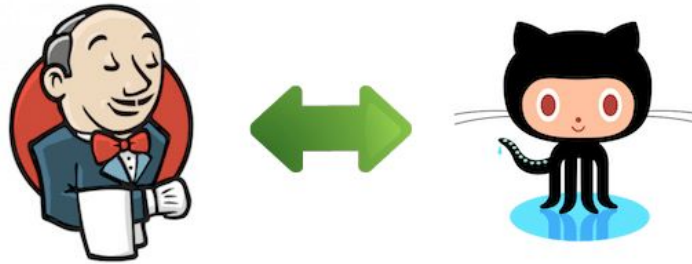


Ivan Danyliuk

**Jenkins, Ansible,
Terraform, AWS**



Jenkins + GitHub



Jenkins + Slack



Jenkins VM



<https://www.jenkins.io/doc/book/installing/linux/>

Local VirtualBox VM (Ubuntu 18 Server)

openssh server

Terraform

Ansible

aws cli

openjdk-11

Jenkins

ddclient

Dashboard [Jenkins] - Mozilla Firefox

Dashboard [Jenkins] x Slack | * devops | Home x +

jenkinsgeo.ddns.net:8080

Jenkins

Search

Dashboard

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Build Queue ^

No builds in the queue.

Build Executor Status ^

1	Idle
2	Idle

Jobs

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☁	DeployGeo	2 hr 49 min #22	2 hr 51 min #21	8 min 14 sec	▶
✓	⚙	DeployGeo_GitHub	2 hr 35 min #8	N/A	2 min 9 sec	▶
✓	⚙	DestroyGeo	2 min 15 sec #8	6 hr 41 min #2	1 min 41 sec	▶

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

REST API Jenkins 2.332.1

Ivan Danyliuk

Project structure

```
ubuntu@jenkins:~/Jenkins$ tree
.
├── ansible
│   ├── ansible.cfg
│   ├── app
│   │   └── deploy
│   ├── config
│   │   ├── pg_hba.conf
│   │   ├── postgresql.conf
│   │   └── tomcat.service
│   ├── group_vars
│   │   ├── app_server
│   │   └── db_server
│   ├── playbook1.yml
│   ├── playbook2.yml
│   └── playbook3.yml
├── config
│   ├── ATC.pem
│   ├── ddclient.conf
│   ├── hosts
│   └── hosts_geo
└── terraform
    ├── aws_Geo.tf
    ├── terraform.tfstate
    └── terraform.tfstate.backup
```

Ansible config file

```
[defaults]
host_key_checking = False
inventory          = ../config/hosts
```

Ansible group_vars files

```
---
ansible_user      : ubuntu
ansible_ssh_private_key_file : /home/ubuntu/Jenkins/config/ATC.pem
```

```
---
ansible_user      : ec2-user
ansible_ssh_private_key_file : /home/ubuntu/Jenkins/config/ATC.pem
```

Ansible inventory file, generated by terraform

```
[app_server]
app_host ansible_host=13.51.156.180
[db_server]
db_host ansible_host=13.49.73.246
```

Script file, generated by terraform, used by deploy script

```
#!/bin/bash

server_ip='13.51.156.180'
db_server_ip='13.49.73.246'
```

Ivan Danyliuk

Terraform awg_Geo.tf

```
1 provider "aws" {
2   shared_config_files = ["/home/ubuntu/Jenkins/.aws/config"]
3   shared_credentials_files = ["/home/ubuntu/Jenkins/.aws/credentials"]
4   profile = "Personal"
5   region = "eu-north-1"
6 }
7 resource "aws_instance" "App_Ubuntu_Terraform" {
8   ami = "ami-000e50175c5f86214"
9   instance_type = "t3.micro"
10  key_name = "ATC"
11  vpc_security_group_ids = [aws_security_group.sg_app.id]
12  credit_specification {
13    cpu_credits = "standard"
14  }
15  tags = {
16    Name = "App_Ubuntu_Terraform"
17    Owner = "Ivan Danyliuk"
18    Project = "Geocitizen"
19  }
20 }
21 > resource "aws_instance" "DB_Amazon_Linux_Terraform" { ...
34 }
35
36 > resource "aws_security_group" "sg_app" { ...
60 }
61 > resource "aws_security_group" "sg_db" { ...
85 }
86
```

```
87 #IP of aws instances copied to a file hosts file in local system
88 resource "local_file" "hosts_file" {
89   content = <<EOT
90   [app_server]
91   app_host ansible_host=${aws_instance.App_Ubuntu_Terraform.public_ip}
92   [db_server]
93   db_host ansible_host=${aws_instance.DB_Amazon_Linux_Terraform.public_ip}
94   EOT
95   filename = "../config/hosts"
96 }
97
98 #IP of aws instances copied to a file hosts file in local system
99 resource "local_file" "hosts_file_ip" {
100   content = <<EOT
101   #!/bin/bash
102
103   server_ip='${aws_instance.App_Ubuntu_Terraform.public_ip}'
104   db_server_ip='${aws_instance.DB_Amazon_Linux_Terraform.public_ip}'
105   EOT
106   filename = "../config/hosts_geo"
107 }
```

Ivan Danyliuk

Ansible playbook playbook1.yml

```
1  ---
2  - name: Configure App Server
3    hosts: app_server
4    become: yes
5    become_method: sudo
6    remote_user: ubuntu
7
8    tasks:
9  >   - name: ping all servers...
11 >   - name: Update APT package manager repositories cache...
15
16 >   - name: Install OpenJDK Java...
20
21 >   - name: add tomcat group...
24
25 >   - name: add tomcat user...
31
32 >   - name: create /opt/tomcat directory...
37
38 >   - name: download & unarchive...
44
45 >   - name: Change ownership...
53
54 >   - name: Copy Tomcat service from local to remote...
59
60 >   - name: Start and enable Tomcat service...
66
67 >   - name: Install Maven...
71
```

Ansible playbook playbook2.yml

```
1  ---
2  - name: Configure Db Server
3    hosts: db_server
4    become: yes
5    become_method: sudo
6    remote_user: ec2-user
7    vars:
8      db_user: ansible
9      db_password: ansibleuser
10     db_name: ansible
11
12   tasks:
13 > - name: install postgres on RHEL based distros ...
25
26 > - name: "Find out if PostgreSQL is initialized" ...
30
31 > - name: "Initialize PostgreSQL" ...
34
35     - name: Copy pg_hba.conf from local to remote
36       copy:
37         force: yes
38         src: ./config/pg_hba.conf
39         dest: /var/lib/pgsql/data/
40         mode: 0600
41         owner: postgres
42         group: postgres
43
```

```
44     - name: Copy postgresql.conf from local to remote
45       copy:
46         force: yes
47         src: ./config/postgresql.conf
48         dest: /var/lib/pgsql/data/
49         mode: 0600
50         owner: postgres
51         group: postgres
52
53 > - name: Start and enable postgres services ...
57
58 > - name: Create first database ...
64
65 > - name: Create db user ...
72
73 > - name: "Grant db user access to app db" ...
82
```

Ansible playbook playbook3.yml

```
1  ---
2  - name: Deploy Application
3    hosts: app_server
4    become: yes
5    become_method: sudo
6    remote_user: ubuntu
7
8    tasks:
9      - name: Copy Deploy script from local to remote
10        copy:
11          force: yes
12          src: ./app/deploy
13          dest: "~"
14          mode: 0700
15          owner: ubuntu
16          group: ubuntu
17
18      - name: Copy hosts file from local to remote
19        copy:
20          force: yes
21          src: /home/ubuntu/Jenkins/config/hosts_geo
22          dest: "~"
23          mode: 0700
24          owner: ubuntu
25          group: ubuntu
26
```

```
27 > ...- name: Stop Tomcat service...
33
34   ...- name: Deploy Application
35     ...- shell: "~/deploy"
36     ...- # register: output
37
38     ...- # --debug:
39     ...- # ...var: output.stdout_lines...
40
41 > ...- name: Start Tomcat service...
47 ...
```

Jenkins Pipelines

Dashboard [Jenkins] - Mozilla Firefox

jenkinsgeo.ddns.net:8080

Jenkins

Search

Ivan Danylyuk log out

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

REST API Jenkins 2.332.1

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	DeployGeo	3 hr 16 min #22	3 hr 18 min #21	8 min 14 sec
✓	🔗	DeployGeo_GitHub	3 hr 2 min #8	N/A	2 min 9 sec
✓	🔗	DestroyGeo	29 min #8	7 hr 9 min #2	1 min 41 sec

Deploy Application to AWS

Auto Deploy Application to AWS
(starts after GitHub repository
has been changed)


Destroy created previously
EC2 Instances on AWS


Ivan Danyliuk

Jenkins Pipeline DeployGeo


```
1 node {  
2   withEnv(["TERRAFORM_FOLDER=/home/ubuntu/Jenkins/terraform",  
3     "ANSIBLE_FOLDER=/home/ubuntu/Jenkins/ansible"]){  
4  
5     stage('Terraform Init') {  
6       slackSend color: "good", message: "Terraform init ..."  
7       sh script: '''  
8         chdir ${TERRAFORM_FOLDER}  
9         terraform init  
10      '''  
11     }  
12     stage('Create AWS VMs with Terraform') {  
13       slackSend color: "good", message: "Terraform apply ..."  
14       sh script: '''  
15         chdir ${TERRAFORM_FOLDER}  
16         terraform apply --auto-approve  
17      '''  
18     }  
19  
20     stage('Configure App VM with Ansible') {  
21       slackSend color: "good", message: "Configure App VM with Ansible ..."  
22       sh script: '''  
23         chdir ${ANSIBLE_FOLDER}  
24         ansible-playbook playbook1.yml  
25      '''  
26     }  
27  
28     stage('Configure Db VM with Ansible') {  
29       slackSend color: "good", message: "Configure Db VM with Ansible ..."  
30       sh script: '''  
31         chdir ${ANSIBLE_FOLDER}  
32         ansible-playbook playbook2.yml  
33      '''  
34     }  
35  
36     stage('Deploy Geo Application with Ansible') {  
37       slackSend color: "good", message: "Deploy Geo Application with Ansible ..."  
38       sh script: '''  
39         chdir ${ANSIBLE_FOLDER}  
40         ansible-playbook playbook3.yml  
41      '''  
42     }  
43   }  
44 }
```


Jenkins Pipeline DeployGeo


 **Jenkins**


Ivan Danylyuk  log out


Dashboard > DeployGeo >


 Back to Dashboard


 Status


 Changes


 Build Now


 Configure


 Delete Pipeline

 Full Stage View


 Rename


 Pipeline Syntax


 Build History **trend** ^

 #22

Pipeline DeployGeo

 Add description



 Recent Changes


Stage View


Average stage times:
(Average full run time: ~9min 17s)

	Terraform Init	Create AWS VMs with Terraform	Configure App VM with Ansible	Configure Db VM with Ansible	Deploy Geo Application with Ansible
Average	5s	27s	1min 18s	55s	6min 27s
#22	5s	7s	1min 41s	1min 10s	5min 8s


Ivan Danyliuk


Jenkins Pipeline DeployGeo_GitHub


 **Jenkins**


Ivan Danylyuk  log out


Dashboard > DeployGeo_GitHub >


 Back to Dashboard


 Status


 Changes


 Build Now


 Configure


 Delete Pipeline


 Full Stage View

 Rename


 Pipeline Syntax

 GitHub Hook Log


 **Build History** trend ^

 **#8**
Mar 23, 2022, 6:26 PM

Pipeline DeployGeo_GitHub

 Add description

Disable Project

 Recent Changes

Stage View

Average stage times:
(Average full run time: ~2min 7s)

	Checkout	Terraform Init	Create AWS VMs with Terraform	Configure App VM with Ansible	Configure Db VM with Ansible	Deploy Geo Application with Ansible
	1s	6s	9s	36s	38s	35s
#8 Mar 23 20:26 1 commit	1s	6s	8s	36s	37s	38s
#7 Mar 23 19:22 1 commit	1s	7s	9s	39s	39s	36s

Ivan Danyliuk

Jenkins Pipeline DeployGeo_GitHub

The screenshot displays the Jenkins web interface. At the top, the Jenkins logo is on the left, a search bar in the center, and the user 'Ivan Danylyuk' with a 'log out' button on the right. Below the header, a breadcrumb trail shows 'Dashboard' and 'DeployGeo_GitHub'. The left sidebar contains a list of actions: 'Back to Dashboard', 'Status', 'Changes' (which is highlighted), 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', 'Pipeline Syntax', and 'GitHub Hook Log'. At the bottom of the sidebar, there is a 'Build History' section with a 'trend' link and a search bar for filtering builds. The main content area is titled 'Changes' and lists a series of build changes, each with a number, a timestamp, and a description of the change (e.g., 'Delete 3', 'Create 3', 'Delete 1', 'Create 1'). The changes are listed in descending order of time, with the most recent change at the top.

Jenkins Search Ivan Danylyuk log out

Dashboard › DeployGeo_GitHub ›

Back to Dashboard

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

GitHub Hook Log

Build History trend ^

Filter builds...

Changes

#8 (Mar 23, 2022, 6:26:00 PM)

1. Delete 3 — noreply / githubweb

#7 (Mar 23, 2022, 5:22:12 PM)

1. Create 3 — noreply / githubweb

#5 (Mar 23, 2022, 4:58:56 PM)

1. Delete 1 — noreply / githubweb

#4 (Mar 23, 2022, 4:31:09 PM)

1. Create 1 — noreply / githubweb

#3 (Mar 23, 2022, 3:37:40 PM)

1. Delete 1 — noreply / githubweb

#2 (Mar 23, 2022, 3:34:55 PM)

1. Create 1 — noreply / githubweb

✓ #8 Mar 23, 2022, 6:26 PM

Ivan Danyliuk

Jenkins Pipeline DeployGeo_GitHub

```
1 properties([pipelineTriggers([githubPush()])])
2
3 node {
4   withEnv(["TERRAFORM_FOLDER=/home/ubuntu/Jenkins/terraform",
5     "ANSIBLE_FOLDER=/home/ubuntu/Jenkins/ansible"]){
6     stage('Checkout'){
7       slackSend color: "good", message: "Project GeoApp build process started (GitHub WebHook ...)"
8       git branch: 'main', url: 'https://github.com/idanylyuk/Geocit134.git'
9     }
10    stage('Terraform Init') {
11      slackSend color: "good", message: "Terraform init ..."
12      sh script: '''
13        chdir ${TERRAFORM_FOLDER}
14        terraform init
15      '''
16    }
17    stage('Create AWS VMs with Terraform') {
18      slackSend color: "good", message: "Terraform apply ..."
19      sh script: '''
20        chdir ${TERRAFORM_FOLDER}
21        terraform apply --auto-approve
22      '''
23    }
24
25    stage('Configure App VM with Ansible') {
26      slackSend color: "good", message: "Configure App VM with Ansible ..."
27      sh script: '''
28        chdir ${ANSIBLE_FOLDER}
29        ansible-playbook playbook1.yml
30      '''
31    }
32
33    stage('Configure Db VM with Ansible') { }
34
35    stage('Deploy Geo Application with Ansible') { }
36  }
37 }
```

1. PUSH events:
GitHub hook trigger for GITScm polling
by GitHub plugin

Build Triggers

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Disable this project ?
- ☐ Quiet period ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

2. The job has to be executed manually one time
in order for the push trigger and the git repo to
be registered

Ivan Danyliuk

Add GitHub Webhook to inform Jenkins

idanylyuk / **Geocit134** Public
forked from mentorchita/Geocit134

<> Code Pull requests Actions Projects Wiki Security Insights **Settings**

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Pages

Security

Code security and analysis

Deploy keys

Secrets

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://jenkinsgeo.ddns.net: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

Ivan Banyliuk

Jenkins Pipeline DestroyGeo

The screenshot shows the Jenkins web interface for a pipeline named "DestroyGeo". The top navigation bar includes the Jenkins logo, a search bar, and the user "Ivan Danylyuk" with a "log out" button. The left sidebar contains navigation links: "Back to Dashboard", "Status", "Changes", "Build Now", "Configure", "Delete Pipeline", "Full Stage View", "Rename", and "Pipeline Syntax". The main content area is titled "Pipeline DestroyGeo" and includes an "Add description" link and a "Disable Project" button. Below this is the "Recent Changes" section. The "Stage View" section displays a table of stages for the "Delete AWS VMs with Terraform" stage, showing average stage times and full run times. The "Build History" section shows a list of builds, with the most recent build (#8) completed on Mar 23, 2022, at 8:59 PM.

Jenkins Pipeline DestroyGeo

Dashboard ▸ DestroyGeo ▸

Back to Dashboard

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Pipeline DestroyGeo

Add description

Disable Project

Recent Changes

Stage View

Average stage times:		
(Average full run time: ~1min 28s)		
#	Time	Changes
#8	Mar 23 22:59	No Changes
#7	Mar 23 20:07	No Changes

1min 8s

1min 41s

2min 13s

Build History trend ^

Filter builds...

#8 Mar 23, 2022, 8:59 PM

#7

Script ?

```
1 node {  
2   stage('Delete AWS VMs with Terraform') {  
3     slackSend color: "good", message: "Destroy AWS instances ..."  
4     sh script: '''  
5       chdir /home/ubuntu/Jenkins/terraform  
6       terraform destroy --auto-approve  
7     '''  
8   }  
9 }
```

Ivan Danyliuk

Jenkins Slack Integration

1. Slack Notification

Provides Jenkins notification integration with Slack or Slack compatible applications

<https://plugins.jenkins.io/slack/#documentation>

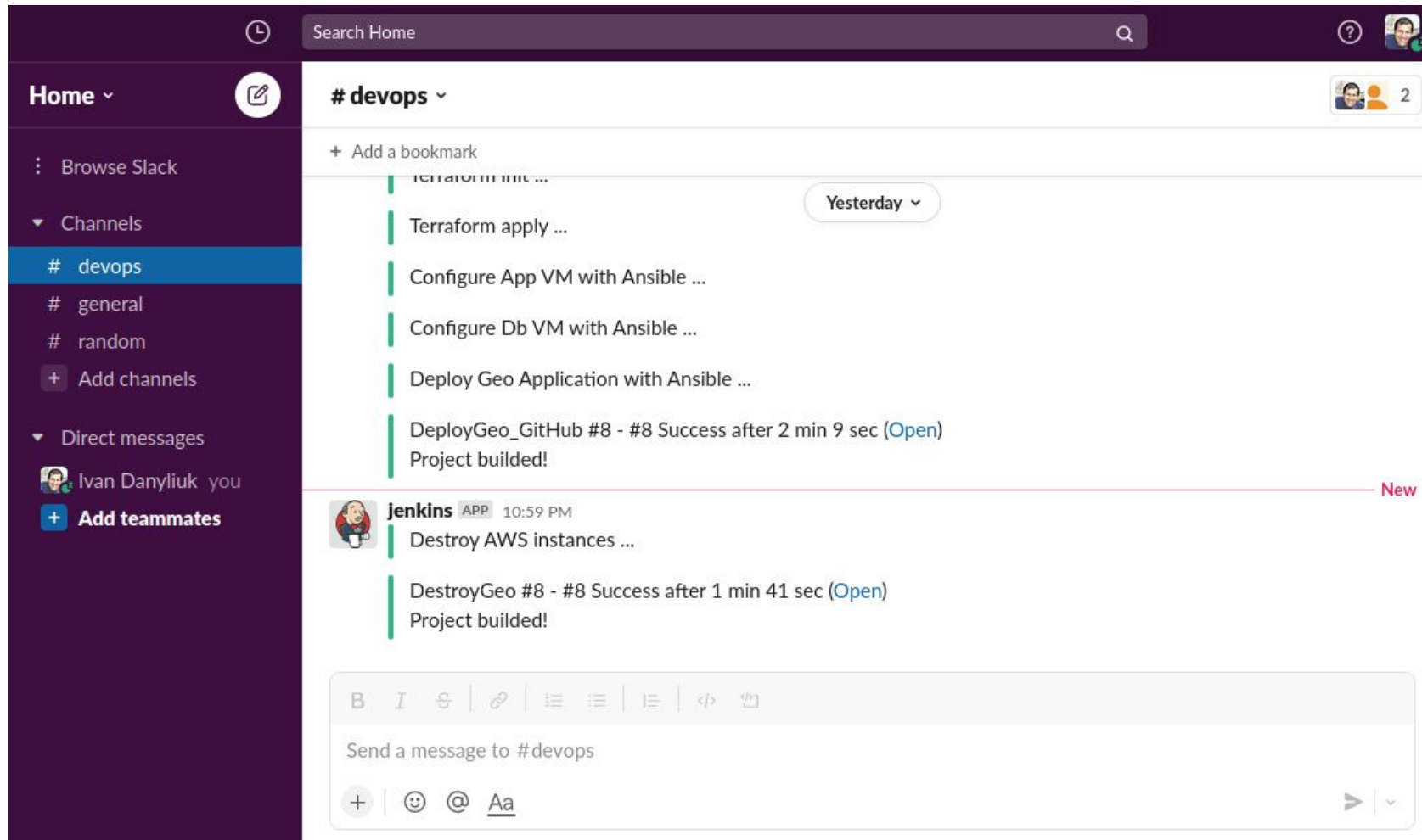
```
1 node {  
2   withEnv(["TERRAFORM_FOLDER=/home/ubuntu/Jenkins/terraform",  
3     "ANSIBLE_FOLDER=/home/ubuntu/Jenkins/ansible"]){  
4  
5     stage('Terraform Init') {  
6       slackSend color: "good", message: "Terraform init ..."  
7       sh script: '''  
8         chdir ${TERRAFORM_FOLDER}  
9         terraform init  
10      '''  
11    }  
}
```

2. Global Slack Notifier

This plugin adds the function to **automatically notify completion to Slack to all jobs.**

<https://plugins.jenkins.io/global-slack-notifier/>

Jenkins Slack Integration



Ivan Danyliuk

