

# **SIT323 Practical Software Development, Trimester 2, 2020**

## **Assessment Task 2**

### **Due Date**

Sunday, 8:00 PM (AEDT), 4<sup>th</sup> October, 2020.

### **Summary**

You need to create software to help determine a set of allocations such that each allocation specifies which tasks are allocated to run on a particular processor. These allocations are based on data from within a configuration file.

Your software will use the following configuration files stored on an Azure blob.

- <https://sit323sa.blob.core.windows.net/at2/TestSmall.cff>
- <https://sit323sa.blob.core.windows.net/at2/TestLarge.cff>
- <https://sit323sa.blob.core.windows.net/at2/TestExtraLarge.cff>

For each configuration file, the ideal goal is to find one or more allocations:

1. that consume a minimum amount of energy
2. where the overall runtime of each allocation does not exceed a program duration
3. where processor RAM constraints are satisfied

There might be one or more allocations that meet these constraints. Also, there might be no allocation that meets these constraints because the program duration is too small for all combinations of task allocations.

### **Application**

As your software might not be able to find the ideal allocation within several years, it must attempt to obtain a set of allocations within 5 minutes, where these allocations have the smallest energy consumption that you can find but also satisfy the time and memory constraints. That is, after 5 minutes, your software must:

- stop searching for other allocations,
- display the low energy consuming allocations that your program found,
- display the runtimes and amount of energy consumed next to each allocation, and
- display the maximum of amount of processor RAM required by the tasks allocated to that processor, and the amount of RAM available to that processor.

Your GUI client software will directly read a configuration file that is stored on Azure. That is, your software must open the file using its URL. Do not download these configuration files, and open them on a local drive.

You will design and implement one or more WCF Services to compute a set of allocations based on the imported configuration data. Each WCF Service will run on an AWS virtual machine. This service will be invoked by your client software. This client software will provide/pass configuration data to this service, and this service will return a set of allocations.

In order to use more compute resources (AWS VMs) to more quickly determine a set of allocations, you will use several AWS virtual machines, AMIs, auto scaling groups, a load balancer, and other AWS resources. Each of these VMs will have your WCF Service installed.

As your software will send several asynchronous requests to your WCF Services, it will receive responses from several WCF Services. Each response should contain a set of allocations. For example:

- WCF Service 1 returns a set of 8 allocations, where each allocation consumes 100.
- WCF Service 2 returns a set of 4 allocations, where each allocation consumes 150.
- WCF Service 3 returns a set of 12 allocations, where each allocation consumes 90.
- WCF Service 4 returns a set of 5 allocations, where each allocation consumes 123.

Your software needs to determine which response contains allocations that consume the smallest amount of energy. For example, the set of 12 allocations (in the above list) consume the smallest amount of energy.

## **AWS Cloud Architecture**

The cloud architecture that you will implement is depicted in Figure 1.

In general, client requests are directly sent to the application load balancer. Each request is sent to one of the Microsoft IIS web servers on a VM, and finally to one of your WCF Services. A response will be delivered to the client.

This architecture will use three different types of VM: t2.nano, t2.micro, and t2.small.

This architecture will use three auto scaling groups. Each auto scaling group:

- uses only one type of VM, and not the same as another group
- commences with one VM
- launches one VM when CPU utilisation exceeds 70%
- does not exceed 4 VMs
- terminates one VM when CPU utilisation is less than 30%
- retains at least 1 VM

Other main AWS resources required for this assessment task are depicted in Figure 1.

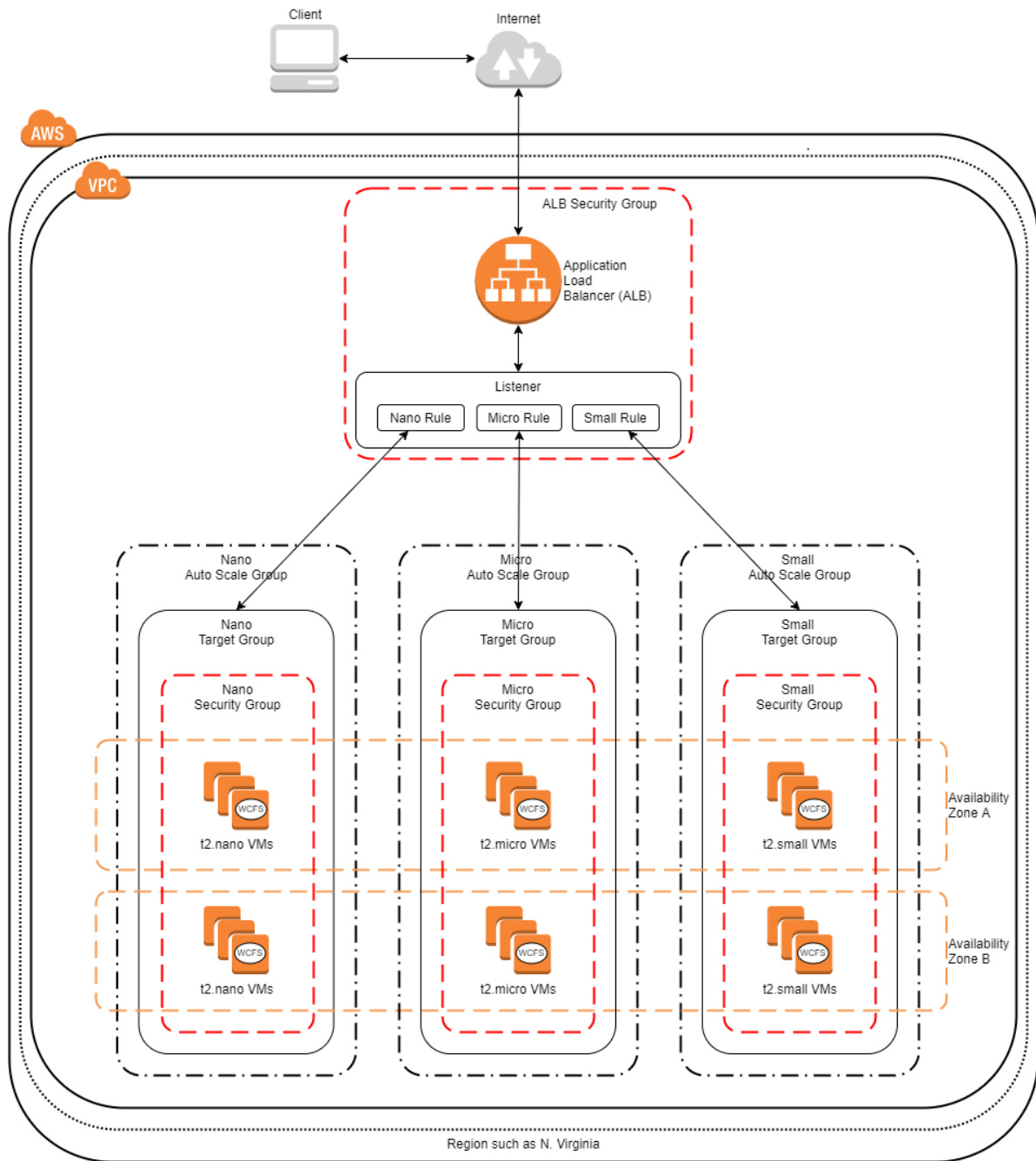


Figure 1.

## Submission

Please submit 2 items. **Details of these 2 items will be published as soon as possible.**

1. A ZIP file of your Visual Studio files and folders.
2. An mp4 video about your cloud system. This video should present AWS resources used in your cloud system, and it should demonstrate your cloud system working.