

# Kubernetes Continuous Deploy Plugin

## Plugin Information

View Kubernetes Continuous Deploy [on the plugin site](#) for more information.

A Jenkins plugin to deploy resource configurations to a Kubernetes cluster.

It provides the following features:

- Fetch the cluster credentials from the master node via SSH. You may also configure it manually.
- Variable substitution for the resource configurations, allowing you to do dynamic resource deployment.
- Docker login credentials management for the private Docker registry.
- No need to install the `kubectl` tool on the Jenkins slave nodes.

## Prerequisites

- A Kubernetes cluster.
- Kubernetes resource configurations to be deployed.

## Configure the Plugin

The screenshot shows the Jenkins configuration interface for the 'Deploy to Kubernetes' plugin. The main title is 'Deploy to Kubernetes'. Below it, there are several configuration fields and sections:

- Kubeconfig:** A dropdown menu currently set to '- none -' with an 'Add' button to its right.
- Config Files:** A text input field for specifying configuration file paths, accompanied by a help icon.
- Enable Variable Substitution in Config:** A checkbox that is checked, with a help icon.
- Deprecated Kubeconfig Settings...** A button to access deprecated settings.
- Docker Container Registry Credentials / Kubernetes Secrets:** A section with a sub-header and several fields:
  - Kubernetes Namespace for Secret:** A text input field containing 'default'.
  - Secret Name:** A text input field.
  - Docker Container Registry Credentials:** A section with a sub-header, a red 'X' close button, and a help icon.
    - Docker registry URL:** A text input field.
    - Registry credentials:** A dropdown menu set to '- none -' with an 'Add' button.
  - Add:** A button at the bottom left of the Docker section.
- Verify Configuration:** A button at the bottom right.

1. Within the Jenkins dashboard, select a Job and then select Configure
2. Scroll down and click the "Add build step" dropdown
3. Select "Deploy to Kubernetes"
4. In the "Kubeconfig" dropdown, select the `kubeconfig` stored in Jenkins. You can click the "Add" button on the right to add new kubeconfig (Kind: `Kubernetes configuration (kubeconfig)`).

You can choose one of the following methods to provide the kubeconfig details:

- Enter the kubeconfig content directly
  - Set the path to the kubeconfig on the Jenkins master
  - Fetch the kubeconfig from a remote SSH server
5. Fill in the "Config Files" with the configuration file paths. Split multiple entries with comma (,). [Ant glob syntax](#) is supported for path patterns.

- By checking "Enable Variable Substitution in Config", the variables (in the form of `$VARIABLE` or ``${VARIABLE}``) in the configuration files will be replaced with the values from corresponding environment variables before they are fed to the Kubernetes management API. This allows you to dynamically update the configurations according to each Jenkins task, for example, using the Jenkins build number as the image tag to be pulled.
- If your Kubernetes resources being deployed need to pull images from private registry, you can click the "Docker Container Registry Credentials / Kubernetes Secrets..." button and configure all the required registry credentials.

- Kubernetes Namespace for Secret:** the namespace in which the secrets will be created with the credentials configured below. By default this will be default.
- Secret Name:** the name of the secret that will be generated or updated if exists. If left blank, a unique name will be generated. The name will be exposed as environment variable `KUBERNETES_SECRET_NAME` and you may reference it in your configuration with the "Enable Variable Substitution in Config" option turned on.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: sample-k8s-deployment
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: sample-k8s-app
    spec:
      containers:
        - name: sample-k8s-app-container
          image: <username or registry URL>/<image_name>:<tag(maybe $BUILD_NUMBER)>
          ports:
            - containerPort: 8080
      imagePullSecrets:
        - name: $KUBERNETES_SECRET_NAME
```

- Docker Container Registry Credentials:** add one entry for each of the required private registry. If it is DockerHub, you may left the "Docker registry URL" as blank. Add or select the "Registry credentials" with type "Username with password".

You may also prepare the [Kubernetes Secrets](#) beforehand. and reference the secret from your resource configurations directly.

## Resource Types

The following resource types are supported by the plugin:

- Deployment
- Replica Set
- Replication Controller - No rolling-update support. If that's required, consider using [Deployment](#).
- Daemon Set
- Pod
- Job
- Service
- Ingress
- Secret - The plugin also provides secrets configuration.

In the context of continuous integration & continuous deployment, only those resources that need to be updated regularly should be placed in Jenkins deployment. So most of the time the plugin should mainly deal with resources of type Deployment.

## Pipeline Support

The `kubernetes-cd` plugin provides function `kubernetesDeploy` for [Jenkins Pipeline](#) support. You can go to the *Snippet Generator* page under *Pipeline Syntax* section in Jenkins, select "kubernetesDeploy: Deploy to Kubernetes" from the "Sample Step" dropdown, and it will provide you configuration interface for the plugin. After filling the entries and click "Generate Pipeline Script" button, you will get the sample scripts which can be used in your Pipeline definition.

```
kubernetesDeploy(kubeconfigId: 'kubeconfig-credentials-id', // REQUIRED

    configs: '<ant-glob-pattern-for-resource-config-paths>', // REQUIRED
    enableConfigSubstitution: false,

    secretNamespace: '<secret-namespace>',
    secretName: '<secret-name>',
    dockerCredentials: [
        [credentialsId: '<credentials-id-for-docker-hub>'],
        [credentialsId: '<credentials-id-for-other-private-registry>', url: '<registry-url>'],
    ]
)
```

The parameters can be divided into the following groups, which you may configure as required.

- Kubeconfig**
  - The credentials ID for the kubeconfig stored in Jenkins credentials store.
- Basic config for the deployments.**

```
kubernetesDeploy(
  ...
  configs: '<ant-glob-pattern-for-resource-config-paths>',
  enableConfigSubstitution: true,
  ...
)
```

- enableConfigSubstitution defaults to true
- Docker Container Registry Credentials / Kubernetes Secrets

```
kubernetesDeploy(
  ...
  secretNamespace: '<secret-namespace>',
  secretName: '<secret-name>',
  dockerCredentials: [
    [credentialsId: '<credentials-id-for-docker-hub>'],
    [credentialsId: '<credentials-id-for-other-private-registry>', url: '<registry-url>'],
  ],
)
```

- secretNamespace will be default if omitted.
- A unique secretName will be generated if omitted, and you need to reference it with variable\$KUBERNETES\_SECRET\_NAME in your resource configurations.

## Kubeconfig Binding

When the kubeconfig is prepared in the Jenkins credentials store, you can utilize the [Credentials Binding](#) plugin to bind the kubeconfig contents to a variable. You can save the contents in the variable to a local file, which can be used to invoke the `kubect1` command (with the `--kubeconfig` argument).

With pipeline:

```
withCredentials([kubeconfigContent(credentialsId: 'acs-ssh-folder', variable: 'KUBECONFIG_CONTENT')]) {
  sh '''echo "$KUBECONFIG_CONTENT" > kubeconfig && cat kubeconfig && rm kubeconfig'''
}
```

## Data/Telemetry

Kubernetes Continuous Deploy Plugin collects usage data and sends it to Microsoft to help improve our products and services. Read our [privacy statement](#) to learn more.

You can turn off usage data collection in Manage Jenkins -> Configure System -> Azure -> Help make Azure Jenkins plugins better by sending anonymous usage statistics to Azure Application Insights.

## Changelog

### Version 0.2.3, 2018-06-08

- Documentation and AI fix

### Version 0.2.2, 2018-05-18

- Support for namespace creation and update
- Fix EnvironmentInjector serialization (JENKINS-51147)

### **Version 0.2.1, 2018-04-20**

- Fix scoped SSH credentials lookup in kubeconfig credentials (#26)
- Fix Kubernetes deploy configuration verification (#29)
- Add support for ConfigMap (#30)
- Fix serialization of 3rd party exceptions thrown from slave (JENKINS-50760)

### **Version 0.2.0, 2018-04-03**

- Configure kubeconfig in the Jenkins credentials store instead of the job configuration (JENKINS-49781)  
  
The original "Kubernetes Cluster Credentials" configuration is deprecated.
- Upgrade Kubernetes Client to 3.1.10
- Use scoped credentials lookup (#19)

### **Version 0.1.5, 2018-02-22**

- Abort build on error (JENKINS-48662 / #12)
- Update Kubernetes Client to 3.1.7

### **Version 0.1.4, 2017-11-07**

- Fix master node SSH password login on Jenkins slave
- Add Third Party Notice

### **Version 0.1.3, 2017-10-10**

- Remove runtime licenses

### **Version 0.1.2, 2017-09-29**

- Fixed a stream closed issue when variable substitution is disabled

### **Version 0.1.1, 2017-09-28**

- Fixed an issue that plugin crashes on fastxml load

### **Version 0.1.0, 2017-09-27**

- Initial release