Migrating Web Application:

* Create web app
  + Select resource group
  + Give name
  + Publish > code
  + Runtime stack > ASP.NET v4.7
  + Region > East US
  + App Service Plan > Windows Plan > new name > Dev/Test > F1 pricing plan
  + No monitoring
  + Review & create
* Open code in VSC
  + Deploy web app to Azure

Lift and Shift Migration with Virtual Machine Scale Sets (VMSS):

* Create Virtual Machine Scale Set
  + Select resource group
  + Give name
  + Region > same as others
  + Size > sort by cost > B1ls (lowest)
  + Authentication > password
    - Username:udacityadmin
    - Password: Password123!@#
  + Got to Networking
    - Load balancer > yes
      * Azure Load balancer
  + Go to Scaling
    - 2 instances
    - Manual
  + Go to Management
    - Boot diagnostics > yes
  + Go to Advanced
    - Copy/paste cloud init
* *#cloud-config*
* package\_upgrade: **true**
* packages:
* - nginx
* - nodejs
* - npm
* write\_files:
* - owner: www-data:www-data
* path: /etc/nginx/sites-available/**default**
* content: |
* server {
* listen 80;
* location / {
* proxy\_pass http:*//localhost:3000;*
* proxy\_http\_version 1.1;
* proxy\_set\_header Upgrade $http\_upgrade;
* proxy\_set\_header Connection keep-alive;
* proxy\_set\_header Host $host;
* proxy\_cache\_bypass $http\_upgrade;
* }
* }
* - owner: azureuser:azureuser
* path: /home/azureuser/myapp/index.js
* content: |
* **var** express = **require**('express')
* **var** app = express()
* **var** os = **require**('os');
* app.get('/', **function** (req, res) {
* res.send('<h1>Welcome to Azure VM Scale Set</h1> <h2>Hello World from host ' + os.hostname() + '! </h2>')
* })
* app.listen(3000, **function** () {
* console.log('Hello world app listening on port 3000!')
* })
* runcmd:
* - service nginx restart
* - cd "/home/azureuser/myapp"
* - npm init
* - npm install express -y
* - nodejs index.js
  + Review and create
    - Go to resource. Go to Networking
      * Add inbound rule
        + Source > any
        + Destination > 80
        + Priority > 100
        + Name > Port\_80
  + Verify in “instances”

Lift and Shift Migration with Load Balancer:

1. Create the Azure Resource Group
   * name: vm-xxx-rg (where xxx is a random number)
   * location: closest region
2. Virtual Network is shared between the 2 VMs and application gateway:
   * A virtual network with address prefix: 10.0.0.0/16
   * 2 subnet prefix:
     + application gateway subnet address prefix: 10.0.1.0/24
     + backend for VM subnet address prefix: 10.0.2.0/24
3. Create 2 virtual machines with the following conditions:
   * names: vm1 and vm2
   * image: UbuntuLTS
   * admin username: book\_admin
   * Use the content in cloud-init.txt for the cloud init custom data which will install prerequistes
   * VM Size: costing less than $10 or sku Standard\_B1ls
   * Each VM will have a public IP address to allow ssh communication for deployment
4. For each Virtual Machine, deploy content to the Website using SCP at following directory /home/book\_admin/web:

e.g: scp -r ./web/ <\*\*adminUsername\*\*>@<\*\*vmIpAddress\*\*>:/home/<\*\*adminUsername\*\*>/web

1. Create an application gateway:
   * Name: ag-xxx (xxx should match the number set in your resource group to stay organized)
   * Tier: Standard\_v2
   * Minimum Scale unit: 2
   * Create a public IP address
   * Use the Virtual Network created for the VMs
   * On the frontend settings, create a public IP
   * On the backend settings, add the 2 virtual machines created in Step 2
   * Configure routing rules to allow HTTP/port 80 communication to the backend
   * Get the Application Gateway IP Address to browse the application

Create Azure SQL Database:

* Create > Azure SQL
  + SQL databases
    - Choose resource group
    - Name
    - Create new Server
    - Compute&storage > basic > 2Gig
    - Review and create
  + Click on server name
    - Firewall and virtual networks
      * Client IP address
        + Create rule name
        + Start: 0.0.0.0
        + End: 255.255.255.255