

1. Configure Webhook

forum-pipe

Pull requests Projects Security Insights **Settings**

updated.

General

Access

Collaborators

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Webhooks / Manage webhook

Settings Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

http://192.168.44.151:8080/github-webhook/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

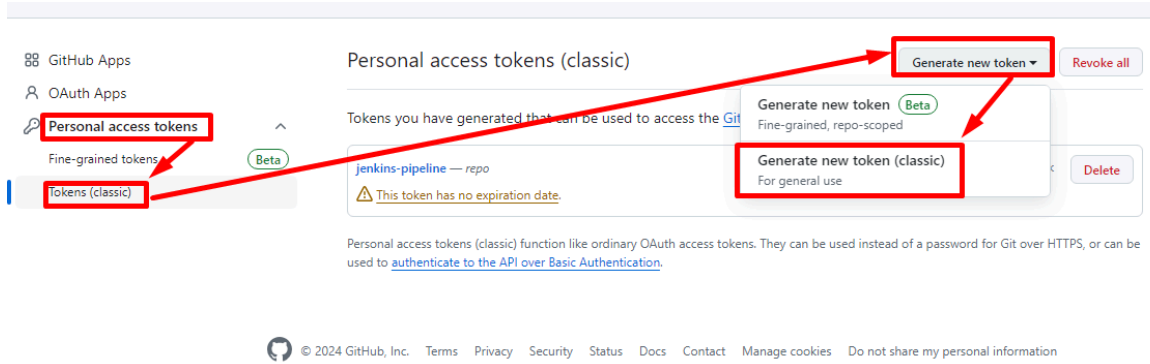
Update webhook Delete webhook

Payload URL: Will be Jenkins server url

Configure GiT Credentials:

Create a Classic Token:

Go to Your Profile>Settings>Developer Settings>Personal Access Token>Token (Classic)



GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

Personal access tokens (classic)

Tokens you have generated that can be used to access the Git

jenkins-pipeline — repo

This token has no expiration date.

Generate new token (Beta)

Generate new token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

cicd

What's this token for?

Expiration *

30 days The token will expire on Thu, Apr 25 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

Click **Generate token**.

[Note: Need to copy newly generated token for future use]

Configure Credentials in the Jenkins:

Go to > Manage Jenkins > Credentials>Global>Add Credentials>

New credentials

Kind

Username with password

Scope ?

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

Username ?

githubuser

☐ Treat username as secret ?

Password ?



github-token

ID ?

Description ?

Create

Credentials

T	P	Store	Domain	ID	Name
		System	(global)	a9a8575b-9023-42a2-9917-31d85ab13b77	iqbalkhan319/*****

Configure SMTP:

Go to > Manage Jenkins > system

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Advanced ^

 Edited

☒ Use SMTP Authentication ?

User Name

buprojecttime@gmail.com

 For security when using authentication it is recommended to enable either TLS or SSL

Password



Concealed

☐ Use SSL ?

☒ Use TLS

SMTP Port ?

587

Reply-To Address

Save

Apply

Connection between Jenkins and Deploy Server:

1. Login to the Jenkins server

```
#vipw
```

Change jenkins user shell from /bin/false to /bin/bash

```
root@ansible-master:~  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:99:99:Nobody:/:/sbin/nologin  
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin  
dbus:x:81:81:System message bus:/:/sbin/nologin  
polkitd:x:999:998:User for polkitd:/:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin  
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin  
iqbal:x:1000:1000:iqbal:/home/iqbal:/bin/bash  
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin  
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false  
nginx:x:998:996:nginx user:/var/cache/nginx:/sbin/nologin  
haproxy:x:188:188:haproxy:/var/lib/haproxy:/sbin/nologin  
ansible:x:1001:1001:/:/home/ansible:/bin/bash  
jenkins:x:997:995:Jenkins Automation Server:/var/lib/jenkins:/bin/bash  
~  
~  
~
```

```
#su - jenkins #switch to jenkins user's home directory  
# ssh-keygen -t rsa -b 2048 # Generate ssh key  
#cat .ssh/id_rsa.pub  
ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQCu3+ElawhWZdr55AXSLMktXhY90Gk165wi  
fLaPMqI+tKzNyCJbi1THd1aoGFdxYJmDV5p7OI1gz0+T/XE+l48eh+BIMXQ32fWqjJwQ  
sNuJ9DQ5xdNFSsJbRmkDAwV8K90zur7lptYDqhz5VeWAxFB39EmpzIJ5RrAGt/JivKngs  
fKZJNp940a7IP9u8+FG8wnQxGtMKZERX/cUo+e1pCRikWSd7iCZOPSSk8s4kxc5UN2C  
2Nd8/JcBleCWriO4JQ2ZHSMVKKzbAlm3YXQvRoDqgfkZ3ViVbHQFi9SNkqPCKnE/2ko  
e3imA6ZaPk02GJmdfgWetAn1aiKCev2ZFYn1 jenkins@ansible-master  
Or  
#ssh-copy-id deploy@deploy-server
```

Login to the deployment server:

```
#adduser deploy  
#su - deploy
```

```
#mkdir .ssh
#chmod 600 .ssh
#cd .ssh
#vi authorized_key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCu3+ElawhWZdr55AXSLMktXhY90Gk165wi
fLaPMql+tKzNyCJbi1THd1aoGFdxYJmDV5p7OI1gz0+T/XE+l48eh+BIMXQ32fWqjJwQ
sNuJ9DQ5xdNFSsJbRmkDAwV8K90zur7lptYDqhz5VeWAxFB39EmpzIJ5RrAGt/JivKngs
fKZJNp940a7IP9u8+FG8wnQxGtMKZErX/cUo+e1pCRikWSd7iCZOPSSk8s4kxc5UN2C
2Nd8/JcBleCWriO4JQ2ZHSMVKKzbAlm3YXQvRoDqgfkZ3ViVbHQFi9SNkqPCKnE/2ko
e3imA6ZaPk02GJmdfgWetAn1aiKCev2ZFYN1 jenkins@ansible-master
#chmod 400 authorized_key
```

```
#visudo
## Same thing without a password
deploy    ALL=(ALL)    NOPASSWD: ALL
```

Configure Build Server:

```
Install php and composer:
#yum install php php-mbstring php-bcmath php-xml php-sqlite3 php-curl -y
#cd /tmp/
#curl -sS https://getcomposer.org/installer | php
# mv composer.phar /usr/local/bin/composer
#composer --version
```

Switch to Jenkins user and do manual testing:

```
# su - jenkins
# cd workspace/forum-git

#cp .env_example .env
#composer install
#php artisan key:generate
#php artisan test
```

Create Jenkinsfile for full pipeline:

```
pipeline {
    agent any

    environment {
        APP_URL="http://forum.csl.com"

        APP_KEY="base64:EGCYLQuhy9xzmuPNnPvPNiXlpuiiOh0y4jl7N6/F5E0="
        CLIENT_APP_URL="http://forum.csl.com"
```

```

        APP_ENV="Production"
APP_DEBUG="False"
        DB_HOST="192.168.44.149"
DB_DATABASE="forum"
DB_USERNAME="root"
DB_PASSWORD=""Time@12345""

DEPLOY_PATH = "/var/www/html/forum"
        SSH_USER      = "deploy"
DEPLOY_SERVER = "192.168.44.210"
}

stages {
    stage('Checkout') {
        steps {
            // Checkout source code from Git repository
            git branch: 'master', credentialsId:
'a9a8575b-9023-42a2-9917-31d85ab13b77', url:
'https://github.com/lqbalkhan319/forum-pipe.git'
        }
    }
    stage('Change env') {
        steps {
            script {
                sh ""
                echo "Updating .env file with Jenkins environment variables..."
                cp .env.example .env
                sed -i "s|^DB_HOST=.*/DB_HOST=${DB_HOST}|" .env
                sed -i "s|^DB_DATABASE=.*/DB_DATABASE=${DB_DATABASE}|" .env
                sed -i "s|^DB_USERNAME=.*/DB_USERNAME=${DB_USERNAME}|" .env
                sed -i "s|^DB_PASSWORD=.*/DB_PASSWORD=${DB_PASSWORD}|" .env
                sed -i "s|^APP_URL=.*/APP_URL=${APP_URL}|" .env
                sed -i
"s|^CLIENT_APP_URL=.*/CLIENT_APP_URL=${CLIENT_APP_URL}|" .env
                sed -i "s|^APP_DEBUG=.*/APP_DEBUG=${APP_DEBUG}|" .env
                sed -i "s|^APP_KEY=.*/APP_KEY=${APP_KEY}|" .env
                sed -i "s|^APP_ENV=.*/APP_ENV=${APP_ENV}|" .env
                ""
            }
        }
    }
    stage('build') {
        steps {
            script {

```

```

        sh ""
        composer install --ignore-platform-reqs
        ""
    }
}

stage('PHPUnit Tests') {
    steps {
        // Run PHPUnit tests
        sh 'php artisan test'
    }
}

stage('Approval') {
    steps {
        input "Do you approve this build?"
    }
}

stage('Deploy') {
    steps {
        script {
            sh ""
            ssh ${SSH_USER}@${DEPLOY_SERVER} 'mkdir -p /var/www/html/forum
&& sudo chown -R deploy. /var/www/html/forum'
            rsync -avhP -e "ssh -o StrictHostKeyChecking=no" --exclude '.git/' .
            ${SSH_USER}@${DEPLOY_SERVER}:/var/www/html/forum
            ssh ${SSH_USER}@${DEPLOY_SERVER} <<'EOF'
            sudo chown -R www-data. /var/www/html/forum
            cd /var/www/html/forum
            sudo -u www-data composer install --ignore-platform-reqs
            sudo -u www-data php artisan config:clear
            sudo -u www-data php artisan cache:clear
            sudo -u www-data php artisan migrate
            << EOF
            ""
        }
    }
}

post {
    success {
        echo 'UAT deployment successful!'
    }
}

```



```

        mail to: 'iqbalkhan319@gmail.com',
        from: 'buprojecttime@brac.net',
        subject: "UAT Pipeline Success: ${currentBuild.displayName}",
        body: "UAT deployment successful for build ${env.BUILD_NUMBER}.\nView
console output at ${env.BUILD_URL}"
    }

    failure {
        echo 'UAT deployment failed!'
        emailx attachLog: true, // Attach the build log to the email
        to: 'iqbalkhan319@gmail.com',
        from: 'buprojecttime@brac.net',
        subject: "UAT Pipeline Failure: ${currentBuild.displayName}",
        body: "UAT deployment failed for build ${env.BUILD_NUMBER}.\nView
console output at ${env.BUILD_URL}"
    }
    always {
        // Clean up after the pipeline run
        cleanWs()
    }
}
}

```