

# AWS Workshop: Highly available & scalable three-tier application deployment on AWS



**Prepared By:**



**Intuitive.Cloud**

Advanced Technology Solutions Group

33 Wood Ave. S, Suite #600

Iselin, NJ, 08830-2717, USA

URL: [www.intuitive.cloud](http://www.intuitive.cloud)

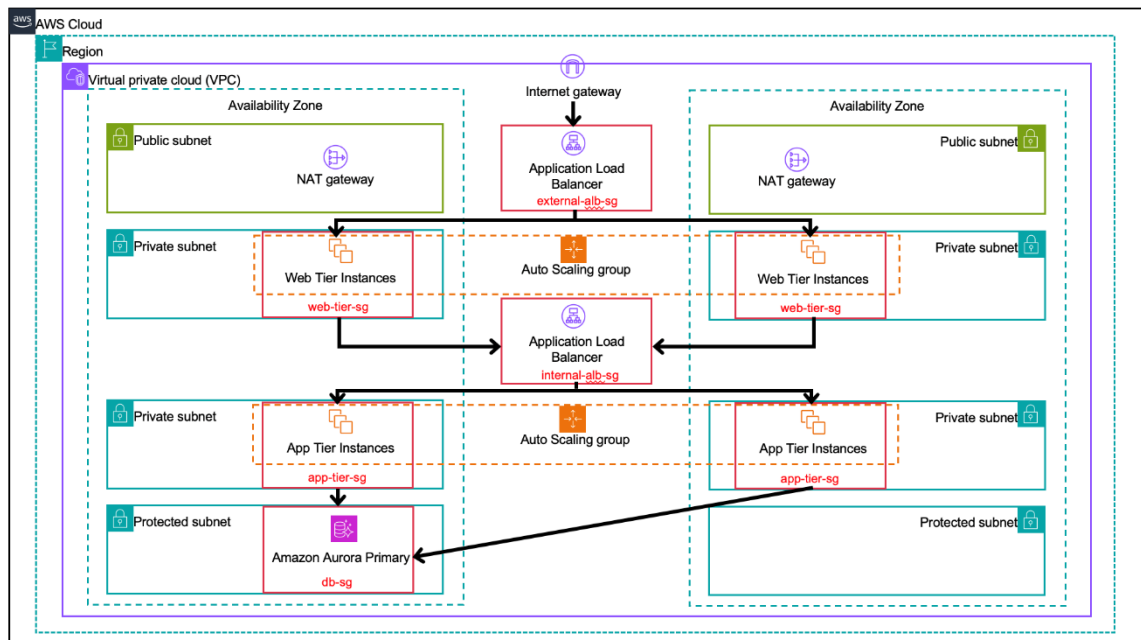


## Table of Contents

AWS Architecture .....	3
Step 1: Setting up networking & IAM roles as pre-requisite.....	4
Step 2: Creating a Web Server using EC2 Instance.....	7



## AWS Architecture



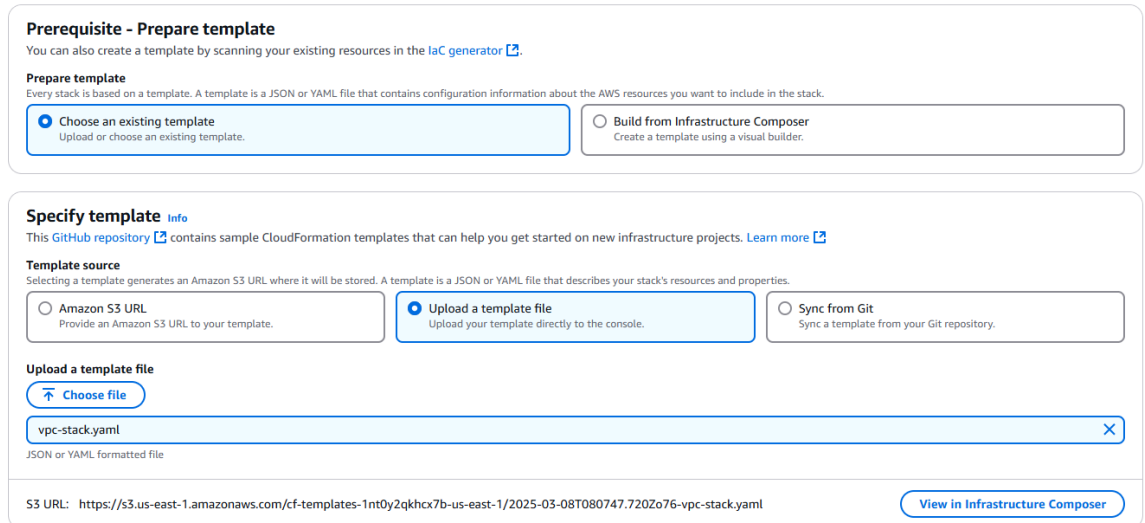
## Step 1: Setting up networking & IAM roles as pre-requisite

1. Go to **CloudFormation** service.
2. Click on **Create a stack With new resources**.



3. Download the '**CFT link**' to deploy the pre-requisite for the LAB
4. Choose **Upload a template file** and upload the file downloaded.

### Create stack



**Prerequisite - Prepare template**  
You can also create a template by scanning your existing resources in the [IaC generator](#).

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Choose an existing template  
Upload or choose an existing template.

☐ Build from Infrastructure Composer  
Create a template using a visual builder.

**Specify template** [info](#)  
This [GitHub repository](#) contains sample CloudFormation templates that can help you get started on new infrastructure projects. [Learn more](#)

**Template source**  
Selecting a template generates an Amazon S3 URL where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

☐ Amazon S3 URL  
Provide an Amazon S3 URL to your template.

☒ Upload a template file  
Upload your template directly to the console.

☐ Sync from Git  
Sync a template from your Git repository.

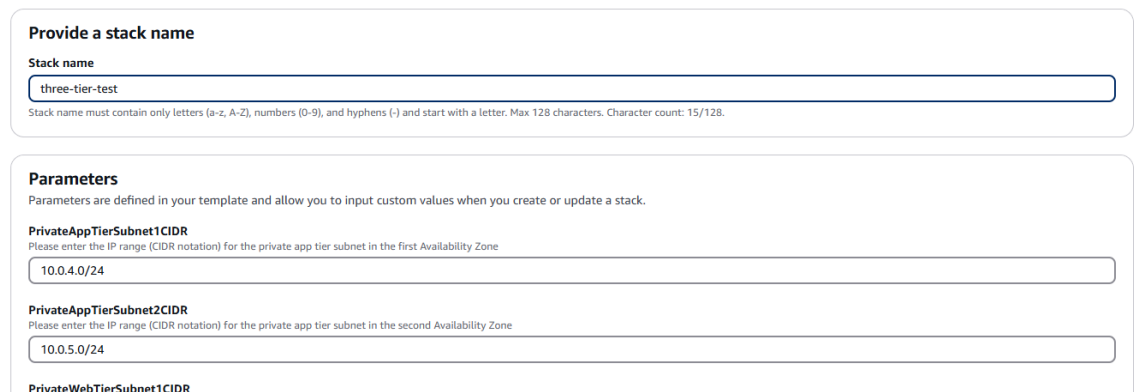
**Upload a template file**  
[Choose file](#)

vpc-stack.yaml  
JSON or YAML formatted file

S3 URL: <https://s3.us-east-1.amazonaws.com/cf-templates-1nt0y2qkxc7b-us-east-1/2025-03-08T080747.720Zo76-vpc-stack.yaml> [View in Infrastructure Composer](#)

5. You can leave all parameters with default values.

### Specify stack details



**Provide a stack name**

Stack name  
three-tier-test  
Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 15/128.

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**PrivateAppTierSubnet1CIDR**  
Please enter the IP range (CIDR notation) for the private app tier subnet in the first Availability Zone  
10.0.4.0/24

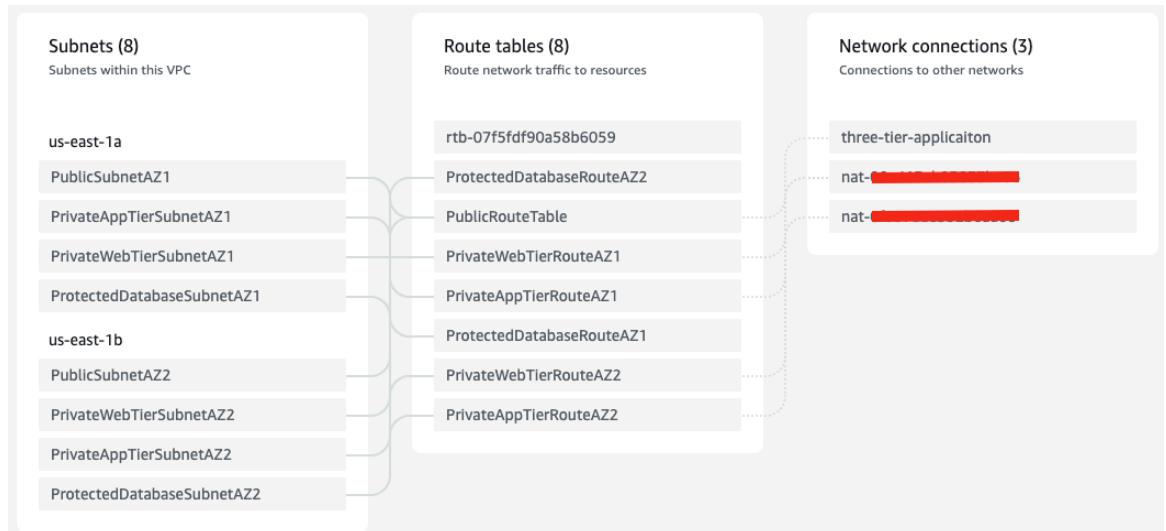
**PrivateAppTierSubnet2CIDR**  
Please enter the IP range (CIDR notation) for the private app tier subnet in the second Availability Zone  
10.0.5.0/24

**PrivateWebTierSubnet1CIDR**

6. Click on **Next** on **Step 2 (Specify stack details)** and **Step 3 (Configure stack options)**.
7. Finally, under review section click on **Submit**.
8. This stack creates the following resources:

- **VPC with 2 public, 4 private, and 2 protected subnets.** Two public subnets would be connected to a common route table, having network connections to

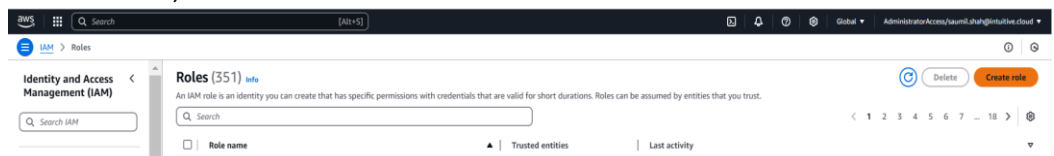
the **internet gateway**. Four private subnet will have 4 separate route tables, each route table will have network connects to to the **NAT gateway**. Protected subnets will have no path to the NAT gateways. Two private subnets will be used for frontend (web tier) logic and the other two private subnets for backend (app tier) logic. Protected subnets will have our RDS database.



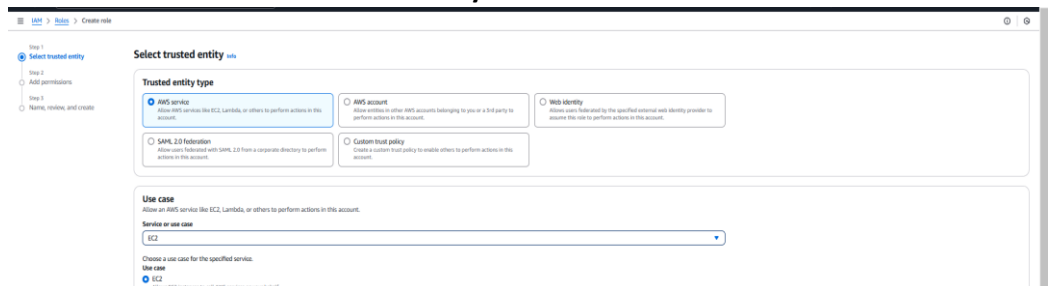
- Security groups namely
  1. WebTierSecurityGroup – To be used for all WebTier resources [EC2, ELB]
  2. AppTierSecurityGroup – To be used for AppTier resources- [EC2]
  3. DatabaseSecurityGroup- To be used for DatabaseTier [RDS]

## 9. Create an **IAM instance profile** for EC2

1. Open AWS Console and go to the **IAM** service.
2. Click on **Roles**, then click **Create role**.



3. Select AWS service as the **trusted entity** and choose **EC2**. Click **Next**.



4. Search for and attach the '**AmazonS3FullAccess**' and '**AmazonSSMManagedInstanceCore**' policies. Click **Next**.

Intuitive > Roles > Create role

Step 1: Select trusted entity  
Step 2: Add permissions  
Step 3: Name, review, and create

### Add permissions

Permissions policies (2/1201) [info](#)

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

Filter by type: All types 1 match

Policy name	Type	Description
AmazonSSMManagedInstanceCore	AWS managed	The policy for Amazon EC2 Role to enable A...

▶ Set permissions boundary - optional

Cancel Previous Next

5. Enter the role name as `flask-ec2-role-<your_user_id>` and click **Create role**.

Intuitive > Roles > Create role

Step 1: Select trusted entity  
Step 2: Add permissions  
Step 3: Name, review, and create

### Name, review, and create

Role details

**Role name**  
Enter a meaningful name to identify this role.  
`flask-ec2-role-test1`

**Description**  
Add a short explanation for this role.  
Allows EC2 instances to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": "ec2.amazonaws.com"
11      }
12    }
13  ]
14 }
15
16

```

Step 2: Add permissions

## Step 2: Creating a Web Server using EC2 Instance

1. Open the Amazon EC2. From the EC2 console dashboard, in the Launch instance pane, choose **Launch instance**.



2. Under Name and tags, for name enter Webserver.

### Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

#### Name and tags [Info](#)

Name

[Add additional tags](#)

3. Under Application and OS Images (Amazon Machine Image). Choose **Quick Start** and then choose the operating system (OS) for your instance. From Amazon Machine Image (AMI), select **Amazon Linux 2AMI**.

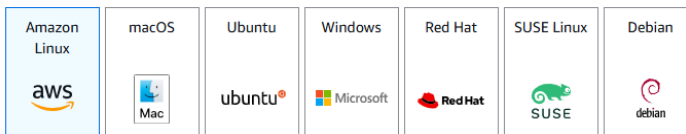
#### ▼ Application and OS Images (Amazon Machine Image) [Info](#)

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

Recents

My AMIs

**Quick Start**



[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-02a53b0d62d37a757 (64-bit (x86)) / ami-08523976443f71beb (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

4. Under Instance type, for Instance type, choose **t2.micro**.

#### ▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

5. Under Key pair (login), **Proceed without a key pair**.

#### ▼ 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.

Key pair name - *required*

Default value

[Create new key pair](#)

- Under Network settings, select **Edit**, under VPC choose **three-tier-application** VPC created in step 1. Under subnet select **PublicSubnetAZ1**. For Auto-assign public IP select **Enable**. Choose Select **existing security group** and choose **WebTierSecurityGroup**.

▼ Network settings

Info

VPC - required

Info

vpc-092494c8b0675df25 (three-tier-application)

10.0.0.0/16

↻

Subnet

Info

subnet-066d4848342aecb96

PublicSubnetAZ1

↻ Create new subnet

VPC: vpc-092494c8b0675df25

Owner: 185713903852

Availability Zone: us-east-1a

Zone type: Availability Zone

IP addresses available: 247

CIDR: 10.0.0.0/24

Auto-assign public IP

Info

Enable

↻

Additional charges apply when outside of free tier allowance

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

Common security groups

Info

Select security groups

↻

three-tier-application-WebTierSecurityGroup-tslFZDj8DRCS sg-08e57ac99e83ab2f2

×

VPC: vpc-092494c8b0675df25

↻ Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

- Under Advanced Details section, for **IAM instance profile** select **flask-ec2-role-  
<your\_user\_id>** created in step-1.

▼ Advanced details

Info

Domain join directory

Info

Select

↻ Create new directory

IAM instance profile

Info

flask-ec2-role-test1

arn:aws:iam:185713903852:instance-profile/flask-ec2-role-test1

↻ Create new IAM profile

Hostname type

Info

IP name

↻

DNS Hostname

Info

☒ Enable IP name IPv4 (A record) DNS requests

- Scroll down to **User Data** and copy contents from this [Link](#).

User data - optional

Info

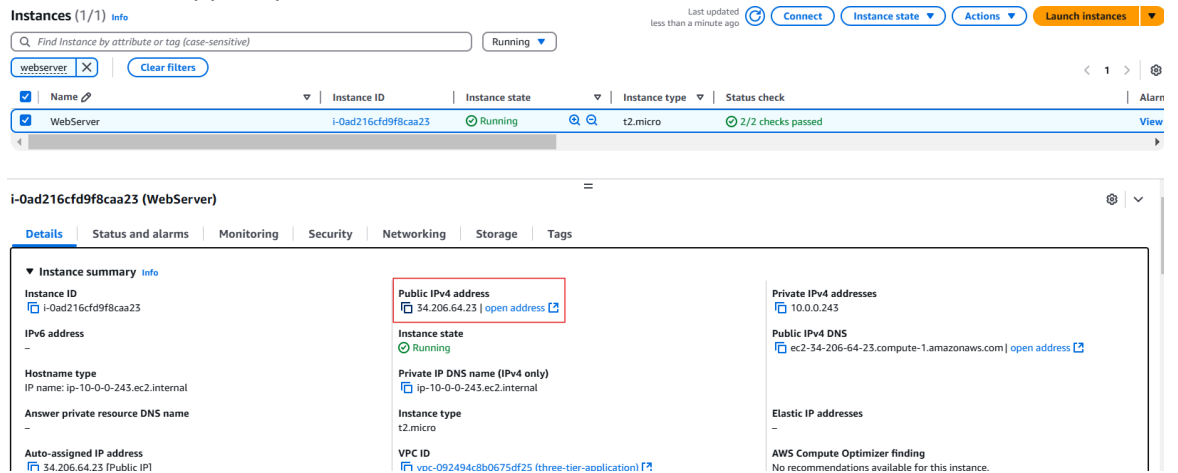
Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash
yum update -y
yum install -y python3 python3-pip git
git clone https://github.com/Saumil-Shah-ity/flask-three-tier.git /home/ec2-user/flaskapp
cd /home/ec2-user/flaskapp/frontend
sudo pip3 install flask flask-cors mysql-connector-python requests pymysql
sudo pip3 install --upgrade urllib3==1.26.16
sudo python3 /home/ec2-user/flaskapp/frontend/frontend.py
```



9. Click on **Launch Instance**.
10. Once the instance is up and running in **Healthy** state with **2/2 checks passed**. Select the instance and copy the public IP.



**Instances (1/1)** [Info](#)

Find Instance by attribute or tag (case-sensitive) Running Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[webserv](#) [Clear filters](#)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm
<input checked="" type="checkbox"/>	WebServer	i-0ad216cfd9f8caa23	Running	t2.micro	2/2 checks passed	<a href="#">View</a>

**i-0ad216cfd9f8caa23 (WebServer)**

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

**▼ Instance summary** [Info](#)

<b>Instance ID</b> <a href="#">i-0ad216cfd9f8caa23</a> <b>IPv6 address</b> - <b>Hostname type</b> IP name: ip-10-0-0-243.ec2.internal <b>Answer private resource DNS name</b> - <b>Auto-assigned IP address</b> <a href="#">34.206.64.23</a> (Public IP)	<b>Public IPv4 address</b> <a href="#">34.206.64.23</a>   <a href="#">open address</a> <b>Instance state</b> <a href="#">Running</a> <b>Private IP DNS name (IPv4 only)</b> <a href="#">ip-10-0-0-243.ec2.internal</a> <b>Instance type</b> t2.micro <b>VPC ID</b> <a href="#">vpc-092494c8b0675df25</a> (three-tier-application)	<b>Private IPv4 addresses</b> <a href="#">10.0.0.243</a> <b>Public IPv4 DNS</b> <a href="#">ec2-34-206-64-23.compute-1.amazonaws.com</a>   <a href="#">open address</a> <b>Elastic IP addresses</b> - <b>AWS Compute Optimizer finding</b> No recommendations available for this instance.
---	--	---

11. Open your browser and search for `http://<public_id>:80`. The screen below should appear.

**To-Do List**

Add Task

Frontend Dummy Task 1 ✖

Frontend Dummy Task 2 ✖