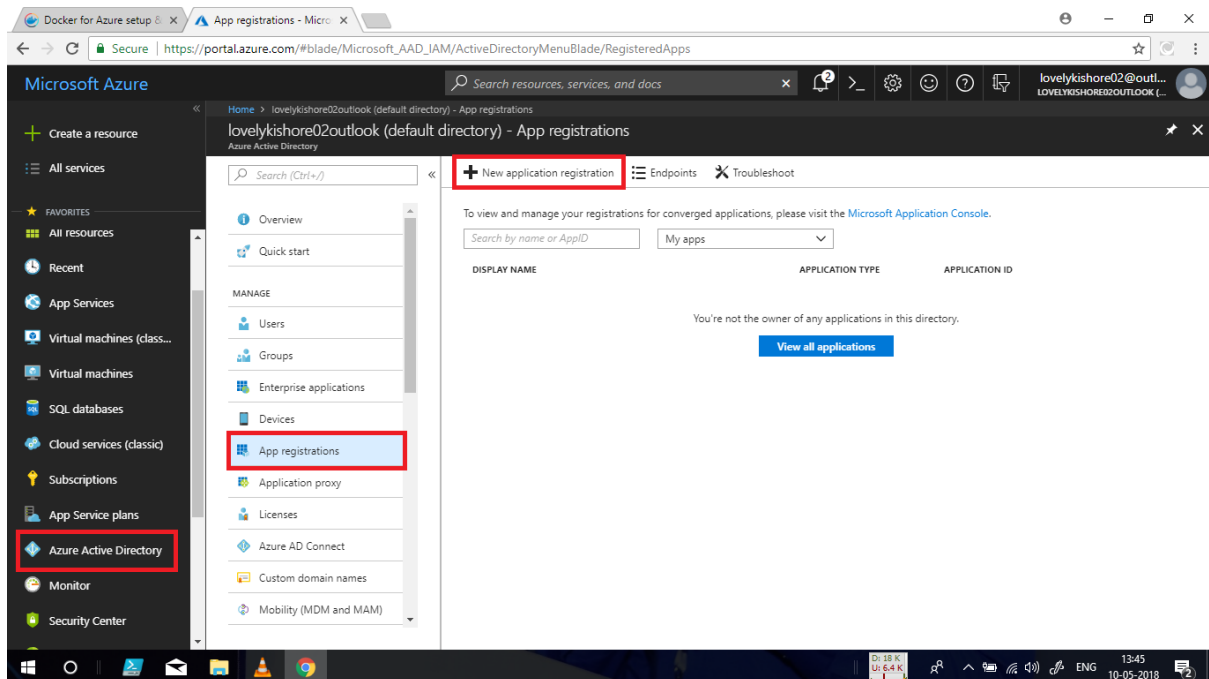


Docker Swarm

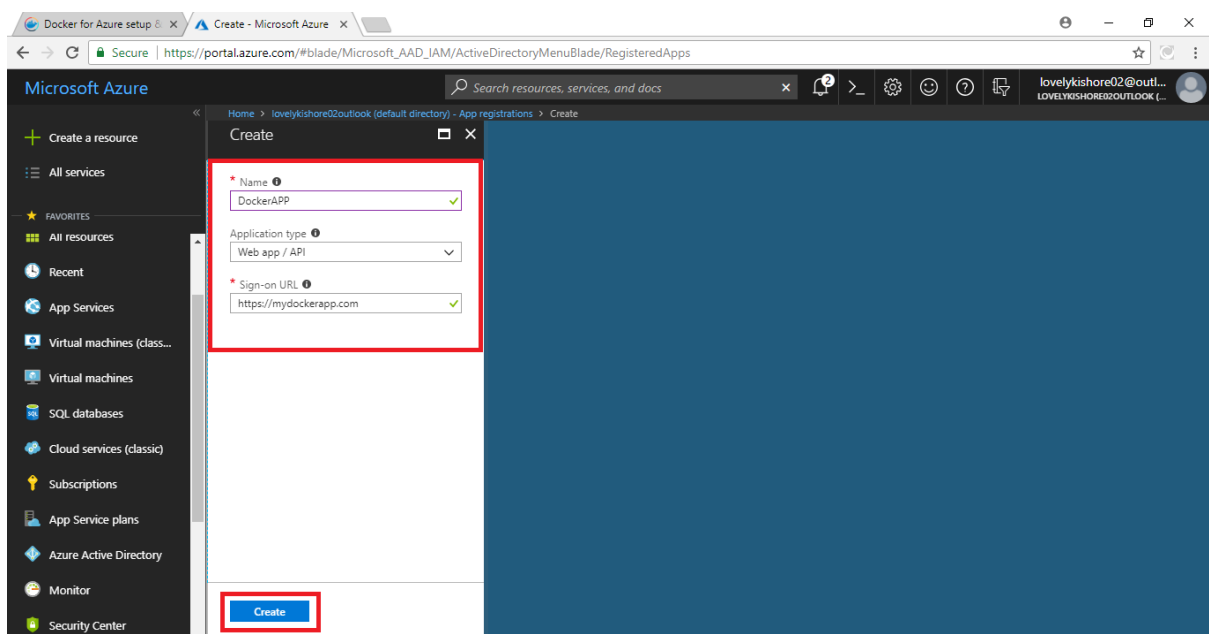
This HOL will guide you in deploying Docker Swarm along and hosting NGINX web server. Followed by the same, load balancing and scaling the number of instances will be shown.

Creating Web API app for role:

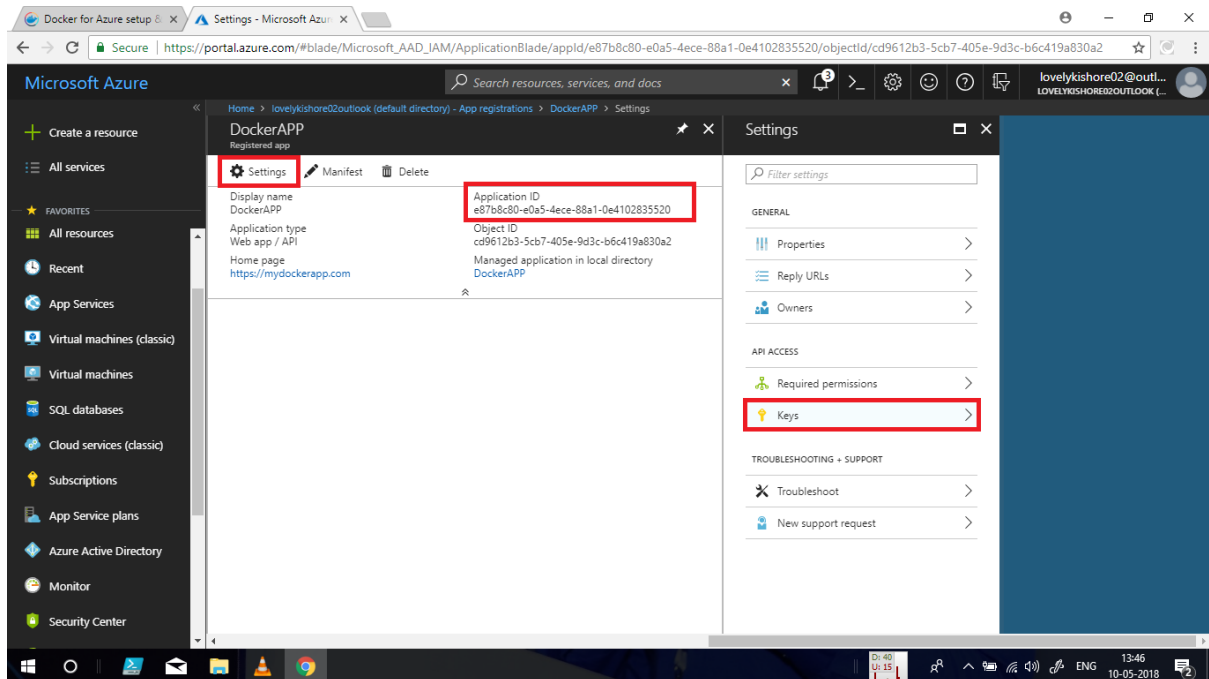
Login into your azure portal and go to **Active Directory->App Registrations->+New application registration**.



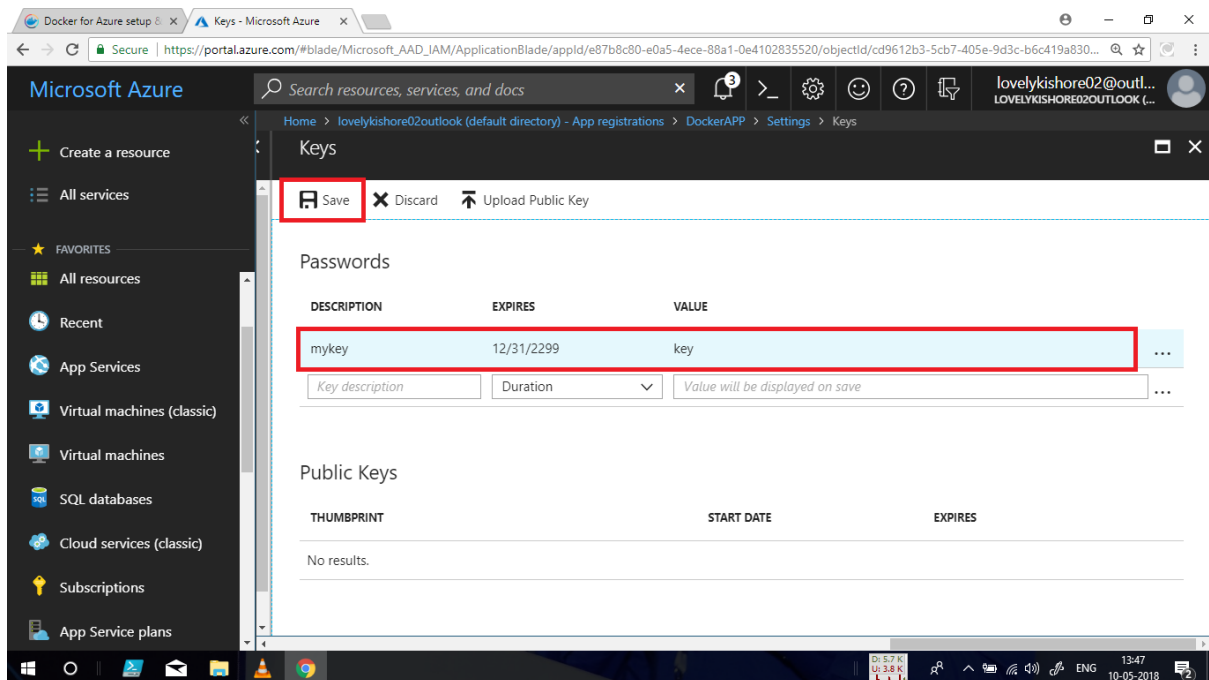
Give a name to your app and choose settings as denoted below. Finally, click on create to deploy an application.



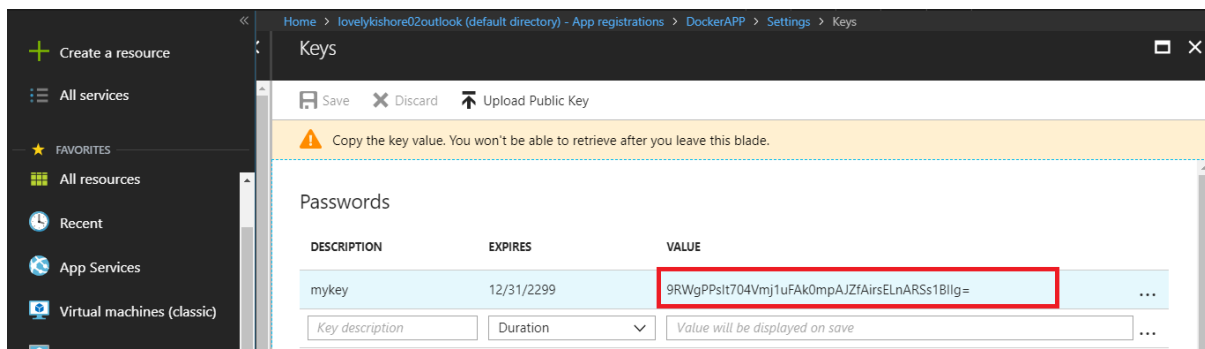
After the app gets created, make note of your **Application ID** and click on settings and go to **keys**.



Give a name for your password and choose an expiry date and click on save button to get your key.

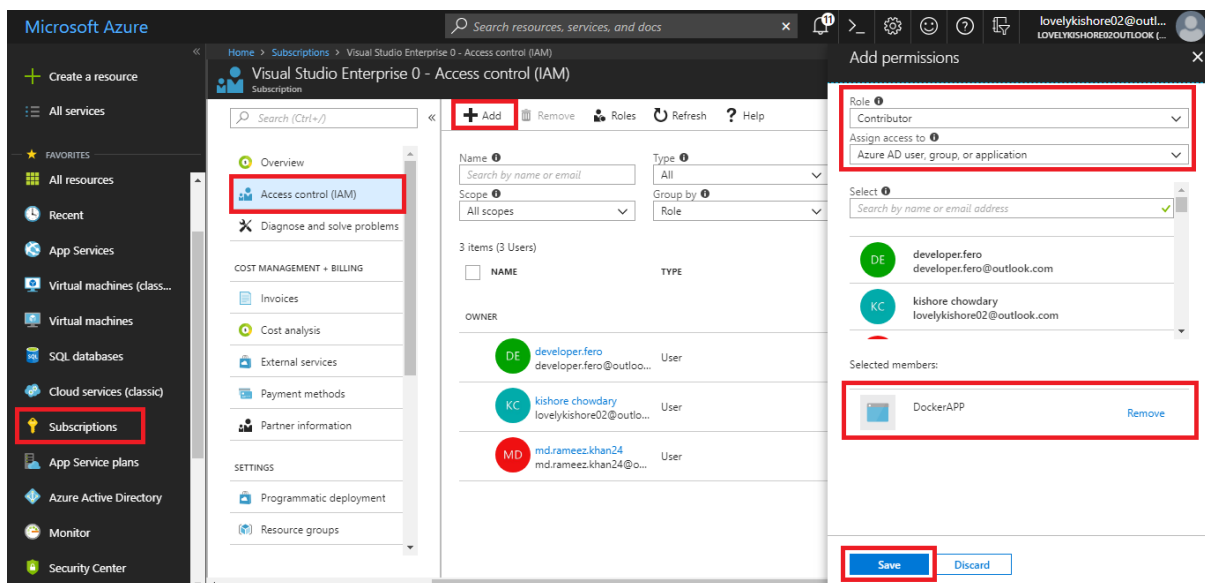


Note the key as well. It is called as app secret and you will be using it in further steps.



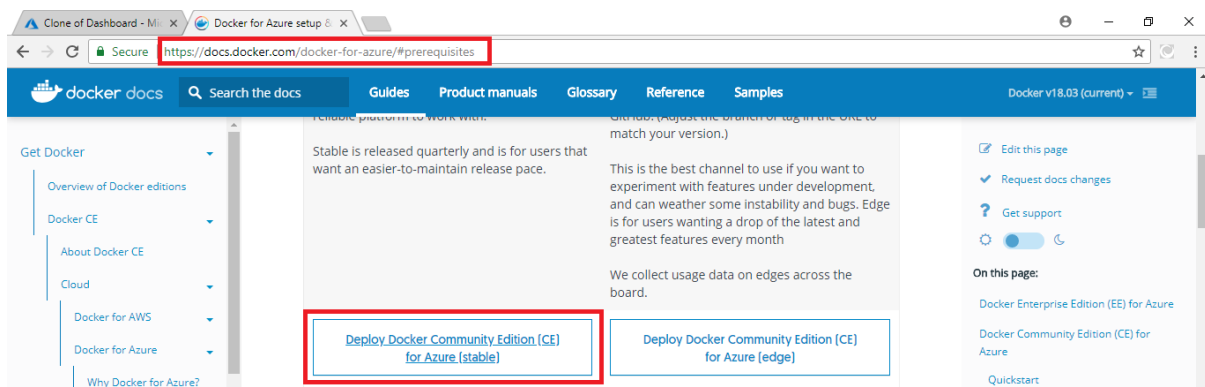
Providing access to Subscription:

Now, let us grant access to subscription for the app that we have created now. For this, go to **subscription->Access Control->Add**. Here, choose the role as contributor and select the app that you created now. Finally, click on **Save** button.



Creating Docker Swarm:

Go to the URL - <https://docs.docker.com/docker-for-azure/#quickstart> and click on the below denoted menu. This will take you into azure portal again.



Choose or create a resource group and select a suitable location. Then fill the **App ID** and **App Key** that you copied in the previous step. This is why we registered an application in AD.

Microsoft Azure

Custom deployment

Deploy from a custom template

TEMPLATE

Customized template
9 resources

Edit template Edit parameters Learn more

BASICS

* Subscription Visual Studio Enterprise 0

* Resource group Create new Use existing
DockerSwarm

* Location South India

SETTINGS

* Ad Service Principal App ID e87b8c80-e0a5-4ece-88a1-0e4102835520

* Ad Service Principal App Secret

Pin to dashboard

Purchase

Generate a public key for your Linux environment and paste it in the **Public key** box. Choose the required number of worker count, manager count and VM size. Fill the needed details, agree the terms and click on **Purchase** button. The deployment will take a minimum of 5 minutes depending on the SKUs that you choose.

Microsoft Azure

Custom deployment

Deploy from a custom template

Enable Ext Logs yes

Enable System Prune no

* Linux SSH Public Key ssh-rsa AAAAB3NzaC1yc2EAAAABQAAQEAz4cy/C2xfyfhQqdzAXZ2i/Bc8nXl...

Linux Worker Count 3

Linux Worker VM Size Standard_A0

Manager Count 3

Manager VM Size Standard_A0

Swarm Name dockerswarm

TERMS AND CONDITIONS

Azure Marketplace Terms Azure Marketplace

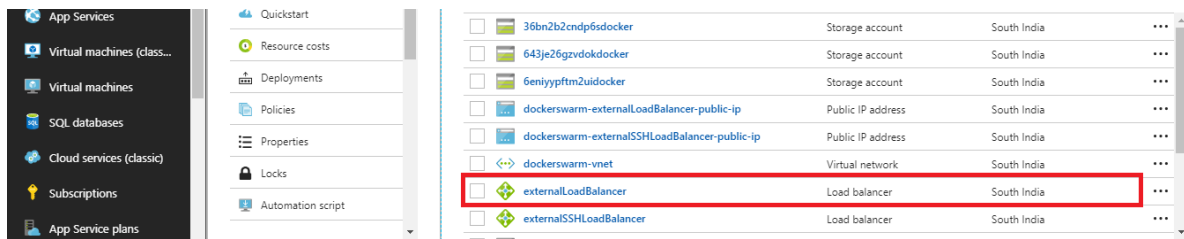
By clicking "Purchase," I agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated with the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

☒ I agree to the terms and conditions stated above.

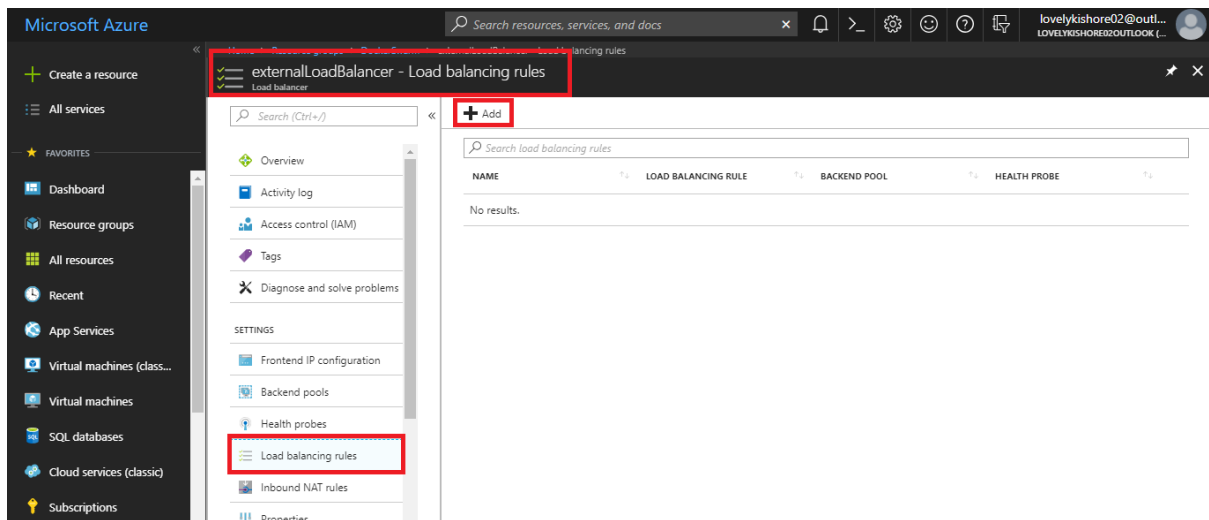
☒ Pin to dashboard

Purchase

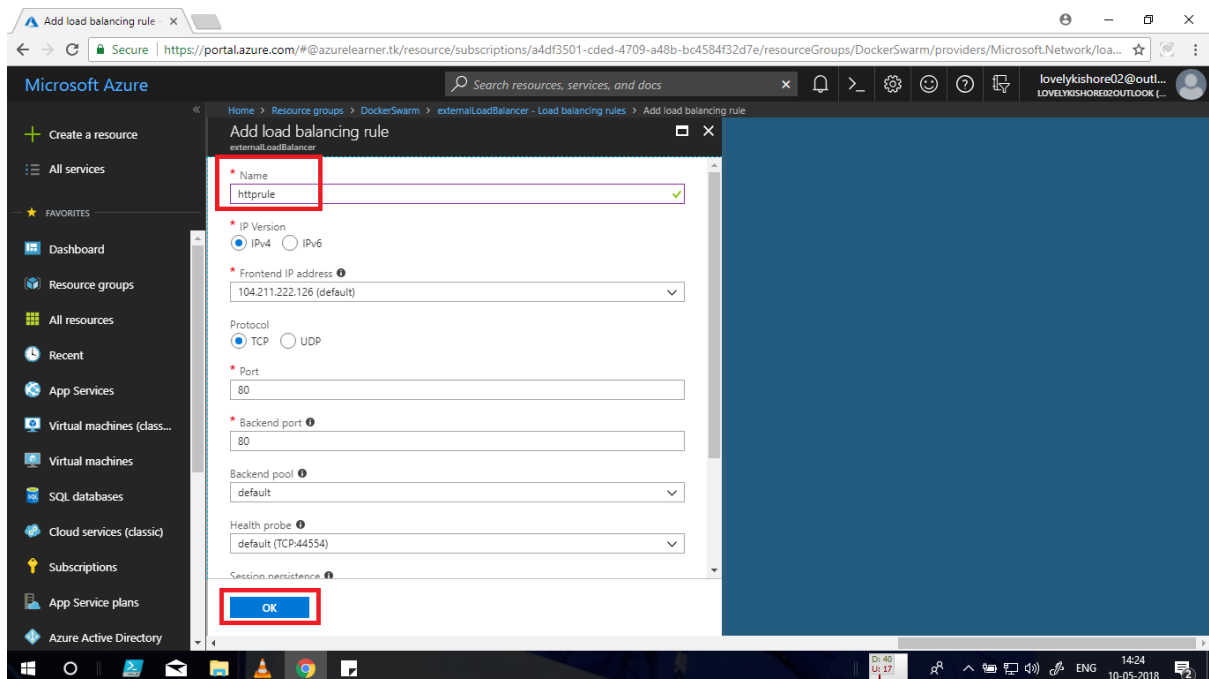
After all the resources get deployed, click on the **externalLoadBalancer**.



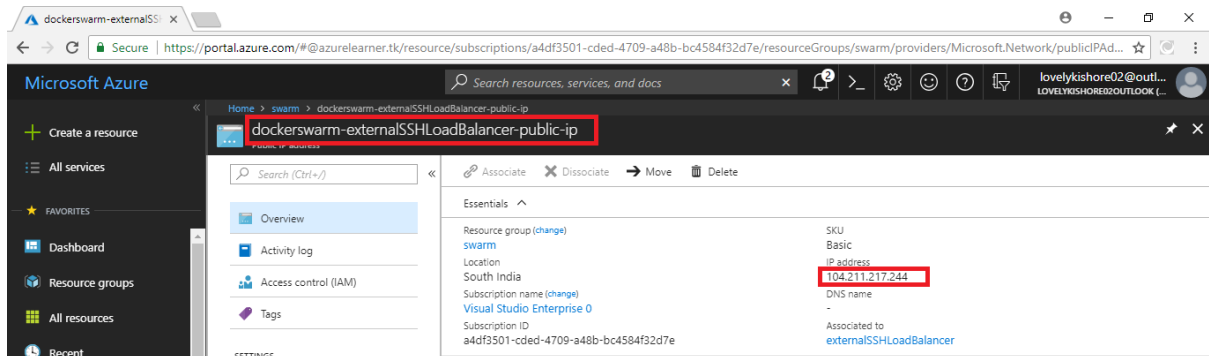
In the left side menu of **externalLoadBalancer** click on **Load Balancing Rules** and add a rule.



Give a name to the rule and click on **OK** without changing any other settings.



Now, copy the public IP address of **externalSSLoadBalancer**. Using the IP address, connect to the Linux VM using Putty.



Make sure to use **Docker** as the user name.



Run the command **docker node ls**. This will show you all the nodes.



Hosting a NGINX server:

Run this command **docker service create --name mywebsite -p 80:80 nginx** to host NGINX server in your container.



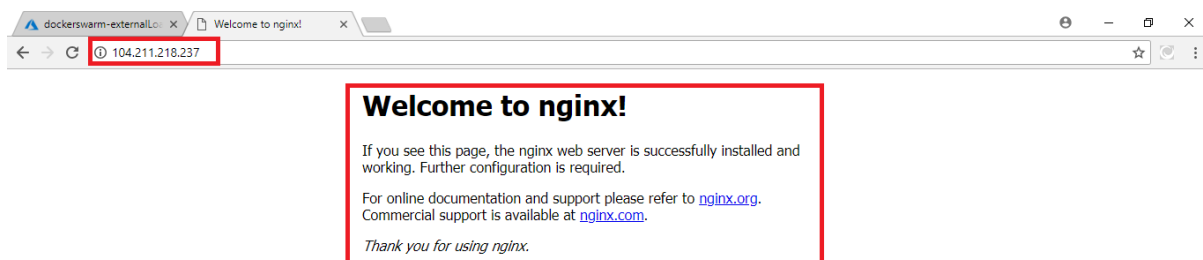
Now, run the command **docker service ls** to view the website that you hosted now.



Run the command **docker service ps my website** to see the number of replications of your site.

```
104.211.217.244 - PuTTY
swarm-manager0000000:~$
swarm-manager0000000:~$ docker service ls
ID                NAME      MODE      REPLICAS  IMAGE      PORTS
lcm4enqb779o     mywebsite replicated 1/1        nginx:latest *:80->80/tcp
swarm-manager0000000:~$ docker service ps mywebsite
ID                NAME      IMAGE      NODE           DESIRED STATE  CURRENT STATE      ERROR      PORTS
92chp2eadbuu     mywebsite.1 nginx:latest swarm-manager0000000 Running         Running 2 minutes ago          ERROR      PORTS
swarm-manager0000000:~$ ^C
swarm-manager0000000:~$
```

Ping the IP of your **externalSSLoadBalancer** in the browser. You will get response from the NGINX server.



Replicating multiple instances of website:

Run this command to get multiple replicas for your website.

docker service update --replicas 10 mywebsite

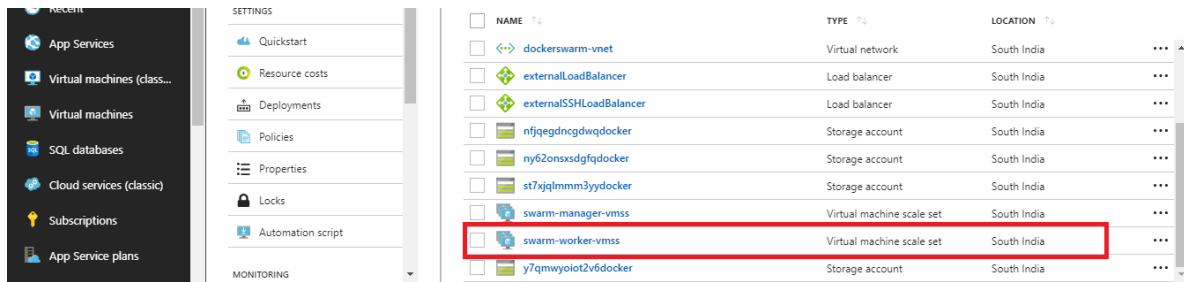
```
104.211.217.244 - PuTTY
swarm-manager0000000:~$
swarm-manager0000000:~$
swarm-manager0000000:~$
swarm-manager0000000:~$ docker service update --replicas 10 mywebsite
mywebsite
overall progress: 10 out of 10 tasks
1/10: running [=====>]
2/10: running [=====>]
3/10: running [=====>]
4/10: running [=====>]
5/10: running [=====>]
6/10: running [=====>]
7/10: running [=====>]
8/10: running [=====>]
9/10: running [=====>]
10/10: running [=====>]
verify: Service converged
swarm-manager0000000:~$
```

Now, run the command **docker service ps my website** . This will show you all the 10 replicas.

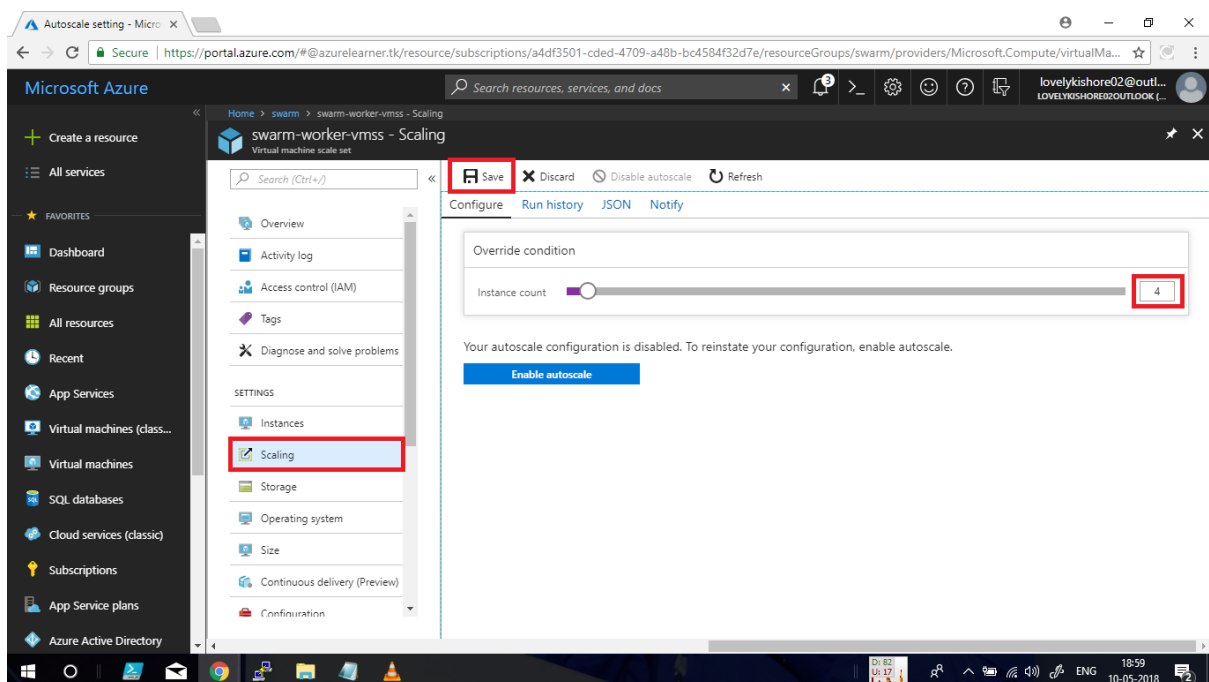
```
swarm-manager0000000:~$ docker service ps mywebsite
ID                NAME      IMAGE      NODE           DESIRED STATE  CURRENT STATE      ERROR      PORTS
92chp2eadbuu     mywebsite.1 nginx:latest swarm-manager0000000 Running         Running 37 minutes ago          ERROR      PORTS
o2ag8am578y     mywebsite.2 nginx:latest swarm-worker0000001 Running         Running 28 minutes ago          ERROR      PORTS
182fu9fezses     mywebsite.3 nginx:latest swarm-manager0000001 Running         Running 28 minutes ago          ERROR      PORTS
4jemc6u9couy     mywebsite.4 nginx:latest swarm-worker0000001 Running         Running 28 minutes ago          ERROR      PORTS
wdiyiq1s9k72     mywebsite.5 nginx:latest swarm-manager0000000 Running         Running 28 minutes ago          ERROR      PORTS
nuk5am5vxje     mywebsite.6 nginx:latest swarm-manager0000002 Running         Running 28 minutes ago          ERROR      PORTS
r3isl14vpmsms     mywebsite.7 nginx:latest swarm-worker0000000 Running         Running 28 minutes ago          ERROR      PORTS
u59yc9ef38xsl     mywebsite.8 nginx:latest swarm-worker0000002 Running         Running 28 minutes ago          ERROR      PORTS
oxe5u5u21fsm3     mywebsite.9 nginx:latest swarm-manager0000002 Running         Running 28 minutes ago          ERROR      PORTS
v8stci2fcol3     mywebsite.10 nginx:latest swarm-worker0000000 Running         Running 28 minutes ago          ERROR      PORTS
swarm-manager0000000:~$
```

Scaling Instances of Worker VMSS:

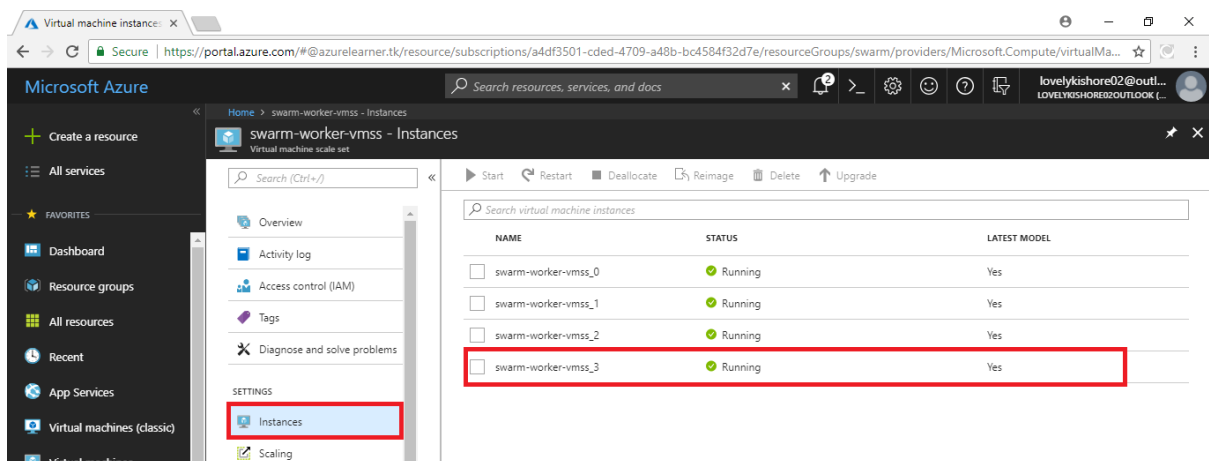
Go to the **Worker-VMSS** in the resources that you created initially.



In the left side menu, click on **Scaling** and change the instance count to your requirement and click on save button.



Click on the instances menu in the left side. Wait for the additional instances to spin up. In a while, you will be able to find the newly scaled instances.



At last run the `docker service ps mywebsite` command to get the state of all your containers. Again run the command `docker service update --replicas 15 mywebsite`. This will be spinning up another 5 instances for your **NGINX** server.

```
104.211.217.244 - PuTTY
inspect      Display detailed information on one or more services
logs         Fetch the logs of a service or task
ls           List services
ps           List the tasks of one or more services
rm           Remove one or more services
rollback     Revert changes to a service's configuration
scale        Scale one or multiple replicated services
update       Update a service

Run 'docker service COMMAND --help' for more information on a command.

swarm-manager000000:~$ docker service ps mywebsite
ID                NAME                IMAGE                NODE                DESIRED STATE       CURRENT STATE        ERROR                PORTS
92cbp2eadbuu      mywebsite.1          nginx:latest         swarm-manager000000 Running             Running about an hour ago
otagh8am878y      mywebsite.2          nginx:latest         swarm-worker000001 Running             Running about an hour ago
182fu9feszsz      mywebsite.3          nginx:latest         swarm-manager000001 Running             Running about an hour ago
4jemc6u9couy      mywebsite.4          nginx:latest         swarm-worker000001 Running             Running about an hour ago
wdiyiq1s9k72      mywebsite.5          nginx:latest         swarm-manager000000 Running             Running about an hour ago
nui1k5am5wxje     mywebsite.6          nginx:latest         swarm-manager000002 Running             Running about an hour ago
r3isl14wpms       mywebsite.7          nginx:latest         swarm-worker000000 Running             Running about an hour ago
u9yc96f38xl       mywebsite.8          nginx:latest         swarm-worker000002 Running             Running about an hour ago
oxe5us2i29m3      mywebsite.9          nginx:latest         swarm-manager000002 Running             Running about an hour ago
vnstti2fsc13      mywebsite.10         nginx:latest         swarm-worker000000 Running             Running about an hour ago
swarm-manager000000:~$ docker service update --replicas 15 mywebsite
mywebsite
overall progress: 15 out of 15 tasks
1/15: running [=====]
2/15: running [=====]
3/15: running [=====]
4/15: running [=====]
5/15: running [=====]
6/15: running [=====]
7/15: running [=====]
8/15: running [=====]
9/15: running [=====]
10/15: running [=====]
11/15: running [=====]
12/15: running [=====]
13/15: running [=====]
14/15: running [=====]
15/15: running [=====]
verify: Service converged
swarm-manager000000:~$
```

At last run the command `docker service ps mywebsite` to get details about every process and the container instance in which it is running.

```
104.211.217.244 - PuTTY
swarm-manager000000:~$
swarm-manager000000:~$
swarm-manager000000:~$ docker service ps mywebsite
ID                NAME                IMAGE                NODE                DESIRED STATE       CURRENT STATE        ERROR                PORTS
92cbp2eadbuu      mywebsite.1          nginx:latest         swarm-manager000000 Running             Running about an hour ago
otagh8am878y      mywebsite.2          nginx:latest         swarm-worker000001 Running             Running about an hour ago
182fu9feszsz      mywebsite.3          nginx:latest         swarm-manager000001 Running             Running about an hour ago
4jemc6u9couy      mywebsite.4          nginx:latest         swarm-worker000001 Running             Running about an hour ago
wdiyiq1s9k72      mywebsite.5          nginx:latest         swarm-manager000000 Running             Running about an hour ago
nui1k5am5wxje     mywebsite.6          nginx:latest         swarm-manager000002 Running             Running about an hour ago
r3isl14wpms       mywebsite.7          nginx:latest         swarm-worker000000 Running             Running about an hour ago
u9yc96f38xl       mywebsite.8          nginx:latest         swarm-worker000002 Running             Running about an hour ago
oxe5us2i29m3      mywebsite.9          nginx:latest         swarm-manager000002 Running             Running about an hour ago
vnstti2fsc13      mywebsite.10         nginx:latest         swarm-worker000000 Running             Running about an hour ago
as5c4vxi2rv3      mywebsite.11         nginx:latest         swarm-worker000003 Running             Running 52 seconds ago
yztuq58i698q      mywebsite.12         nginx:latest         swarm-worker000002 Running             Running about a minute ago
fchj38ztzqgh1     mywebsite.13         nginx:latest         swarm-manager000001 Running             Running about a minute ago
qubuxrx081kz      mywebsite.14         nginx:latest         swarm-manager000001 Running             Running about a minute ago
lundmhtja7da      mywebsite.15         nginx:latest         swarm-worker000003 Running             Running 50 seconds ago
swarm-manager000000:~$
```