



# Microsoft Cloud Workshop

Linux lift and shift

Hands-on lab step-by-step

January 2018

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2018 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners

# Contents

Linux lift and shift hands-on lab step-by-step .....	1
Abstract and learning objectives.....	1
Overview.....	1
Requirements .....	2
Help References.....	2
Before the hands-on lab.....	3
Task 1: Create a virtual machine to execute the lab.....	3
Task 2: Install the MySQL Workbench .....	8
Exercise 1: Deploy the on-premises OsTicket VM application.....	9
Task 1: Deploy the OnPremVM .....	9
Task 2: Export the osticket database.....	12
Exercise 2: Migrate to Azure IaaS VM Scale Sets and MySQL cluster .....	17
Task 1: Deploy the MySQL HA Cluster .....	17
Task 2: Connect to the MySQL cluster and restore the database.....	18
Task 3: Deploy the Virtual Machine Scale Set for the OsTicket Application .....	23
Task 4: Connect the MySQLVNet to the Scale Sets VNet .....	26
Task 5: Export the osticket database from the MySQL cluster .....	30
Exercise 3: Migrate the OsTicket application from Azure IaaS to PaaS.....	33
Task 1: Create the MySQL database .....	33
Task 2: Restore the osticket database to MySQL PaaS .....	35
Task 3: Create the Web App .....	40
Task 4: Configure the OsTicket Web App .....	41
After the hands-on lab.....	48



# Linux lift and shift hands-on lab step-by-step

## Abstract and learning objectives

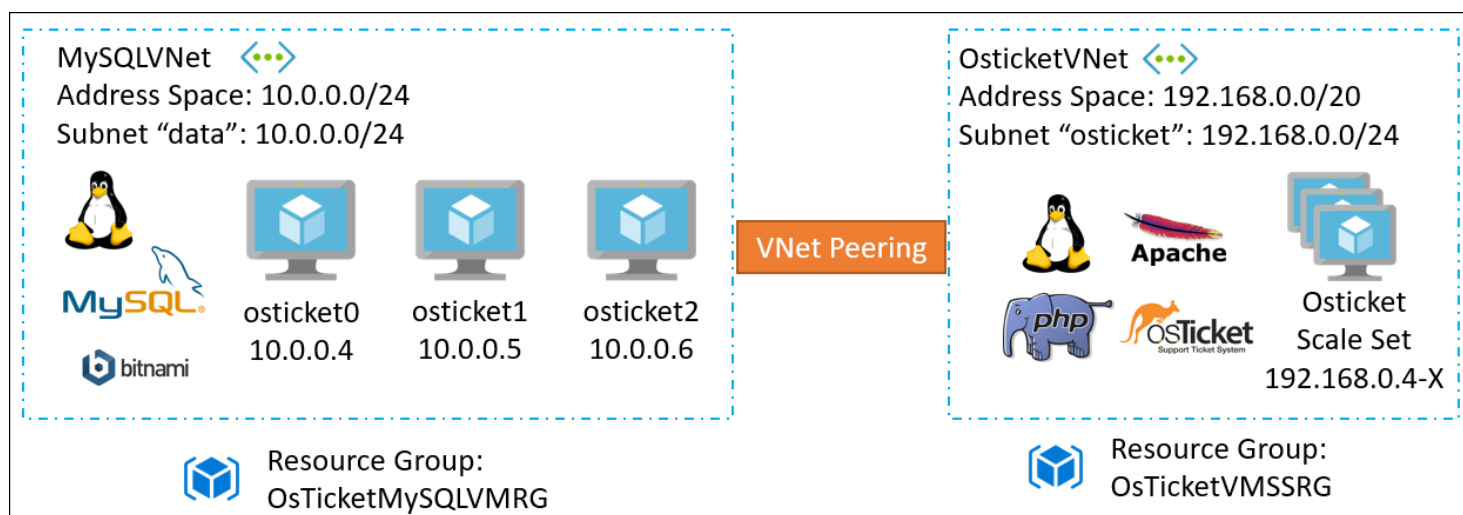
In this Step-by-step Microsoft Cloud Workshop, you will migrate a Linux based application to the Azure Cloud. This will include the use of Azure IaaS Virtual Machines and Virtual Machine Scale Sets. Azure PaaS will also be leveraged including: Azure App Services (Web App), and Azure Database for MySQL. The student will leverage Azure Resource Manager templates, the Linux custom script extension, Github and a Linux Docker Container in the App Service.

## Overview

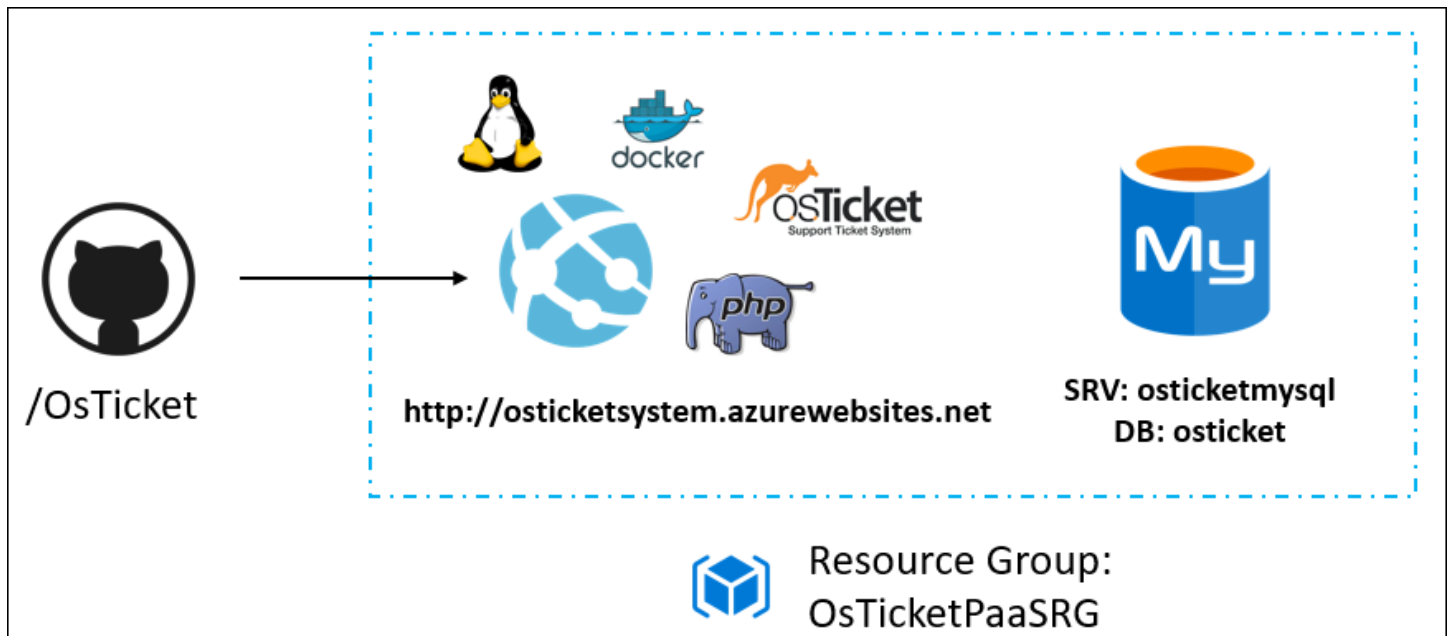
In this hands-on Lab Step by Step, you will migrate an on-premises based helpdesk application called OsTicket to Azure. This will be a two-phase project to lift and shift the application into Azure IaaS and then migrate it to Azure PaaS. The application is Linux based using Apache, PHP and MySQL (LAMP). During the process of these Phases you will ensure zero data loss.

- **Phase I:** Lift and Shift the Application from on-premises to Azure IaaS using an auto scaling Virtual Machine Scale Set and a MySQL cluster with 3 nodes.
- **Phase II:** Migrate to PaaS using Azure App Services with a Linux Docker Container and Azure Database for MySQL.

Phase I will result in an environment resembling this diagram:



Phase II will result in an environment resembling this diagram:



## Requirements

You must have a working Azure subscription to carry out this hands-on Lab step-by-step.

## Help References

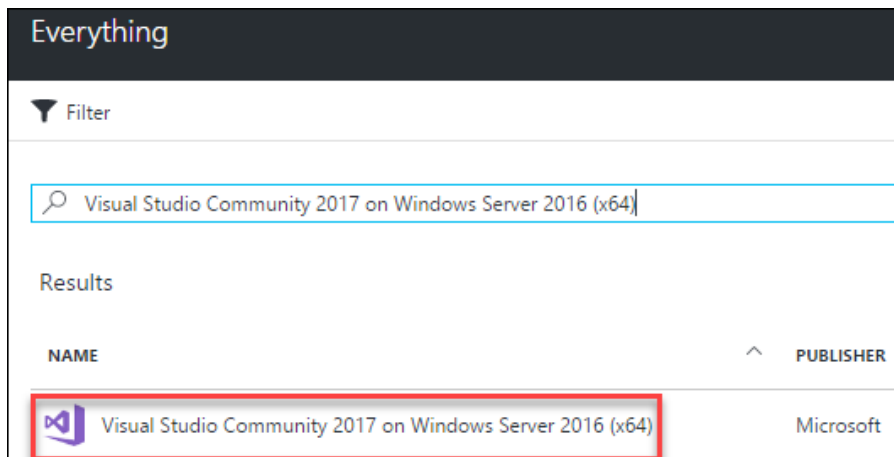
Description	Links
Azure Database for MySQL	<a href="https://docs.microsoft.com/en-us/azure/mysql/">https://docs.microsoft.com/en-us/azure/mysql/</a>
Azure Database for MySQL	<a href="https://docs.microsoft.com/en-us/azure/mysql/howto-create-manage-server-portal">https://docs.microsoft.com/en-us/azure/mysql/howto-create-manage-server-portal</a>
Azure Database for MySQL	<a href="https://docs.microsoft.com/en-us/azure/mysql/howto-manage-firewall-using-portal">https://docs.microsoft.com/en-us/azure/mysql/howto-manage-firewall-using-portal</a>
Connect Azure Web App to Azure Database for MySQL	<a href="https://docs.microsoft.com/en-us/azure/mysql/howto-connect-webapp">https://docs.microsoft.com/en-us/azure/mysql/howto-connect-webapp</a>
Virtual Machine Scale Sets	<a href="https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/">https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/</a>
Using VM Extensions with Virtual Machine Scale Sets	<a href="https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-deploy-scaling-app-template">https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-deploy-scaling-app-template</a>
App Service for Linux	<a href="https://docs.microsoft.com/en-us/azure/app-service/containers/app-service-linux-intro">https://docs.microsoft.com/en-us/azure/app-service/containers/app-service-linux-intro</a>
Azure CLI	<a href="https://docs.microsoft.com/en-us/cli/azure/install-azure-cli">https://docs.microsoft.com/en-us/cli/azure/install-azure-cli</a>

## Before the hands-on lab

Duration: 30 Minutes

### Task 1: Create a virtual machine to execute the lab

1. Launch a browser and navigate to <https://portal.azure.com>. Once prompted, login with your Microsoft Azure credentials. If prompted, choose whether your account is an organization account or just a Microsoft Account.
2. Click on **+NEW**, and in the search box type in **Visual Studio Community 2017 on Windows Server 2016 (x64)** and press enter. Click the Visual Studio Community 2017 image running on Windows Server 2016 and with the latest update.
3. In the returned search results, click the image name.



4. Click **Create**.
5. Set the following configuration on the Basics tab and click **OK**.
  - Name: **LABVM**
  - VM disk type: **SSD**
  - User name: **demouser**
  - Password: **demo@pass123**
  - Subscription: **If you have multiple subscriptions choose the subscription to execute your labs in.**
  - Resource Group: **OPSLABRG**

- Location: **Choose the closest Azure region to you.**

The screenshot shows the 'Basics' configuration page for an Azure Virtual Machine. The following fields are highlighted with red boxes:

- Name:** LABVM (with a green checkmark)
- VM disk type:** SSD (with a dropdown arrow)
- User name:** demouser
- Password:** (masked with dots)
- Confirm password:** (masked with dots)
- Subscription:** (dropdown menu)
- Resource group:** Create new (selected), OPSLABRG (with a green checkmark)
- Location:** South Central US (with a dropdown arrow)

6. Choose the **DS1\_V2 Standard** instance size on the Size blade.

**Note:** You may have to click the View All link to see the instance sizes.



**Choose a size**  
Browse the available sizes and their features

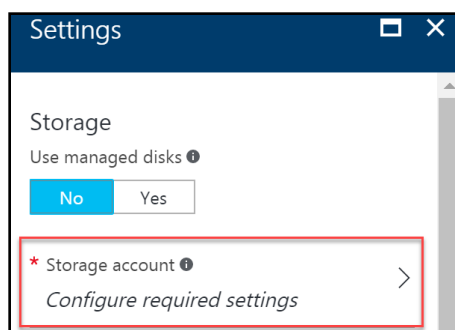
Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Recommended sizes are determined by the publisher of the selected image based on hardware and software requirements.

★ Recommended | View all

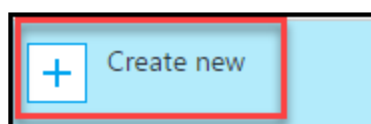
DS1_V2 Standard		DS2_V2 Standard ★		DS3_V2 Standard	
1	Core	2	Cores	4	Cores
3.5	GB	7	GB	14	GB
2	Data disks	4	Data disks	8	Data disks
3200	Max IOPS	6400	Max IOPS	12800	Max IOPS
7 GB	Local SSD	14 GB	Local SSD	28 GB	Local SSD
Load balancing		Load balancing		Load balancing	
Premium disk support		Premium disk support		Premium disk support	
47.62		94.49		189.72	
USD/MONTH (ESTIMATED)		USD/MONTH (ESTIMATED)		USD/MONTH (ESTIMATED)	

**Note:** If the Azure Subscription you are using is NOT a trial Azure subscription you may want to chose the DS2\_V2 to have more power in this LABMV. If you are using a Trial Subscription or one that you know has a restriction on the number of cores stick with the DS1\_V2.

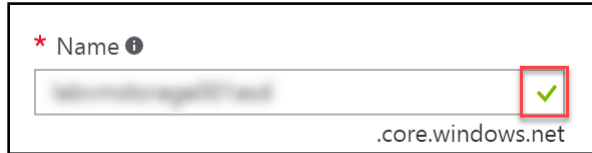
- Click **Storage Account** *Configure required settings* to specify a storage account for your virtual machine if a storage account name is not automatically selected for you.



- Click **Create New**

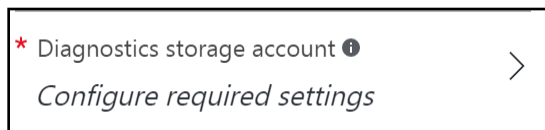


9. Specify a unique name for the storage account (all lower letters and alphanumeric characters) and ensure the green checkmark shows the name is valid.



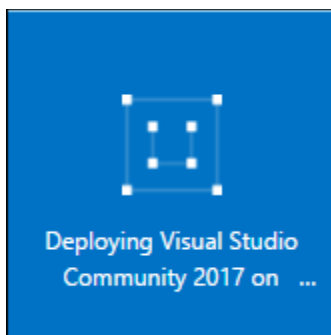
A screenshot of a web form for creating a storage account. It features a text input field with a red asterisk and the label "Name" followed by an information icon. The input field contains a blurred alphanumeric string. To the right of the input field is a green checkmark icon, which is highlighted by a red square. Below the input field, the text ".core.windows.net" is visible.

10. Click **OK** to continue.
11. Click **Diagnostics Storage Account** *Configure required settings* for the Diagnostics storage account if a storage account name is not automatically selected for you. Repeat the previous steps to select a unique storage account name. This storage account will hold diagnostic logs about your virtual machine that you can use for troubleshooting purposes.



A screenshot of a button for configuring the Diagnostics storage account. The button has a red asterisk, the text "Diagnostics storage account" followed by an information icon, and the text "Configure required settings" in italics. A right-pointing chevron icon is located to the right of the text.

12. Accept the remaining default values on the Settings blade and click **OK**. On the Summary page click **Create**. The deployment should begin provisioning. It may take 10+ minutes for the virtual machine to complete provisioning.



**Note:** Please wait for the LABVM to be provisioned prior to moving to the next step.

13. Move back to the Portal page on your local machine and wait for **LABVM** to show the Status of **Running**. Click **Connect** to establish a new Remote Desktop Session.

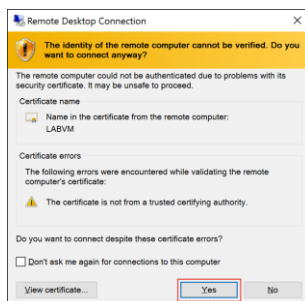


14. Depending on your remote desktop protocol client and browser configuration you will either be prompted to open an RDP file, or you will need to download it and then open it separately to connect.

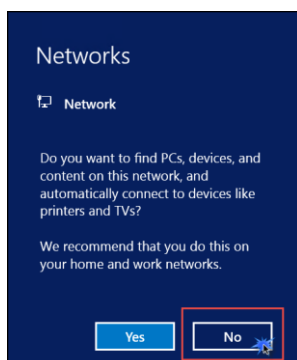
15. Log in with the credentials specified during creation:

- a. User: **demouser**
- b. Password: **demo@pass123**

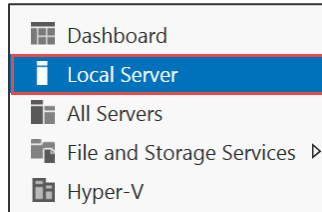
16. You will be presented with a Remote Desktop Connection warning because of a certificate trust issue. Click **Yes** to continue with the connection.



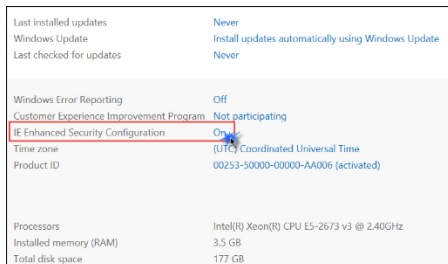
17. When logging on for the first time you will see a prompt on the right asking about network discovery. Click **No**.



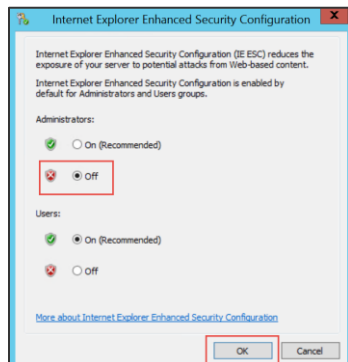
18. Notice that Server Manager opens by default. On the left, click **Local Server**.



19. On the right side of the pane, click **On** by **IE Enhanced Security Configuration**.



20. Change to **Off** for Administrators and click **OK**.



## Task 2: Install the MySQL Workbench

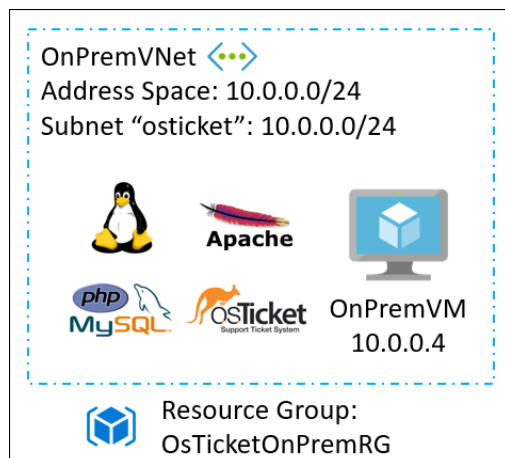
1. While logged into **LABVM** via remote desktop, open Internet Explorer and navigate to <https://dev.mysql.com/get/Downloads/MySQLGUITools/mysql-workbench-community-6.3.10-winx64.msi> this will download an executable. After the download is finished, click **Run** to execute it.
2. Follow the directions of the installer to complete the installation of MySQL Workbench.
3. After the installation is complete, **reboot** the machine.

You should follow all steps provided *before* attending the Hands-on lab.

## Exercise 1: Deploy the on-premises OsTicket VM application

Duration: 45 Minutes

In this exercise, you will deploy a VM using an ARM template that will act as the on-premises installation of the OsTicket application. This will consist of a single Ubuntu Linux 16.04-LTS VM with Apache, PHP and MySQL installed (LAMP). This application is a helpdesk management software that you will lift and shift into Azure. There is sample data in the application that will be retained throughout the two phases of the project.



### Task 1: Deploy the OnPremVM

1. From the Azure portal, click on the Cloud Shell icon on the top navigation.



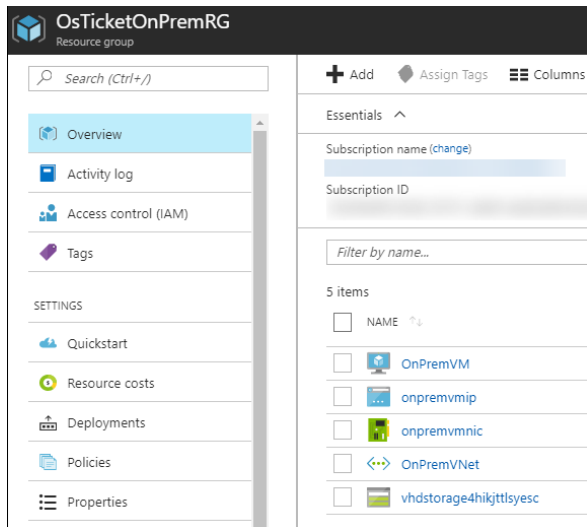
2. Execute the following command to create a resource group that will contain the application. Note: You can also specify an alternate region.

```
az group create --name OsTicketOnPremRG --location "East US"
```

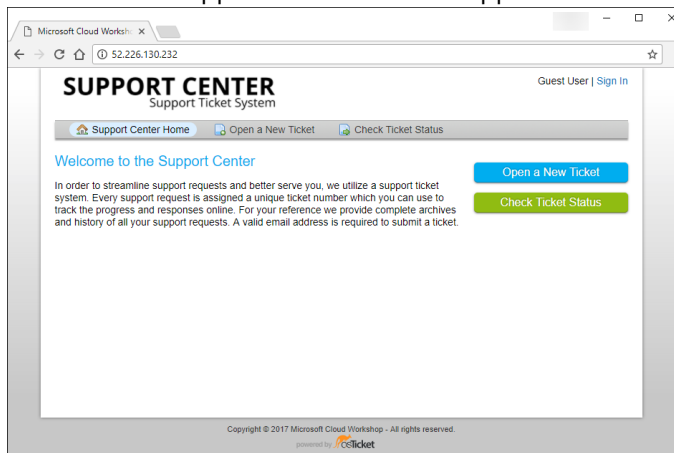
3. Execute the following command to deploy the ARM template.

```
az group deployment create --name OsTicketOnPremRG --resource-group  
OsTicketOnPremRG --template-uri "  
https://cloudworkshop.blob.core.windows.net/linux-lift-  
shift/onprevmdeploy.json"
```

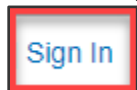
4. This deployment will take about 5 minutes to complete. Wait until it has been deployed before moving on to the next step.
5. Once the deployment has completed open the resource group **OsTicketOnPremRG** and review the deployment.



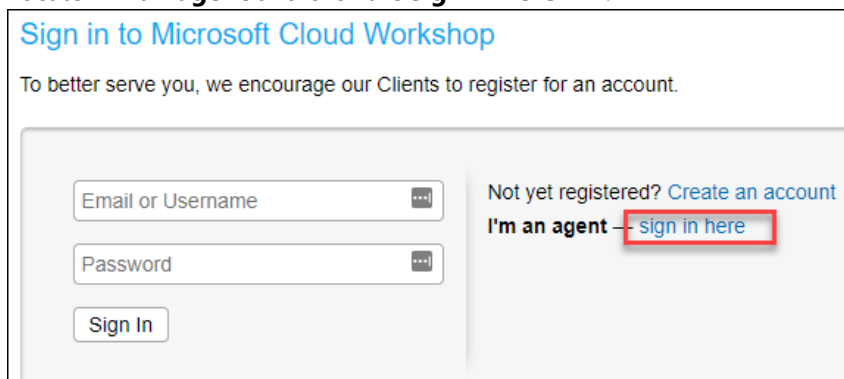
6. Next click on the **onprevmip** Public IP address. Locate the IP address and paste this into a new tab of your web browser. The Support Center OsTicket application should load.



7. Click the Sign in link.

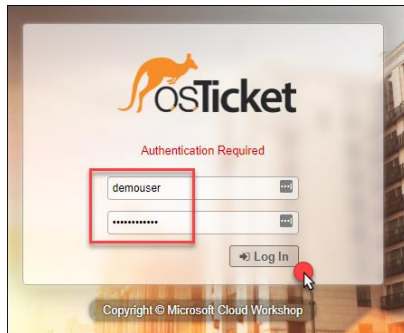


8. Locate **I'm an agent** and click the **sign in here** link.

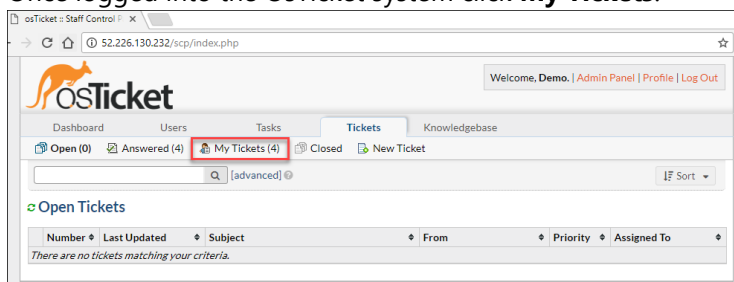


9. At the OsTicket screen enter the **username** and **password** and click **Log In**.

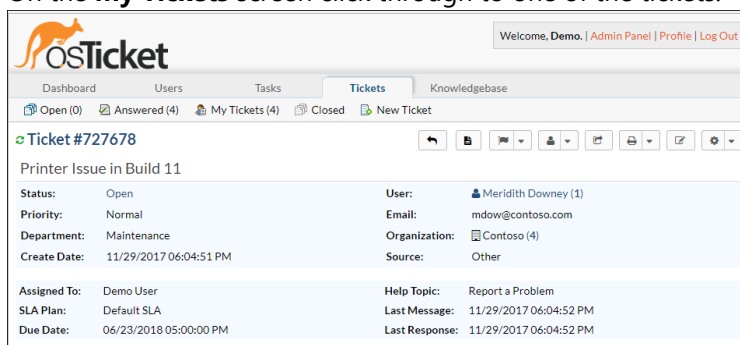
- a. Username: **demouser**
- b. Password: **demo@pass123**



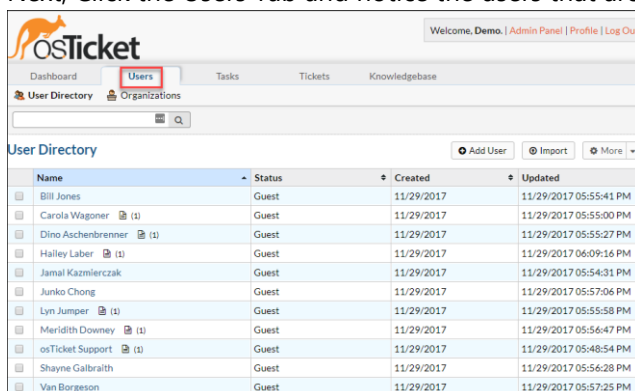
10. Once logged into the OsTicket system click **My Tickets**.



11. On the **My Tickets** screen click through to one of the tickets.



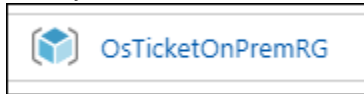
12. Next, Click the Users Tab and notice the users that are entered into the system.



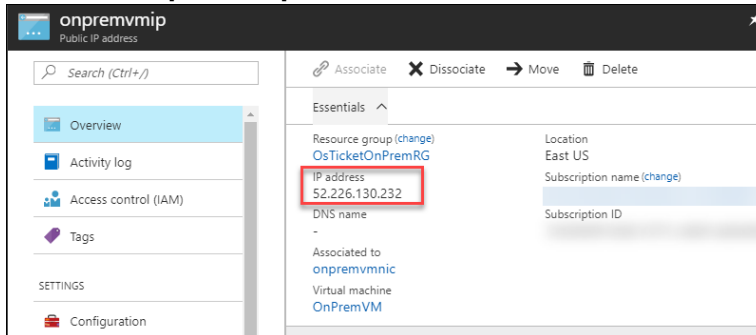
13. Feel free to create new tickets or new users to add to your dataset.

## Task 2: Export the osticket database

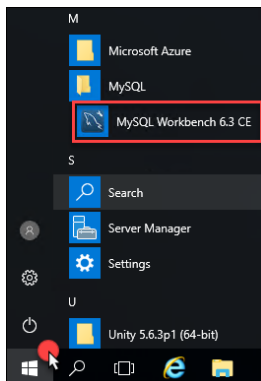
1. From your **LABVM**, connect to the Azure portal and then open the **OsTicketOnPremRG** resource group.



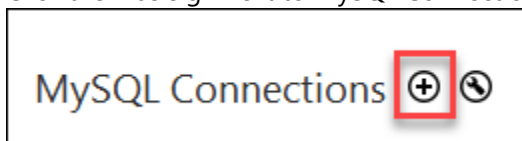
2. Locate the **onpremvmpip** Public IP address and **take note of the address**.



3. Next on the LABVM click Start and then locate the MySQL Workbench.



4. Click the Plus sign next to MySQL Connections on the Workbench.



5. Enter the following information to configure to connect to your OnPremVM.
  - a. Connection Name: **OnPremVM**
  - b. Connection Method: **Standard TCP/IP over SSH**
  - c. SSH Hostname: **<enter the Public ip address>**
  - d. SSH Username: **demouser**
  - e. SSH Password: **Click Store in Vault: demo@pass123**
  - f. MySQL Hostname: **127.0.0.1**
  - g. MySQL Server Port: **3306**
  - h. Username: **osticket**



- i. Password: **Click Store in Vault: demo@pass123**

Setup New Connection

Connection Name: OnPremVM

Connection Method: Standard TCP/IP over SSH

SSH Hostname: 52.226.130.232:22

SSH Username: demouser

SSH Password: demo@pass123

MySQL Hostname: 127.0.0.1

MySQL Server Port: 3306

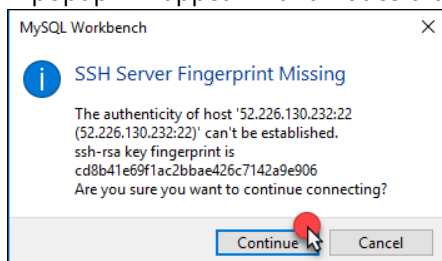
Username: osticket

Password: demo@pass123

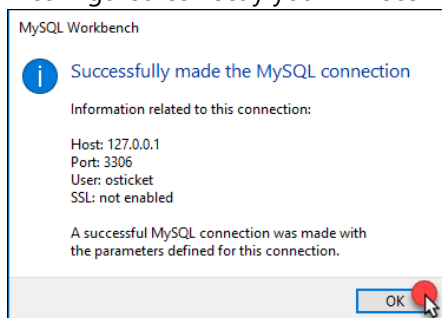
Default Schema:

Test Connection

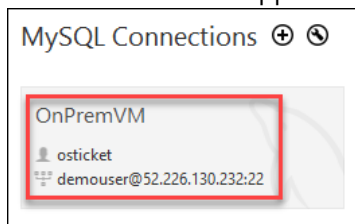
6. Once configured click **Test Connection**.
7. A popup will appear with a notice that the **SSH Server Fingerprint Missing**, click **continue**.



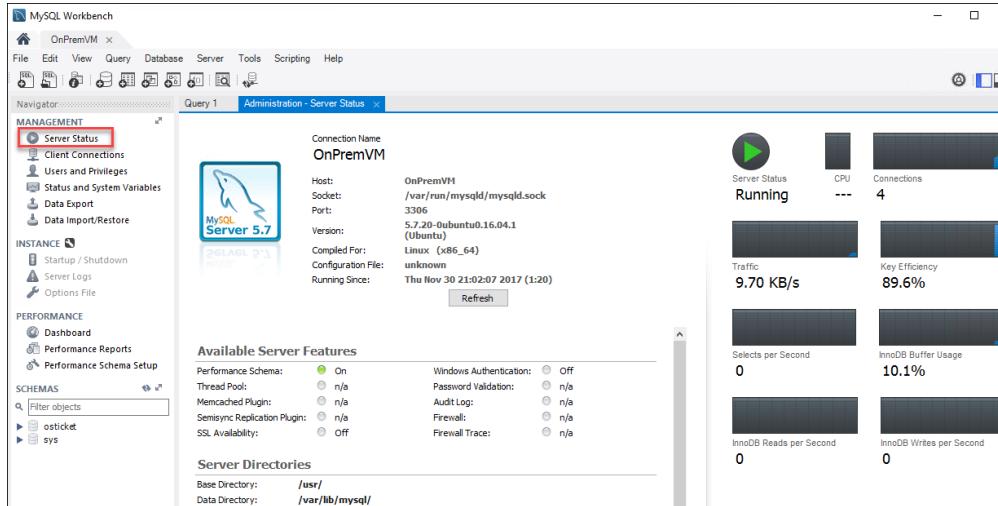
8. If configured correctly you will receive a message: **Successfully made the MySQL Connection**. Click OK.



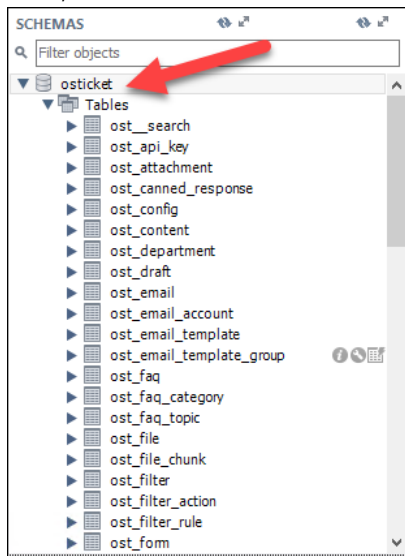
9. The Connection will appear. Double-click to start a session with the MySQL instance running on the **OnPremVM**.



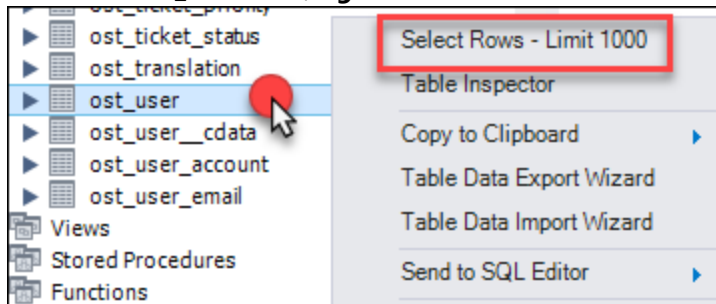
10. Once the Workbench loads, click **Server Status**. Review the details of the server.



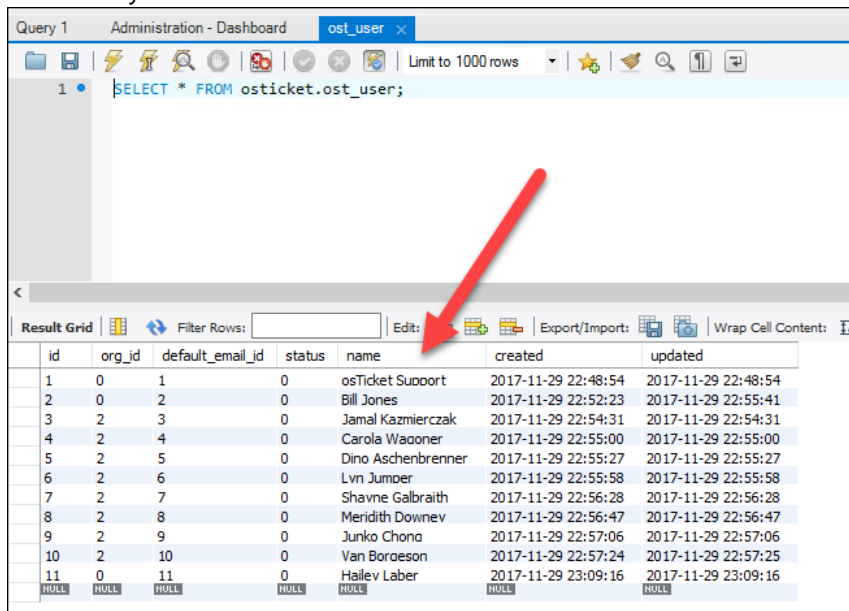
11. Next, click the **osticket** database under Schemas and expand to see the tables.



12. Locate the **ost\_user** table, **right-click** and then click **Select Rows – Limit 1000**.



13. This will launch a query in the Workbench and list all of the users on the system. Notice that these are the same users that you saw in the OsTicket UI.



Query 1 Administration - Dashboard ost\_user x

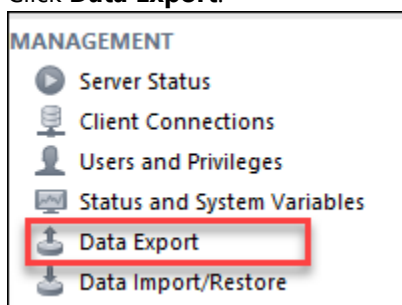
Limit to 1000 rows

1 • `SELECT * FROM osticket.ost_user;`

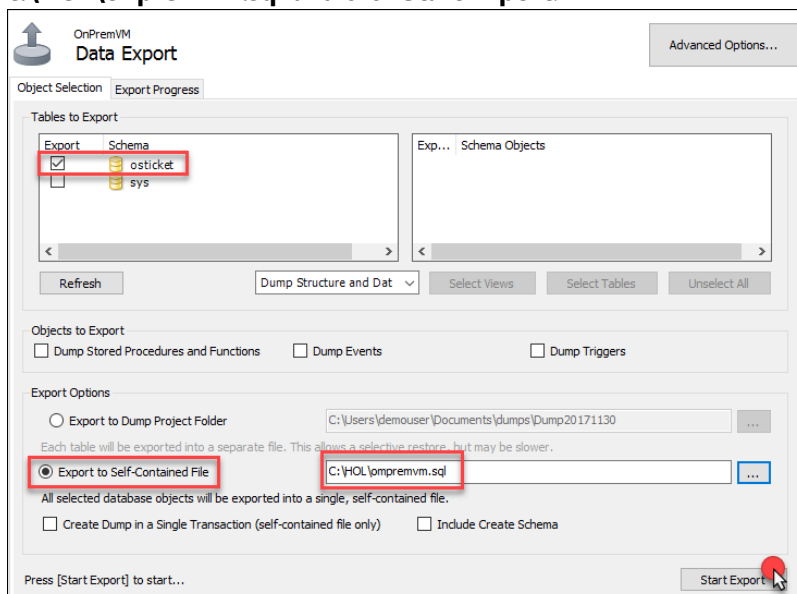
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	org_id	default_email_id	status	name	created	updated
1	0	1	0	0	osTicket Support	2017-11-29 22:48:54	2017-11-29 22:48:54
2	0	2	0	0	Bill Jones	2017-11-29 22:52:23	2017-11-29 22:55:41
3	2	3	0	0	Jamal Kazmierczak	2017-11-29 22:54:31	2017-11-29 22:54:31
4	2	4	0	0	Carola Waggoner	2017-11-29 22:55:00	2017-11-29 22:55:00
5	2	5	0	0	Dino Aschenbrenner	2017-11-29 22:55:27	2017-11-29 22:55:27
6	2	6	0	0	Lvn Jumper	2017-11-29 22:55:58	2017-11-29 22:55:58
7	2	7	0	0	Shavne Galbraith	2017-11-29 22:56:28	2017-11-29 22:56:28
8	2	8	0	0	Meridith Downey	2017-11-29 22:56:47	2017-11-29 22:56:47
9	2	9	0	0	Junko Chono	2017-11-29 22:57:06	2017-11-29 22:57:06
10	2	10	0	0	Van Boroeseon	2017-11-29 22:57:24	2017-11-29 22:57:25
11	0	11	0	0	Hailev Laber	2017-11-29 23:09:16	2017-11-29 23:09:16
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

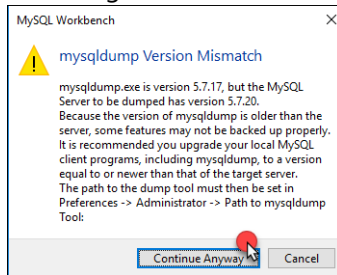
14. Click **Data Export**.



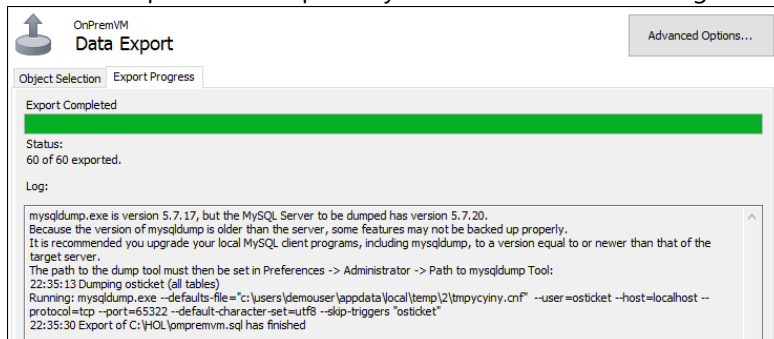
15. On the **Data Export** screen, select the **osticket**. Then select **Export to Self-Contained File**, and named the file **c:\HOL\onpremvmsql** and click **Start Export**.



16. You will get a version mismatch warning, click **Continue Anyway**.



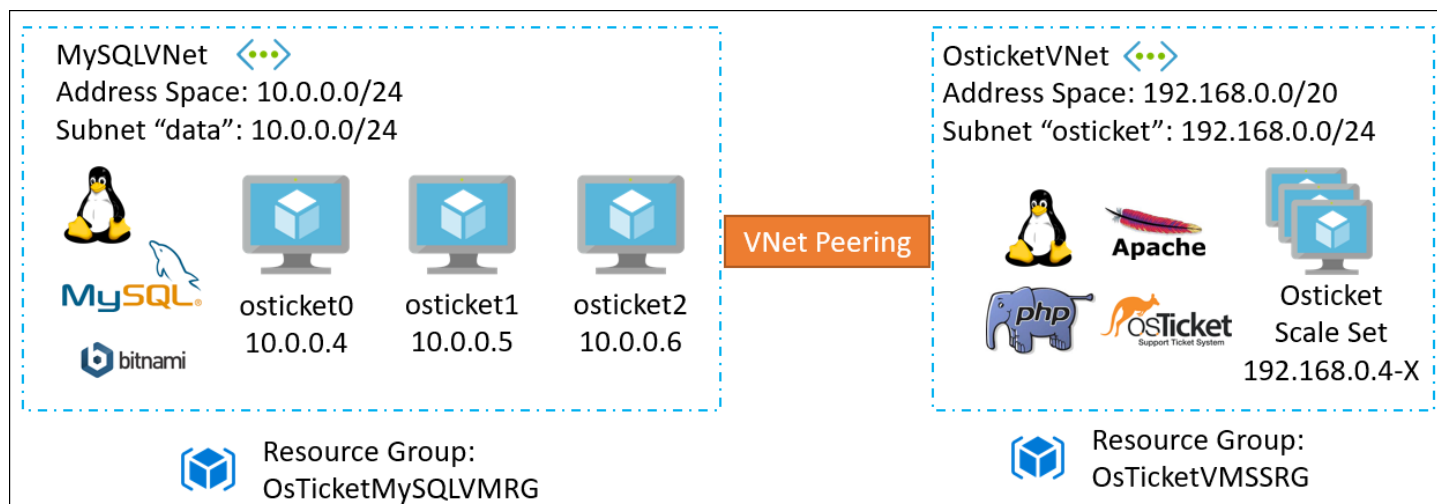
17. Once the export has completed you will receive this message.



## Exercise 2: Migrate to Azure IaaS VM Scale Sets and MySQL cluster

Duration: 90 Minutes

In this exercise, you will deploy the OsTicket application to Azure IaaS. In the first task, you will deploy and configure a MySQL Cluster with multiple nodes. Next, you will restore the database that was exported using MySQL Workbench. After the data tier of your application has been configured you will deploy a Virtual Machine Scale Set. The OsTicket application will be installed on the instances using the Custom Linux Extension and will be self-configuring. The Scale Set will be set up to autoscale and the VMs that spin up will also self-configuring.



### Task 1: Deploy the MySQL HA Cluster

1. From the Azure portal, click on the Cloud Shell icon on the top navigation.



2. Execute the following command to create a resource group that will contain the MySQL HA environment. Note: Ensure you that use the same region as the OsTicket application.

```
az group create --name OsTicketMySQLVMRG --location "East US"
```

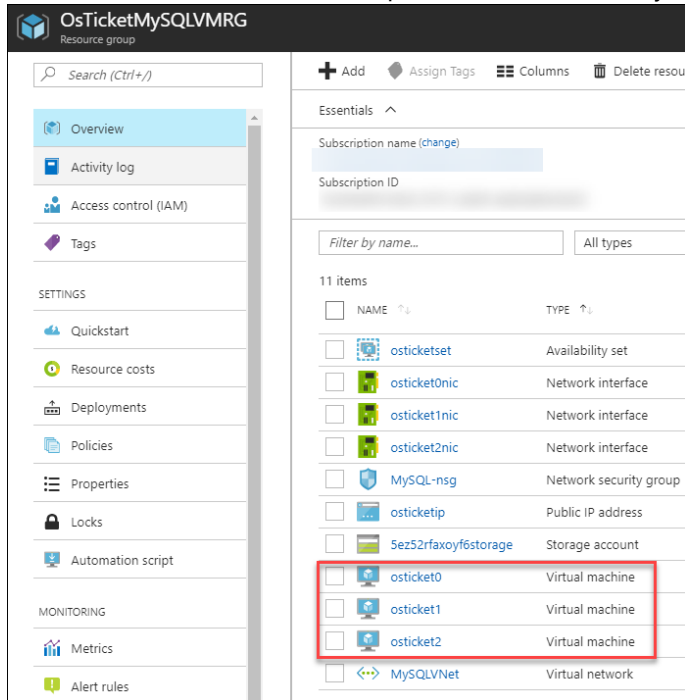
3. Execute the following command to deploy the ARM template.

```
az group deployment create --name OsTicketMySQLVMRG --resource-group
OsTicketMySQLVMRG --template-uri
"https://cloudworkshop.blob.core.windows.net/linux-lift-
shift/mysqlhadeploy.json"
```

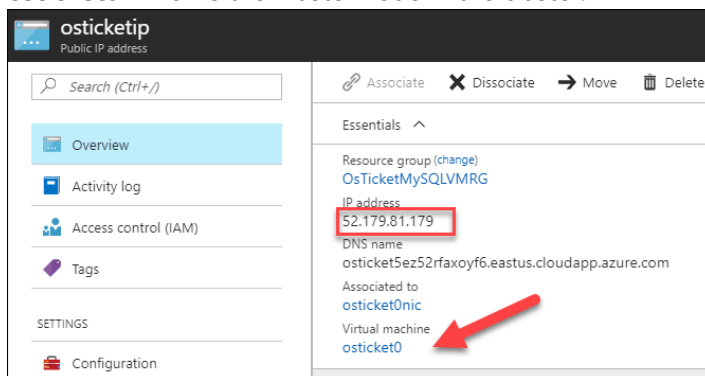
Note the settings that are being deployed as part of the template. They will be referenced later in the lab.

- a. OS User Name: **bitnami**
- b. OS Admin Password: **demo@pass123**
- c. App Database: **osticket**
- d. App Password: **demo@pass123**

- This deployment will take about 15 minutes to complete. Wait until it has been deployed before moving on to the next step.
- Once the deployment has completed open the **OsTicketMySQLVMRG** and review the deployment. Notice that there are three VMs which are a part of a three node MySQL cluster.

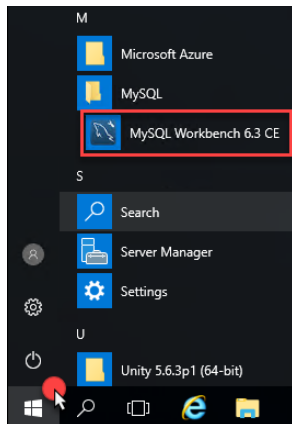


- Locate the **osticketip** Public IP address and **take note of the address**. Notice that this is attached the VM **osticket0** which is the master node in the cluster.

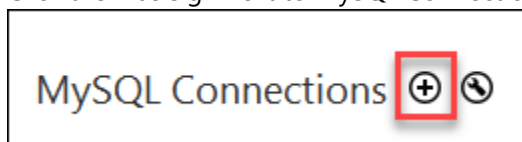


## Task 2: Connect to the MySQL cluster and restore the database

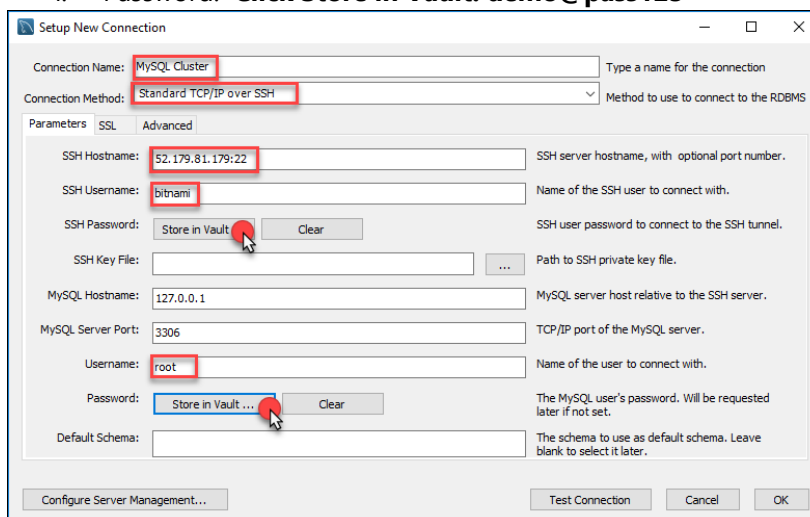
- On the **LABVM**, click Start and then locate the MySQL Workbench.



2. Click the Plus sign next to MySQL Connections on the Workbench.

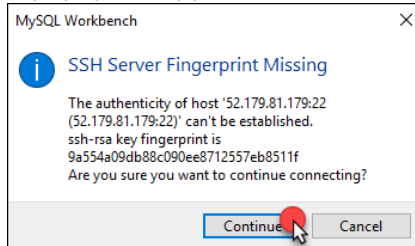


3. Enter the following information to connect to your MySQL master node.
- a. Connection Name: **MySQL Cluster**
  - b. Connection Method: **Standard TCP/IP over SSH**
  - c. SSH Hostname: **<enter the Public ip address>**
  - d. SSH Username: **bitnami**
  - e. SSH Password: Click Store in Vault: **demo@pass123**
  - f. MySQL Hostname: **127.0.0.1**
  - g. MySQL Server Port: **3306**
  - h. Username: **root**
  - i. Password: **Click Store in Vault: demo@pass123**

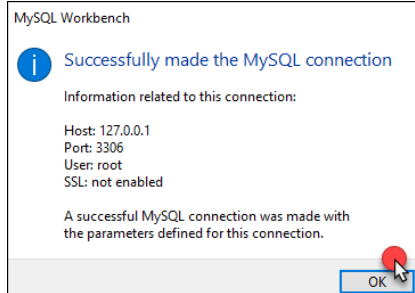


4. Once configured click **Test Connection**.

5. A popup will appear with a notice that the **SSH Server Fingerprint Missing**, click **continue**.



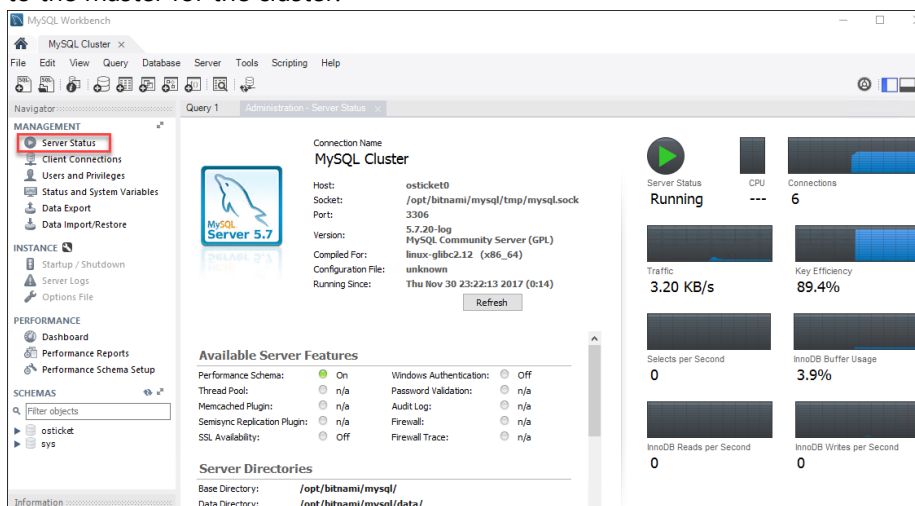
6. If configured correctly you will receive a message: **Successfully made the MySQL Connection**, Click **OK**.



7. Click **OK** to save the connection that you just configured.
8. The Connection will appear. Double-click to start a session with the MySQL Cluster instance running on the Azure IaaS.

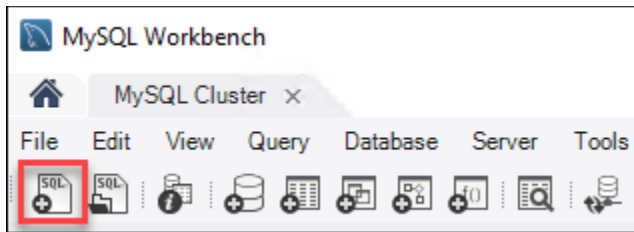


9. Once the Workbench loads, click **Server Status**. Review the details of the cluster. Remember you are connected to the master for the cluster.



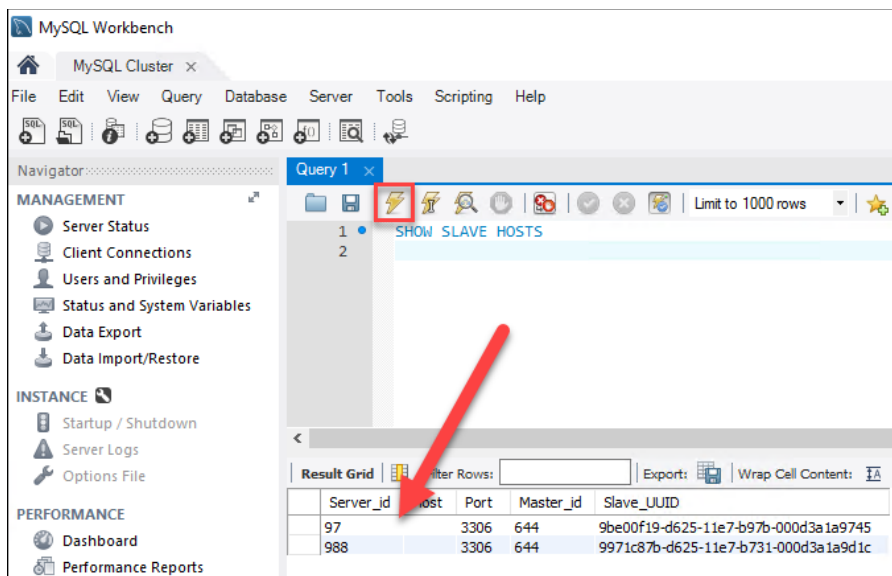
10. Click the New Query button.



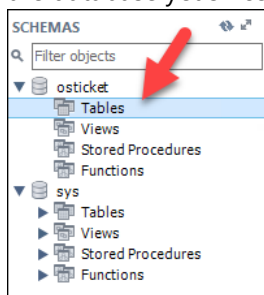


11. Execute the following query to show the status of the cluster nodes known as “Slaves”. Click the **Lightning Bolt** to run the query. Notice that the two replication partners appear in the results window as well as their TCP port of 3306.

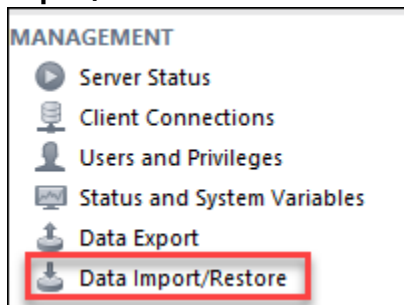
#### SHOW SLAVE HOSTS



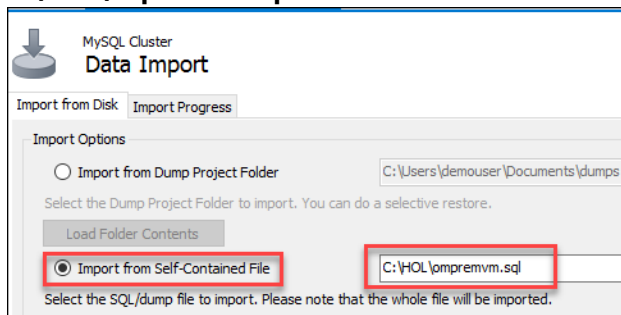
12. Next, click the **osticket** database under Schemas and expand to see the tables. This time notice that nothing is in the database yet since you have not restored it yet.



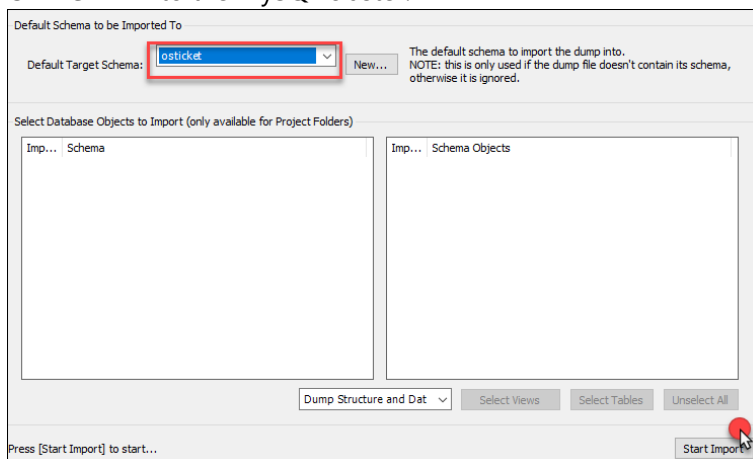
13. This is the part of the lift and shift where we will restore the existing database for the application. Click the **Data Import/Restore** button.



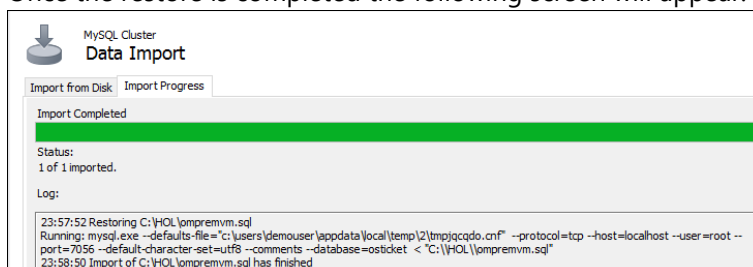
14. On the Data Import screen click the **Import from Self-Contained File**, and then select the **C:\HOL\onprevmvm.sql** datafile.



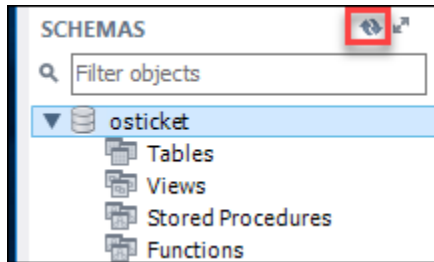
15. Next, set the Default Target Schema as **osticket** and click **Start Import**. This will restore the data from the **OnPremVM** to the MySQL cluster.



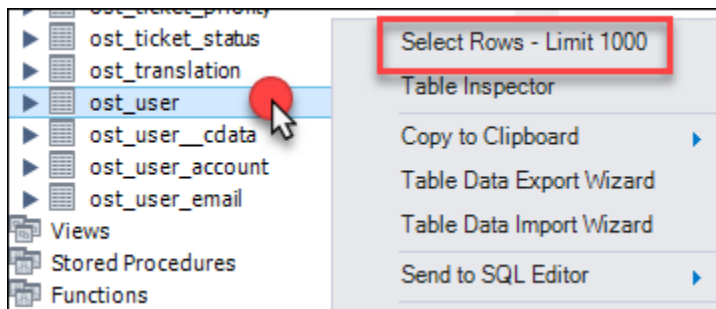
16. Once the restore is completed the following screen will appear.



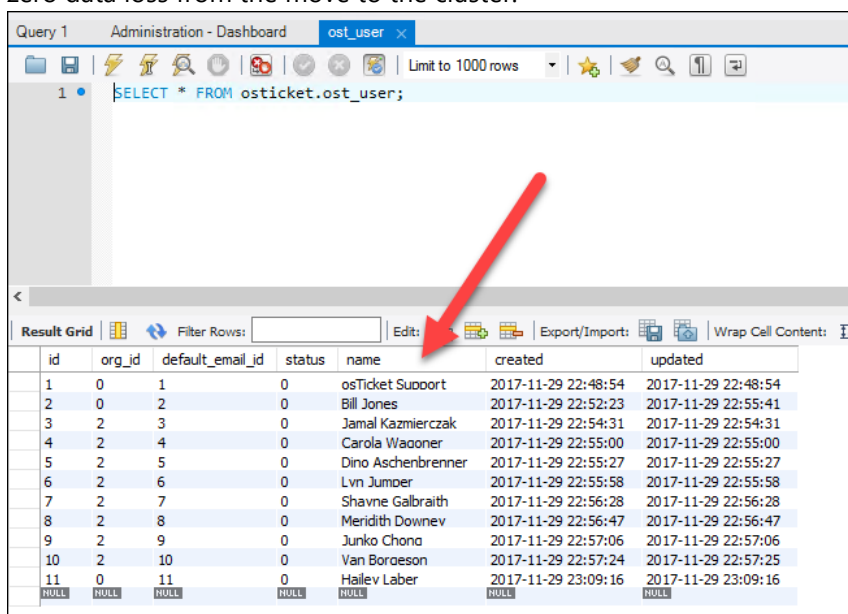
17. Move back to the Schemas area of the MySQL Workbench, and click the refresh icon.



18. The tables of the database now appear since they have been restored. Locate the **ost\_user** table, **right-click** and then click **Select Rows – Limit 1000**.



19. This will launch a query in the Workbench and list all the users on the system. Notice that now the data from the application has been Lifted and Shifted into a MySQL Cluster running in Azure IaaS. This means that there was zero data loss from the move to the cluster.



### Task 3: Deploy the Virtual Machine Scale Set for the OsTicket Application

1. From the Azure portal, click on the Cloud Shell icon on the top navigation.



2. Execute the following command to create a resource group that will contain the MySQL HA environment. Note: Ensure you that use the same region as the OsTicket application.

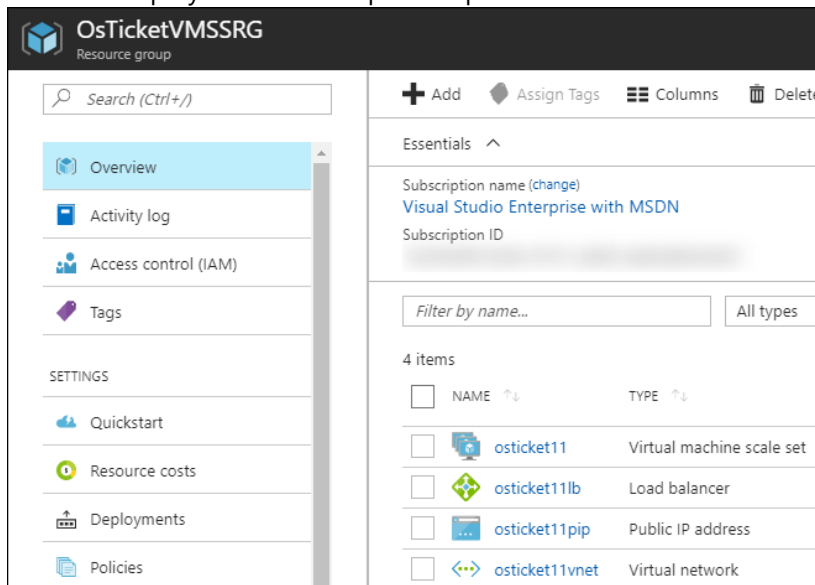
```
az group create --name OsTicketVMSSRG --location "East US"
```

3. Execute the following command to deploy the ARM template. This command requires a parameter to be passed to the template. In the example, osticketXX is used. Replace that value with a unique value that is lowercase and less than 10 characters.

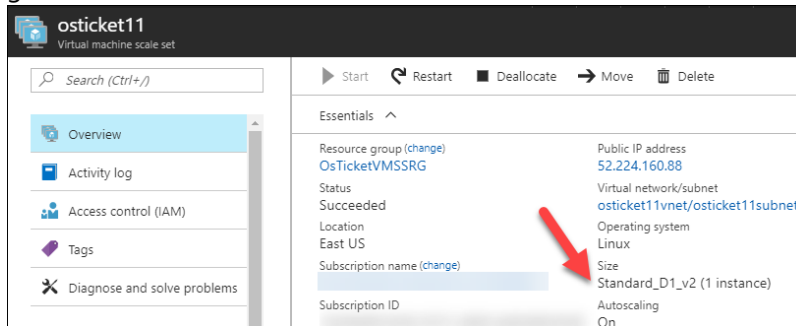
```
az group deployment create --name OsTicketVMSSRG --resource-group  
OsTicketVMSSRG --template-uri  
"https://cloudworkshop.blob.core.windows.net/linux-lift-  
shift/scalesetdeploy.json" --parameters vmssName=osticketXX
```

Take note of the credentials for the VMSS

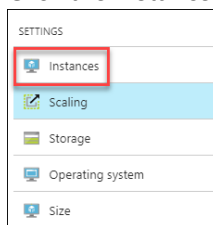
- a. Admin Username: **demouser**
  - b. OS Admin Password: **demo@pass123**
4. Once the deployment has completed open the **OsTicketVMSSRG** and review the deployment.



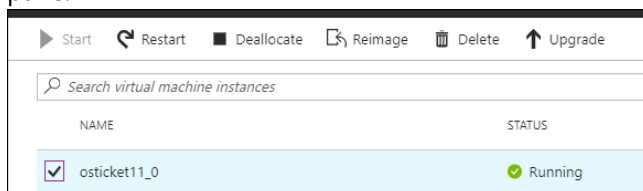
- Open the Virtual Machine Scale Set. Notice that there are two instances (to start out with), which are a part of a VM Scale Set and that autoscaling has been enabled. After some time, it will be scaled down to only one instance given the lack of traffic to the site.



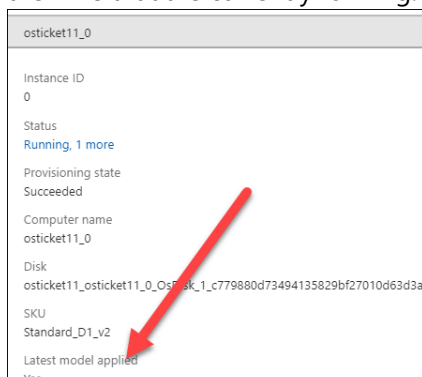
- Take note of the Public IP address.
- If you browse to the IP address in a new tab you will not be able to connect to the webpage. This is due to the lack of connectivity to the MySQL Cluster.
- Click the Instances link in Settings.



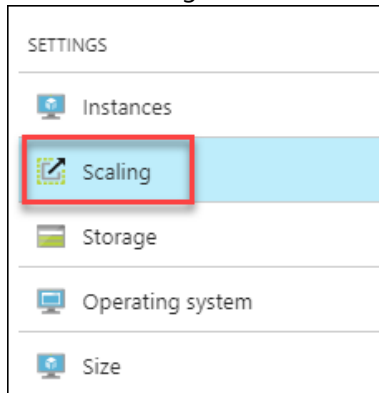
- Here you can take different actions on the Scale Set instances. Click on the instances, and review the information pane.



- In the information pane notice the "Latest model applied" information. This is where you can update the image being used for the OS. By selecting the instances and then clicking the update, you can apply system updates to the VMs that are currently running.



11. Click the Scaling link in the Settings area of the Scale Set.



12. Review the Rules for Scale Out and Scale In.

Scale mode	<input checked="" type="radio"/> Scale based on a metric <input type="radio"/> Scale to a specific instance count		
Rules	Scale out		
	When	osticket11      (Average) Percentage CPU > 60	Increase instance count by 1
	Scale in		
	When	osticket11      (Average) Percentage CPU < 30	Decrease instance count by 1

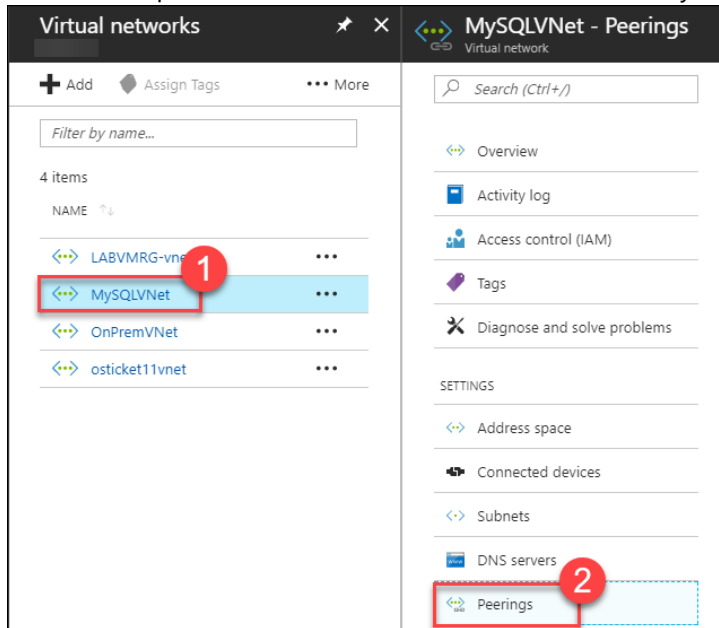
13. Click the **Notify** tab on this page, click the **Email Administrators**, **Email co-administrators**, and add any additional emails you wish. Also, notice that a Webhook could be entered here. Anytime an event happens with autoscaling Azure will send a notification. Click **Save** to configure.

A screenshot of the 'Notify' tab in the Azure portal. At the top, there are buttons for 'Save', 'Discard', 'Disable autoscale', and 'Refresh'. Below these are tabs for 'Configure', 'Run history', 'JSON', and 'Notify'. The 'Notify' tab is active. It contains two checked checkboxes: 'Email administrators' and 'Email co-administrators'. Below these is a text input field for 'Additional administrator email(s)' containing the email 'billg@microsoft.com'. At the bottom is a 'Webhook' section with a text input field for an 'HTTP or HTTPS endpoint to route autoscale notifications to', which is currently empty.

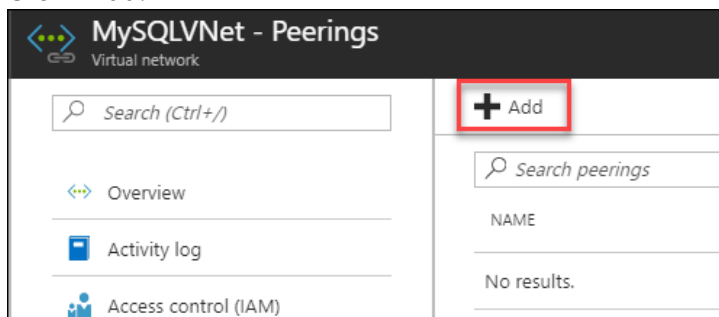
## Task 4: Connect the MySQLVNet to the Scale Sets VNet

The MySQL Cluster and the Scale Set are running in isolated VNets. To bring the OsTicket application online, you will need to create a peering between these VNets. This was the reason that both deployments needed to be in the same regions along with performance hit if they were in different regions.

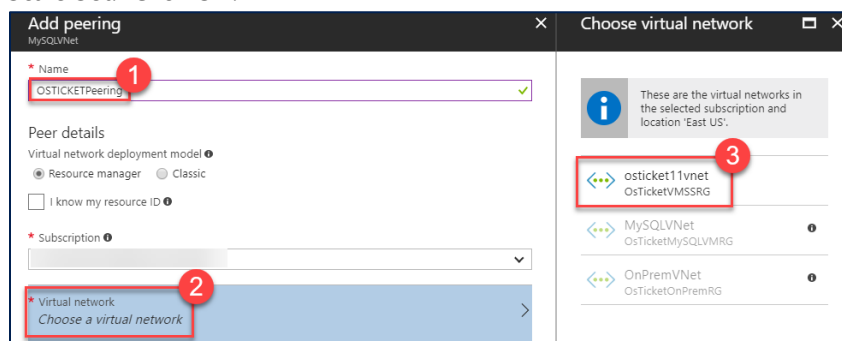
1. In the Azure portal, click on **Virtual Networks** followed by **MySQLVNet** and **Peerings**.



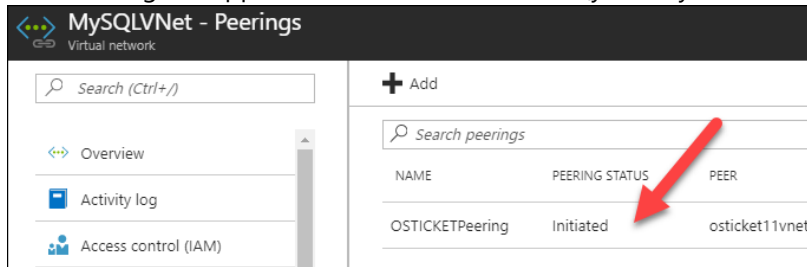
2. Click **+Add**.



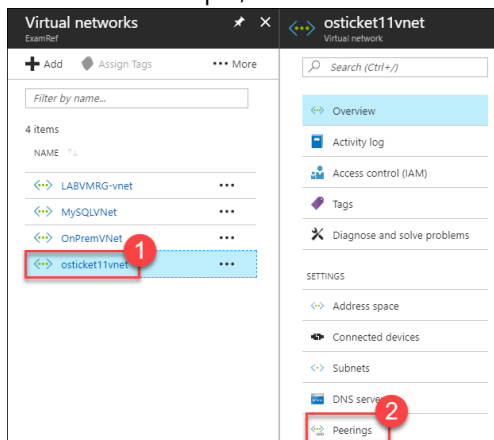
3. Enter the name **OSTicketPeering** followed by **Choose a virtual network**. Select the VNet you created with your Scale Set. Click **OK**.



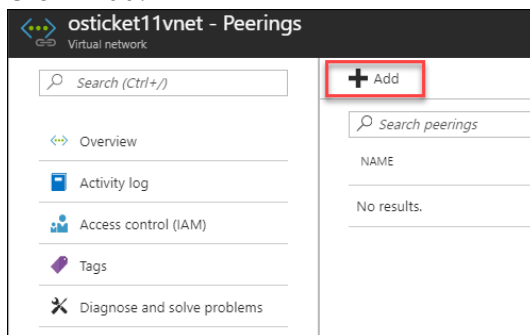
4. Your Peering will appear in the blade as Initiated (you may have to refresh the blade to see this update).



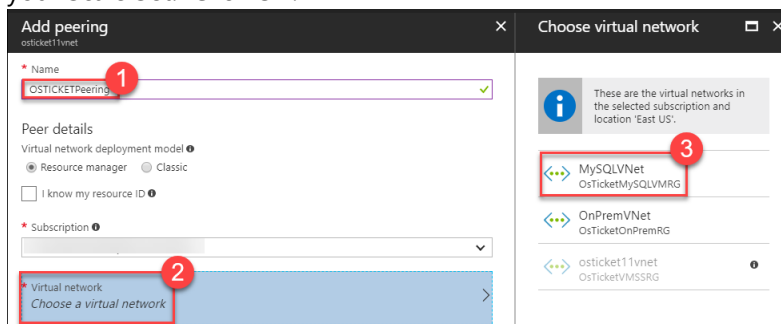
5. In the Azure portal click on **Virtual Networks** followed by the name of your **Scale Set VNet** and **Peerings**. In the case of the example, the VNet is named **osticket11vnet**.



6. Click **+Add**.

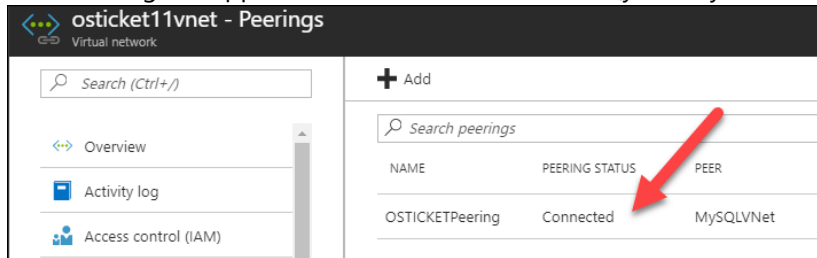


7. Enter the name **OSTicketPeering** followed by **Choose a virtual network**. Select the VNet that you created with your Scale Set. Click **OK**.

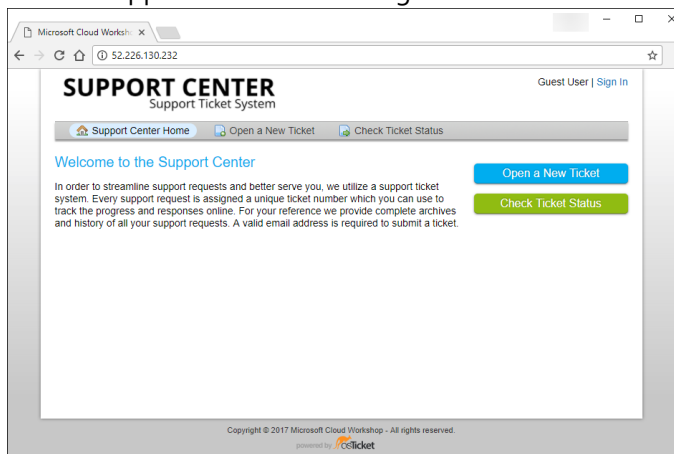




8. Your Peering will appear in the blade as Connected (you may have to refresh the blade to see this update).



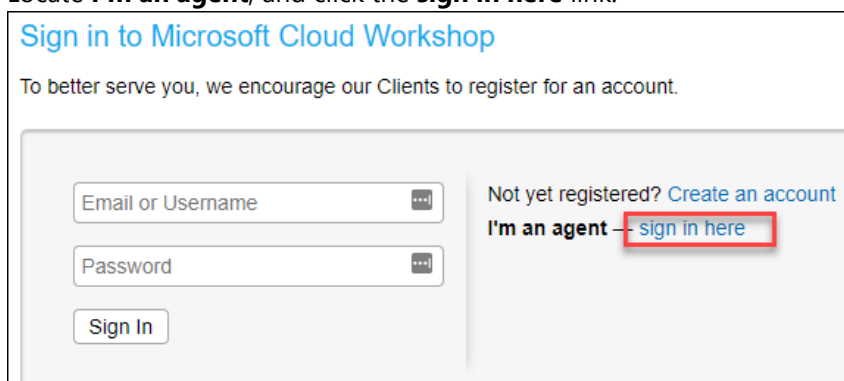
9. Immediately, the two networks can see each other which means the Scale Set will be able to see the MySQL Cluster. Open a new browser tab, and attempt to connect to the Public IP address of the Scale Set. You should see the Support Center website again.



10. Click the Sign in link.

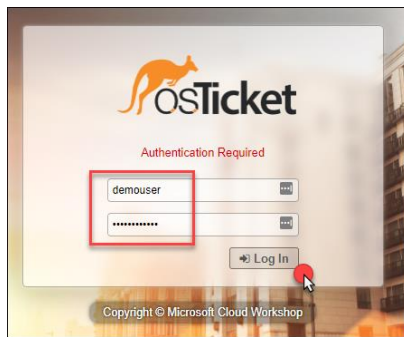


11. Locate **I'm an agent**, and click the **sign in here** link.

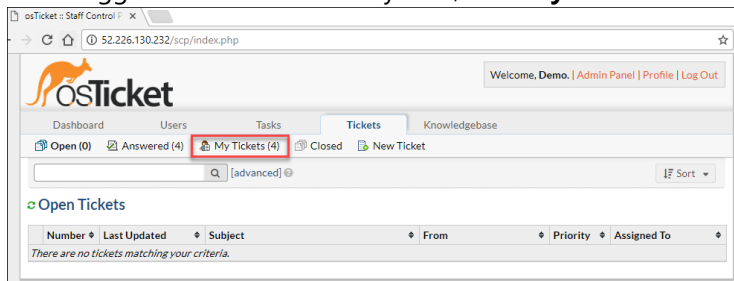


12. At the OsTicket screen enter the **username** and **password**, and click **Log In**.
- Username: **demouser**

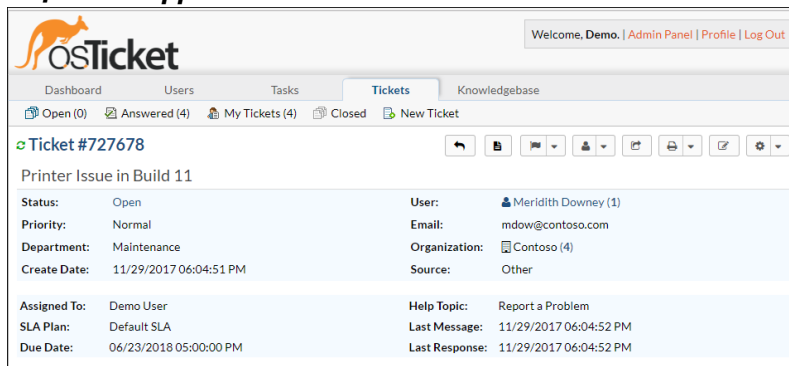
b. Password: **demo@pass123**



13. Once logged into the OsTicket system, click **My Tickets**.

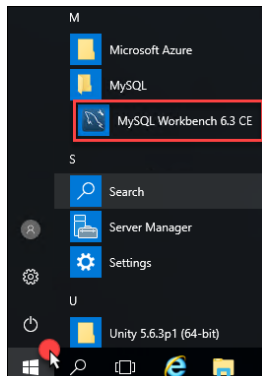


14. On the **My Tickets** screen, click through to one of the tickets. Once again, you see that the data from the on-premises installation of the OsTicket system is preserved which means that ***you have successfully lifted and shifted the application to Azure IaaS!***

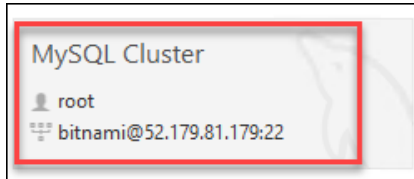


## Task 5: Export the osticket database from the MySQL cluster

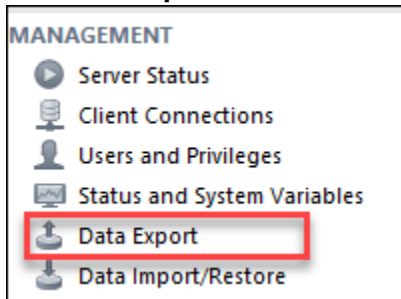
1. Next on the **LABVM** click Start and then locate the MySQL Workbench.



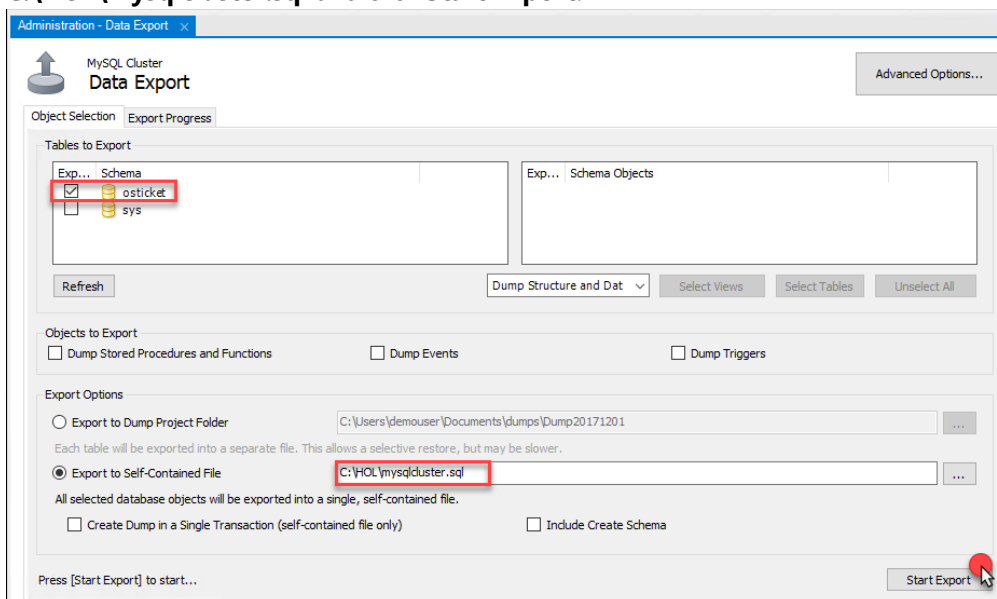
2. Double-click to start a session with the MySQL cluster.



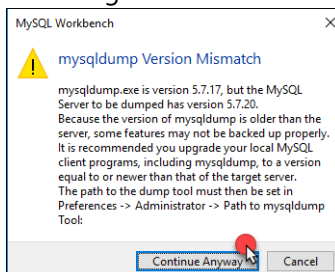
3. Click **Data Export**.



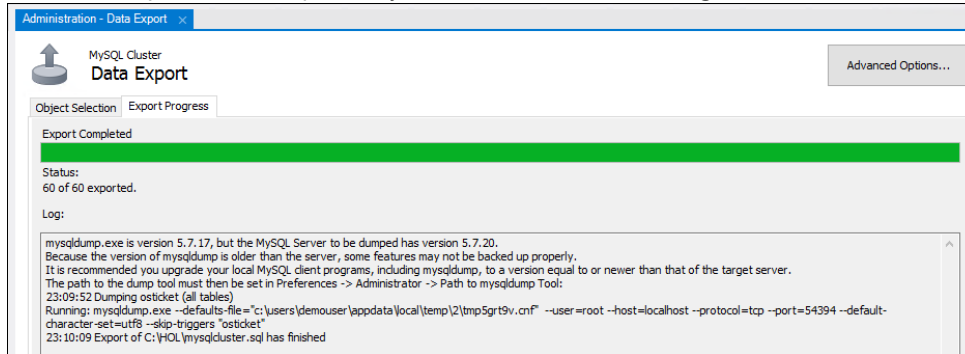
4. On the **Data Export** screen, select the **osticket**. Then select **Export to Self-Contained File**, and name the file **C:\HOL\mysqlcluster.sql** and click **Start Export**.



5. You will get a version mismatch warning, click **Continue Anyway**.



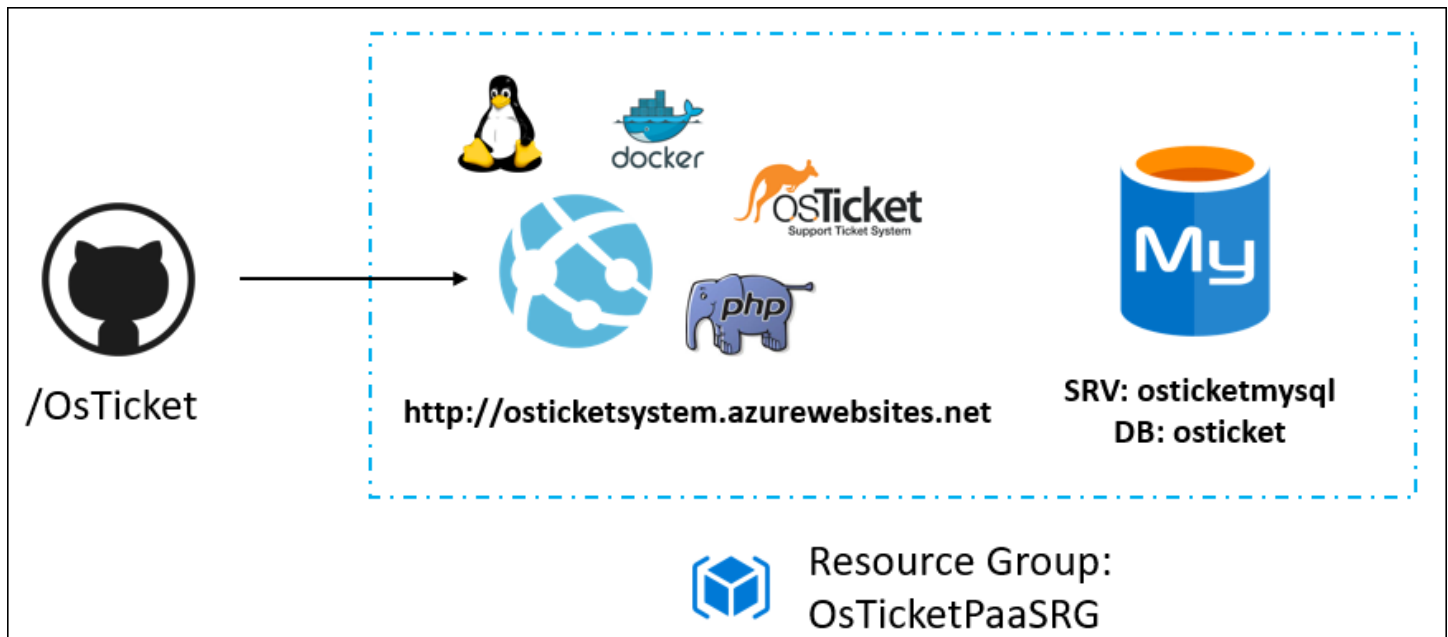
6. Once the export has completed, you will receive this message.



## Exercise 3: Migrate the OsTicket application from Azure IaaS to PaaS

Duration: 60 Minutes

In this exercise, you will implement Phase II of the migration to Azure. Here you will retire the Azure IaaS implementation and move to Azure App Service and Azure Database for MySQL. The first steps will be to build the MySQL DB and again migrate the data using MySQL Workbench. Then, you will create the Azure Web App and connect it to GitHub Repo to download the app using a Docker Container with PHP 7.0.



### Task 1: Create the MySQL database

1. From the Azure portal, click on the Cloud Shell icon on the top navigation.



2. Execute the following command to create a resource group to contain the MySQL DB.

```
az group create --name OsTicketPaaS --location "East US"
```

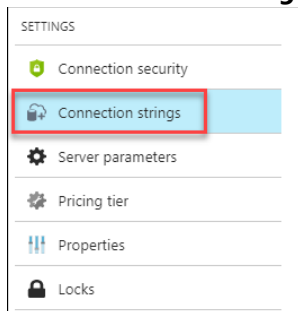
3. Execute the following command to create a MySQL Database. Note: You must choose a unique name for the MySQL server. Replace **osticketsrv01** with a more unique value.

```
az mysql server create --resource-group OsTicketPaaS --name osticketsrv01
--location "East US" --admin-user demouser --admin-password demo@pass123 --
performance-tier Basic --storage-size 51200 --ssl-enforcement Disabled
```

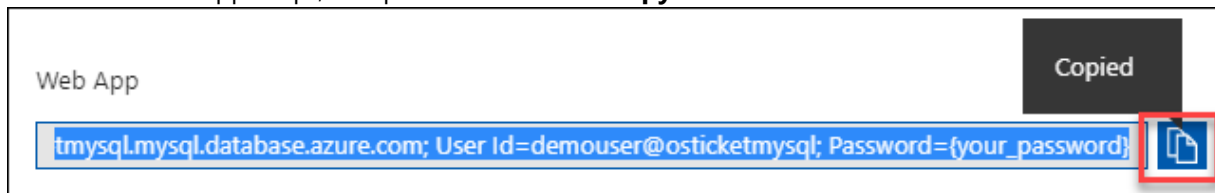
4. Add an open firewall rule to the database by executing the following command. Ensure you replace the server name with the unique value from the previous step.

```
az mysql server firewall-rule create --resource-group OsTicketPaaSRG --server-name osticketmysql --name Internet --start-ip-address 0.0.0.0 --end-ip-address 255.255.255.255
```

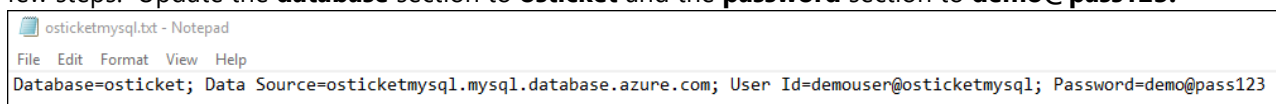
5. Once the MySQL database has been deployed, locate and open it from the **OsTicketPaaSRG** resource group using the Azure Portal.
6. Click **Connection Strings**.



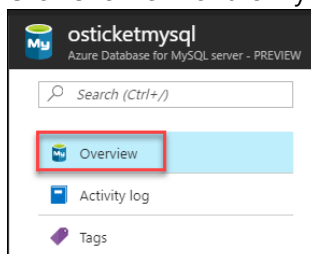
7. Locate the Web App script, and press the **Click the copy** button.



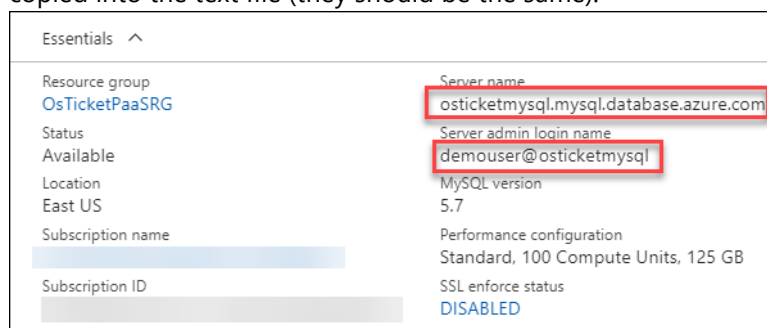
8. Open a new notepad window and paste this into a new file to retain this string and more information in the next few steps. Update the **database** section to **osticket** and the **password** section to **demo@pass123**.



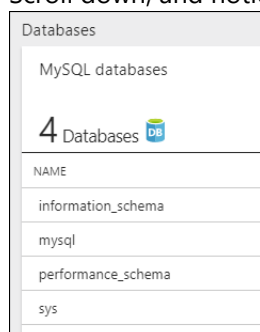
9. Click Overview for the MySQL server.



10. Notice the **Server name** and **Server Admin Login** name. You can compare them the connection string that you copied into the text file (they should be the same).

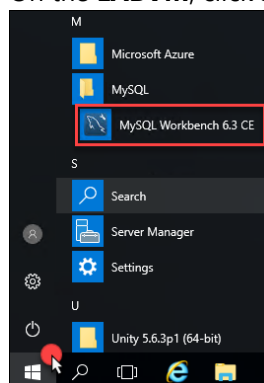


11. Scroll down, and notice that there are currently four databases that are running on your server.

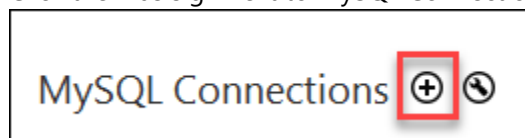


## Task 2: Restore the osticket database to MySQL PaaS

1. On the **LABVM**, click Start and then, locate the MySQL Workbench.

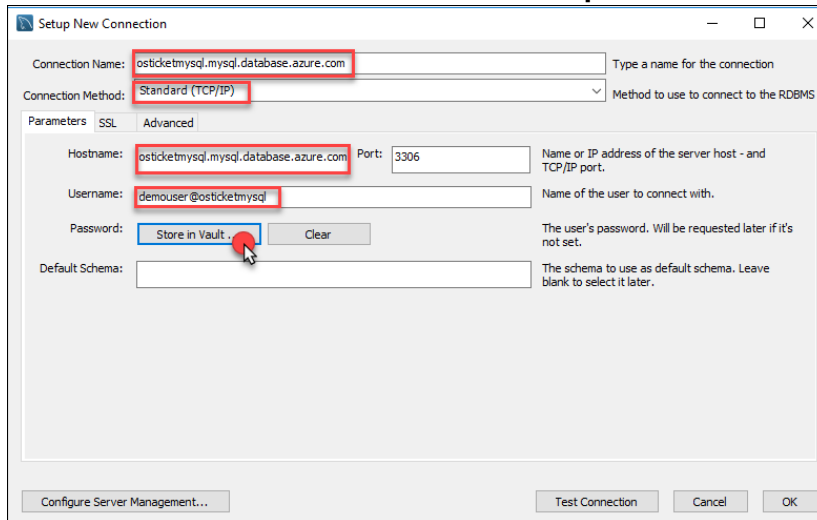


2. Click the Plus sign next to MySQL Connections on the Workbench.

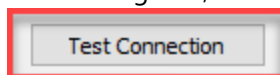


3. Enter the following information to configure to connect to your Server.
- Connection Name: **<enter your MySQL Server DNS Name – found in the connection string >**
  - Connection Method: **Standard TCP/IP**
  - MySQL Hostname: **<enter your MySQL Server DNS Name – found in the connection string >**

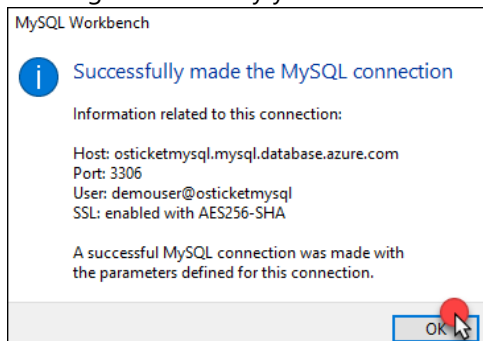
- d. MySQL Server Port: **3306**
- e. Username: **<enter your user name – found in the connection string>**
- f. Password: **Click Store in Vault: demo@pass123**



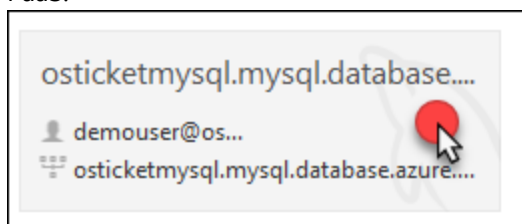
4. Once configured, click the Test Connection Button.



5. If configured correctly you will receive a message: **Successfully made the MySQL Connection**, Click **OK**.

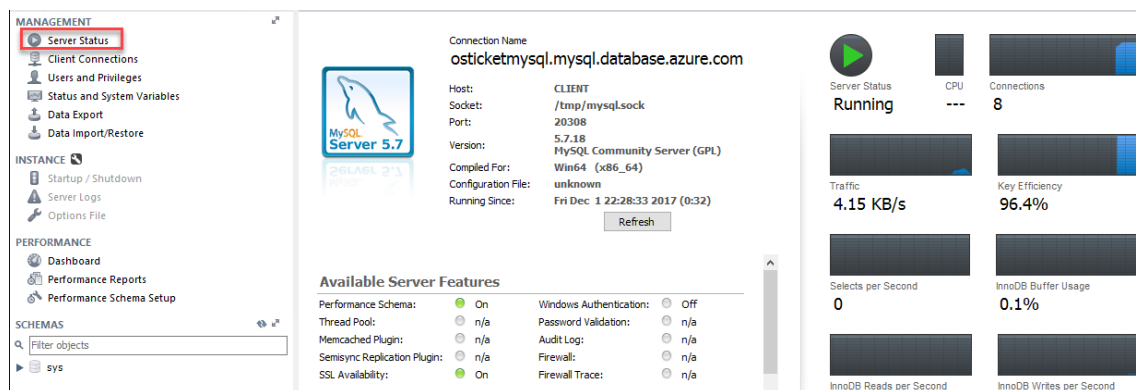


6. Click **OK** to save the connection that you just configured.
7. The Connection will appear. Double-click to start a session with the MySQL database server running on the Azure PaaS.

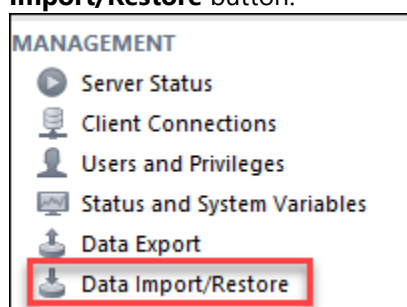


8. Once the Workbench loads, click **Server Status**. Review the details of the MySQL PaaS Server.

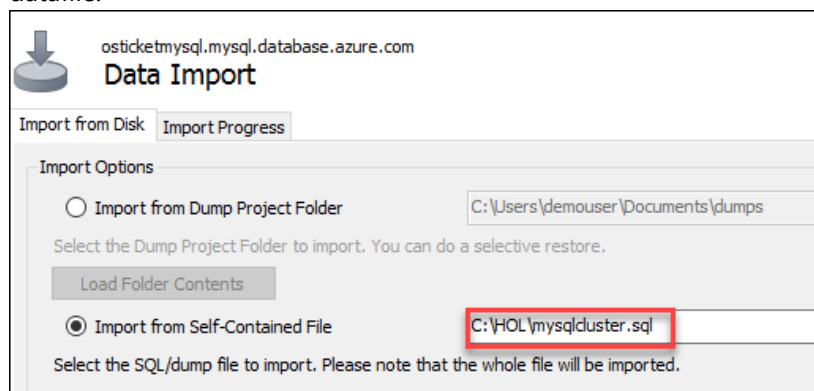




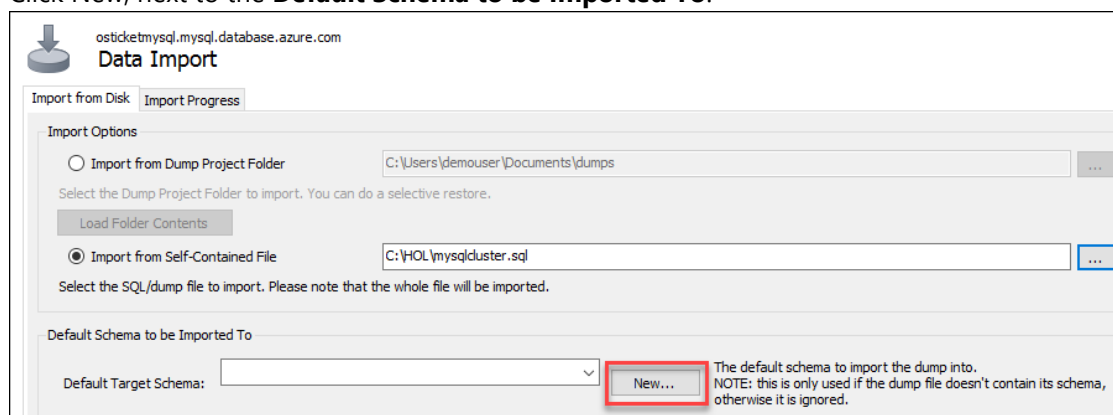
9. This is the part of the lift and shift where we will restore the existing database for the application. Click the **Data Import/Restore** button.



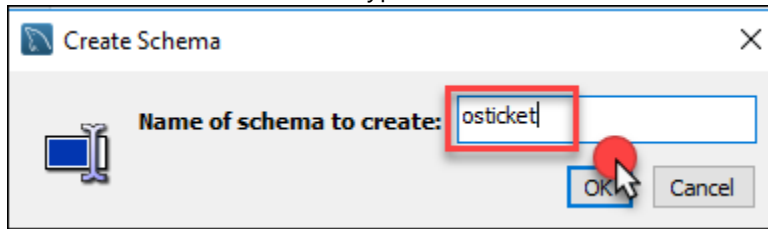
10. On the Data Import screen, click the **Import from Self-Contained File**, and select the **c:\HOL\mysqlcluster.sql** datafile.



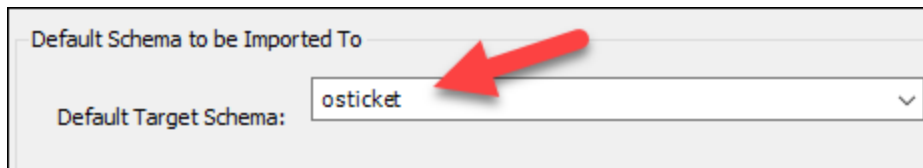
11. Click New, next to the **Default Schema to be Imported To**.



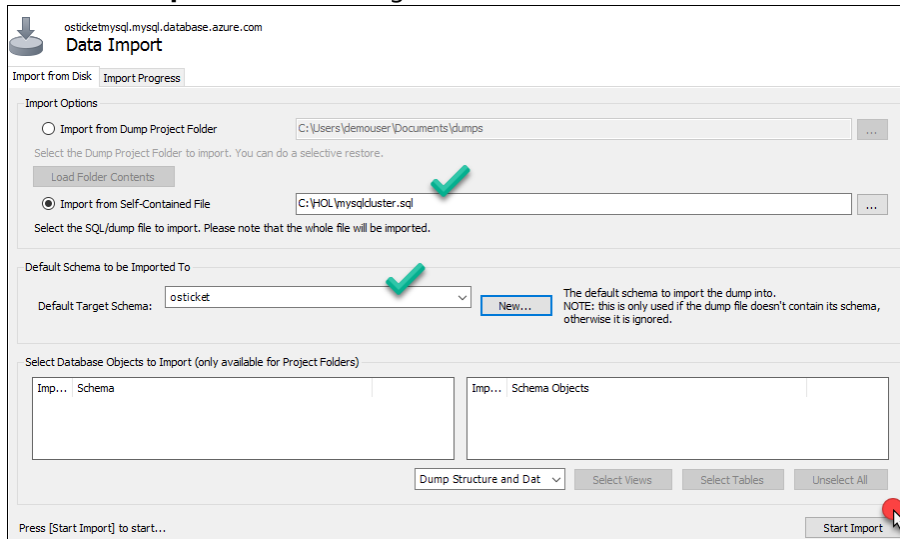
12. On the Create Schema menu, type **osticket** and click OK.



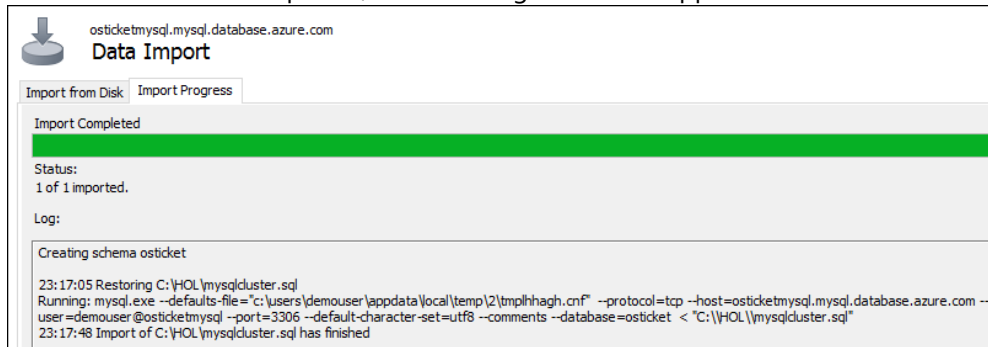
13. MySQL Workbench will create the Schema (database), on the server for you and select it as the Default Target Schema for the restore.



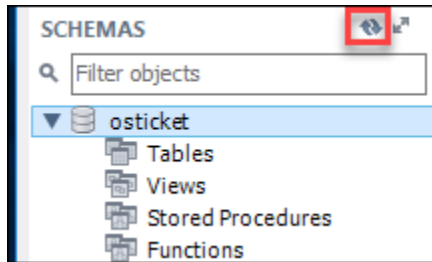
14. Click **Start Import** after reviewing the screen.



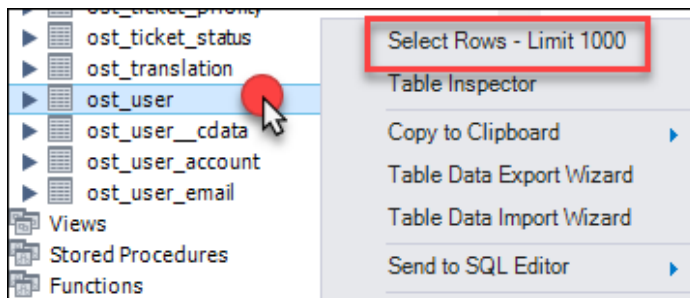
15. Once the restore is completed, the following screen will appear.



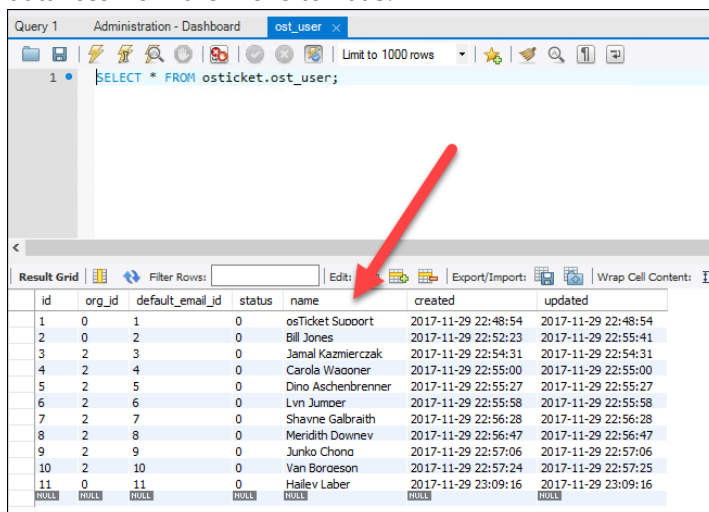
16. Move back to the Schemas area of the MySQL Workbench, and click the refresh icon.



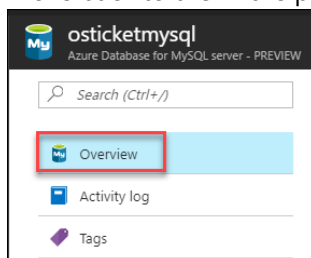
17. The tables of the database now appear since they have been restored. Locate the **ost\_user** table, **right-click**, and click **Select Rows – Limit 1000**.





18. This will launch a query in the Workbench and list all of the users on the system. Notice that now the data from the application has been Lifted and Shifted into a MySQL server running in Azure PaaS. This means there was zero data loss from the move to PaaS.



19. Move back to the Azure portal, and click Overview for the MySQL server.



20. Scroll down and notice now, there are five databases and the addition of the **osticket**.

Databases	
MySQL databases	
5 Databases 	
NAME	
information_schema	
mysql	
osticket	
performance_schema	
sys	

### Task 3: Create the Web App

- From the Azure portal, click on the Cloud Shell icon on the top navigation.



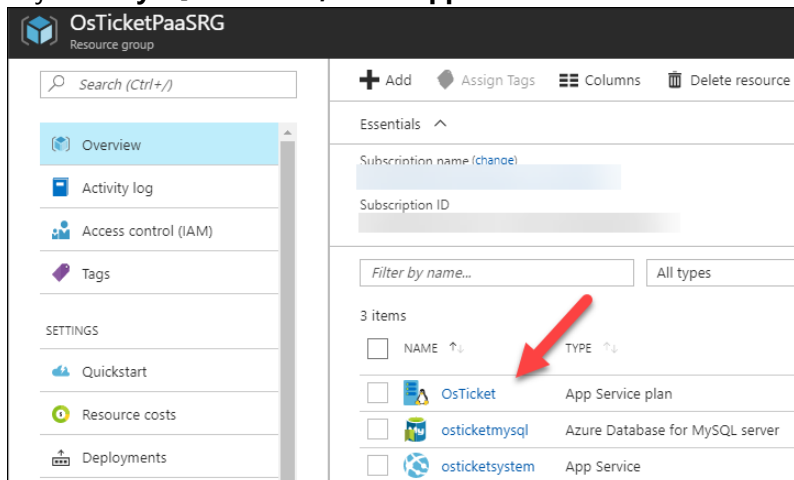
- Execute the following command to create a Linux-based App Service Plan for the new web app.

```
az appservice plan create -n OsTicket -g OsTicketPaaS -l "East US 2" --sku S1 --number-of-workers 1
```

- Execute the following command to create a new web app configured for PHP 7.0 inside of the new app service plan. The name of the web app must be unique, so specify some numbers at the end to make it a more unique value.

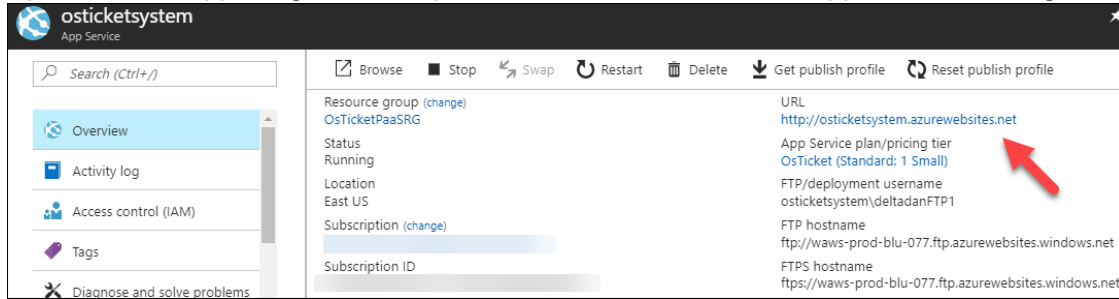
```
az webapp create -n osticketsystem -g OsTicketPaaS -p OsTicket -r "php|7.0"
```

- Once the deployment has completed, open the **OsTicketPaaS** resource group. Notice there are now three objects: **MySQL database**, **Linux App Service Plan** and the **Web App**.

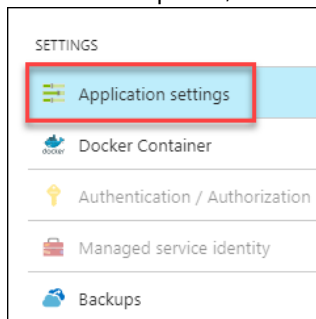


## Task 4: Configure the OsTicket Web App

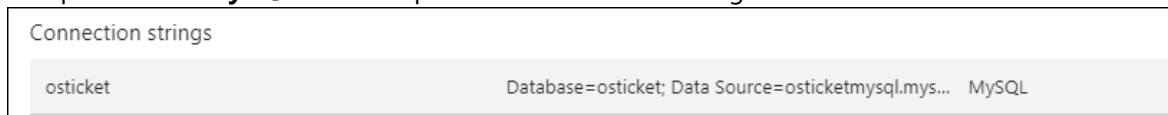
1. Open the Web App using the Azure portal. Notice the details of the application including the **URL**.



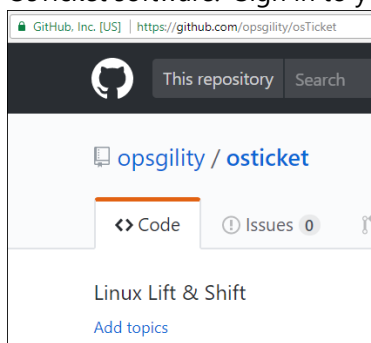
2. If you click the **URL**, the default webpage will load.
3. In the Azure portal, click **Application settings** in the Settings area.



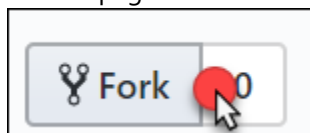
4. Locate the Connection Strings section. Enter the name **osticket**, and copy the string into the **value area** from notepad. Select **MySQL** in the dropdown list next to the string. Click **Save**.



5. Open a new browser tab and connect to <https://github.com/opsgility/osTicket>. This is a public repo for the OsTicket software. Sign in to your GitHub account or create a new one.



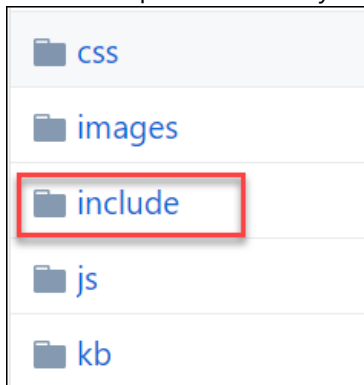
6. On this page locate and then click the **Fork** button.



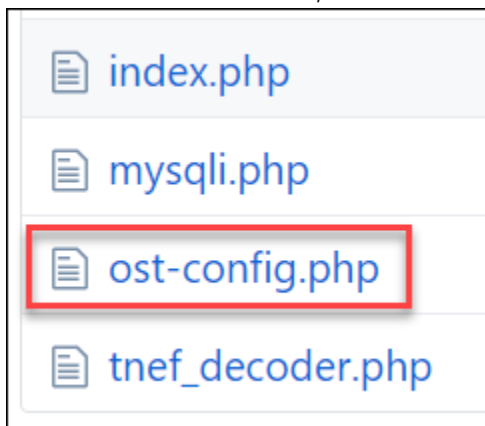
7. If prompted, select your personal account when prompted with “**Where should we fork this repository?**”



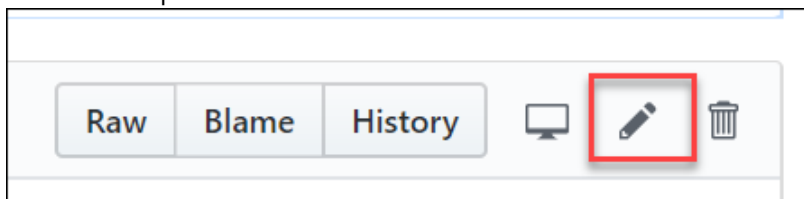
8. After the repo is forked to your GitHub account, scroll down and locate the **include** folder and click it.



9. Once in the **include** folder, scroll down and locate the file named **ost-config.php**.



10. The file will open in the browser. Click the **Pencil** icon to edit this file.



11. The file will open in an editor. Scroll down to the Database options area of the file. Update the text in this file with your MySQL database settings from your notepad file. The **DBHOST** name and the **DBUSER** should be updated. See below for the before and after comparison.

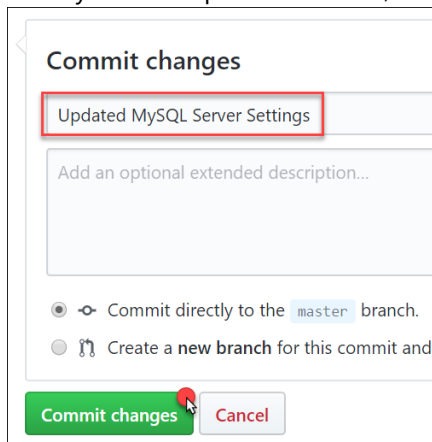
Before:

```
38 # Database Options
39 # -----
40 # Mysql Login info
41 define('DBTYPE','mysql');
42 define('DBHOST','yourservername.mysql.database.azure.com');
43 define('DBNAME','osticket');
44 define('DBUSER','demouser@yourservername');
45 define('DBPASS','demo@pass123');
```

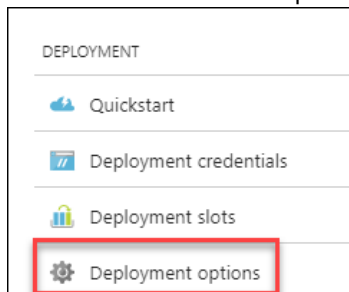
After:

```
38 # Database Options
39 # -----
40 # Mysql Login info
41 define('DBTYPE','mysql');
42 define('DBHOST','osticketmysql.mysql.database.azure.com');
43 define('DBNAME','osticket');
44 define('DBUSER','demouser@osticketmysql');
45 define('DBPASS','demo@pass123');
```

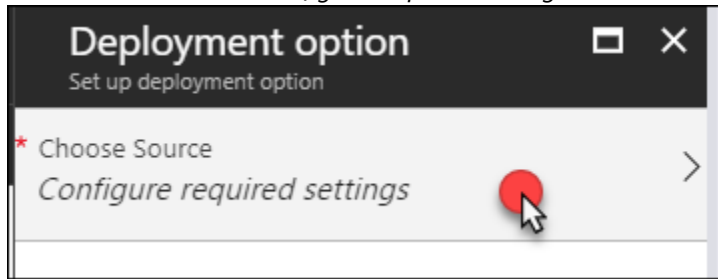
12. Once you have updated the text, scroll down enter a command and click **Commit changes**.



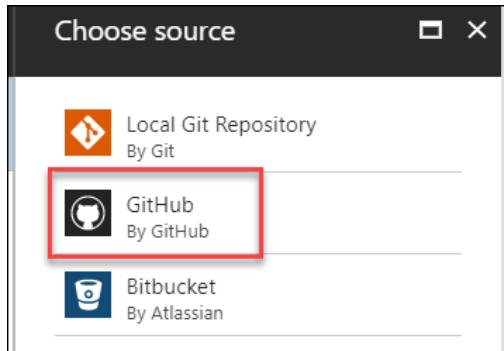
13. Move back to the Azure portal on the Web App, and click **Deployment options** under the Deployment area.



14. Click Choose Source, *Configure required settings*



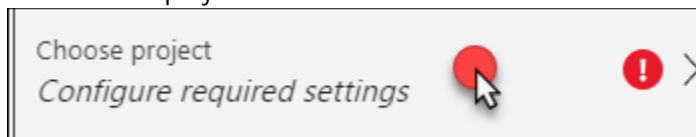
15. Click GitHub.



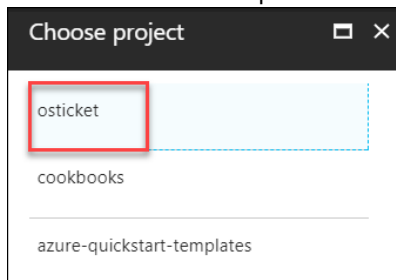
16. Click Authorization if you have not connected your GitHub account to the Azure portal follow the prompts.

17. Click Choose your organization if your GitHub personal account is not shown.

18. Click Choose project.

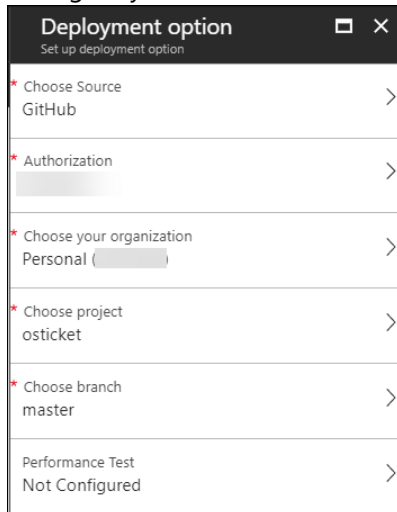


19. Select the **osticket** repo.





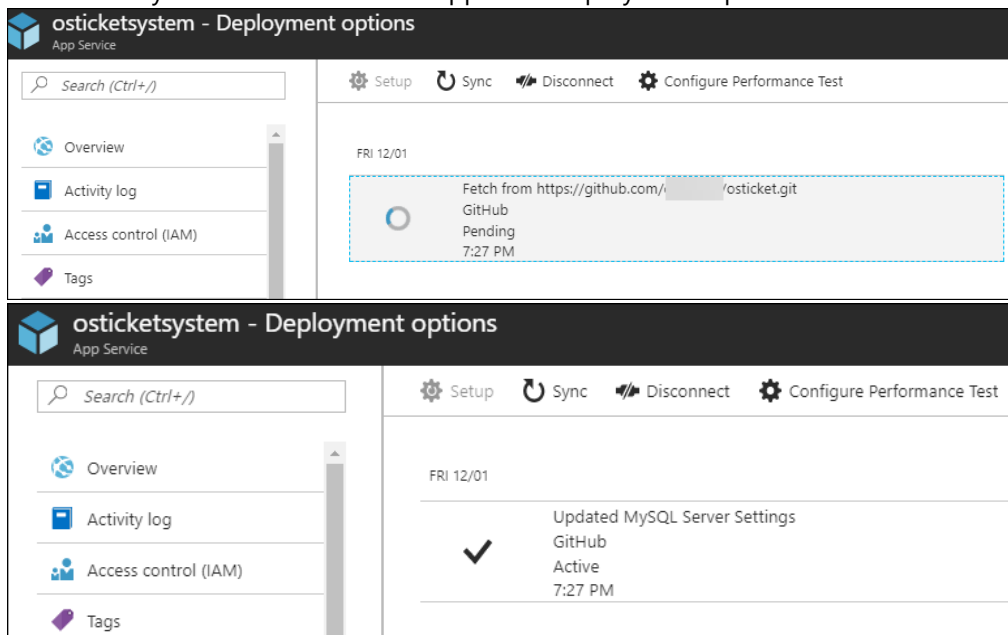
20. Configure your selections and then click **OK**.



A dialog box titled "Deployment option" with the subtitle "Set up deployment option". It contains a list of configuration items, each with a red asterisk and a right-pointing arrow:

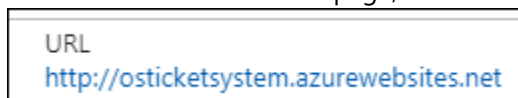
- \* Choose Source: GitHub
- \* Authorization: [Redacted]
- \* Choose your organization: Personal ([Redacted])
- \* Choose project: osticket
- \* Choose branch: master
- Performance Test: Not Configured

21. The OsTicket application will be downloaded from the GitHub account. First, it will show as Pending and then Active. It may take a minute for it to appear in Deployment Options.



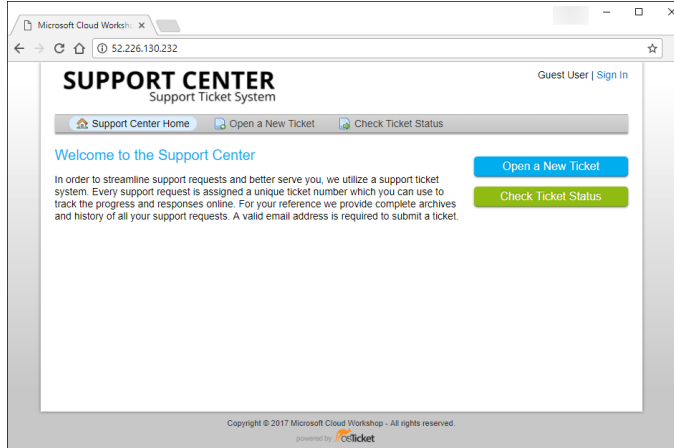
The screenshot shows the "osticketsystem - Deployment options" page in the Azure App Service portal. The left sidebar contains links for Overview, Activity log, Access control (IAM), and Tags. The main content area has tabs for Setup, Sync, Disconnect, and Configure Performance Test. A search bar is at the top left of the main area. The main content area displays a list of deployment options. The first entry, dated FRI 12/01, shows a pending fetch from a GitHub repository, with a status of "Pending" and a time of 7:27 PM. The second entry, also dated FRI 12/01, shows a successful update of MySQL Server Settings, with a status of "Active" and a time of 7:27 PM.

22. Click back to the **Overview** page, and then click the **URL** for the Web App.



A text input field labeled "URL" containing the text <http://osticketsystem.azurewebsites.net>.

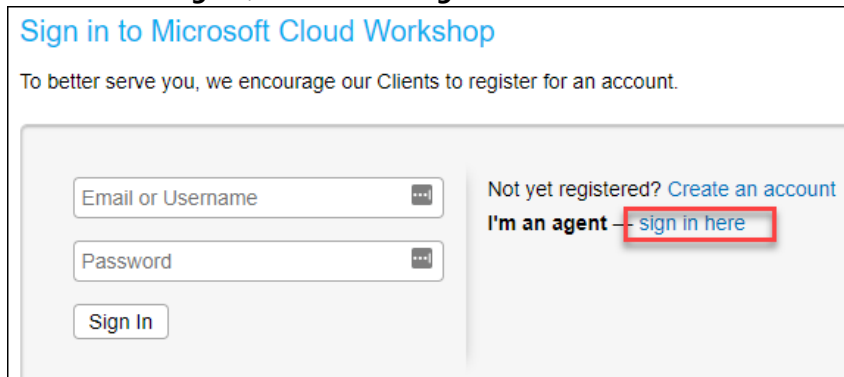
23. Immediately the Web App will load you should see the Support Center website again.



24. Click the Sign in link.

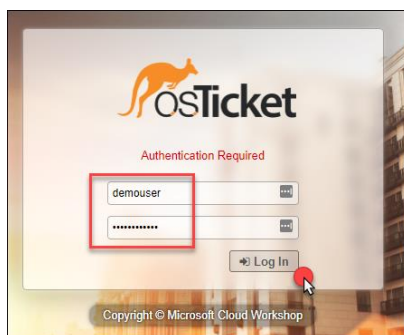


25. Locate **I'm an agent**, and click the **sign in here** link.

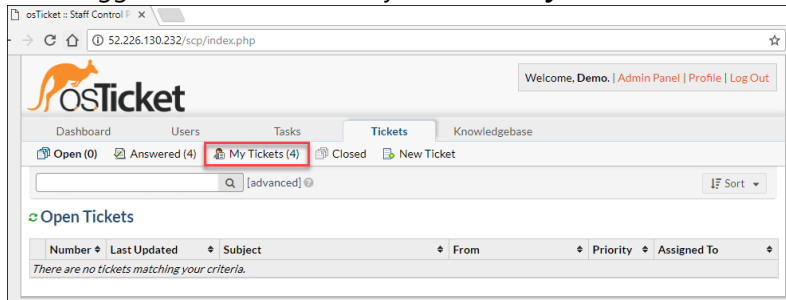


26. At the OsTicket screen, enter the **username** and **password** and click **Log In**.

- a. Username: **demouser**
- b. Password: **demo@pass123**



27. Once logged into the OsTicket system click **My Tickets**.



28. On the **My Tickets** screen, click through to one of the tickets. Once again, you see that the data from the IaaS installation of the OsTicket system is preserved which means that you have successfully lifted and shifted the application to Azure PaaS!

## After the hands-on lab

Duration: 10 minutes

After you have successfully completed the Linux Lift & Shift Azure hands-on lab step-by-step, you will want to delete the Resource Groups. This will free up your subscription from future charges.

- OPSLABRG
- OsTicketMySQLVMRG
- OsTicketOnPremRG
- OsTicketPaaS
- OsTicketVMSSRG

You should follow all steps provided *after* attending the Hands-on lab.