

Infra CI/CD Jenkins 설정

Java 설치

1. Java 8 설치

```
sudo yum install -y java-1.8.0-openjdk-devel.x86_64
```

2. Java version 변경

```
sudo /usr/sbin/alternatives --config java  
2 + Enter
```

3. Java 7 삭제

```
sudo yum remove java-1.7.0-openjdk
```

4. Java 8 확인

```
java -version
```

Jenkins 설치

1. yum update

```
sudo yum update
```

2. Jenkins 다운로드

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

3. Jenkins 저장소 키를 등록

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

4. Jenkins 설치

```
sudo yum install jenkins
```

5. Jenkins Password 코드를 설정

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Nginx 설치

1. Nginx 설치

```
sudo yum install nginx
```

2. proxy pass 설정

```
sudo vim /etc/nginx/nginx.conf
```

```
proxy_pass http://localhost:8080;  
proxy_set_header X-Real-IP $remote_addr;  
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
proxy_set_header Host $http_host;
```

3. Nginx 실행

```
sudo service nginx start
```

Terraform 설치

```
sudo yum update  
wget  
https://releases.hashicorp.com/terraform/0.12.20/terraform_0.12.20_linux_amd64.zip  
unzip terraform_0.12.20_linux_amd64.zip  
sudo mv terraform /usr/bin/
```

Git 설치

```
sudo yum install git  
which git          # /usr/bin/git 이면 ok
```

Plugin 설치 목록

- [AWS CodePipeline Plugin](#)
- [Pipeline: AWS Steps](#)
- [Blue Ocean](#)
- [GitHub Authentication plugin](#)
- [Pipeline](#)
- [Terraform Plugin](#)
- [Deploy to container Plugin](#)

Global Tool Configuration

Git

Git installations



Git

Name

Default

Path to Git executable

git



Install automatically



Delete Git

Add Git ▼

Terraform

Terraform installations

Add Terraform



Terraform

Name

Terraform



Install automatically



Install from bintray.com

Version

Terraform 0.12.21 darwin (amd64) ▼

Delete Installer

Add Installer ▼

Delete Terraform

Add Terraform

List of Terraform installations on this system

계정 정보 설정


aws 계정 정보 설정

 [Back to Global credentials \(unrestricted\)](#)

 [Update](#)

 [Delete](#)

 [Move](#)

Scope 

ID 

Description 

Access Key ID 

Secret Access Key

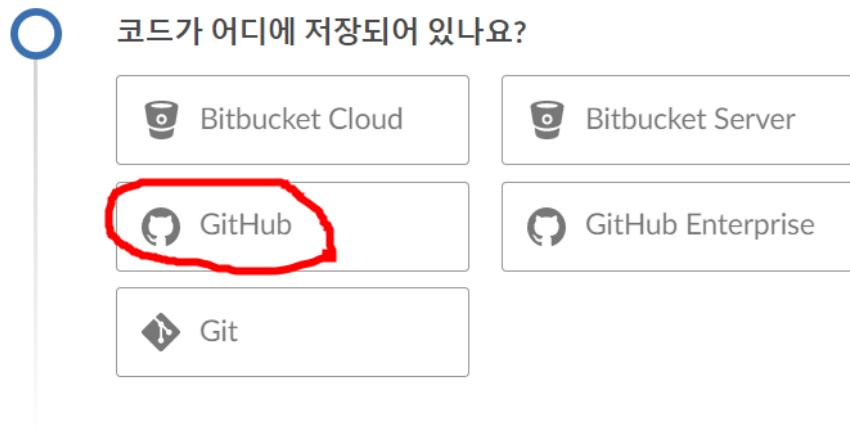
These credentials are valid and have access to 6 availability zones

IAM Role Support

[고급...](#)

[Save](#)

Blue ocean



계정 정보 설정

/var/lib/jenkins/workspace/ 에 다음과 같은 변수 값을 지닌 var.json파일을 미리 생성해 두어야 한다.

```
{  "aws_access_key" : "access key value",  "aws_secret_key" : "secret key value",  "db_password"    : "DB password",  "db_username"    : "DB master user name"}
```

CICD 수행 과정

1. Git clone



Git cloning - 1s

Git cloning 재시작

다운로드

2s

2s

```
1 using credential github
2 > git rev-parse --is-inside-work-tree # timeout=10
3 fetching changes from the remote git repository
4 > git config remote.origin.url https://github.com/cj905/practice.git # timeout=10
5 Cleaning workspace
6 > git rev-parse --verify HEAD # timeout=10
7 Resetting working tree
8 > git reset --hard # timeout=10
9 > git clean -fdx # timeout=10
10 Fetching upstream changes from https://github.com/cj905/practice.git
11 > git --version # timeout=10
12 using GIT_ASKPASS to set credentials GitHub Access Token
13 > git fetch --no-tags --progress -- https://github.com/cj905/practice.git +refs/heads/master:refs/remotes/origin/master # timeout=10
14 Checking out Revision 76da55f25a8a0074a82a6dc8d529d52b366fb4c (master)
15 > git config core.sparsecheckout # timeout=10
16 > git checkout -f 76da55f25a8a0074a82a6dc8d529d52b366fb4c # timeout=10
17 > git branch -a -v --no-abrev # timeout=10
18 > git branch -D master # timeout=10
19 > git checkout -b master 76da55f25a8a0074a82a6dc8d529d52b366fb4c # timeout=10
20 Commit message: "Added Jenkinsfile"
21 > git rev-list --no-walk 4ba07490804905da77b0b462d78c230f7f7f704 # timeout=10
22 Cleaning workspace
23 > git rev-parse --verify HEAD # timeout=10
24 Resetting working tree
25 > git reset --hard # timeout=10
26 > git clean -fdx # timeout=10
```

Git cloning 재시작

다운로드

```
1 Warning: CredentialId "cj905" could not be found.
2 > git rev-parse --is-inside-work-tree # timeout=10
3 fetching changes from the remote git repository
4 > git config remote.origin.url https://github.com/cj905/practice.git # timeout=10
5 Fetching upstream changes from https://github.com/cj905/practice.git
6 > git --version # timeout=10
7 > git fetch --no-tags --progress -- https://github.com/cj905/practice.git +refs/heads/*:refs/remotes/origin/* # timeout=10
8 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
9 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
10 Checking out Revision 76da55f25a8a0074a82a6dc8d529d52b366fb4c (refs/remotes/origin/master)
11 > git config core.sparsecheckout # timeout=10
12 > git checkout -f 76da55f25a8a0074a82a6dc8d529d52b366fb4c # timeout=10
13 > git branch -a -v --no-abrev # timeout=10
14 > git branch -D master # timeout=10
15 > git checkout -b master 76da55f25a8a0074a82a6dc8d529d52b366fb4c # timeout=10
16 Commit message: "Added Jenkinsfile"
```

2. terraform init



Init - 1s

Init 재시작

다운로드

1s

```
1 # cd /var/lib/jenkins/workspace
2 # terraform init -lock=false -var-file=var-json /var/lib/jenkins/workspace/practice_master
3
4 Initializing the backend...
5
6 Initializing provider plugins...
7
8 The following providers do not have any version constraints in configuration,
9 so the latest version was installed:
10
11 To prevent automatic upgrades to new major versions that may contain breaking
12 changes, it is recommended to add version = "x.y.z" constraints to the
13 corresponding provider blocks in configuration, with the constraint strings
14 suggested below.
15
16 * provider.aws: version = ">= 2.51"
17
18 Terraform has been successfully initialized!
19
20 You may now begin working with Terraform. Try running "terraform plan" to see
21 any changes that are required for your infrastructure. All Terraform commands
22 should now work.
23
24 If you ever set or change modules or backend configuration for Terraform,
25 rerun this command to reinitialize your working directory. If you forget, other
26 commands will detect it and remind you to do so if necessary.
```

3. terraform 실행



Apply - 10s

Apply 재시작

다운로드

5s

5s

```
1 # cd /var/lib/jenkins/workspace
2 # terraform plan -lock=false -var-file=var-json /var/lib/jenkins/workspace/practice_master
3
4 The refreshed state will be used to calculate this plan, but will not be
5 persisted to local or remote state storage.
6
7 aws_vpc.CRB52-vpc: Refreshing state... [id=vpc-0c2bbbc51d7b5410e]
8 aws_elb.CRB52-elb: Refreshing state... [id=elb-0ee86f9bbd65eb59]
9 aws_security_group.CRB52-security_group-private: Refreshing state... [id=sg-017b500e287079320]
10 aws_security_group.CRB52-security_group-public: Refreshing state... [id=sg-0f3f2d3ef9a8f6f67]
11 aws_lb_target_group.CRB52-U1: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:targetgroup/CRB52-U1/ad145ffb536590c1]
12 aws_subnet.CRB52-subnet-private-a: Refreshing state... [id=subnet-00f504875c1c3c02]
13 aws_subnet.CRB52-subnet-public-c: Refreshing state... [id=subnet-0bc8ecf50ca559485]
14 aws_lb_target_group.CRB52-API: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:targetgroup/CRB52-API/c86c75c183660bf]
15 aws_internet_gateway.CRB52-igw: Refreshing state... [id=igw-00009bc5db260db0a]
16 aws_subnet.CRB52-subnet-private-c: Refreshing state... [id=subnet-0eb7a3b961e052875]
17 aws_route_table.CRB52-route_table-public: Refreshing state... [id=rtb-02905a0528024d3da]
18 aws_security_group.rule.private-ingress-MySQL: Refreshing state... [id=sgrule-1596853271]
19 aws_instance.CRB52-private-c: Refreshing state... [id=i-0c1f759317a6dau1b]
20 aws_lb.CRB52-internal: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:loadbalancer/app/CRB52-internal/92926784371f558c]
21 aws_instance.CRB52-public-c: Refreshing state... [id=i-0e90c564c8cbef7d]
22 aws_security_group.rule.private-ingress-HTTP: Refreshing state... [id=sgrule-3384135306]
23 aws_security_group.rule.private-ingress-HTTPS: Refreshing state... [id=sgrule-3348441307]
24 aws_security_group.rule.private-egress-MySQL: Refreshing state... [id=sgrule-436783349]
25 aws_security_group.rule.public-egress-MySQL: Refreshing state... [id=sgrule-218485536]
26 aws_network_acl.CRB52-acl-private: Refreshing state... [id=acl-0a144e7cb650a7506]
27 aws_instance.CRB52-private-a: Refreshing state... [id=i-08b10a7c26a89cc5c]
28 aws_db_subnet_group.CRB52-rds-subnet-group: Refreshing state... [id=rds-rds-subnet-group]
29 aws_network_acl.CRB52-acl-public: Refreshing state... [id=acl-08624df6d61ba8f302]
30 aws_lb.CRB52-external: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:loadbalancer/app/CRB52-external/8c7f2ec80006dd93]
31 aws_instance.CRB52-public-a: Refreshing state... [id=i-01d1d3998013207b1]
32 aws_instance.crb5-bastion: Refreshing state... [id=i-04188f7bad79a3381]
33 aws_nat_gateway.CRB52-nat: Refreshing state... [id=nat-0b6fcd1f21b35a08]
34 aws_route_table_association.CRB52-route_table_associationpublic-a: Refreshing state... [id=rtbassoc-06ee79f239bc4035c]
35 aws_route_table_association.CRB52-route_table_associationpublic-c: Refreshing state... [id=rtbassoc-0ebcd38e08c97c7f0]
```

2. terraform apply

```
Apply - 10s
✓ > Shell Script 5s
✓ > Shell Script 5s

1 + cd /var/lib/jenkins/workspace
2 + terraform apply -auto-approve -lock=false -var-file=var-json /var/lib/jenkins/workspace/practice_master
3 aws_vpc.CRB52-vpc: Refreshing state... [id=vpc-0c2bbbc61d7b5410e]
4 aws_subnet.CRB52-subnet-public-c: Refreshing state... [id=subnet-0bc8ecf50ca559d85]
5 aws_internet_gateway.CRB52-igw: Refreshing state... [id=igw-000b98c5d8265e8a]
6 aws_security_group.CRB52-security-group-private: Refreshing state... [id=sg-017b500e287079320]
7 aws_subnet.CRB52-subnet-private-a: Refreshing state... [id=subnet-00f5bf7531bc3e2]
8 aws_subnet.CRB52-subnet-public-a: Refreshing state... [id=subnet-0c86b9daad74f3ac8]
9 aws_lb_target_group.CRB52-UI: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:targetgroup/CRB52-UI/8d145f7b53b590c1]
10 aws_security_group.CRB52-security-group-public: Refreshing state... [id=sg-0f3f2d3e9a8f6f67]
11 aws_subnet.CRB52-subnet-private-c: Refreshing state... [id=subnet-0eb7d3b961e052875]
12 aws_route_table.CRB52-route-table-public: Refreshing state... [id=rtb-02905a0528024d3da]
13 aws_nat_gateway.CRB52-nat: Refreshing state... [id=nat-00f6c1e0711b35d08]
14 aws_network_acl.CRB52-acl-public: Refreshing state... [id=acl-08624d6d61ba8f382]
15 aws_network_acl.CRB52-acl-private: Refreshing state... [id=acl-0a144e7c650a7506]
16 aws_db_subnet_group.CRB52-rds-subnet-group: Refreshing state... [id=crbs-rds-subnet-group]
17 aws_route_table_association.CRB52-route-table-association-public-a: Refreshing state... [id=rtaassoc-06ac79f239bc0035c]
18 aws_route_table_association.CRB52-route-table-association-public-c: Refreshing state... [id=rtaassoc-0ebcd38e08c97c7f0]
19 aws_instance.CRB52-public-a: Refreshing state... [id=i-01d13998013207b1]
20 aws_lb.CRB52-internal: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:loadbalancer/app/CRB52-internal/9292b784371f558c]
21 aws_lb.CRB52-external: Refreshing state... [id=arn:aws:elasticloadbalancing:ap-northeast-2:479011694316:loadbalancer/app/CRB52-external/8c7f6cc80006d93]
22 aws_instance.crbs-bastion: Refreshing state... [id=i-0a11807f0ad79a338]
23 aws_instance.CRB52-public-c: Refreshing state... [id=i-0e90c564c48cef7d]
24 aws_security_group_rule.private-ingress-HTTP: Refreshing state... [id=sgrule-3384135306]
25 aws_instance.CRB52-private-a: Refreshing state... [id=i-00b18a7c26a89cc5c]
26 aws_security_group_rule.private-ingress-HTTPS: Refreshing state... [id=sgrule-3348443307]
27 aws_instance.CRB52-private-c: Refreshing state... [id=i-0c1f759317e4ead12]
28 aws_security_group_rule.private-egress-MYSQL: Refreshing state... [id=sgrule-436751349]
29 aws_db_instance.CRB52-rds-instance: Refreshing state... [id=crbs-rds-instance]
30 aws_route_table.CRB52-route-table-private: Refreshing state... [id=rtb-0e72408e742f7b40]
31 aws_security_group_rule.public-egress-MYSQL: Refreshing state... [id=sgrule-2184805536]
32 aws_security_group_rule.private-ingress-MYSQL: Refreshing state... [id=sgrule-1596853271]
```

4. 생성된 리소스 확인

1. 생성된 리소스 결과값 받기
2. Git에서 받아온 shell 파일 workspace로 이동
3. shell 파일과 리소스 결과값으로 resource cli script 완성

Check - 9s

```
✓ > cd /var/lib/jenkins/workspace/terraform output > id.txtsed 's/ //g' id.txt > newid.txt — Shell Script

1 + cd /var/lib/jenkins/workspace/
2 + terraform output
3 + sed 's/ //g' id.txt

✓ > Shell Script

1 + cd /var/lib/jenkins/workspace/practice_master
2 + cp key.sh /var/lib/jenkins/workspace
3 + cp cli.sh /var/lib/jenkins/workspace

✓ > cd /var/lib/jenkins/workspacecat key.sh newid.txt > ids.shchmod +x ids.shcat ids.sh cli.sh > script.shchmod +x

1 + cd /var/lib/jenkins/workspace
2 + cat key.sh newid.txt
3 + chmod +x ids.sh
4 + cat ids.sh cli.sh
5 + chmod +x script.sh
```

4. script 결과물

✓ v cd /var/lib/jenkins/workspace/./script.sh — Shell Script

```
1 + cd /var/lib/jenkins/workspace/
2 + ./script.sh
3 [
4   [
5     [
6       {
7         "Value": "CRBS2-vpc",
8         "Key": "Name"
9       }
10    ]
11  ]
12 ]
13 [
14   [
15     "CRBS2-public-a"
16   ]
17 ]
18 [
19   [
20     "CRBS2-public-c"
21   ]
22 ]
23 [
24   [
25     "CRBS2-private-a"
26   ]
27 ]
28 [
29   [
30     "CRBS2-private-c"
31   ]
32 ]
33 [
34   [
35     "CRBS2-igw"
36   ]
37 ]
38 [
39   [
40     "CRBS2-nat"
41   ]
42 ]
43 ]
```