Jenkins

CI（継続的インテグレーション）を行うためのツール。
本体は、war ファイル 1 つだけで起動も簡単。

起動

```
java -jar jenkins.war
```

で起動。

```
http://localhost:8080
```

でアクセスできる。もしポートを変更したい場合は httpPort オプションで変更できる。

```
java -jar jenkins.war &amp;#8211;httpPort=8081
```

その他の起動オプション

https://wiki.jenkins-ci.org/display/JA/Starting+and+Accessing+Jenkins

コマンドライン・パラメータ	説明
--javaHome=$JAVA_HOME	Java や Ant が必要なビルドを実行するときに使用する $JAVA_HOME を Jenkins に渡します。これは Jenkins を起動する JRE ではありません (Jenkins 上で使用する JRE を指定します)。デフォルトは、Jenkins を実行するのに使用した JRE です。
--httpPort=$HTTP_PORT	Jenkins が使用する HTTP プロトコルのポート番号を指定します。デフォルトは 8080 です。これを使用しない (_HTTPS_ を使用) ようにするには、-1 を指定します。
--httpListenAddress=$HTTP_HOST	このパラメータで、Jenkins がリクエストを受け付ける IP アドレスを指定できます。環境変数 $HTTP_HOST が指定されていれば、それを利用します。デフォルトでは 0.0.0.0 です。--- この場合、すべての有効なインタフェース上で (httpPort で指定された) ポートをリスンします。
--httpsPort=$HTTP_PORT	HTTPS プロトコルで使用するポート番号を指定します。
--httpsListenAddress=$HTTPS_HOST	Jenkins が HTTPS プロトコルでリクエストを受け付ける IP アドレスを指定できます。環境変数 $HTTPS_HOST が指定されていれば、それを利用します。

--ajp13Port=$AJP_PORT	標準の AJP13 プロトコルを使う場合、$AJP_PORTR を利用して、Jenkins がリスンするポートを指定できます。デフォルトは 8009 です。(_HTTPS_ を使用するため)　これを使用しないようにするには、-1 を指定します。
--ajp13ListenAddress=$AJP_HOST	Jenkins が AJP13 プロトコルのリクエストを受け付ける、IP アドレスを指定できます。デフォルトでは 0.0.0.0 です。 --- i.e. この場合、すべての有効なインタフェース上で (ajpPort で指定された) ポートをリスンします。
--argumentsRealm.passwd.$ADMIN_USER=password	ユーザー $ADMIN_USER のパスワードを設定します。Jenkins のセキュリティが有効化されている場合、Jenkins やプロジェクトを設定するために、$ADMIN_USER でログインする必要があります。注意 : 管理者権限を持つユーザーも指定しなければなりません。(以下を参照してください)
--argumentsRealm.roles.$ADMIN_USER=admin	$ADMIN_USER を管理者として設定し、Jenkins のセキュリティが有効化されていれば、Jenkins を設定できるようにします。詳しくは Jenkins のセキュリティを参照してください。

デーモン化

Linux

https://gist.github.com/wataru420/1757063
実行するユーザと jenkins の各ファイルのパスを決める
ディレクトリ構成や各パスは自由。
とりあえず、ここでは以下の構成にする。

```
jenkins のルート
├── bin <--- 実行シェルとか jenkins.war とかを置く
├── home <--- JENKINS_HOME
└── log
```

■ start-jenkins.sh の作成

```
#!/bin/bash
JENKINS_ROOT=/home/test/jenkins
JENKINS_WAR=${JENKINS_ROOT}/bin/jenkins.war
JENKINS_LOG=${JENKINS_ROOT}/log/jenkins.log
JENKINS_OPT="--httpPort=8081  --ajp13Port=-1"
 export JENKINS_HOME=${JENKINS_ROOT}/home
JAVA=/usr/bin/java
nohup nice $JAVA -jar $JENKINS_WAR $JENKINS_OPT > $JENKINS_LOG 2>&1 &
```

■ stop-jenkins.sh

```
#!/bin/bash
```

```
kill `ps -ef | grep [j]enkins.war | awk '{ print $2 }'`
```

■ /etc/init.d/jenkins

```
#! /bin/bash
#
# chkconfig: - 85 15
# description: Jakarta Tomcat Java Servlets and JSP server
# jenkins Start/Stop the Jenkins Continuous Integration server.
# Source function library.
. /etc/rc.d/init.d/functions
# Get config.
. /etc/sysconfig/network
# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 0
startup=/home/test/jenkins/bin/start-jenkins.sh
shutdown=/home/test/jenkins/bin/stop-jenkins.sh
#export JAVA_HOME=/usr/local/java/
JENKINS_USER=test
start(){
  echo -n $"Starting Jenkins service: "
  su - $JENKINS_USER -c $startup
  RETVAL=$?
  echo
}
stop(){
  action $"Stopping Jenkins service: "
  su - $JENKINS_USER -c $shutdown
  RETVAL=$?
  echo
}
status(){
  numproc=`ps -ef | grep [j]enkins.war | wc -l`
  if [ $numproc -gt 0 ]; then
  echo "Jenkins is running..."
  else
    echo "Jenkins is stopped..."
  fi
}
restart(){
  stop
  sleep 5
  start
}
# See how we were called.
case "$1" in
start)
start
;;
stop)
stop
;;
status)
status
;;
restart)
restart
;;
*)
echo $"Usage: $0 {start|stop|status|restart}"
exit 1
esac
exit 0
```

■パーミッション変更とデーモン追加

各シェルのパーミッションを設定して

```
chkconfig jenkins on
```

でデーモン追加。

基本的な設定

workspace の設定

```
[Jenkins の管理 ] -> [ システムの設定 ] -> [ 高度な設定 ]
```

にあるワークスペース・ルートディレクトリを

```
${ITEM_ROOTDIR}/workspace
```

にすると、jobs の各ジョブの配下にワークスペースが作成される。
バックアップ等の都合を考えて、状況に応じて変更すると良い。

プラグイン

状態によって jenkins の表示を変える

http://natural-born-minority.blogspot.jp/2012/12/blog-post.html

```
Emotional Jenkins Plugin
```

というプラグインを入れる。ただし、

```
1.454
```

以前のバージョンでないと機能しない。

状態を判定する閾値を設定する

http://shokos.hatenablog.com/entry/2012/08/15/112833
テストが失敗しても状態が「問題ない」と判定される場合は、閾値を設定する。

```
xUnit Plugin
```

を導入して、閾値を全て 0 にする。