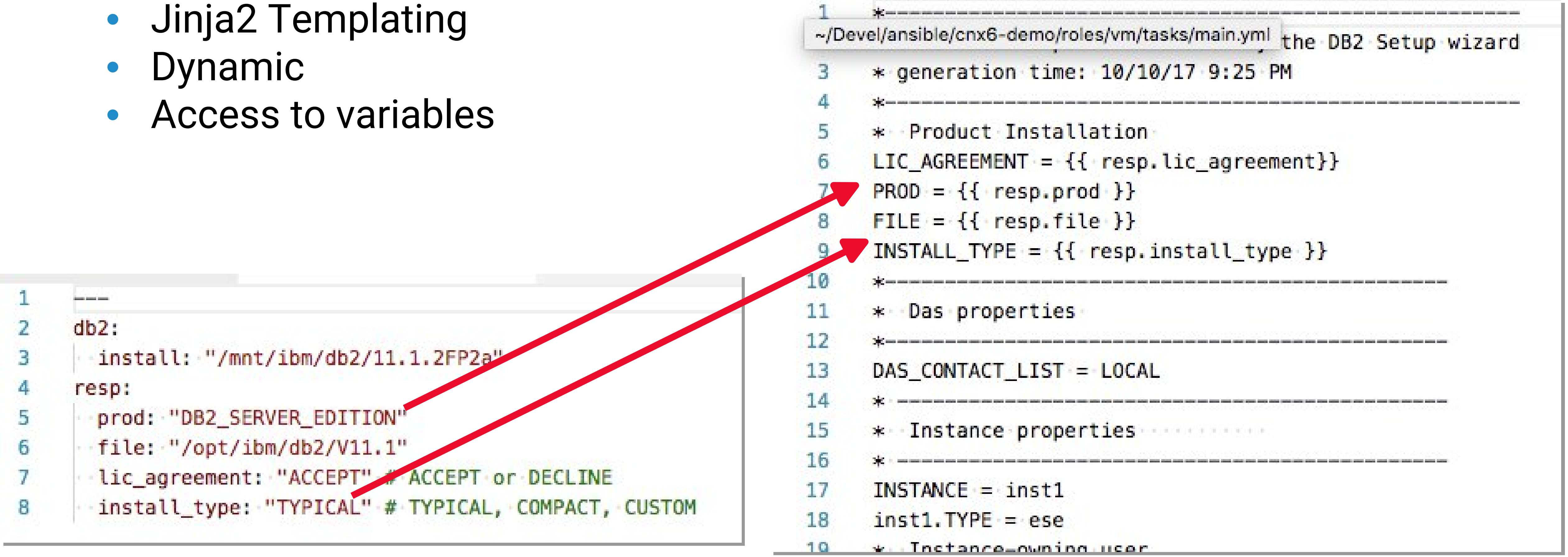
```
---
- name: Install WebSphere Application Server Network Deployment
  ibmim:
    id: "com.ibm.websphere.ND.v85 com.ibm.websphere.IBMJAVA.v80"
    repositories: "{{ ibmrepositories }}"
    properties: "{{ wasnd.properties }}"

- name: Update all WebSphere packages
  ibmim:
    id: null
    state: update
    repositories: "{{ ibmrepositories }}"
    properties: "{{ wasnd.properties }}"

```



```
---
wasnd:
  properties: "user.wasjava=java8"
```

Templates – Jinja2

- You can use response files as templates
- Jinja2 Templating
- Dynamic
- Access to variables

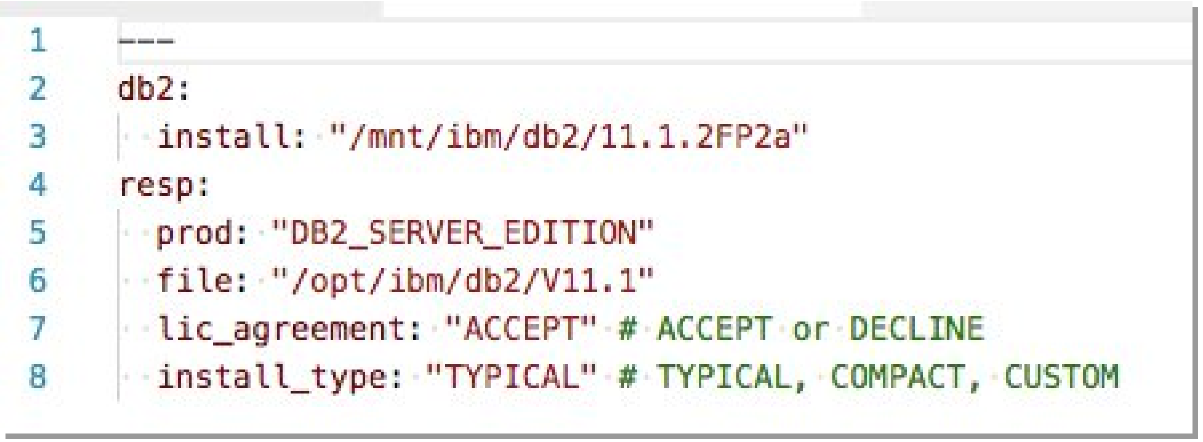


```
---
db2:
  install: "/mnt/ibm/db2/11.1.2FP2a"
resp:
  prod: "DB2_SERVER_EDITION"
  file: "/opt/ibm/db2/V11.1"
  lic_agreement: "ACCEPT" # ACCEPT or DECLINE
  install_type: "TYPICAL" # TYPICAL, COMPACT, CUSTOM
```

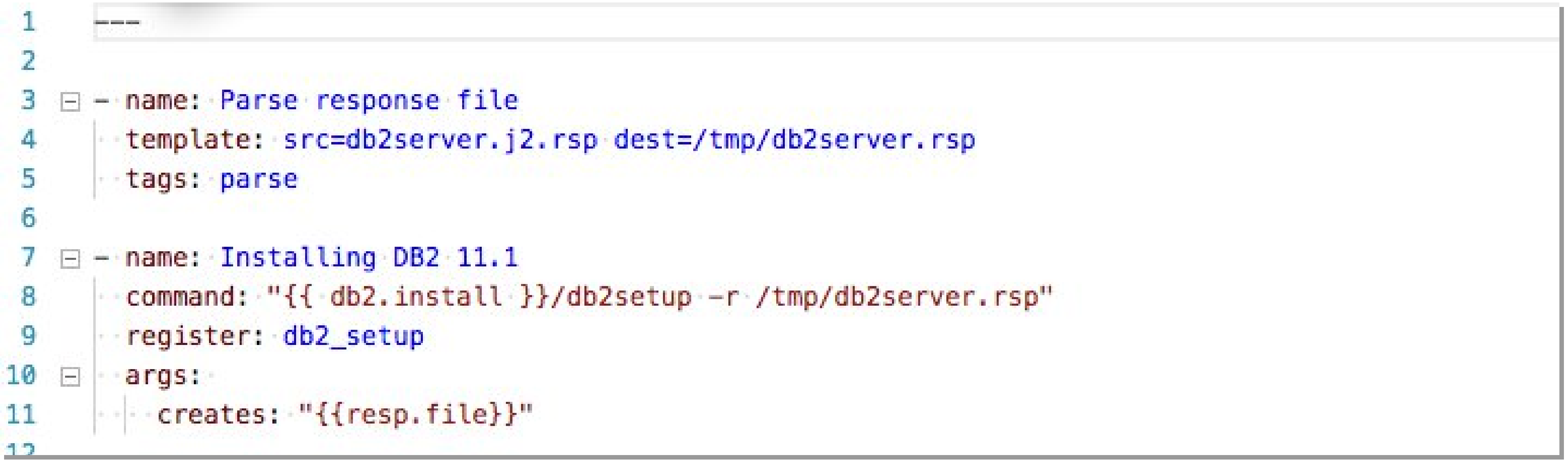
```
*-----------------------------------------------------
~/Devel/ansible/cnx6-demo/roles/vm/tasks/main.yml the DB2 Setup wizard
* generation time: 10/10/17 9:25 PM
*-----------------------------------------------------
*  Product Installation
LIC_AGREEMENT = {{ resp.lic_agreement}}
PROD = {{ resp.prod }}
FILE = {{ resp.file }}
INSTALL_TYPE = {{ resp.install_type }}
*-------------------------------------------------
*  Das properties
*-------------------------------------------------
DAS_CONTACT_LIST = LOCAL
* -----------------------------------------------
*  Instance properties
* -----------------------------------------------
INSTANCE = inst1
inst1.TYPE = ese
*  Instance-owning user
```

Install DB2

- Parsed Templates
- Use as Reponse File



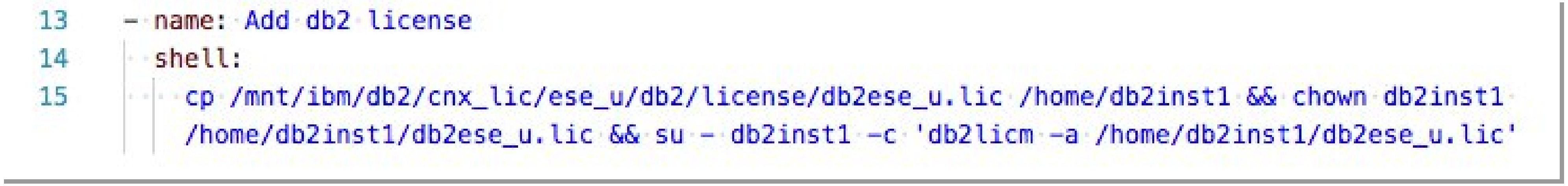
```
---
db2:
  install: "/mnt/ibm/db2/11.1.2FP2a"
resp:
  prod: "DB2_SERVER_EDITION"
  file: "/opt/ibm/db2/V11.1"
  lic_agreement: "ACCEPT" # ACCEPT or DECLINE
  install_type: "TYPICAL" # TYPICAL, COMPACT, CUSTOM
```



```
---

- name: Parse response file
  template: src=db2server.j2.rsp dest=/tmp/db2server.rsp
  tags: parse

- name: Installing DB2 11.1
  command: "{{ db2.install }}/db2setup -r /tmp/db2server.rsp"
  register: db2_setup
  args:
    creates: "{{resp.file}}"
```

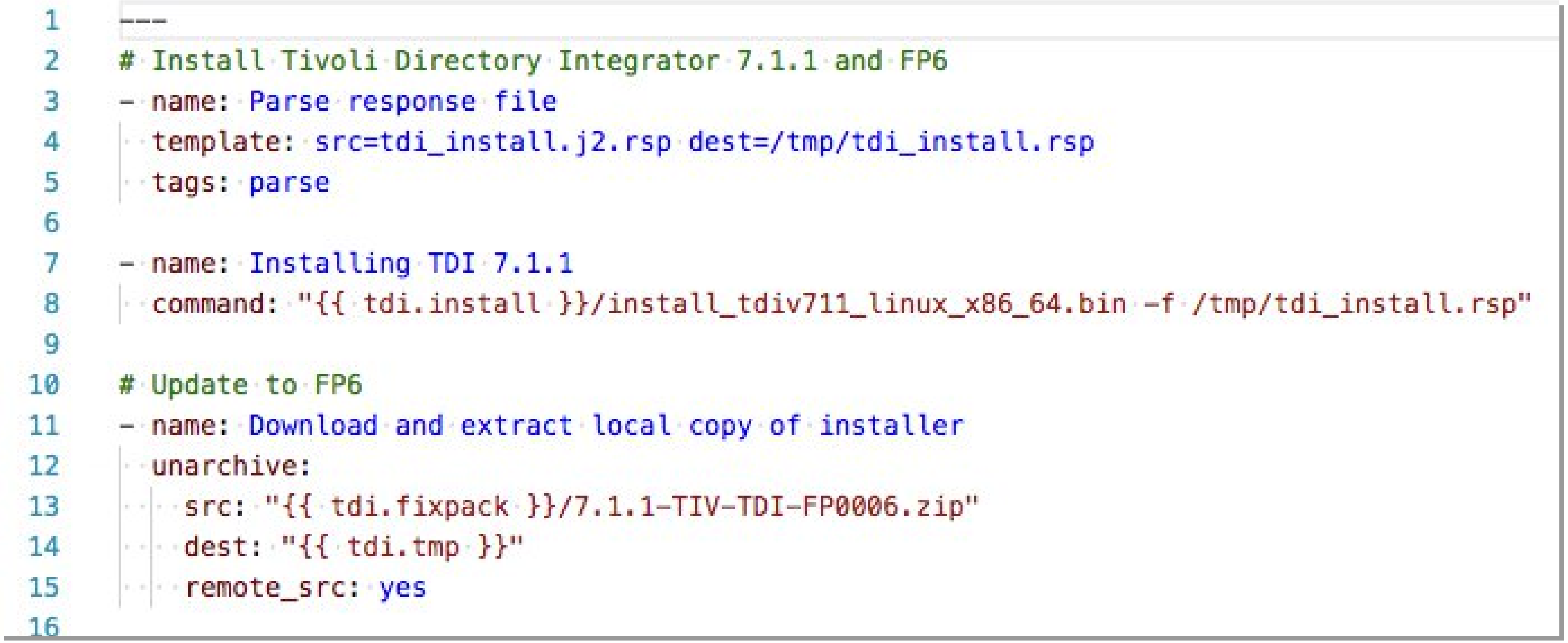
Import DB2 License

- Use a shell command



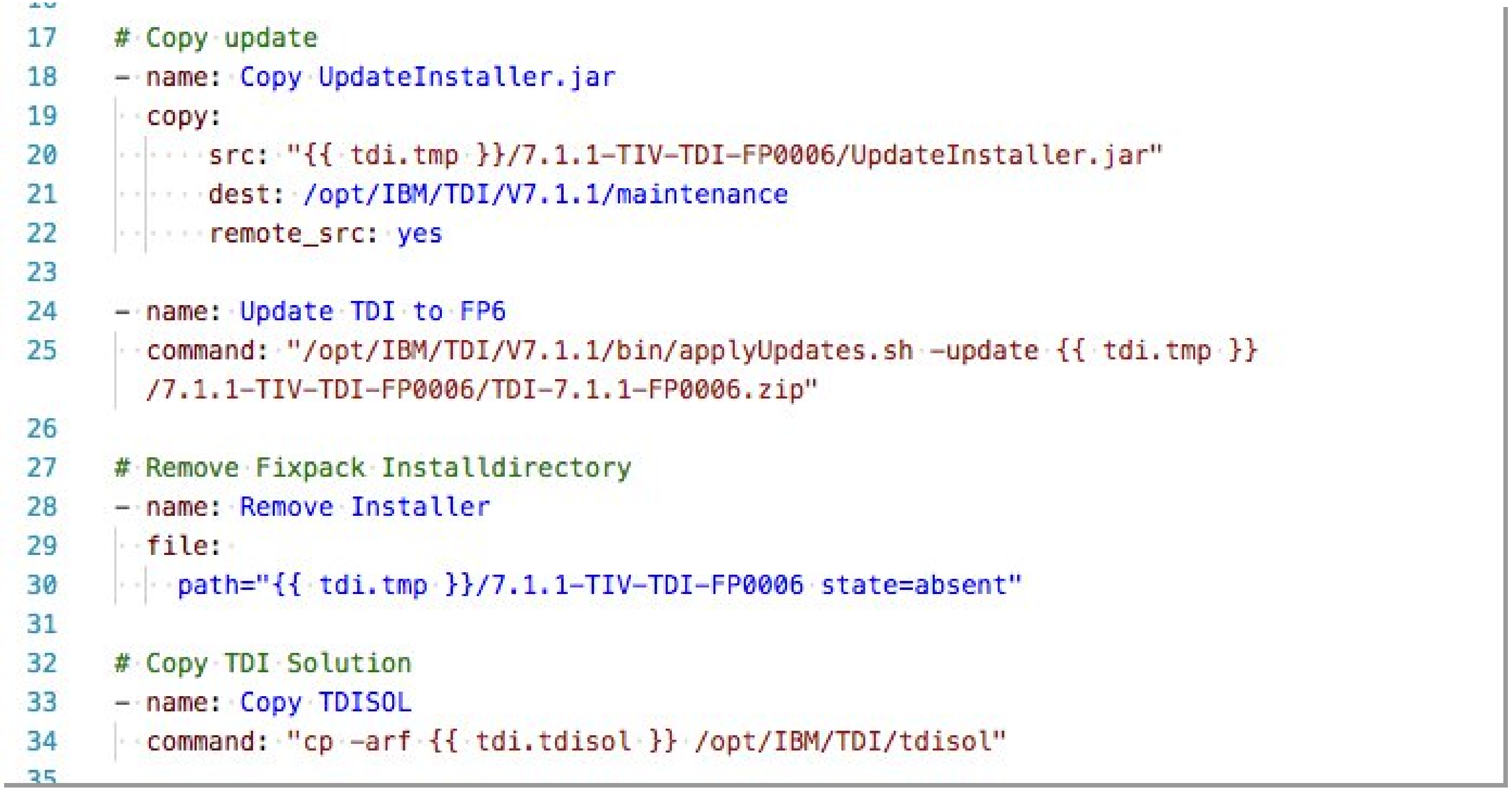
```
- name: Add db2 license
  shell:
    cp /mnt/ibm/db2/cnx_lic/ese_u/db2/license/db2ese_u.lic /home/db2inst1 && chown db2inst1 /home/db2inst1/db2ese_u.lic && su - db2inst1 -c 'db2licm -a /home/db2inst1/db2ese_u.lic'
```

Install and Update TDI



```
---
# Install Tivoli Directory Integrator 7.1.1 and FP6
- name: Parse response file
  template: src=tdi_install.j2.rsp dest=/tmp/tdi_install.rsp
  tags: parse

- name: Installing TDI 7.1.1
  command: "{{ tdi.install }}/install_tdiv711_linux_x86_64.bin -f /tmp/tdi_install.rsp"

# Update to FP6
- name: Download and extract local copy of installer
  unarchive:
    src: "{{ tdi.fixpack }}/7.1.1-TIV-TDI-FP0006.zip"
    dest: "{{ tdi.tmp }}"
    remote_src: yes

```

Install and Update TDI



```
# Copy update
- name: Copy UpdateInstaller.jar
  copy:
      src: "{{ tdi.tmp }}/7.1.1-TIV-TDI-FP0006/UpdateInstaller.jar"
      dest: /opt/IBM/TDI/V7.1.1/maintenance
      remote_src: yes

- name: Update TDI to FP6
  command: "/opt/IBM/TDI/V7.1.1/bin/applyUpdates.sh -update {{ tdi.tmp }}
  /7.1.1-TIV-TDI-FP0006/TDI-7.1.1-FP0006.zip"

# Remove Fixpack Installdirectory
- name: Remove Installer
  file:
    path="{{ tdi.tmp }}/7.1.1-TIV-TDI-FP0006 state=absent"

# Copy TDI Solution
- name: Copy TDISOL
  command: "cp -arf {{ tdi.tdisol }} /opt/IBM/TDI/tdisol"
```

DEMO TIME

Nearly everything is possible

- Put passwords in an encrypted vault
  - http://docs.ansible.com/ansible/latest/playbooks_vault.html
- Manage Docker container
- Reboot your Systems

Works with Windows too

- Ansible can used with Windows (WinRM / Remote Powershell)
  - Gather facts on Windows hosts
  - Install and uninstall MSIs
  - Enable and disable Windows Features
  - Start, stop, and manage Windows services
  - Create and manage local users and groups
  - Manage Windows packages via the Chocolatey package manager
  - Manage and install Windows updates
  - Fetch files from remote sites
  - Push and execute any PowerShell scripts you write

Administrators / Developers

- Have a look at Ansible
  - Saves you time
  - Easy to deploy and use in different environments
    - QA
    - Testing
    - Production
- KISS
  - Keep it simple stupid

Questions?

@stoeps
linkedin.com/in/christophstoettner
www.stoeps.de

christophstoettner

+49 173 8588719

christoph.stoettner@panagenda.com

Presentation download:

https://slideshare.net/christophstoettner

Headquarters, Austria:
panagenda GmbH (Ltd.)
Schreyvogelgasse 3/10
AT 1010 Vienna

Phone: +43 1 89 012 89
Fax: +43 1 89 012 89-15
E-Mail: info@panagenda.com

Headquarters, Germany:
panagenda GmbH (Ltd.)
Lahnstraße 17
DE 64646 Heppenheim

Phone: +49 6252 67 939-00
Fax: +49 6252 67 939-16
E-Mail: info@panagenda.com

USA:
panagenda Inc.
60 State Street, Suite 700
MA 02109 Boston

Phone: +1 617 855 5961
Fax: +1 617 488 2292
E-Mail: info@panagenda.com

Germany:
panagenda Consulting GmbH (Ltd.)
Donnersbergstraße 1
DE 64646 Heppenheim

Phone: +49 6252 67 939-86
Fax: +49 6252 67 939-16
E-Mail: info@panagenda.com

The Netherlands:
Trust Factory B.V.
11th Floor,
Koningin Julianaplein 10
NL 2595 AA The Hague

Phone: +31 70 80 801 96
E-Mail: info@trust-factory.com

Thank You

Make Your Data Work for You

PLATINUM SPONSORS

IBM®

panagenda

GOLD SPONSORS

Kudos

BOARDS
BADGES
BUZZY
ANALYTICS
IDEAS

INFOWARE

Appfusions™

Reverso

SILVER SPONSORS

OnTime®
IF TIME MATTERS...

T-Matic computerplus

holistic-net

TIMETOACT GROUP
SOFTWARE & CONSULTING

riva
Riva CRM Integration

hedersoft
We understand Data!

Belsoft
collaboration

BCC

workplaceON

groupwave

opentopic

BRONZE SPONSORS

RAMSIT

Whitesoft