

DevOps - SQLABS

Project 1

Mark Kiezhner

All the code can be found here:

<https://github.com/MarkeyBass/todos-docker-compose.git>

● Create an IAM user that you will use for all of the AWS implementations.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Specify user details

User details

User name
OscarWilde

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Are you providing console access to a person?

☐ Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☐ Autogenerated password
You can view the password after you create the user.

☒ Custom password
Enter a custom password for the user:

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + = , @ _ - (hyphen) [] { } ' "

☐ Show password

☒ Users must create a new password at next sign-in (recommended).
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1050)

Choose one or more policies to attach to your new user.

Filter distributions by text, property or value 4 matches

[AdministratorAccess](#) Clear filters

Policy name	Type	Attached entities
<input type="checkbox"/> AdministratorAccess	AWS managed - job function	3

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name OscarWilde	Console password type Custom password	Require password reset Yes
-------------------------	--	-------------------------------

Permissions summary

Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional
Tag and key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 custom tags.

Cancel Previous **Create user**

User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Retrieve password

You can view and download the user's password below or email users instructions

Console sign-in details

Console sign-in URL
<https://504406221982.signin.aws.amazon.com/console>

User name
OscarWilde

Console password
***** [Show](#)

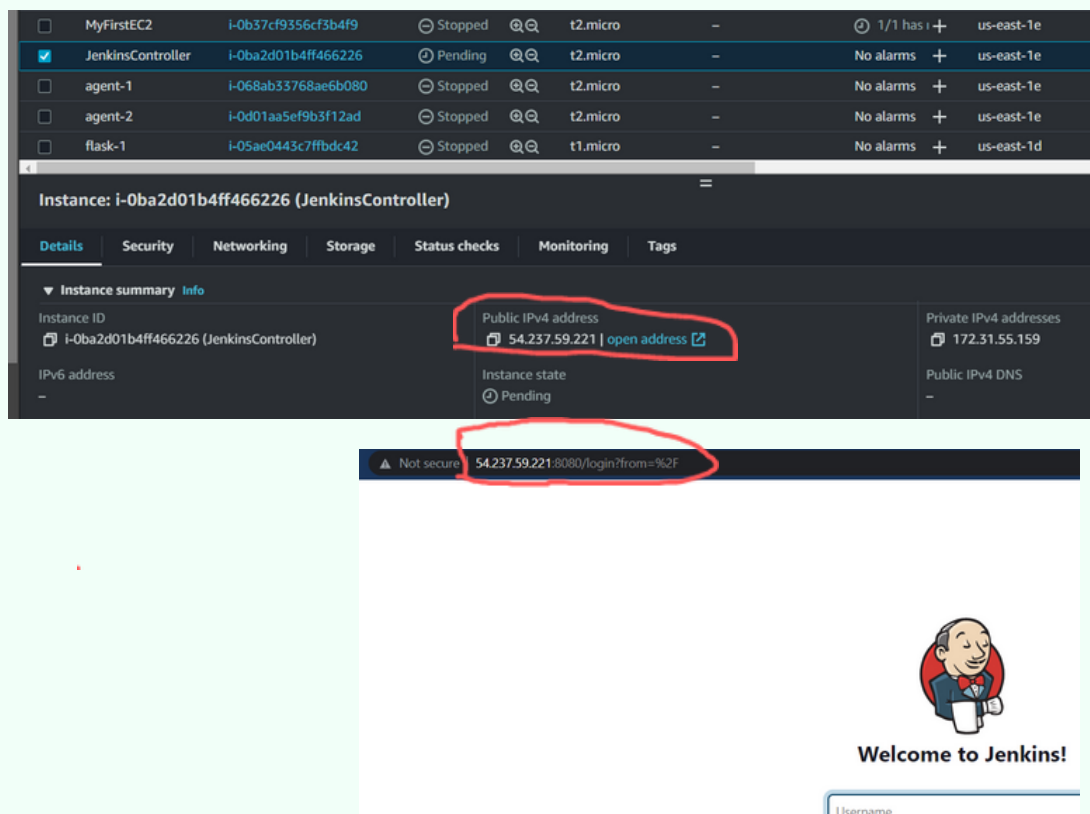
Permissions policies (2)

Permissions are defined by policies attached to the user directly or through groups.

< 1 >

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	Group SablesGroup
<input type="checkbox"/>	IAMUserChangePassword	AWS managed	Directly

- **Set up a Jenkins server inside a Docker container on an EC2 instance.**



- **Create three different users on your Jenkins server that will be on the same group. You will use only these users for all of the Jenkins implementations.**

JENKINS PLUGINS

Role-Based Authorization Strategy

---> download without restart of role plugin

--> Manage Jenkins

---> Configure Global Security

---> Authorization

----> Role-Based Strategy

Manage Jenkins

----> Manage and Assign Roles

----> Manage Roles ----> add new role

Manage Jenkins

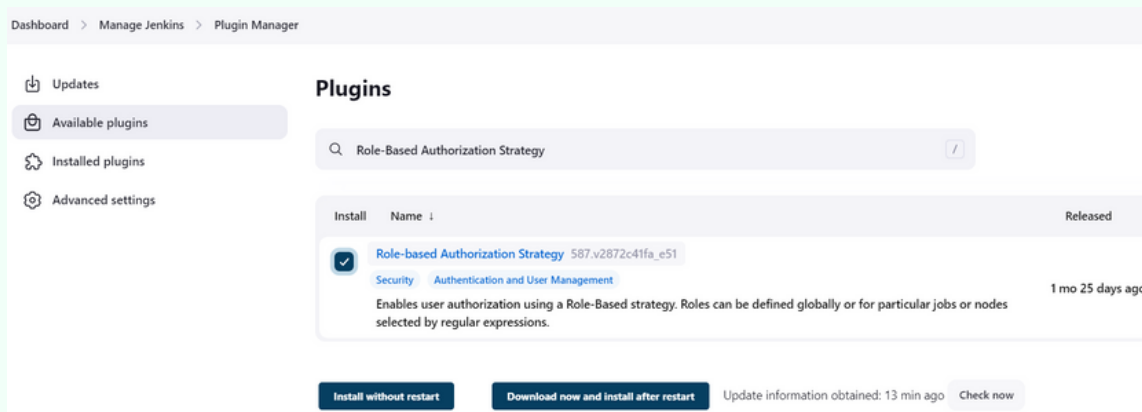
----> Jenkins' own user database

----> create user

----> Manage and Assign Roles

----> Assign Roles

----> User/group to add (Add the user)



Manage Jenkins > Jenkins' own user database > Create User

Create User

Username
MrA

Password

Confirm password

Full name
Mr A

E-mail address
markeybass@gmail.com

Jenkins' own user database > Create User

Create User

Username
MrB

Password

Confirm password

Full name
Mr B

E-mail address
markeybass@gmail.com

Create User

Jenkins' own user database > Create User

Create User

Username
MrC

Password

Confirm password

Full name
Mr C

E-mail address
markeybass@gmail.com

Dashboard > Manage Jenkins > Configure Global Security

Authentication

☐ Disable remember me

Security Realm
Jenkins' own user database

☐ Allow users to sign up ?

Authorization

Role-Based Strategy

Dashboard > Manage Jenkins > Manage and Assign Roles

Global roles

Role	Overall	Credentials	Agent	Job	Run	View	SCM
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AdminAccessUser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Build Queue
No builds in the queue.

Build Executor Status

User ID	Name
markeybass	Mark Kirzhner
MrA	Mr A
MrB	Mr B
MrC	Mr C

Assign Roles

Global roles

User/group	admin	AdminAccessUser
Mark Kirzhner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>
Mr A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mr B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mr C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

User/group to add ?
MrC

Role base Strategy didn't work. Mabe some BUG - so I Chaned Authorization to Matrix-based security - Now it works

Dashboard > Manage Jenkins > Configure Global Security

Matrix-based security

User/group	Overall	Credentials	Agent	Job	Run	View	SCM
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Mark Kirzhner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mr A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mr B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mr C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Markup Formatter
Plain text

- **Create a Jenkins pipeline job that pulls the code from the GitHub repository, builds the Docker image, and runs unit tests on the application.**

Dashboard > todos-test-and-deploy > Configuration

Configure

General | Advanced Project Options | Pipeline

General Enabled ☒

Description

Create a Jenkins pipeline job that pulls the code from the GitHub repository (git@github.com:MarkeyBass/todos-docker-compose.git), builds the Docker image, and runs unit tests on the application

The test result will be saved into a csv file with the following information the name of the user that ran the job the current date and status of the test.

The job will upload the text file with the results to the dedicated S3 bucket on AWS.

Configure the job to trigger automatically whenever a new commit is pushed to the Git repository.

[Plain text] [Preview](#)

Build Triggers

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

- ✓ Install Java on agent-1 and agent-2 EC2 machines
- sudo apt-get update
- sudo apt install default-jre

- ✓ Install Docker on agent-1 and agent-2 machines
- <https://docs.docker.com/engine/install/ubuntu/>

- ✓ docker check on both instances:
- sudo docker run hello-world

inside JenkinsController machine:

```
docker exec -it JenkinsController bash
ssh-keygen
```

```
cd /var/jenkins_home/.ssh
```

```
jenkins@32b8a66b46ff:~/.ssh$ cat id_rsa
```

```
-----BEGIN OPENSSH PRIVATE KEY-----
```

```
-----END OPENSSH PRIVATE KEY-----
```

add this private key to the configure credentials --> add this creds ID (controller-node) in the node (agent-1, agent-2, Prod-1-todos, Prod-2-todos) configuration

Dashboard > Manage Jenkins > Nodes > agent-1 > Configure

Build Executor Status

1 idle

Name ?

agent-1

Description ?

agent-1 of [JenkinsController](#)

[Plain text] [Preview](#)

Number of executors ?

1

Remote root directory ?

/home/ubuntu/jenkins-agent

Labels ?

agent-1 simple-rows

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agents via SSH

Host ?

172.31.56.239

Credentials ?

ubuntu (controller-node)

[Add](#)

Host Key Verification Strategy ?

Known hosts file Verification Strategy

[Advanced](#) [Edited](#)

Availability ?

Keep this agent online as much as possible

This part of the pipeline the agent connects to GitHub Downloads the repo Performs the tests.

```
pipeline {
  agent {label "agent-1"}

  stages {
    stage('CHECKOUT SCM') {
      steps {
        sshagent(credentials: ['controller-node']) {
          checkout([
            $class: 'GitSCM',
            branches: [[name: 'main']],
            userRemoteConfigs: [[
              url: 'git@github.com:MarkeyBass/todos-docker-compose.git',
              credentialsId: 'controller-node'
            ]]
          ])
        }
      }
    }

    stage('Build') {
      steps {
        sh 'sudo docker compose up -d'
      }
    }

    stage('Test') {
      steps {
        sh 'sudo docker compose exec server python test_server.py > test-results.txt 2>&1'
        sh 'cat test-results.txt'
        script {
          def fileContents = readFile(file: 'test-results.txt', encoding: 'UTF-8').trim()
          def lines = fileContents.split('\n')
          def test_statistics = lines[2].trim()
          def test_status = lines[4].trim()

          def testMap = [:]
          testMap['username'] = "${env.owner}"
          testMap['timestamp'] = new Date().getTime()
          testMap['datetime'] = new Date(testMap['timestamp']).toString()
          testMap['test_statistics'] = test_statistics
          testMap['test_status'] = test_status

          def jsonString = groovy.json.JsonOutput.toJson(testMap)
          writeFile file: 'test-results.json', text: jsonString

          println(jsonString)

          def csvString = "username,timestamp,datetime,test_statistics,test_status\n"
          csvString += "${testMap['username']},${testMap['timestamp']},${testMap['datetime']},${testMap['test_statistics']},${testMap['test_status']}\n"
          writeFile file: 'test-results.csv', text: csvString
        }
      }
    }
  }
}
```

The test result will be saved into a csv file with the following information the name of the user that ran the job the current date and status of the test.

- **The job will upload the text file with the results to the dedicated S3 bucket on AWS.**

Amazon S3 > Buckets > Create bucket

Create bucket [info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

☒ **jenkins-sqlabs-markk**

Creation date: February 16, 2023, 15:55:33 (UTC+02:00)

To remove the copied bucket settings, and restore the following configuration to the defaults settings, choose **Restore defaults**.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on **Block all public access**. These settings apply only to this bucket and its access points. *AWS recommends that you turn on **Block all public access**, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases.* [Learn more](#)

☐ **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☐ **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☐ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Upcoming permission changes to disable any Block Public Access setting

Starting in April 2023, to disable any Block Public Access setting when creating buckets by using the S3 console, you must have the `s3:PutBucketPublicAccessBlock` permission. [Learn more](#)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☒ Disable

☐ Enable

Tags (0) - optional

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Default encryption [info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption key type [info](#)

☒ Amazon S3 managed keys (SSE-S3)

☐ AWS Key Management Service key (SSE-KMS)

Bucket Key

When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. [Learn more](#)

☐ Disable

☒ Enable

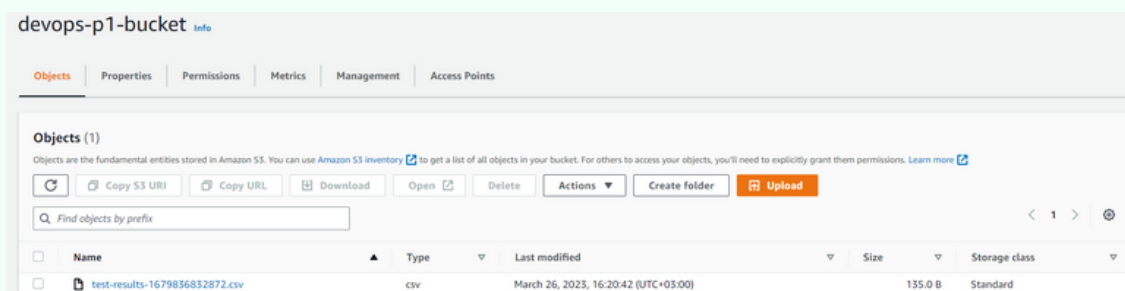
The stage of uploading to S3

(Instead of the text file I have uploaded the csv file)

```
stage('Upload Test in csv format to S3') {
  steps {
    script {
      def timestamp
      if (fileExists('test-results.csv')) {
        def fileContents = readFile(file: 'test-results.csv', encoding: 'UTF-8').trim()
        def lines = fileContents.split('\n')
        def csvData = lines[1].trim().split(',')
        timestamp = csvData[1].trim()
      } else {
        error('No test results file found')
      }
    }

    withAWS(credentials: 'awscredentials', region: 'us-east-1') {
      s3Upload(
        file: "test-results.csv",
        bucket: "devops-p1-bucket",
        path: "test-results-${timestamp}.csv"
      )
    }
  }
}
```

The result is in the bucket



- Configure the job to trigger automatically whenever a new commit is pushed to the Git repository

Dashboard > todos-test-and-deploy > Configuration

Configure

⚙️ General

🔑 Advanced Project Options

📄 Pipeline

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☒ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

github.com/MarkeyBass/todos-docker-compose/settings/hooks/new

Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub apps

Webhooks / Add webhook

We'll send a POST request to the URL below with details of a commit in the format you'd like to receive (JSON, x-www-form-urlencoded, or Atom). See the [documentation](#) for more details.

Payload URL *

Content type

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.

Add webhook

```
marke@DESKTOP-1P12AB4 MINGW64 /c/dev/DevOps/Projects/todos (main)
$ git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.43 KiB | 1.22 MiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:MarkeyBass/todos-docker-compose.git
 826d166..8ff7ef2  main -> main

marke@DESKTOP-1P12AB4 MINGW64 /c/dev/DevOps/Projects/todos (main)
$
```

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

GitHub Hook Log

Stage View

Average stage times:
(Average full run time: ~1min 0s)

	CHECKOUT SCM	Build	Test	Upload Test in csv format to S3	Declarative: Post Actions
#35 Mar 27 08:41 1 commit	5s	23s			
#34 Mar 26 16:19 No Changes	5s	36s	3s	8s	407ms
#33 Mar 26 16:15 No Changes					

Build History

trend

Filter builds...

#35
Mar 27, 2023, 5:41 AM

#34
Mar 26, 2023, 1:19 PM

#33
Mar 26, 2023, 1:15 PM

- Set up another EC2 instance to act as the production server.
- Create another EC2 production service.

Name and tags [info](#)

Name
Prod-1-todos [Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image)** [info](#)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or browse for AMIs if you don't see what you are looking for below.

Search your full catalog including 1000s of application and OS images

Recently **Quick Start**

Amazon Linux **aws** macOS **Mac** Ubuntu **ubuntu** Windows **Microsoft** Red Hat **Red Hat** S **S** [Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type [Free tier eligible](#)
ami-0557a15b87f6559cf (64-bit x86_64) / ami-0f9a090a2a642b (64-bit x86_64) Root device type: ebs

Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-02-08

Architecture **64-bit (x86)** AMI ID **ami-0557a15b87f6559cf** **Verified provider**

▼ **Instance type** [info](#)

Instance type
t2.small
Family: t2 1 vCPU 2 GiB Memory
On-Demand Linux pricing: 0.035 USD per Hour
On-Demand Linux pricing: 0.035 USD per Hour
On-Demand RHEL pricing: 0.083 USD per Hour
On-Demand SUSE pricing: 0.053 USD per Hour

[Compare instance types](#)

► **Key pair (login)** [info](#)
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

▼ **Network settings** [info](#) [Edit](#)

Network [info](#)
vpc-0424a6a9904e34a6d

Subnet [info](#)
No preference (Default subnet in all availability zones)

Auto-assign public IP [info](#)
Enable

Firewall (security groups) [info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Security groups [info](#)
Select security groups
AutoScaling-Security-Group-1 sg-0daffb3a49a5236d2 [X](#)
VPC: vpc-0424a6a9904e34a6d [Compare security group rules](#)

▼ **Configure storage** [info](#) [Advanced](#)

1x 16 GiB gp2 Root volume (Not encrypted)

[Free tier eligible customers can get up to 30 GiB of EBS General Purpose \(SSD\) or Magnetic storage](#) [X](#)

[Add new volume](#)

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems [Edit](#)

► **Advanced details** [info](#)

▼ **Summary**

Number of instances [info](#)
2

When launching more than 1 instance, consider [EC2 Auto Scaling](#).

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)
ami-0557a15b87f6559cf

Virtual server type (instance type)
t2.small

Firewall (security group)
AutoScaling-Security-Group-1

Storage (volumes)
1 volume(s) - 16 GiB

[Free tier](#): In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier

[Cancel](#) [Launch instance](#)

Instance ID	AMI	Instance type	Network	Subnet	Public IP	Private IP	Security groups	Key pair	Launch time	State	Tags
Prod-1-todos	i-06254d296e1e6f8e1	Pending	us-east-1c	t2.small	—	No alarms	+	us-east-1c	ec2-3-83-3-0-0-0		
Prod-2-todos	i-0574bdf79076bf77b	Pending	us-east-1c	t2.small	—	No alarms	+	us-east-1c	ec2-3-86-203-11		

Change security groups [info](#)

Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.

Instance details

Instance ID
i-0574bdf79076bf77b (Prod-2-todos)

Network interface ID
eni-0c7fbabdf84dcf186

Associated security groups
Add one or more security groups to the network interface. You can also remove security groups.

Search sg-0c71e220b685cc3c4 [X](#) [Add security group](#)

Security groups associated with the network interface (eni-0c7fbabdf84dcf186)

Security group name	Security group ID	
AutoScaling-Security-Group-1	sg-0daffb3a49a5236d2	Remove
launch-wizard-2	sg-0c71e220b685cc3c4	Remove

✓ Installing Java on prod-1-todos and prod-2-todos

```
sudo apt-get update
```

```
sudo apt install default-jre
```

✓ Installing Docker on prod-1-todos and prod-2-todos

<https://docs.docker.com/engine/install/ubuntu/>

Setting an agent connection between JenkinsController and prod-1-todos and prod-2-todos



Configuring agent:

Dashboard > Nodes >

Name ?
Prod-1-todos

Description ?
Production server for todos app
[Plain text] Preview

Number of executors ?
1

Remote root directory ?
/home/ubuntu/jenkins-agent

Labels ?
Prod-1-todos

Usage ?
Only build jobs with label expressions matching this node

Launch method ?
Launch agents via SSH

Host ?
172.31.91.118

Credentials ?
ubuntu (controller-node)
Add +

Host Key Verification Strategy ?
Known hosts file Verification Strategy

Advanced ▾

Availability ?
Keep this agent online as much as possible

Copy configuration to Prod-2-todos agent

Dashboard > Nodes > New node

New node

Node name
Prod-2-todos

Type

☐ Permanent Agent
Adds a plain, permanent agent to these agents, such as dynamic provisioning a physical computer, virtual machine, or container.

☒ Copy Existing Node
Prod-1-todos

Create

Change ip adress and Labels to Prod-2-todos agent

Dashboard > Nodes > Prod-2-todos > Configure

Host ?
172.31.93.92

Labels ?
Prod-2-todos

connect to prod-1-todos and prod-2-todos instances

add to .ssh/authorised_keys the Public Key of JenkinsController container

(!important - get it from inside the container)

Get it in advanced from the ---> docker exec -it JenkinsController bash -->

jenkins@32b8a66b46ff:~/.ssh\$ cat ~/.ssh/id_rsa.pub

```
ubuntu@ip-172-31-55-159:~$ sudo docker exec -it JenkinsController bash
jenkins@32b8a66b46ff:/$ cat ~/.ssh/id_rsa.pub~
cat: /var/jenkins_home/.ssh/id_rsa.pub~: No such file or directory
jenkins@32b8a66b46ff:/$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDWR0xewar16ucGCg9xX/aOBDBiyhoKJoq
je8i4jhs4bah/KCaoEsR0EfXCmLojUE/XIAOa7OnHKrhxnaPatyUeCZN201EzAtQ6fP0kN
rHGFTV8Ksx8B62kogWR0vDZSmkYwJWmuFq8/6DAQlAlU4fAx/5fKjU9hNLHgINCmZPm9FX2
```

```
Last login: Mon Mar 27 06:46:41 2023 from 2.52.151.17
ubuntu@ip-172-31-91-118:~$ cd ~/.ssh
ubuntu@ip-172-31-91-118:~/.ssh$ nano authorized_keys
ubuntu@ip-172-31-91-118:~/.ssh$ █

Last login: Mon Mar 27 06:47:37 2023 from 2.52.151.17
ubuntu@ip-172-31-93-92:~$ cd ~/.ssh
ubuntu@ip-172-31-93-92:~/.ssh$ nano authorized_keys
ubuntu@ip-172-31-93-92:~/.ssh$ █
```

THEN

---> connect from the Jenkins Container (JenkinsController) via ssh
to the agent (prod-1-todos and prod-2-todos) and leave a
fingerprint

Only now after leaving a fingerprint the JenkinsController will be
able to connect to its agent

```
jenkins@32b8a66b46ff:/$ ssh ubuntu@172.31.91.118
The authenticity of host '172.31.91.118 (172.31.91.118)' can't be established.
ECDSA key fingerprint is SHA256:ozKT7G1ONYAr1WmluiAsLErbfgKXgPB6egPG3dDy5d4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.91.118' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1028-aws x86_64)
```

```
jenkins@32b8a66b46ff:/$ ssh ubuntu@172.31.93.92
The authenticity of host '172.31.93.92 (172.31.93.92)' can't be established.
ECDSA key fingerprint is SHA256:JiHJIZb4OoBpG0BhL/IW3S2/q2TmPrE+Z4FhuJ6RYBA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.93.92' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1028-aws x86_64)
```

Launch the agents

Dashboard > Manage Jenkins > Nodes > Prod-1-todos > Log

```
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

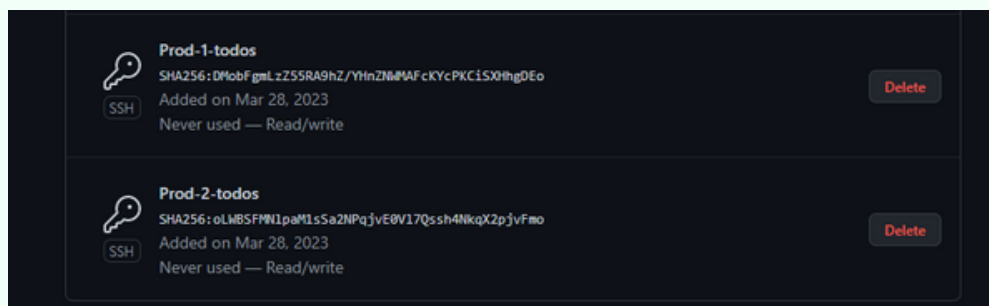
Dashboard > Manage Jenkins > Nodes > Prod-2-todos > Log

```
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

Adding prod-1-todos and prod-2-todos ssh pub keys to github

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ubuntu/.ssh/id_rsa  
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub  
The key's fingerprint is:  
SHA256:D0bfzGMLZ5SRA9HZ/YHzZWfKcPKCSl0HqEo ubuntu@ip-172-31-91-118  
The key's randormat Image is:  
  
+-----[RSA 3072]-----  
|  
|-E=+-+...+B.  
|BB.*B..+.O.  
|+..+O.+*..  
|. .+ O O ..  
|+ * S  
|  
|O O  
|  
|.  
|  
+-----[SHA256]-----  
ubuntu@ip-172-31-91-118:~$
```

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ubuntu/.ssh/id_rsa  
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub  
The key's fingerprint is:  
SHA256:oUBSFmNlpMtTsA2NpqJvEVL7QssH4kqCzpJvmo ubuntu@ip-172-31-93-92  
The key's randormat Image is:  
  
+-----[RSA 3072]-----  
|  
|dB==  
|..+B0.  
|+..+..  
|..+O..+..  
|..*+0 0 S  
|tB+..  
|O==+O  
|Eoo  
|O..+O  
+-----[SHA256]-----  
ubuntu@ip-172-31-93-92:~$
```



Adding GitHub to the agents known_hosts file:

```
ssh-keyscan github.com >> ~/.ssh/known_hosts
```

```

Last login: Tue Mar 28 09:25:09 2023 from 147.235.210.167
ubuntu@ip-172-31-91-118:~$ cd ~/.ssh
ubuntu@ip-172-31-91-118:~/.ssh$ ssh-keyscan github.com >> ~/.ssh/known_hosts
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
ubuntu@ip-172-31-91-118:~/.ssh$ cat known_hosts
cat: known_hosts: No such file or directory
ubuntu@ip-172-31-91-92:~$ ssh ls
authorized_keys id_rsa id_rsa.pub
ubuntu@ip-172-31-91-92:~$ ssh-keyscan github.com >> ~/.ssh/known_hosts
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
# github.com:22 SSH-2.0-babeld-9c57f613
ubuntu@ip-172-31-91-92:~$ ssh
```

- Upload the data from the S3 bucket csv file with the tests status into the DynamoDB service. And validate that you can see the collection with the whole results..

DynamoDB

>

Tables

>

Create table

Create table

Table details

info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

todos-test-results

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

id

Number

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name

String

1 to 255 characters and case sensitive.

Create item

You can add, remove, or edit the attributes of an item. [more](#)

Form

JSON view

Attributes

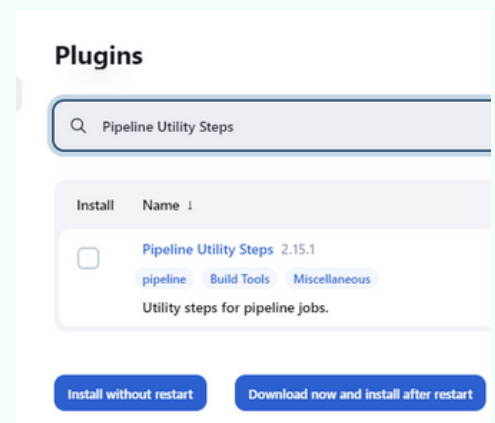
☒ View DynamoDB JSON

```
1 ▼ {
2   "Id": {
3     "N": "0"
4   },
5   "datetime": {
6     "S": ""
7   },
8   "test_statistics": {
9     "S": ""
10  },
11  "test_status": {
12    "S": ""
13  },
14  "timestamp": {
15    "S": ""
16  },
17  "username": {
18    "S": ""
19  }
20 }
```

Installing "Pipeline Utility Steps" plugin to allow parsing Map from JSON
(def props = readJSON text: jsonString)

Installing AWS CLI inside my agent (agent-1 EC2)

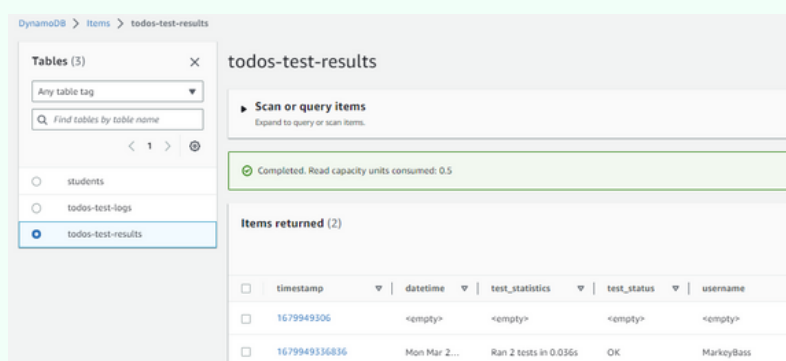
```
sudo apt-get update
sudo apt-get install awscli
aws --version
```



```
stage('Upload Test to DynamoDB') {
    steps {
        script {
            if (fileExists('test-results.csv')) {
                def testResultMap = readJSON file: './test-results.json'

                def timestamp = testResultMap['timestamp']
                def datetime = testResultMap['datetime']
                def username = testResultMap['username']
                def test_statistics = testResultMap['test_statistics']
                def test_status = testResultMap['test_status']

                withAWS(credentials: 'awscredentials', region: 'us-east-1') {
                    sh """
                        aws dynamodb put-item \
                        --table-name todos-test-results \
                        --item '{
                            \"timestamp\": {\"S\": \"${timestamp}\"},
                            \"datetime\": {\"S\": \"${datetime}\"},
                            \"username\": {\"S\": \"${username}\"},
                            \"test_statistics\": {\"S\": \"${test_statistics}\"},
                            \"test_status\": {\"S\": \"${test_status}\"}
                        }'
                    """
                }
            } else {
                error('No test results file found')
            }
        }
    }
}
```



	CHECKOUT SCM	Test	Upload Test in csv format to S3	Upload Test to DynamoDB	Declarative: Post Actions
Average stage times: (Average full run time: ~5s)	1s	2s	1s	942ms	689ms
<div>890</div> <div>Mar 27 23:38</div> <div>No Changes</div>	851ms	1s	370ms	1s	639ms

- Create a Jenkins job that deploys the Docker image to the production server.
- Configure the job to trigger automatically whenever a new test build is successfully tested

Added the trigger line to the success clause in the post clause:

```
post {  
  success {  
    build job: "${DEPLOY_JOB_ON_SUCCESS}", wait: false  
  }  
}
```

Create the deploy to prod job:

Dashboard > todos-deploy-to-prod > Configuration

Description

todos-deploy-to-prod job will be triggered by todos-test-and-deploy job only if todos-test-and-deploy finishes with success.
todos-deploy-to-prod will deploy the todos application on the production server.
The production server is one of the EC2 machines (Jenkins Agent nodes) Prod-1-todos or Prod-2-todos.

[Plain text] [Preview](#)

☒ Discard old builds ?

Strategy

Log Rotation ▼

Days to keep builds
if not empty, build records are only kept up to this number of days

Max # of builds to keep
if not empty, only up to this number of build records are kept

The deploy to prod job pipeline:

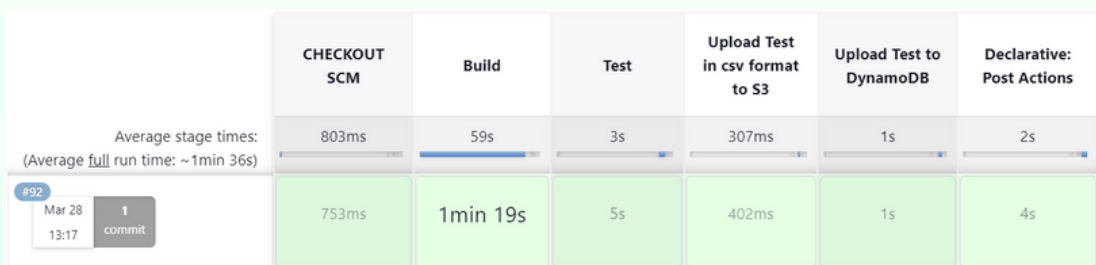
```
pipeline {
  agent {label 'Prod-1-todos'}
  environment {
    GIT_REPO_URL_SSH = 'git@github.com:MarkeyBass/todos-docker-compose.git'
  }

  stages {
    stage('CHECKOUT SCM') {
      steps {
        sshagent(credentials: ['controller-node']) {
          checkout([
            $class: 'GitSCM',
            branches: [[name: 'main']],
            userRemoteConfigs: [[
              url: "${GIT_REPO_URL_SSH}",
              credentialsId: 'controller-node'
            ]]
          ])
        }
      }
    }
    stage('Build') {
      steps {
        sh 'sudo docker compose -f docker-compose-prod.yml down'
        sh 'sudo docker compose -f docker-compose-prod.yml up -d'
      }
    }
  }
}
```

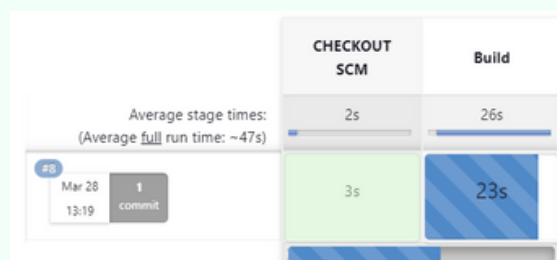
Pushing to github

```
marke@DESKTOP-1PI2AB4 MINGW64 /c/dev/DevOps/Projects/todos (main)
$ git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.20 KiB | 1.20 MiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:MarkeyBass/todos-docker-compose.git
83a3cb6..1a4f8ec main -> main
```

todos-test-and-deploy job triggered



todos-deploy-to-prod job triggered



Production app is running on port 80

Todos App

Not secure | 44.211.125.196

Update

TODO List!!!

Insert

userId	Title	Description	Date Time	Update	Update
1	1	1	2023-03-2...	Update	Delete

Previous

Page 1 of 1

5 rows

Next

● *Create a parameter in your Jenkins job that will allow for you to choose on which of the production services you want to deploy.*

○ *The first one*

○ *The second one*

○ *Both*

todos-deploy-to-prod pipeline modified

The parameters block defines the parameter and its available options

The input step inside 'Manually add agent lable' stage ensures that the parameter is set

```
def gitCheckoutSCM(gitRepoUrl) {
    sshagent(credentials: ['controller-node']) {
        checkout([
            $class: 'GitSCM',
            branches: [[name: 'main']],
            userRemoteConfigs: [[
                url: "${gitRepoUrl}",
                credentialsId: 'controller-node'
            ]]
        ])
    }
}

def deployToProd() {
    sh 'sudo docker compose -f docker-compose-prod.yml down'
    sh 'sudo docker compose -f docker-compose-prod.yml up -d'
}

pipeline {
    agent none

    environment {
        GIT_REPO_URL_SSH = 'git@github.com:MarkeyBass/todos-docker-compose.git'
    }

    parameters {
        choice(name: 'KEY', choices: ['ONE', 'TWO', 'BOTH'], description: 'Select on which agent to run the deploy')
    }

    stages {
        stage('Input Key') {
            when {
                expression { params.KEY == null }
            }
            steps {
                input message: 'Please select KEY value', parameters: [choice(name: 'KEY', choices: ['ONE', 'TWO', 'BOTH'], description: 'Select on which agent to run the deploy')]
            }
        }
        stage('Run on Prod-1-todos') {
            when {
                expression { params.KEY == 'ONE' || params.KEY == 'BOTH' }
            }
            agent {
                label 'Prod-1-todos'
            }
            steps {
                gitCheckoutSCM(GIT_REPO_URL_SSH)
                deployToProd()
            }
        }
        stage('Run on Prod-2-todos') {
            when {
                expression { params.KEY == 'TWO' || params.KEY == 'BOTH' }
            }
            agent {
                label 'Prod-2-todos'
            }
            steps {
                gitCheckoutSCM(GIT_REPO_URL_SSH)
                deployToProd()
            }
        }
    }
}
```

todos-test-and-deploy modified

added parameter when triggering the next job

```
build job: "${DEPLOY_JOB_ON_SUCCESS}", wait: false, parameters: [
    [$class: 'StringParameterValue', name: 'KEY', value: 'BOTH']
]
```

- Create a load balancer between your both production servers. Separate equally the traffic between both of these instances.

Creating a target group

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

Target group name

todos-prod-target-group

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Port

HTTP

:

80

VPC

Select the VPC with the instances that you want to include in the target group.

vpc-0424a6a9904e34a6d
IPv4: 172.31.0.0/16

Protocol version

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

Review targets

Targets (2)

Remove all pending

All

Filter resources by property or value

< 1 >

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
X	Pending	I-0574bdf79076bf77b	Prod-2-todos	80	running	AutoScaling-Security-Group-1, launch-wizard-2	us-east-1c	subnet-05a9689ef64285acb
X	Pending	I-06254d296e1e6ffe1	Prod-1-todos	80	running	AutoScaling-Security-Group-1, launch-wizard-2	us-east-1c	subnet-05a9689ef64285acb

2 pending

Cancel Previous Create target group

EC2 > Target groups

Target groups (1) Info

Actions

Create target group

Search or filter target groups

< 1 >

	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	todos-prod-target-group	arn:aws:elasticloadbalancing...	80	HTTP	Instance	None associated	vpc-0424a6a9904e34a6c

Creating a Load Balancer

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservice on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which applicable, it selects a target from the target group for the rule action.

► **How Elastic Load balancing works**

Basic configuration

Load balancer name
Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

▼ Listener HTTP:80 [Remove](#)

Protocol	Port	Default action	Info
HTTP ▼	80 1-65535	Forward to todos-prod-target-group Target type: Instance, IPv4	HTTP ▼ ↺

[Create target group](#) [↗](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Round Robin algorithm is configured - Separate equally the traffic between both of these instances.

EC2 > Target groups > todos-prod-target-group > Edit target group attributes

Edit target group attributes [Info](#)

[Restore defaults](#)

Target configuration

Deregistration delay
The time to wait for in-flight requests to complete while deregistering a target. During this time, the state of the target is draining.

seconds
0-3600

Traffic configuration

Slow start duration
During this period, a newly registered target receives an increasing share of requests. This attribute cannot be combined with the Least outstanding requests algorithm.

seconds
30-900 seconds or 0 to disable.

Load balancing algorithm
Determines how the load balancer selects targets from this target group when routing requests.

☒ Round robin
☐ Least outstanding requests
Cannot be combined with the Slow start duration attribute.

Target selection configuration

Entering the application from the load balancer DNS name

todos-prod-balancer-256120987.us-east-1.elb.amazonaws.com

EC2 > Load balancers > todos-prod-balancer

todos-prod-balancer

Details

arn:aws:elasticloadbalancing:us-east-1:504406221982:loadbalancer/app/todos-prod-balancer/65379603ab877317

Load balancer type Application	DNS name todos-prod-balancer-256120987.us-east-1.elb.amazonaws.com (A Record)	Status Active	VPC vpc-0424a6a9904e34a6d
IP address type IPv4	Scheme Internet-facing	Availability Zones subnet-05a9689ef64285acb us-east-1c (use 1-az2) subnet-0b84b54f59135a6c6 us-east-1a (use 1-az6)	Hosted zone Z35SXDOTRQ7X7K
Date created March 29, 2023, 01:04 (UTC+03:00)			

Items | S3 Ma | Load b | Load b | Mana | Target | Load b | Load b | Target | To: x +

Not secure | todos-prod-balancer-256120987.us-east-1.elb.amazonaws.com

TODO List!!!

Insert

userId	Title	Description	Date Time	Update	Update
1	1	1	2023-03-28T10:04:08	Update	Delete

Items | S3 Ma | Load b | Load b | Mana | Target | Load b | Load b | Target | To: x +

Not secure | todos-prod-balancer-256120987.us-east-1.elb.amazonaws.com

TODO List!!!

Insert

userId	Title	Description	Date Time	Update	Update
1	2	2	2023-03-28T22:22:50	Update	Delete

- Create a cloud watch service that will inspect your billing, the CPU of your Jenkins EC2 instance and the CPU of your production service.

