**上海商業銀行股份有限公司**

**新核心系統INFRA**

**軟硬體設備建置專案服務**

**CICD安裝手冊**

**IBM Confidential**

台灣國際商業機器股份有限公司

目錄

[一. Gitlab 3](#_Toc126255991)

[二. Jenkins 5](#_Toc126255992)

[三. Redmine 7](#_Toc126255993)

[四. Sonarqube 11](#_Toc126255994)

[五. Nexus 15](#_Toc126255995)

1. **Gitlab**
2. 安裝套件

* 安裝基礎套件

$ yum install -y curl policycoreutils openssh-server perl postfix

* 啟動ssh服務

$ yum install -y curl policycoreutils openssh-server perl postfix

* 新增Gitlab YUM Repository及安裝

$ curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash

$ export LC\_ALL="en\_US.UTF-8"

$ yum install -y gitlab-ce

1. 配置Gitlab

修改Gitlab參數都在/etc/gitlab/gitlab.rb設定檔編輯修改。

external\_url "https://ddpsg01.cbsd.scsb.com.tw"

* 設定網址
* 開啟HTTPS

nginx['redirect\_http\_to\_https'] = true

registry\_nginx['redirect\_http\_to\_https'] = true

mattermost\_nginx['redirect\_http\_to\_https'] = true

* 設定ssl憑證

將憑證放入目錄

$ mkdir -p /etc/gitlab/ssl

$ chmod 755 /etc/gitlab/ssl

$ cp ca.crt ddpsg01.cbsd.scsb.com.tw.key ddpsg01.cbsd.scsb.com.tw.crt /etc/gitlab/ssl/

修改Gitlab參數檔

* 修改預設git資料儲存位置

$ gitlab-ctl reconfigure

git\_data\_dirs({ "default" => { "path" => "/data/gitlab/git-data" } })

letsencrypt['enable'] = false

nginx['ssl\_client\_certificate'] = "/etc/gitlab/ssl/ca.crt"

nginx['ssl\_certificate'] = "/etc/gitlab/ssl/ddpsg01.cbsd.scsb.com.tw.key.crt"

nginx['ssl\_certificate\_key'] = "/etc/gitlab/ssl/ddpsg01.cbsd.scsb.com.tw.key"

上述項目如果有更改都需要執行底下指令進行重新配置生效。

1. Gitlab指令基本介紹

Gitlab的控制都是透過gitlab-ctl進行操作，底下列出幾個常用參數。

* gitlab-ctl reconfigure: 重新配置
* gitlab-ctl stop: 停止服務
* gitlab-ctl start: 啟動服務
* gitlab-ctl status: 查看服務狀態

1. **Jenkins**
2. 安裝

安裝java和jenkins

$ wget -O /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

$ rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

$ yum upgrade

$ yum install java-11-openjdk

$ yum install jenkins

$ systemctl daemon-reload

啟動Jenkins

$ systemctl enable jenkins

$ systemctl start jenkins

1. 修改配置

編輯Jenkins systemd檔案/usr/lib/systemd/system/jenkins.service，並修改底下參數。

* 修改Jenkins工作目錄

Environment="JENKINS\_HOME=/datam/jenkins"

WorkingDirectory=/datam/jenkins

之後記得建立目錄。

$ mkdir /datam/jenkins

$ chown jenkins:jenkins /datam/jenkins

* 設定HTTPS和憑證

將ssl憑證轉換jks格式可參考底下。

Converts SSL certs to intermediate PKCS12 format

$ openssl pkcs12 -export -out brobridge.lab.p12 -passout 'pass:brobridge' -inkey brobridge.lab.key -in brobridge.lab.crt -certfile ca.crt -name jenkins.brobridge.lab

Convert PKCS12 to JKS format

$ keytool -importkeystore -srckeystore brobridge.lab.p12 -srcstorepass 'brobridge' -srcstoretype PKCS12 -srcalias jenkins.brobridge.lab -deststoretype JKS -destkeystore jenkins.brobridge.lab.jks -deststorepass 'brobridge' -destalias jenkins.brobridge.lab

$ vim /usr/lib/systemd/system/jenkins.service

Environment="JENKINS\_PORT=80"

Environment="JENKINS\_HTTPS\_LISTEN\_ADDRESS=0.0.0.0"

Environment="JENKINS\_HTTPS\_PORT=443"

Environment="JENKINS\_HTTPS\_KEYSTORE=/etc/jenkins/ssl/ddpsj01.cbsd.scsb.com.tw.jks"

Environment="JENKINS\_HTTPS\_KEYSTORE\_PASSWORD=brobridge"

AmbientCapabilities=CAP\_NET\_BIND\_SERVICE

將憑證放置憑證目錄。

$ mkdir -p /etc/jenkins/ssl

$ cp ddpsj01.cbsd.scsb.com.tw.jks /etc/jenkins/ssl

$ chown -R jenkins: /etc/jenkins

$ chmod 600 /etc/jenkins/ssl/

$ chmod 600 /etc/jenkins/ddpsj01.cbsd.scsb.com.tw.jks

上述設定完畢重新啟動Jenkins服務。

$ systemctl daemon-reload

$ systemctl restart jenkins.service

1. Plugin安裝

* 網頁安裝

下載Jenkins plugin後在 Web(管理Jenkins>管理外掛程式)點”進階”。



Web找到”部署外掛”，點擊”檔案”選擇檔案位置，選擇完畢點擊”部署”開始安裝。



安裝完畢確認無Job執行後重新啟動Jenkins，再回到外掛管理頁面中的已安裝確認安裝結果。



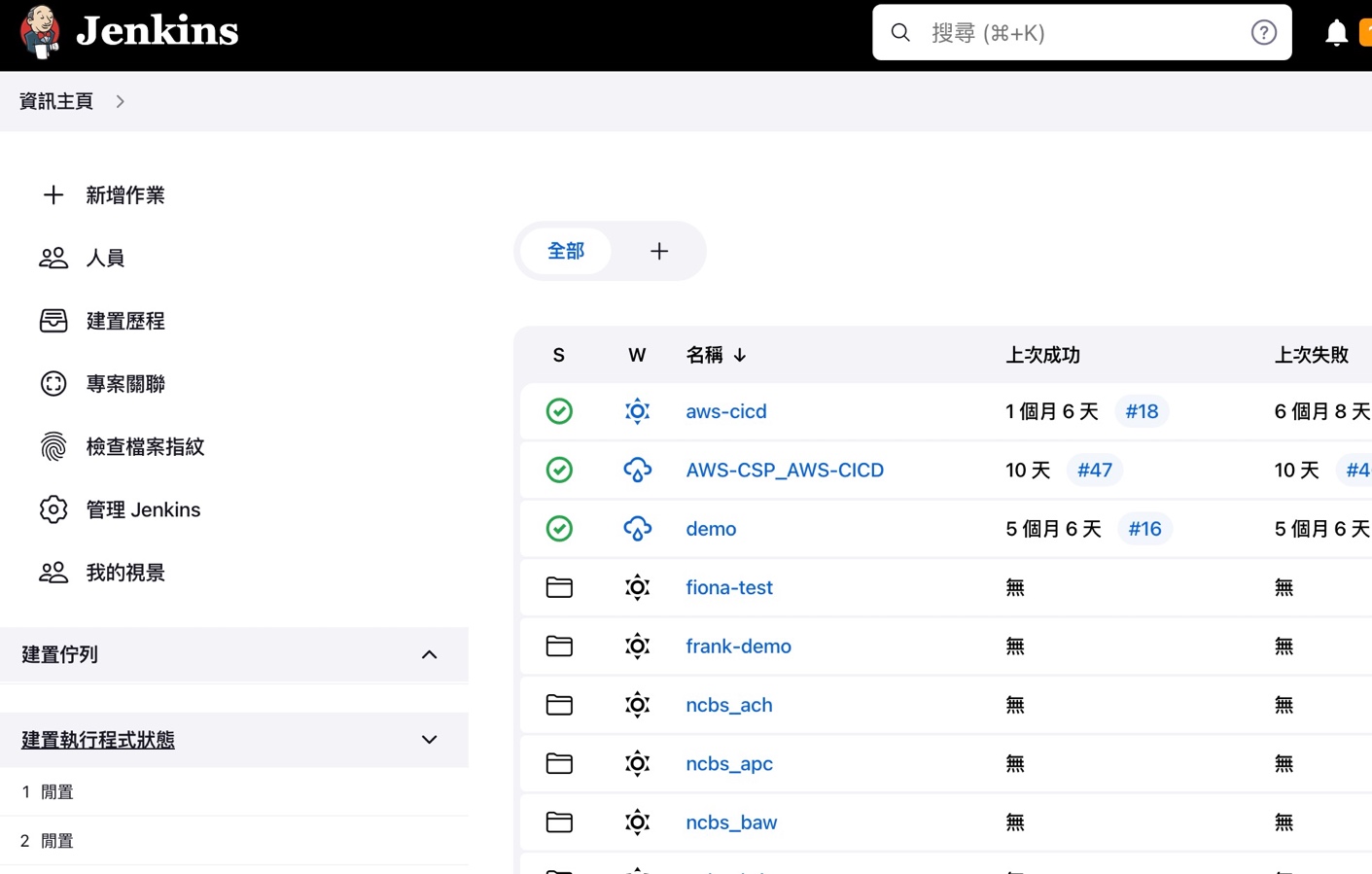
* 本地系統安裝

1. 將Plugin檔案複製到”/datam/jenkins/plugins”

$ cp <plugin\_name> /datam/jenkins/plugins

1. 修改使用者權限

$ chown jenkins:jenkins /datam/jenkins/plugins/<plugin\_name>

1. 到Web確認無Job執行中  
   
2. 重新啟動Jenkins 執行

$ systemctl restart jenkins

1. Web在外掛管理頁面的已安裝可確認安裝結果



1. **Redmine**
2. 安裝

安裝 nginx 和 docker套件

$ yum install nginx docker

1. Docker compose執行redmine服務

啟動Docker。

$ systemctl enable docker.service

$ systemctl start docker.service

建立工作目錄。

$ mkdir /datam/mariadb /datam/redmine

匯入redmine docker image。

$ docker load < mariadb.tar

$ docker load < redmine.tar

編輯docker-compose，修改volume掛載位置。

version: '2'

services:

mariadb:

image: docker.io/bitnami/mariadb:10.6

restart: always

volumes:

- /datam/mariadb:/bitnami/mariadb:Z

environment:

# ALLOW\_EMPTY\_PASSWORD is recommended only for development.

- ALLOW\_EMPTY\_PASSWORD=yes

- MARIADB\_USER=bn\_redmine

- MARIADB\_DATABASE=bitnami\_redmine

redmine:

image: docker.io/bitnami/redmine:5.0.1

restart: always

ports:

- '8033:3000'

volumes:

- 'redmine\_data:/bitnami/redmine'

depends\_on:

- mariadb

environment:

# ALLOW\_EMPTY\_PASSWORD is recommended only for development.

- ALLOW\_EMPTY\_PASSWORD=yes

- REDMINE\_DATABASE\_HOST=mariadb

- REDMINE\_DATABASE\_PORT\_NUMBER=3306

- REDMINE\_DATABASE\_USER=bn\_redmine

- REDMINE\_DATABASE\_NAME=bitnami\_redmine

- REDMINE\_USERNAME=admin

- REDMINE\_PASSWORD=admin

volumes:

- /datam/redmine:/bitnami/redmine:Z

執行。

$ docker-compose up -d

1. Nginx proxy設定80和443 port導向、憑證及TLS Version

設定nginx檔(/etc/nginx/conf.d/redmine.conf)，如下。

# the server directive is Nginx's virtual host directive

server {

listen 80;

server\_name ddpsr01.cbsd.scsb.com.tw;

# redirect http to https www

return 301 https://ddpsr01.cbsd.scsb.com.tw$request\_uri;

}

server {

listen 443 ssl;

server\_name ddpsr01.cbsd.scsb.com.tw;

access\_log /var/log/nginx/redmine.access.log;

error\_log /var/log/nginx/redmine.error.log;

ssl\_protocols TLSv1.2;

ssl\_certificate /etc/redmine/ssl/ddpsr01.cbsd.scsb.com.tw.crt;

ssl\_certificate\_key /etc/redmine/ssl/ddpsr01.cbsd.scsb.com.tw.key;

location / {

proxy\_pass http://127.0.0.1:3000;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header X-Forwarded-Proto https;

}

}

建立憑證目錄並將憑證放入。

$ mkdir -p /etc/redmine/ssl

$ cp ddpsr01.cbsd.scsb.com.tw.crt ddpsr01.cbsd.scsb.com.tw.key /etc/redmine/ssl

$ chmod 600 /etc/redmine/ssl/ddpsr01.cbsd.scsb.com.tw.\*

啟動Nginx服務。

$ systemctl enable nginx.service

$ systemctl start nginx.service

1. **Sonarqube**
2. Linux設定需求

$ vim /etc/sysctl.conf

vm.max\_map\_count=524288

fs.file-max=131072

$ vim /etc/security/limits.conf

sonarqube - nofile 65536

sonarqube - nproc 4096

立即套用設定

$ sysctl -w vm.max\_map\_count=524288

$ sysctl -w fs.file-max=131072

$ ulimit -n 131072

$ ulimit -u 8192

1. 安裝Podman、Java及Nginx

$ yum install podman podman-plugins nginx java-11-openjdk

1. Podman建立Sonarqube服務

建立工作目錄。

$ mkdir -p /datam/postgresql /datam/sonarqube/data /datam/sonarqube/extensions /datam/sonarqube/logs /datam/sonarqube/ssl

匯入Image。

$ podman load < sonarqube.tar

$ podman load < postgres.tar

Podman建立network。

$ podman network create sonar\_network

Podman運行PostgreSQL。

$ podman run -d --name postgresql \

-e POSTGRES\_USER=sonar \

-e POSTGRES\_PASSWORD=sonar \

-e POSTGRES\_DB=sonar \

-v /datam/postgresql:/var/lib/postgresql:Z \

-v /datam/postgresql/data:/var/lib/postgresql/data:Z \

--network sonar\_network \

--network-alias db \

--hostname postgresql \

postgres:12

Podman運行Sonarqube。

$ podman run -d --name sonarqube \

-p 9000:9000 \

-e SONAR\_JDBC\_URL=jdbc:postgresql://postgresql:5432/sonar \

-e SONAR\_JDBC\_USERNAME=sonar \

-e SONAR\_JDBC\_PASSWORD=sonar \

-v /datam/sonarqube/data:/opt/sonarqube/data:Z \

-v /datam/sonarqube/extensions:/opt/sonarqube/extensions:Z \

-v /datam/sonarqube/logs:/opt/sonarqube/logs:Z \

--network sonar\_network \

--network-alias sonarqube \

--hostname sonarqube \

sonarqube:9.6.1-community

建立Systemd服務。

$ podman generate systemd --files --name postgresql

$ podman generate systemd --files --after postgresql --name sonarqube

$ cp container-postgresql.service container-sonarqube.service /etc/systemd/system/

$ systemctl daemon-reload

之後可透過systemctl執行、關閉或重啟服務。

$ systemctl start container-postgresql.service container-sonarqube.service

$ systemctl enable container-postgresql.service container-sonarqube.service

1. Nginx proxy設定80和443 port導向及憑證、憑證及TLS Version

設定nginx檔(/etc/nginx/conf.d/sonarqube.conf)，如下。

# the server directive is Nginx's virtual host directive

server {

listen 80;

server\_name ddpsq01.cbsd.scsb.com.tw;

client\_max\_body\_size 128M;

# redirect http to https www

return 301 https://ddpsq01.cbsd.scsb.com.tw$request\_uri;

}

server {

listen 443 ssl;

server\_name ddpsq01.cbsd.scsb.com.tw;

client\_max\_body\_size 128M;

access\_log /var/log/nginx/sonarqube.access.log;

error\_log /var/log/nginx/sonarqube.error.log;

ssl\_protocols TLSv1.2;

ssl\_certificate /etc/sonarqube/ssl/ddpsq01.cbsd.scsb.com.tw

ssl\_certificate\_key /etc/sonarqube/ssl/ddpsq01.cbsd.scsb.com.tw.key;

location / {

proxy\_pass http://127.0.0.1:9000;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header X-Forwarded-Proto https;

}

}

建立憑證目錄並將憑證放入。

$ mkdir -p /etc/sonarqube/ssl

$ cp ddpsq01.cbsd.scsb.com.tw.crt ddpsq01.cbsd.scsb.com.tw.key /etc/sonarqube/ssl

$ chmod 600 /etc/sonarqube/ssl/ddpsq01.cbsd.scsb.com.tw.\*

啟動Nginx服務。

$ systemctl enable nginx.service

$ systemctl start nginx.service

1. **Nexus**
2. 安裝Podman

$ yum install podman

1. Podman運行Nexus

建立工作目錄。

$ mkdir /datam/nexus-data && chown -R 200 /datam/nexus-data

匯入Image。

$ podman load < nexus.tar

運行Nexus。

$ podman run -d -p 8081:8081 --name nexus -v /datam/nexus-data:/nexus-data sonatype/nexus3:3.42.0

建立Systemd服務。

$ podman generate systemd --files --name nexus

$ cp container-nexus.service /etc/systemd/system/

$ systemctl daemon-reload

之後可透過systemctl執行、關閉或重啟服務。

$ systemctl start container-nexus.service

$ systemctl enable container-nexus.service

1. Nginx proxy設定80和443 port導向及憑證、憑證及TLS Version

設定nginx檔(/etc/nginx/conf.d/nexus.conf)，如下。

# the server directive is Nginx's virtual host directive

server {

listen 80;

server\_name ddpsn01.cbsd.scsb.com.tw;

client\_max\_body\_size 128M;

# redirect http to https www

return 301 https://ddpsn01.cbsd.scsb.com.tw$request\_uri;

}

server {

listen 443 ssl;

server\_name ddpsn01.cbsd.scsb.com.tw;

client\_max\_body\_size 128M;

access\_log /var/log/nginx/nexus.access.log;

error\_log /var/log/nginx/nexus.error.log;

ssl\_protocols TLSv1.2;

ssl\_certificate /etc/nexus/ssl/ddpsn01.cbsd.scsb.com.tw

ssl\_certificate\_key /etc/nexus/ssl/ddpsn01.cbsd.scsb.com.tw.key;

location / {

proxy\_pass http://127.0.0.1:8081;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header X-Forwarded-Proto https;

}

}

建立憑證目錄並將憑證放入。

$ mkdir -p /etc/nexus/ssl

$ cp ddpsn01.cbsd.scsb.com.tw.crt ddpsn01.cbsd.scsb.com.tw.key /etc/nexus/ssl

$ chmod 600 /etc/nexus/ssl/ddpsn01.cbsd.scsb.com.tw.\*

啟動Nginx服務。

$ systemctl enable nginx.service

$ systemctl start nginx.service