
Setting up BMC Deployment Engine

Version 14.2 authored by  [superadmin](#) on 2025/02/25 04:38:16

Table of Contents

Before you begin	5
Creating and configuring users	7
To add users	7
To add the git user to the jenkins group and the jenkins user to the git group	8
To provide sudo access to the git user	8
To configure passwordless sudo access for the git user	8
Performing post-installation configurations	10
To update the Jenkins credentials	10
To update the node configuration in a cloud environment	12
To update the node configuration in a non cloud environment	12
To add the Jenkins libraries	12
Running the deployment pipelines in dry-run mode	13
To add the HELIX_FULL_STACK_UPGRADE deployment pipeline	14
Troubleshooting	15
Where to go from here	20

Set up BMC Deployment Engine to deploy BMC Helix Service Management applications.

BMC Deployment Engine is set up on a single virtual machine. For Jenkins installation and configuration, the host and hostname reference point to this single server. Several tools, such as Jenkins, git, Helm, and kubectl are installed on this server, and these tools are used by the Jenkins pipelines to deploy BMC Helix Service Management applications.

Git is required to maintain the automation scripts. kubectl is the Kubernetes client that is used to run the Kubernetes commands. Helm is the package manager used to deploy applications in Kubernetes.

The Deployment Engine automation script sets up the Jenkins pipeline framework, which is used to set up BMC Helix Service Management applications.

Related topics

[System-requirements](#)

[BMC-Deployment-Engine-overview](#)

[FAQ](#)

✓ Skills required

- Using Jenkins
- Using Linux

See [Recommended-skill-set-and-trainings](#)

The steps to set up BMC Deployment Engine are shown in the following figure:



Before you begin

! Important

- To login to the Jenkins server, use SSH clients, such as PuTTY, MobaXterm, or mRemoteNG. This server is referred to as Jenkins server or `<Deployment Engine Server>` in the rest of the document.
- By default, all the commands must be executed as non root user and the non root user must be configured for sudo access.
- Unless specified, user refers to non root user for this setup.
- Make sure that the user has default shell as sh or bash.
- Make sure that the Jenkins server has access to the [internet](#) to access the online Jenkins repository for OS software update or upgrade as well as installing tools required to setup Deployment Engine and supporting binaries.
- The nomenclature used in pipeline development is as follows:
 - The Home directory of non root user is referred to as `<HOME>` in all subsequent steps. By default, it is `/home/git`.
 - `<user>` refers to the git repo owner. Jenkins user owns the Jenkins setup. The git user is a member of the git and Jenkins groups. The Jenkins user is member of the Jenkins and git groups.
 - `<hostname>` is the host name/IP address of the Jenkins server, which is resolvable by all participating entities, such as databases and clusters in this environment.
 - `<GIT_REPO_DIR>` is the location where the entire Deployment Engine git code is copied by the installer for setting up the pipelines. The recommended size of this directory must be 10GB or more.
 - `<Library_REPO_DIR>` is the location where the installer copies the required libraries that need to be configured manually after installation.

Ensure that your environment meets the following requirements:

- **Virtual machine**—A fresh and dedicated physical server or virtual machine with the following configuration for the Jenkins server:

Component	No of	vCPU	Operating System	RAM (GB)	Disk (GB)
Jenkins server	1	2	<ul style="list-style-type: none"> ◦ Cent OS 7.x and 8.x are certified. ◦ RHEL 7.x and 8.x are certified. ◦ Equivalent releases of Fedora and Oracle Linux are supported. 	Minimum 8 GB	100

- **Kubernetes cluster**—Kubernetes cluster is accessible from the Deployment Engine server. For Kubernetes cluster requirements, see [System Requirements](#).
- **Ansible 2.9**—is configured, and working fine. You must install Ansible 2.9 for setting up BMC Deployment Engine.

- **Yum utility**—is installed, configured, and working fine.

Make sure that the yum client is installed and is accessible, and you can install certified repos stable version of applications.

Run the following commands to update your system libraries and packages to the latest available version.

```
yum update -y
yum upgrade -y
yum clean all
```

- **Unzip utility**—is installed.

Use the following command to install the unzip utility:

```
sudo yum install unzip
```

- **Perl**—Perl (version 5.x +) is installed and is accessible from shell prompt.

Use the following command to set up Perl:

```
sudo yum install perl
```

- **Perl-Data-Dumper package**—Set up Perl-Data-Dumper package by using the following commands:

```
sudo yum makecache
sudo yum -y install perl-Data-Dumper
```

- Make sure that /etc/resolv.conf has proper DNS entries.
- From any remote machine run the following commands:

```
nslookup <jenkins-server-name>
nslookup <jenkins-server-ip>
```

both the commands should resolve via DNS from all computers where clusters are setup.

- from /etc/hosts make sure to have only relevant entries in this file. If an entry is not correct, delete the entry.
- Remove the ipv6 entry for the localhost and ensure that the hostname resolves to the correct DNS IP address.
- Run the `yum repolist` command to view all accessible and OS matching repositories. The `yum` commands should be able to install the required repository software.

- Download the **BMC_Helix_Innovation_Suite_And_Service_Management_Apps_Version_21.3.10.zip** file from EPD to the Deployment Engine server.
This file contains the following files:

- **BMC_Remedey_Deployment_Manager_Configuration_Release_21.3.10.zip** - This file contains the repositories that go in to git.
- **BMC_Remedey_Deployment_Engine_Setup_21.3.10.zip**
- Download the **BMC_Remedey_Deployment_Engine_Setup_21.3.10.zip** file. This file contains the BMC Deployment Engine automation script.

For details, see [Downloading-the-installation-files](#).

Ensure that the following repository URLs are whitelisted by your administrator so that BMC Deployment Engine can install these applications while running the BMC Deployment Engine automation script:

RHEL8	Product Documentation for Red Hat Enterprise Linux 8
CENTOS8	CentOS Vault
MSSQL	https://packages.microsoft.com
Jenkins server	https://pkg.jenkins.io/
Jenkins cli	https://github.com/jenkinsci
Kubectl	https://s3.us-west-2.amazonaws.com/amazon-eks
Helm	https://get.helm.sh
Groovy	https://dlcdn.apache.org/groovy
EPEL	https://mirrors.fedoraproject.org/

[Back to top](#)

Creating and configuring users

BMC Deployment Engine uses the following two users:

- git—Owner of git repositories where deployment code is present. This user requires sudo access.
- jenkins—Jenkins server runs as jenkins user. This is a non-interactive user.

You must create and configure these users.

To add users

Perform the following steps to add the users:

```
sudo useradd git -m
sudo passwd git
sudo useradd jenkins -m
sudo passwd jenkins
```

To add the git user to the jenkins group and the jenkins user to the git group
Perform the following steps to add the users:

```
sudo usermod -a -G git jenkins  
sudo usermod -a -G jenkins git
```

To provide sudo access to the git user

Run the following command to provide sudo access to the git user:

```
sudo usermod -aG wheel git
```

To configure passwordless sudo access for the git user

Perform the following steps to configure passwordless sudo access to the git user:

1. Run the following command:

```
sudo visudo
```

2. Uncomment the following line:

```
%wheel ALL=(ALL) NOPASSWD: ALL
```

3. Comment the other lines that start with %wheel.
For example:

```
%wheel ALL=(ALL) ALL
```

4. Save the changes.

[Back to top](#)

Running the BMC Deployment Engine automation script

The BMC Deployment Engine automation script installs the following software:

- Python
- Git
- kubectl
- unzip
- Helm
- Java
- additional packages, such as MS SQL tools, unixODBC, xmlstarlet, dos2unix, jq

The automation script also installs Jenkins locally on a node (In earlier versions of BMC Helix Service Management Deployment, you had to manually set up Jenkins on a local node).

Perform the following steps to run the BMC Deployment Engine automation script:

1. Login as git user:

```
ssh git@<hostname>
```

or switch to the git user by using the following command:

```
su - git
```

2. Copy the **BMC_Helix_Innovation_Suite_And_Service_Management_Apps_Version_21.3.10.zip** file downloaded from EPD and extract the files to the git user home directory or any other directory owned by the git user.
 - **BMC_Remedey_Deployment_Manager_Configuration_Release_21.3.10.zip**
 - **BMC_Remedey_Deployment_Engine_Setup_21.3.10.zip**
3. Copy the **BMC_Remedey_Deployment_Engine_Setup_21.3.10.zip** file downloaded from EPD and extract the files to the git user home directory or any other directory owned by the git user.
4. Change the ownership of these files to git user by using the following command:

```
sudo chown -R git:git <filename>
```

5. Unzip **BMC_Remedey_Deployment_Engine_Setup_21.3.10.zip** file.
6. Change the directory to DE1.0.
7. Update **build.properties** and customize the following parameters according to your requirement:
 - ITSM_REPO_GIT_ZIP=
</path/to/BMC_Remedey_Deployment_Manager_Configuration_Release_21.3.xx.zip>
 - GIT_REPO_USER=git
 - GIT_USER_HOME_DIR=~git
 - JENKINS_USER=jenkins
 - JENKINS_INSTALL_DIR=/var/lib/jenkins
 - JENKINS_HOSTNAME=<fully-qualified-hostname-or-ip-address-where-DE-setup-is-planned>
 - ANSIBLE_NODE_ROOT_DIR=~/.ansible_node_root_dir
 - JENKINS_NODE_ROOT_DIR=~/.jenkins_node_root_dir
(This is the path that the Jenkins_node needs for various operations that are internal to Jenkins)
 - GIT_REPO_DIR=~/.git_repo/
(This is the location where the entire Deployment Engine git code is copied by the installer for setting up the pipelines.)
 - JENKINS_CONFIG_FILES_DIR=~/.Jenkins_Config_Files
(The Jenkins job pipeline uses this location to store its pipeline jobs.)
8. After updating the build.properties file, run the BMC Deployment Engine automation script to setup the Jenkins job pipeline framework.
The log file is generated in the Home directory and it can be used for debugging setup related issues.

```
$ sudo perl setup-Helix-ITSM-onPrem.pl 2>&1 | tee ~/BMC-HELIX-DE-AUTO.log.$$
```

! Important

The build.properties file of BMC Deployment Engine automation script has a KUBERNETES_VERSION parameter which only accepts Kubernetes versions 1.21 and earlier. We recommend you to set the value `KUBERNETES_VERSION` parameter as 1.21. This step will setup the kubectl version as 1.21.

After the script completes running, update kubectl version to 1.23 or 1.24 as per the cluster version.

[Back to top](#)

Performing post-installation configurations

To install plugins

1. Login to Jenkins server by using the following URL:

`http://<jenkins_server_host_name>:<jenkins_port>`



`$JENKINS_INSTALL_DIR/secrets/initialAdminPassword`

For example:

2. On the Jenkins User Interface, select **Install Suggested Plugins**.
3. In the **Admin User Creation** wizard, provide a preferred username and password and run the wizard.
4. Verify that ssh works fine with passwordless login from root user to `git@<jenkins_server_name>` and git user to `git@<jenkins_server_name>` by using the following command:

```
sudo ssh git@<jenkins_server_name>
ssh git@<jenkins_server_name>
```

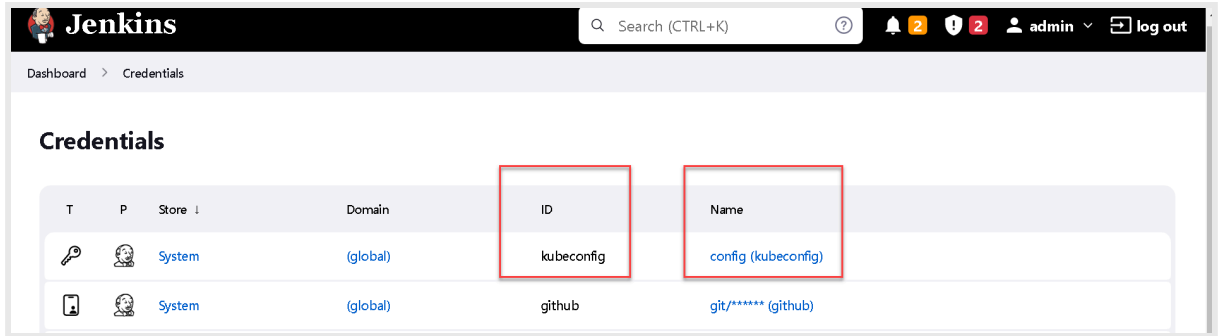
5. Add the kubeconfig file on the Jenkins server.
 1. Download the **kubeconfig.yaml** file to the \$HOME directory.
 2. Copy the **kubeconfig.yaml** file to \$HOME/.kube/config.

```
cp <kubeconfig complete file path> $HOME/.kube/config
```

To update the Jenkins credentials

1. Login to the Jenkins server by using the following URL:
`http://<Jenkins server host name>:<Jenkins port>/credentials`
2. Add the credentials in the **ID: kubeconfig** file.

1. Select kubeconfig and click the name of the kubeconfig credential.



2. Click **Update**.
3. Select the **Replace** checkbox.
4. Click **Choose File** and select the kubeconfig file
5. Click **Save**.
3. Add the credentials in the **ID: github** account.
 1. Select **github** and click the name of the github credential.
 2. Click **Update**.
 3. Click **Change Password** and enter the git user password.
 4. Click **Save**.
4. Add the credentials in the **ID: ansible_host** account.
 1. Select **ansible_host** and click the name of the ansible_host credential
 2. Click **Update**.
 3. Click **Change Password** and enter the git user password.
 4. Click **Save**.
5. Add the credentials in the **ID: ansible** file.
 1. Select **ansible** and click the name of the ansible credential.
 2. Click **Update**.
 3. Click **Change Password** and enter the git user password.
 4. Click **Save**.
6. Add credentials in the **ID: git** account.
 1. Select **git** and click the name of the git credential.
 2. Click **Update**.
 3. Click **Change Password** and enter the git user password.
 4. Click **Save**.
7. (For cloud environments) Update git user private key credentials
 1. Navigate to `http://<Jenkins server host name>:<Jenkins port>/credentials/store/system/domain/_/`
 2. Click **Add Credentials**.
 3. Enter the following details:

Kind	SSH Username with private key
ID	git_pk
Username	git
Private Key	<ol style="list-style-type: none"> 1. Click Enter. 2. Paste the contents of git user private key value (/home/git/.ssh/id_rsa)

4. Save the details.

Depending on your environment, you can choose to update the node configuration in a cloud or non cloud environment.

To update the node configuration in a cloud environment

1. Navigate to `http://<Jenkins server host name>:<Jenkins port>/computer`.
2. Perform the following steps for both nodes, the one with the actual hostname and another with the git hostname.
 1. Click the node name.
 2. Click **Configure**.
 3. Under Credentials, select the git user ssh key credential and click **Save**.
 4. Click **Launch Agent**.

To update the node configuration in a non cloud environment

1. Navigate to `http://<Jenkins server host name>:<Jenkins port>/computer`.
2. Perform the following steps for both nodes, the one with actual hostname and another with git-hostname.
 1. Click the node name.
 2. Click **Launch Agent**.

! If the status of the node is already 'online', then skip Step **b**.

To add the Jenkins libraries

Perform the following steps to add the pipeline-framework library:

1. On the Jenkins home page, click **Manage Jenkins**.
2. Select **Configure System**.
3. In Global Pipeline Libraries, add the pipeline-framework library as shown in the following figure:
4. In the Project Repository field, specify the complete path of pipeline-**framework.git** according to the environment.

<GIT_REPO_DIR> value can be fetched from **build.properties** file.

The screenshot shows the Jenkins Library configuration page for a library named 'pipeline-framework'. The 'Name' field is 'pipeline-framework'. The 'Default version' is 'master', with a note 'Currently maps to revision: 671c40bd62fb0e8cd942e9d340c86db63bcf42a1'. There are three checkboxes: 'Load implicitly' (unchecked), 'Allow default version to be overridden' (checked), and 'Include @Library changes in job recent changes' (checked). The 'Retrieval method' is 'Modern SCM'. The 'Source Code Management' section shows 'Git' as the provider and the 'Project Repository' as 'ssh://git@<Jenkins server name>/<GIT_REPO_DIR>/LIBRARY_REPO/pipeline-framework/pipeline-framework.git'.

5. Make sure you provide exact location of pipeline-**framework.git** according to the environment.
6. Add JENKINS-27413-workaround-library as shown in the following figure:
7. In the Project Repository field, specify the complete path of **JENKINS-27413-workaround-library.git** according to the environment.
8. Select **Load implicitly**.

The screenshot shows the Jenkins Library configuration page for a library named 'JENKINS-27413-workaround-library'. The 'Name' field is 'JENKINS-27413-workaround-library'. The 'Default version' is 'master', with a note 'Currently maps to revision: 6e99c461d051e3c27cc6c4a215ad4769f0313f73'. There are four checkboxes: 'Load implicitly' (checked), 'Allow default version to be overridden' (checked), 'Include @Library changes in job recent changes' (checked), and 'Cache fetched versions on controller for quick retrieval' (unchecked). The 'Retrieval method' is 'Modern SCM'. The 'Source Code Management' section shows 'Git' as the provider and the 'Project Repository' as 'ssh://git@<jenkins server name>/<GIT_REPO_DIR>/LIBRARY_REPO/jenkins-workaround/JENKINS-27413-workaround-library.git'. The 'Credentials' section shows a dropdown menu with '- none -' and an 'Add' button.

9. Save the changes.

Running the deployment pipelines in dry-run mode

Dry-run is a mandatory step to update the pipeline configuration for any changes to the BMC Helix Innovation Suite and Service Management Installer.

Important

Though you see the **Build with Parameters** option for all the parameters, you have to still perform a dry-run.

To run the deployment pipelines in dry-run mode

1. Navigate to Jenkins Dashboards to view all the pipelines required for deployment.
2. Select each pipeline and click **Build with Parameters**.
3. Click **Build**.

The build job status fails, which is expected.



- The agent-add-pipeline and HELIX_DR pipelines do not require a dry-run.
- For HELIX_ONPREM_DEPLOYMENT pipeline, enter the value of the AGENT parameter.
- The AGENT parameter value is the Jenkins agent to run the pipeline. Provide the value of the node that has the name `git-<hostname>`.
- Do not select any pipelines in the PRODUCT_DEPLOY section.
- For the other pipelines, run **Build with Parameters** with default values present in the pipeline.

To add the HELIX_FULL_STACK_UPGRADE deployment pipeline

1. Log in to the Jenkins server by using the following URL:
`http://<Jenkins server host name>:8080`
2. On the Jenkins home page, click **New Item**.
3. In the **Enter an item name** field, enter the pipeline name as **HELIX_FULL_STACK_UPGRADE**.
4. Select **Pipeline** and click **OK**.
5. Click the **Pipeline** tab.
6. Enter the following information:

Field	Description
Definition	From the Definition list, select Pipeline script from SCM .
SCM	From the SCM list, select Git .
Repository URL	Enter the Repository URL as the path of your local Git repository in the format <code>ssh://git@<jenkins_server>/<path to itsm-on-premise-installer.git></code> . Example: <code>ssh://git@<Jenkins server host name>/home/git/Git_Repo/ITSM_REPO/itsm-on-premise-installer.git</code> .
Credentials	Enter the Git server credentials.

Script Path	Specify the script path. Example: pipeline/jenkinsfile/HELIX_FULL_STACK_UPGRADE.jenkinsfile
--------------------	---

7. Click **Apply** and then **Save**.

After the pipeline is created, make sure that the pipeline is selected from Jenkins home page.

8. Click **Build Now**.

The first build job fails because it needs to run the first time to load all the parameters of the pipeline script.

9. After the build job fails, select the pipeline name again from the Jenkins home page.

The **Build Now** option changes to **Build With Parameters**.

[Back to top](#)

Troubleshooting

Refer to the troubleshooting information if you encounter any of the following issues while setting up BMC Deployment Engine.

! Important

When logging a case with BMC Support for any issues that occur while setting up BMC Deployment Engine, we recommend you to include the log file and build.properties file to facilitate debugging the issues.

Issue symptom

During .pl script execution, you might receive the following message:

This system is not registered with an entitlement server. You can use subscription-manager to register.

Log: Query: Please, suggest should we disable subscription manager and proceed with setting up prereq for ITSM onprem deployment, Y(yes) or N(no)?:

Resolution

- If the Jenkins server is configured with a local repository from where the required packages can be downloaded, enter 'yes' and proceed.
- If the Jenkins server is registered with any subscription manager, enter 'no' and proceed. In this case, the required packages are retrieved online through the subscription manager.

Issue symptom

The installation script fails or aborts for some reason.

Resolution

1. Review the logs to check for errors.
2. Stop Jenkins by using the `systemctl` command.

3. Check if port 8080 is free by using `netstat -anp | grep 8080` command and then rerun automation script command.
4. Make sure to save log files.

Issue symptom

HELIX_ONPREM_DEPLOYMENT pipeline fails with the following scripts not permitted error.

```
method hudson.model.Run getLog
method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild
```

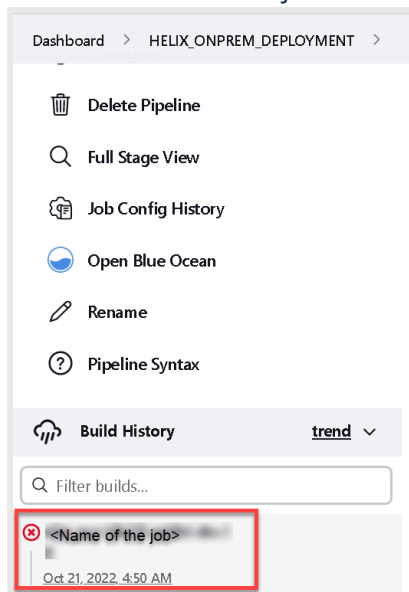
Resolution

1. Right click the error message link and open it in a new tab.
2. Click **Approve**.

You can also remove all previous script approvals: [Clear Approvals](#)

Approve / [Approve assuming permission check](#) / [Deny](#) signature: method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild **Approving**
 this signature may introduce a security vulnerability! You are advised to deny it.

3. Click the latest failed job on the HELIX_ONPREM_DEPLOYMENT pipeline tab.



Troubleshooting

Refer to the troubleshooting information if you encounter any of the following issues while setting up BMC Deployment Engine.

❗ Important

When logging a case with BMC Support for any issues that occur while setting up BMC Deployment Engine, we recommend you to include the log file and build.properties file to facilitate debugging the issues.

Issue symptom

During .pl script execution, you might receive the following message:

This system is not registered with an entitlement server. You can use subscription-manager to register.

Log: Query : Please, suggest should we disable subscription manager and proceed with setting up prereq for ITSM onprem deployment, Y(yes) or N(no) ? :

Resolution

- If the Jenkins server is configured with a local repository from where the required packages can be downloaded, enter 'yes' and proceed.
- If the Jenkins server is registered with any subscription manager, enter 'no' and proceed. In this case, the required packages are retrieved online through the subscription manager.

Issue symptom

The installation script fails or aborts for some reason.

Resolution

1. Review the logs to check for errors.
2. Stop Jenkins by using the `systemctl` command.
3. Check if port 8080 is free by using `netstat -anp | grep 8080` command and then rerun automation script command.
4. Make sure to save log files.

Issue symptom

HELIX_ONPREM_DEPLOYMENT pipeline fails with the following scripts not permitted error.

```
method hudson.model.Run getLog
method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild
```

Resolution

1. Right click the error message link and open it in a new tab.

2. Click **Approve**.

You can also remove all previous script approvals: [Clear Approvals](#)

Approve / Approve assuming permission check / Deny signature:method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild **Approving**
this signature may introduce a security vulnerability! You are advised to deny it.



3. Click the latest failed job on the HELIX_ONPREM_DEPLOYMENT pipeline tab.

Dashboard > HELIX_ONPREM_DEPLOYMENT >


- Delete Pipeline
- Full Stage View
- Job Config History
- Open Blue Ocean
- Rename
- Pipeline Syntax

Build History [trend](#) v




Filter builds...

 <Name of the job> 
 Oct 21, 2022, 4:50 AM

1. Click **Rebuild**.

 **Jenkins**

Dashboard > HELIX_ONPREM_DEPLOYMENT >

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build   
- Parameters
- Git Build Data
- Git Build Data
- Rebuild**
- Open Blue Ocean

2. Enter the required parameters and click **Rebuild**.

The screenshot shows a web interface for BMC Deployment Engine. At the top, there is a breadcrumb navigation bar: Dashboard > HELIX_ONPREM_DEPLOYMENT > <Name of job> > Rebuild. Below this, there are four input fields for configuration parameters: SIDE CAR FLUENT SPLUNK EVENT INDEX, SIDE CAR FLUENT SPLUNK EVENT FIELD, SIDE CAR FLUENT SPLUNK EVENT SOURCE, and SIDE CAR FLUENT SPLUNK EVENT SOURCETYPE. At the bottom of the form, there is a blue button labeled 'Rebuild', which is highlighted with a red rectangular box.

3. You might get other scripts not permitted error. Repeat the same steps to approve the script execution and rebuild HELIX_ONPREM_DEPLOYMENT jobs.

For other issues, see [Troubleshooting-deployment-pipeline-failure-issues](#).

For additional questions about BMC Deployment Engine, see [FAQ](#).

Where to go from here

Next task Proceed with [Installing-BMC-Helix-Platform-Common-Services-23-2-02](#).

Back to process If you are finished installing BMC Helix Platform services, return to the appropriate installation or upgrade process:

- [Installing](#)
- [Migrating-Remedy-on-premises-to-BMC-Helix-Service-Management-on-premises](#)