

Integrate Jenkins with Password Safe

Overview

[Jenkins](#) is a leading open source automation server used by many DevOps teams and CI/CD workflows. The steps often included in a Jenkins pipeline require credentials or secrets for executing various tasks, such as logging into a remote system, providing credentials for connecting to database files, and many other privileged tasks. A critical security step for protecting these secrets is to ensure they are not hard coded in pipeline configurations or stored insecurely, as they are leveraged in these automated workflows. This integration guide provides example steps for leveraging secrets managed externally by Password Safe within your Jenkins CI/CD pipelines.

Prerequisites

Before implementing the integration, ensure the following prerequisites are met:

- A Jenkins environment with the ability to execute pipelines with steps, including shell commands.
- A Password Safe environment with managed SSH credentials.
- The PSRUN Password Safe utility available where the Jenkins Pipeline executes.

Configure Password Safe

In order to retrieve credentials from Password Safe, using either the API or the PSRUN utility, an API registration must be configured. This API registration must also be enabled for a group that provides access to the necessary SSH credentials for connecting to the remote build system. The group must have an access policy that is auto-approved to view the credentials.

The API registration must include an authentication rule that allows connectivity from the Jenkins agent node.



For more information on configuring the API account and applying the required authentication rules for use with PSRUN, please see [Configure API Registration](https://www.beyondtrust.com/docs/beyondinsight-password-safe/ps/admin/configure-api-registration.htm) at <https://www.beyondtrust.com/docs/beyondinsight-password-safe/ps/admin/configure-api-registration.htm>.

Setup Jenkins

In this example Jenkins Pipeline, the PSRUN utility is used to retrieve managed SSH credentials. These credentials are used to connect to a remote system to execute a build step in the pipeline.

Below is an example Jenkins file that leverages the Jenkins credential plugin to provide the initial API key information. These can be provided to the pipeline in a number of ways, but the following values are required:

- **Host:** URL for the Password Safe site.
- **API Key:** API Key generated during the API registration within Password Safe.
- **RunAs User:** The Password Safe user account with permissions to use the API key and access to the managed secrets.

In the following example, the Jenkins agent node has the PSRUN utility installed to `/usr/bin` and `sshpass` is included for remote connectivity.

It may also be necessary to include the remote host's fingerprint in the Jenkins known_hosts file.

```

pipeline {
  agent any

  environment {
    PSCLLOUD = credentials('pscloud')
  }

  stages {
    stage('build') {
      steps {
        sh '''
        #remove log output
        set +x

        #use psrun to retrieve credentials
        BPASS=$(/usr/bin/psrun -i ps.example.com $PSCLLOUD_PSW $PSCLLOUD_
        USR RetrievePassword build01 builder "Jenkins Build")

        #execute remote command using managed credentials
        /usr/bin/sshpass -p$BPASS ssh builder@build01 make -f projects/app01/Makefile

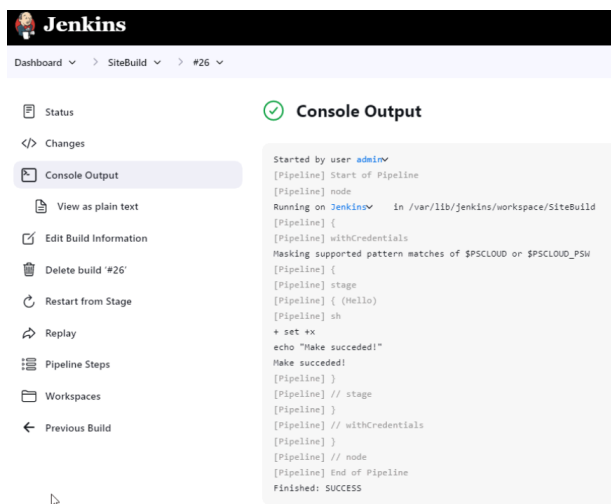
        '''
      }
    }
  }
}

```

This Jenkins pipeline registers stored credentials with the ID of *PSCLLOUD*. This credential is of type user/pass with the runas user stored in the username and the API key stored as the password.

PSRUN then uses this API key to retrieve ssh credentials for the *builder* managed account on the remote build environment *build01*.

The final step is to use the retrieved managed credentials to remotely execute a build step for the application. This simple workflow can be adapted to retrieve managed credentials of any type to facilitate any number of pipeline steps.



The screenshot shows the Jenkins web interface. The top navigation bar includes 'Dashboard', 'SiteBuild', and '#26'. The left sidebar contains various actions like 'Status', 'Changes', 'Console Output', 'View as plain text', 'Edit Build Information', 'Delete build', 'Restart from Stage', 'Replay', 'Pipeline Steps', 'Workspaces', and 'Previous Build'. The main area displays the 'Console Output' for a pipeline run, which includes the following text:

```

Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/SiteBuild
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $PSCLLOUD or $PSCLLOUD_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] sh
+ set +x
echo "Make succeeded!"
Make succeeded!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```



For more information on deployment and advanced configuration options, please see the [PSRUN User Guide](https://www.beyondtrust.com/docs/beyondinsight-password-safe/ps/psrun/index.htm) at <https://www.beyondtrust.com/docs/beyondinsight-password-safe/ps/psrun/index.htm>.