02 Compute Fundementals

July 1, 2020

1 Servers In The Cloud

Servers in the cloud have revolutionized the IT industry - Scale capacity up and down based on demands - Storage, more memory, and computing power can be added as needed - Obtain servers in minutes - No need for onsite hardware or capital expenses

2 Elastic Cloud Compute (EC2)

EC2 is a foundational piece of AWS cloud computing platform and is a service that provides servers for rent in the cloud

They are called instances. Instances are physical servers in a data center and are not to be confused with serverless

2.1 Pricing Options

There are several pricing options for EC2 - On Demand- Pay as you go, no contract - Dedicated Hosts- you have your own dedicated hardware and don't share it with others - Spot- you place a bid on an instance price. If there is extra capacity that falls below your bid, and EC2 instance is provisioned. If the price goes above your bid while the instance is running, the instance is terminated - Reserved Instances- You earn huge discounts if you pay up front and sign a 1-year or 3-year contract

2.1.1 TIPS

- Spot instances can save you up to $ 90$% off the on-demand pricing
- There are several instance types that provide varying combinations of $ CPU $, memory, storage and networking capacity

3 Elastic Block Store

Elastic Block Store (EBS) is a storage solution for EC2 instances and is a physical hard drive that is attached to the EC2 instance to increase storage

3.1 Tips

- EBS is found on the EC2 dashboard
- There are several EBS volume types that fall under the categories of Solid State Drive (SSD) and Hard Disk Drive (HDD)

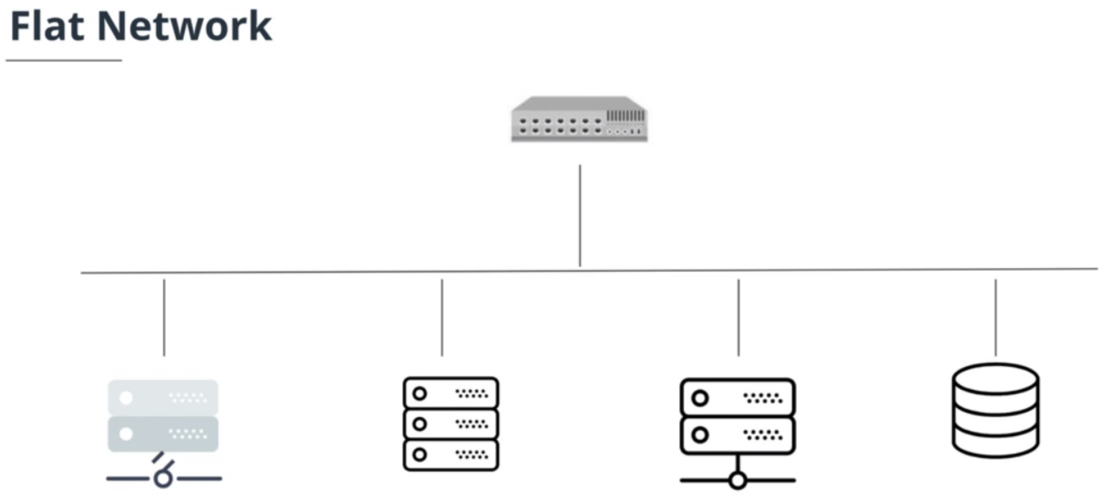
4 Security

Security in the cloud allows you to have complete control over your virtual networking enviorment

- Configure your virtual network with public or private facing subnets
- Launch your servers in the selected network to secure access

5 Virtual Private Cloud (VPC)

To Save costs, everything is part of the same broadcast network and communicate with each other. If you want more control/security, use a VPC



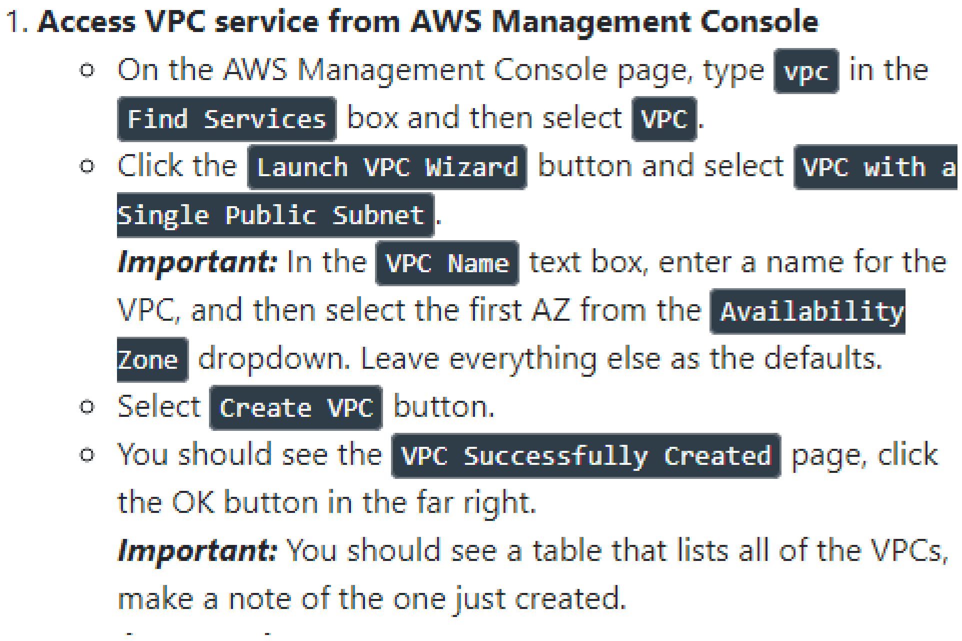
Virtual Private Cloud or VPC allows you to create your own private network in the cloud. You can launch services, like EC2, inside of that private network. A VPC spans all avalibility zones in the region

VPC allows you to control your virtual networking enviorment wich includes - IP adress ranges - subnets - route tables - network gateways

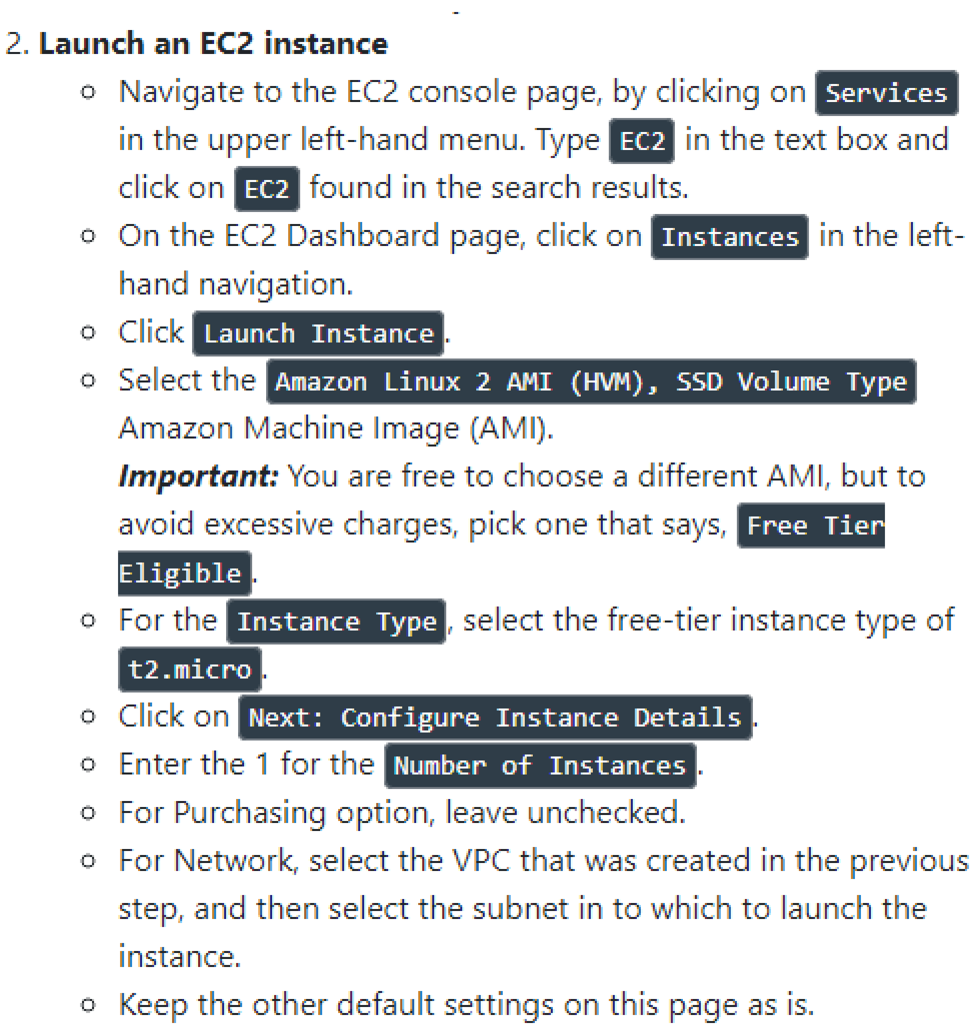
5.1 Tips

- VPC is found under networking and content delivery section of the console
- The default limit is 5 VPC's per region. You can request an increase
- Your AWS resources are automatically provisioned in a default VPC
- There are no additional charges for creating and using the VPC
- You can store data in the Amazon S3 and restrict access so that its only accessible for instance in your VPC

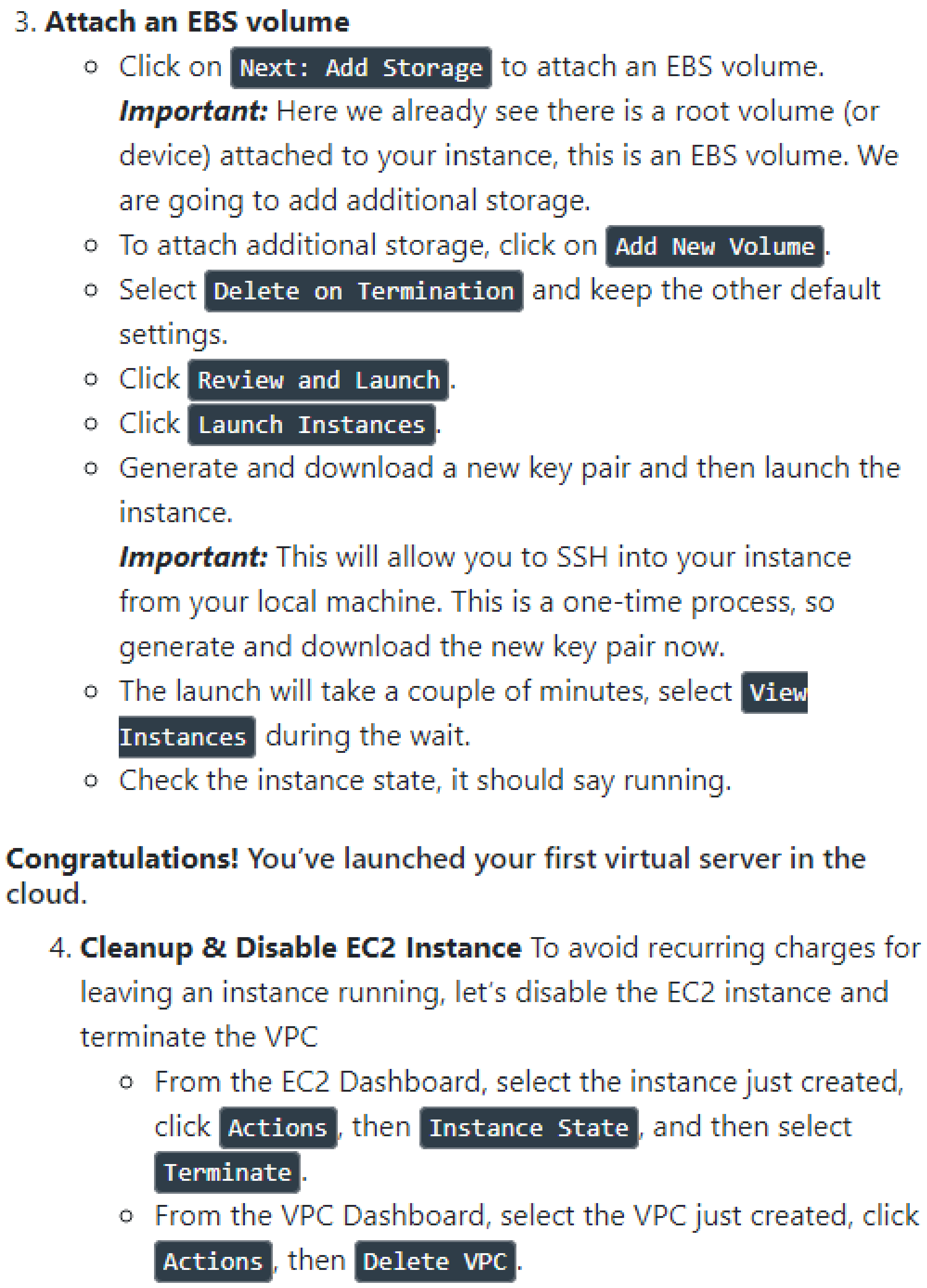
6 Launch an EC2 Instance



1. **Access VPC service from AWS Management Console**
    - On the AWS Management Console page, type `vpc` in the `Find Services` box and then select `VPC`.
    - Click the `Launch VPC Wizard` button and select `VPC with a Single Public Subnet`.
      Important: In the `VPC Name` text box, enter a name for the VPC, and then select the first AZ from the `Availability Zone` dropdown. Leave everything else as the defaults.
    - Select `Create VPC` button.
    - You should see the `VPC Successfully Created` page, click the OK button in the far right.
      Important: You should see a table that lists all of the VPCs, make a note of the one just created.



2. **Launch an EC2 instance**
   - Navigate to the EC2 console page, by clicking on `Services` in the upper left-hand menu. Type `EC2` in the text box and click on `EC2` found in the search results.
   - On the EC2 Dashboard page, click on `Instances` in the left-hand navigation.
   - Click `Launch Instance`.
   - Select the `Amazon Linux 2 AMI (HVM), SSD Volume Type` Amazon Machine Image (AMI).
     Important: You are free to choose a different AMI, but to avoid excessive charges, pick one that says, `Free Tier Eligible`.
   - For the `Instance Type`, select the free-tier instance type of `t2.micro`.
   - Click on `Next: Configure Instance Details`.
   - Enter the 1 for the `Number of Instances`.
   - For Purchasing option, leave unchecked.
   - For Network, select the VPC that was created in the previous step, and then select the subnet in to which to launch the instance.
   - Keep the other default settings on this page as is.



3. **Attach an EBS volume**
   - Click on `Next: Add Storage` to attach an EBS volume.
     Important: Here we already see there is a root volume (or device) attached to your instance, this is an EBS volume. We are going to add additional storage.
   - To attach additional storage, click on `Add New Volume`.
   - Select `Delete on Termination` and keep the other default settings.
   - Click `Review and Launch`.
   - Click `Launch Instances`.
   - Generate and download a new key pair and then launch the instance.
     Important: This will allow you to SSH into your instance from your local machine. This is a one-time process, so generate and download the new key pair now.
   - The launch will take a couple of minutes, select `View Instances` during the wait.
   - Check the instance state, it should say running.

Congratulations! You've launched your first virtual server in the cloud.

4. **Cleanup & Disable EC2 Instance** To avoid recurring charges for leaving an instance running, let's disable the EC2 instance and terminate the VPC
   - From the EC2 Dashboard, select the instance just created, click `Actions`, then `Instance State`, and then select `Terminate`.
   - From the VPC Dashboard, select the VPC just created, click `Actions`, then `Delete VPC`.

7 Compute Power

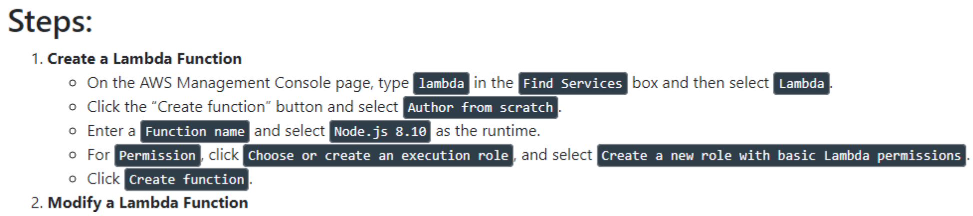
Compute power in the cloud is a faster way to build applications, providing: - no servers to manage (serverless) - ability to continously scale - ability to run code on demand in response to events - pay only when your code runs

7.1 Lambda

AWS Lambda provides you with computing power in the cloud by allowing you to execute code without standing up or managing servers

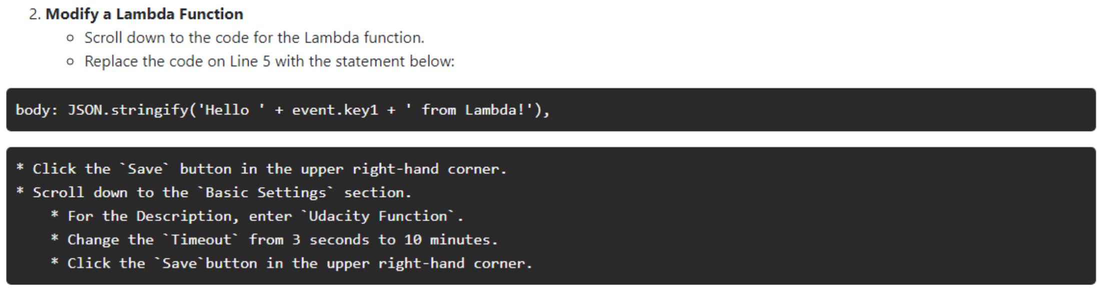
7.1.1 Tips

- Lambda is found under the compute section of the console
- Lambda have a time limit of 15 mins
- The code you run on Lambda is called a "Lambda Function"
- Lambda code can be triggered by other AWS service
- AWS Lambda code can be triggered by other AWS services
- Lambda supports Python, Go, PowerShell, Node, C#, etc.
- Lambda code can be authored via the console



Steps:

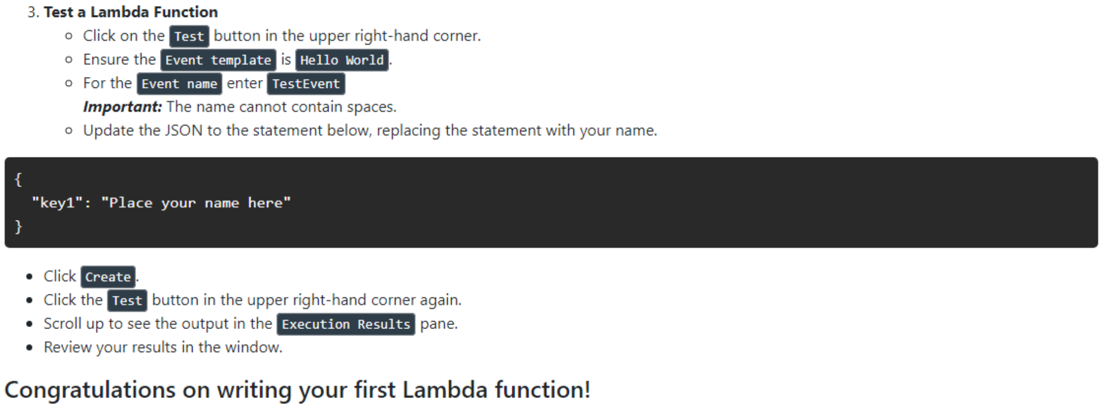
1. **Create a Lambda Function**
   - On the AWS Management Console page, type `lambda` in the `Find Services` box and then select `Lambda`.
   - Click the "Create function" button and select `Author from scratch`.
   - Enter a `Function name` and select `Node.js 8.10` as the runtime.
   - For `Permission`, click `Choose or create an execution role`, and select `Create a new role with basic Lambda permissions`.
   - Click `Create function`.
2. **Modify a Lambda Function**



2. **Modify a Lambda Function**
   - Scroll down to the code for the Lambda function.
   - Replace the code on Line 5 with the statement below:

```
body: JSON.stringify('Hello ' + event.key1 + ' from Lambda!'),
```

```
* Click the `Save` button in the upper right-hand corner.
* Scroll down to the `Basic Settings` section.
    * For the Description, enter `Udacity Function`.
    * Change the `Timeout` from 3 seconds to 10 minutes.
    * Click the `Save`button in the upper right-hand corner.
```



3. **Test a Lambda Function**
   - Click on the `Test` button in the upper right-hand corner.
   - Ensure the `Event template` is `Hello World`.
   - For the `Event name` enter `TestEvent`
     Important: The name cannot contain spaces.
   - Update the JSON to the statement below, replacing the statement with your name.

```
{
  "key1": "Place your name here"
}
```

- Click `Create`.
- Click the `Test` button in the upper right-hand corner again.
- Scroll up to see the output in the `Execution Results` pane.
- Review your results in the window.

Congratulations on writing your first Lambda function!

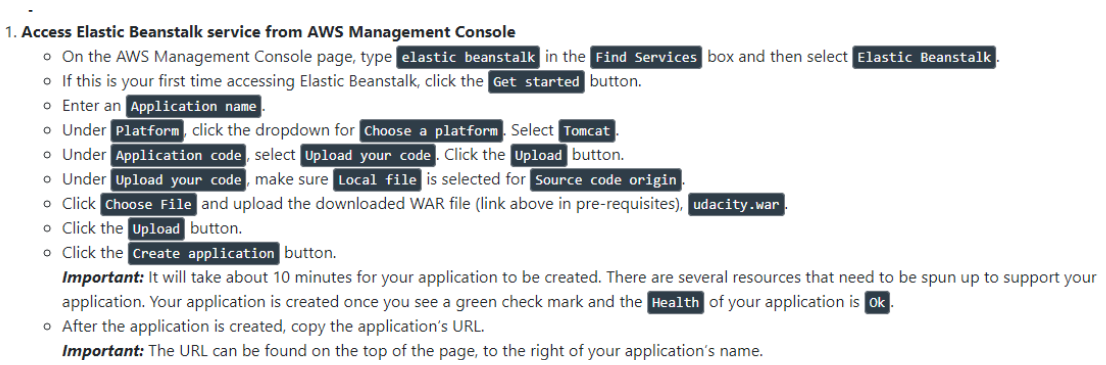
8 Elastic Beanstalk

Elastic Beanstalk is an orchestration service that allows you to deploy a web application at the touch of a button by spinning up (or provisioning) all of the services that you need to run your application

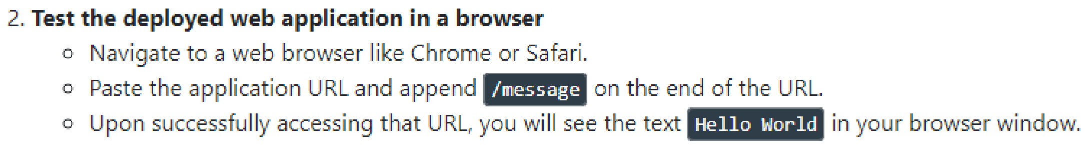
8.1 Tips

- Elastic Beanstalk is found under the compute section of the console
- Elastic Beanstalk can be used with backend technologies such as Python and Docker
- You can run your application in a VPC

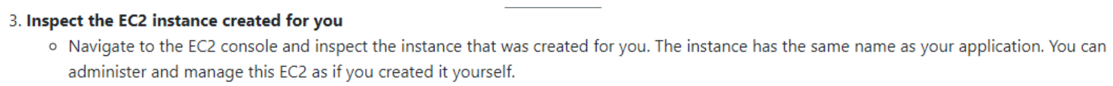
9 Deploy App to Beanstalk



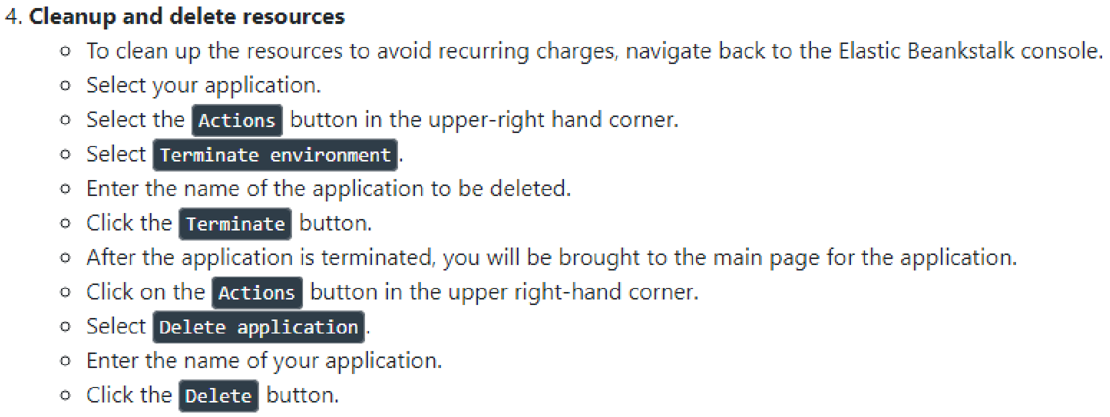
1. **Access Elastic Beanstalk service from AWS Management Console**
   - On the AWS Management Console page, type `elastic beanstalk` in the `Find Services` box and then select `Elastic Beanstalk`.
   - If this is your first time accessing Elastic Beanstalk, click the `Get started` button.
   - Enter an `Application name`.
   - Under `Platform`, click the dropdown for `Choose a platform`. Select `Tomcat`.
   - Under `Application code`, select `Upload your code`. Click the `Upload` button.
   - Under `Upload your code`, make sure `Local file` is selected for `Source code origin`.
   - Click `Choose File` and upload the downloaded WAR file (link above in pre-requisites), `udacity.war`.
   - Click the `Upload` button.
   - Click the `Create application` button.
     Important: It will take about 10 minutes for your application to be created. There are several resources that need to be spun up to support your application. Your application is created once you see a green check mark and the `Health` of your application is `Ok`.
   - After the application is created, copy the application's URL.
     Important: The URL can be found on the top of the page, to the right of your application's name.



2. **Test the deployed web application in a browser**
   - Navigate to a web browser like Chrome or Safari.
   - Paste the application URL and append `/message` on the end of the URL.
   - Upon successfully accessing that URL, you will see the text `Hello World` in your browser window.



3. **Inspect the EC2 instance created for you**
   - Navigate to the EC2 console and inspect the instance that was created for you. The instance has the same name as your application. You can administer and manage this EC2 as if you created it yourself.



4. **Cleanup and delete resources**
   - To clean up the resources to avoid recurring charges, navigate back to the Elastic Beankstalk console.
   - Select your application.
   - Select the `Actions` button in the upper-right hand corner.
   - Select `Terminate environment`.
   - Enter the name of the application to be deleted.
   - Click the `Terminate` button.
   - After the application is terminated, you will be brought to the main page for the application.
   - Click on the `Actions` button in the upper right-hand corner.
   - Select `Delete application`.
   - Enter the name of your application.
   - Click the `Delete` button.