|  |
| --- |
| BC Pension Corporation |
| Ansible Tower Configuration |
| Installation and Configuration |

|  |
| --- |
| Charchuk, Mike PENC:EX  8-30-2019 |

Contents

[Ansible Tower Install 2](#_Toc18062768)

[Ansible Tower Firewall Requests 3](#_Toc18062769)

[Ansible Tower SSL Configuration 3](#_Toc18062770)

[Ansible Tower Linux Configuration 4](#_Toc18062771)

[Ansible Tower Windows configuration 4](#_Toc18062772)

[Ansible Tower LDAP Configuration 6](#_Toc18062773)

[Postgresql Configuration - Required 9](#_Toc18062774)

[Postgres configuration for Point in Time Recovery – Not Required 9](#_Toc18062775)

# Ansible Tower Install

1. The Bundled Tower Installation will be used – See <https://docs.ansible.com/ansible-tower/latest/html/quickinstall/download_tower.html#bundled-install>.
2. Pull the latest bundle release from <https://docs.ansible.com/ansible-tower/latest/html/quickinstall/download_tower.html#bundled-install/>
3. Unzip the bundle into /pencmw\_common/ansible\_tower.
4. cd /pencmw\_common/ansible\_tower/ansible-tower-setup-bundle-#####
5. The Red Hat ‘extras’ repository needs to be enabled – at least for CENTOS install.
6. root@localhost:~$ yum install http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm.
7. Tower Installation Scenario is ‘An Integrated Installation’.
8. Edit the inventory file and change the following (the admin password can be changed relatively easily – the pg\_password and rabbitmq\_password cannot):
   1. admin\_password=’initial\_password’
   2. pg\_password=’initial\_password’
   3. rabbitmq\_password=’initial\_password’

Sample Inventory File

[tower]

localhost ansible\_connection**=**local

[database]

[all:vars]

admin\_password**=**'password'

pg\_host**=**''

pg\_port**=**''

pg\_database**=**'awx'

pg\_username**=**'awx'

pg\_password**=**'password'

rabbitmq\_port**=**5672

rabbitmq\_vhost**=**tower

rabbitmq\_username**=**tower

rabbitmq\_password**=**'password'

rabbitmq\_cookie**=**rabbitmqcookie

*# Needs to be true for fqdns and ip addresses*

rabbitmq\_use\_long\_name**=**false

*# Needs to remain false if you are using localhost*

1. As root execute ./setup.sh – this will install ansible tower.

# Ansible Tower Firewall Requests

The following ports need to be opened up for Ansible Tower and Jenkins

1. 80 http for Ansible Tower
2. 4352 postgresql listener for Ansible Tower
3. 443 https for Ansible Tower – currently 4443 because of a typo in my firewall request (mgc)
4. 8080 http for Jenkins
5. 8443 https for Jenkins

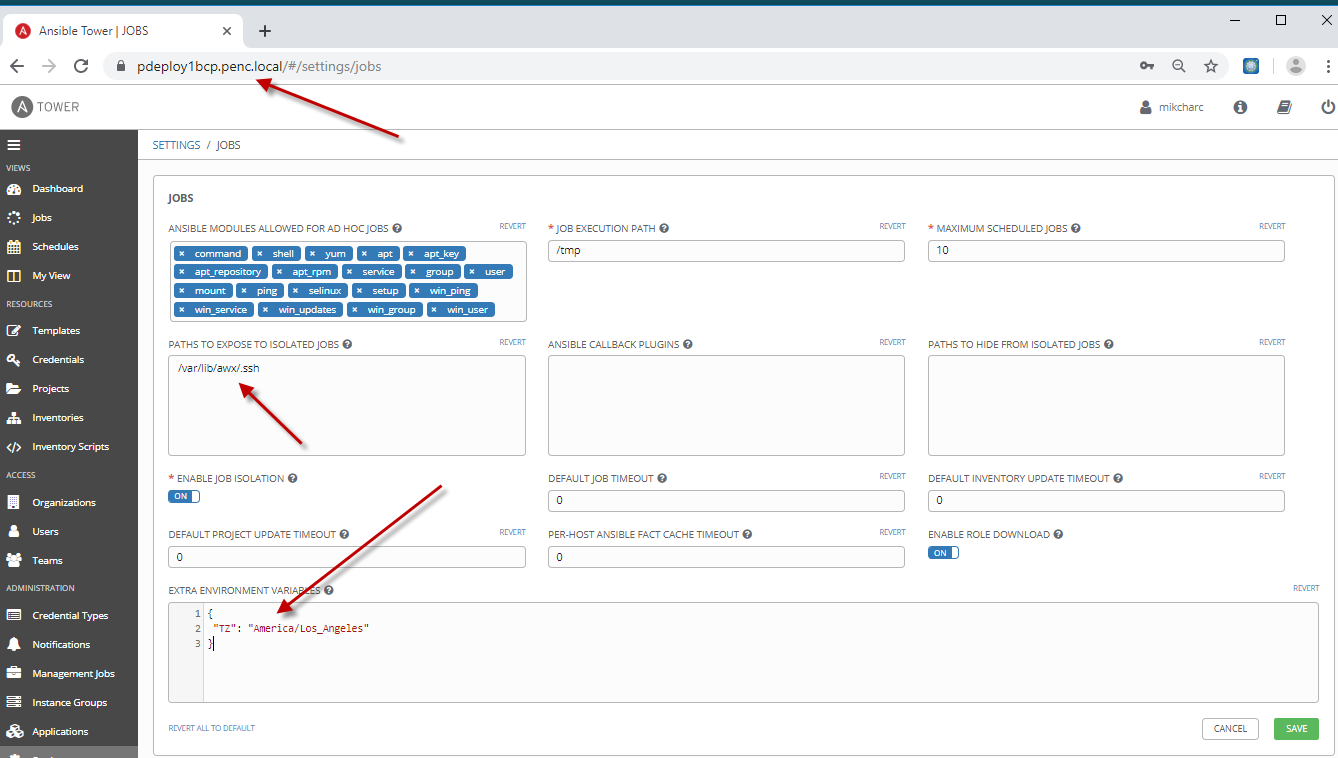
# Ansible Tower SSL Configuration

The ownership of the /etc/tower directory and included files should be changed to awx:awx by the Linux SysAdmins.

1. sudo su – awx
2. cp $Stage\_Dir/Wildcard.cer /etc/tower
3. cp $Stage\_Dir/wildcard.penc.local.aes256.key /etc/tower
4. cd /etc/tower
5. openssl rsa -in wildcard.penc.local.aes256.key -out wild.key
6. Use the PEM Pass Phrase that was used to create the wildcard cert (in the Vault Portal\_AS\_pwds.xls file)
7. cp tower.key tower.key.DDMonYYYY
8. cp tower.cert tower.cert.DDMonYYYY
9. cp wild.key tower.key
10. cp Wildcard.cer tower.cert
11. exit
12. sudo systemctl stop nginx
13. sudo systemctl start nginx

# Ansible Tower Linux Configuration

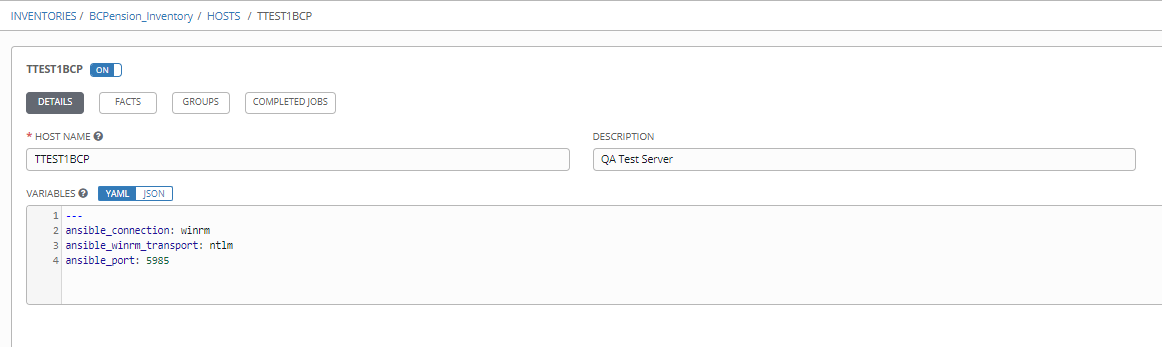
1. Create ssh keys for user awx
   1. sudo su – awx
   2. ssh-key**gen** -b 4096 -o -t rsa
2. In <https://pdeploy1bcp.penc.local/#/configuration/jobs>. 'Enable Job Isolation' is ON by default. In order for jobs to connect via ssh keys the awx .ssh location needs to be exposed to the jobs. Set 'Paths to Expose to Isolated Jobs' to /var/lib/awx/.ssh.



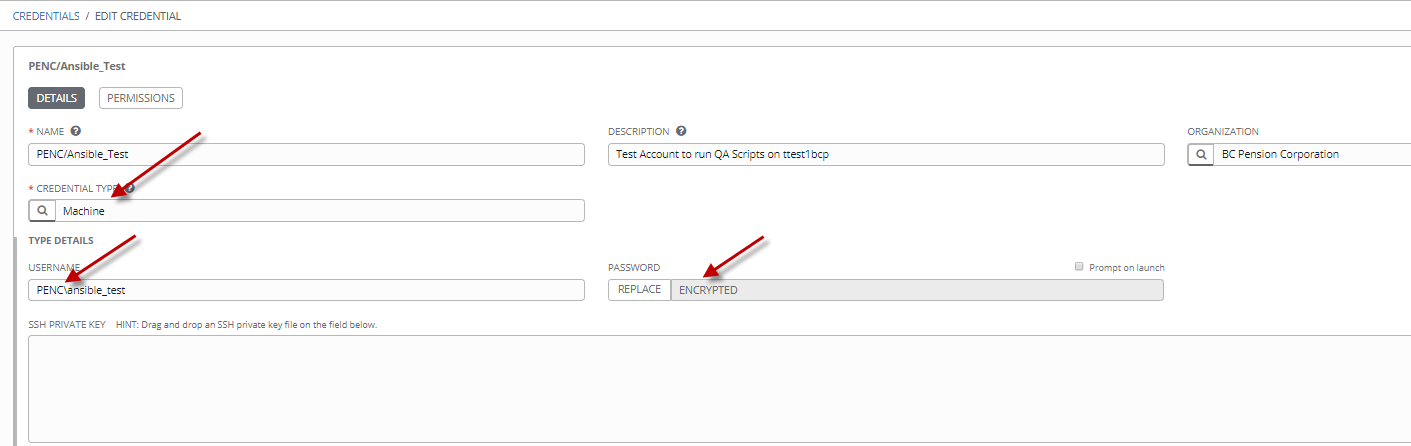
1. The awx ssh public key can then be copied into any Linux users .ssh/authorized\_keys file to allow ansible tower to access the host and execute playbooks.
2. Set the Extra Environment Variables to “TZ”: “America/Los\_Angeles”, this will display all date/time entries with the appropriate Timezone.

# Ansible Tower Windows configuration

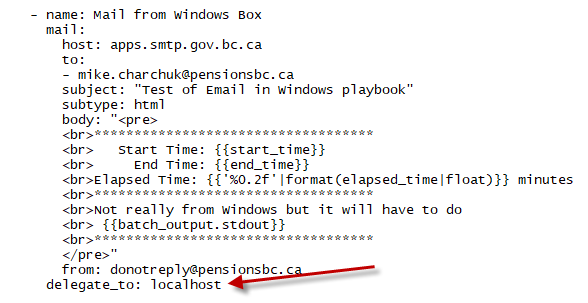
1. Firewall Rule – WINRM is used to execute Ansible Playbooks on Windows servers/workstations so Port http/5985 and https/5986 need to be opened from the Ansible Tower Server to the Windows servers/workstations. The current Pensions configuration only uses port 5985, as winrm is not configured to listen over https.
2. Inventory Configuration – When the windows server is entered into the Ansible Tower Inventory three variables must be set: ansible\_connection, ansible\_winrm\_transport and ansible\_port.



1. Credential configuration – The machine credential needs to have the Domain\username format and a password must be defined.

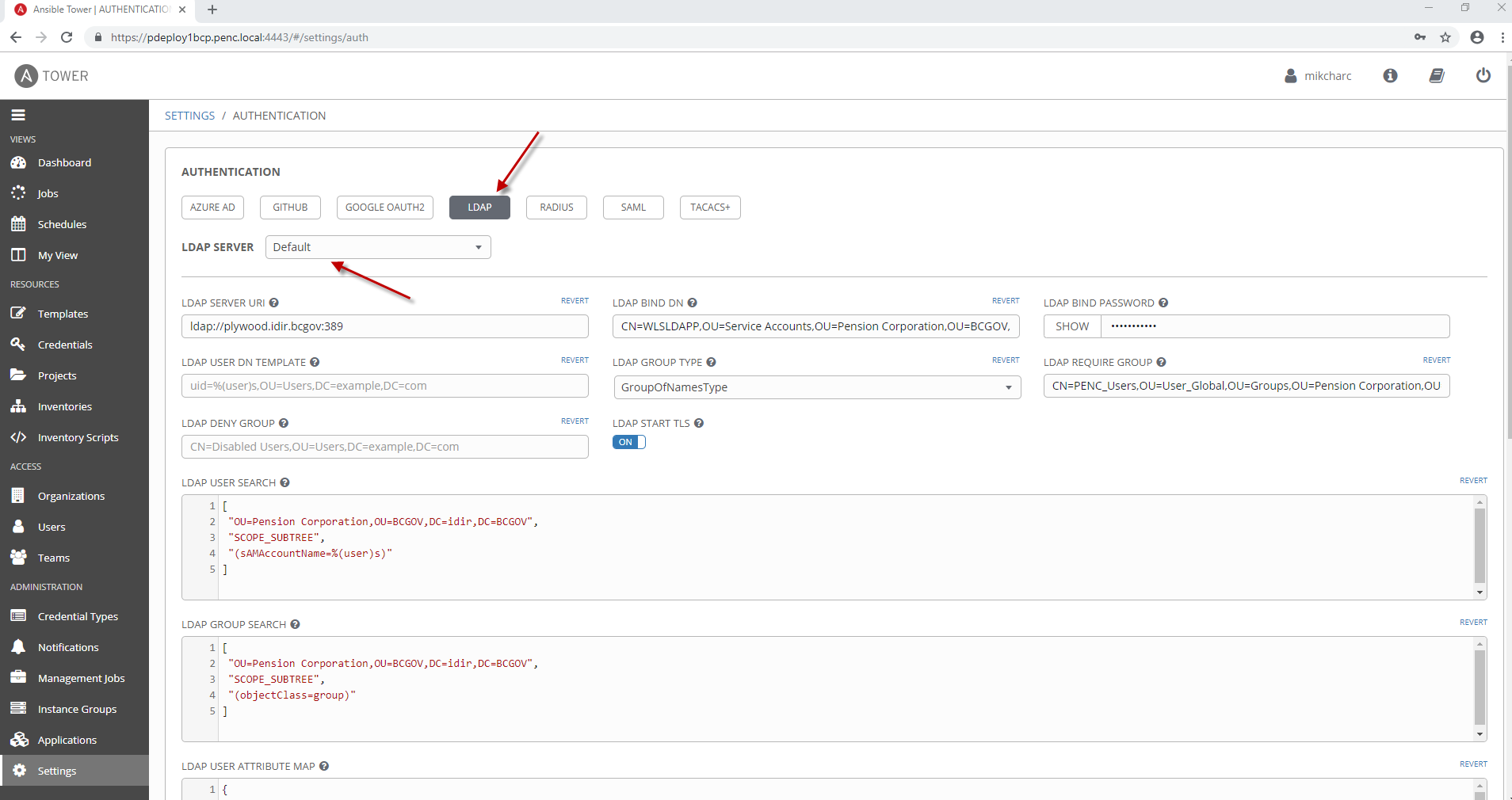


1. Playbook configuration – Windows can only use Ansible modules prefixed with win\_ (see the full full windows module index [here](https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html). As such any playbooks that send email notifications have to delegate the mail task to the localhost (i.e. Ansible Tower server).



# Ansible Tower LDAP Configuration

1. Got to - https://pdeploy1bcp.penc.local:4443/#/settings/auth ‘LDAP’
2. LDAP Server – ‘Default’



1. LDAP SERVER URI - ldap://plywood.idir.bcgov:389
2. LDAP BIND DN - CN=WLSLDAPP,OU=Service Accounts,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV
3. LDAP GROUP TYPE – GroupOfNamesType
4. LDAP REQUIRE GROUP - CN=PENC\_Users,OU=User\_Global,OU=Groups,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV
5. LDAP USER SEARCH –

[

"OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV",

"SCOPE\_SUBTREE",

"(sAMAccountName=%(user)s)"

]

1. LDAP GROUP SEARCH –

[

"OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV",

"SCOPE\_SUBTREE",

"(objectClass=group)"

]

1. LDAP USER ATTRIBUTE MAP –

{

"first\_name": "givenName",

"last\_name": "sn",

"email": "mail"

}

1. LDAP USER FLAGS BY GROUP –

{

"is\_superuser": [

"CN=PENC ITS TSS Midtier,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV"

],

"is\_system\_auditor": [

"CN=PENC ITS ADM Release Management,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV"

]

}

1. LDAP ORGANIZATION MAP –

{

"Pensions Corporation": {

"admins": "CN=PENC ITS TSS Midtier,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV",

"remove\_admins": true,

"users": [

"CN=PENC ITS ADM Release Management,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV",

"CN=PENC ITS TSS Midtier,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV"

]

}

}

1. LDAP TEAM MAP –

{

"DeploymentOps": {

"organization": "Pensions Corporation",

"users": [

"CN=PENC ITS ADM Release Management,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV",

"CN=PENC ITS TSS Midtier,OU=Distribution Lists,OU=Pension Corporation,OU=BCGOV,DC=idir,DC=BCGOV"

],

"remove": true

}

}

1. Save Configuration
2. Open <https://pdeploy1bcp.penc.local:4443/api/v2/settings/ldap/>
   1. Login as a System Administrator (admin)
      1. In the Content Area at the bottom of the page change:

"AUTH\_LDAP\_CONNECTION\_OPTIONS": {

"OPT\_NETWORK\_TIMEOUT": 30,

"OPT\_REFERRALS": 0

},

To:

"AUTH\_LDAP\_CONNECTION\_OPTIONS": {

"OPT\_X\_TLS\_REQUIRE\_CERT": 0,

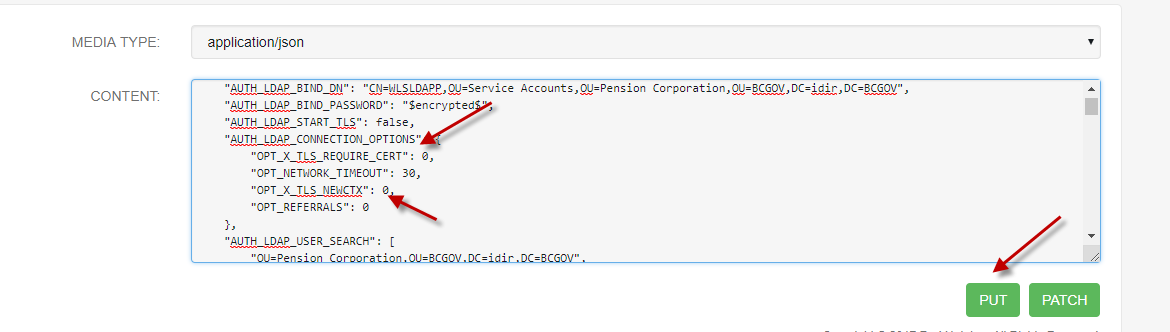
"OPT\_NETWORK\_TIMEOUT": 30,

"OPT\_X\_TLS\_NEWCTX": 0,

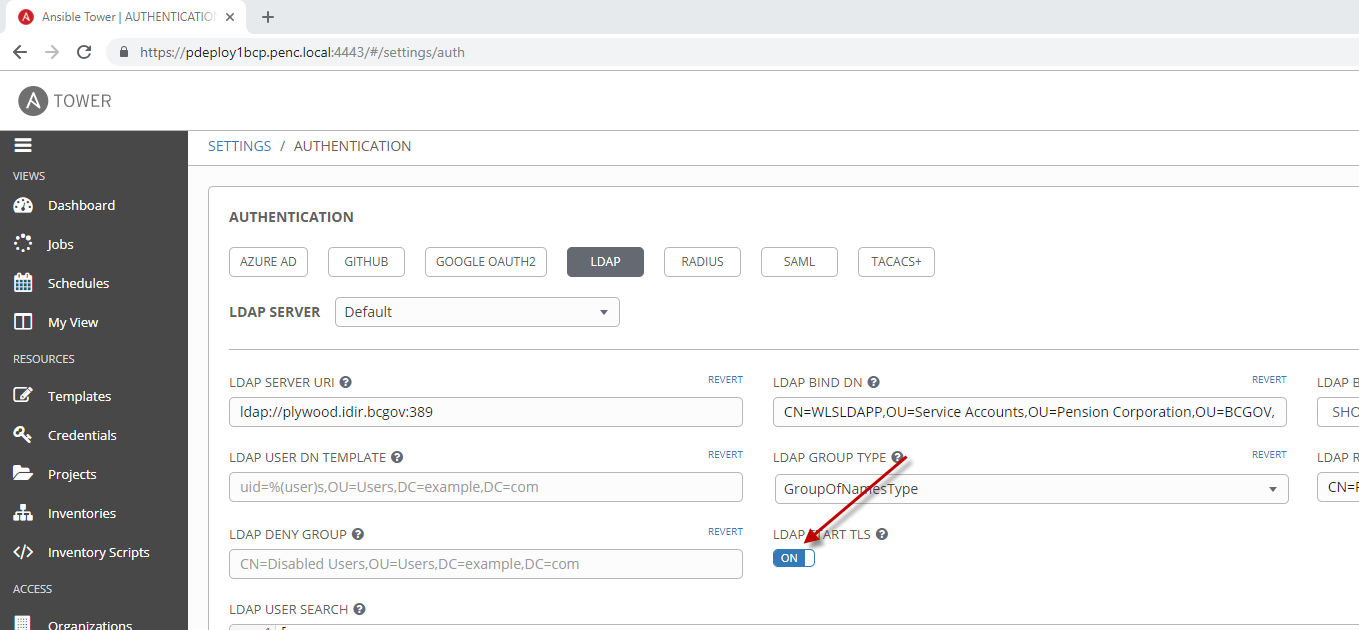
"OPT\_REFERRALS": 0

},

* + 1. And then press PUT



* + 1. Go to <https://pdeploy1bcp.penc.local:4443/#/settings/auth> and click LDAP - Set LDAP START TLS to ON and press Save



# Postgresql Configuration - Required

For future reporting access to the postgresql database will probably need to carried out remotely.

As the postgres user edit the pg\_hba.conf - add the following line - it will allow connection to the database remotely as long as a password is provided.

host all all 0.0.0.0/0 password

1. sudo su – postgres
2. cd /var/lib/pgsql/9.6/data
3. vi pg\_hba.conf
4. shift G
5. shift A
6. host all all 0.0.0.0/0 password
7. ‘ESC’
8. shift ZZ

# Postgres configuration for Point in Time Recovery – Not Required

1. postgresql.conf - add the following lines

# The archive\_mode must be set to on for archiving to happen.

archive\_mode = on

# This is the command to invoke for each WAL file to be archived.

archive\_command = 'test ! -f /backups/archives/%f && cp %p /backups/archives/%f'

# Ensure there is at least one WAL file for each "archive\_timeout" duration, it forces the server to switch to a new WAL segment file at least that often, archive\_timeout settings of a minute or so are usually reasonable

archive\_timeout = 900

# Set high enough to leave at least one session available for the backup

max\_wal\_senders = 3

# All WAL records required for the backup must contain sufficient full-page writes

full\_page\_writes = on

# Keep around at least these many WAL files (aka segments). 16MB each; 0 disables, should be set high enough that the log is not removed before the end of the backup.

wal\_keep\_segments = 64

# The WAL level must be archive or higher for continuous WAL archiving, hot\_standby or higher for standby, posible values are minimal, archive, hot\_standby, logical.

wal\_level = archive

1. pg\_hba.conf - add the following lines

host replication postgres 127.0.0.1/32 trust

host replication postgres ::1/128 trust

1. create /var/lib/pgsql/hot\_backup.sh with the following 4 lines

#!/bin/bash

. /var/lib/pgsql/.bash\_profile

Backup\_Location="/backups/database/backup\_`date +%d%b%y\_%T`"

/usr/pgsql-9.6/bin/pg\_basebackup --host=localhost --port=5432 --username=postgres --xlog --progress --pgdata=$Backup\_Location --no-password

1. create crontab entry for hot backups and log maintenance

#

# Carry out a Daily Hot Backup

00 09 \* \* \* /var/lib/pgsql/hot\_backup.sh > /backups/logs/hot\_backup\_`date +\%d\%b\%y\_\%T`.log 2>&1

#

# Remove postgresql archive logs older than 30 days

00 09 \* \* \* /usr/bin/find /backups/archives/ -name 00\\* -mtime +30 -exec rm {} \; > /backups/logs/archive\_cleanup\_`date +\%d\%b\%y\_\%T`.log 2>&1

#

# Remove postgresql hot backups older than 30 days

00 09 \* \* \* /usr/bin/find /backups/database/ -name backup\_\\* -type d -mtime +30 -exec rm -rf {} \; > /backups/logs/backup\_cleanup\_`date +\%d\%b\%y\_\%T`.log 2>&1

#

# Remove postgresql backup logs older than 30 days

00 09 \* \* \* /usr/bin/find /backups/logs/ -name \\*.log -mtime +30 -exec rm {} \; > /backups/logs/archive\_cleanup\_`date +\%d\%b\%y\_\%T`.log 2>&1