

# Project 0

## Initial setup and creation

### Introduction

In this project you will setup the base for the rest of projects during this course. There will be both individual and team based tasks that will need to be completed.

### Tasks

Below are a set of tasks that need to be completed for Project 0. There are both individual (I) and team (T) based tasks.

### Setup

The following tasks will be completed once but you will use the results of these tasks throughout the rest of the class.

#### Build a Team (T)

You will need to create a team of 3 - 4 people. Once you have found your team members, you will need to come up with a team name. Finally, you will need to let the professor know the members of your team and its name.

#### GitHub (I)

Git is a popular distributed version control system (DVCS) that allows a group of users to work together on a common set of code. We will be using git as part of this class so it will be helpful to learn about it as quickly as possible. You should have git installed locally on your computer and it should be used to track the work you do with your team.

GitHub is an online platform that allows people around the world to work on projects together. We will be utilizing GitHub in the this course to track team work. You will need to sign up for a GitHub account (free) and send the usernames of your group to the professor.

GitHub gives teams a few extra resources such as an issue/task tracker and a peer review system for changes made (pull requests). We will discuss this in more depth during class.

It is also suggested that you sign up for the GitHub Student Developer pack. This provides you with a set of free and discounted software tools that can be useful in future endeavors.

<https://education.github.com/pack>

## Amazon Web Services Account (I)

Amazon Web Services (AWS) will be used during this course. AWS allows us to create a professional and public environment with little overhead. Each member of the team will create their own account. AWS Educate is a program that gives students a \$100 credit to use when learning AWS. We will combine these credits together in this course so that all of the projects are free.

You will need to create a full account and not use an AWS starter account. This will require each member of the team to have a credit card for initial account creation. As long as all of the advice is followed in this class, your credit card should never be charged.

You will be sent a code that can be used to redeem a \$100 credit for AWS. You can hold onto this code and apply all of them to a single account. It is up to the group. If each person applies this to their account, you will need to look into an AWS Organization so that you can chain together all of the accounts.

## Slack

Your team can use Slack to communicate about work that needs to be done. You can sign-up at <https://comp-490.slack.com/signup> using your @my.csun.edu address. You should join the #cit480-11am channel and you can create another channel for your group. Please don't make these channels private groups and do not share secrets inside of slack.

Also note that this Slack is free so there is no long term history. You should keep track of important conversations you have in a different medium too for longevity.

## Leader of the Project

One member of the team should be appointed the "leader" for this project. This leader should change for each project between the different members of the group. The lead is responsible for making sure that work is completed by other members in a timely manner. The lead will also need to work on other aspects of the project but must also keep track of everything that is happening. The lead will need to coordinate work as it is done.

## Project Tasks

These tasks are specific to Project 0 and will result in the graded material.

You will need to break down the different steps below add a set of issues in your repo on GitHub for each task and assign them to members of the team. Team members should track progress in these issues and attach them to pull requests if applicable. Issues can work as a log that tracks the work done over the past week.

## Purchase a Domain (T)

You will need to purchase a public domain for your team to use for the course. A domain name is something like csun.edu or google.com. You can use any Top-level Domain (TLD) that you'd like as these tend to be cheaper. Ideally, the domain name your choose should allow the professor to easily identify which team owns it.

NameCheap is a domain registrar that allows you to search for your preferred domain name. It also has deals where you can get a domain name for as little as \$0.99 for a year.

## Setup AWS Virtual Private Cloud (VPC)

An AWS VPC allows every organization to setup one or more networks inside of AWS. You should be given a default VPC with a given subnet (Usually 172.31.0.0/16). You will need to break these networks into public and private subnetworks to allow services in AWS to function properly.

Design your network so that there are the following subnets:

- 3 public subnets in 3 different availability zones (AZ) with at least 1024 available IP addresses per subnet.
- 3 private subnets in 3 different availability zones (AZ) with at least 4096 available IP addresses per subnet.
- 1 route table for the public subnets
- 1 route table for the private subnets
- 1 Internet Gateway (IG) attached to the default route of the public subnet routing table

## Setup a Web Server on AWS Elastic Compute Cloud (EC2)

AWS Elastic Compute Cloud allows us to easily create virtual machines (VM), referred to as instances in AWS, with a base OS image (Amazon Machine Image or AMI). There are different types of instances that define the number of vCPU cores, RAM and storage are provided by default. Instances are currently charged for by the second but when looking up the price you will see it provided for each hour of service the instance uses.

You will need to choose the OS that you'd like to use as well as the web server that will be serving up your pages.

You also need to think of a secure way to allow all users to access your machine while not sharing credentials. You should look at the EC2 Userdata service and the cloud-init process on most machines. You should be able to determine a way to provide SSH access without the use of passwords. All users can still use the default account provided on the system.

## Setup your Domain

Your domain should use AWS Route53 to serve requests. You will need to setup the provider you purchased your domain from point to your domain on the Route53 servers. You will need to setup the following records in AWS Route53 that will all be served from your webserver.

- Apex domain
- www.<yourdomain>
- blog.<yourdomain>

## Setup your Blog

We will be creating our blog by using a "static site generator" named Hugo. Hugo allows us to write pages in a language called "markdown" that is interpreted and built into a website. Each post will be its own file and the site will be generated using the command-line tool provided by Hugo. The result will be a set of HTML and other files used to display the website.

You will also need to pick out a template to define the style of your website. You can also add images and other features to the site as you see fit.

Each blog entry should be technical in nature and discuss in detail a task performed during the week. Every blog entry should be added to the repo for your site and should be reviewed by at least one other member of the team. The other member should read through your post and approve it to be merged into the team's repo.

You can complete the blog entries at any time but they must be done each week by 11:59:59 p.m. on Friday. You will provide a link to the URL on the site for your entry.

## Generate TLS Certificates

It is important to protect users to any website by encrypting traffic to and from it. We do this using Transport Layer Security (TLS) certificates. For a long time, buying a TLS certificate was very expensive but recent efforts have made these certificates free. One such project that has helped this is called Let's Encrypt. Let's Encrypt will allow the owner of a domain to verify ownership and be able to get a TLS certificate for free.

You must generate a set of security certificates for your domain that will work for each of the DNS addresses mentioned in the section "Setup your Domain". A single certificate should be used for all of the addresses.

The private key of your TLS certificate should be kept secret and not shared with anyone outside of your group to ensure that your users are protected. You must determine a safe and effective way to store and protect your certificates while also making them available to your instance to use. You can look at other AWS services such as the Secret Store to help with this.

## Deploy your blog

You must deploy your to your EC2 instance so that it is publically available. You are not allowed to do a "git clone" on the web server for your blog. You must create a tarball or zip file and upload that to your web server and deploy the files to the correct location.

## Task Breakdown

You will need to break down all of your tasks into a set of issues inside of GitHub. These tasks should be broken down into subtasks as needed and assigned to different members of the team. Each member will be responsible for keeping the task up to date with information about how the task is being completed and should close the task when it is done.

You should show the professor your task breakdown.

## Deliverables

These are the requirements for grading for Project 0.

1. Complete all of the tasks above.
2. A running website on a service in AWS EC2 that is publicly available over your DNS address using HTTPS.
3. Individual blog posts (1 each week by each member, 2 total per person) saved to your blog repo and publicly available on your site.
4. Complete documentation of all tasks performed. Documentation must be clear, concise and complete as to not leave anything out.

You should have the following documents in your infrastructure repository as a set of markdown files.

- a. README.md for each repo giving basic information
- b. AWS setup
- c. AWS services setup (EC2, VPC, Route53, etc)
- d. AWS service diagram showing each aspect of the infrastructure
- e. How to write a blog post
- f. How to deploy a new blog post

Documentation is separate from the individual blog posts and should be structured differently.

5. Group presentation at the end of the project.

## Grading

30 pts	All tasks completed
30 pts	Complete documentation that is easy to understand
20 pts	Group presentations
20 pts	Individual blog posts
<b>100 pts</b>	<b>Total</b>