

Jenkins: CI/CD



Shailendra Chauhan

Microsoft MVP, Technical Consultant and Corporate Trainer

Jenkins

- An open source continuous integration tool written in Java. The project was forked from Hudson after a dispute with Oracle.
- Jenkins provides continuous integration services for software development.
- It is a server-based system running in a servlet container such as Apache Tomcat.
- An award-winning tool that monitors executions of repeated jobs, such as building a software project or jobs run by cron.

Jenkins with .NET

- Install MSBuild, MSTest, MSTestRunner, VSTestRunner, and Git plugin using Plugin manager.
- Jenkins has a Master/Slave architecture.
- You can add as many nodes as needed.
- Jenkins can be configured to run on the cloud like Azure, AWS etc.

```
> service jenkins restart  
- Browse: http://localhost:8080/restart
```

Jenkins Login



Welcome to Jenkins!

admin

.....

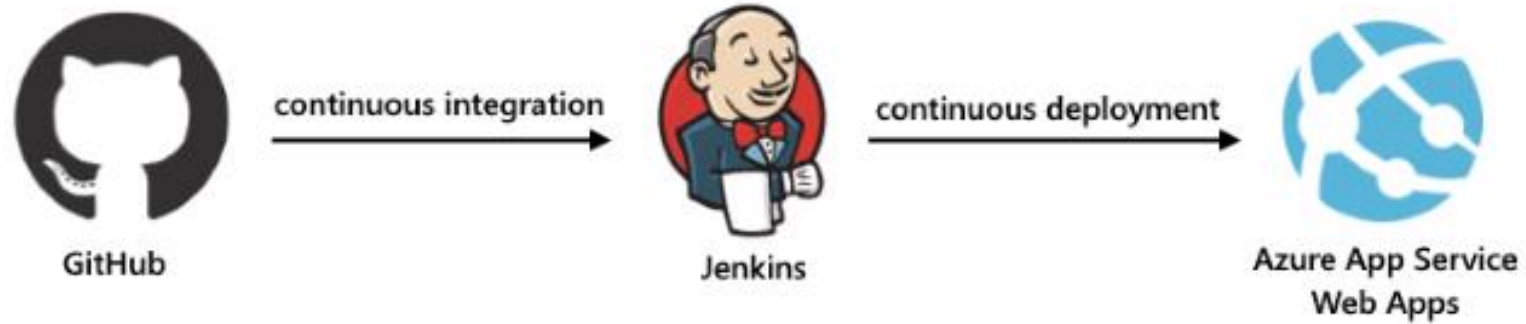
Sign in

☐ Keep me signed in

Jenkins Pipeline

```
pipeline {  
  agent any  
  environment {  
    dotnet = 'C:\\Program Files\\dotnet\\dotnet.exe'  
  }  
  stages {  
    stage('Checkout') {  
      steps {  
        git url: 'https://github.com/proshailendra/ASPNetCoreApp', branch: 'master'  
      }  
    }  
    stage('Build') {  
      steps {  
        bat 'dotnet build ASPNetCoreApp.sln --configuration Release'  
      }  
    }  
  }  
}
```

Jenkins CI/CD with Azure Web App



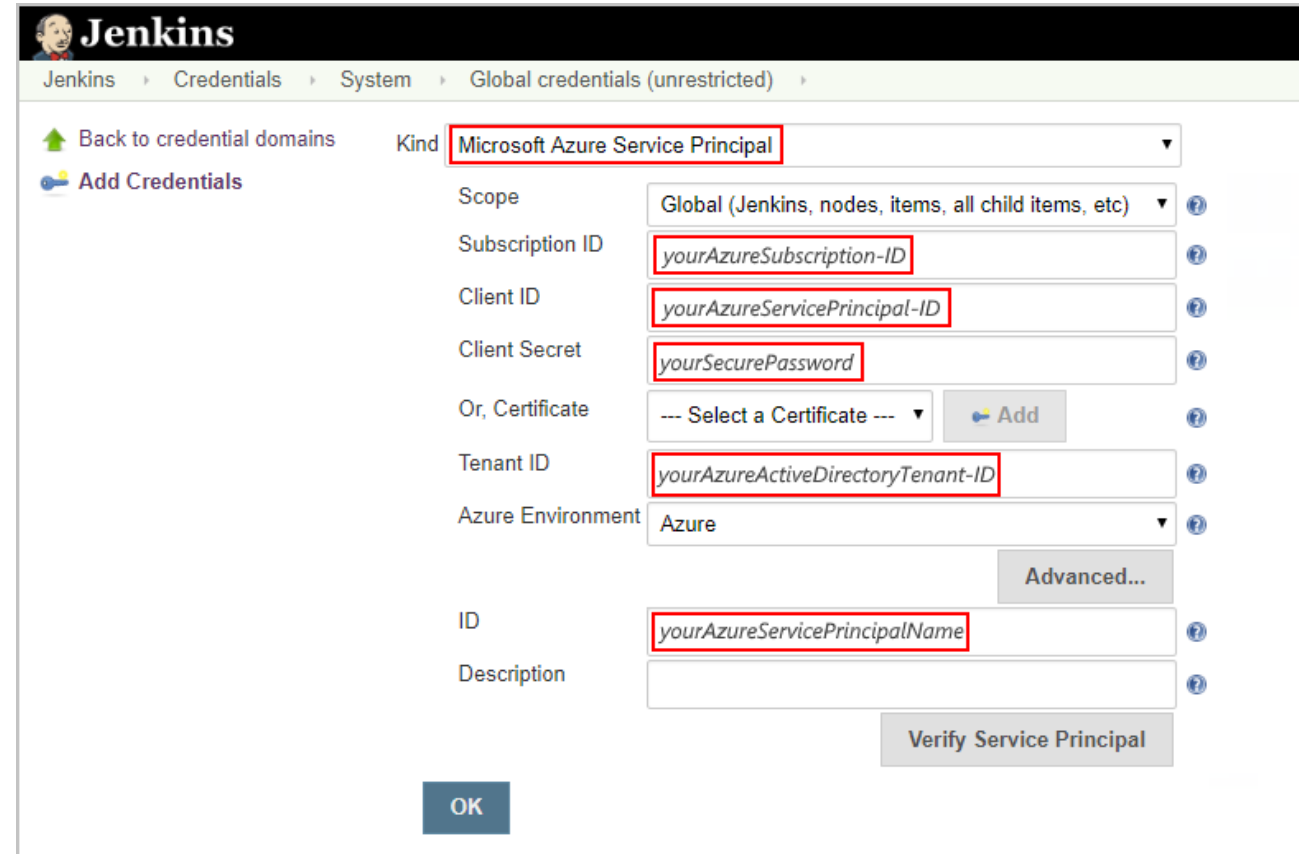
Jenkins CI/CD Plugins For Azure

- Azure Credentials
- Azure App Service
- Environment Injector
- Build Pipeline
- Jenkins Git client plugin

Setting Jenkins For Azure Deployment

```
> az ad sp create-for-rbac --name "webAppServicePrincipal"
```

```
{  
  "appId": "37cfd6cb-3535-4904-972e-af33682676d2",  
  "displayName": "webAppServicePrincipal",  
  "name": "http://webAppServicePrincipal",  
  "password": "25254d8f-7a51-4dc2-8a07-c2ebca78d111",  
  "tenant": "d95c2423-fd55-4495-8a6f-0a9283991dca"  
}
```



The screenshot shows the Jenkins web interface for adding a new credential. The breadcrumb trail is 'Jenkins > Credentials > System > Global credentials (unrestricted)'. The 'Kind' dropdown is set to 'Microsoft Azure Service Principal'. The 'Scope' is 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Subscription ID' field contains 'yourAzureSubscription-ID'. The 'Client ID' field contains 'yourAzureServicePrincipal-ID'. The 'Client Secret' field contains 'yourSecurePassword'. The 'Or, Certificate' dropdown is set to '--- Select a Certificate ---'. The 'Tenant ID' field contains 'yourAzureActiveDirectoryTenant-ID'. The 'Azure Environment' dropdown is set to 'Azure'. The 'ID' field contains 'yourAzureServicePrincipalName'. The 'Description' field is empty. There are 'Advanced...' and 'Verify Service Principal' buttons. An 'OK' button is at the bottom left.

Jenkins

Jenkins > Credentials > System > Global credentials (unrestricted)

Back to credential domains

Add Credentials

Kind: Microsoft Azure Service Principal

Scope: Global (Jenkins, nodes, items, all child items, etc)

Subscription ID: yourAzureSubscription-ID

Client ID: yourAzureServicePrincipal-ID

Client Secret: yourSecurePassword

Or, Certificate: --- Select a Certificate --- Add

Tenant ID: yourAzureActiveDirectoryTenant-ID

Azure Environment: Azure

Advanced...

ID: yourAzureServicePrincipalName

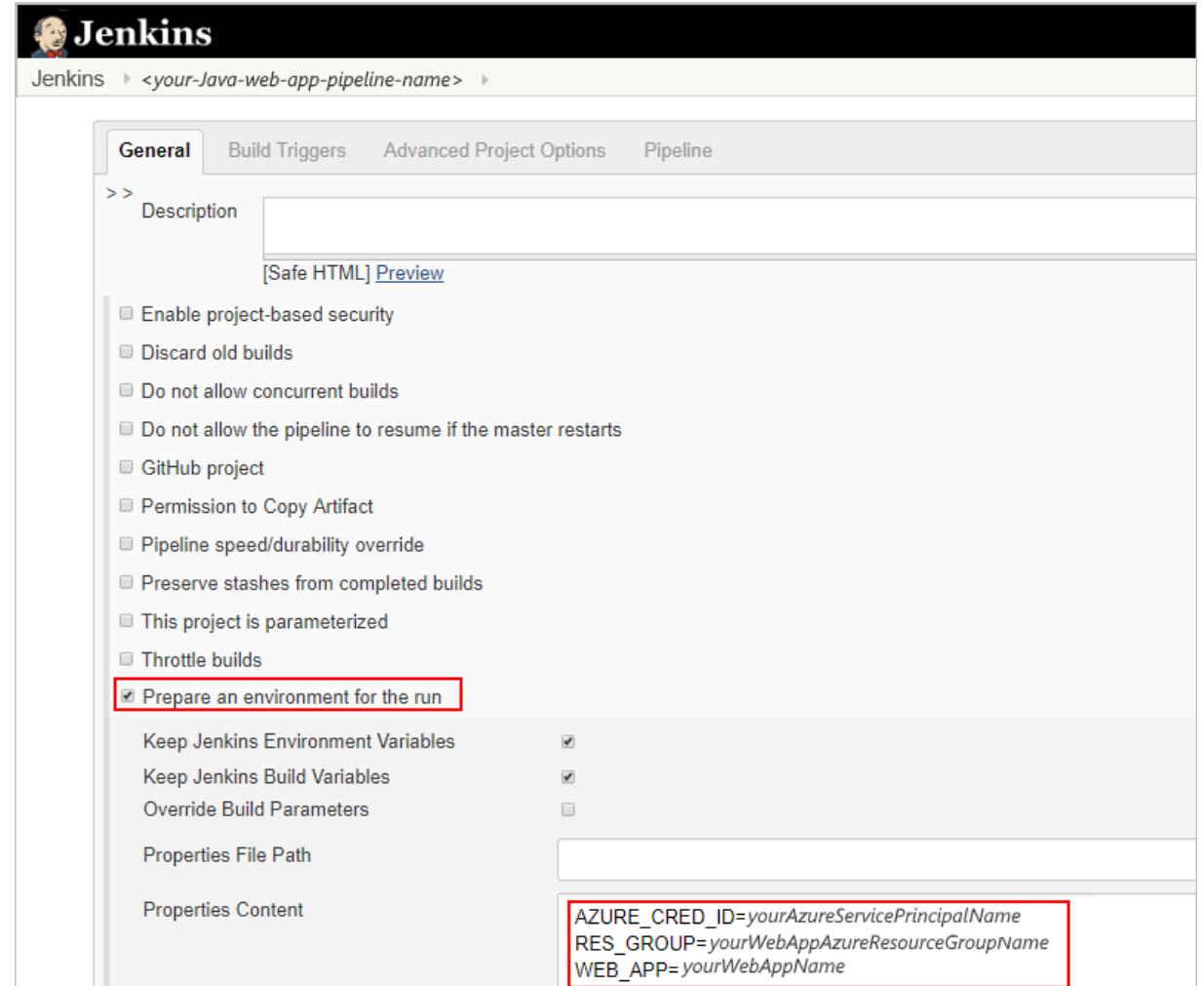
Description:

Verify Service Principal

OK

Defining Environment Variable

AZURE_CRED_ID=webAppServicePrincipal
RES_GROUP=TestDNTRG
WEB_APP=testdnt



The screenshot shows the Jenkins web interface for configuring a pipeline. The 'General' tab is selected, and the 'Prepare an environment for the run' checkbox is checked. Below this, the 'Keep Jenkins Environment Variables' checkbox is also checked. The 'Properties Content' field at the bottom contains the environment variable definitions: AZURE_CRED_ID=yourAzureServicePrincipalName, RES_GROUP=yourWebAppAzureResourceGroupName, and WEB_APP= yourWebAppName. Red boxes highlight the 'Prepare an environment for the run' checkbox, the 'Keep Jenkins Environment Variables' checkbox, and the 'Properties Content' field.

Jenkins > <your-Java-web-app-pipeline-name>

General Build Triggers Advanced Project Options Pipeline

> > Description

[Safe HTML] [Preview](#)

- ☐ Enable project-based security
- ☐ Discard old builds
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the master restarts
- ☐ GitHub project
- ☐ Permission to Copy Artifact
- ☐ Pipeline speed/durability override
- ☐ Preserve stashes from completed builds
- ☐ This project is parameterized
- ☐ Throttle builds
- ☒ Prepare an environment for the run

Keep Jenkins Environment Variables ☒

Keep Jenkins Build Variables ☒

Override Build Parameters ☐

Properties File Path

Properties Content

AZURE_CRED_ID=yourAzureServicePrincipalName
RES_GROUP=yourWebAppAzureResourceGroupName
WEB_APP= yourWebAppName

Setting Source Control Manager (SCM)

The screenshot shows the 'Advanced Project Options' tab in the Jenkins configuration interface. The 'Pipeline' section is active, showing the 'Definition' as 'Pipeline script from SCM'. The 'SCM' is set to 'Git'. Under 'Repositories', the 'Repository URL' is set to 'https://github.com/proshailendra/ASPNetCoreApp', and the 'Credentials' are set to 'proshailendra/***** (Github Connect)'. The 'Branches to build' section shows a 'Branch Specifier (blank for \'any\')' set to '*/master'. The 'Repository browser' is set to '(Auto)'. The 'Script Path' is set to 'Jenkinsfile'. The 'Lightweight checkout' checkbox is checked. There are several help icons (question marks) and buttons like 'Advanced...', 'Add Repository', 'Add Branch', and 'Add' throughout the form.

General Build Triggers **Advanced Project Options** Pipeline

Advanced Project Options

Advanced...

Pipeline

Definition

Pipeline script from SCM

SCM Git

Repositories

Repository URL https://github.com/proshailendra/ASPNetCoreApp

Credentials proshailendra/***** (Github Connect)

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any') */master

Add Branch

Repository browser (Auto)

Additional Behaviours Add

Script Path Jenkinsfile

Lightweight checkout ☒

Octopus

- Octopus integrates nicely with build server like Jenkins, Teamcity and Azure DevOps
- Octopus makes it easy to handle configuration across environments.
- Octopus uses the same deployment packages across environments and makes it easy to track their progress across those environments.