Contents

- Lab Services documentation
- Overview
  - About Lab Services
- Quickstarts
  - Get started
- Tutorials
  - Set up a lab account
  - Set up a classroom lab
  - Access a classroom lab
  - Track usage of a classroom lab
- Concepts
  - Labs overview
  - Labs concepts
  - Labs architecture fundamentals
  - Capacity limits
  - Cost management for labs
  - Scenarios
    - Use Lab Services for conducting classes
    - Use Lab Services for hackathon
    - Lab class types
      - Class types overview
      - Adobe Creative Cloud
      - ArcGIS
      - Autodesk
      - Big data analytics
      - Database management
      - Deep learning in natural language processing
      - Ethical hacking with Hyper-V
      - Ethical hacking with VirtualBox

- MATLAB
- Networking with GNS3
- Project Lead the Way
- Python and Jupyter Notebooks
- React web development
  - React (Linux)
  - React (Windows)
- RStudio
  - RStudio (Linux)
  - RStudio (Windows)
- Shell scripting on Linux
- SolidWorks computer-aided design (CAD)
- Sql database management and development

Use Azure Lab Services within Teams

- Overview

How-to guides

- Administrator guide
- Accelerated lab account setup guide
- Accelerated lab setup guide
- Create and configure lab accounts (lab account owner)
  - Create and manage lab accounts
  - Manage labs in a lab account
  - Configure auto-shutdown of VMs for a lab account
  - Add a lab creator
  - Allow lab creator to pick lab location
  - Connect to a peer virtual network
  - How to create a lab with a shared resource
  - Use shared image gallery for custom images
    - Attach or detach a shared image gallery
    - Create custom images for labs
    - Windows
      - Bring a custom image from an Azure VM

Bring a custom image from a physical environment
Linux
Bring a custom image from an Azure VM
Bring a custom image from a physical environment
Add additional owners to a classroom lab
Configure firewall settings
Specify Marketplace images for labs
Specify support information (lab account owner)
Create and configure labs (educator)
Create and manage labs
Use dashboard
Set up, customize, and publish templates
Set up and manage virtual machine pool
Add and manage lab users
Create and manage schedules
Use a shared image gallery
Enable remote desktop connection for Linux
Configure auto-shutdown of VMs for a lab
Enable nested virtualization on a template VM
Enable nested virtualization on a template VM overview
Enable nested virtualization on a template VM using the UI
Enable nested virtualization on a template VM using a script
Set up a lab with GPUs
Prepare a Windows template VM
Use external file storage
Control Windows shutdown behavior
View support information (educator)
Access student VMs from educator view
Connect to VMs in labs (student)
Access labs
Connect to Linux VMs
Connect to a VM from Mac using RDP

    - Connect to a VM from Chromebook using RDP
    - Set or reset passwords for virtual machines
    - View support information (student)
  - Use labs within Teams
    - Create a lab from Teams
    - Manage user lists
    - Manage VM pool
    - Delete labs
    - Schedules and settings
    - Access a VM (student view)
- Reference
  - REST API
  - Az.LabServices PowerShell module (preview)
  - Azure CLI
- Resources
  - Azure Lab Services Blog
  - Azure updates
  - Pricing
  - Tech community forum
  - Frequently asked questions

An introduction to Azure Lab Services

11/2/2020 • 3 minutes to read • Edit Online

Azure Lab Services lets you create labs whose infrastructure is managed by Azure. Currently, classroom lab is the only type of managed lab that's supported by Azure Lab Services. The service itself handles all the infrastructure management for a managed lab type, from spinning up VMs to handling errors and scaling the infrastructure. After an IT admin creates a lab account in Azure Lab Services, an instructor can quickly set up a lab for the class, specify the number and type of VMs that are needed for exercises in the class, and add users to the class. Once a user registers to the class, the user can access the VM to do exercises for the class.

Key capabilities

Azure Lab Services supports the following key capabilities/features:

- **Fast and flexible setup of a lab**. Using Azure Lab Services, lab owners can quickly set up a lab for their needs. The service offers the option to take care of all Azure infrastructure work for managed lab types. The service provides built-in scaling and resiliency of infrastructure for labs that the service manages for you.
- **Simplified experience for lab users**. Lab users can register to a lab with a registration code and can access the lab anytime to use the lab's resources.
- **Cost optimization and analysis**. A lab owner can set lab schedules to automatically shut down and start up virtual machines. The lab owner can set a schedule to specify the time slots when the lab's virtual machines are accessible to users and set usage policies per user or per lab to optimize cost.

If you want to just input what you need in a lab and let the service set up and manage infrastructure required for the lab, choose from one of the **managed lab types**. Currently, **classroom lab** is the only managed lab type that you can create with Azure Lab Services.

The following sections provide more details about these labs.

Managed lab types

Azure Lab Services allows you to create labs whose infrastructure is managed by Azure. This article refers to them as managed lab types. Managed lab types offer different types of labs that fit for your specific need. Currently, the only managed lab type that's supported is **classroom lab**.

Managed lab types enable you to get started right away, with minimal setup. The service itself handles all the management of the infrastructure for the lab, from spinning up the VMs to handling errors and scaling the infrastructure. To create a managed lab type such as a classroom lab, you need to create a lab account for your organization first. The lab account serves as the central account in which all labs in the organization are managed.

When you create and use Azure resources in these managed lab types, the service creates and manages resources in internal Microsoft subscriptions. They are not created in your own Azure subscription. The service keeps track of usage of these resources in internal Microsoft subscriptions. This usage is billed back to your Azure subscription that contains the lab account.

Here are some of the **use cases for managed lab types**:

- Provide students with a lab of virtual machines that are configured with exactly what's needed for a class. Give each student a limited number of hours for using the VMs for homework or personal projects.
- Set up a pool of high performance compute VMs to perform compute-intensive or graphics-intensive

research. Run the VMs as needed, and clean up the machines once you are done.

- Move your school's physical computer lab into the cloud. Automatically scale the number of VMs only to the maximum usage and cost threshold that you set on the lab.
- Quickly provision a lab of virtual machines for hosting a hackathon. Delete the lab with a single click once you're done.

Next steps

See the following tutorials for step-by-step instructions to create a lab account, and create a classroom lab.

- Tutorial: setup a lab account
- Tutorial: create a classroom lab

Get started with Lab Services

3/5/2021 • 5 minutes to read • Edit Online

Azure Lab Services provides students and teachers with access to virtual computer labs directly from their own computers.

Teachers need to know how to teach students/parents to utilize Lab Services in their instruction through one-to-one student issued hardware. As a result, students would be able to access industry-standard software required for their programs of study through Virtual Machines (VM).

A VM is a virtual environment that acts as a virtual computer. VMs have their own processor, memory, and storage. VMs provide a substitute for a real machine and can give users access to operating systems and software without the need to have them on their own device. Azure Lab Services provides a tool for students to access and navigate VMs and for staff to manage their virtual computer labs.

This article provides information for teaching staff on how to access, manage, and teach students/parent to utilize Azure Lab Services.

Key concepts

Quota hours

Students can access their VMs at any time during scheduled class time without impacting their quota hours. Quota hours are set for the entire semester and determine the number of hours a student can use their VM outside of regularly scheduled class time.

8 Hrs per week, resets on Sunday - not cumulative.

For more information, see Set quota.

Automatic shut-down

To help keep down costs and save students' quota hours, automatic shutdowns are enabled for the labs. Auto-shutdowns will turn VMs off after a period of inactivity (no mouse or keyboard inputs). Auto-shutdowns work in two stages, first a student will be disconnected from the VM after a period of inactivity. At this point, the VM is still `Running` and the students are able to connect. After another period of inactivity once disconnected, the VM will shut itself down.

Auto-shutdowns are an important cost-saving tool, however they do present a challenge for students in regard to saving their work and rendering large project files. If your students are frequently being disconnected or VMs are turning off too quickly, reach out to your CTE administrator.

For more information, see Configure automatic shutdown of VMs for a lab account.

Managing Virtual Machines

Managing the lab allows teachers to control things like lab capacity (the number of VMs available for students) and manually starting, stopping, or resetting VMs. teachers can also connect to VMs to experience student interface, access files and troubleshoot issues with software or the VM itself.

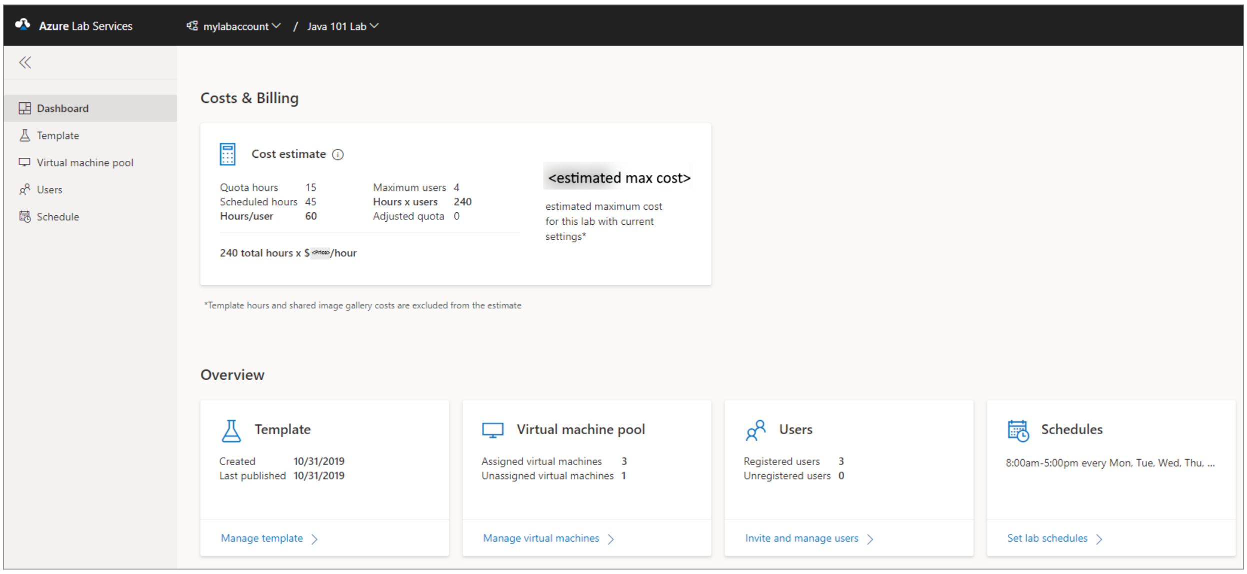
The most important thing to remember when managing the VMs is that anytime a machine is `Running`, we are incurring costs, even if no one is connected to the VM.

Lab dashboards

Overview

Dashboards for labs in Azure Lab Services provide a snapshot of different aspects of a particular lab including, VM information, number of assigned and unassigned VMs, number of registered and unregistered users and information about lab schedules.

> **NOTE**
>
> While most administrative aspects of the dashboard and the Azure Lab Services website will be visible to teachers, permissions specific to your role may impact your ability to modify certain criteria in the dashboard. If you encounter an issue with your particular lab set-up, reach out to your CTE administrator.



Examine a dashboard

1. Navigate and sign in to the Azure Lab Services website.

2. Select your lab.

3. You will see a **Dashboard** on the left-hand side of the window. Click on **Dashboard** and you will see a number of tiles in your dashboard.

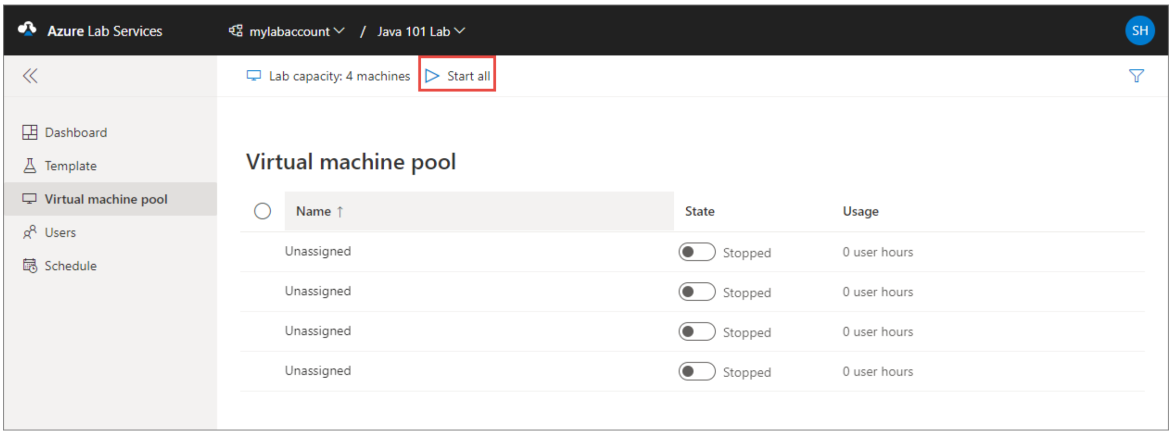
4. Below the **Costs & Billing** tile, there are also tiles for Templates, Virtual Machine Pools, Users, and Schedules, which allow you to modify aspects and view more details on the Classroom Lab.

   - Template - describes the date the template was created, and last published.
   - Virtual Machine Pool - number of assigned and unassigned VMs.
   - Users - number of registered users and users who have been added to the lab, but not registered.
   - Schedules - displays upcoming scheduled events for the lab and a link to view more events.

For more information, see Use dashboard.

Manually starting VMs

1. From the **Virtual machine pool** page, you can start all VMs in a lab by clicking the **Start all** button at the top of the page.



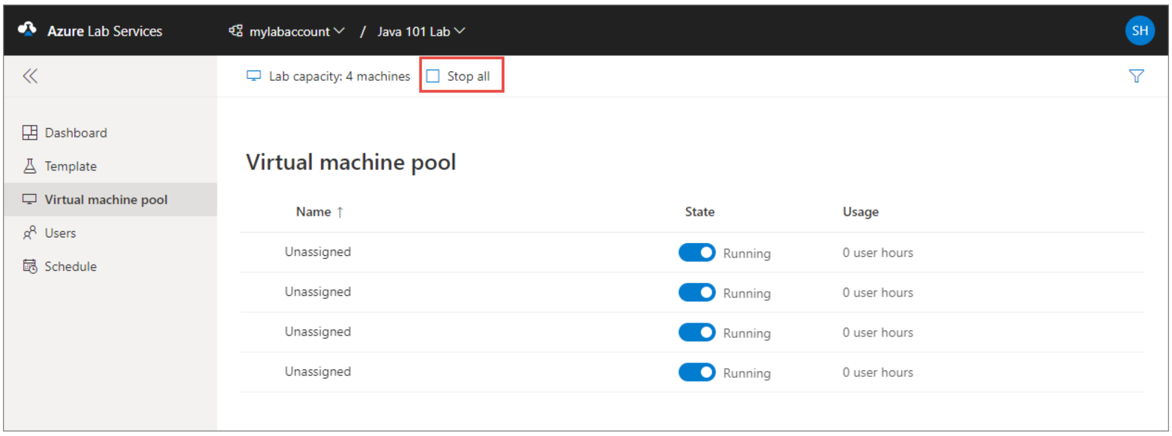
2. Individual VMs can be started by clicking the state toggle.

   The toggle will read **Starting** as the VM starts up, and then **Running** once the VM has started.

3. You can also select a number of VMs using the checks to the left of the **Name** column.

   Once you have selected the desired VMs, click the **Start** button at the top of the screen.

4. Once started, you can click the **Stop all** button to stop all of the VMs.



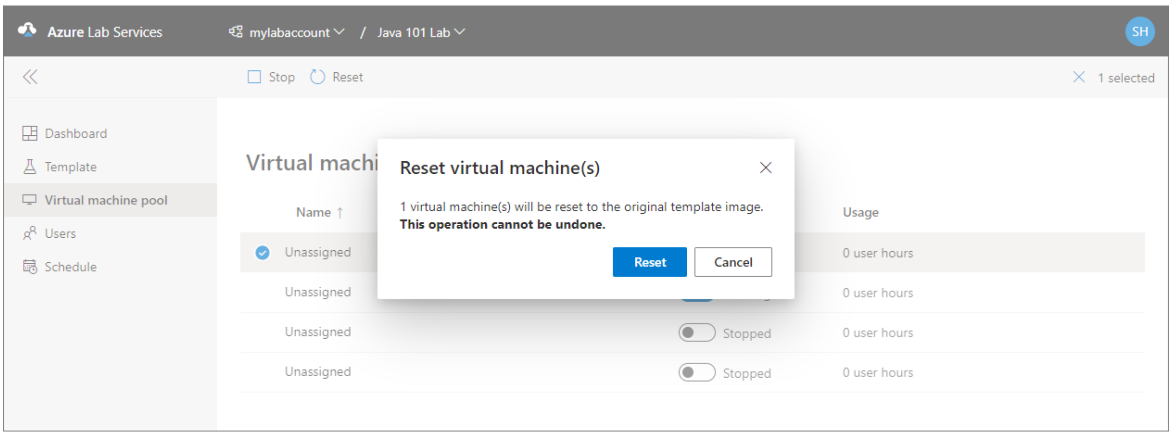
Stopping and resetting VMs

- You can stop individual VMs by clicking the state toggle.
- You can also stop multiple VMs by using the checks and clicking the "Stop" button at the top of the screen.

If a student is experiencing difficulties connecting to the VM, or the VM needs to be reset for any other reason, you can use the reset function.

1. To reset one or more VMs, select them using the checks, then click the **Reset** button at the top of the page.

2. In the pop-up window, click **Reset**.

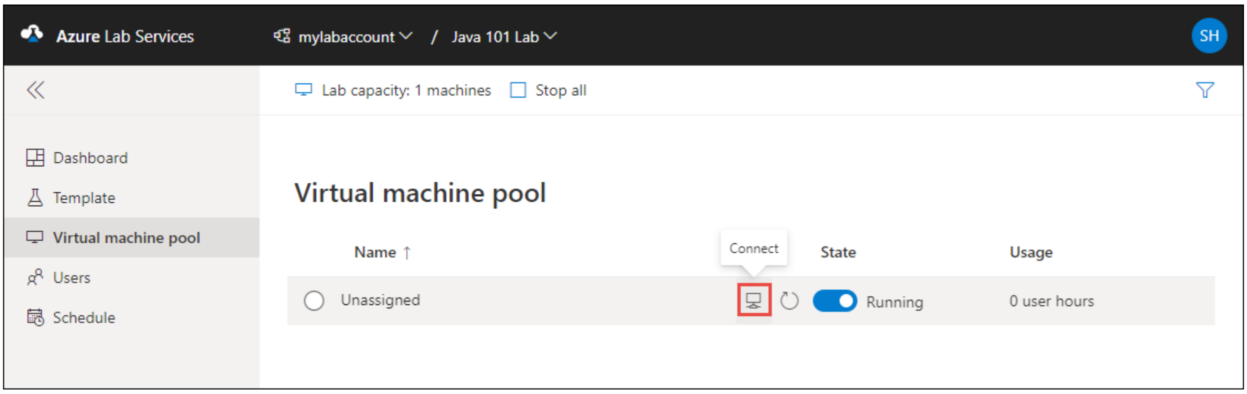


> **NOTE**
>
> Turning on a student VM will not affect the quota for the student. Quotas for users specifies the number of lab hours available to the user outside of the scheduled class time.

Connect to VMs

Teachers are able to connect to a student VM as long as it is turned on, and the student is NOT connected to the VM. By connecting to the VM, you will be able to access local files on the VM and help students troubleshoot issues.

1. To connect to the student VM, hover the mouse on the VM in the list and click the **Connect** button.
2. Then follow the getting started guide for students for either Chromebooks, Macs or PCs



Manage users in a lab

Teachers are able to add student users to a lab and monitor their hour quotas. For details on how to add users by email address or by using a spreadsheet list and register users, see [Add and manage lab users](#).

After you have invited users or shared the link, you will be able to monitor which users have registered successfully in the **Users** page in the **Status** column.

Clean up resources

If you're not going to continue to use resources that you created in this quickstart, delete the resources.

Next steps

Set up a lab account

Tutorial: Set up a lab account with Azure Lab Services

8/20/2021 • 2 minutes to read • Edit Online

In Azure Lab Services, a lab account serves as the central account in which your organization's labs are managed. In your lab account, give permission to others to create labs, and set policies that apply to all labs under the lab account. In this tutorial, learn how to create a lab account.

In this tutorial, you do the following actions:

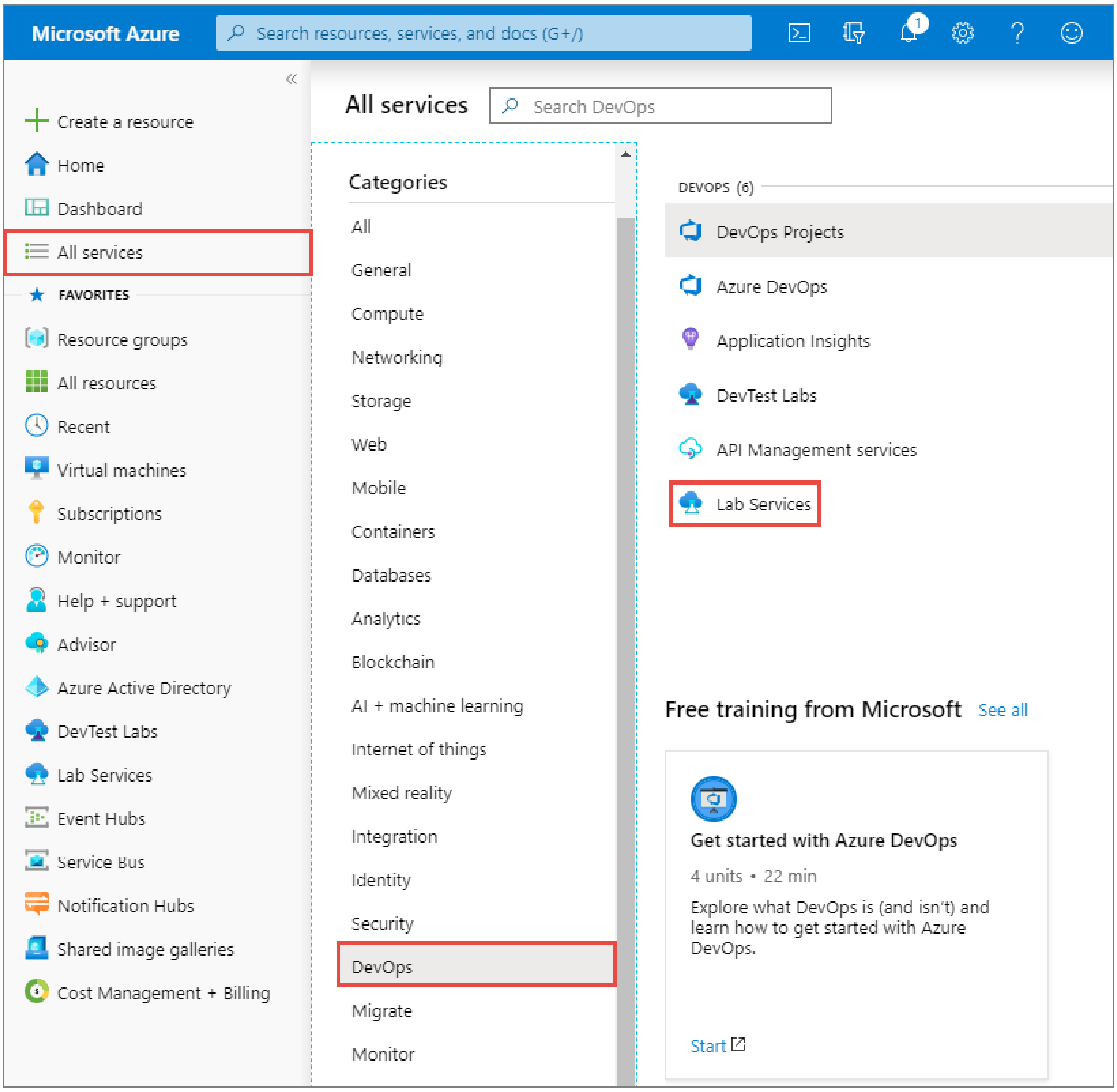
- Create a lab account
- Add a user to the Lab Creator role

If you don't have an Azure subscription, create a free account before you begin.

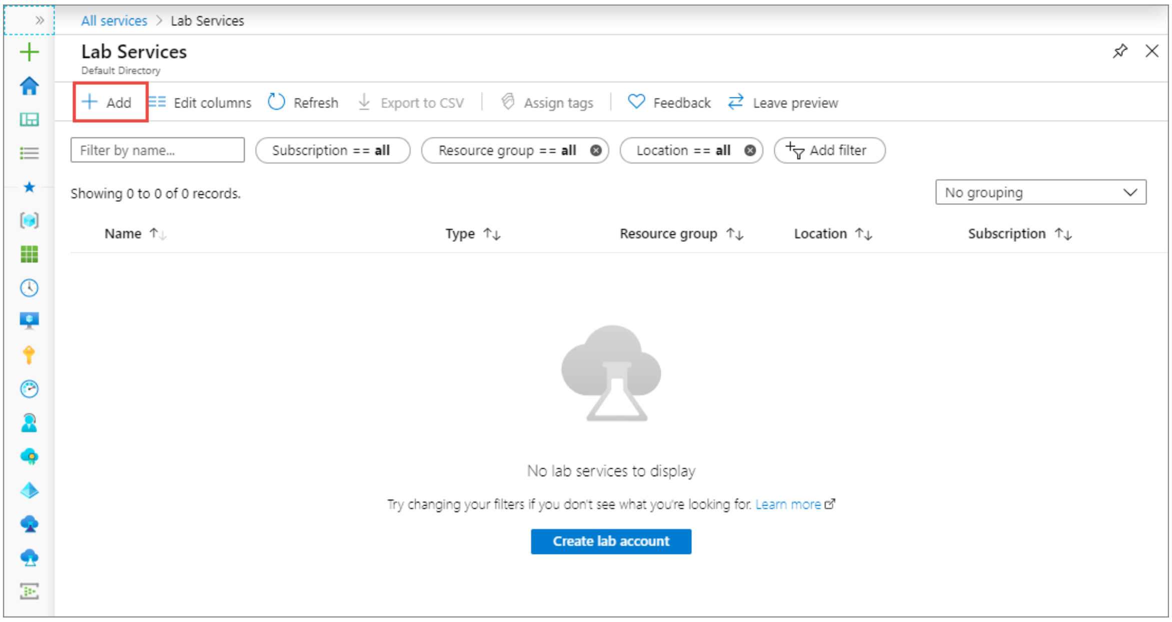
Create a lab account

The following steps illustrate how to use the Azure portal to create a lab account with Azure Lab Services.

1. Sign in to the Azure portal.
2. Select **All Services** on the left menu. Select **DevOps** from **Categories**. Then, select **Lab Services**. If you select star (`*`) next to **Lab Services**, it's added to the **FAVORITES** section on the left menu. From the next time onwards, you select **Lab Services** under **FAVORITES**.



3. On the **Lab Services** page, select **Add** on the toolbar or select **Create lab account** button on the page.



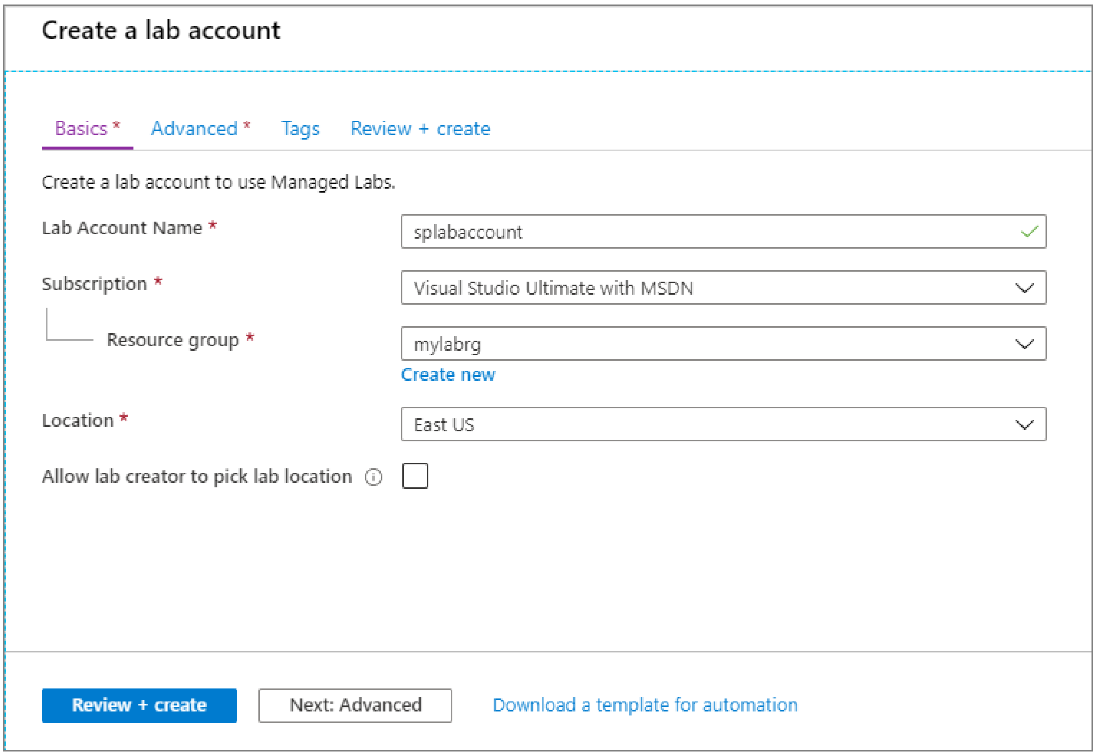
4. On the **Basics** tab of the **Create a lab account** page, do the following actions:

   a. For **Lab account name**, enter a name.

   b. Select the **Azure subscription** in which you want to create the lab account.

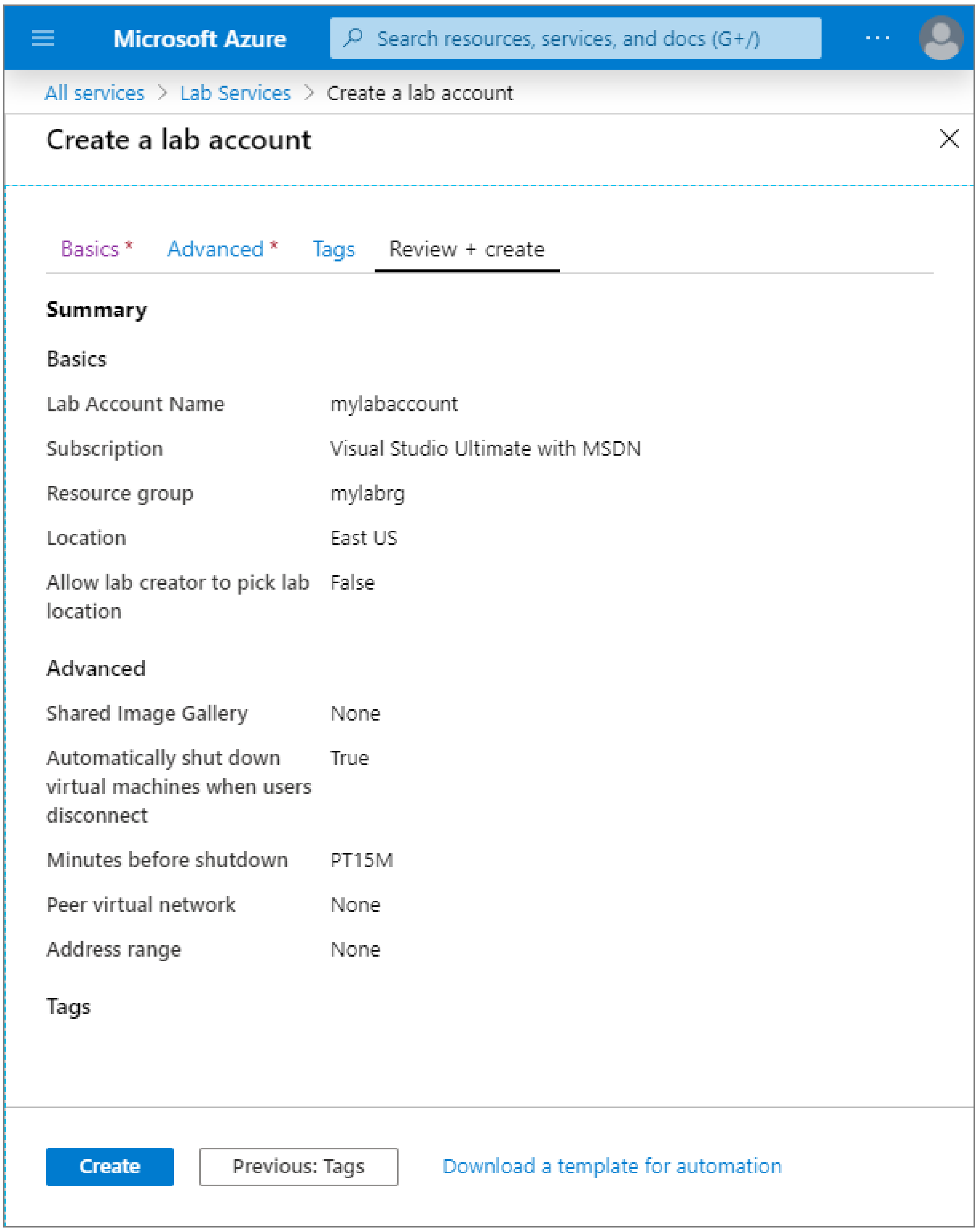
c. For **Resource group**, select an existing resource group or select **Create new**, and enter a name for the resource group.

d. For **Location**, select a location/region in which you want to create the lab account.

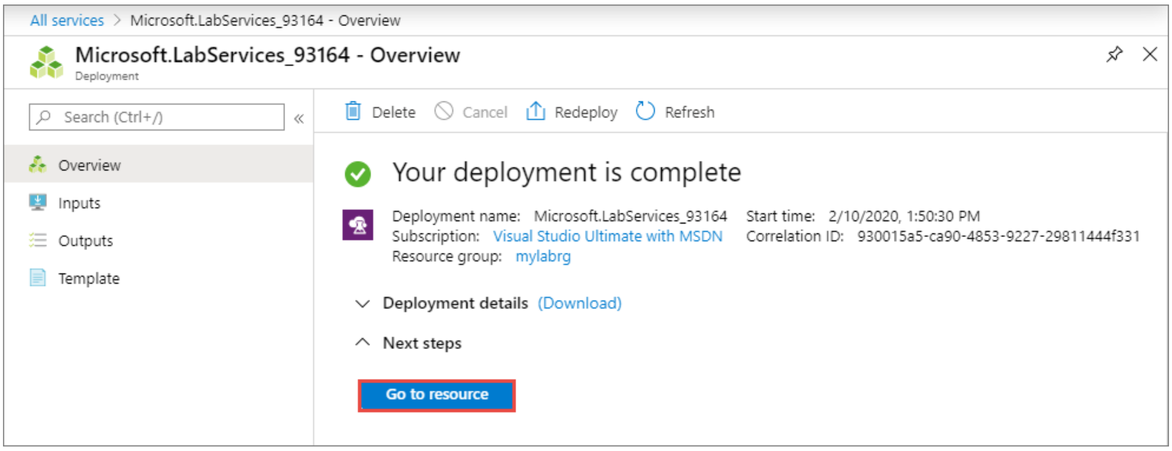


e. Select **Review + create**.

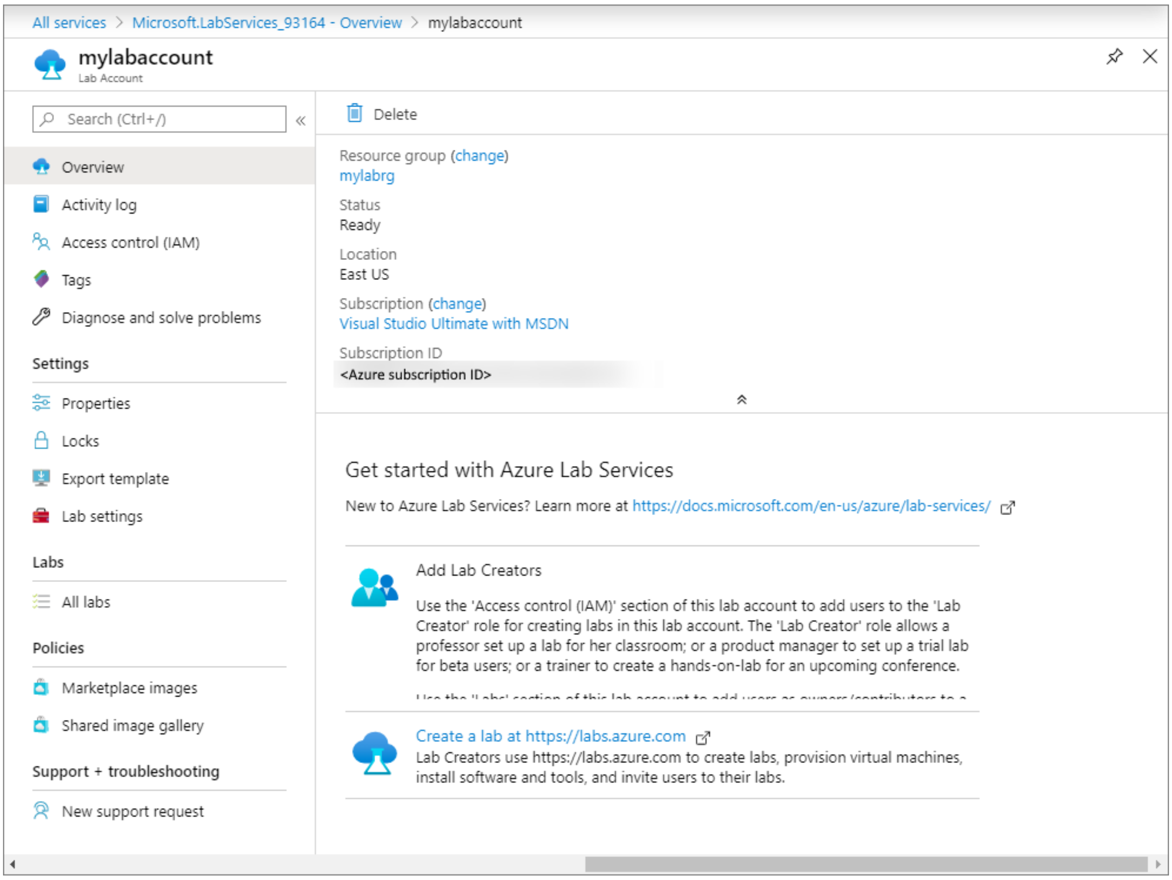
f. Review the summary, and select **Create**.



5. When the deployment is complete, expand **Next steps**, and select **Go to resource**.



6. Confirm that you see the **Lab Account** page.



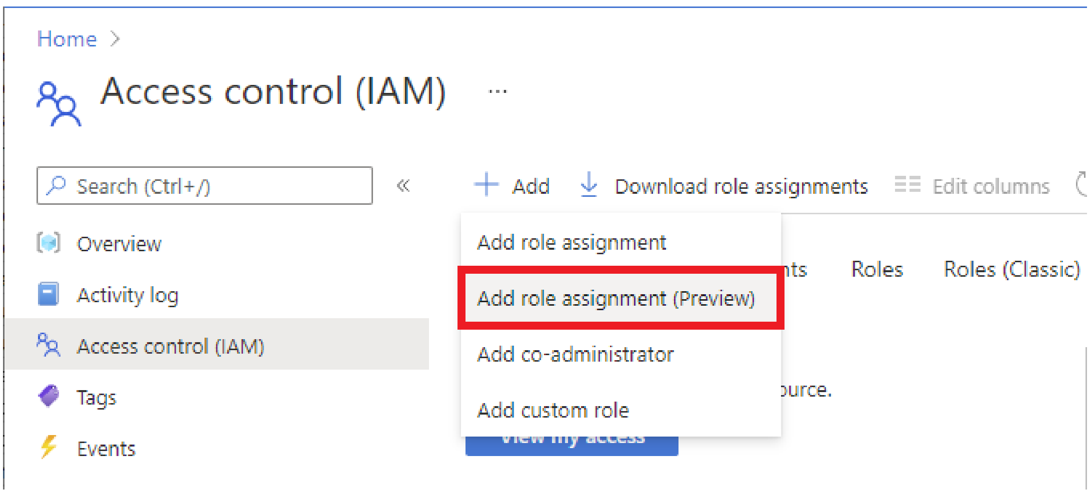
Add a user to the Lab Creator role

To set up a classroom lab in a lab account, the user must be a member of the **Lab Creator** role in the lab account. To provide educators the permission to create labs for their classes, add them to the **Lab Creator** role: For detailed steps, see Assign Azure roles using the Azure portal.

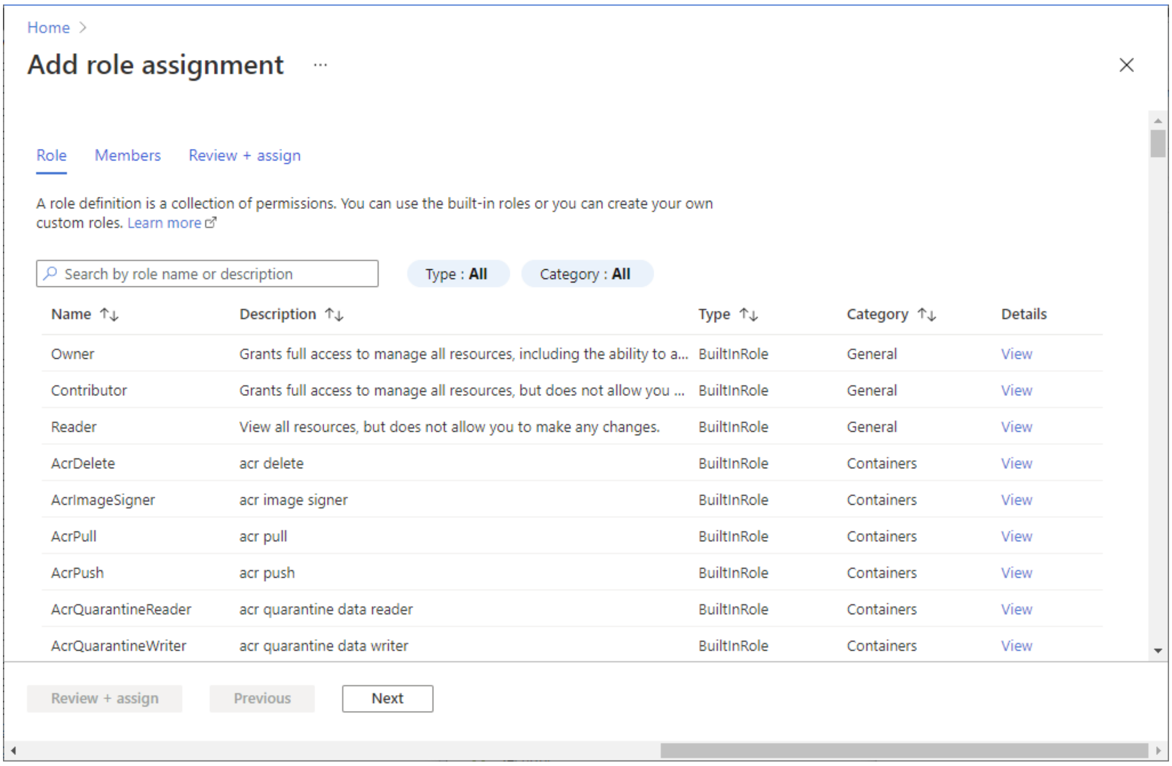
> **NOTE**
>
> The account you used to create the lab account is automatically added to this role. If you are planning to use the same user account to create a classroom lab in this tutorial, skip this step.

1. On the **Lab Account** page, select **Access control (IAM)**

2. Select **Add** > **Add role assignment (Preview)**.



3. On the **Role** tab, select the **Lab Creator** role.



4. On the **Members** tab, select the user you want to add to the Lab Creators role
5. On the **Review + assign** tab, select **Review + assign** to assign the role.

Next steps

In this tutorial, you created a lab account. To learn about how to create a classroom lab as an educator, advance to the next tutorial:

Set up a classroom lab

Tutorial: Set up a classroom lab

3/5/2021 • 9 minutes to read • Edit Online

In this tutorial, you set up a classroom lab with virtual machines that are used by students in the classroom.

In this tutorial, you do the following actions:

- Create a classroom lab
- Add users to the lab
- Set schedule for the lab
- Send invitation email to students

Prerequisites

In this tutorial, you set up a lab with virtual machines for your class. To set up a classroom lab in a lab account, you must be a member of one of these roles in the lab account: Owner, Lab Creator, or Contributor. The account you used to create a lab account is automatically added to the owner role. So, you can use the user account that you used to create a lab account to create a classroom lab.

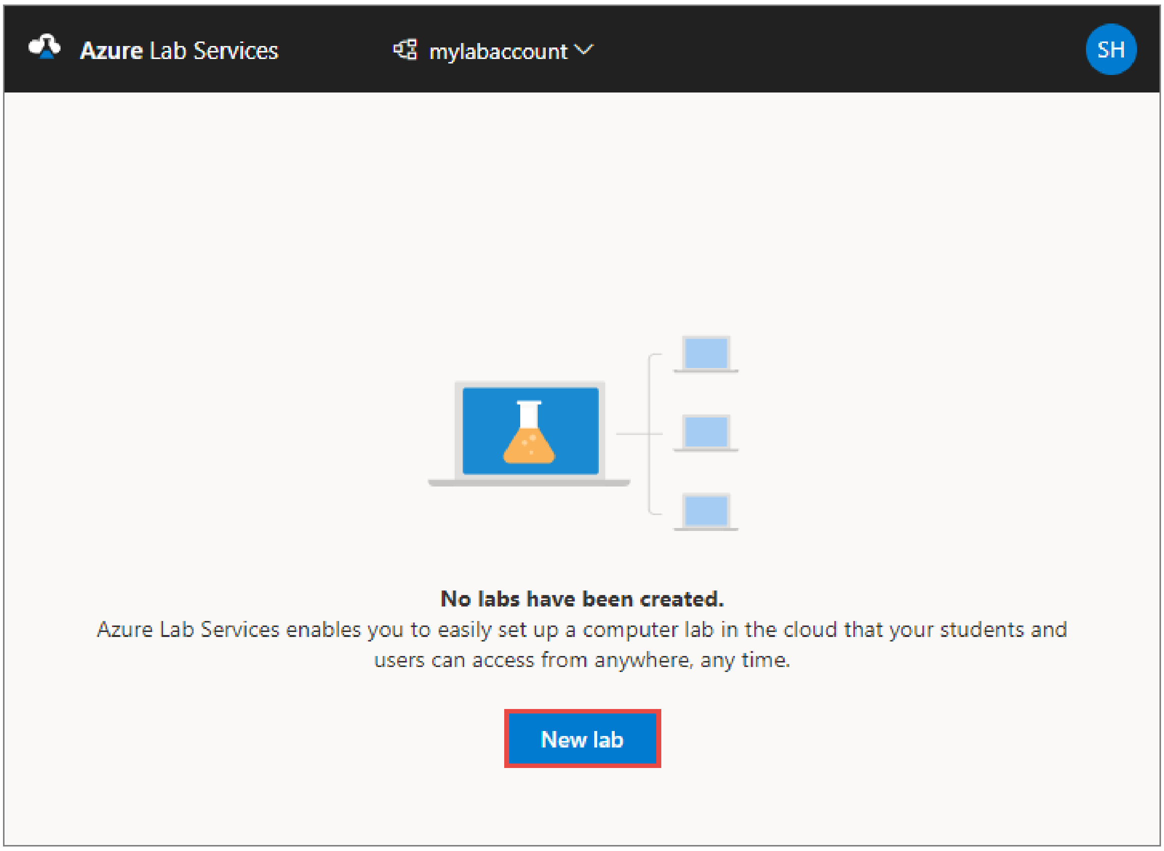
Here is the typical workflow when using Azure Lab Services:

1. A lab account creator adds other users to the **Lab Creator** role. For example, the lab account creator/admin adds educators to the **Lab Creator** role so that they can create labs for their classes.
2. Then, the educators create labs with VMs for their classes and send registration links to students in the class.
3. Students use the registration link that they receive from educators to register to the lab. Once they are registered, they can use VMs in the labs to do the class work and home work.

Create a classroom lab

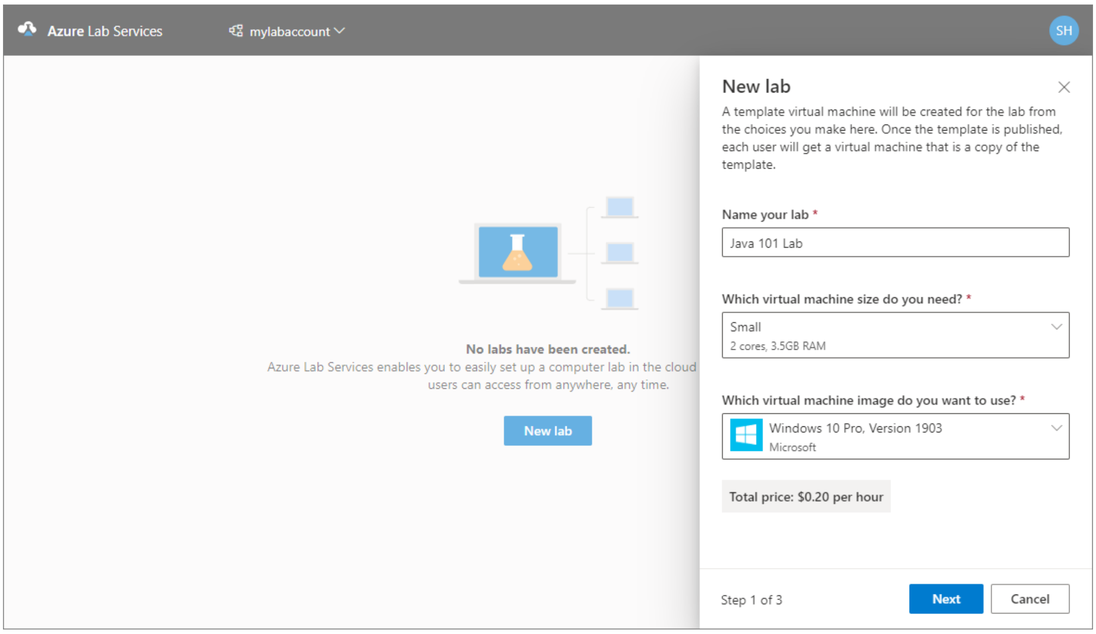
In this step, you create a lab for your class in Azure.

1. Navigate to Azure Lab Services website. Note that Internet Explorer 11 is not supported yet.
2. Select **Sign in** and enter your credentials. Azure Lab Services supports organizational accounts and Microsoft accounts.
3. Select **New lab**.

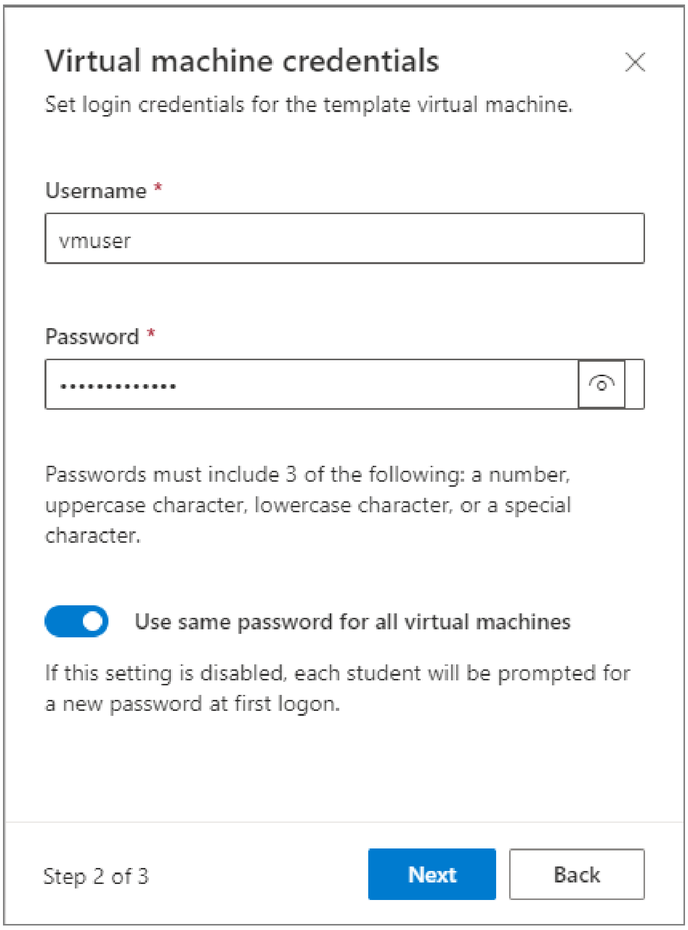


4. In the **New Lab** window, do the following actions:

   a. Specify a **name** for your lab, and select **Next**.



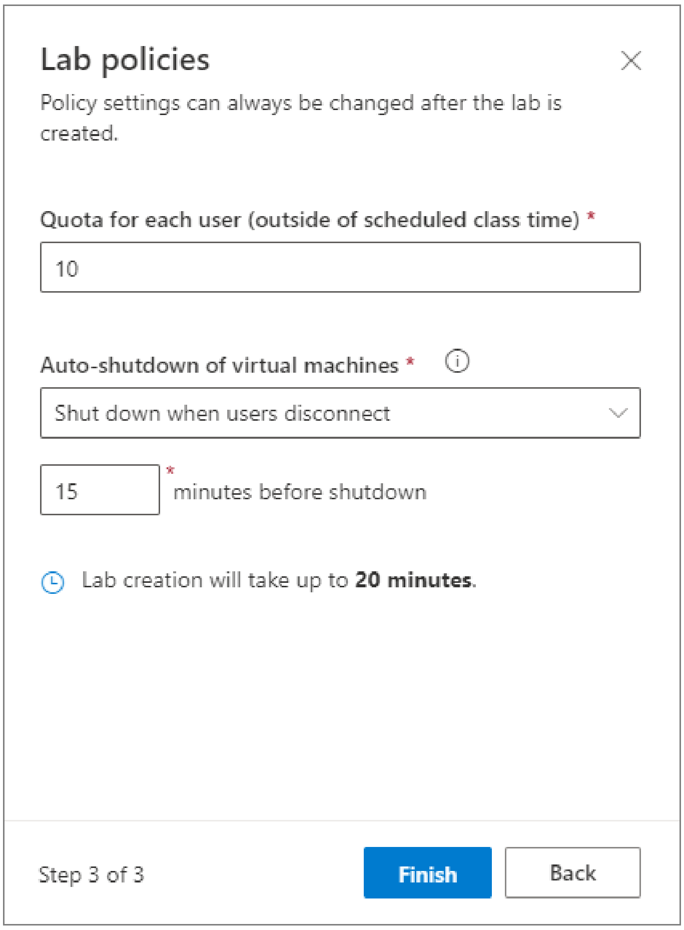
   b. On the **Virtual machine credentials** page, specify default credentials for all VMs in the lab. Specify the **name** and the **password** for the user, and then select **Next**.



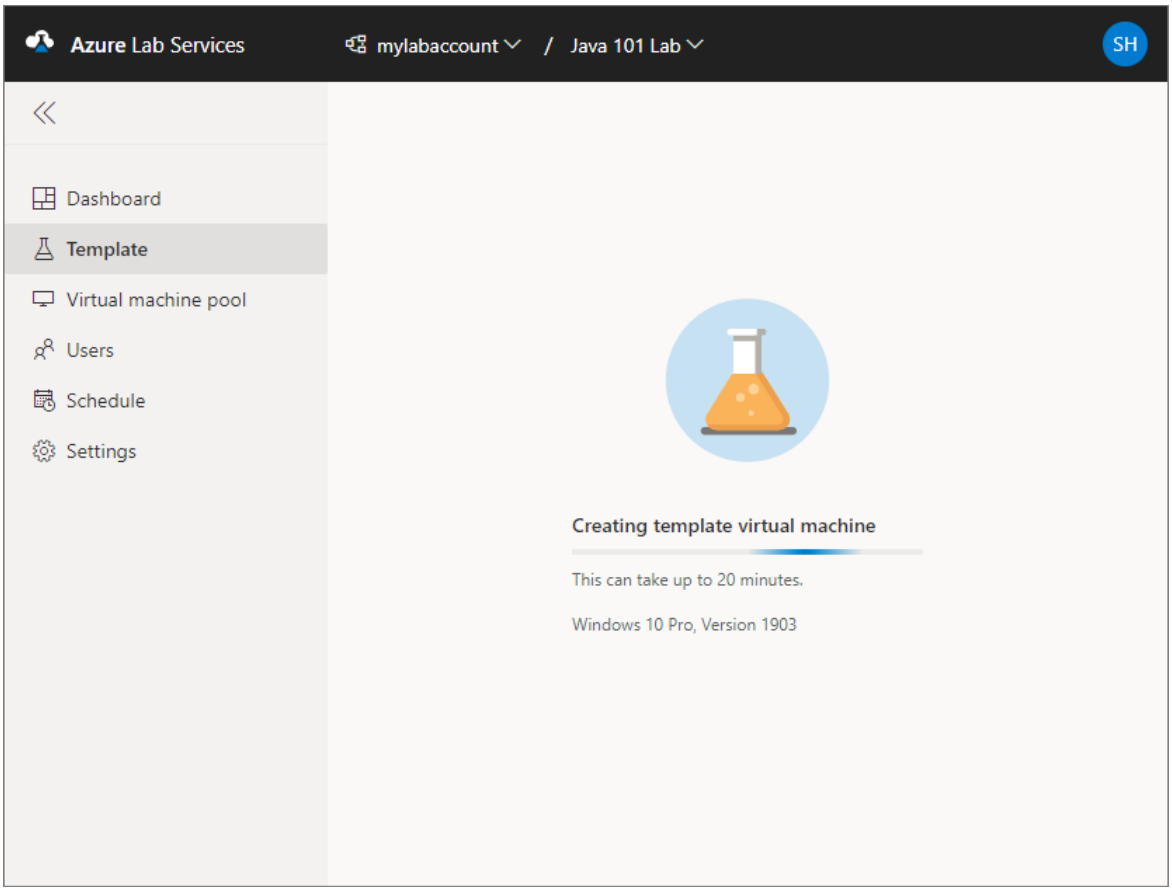
IMPORTANT

Make a note of user name and password. They won't be shown again.

c. On the **Lab policies** page, select **Finish**.



5. You should see the following screen that shows the status of the template VM creation. This operation takes up to 20 minutes.



6. On the **Template** page, do the following steps: These steps are **optional** for the tutorial.

   a. Connect to the template VM by selecting **Connect**. If it's a Linux template VM, you choose whether

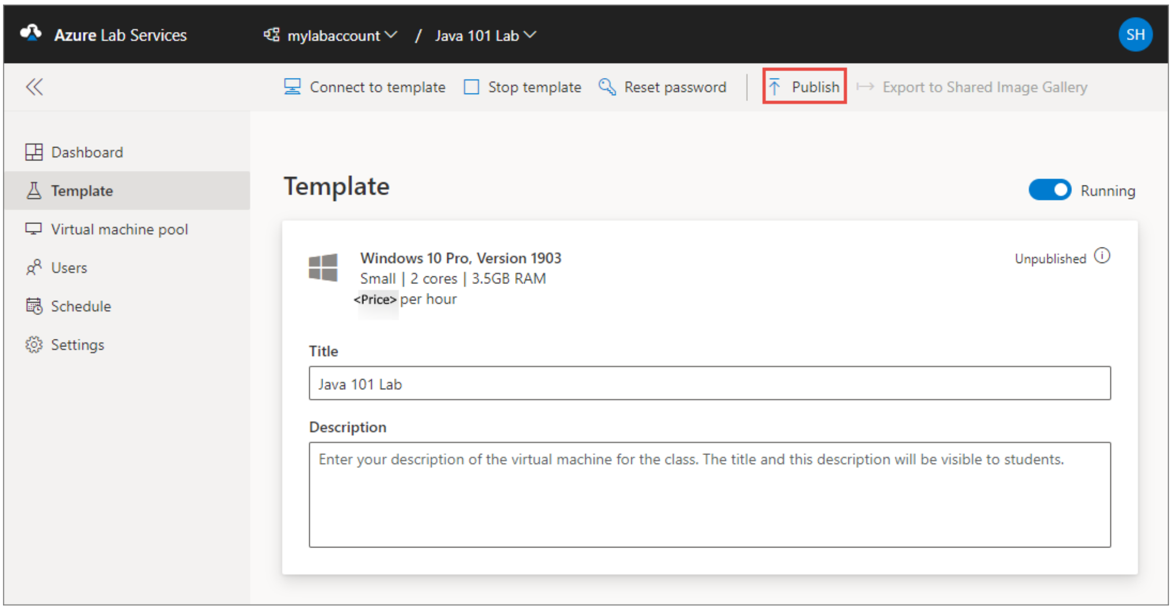
you want to connect using SSH or RDP (if RDP is enabled).
b. Install and configure software required for your class on the template VM.
c. **Stop** the template VM.

> **NOTE**
>
> Template VMs incur **cost** when running, so ensure that the template VM is shutdown when you don't need it to be running.

Publish the template VM

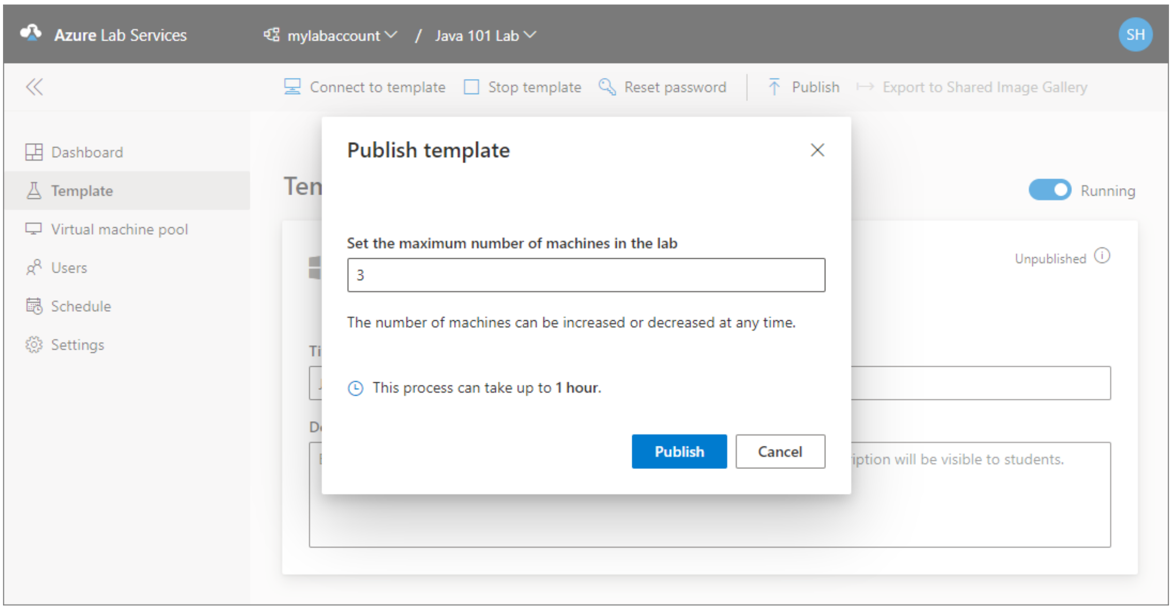
In this step, you publish the template VM. When you publish the template VM, Azure Lab Services creates VMs in the lab by using the template. All virtual machines have the same configuration as the template.

1. On the **Template** page, select **Publish** on the toolbar.

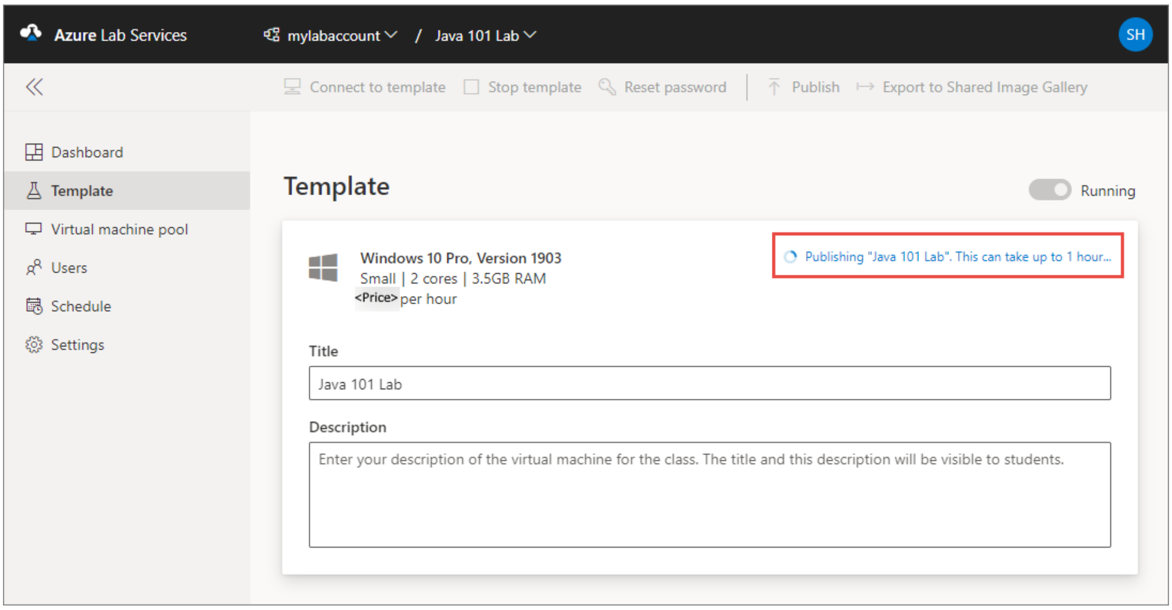


> **WARNING**
>
> Once you publish, you can't unpublish.

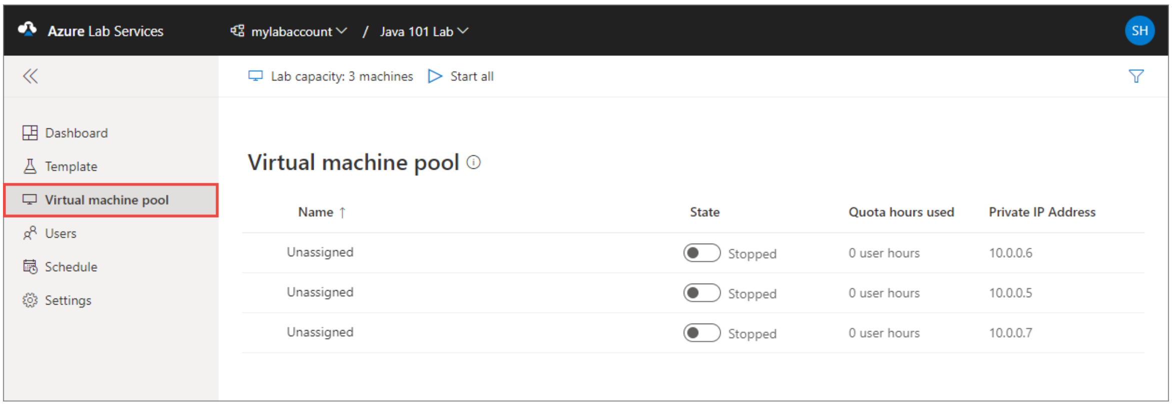
2. On the **Publish template** page, enter the number of virtual machines you want to create in the lab, and then select **Publish**.



3. You see the **status of publishing** the template on page. This process can take up to an hour.



4. Wait until the publishing is complete and then switch to the **Virtual machines pool** page by selecting **Virtual machines** on the left menu or by selecting **Virtual machines** tile. Confirm that you see virtual machines that are in **Unassigned** state. These VMs are not assigned to students yet. They should be in **Stopped** state. You can start a student VM, connect to the VM, stop the VM, and delete the VM on this page. You can start them in this page or let your students start the VMs.

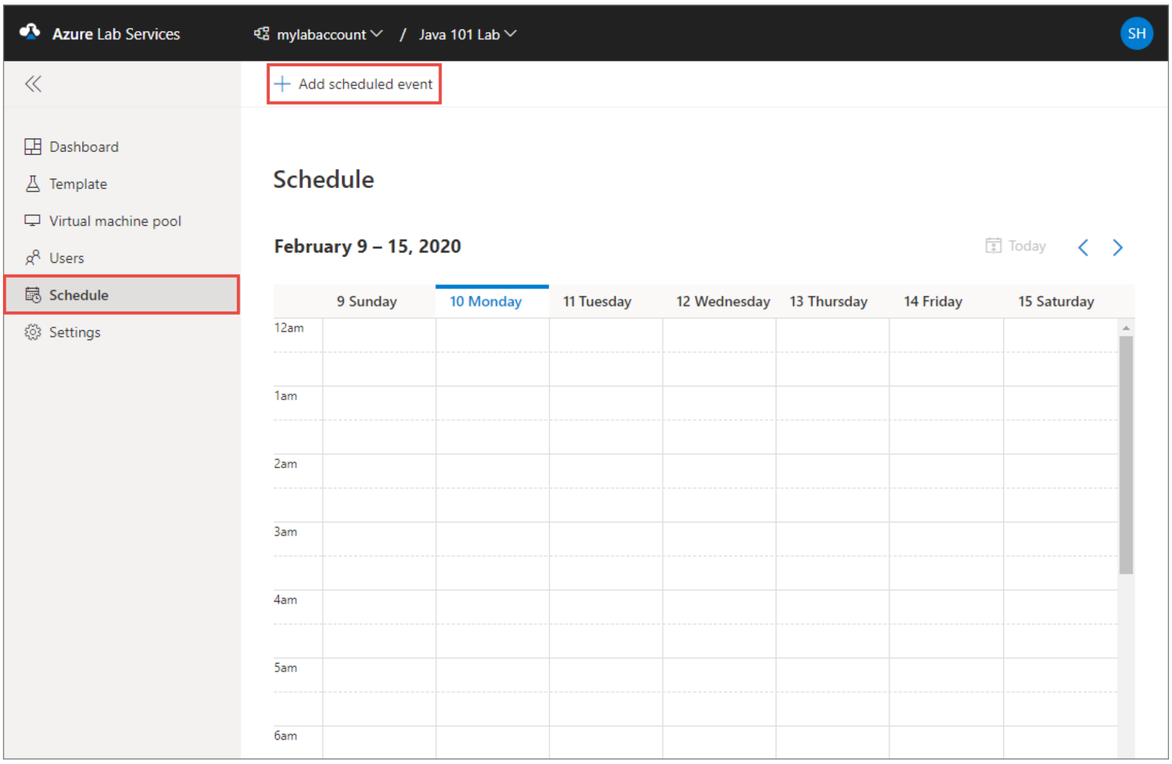


> **NOTE**
>
> When an educator turns on a student VM, quota for the student isn't affected. Quota for a user specifies the number of lab hours available to the user outside of the scheduled class time. For more information on quotas, see Set quotas for users.

Set a schedule for the lab

Create a scheduled event for the lab so that VMs in the lab are automatically started/stopped at specific times. The user quota (default: 10 hours) you specified earlier is the additional time assigned to each user outside this scheduled time.

1. Switch to the **Schedules** page, and select **Add scheduled event** on the toolbar.



2. On the **Add scheduled event** page, do the following steps:

   a. Confirm that **Standard** is selected the **Event type**.

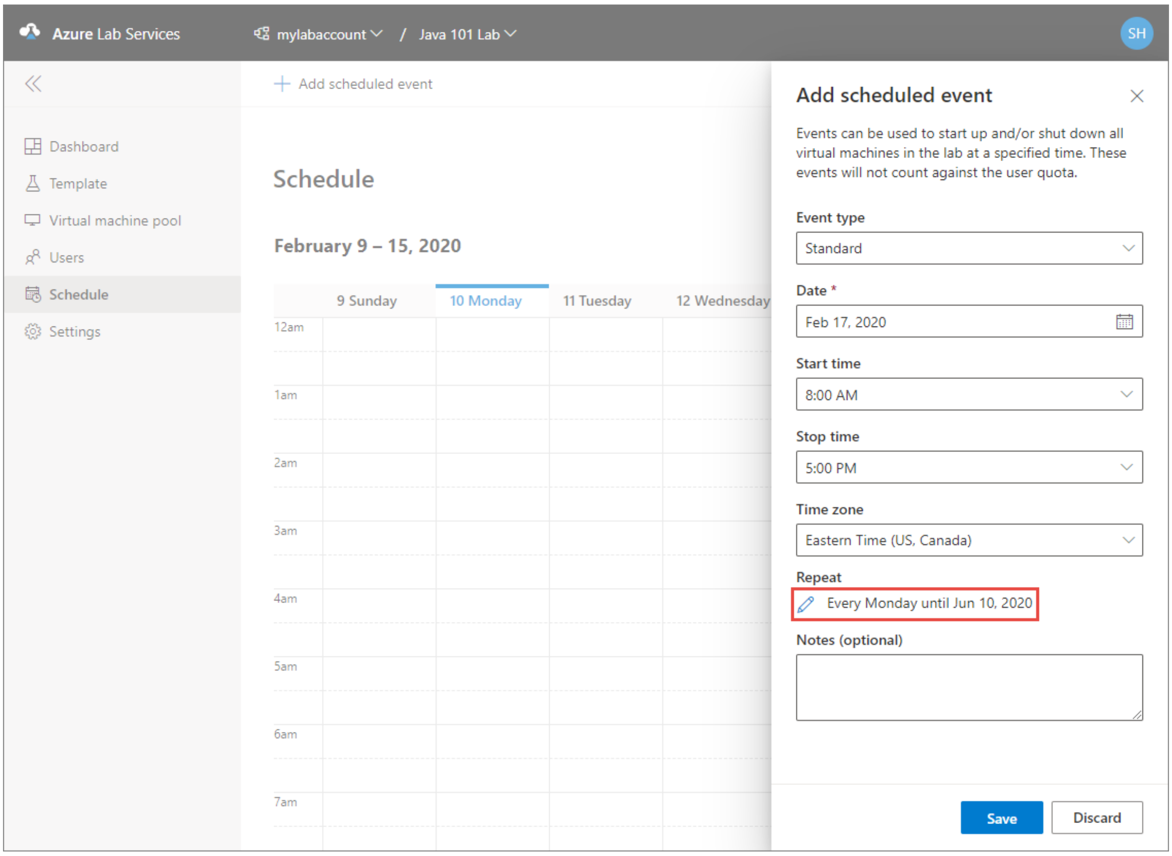
   b. Select the **start date** for the class.

   c. Select the **start time** at which you want the VMs to be started.

   d. Select the **stop time** at which the VMs are to be shut down.

   e. Select the **time zone** for the start and stop times you specified.

3. On the same **Add scheduled event** page, select the current schedule in the **Repeat** section.



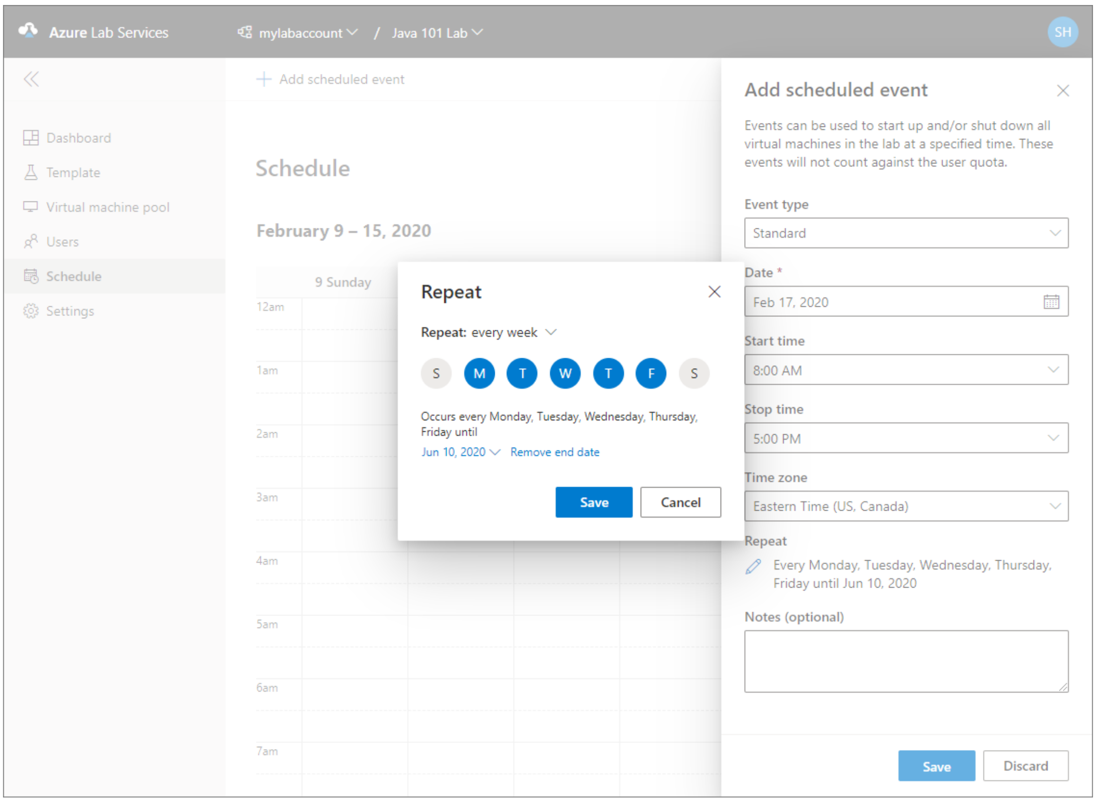
4. On the **Repeat** dialog box, do the following steps:

   a. Confirm that **every week** is set for the **Repeat** field.

   b. Select the days on which you want the schedule to take effect. In the following example, Monday-Friday is selected.

   c. Select an **end date** for the schedule.

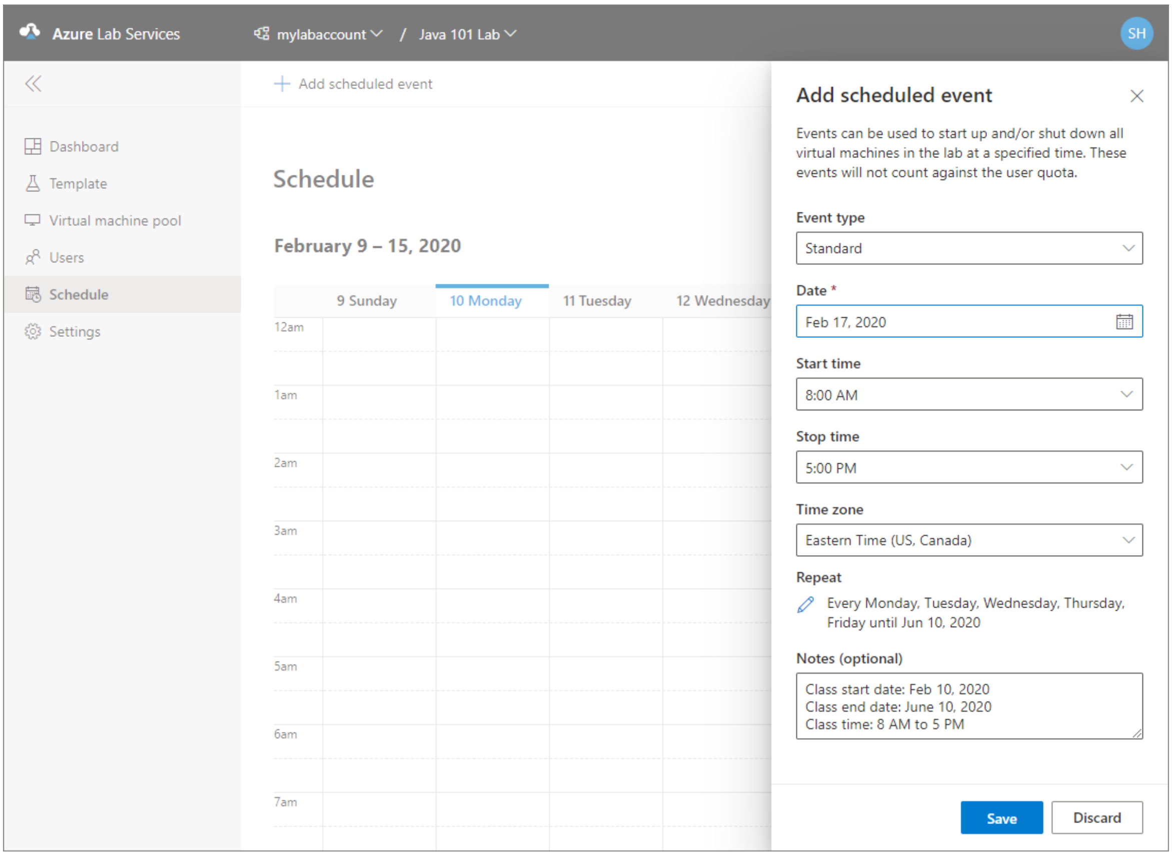
   d. Select **Save**.



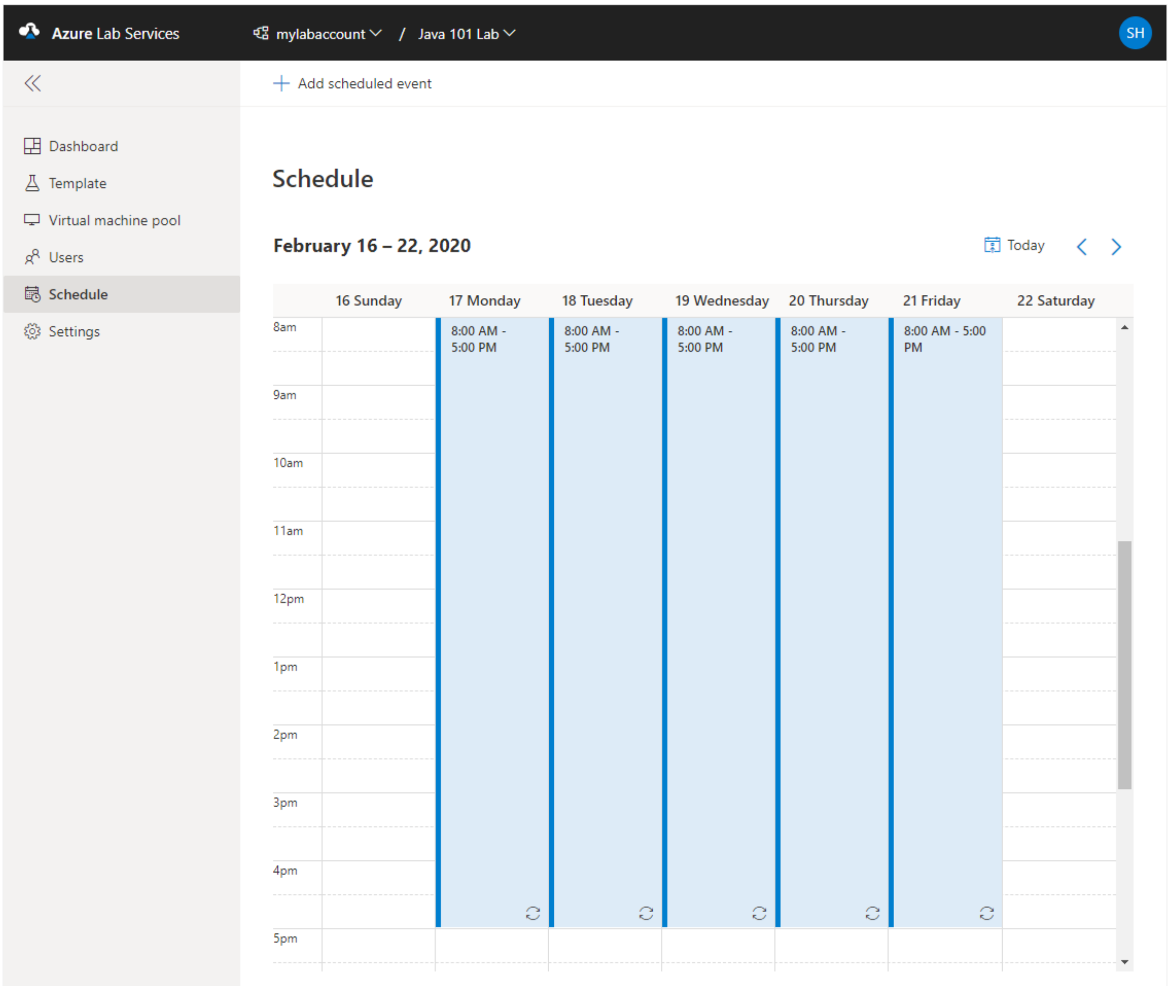
5. Now, on the **Add scheduled event** page, for **Notes (optional)**, enter any description or notes for the

schedule.

6. On the **Add scheduled event** page, select **Save**.



7. Navigate to the start date in the calendar to verify that the schedule is set.



For more information about creating and managing schedules for a class, see Create and manage schedule for labs.

Add users to the lab

When you add users, by default, the **Restrict access** option is turned on and, unless they're in the list of users, students can't register with the lab even if they have a registration link. Only listed users can register with the lab by using the registration link you send. You can turn off **Restrict access**, which allows students to register with the lab as long as they have the registration link.

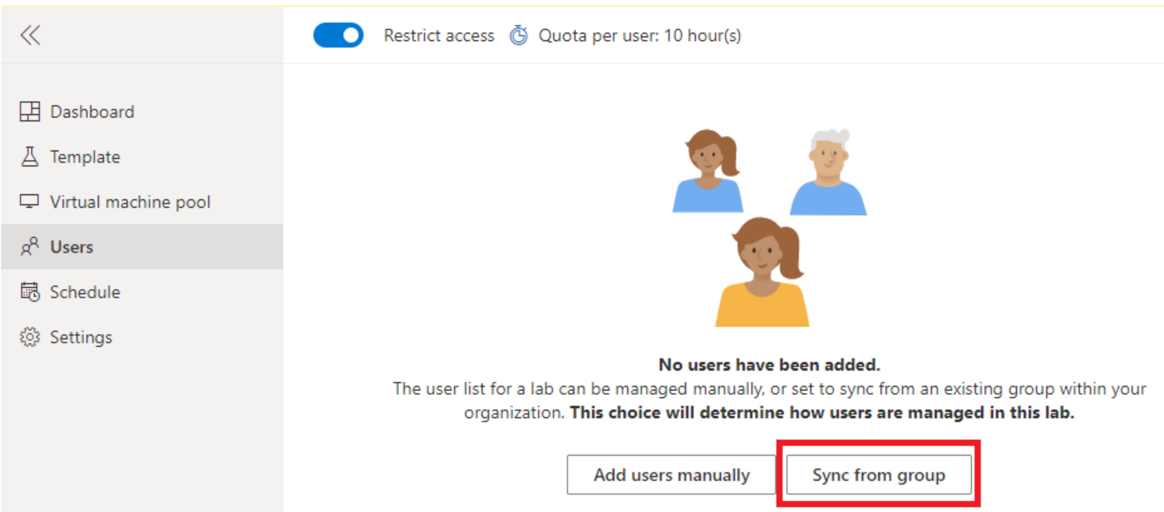
Add users from an Azure AD group

You can sync a lab user list to an existing Azure Active Directory (Azure AD) group so that you do not have to manually add or delete users.

An Azure AD group can be created within your organization's Azure Active Directory to manage access to organizational resources and cloud-based apps. To learn more, see Azure AD groups. If your organization uses Microsoft Office 365 or Azure services, your organization will already have admins who manage your Azure Active Directory.

> **IMPORTANT**
> Make sure the user list is empty. If there are existing users inside a lab that you added manually or through importing a CSV file, the option to sync the lab to an existing group will not appear.

1. In the left pane, select **Users**.
2. Click **Sync from group**.



3. You will be prompted to pick an existing Azure AD group to sync your lab to.

   If you don't see an Azure AD group in the list, could be because of the following reasons:

   - If you are a guest user for an Azure Active Directory (usually if you're outside the organization that owns the Azure AD), and you are not able to to search for groups inside the Azure AD. In this case, you won't be able to add an Azure AD group to the lab in this case.
   - Azure AD groups created through Teams do not show up in this list. You can add the Azure Lab Services app inside Teams to create and manage labs directly from within it. See more information about managing a lab's user list from within Teams.
4. Once you picked the Azure AD group to sync your lab to, click **Add**.
5. Once a lab is synced, it will pull everyone inside the Azure AD group into the lab as users, and you will

see the user list updated. Only the people in this Azure AD group will have access to your lab. The user list will refresh every 24 hours to match the latest membership of the Azure AD group. You can also click on the Sync button in the Users tab to manually sync to the latest changes in the Azure AD group.

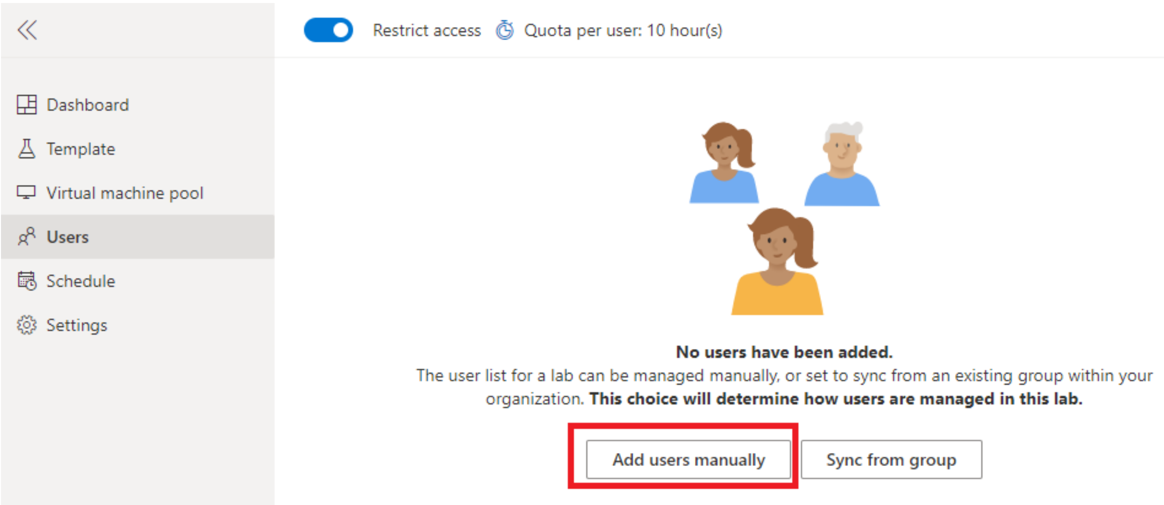
6. Invite the users to your lab by clicking on the **Invite All** button, which will send an email to all users with the registration link to the lab.

Add users manually from email(s) or CSV file

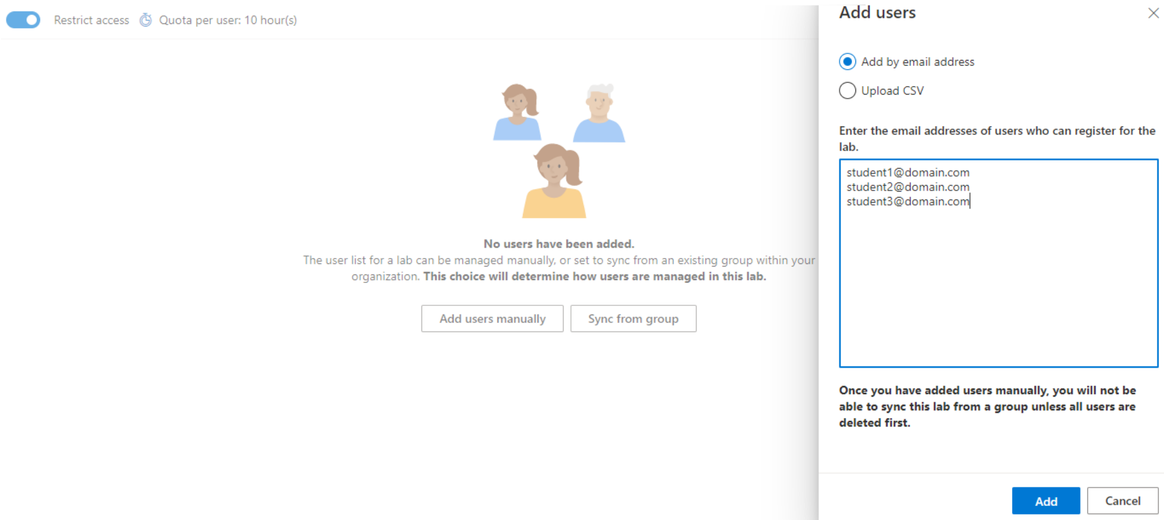
In this section, you add students manually (by email address or by uploading a CSV file).

Add users by email address

1. In the left pane, select **Users**.
2. Click **Add users manually**.

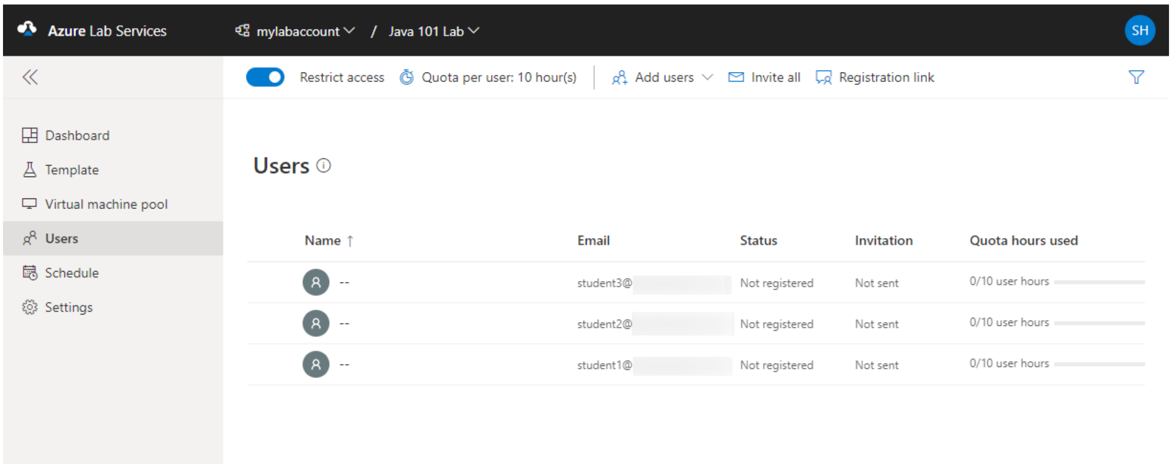


3. Select **Add by email address** (default), enter the students' email addresses on separate lines or on a single line separated by semicolons.



4. Select **Save**.

   The list displays the email addresses and statuses of the current users, whether they're registered with the lab or not.



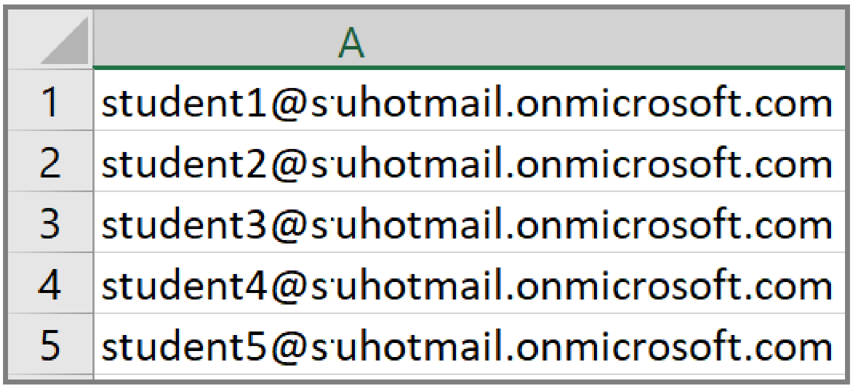
> **NOTE**
>
> After the students are registered with the lab, the list displays their names. The name that's shown in the list is constructed by using the first and last names of the students in Azure Active Directory.

Add users by uploading a CSV file

You can also add users by uploading a CSV file that contains their email addresses.

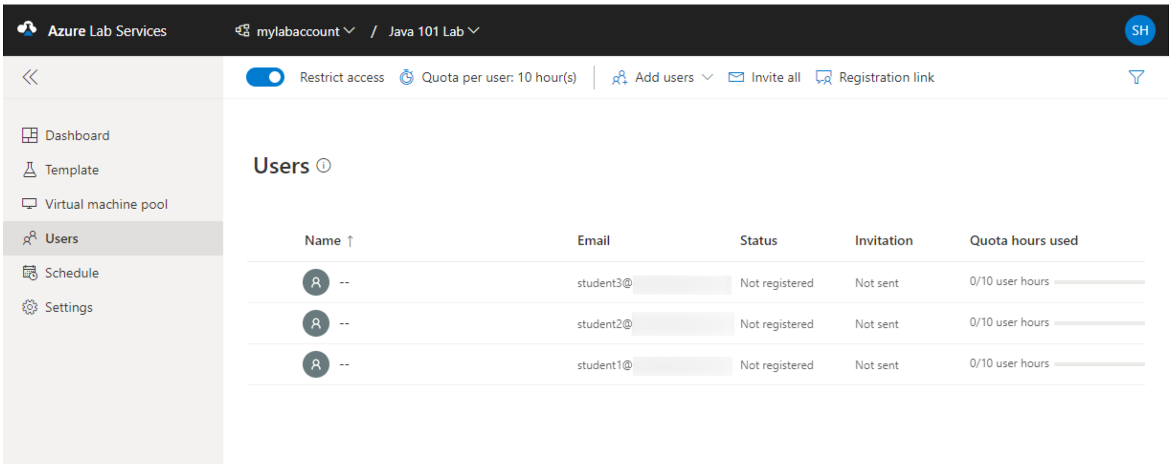
A CSV text file is used to store comma-separated (CSV) tabular data (numbers and text). Instead of storing information in columns fields (such as in spreadsheets), a CSV file stores information separated by commas. Each line in a CSV file will have the same number of comma-separated "fields." You can use Excel to easily create and edit CSV files.

1. In Microsoft Excel, create a CSV file that lists students' email addresses in one column.



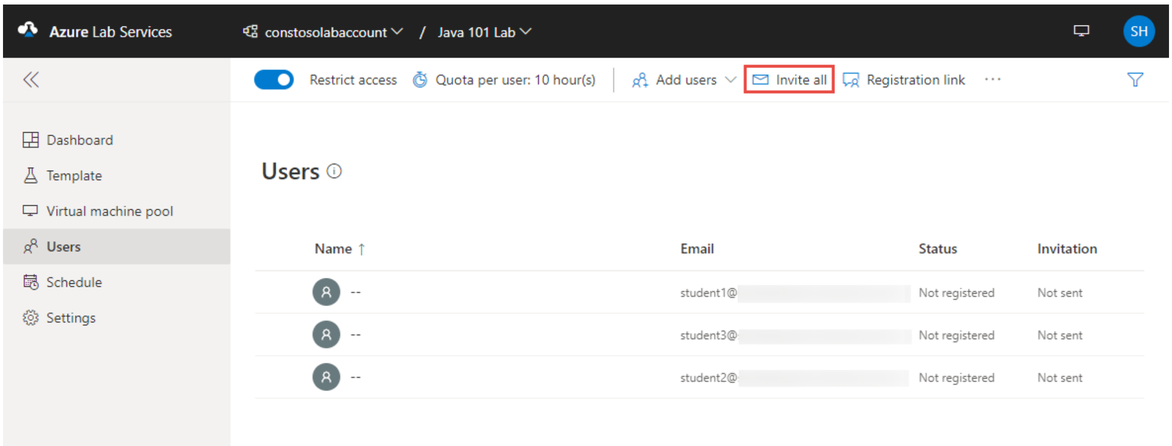
2. At the top of the Users pane, select **Add users**, and then select **Upload CSV**.
3. Select the CSV file that contains the students' email addresses, and then select **Open**.

   The **Add users** window displays the email address list from the CSV file.
4. Select **Save**.
5. In the **Users** pane, view the list of added students.

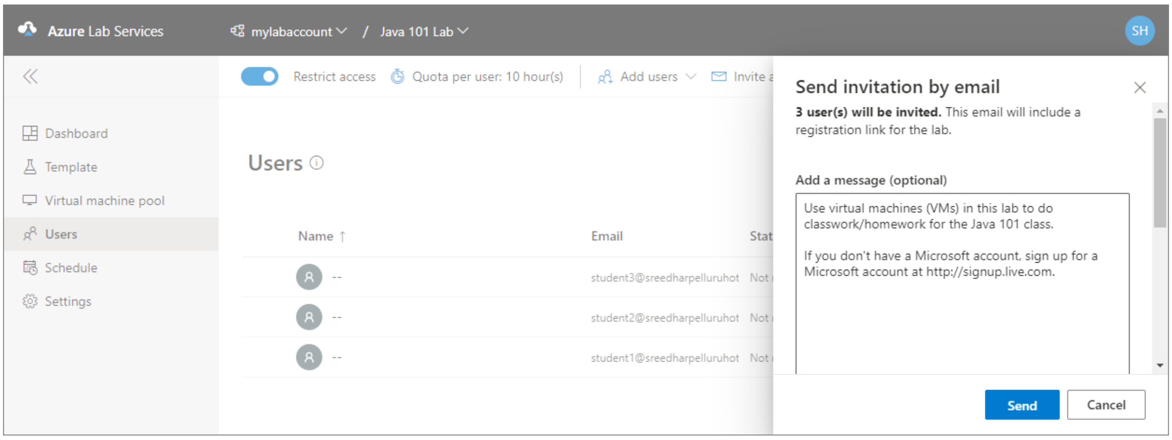


Send invitation emails to users

1. Switch to the **Users** view if you are not on the page already, and select **Invite all** on the toolbar.



2. On the **Send invitation by email** page, enter an optional message, and then select **Send**. The email automatically includes the registration link. You can get this registration link by selecting **... (ellipsis)** on the toolbar, and **Registration link**.



3. You see the status of **invitation** in the **Users** list. The status should change to **Sending** and then to **Sent on <date>**.

For more information about adding students to a class and managing their usage of the lab, see How to configure student usage.

Next steps

In this tutorial, you created a lab for your class in Azure. To learn how a student can access a VM in the lab using the registration link, advance to the next tutorial:

Connect to a VM in the classroom lab

Tutorial: Access a classroom lab in Azure Lab Services

11/2/2020 • 2 minutes to read • Edit Online

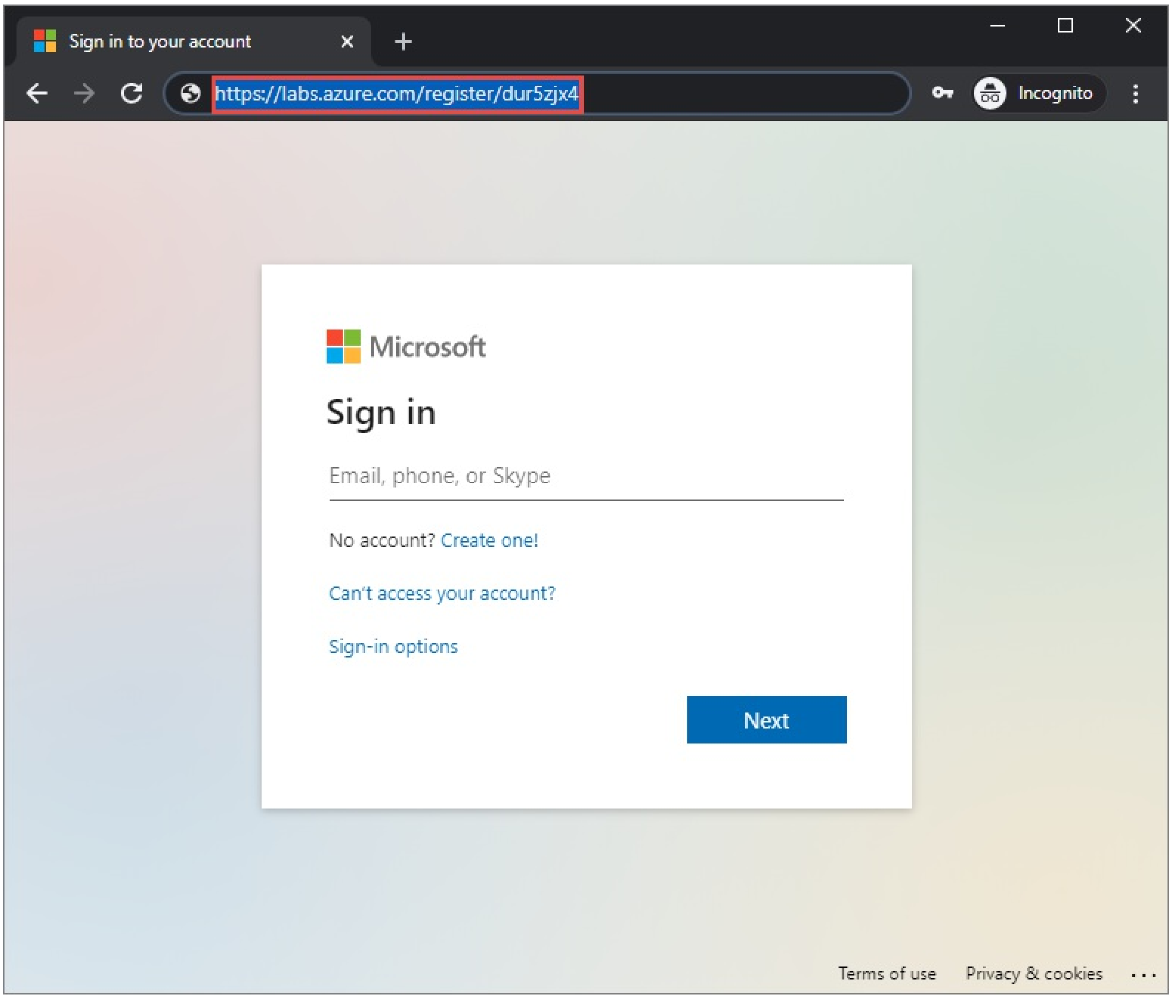
In this tutorial, you, as a student, connect to a virtual machine (VM) in a classroom lab.

In this tutorial, you do the following actions:

- Register to the lab
- Start the VM
- Connect to the VM

Register to the lab

1. Navigate to the **registration URL** that you received from the educator. You don't need to use the registration URL after you complete the registration. Instead, use the URL: https://labs.azure.com. Internet Explorer 11 isn't supported yet.

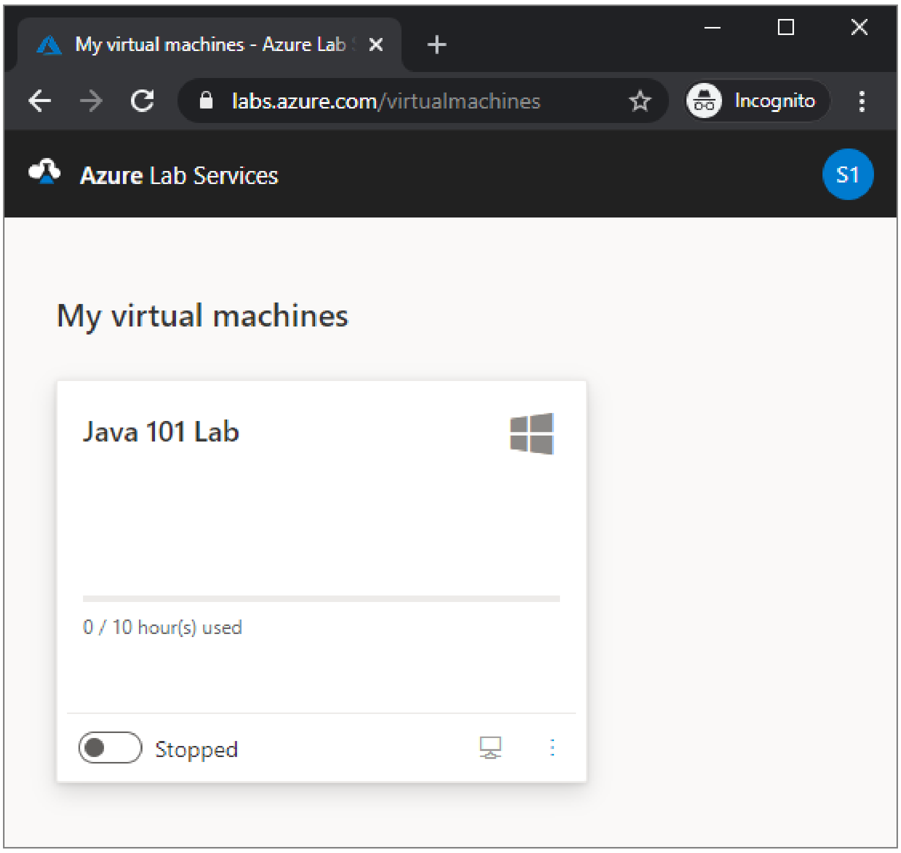


2. Sign in to the service using your school account to complete the registration.

NOTE

A Microsoft account is required for using Azure Lab Services. If you are trying to use your non-Microsoft account such as Yahoo or Google accounts to sign in to the portal, follow instructions to create a Microsoft account that will be linked to your non-Microsoft account. Then, follow the steps to complete the registration process.

3. Once registered, confirm that you see the virtual machine for the lab you have access to.



4. Wait until the virtual machine is ready. On the VM tile, notice the following fields:

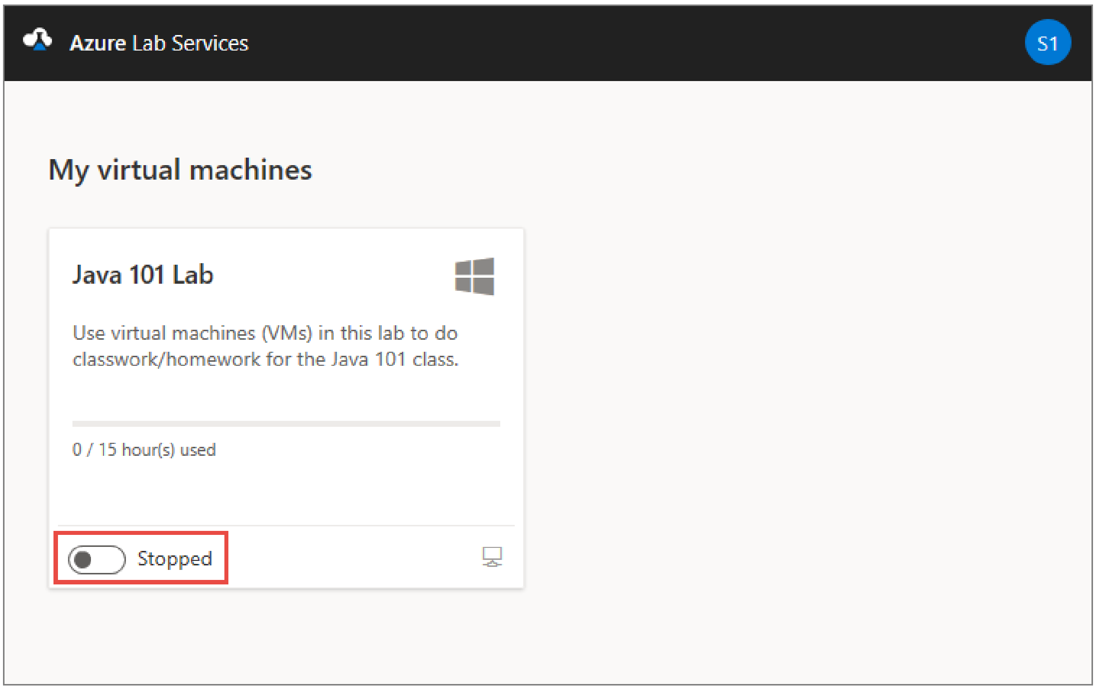
   a. At the top of the tile, you see the **name of the lab**.

   b. To its right, you see the icon representing the **operating system (OS)** of the VM. In this example, it's Windows OS.

   c. The progress bar on the tile shows the number of hours used against the number of quota hours assigned to you. This time is the additional time allotted to you in addition to the scheduled time for the lab.

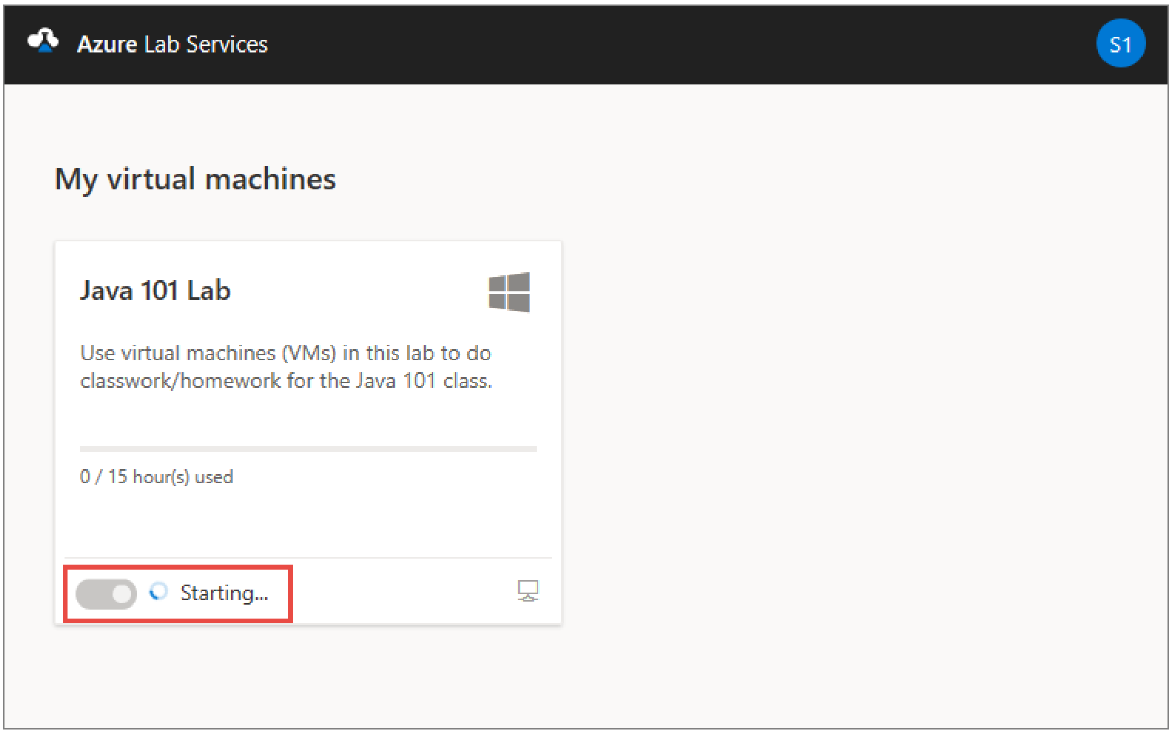
   d. You see icons/buttons at the bottom of the tile to start/stop the VM, and connect to the VM.

   e. To the right of the buttons, you see the status of the VM. Confirm that you see the status of the VM is **Stopped**.

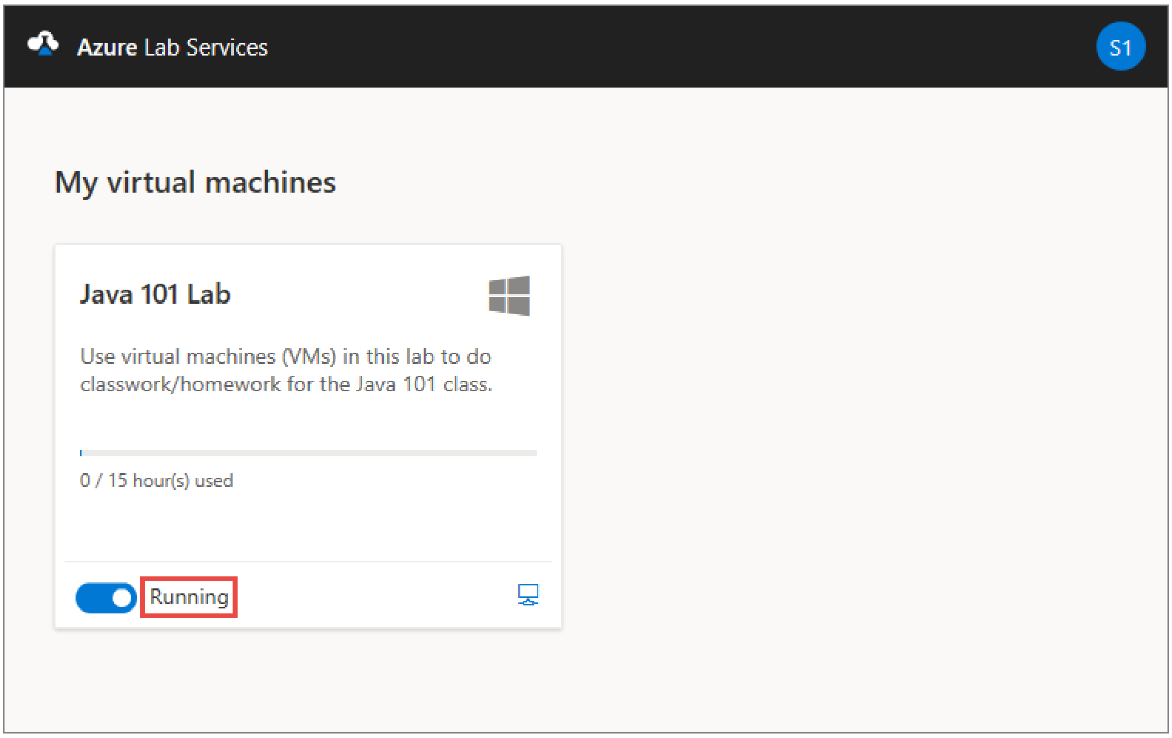


Start the VM

1. **Start** the VM by selecting the first button as shown in the following image. This process takes some time.



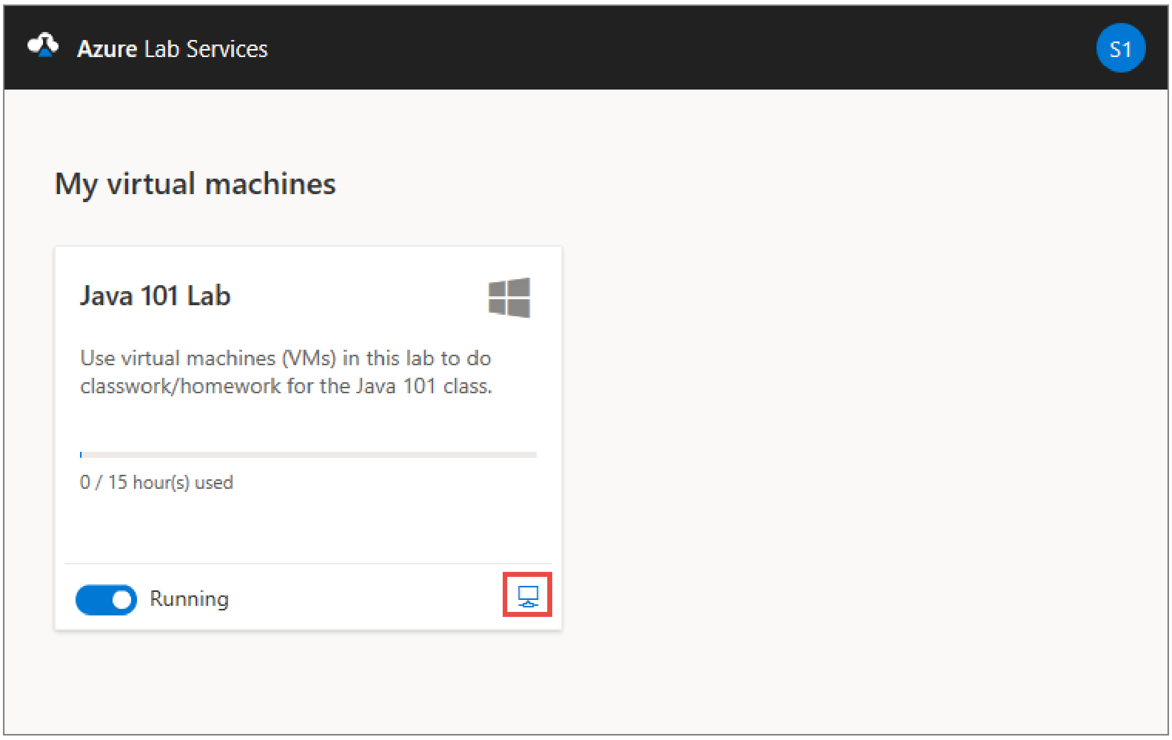
2. Confirm that the status of the VM is set to **Running**.



Notice that the icon of the first button changed to represent a **stop** operation. You can select this button to stop the VM.

Connect to the VM

1. Select the second button as shown in the following image to **connect** to the lab's VM.



2. Do one of the following steps:

   a. For **Windows** virtual machines, save the **RDP** file to the hard disk. Open the RDP file to connect to the virtual machine. Use the **user name** and **password** you get from your educator to sign in to the machine.

   b. For **Linux** virtual machines, you can use **SSH** or **RDP** (if it's enabled) to connect to them. For more information, see Enable remote desktop connection for Linux machines.

Next steps

In this tutorial, you accessed a classroom lab using the registration link you get from your educator.

As a lab owner, you want to view who has registered with your lab and track the usage of VMs. Advance to the next tutorial to learn how to track the usage of the lab:

Track usage of a lab

Tutorial: Track usage of a lab in Azure Lab Service

3/5/2021 • 2 minutes to read • Edit Online

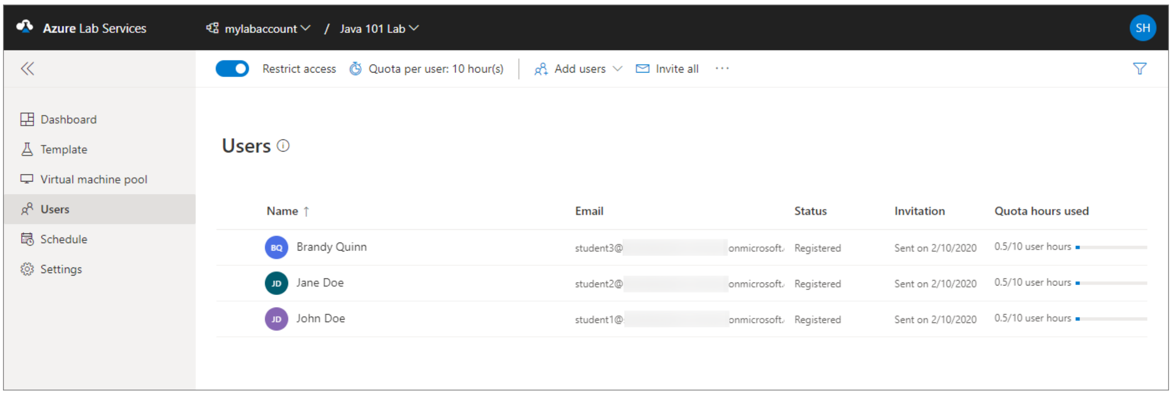
This tutorial shows you how a lab creator/owner can track usage of a lab.

In this tutorial, you do the following actions:

- View users registered with your lab
- View the usage of VMs in the lab
- Manage student VMs

View registered users

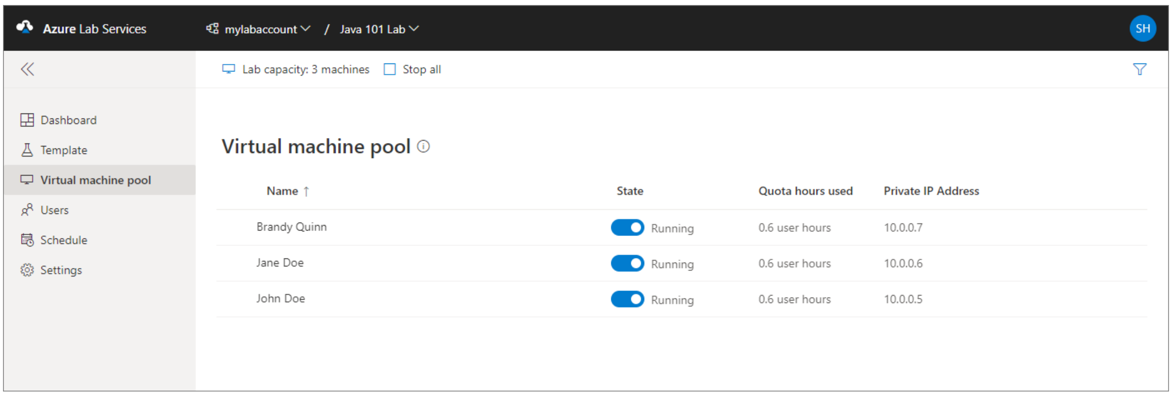
1. Navigate to Azure Lab Services website.
2. Select **Sign in** and enter your credentials. Azure Lab Services supports organizational accounts and Microsoft accounts.
3. On the **My labs** page, select the lab for which you want to track the usage.
4. Select **Users** on the left menu or **Users** tile. You see students who have registered with your lab.



For more information about adding and managing users for the lab, see Add and manage lab users.

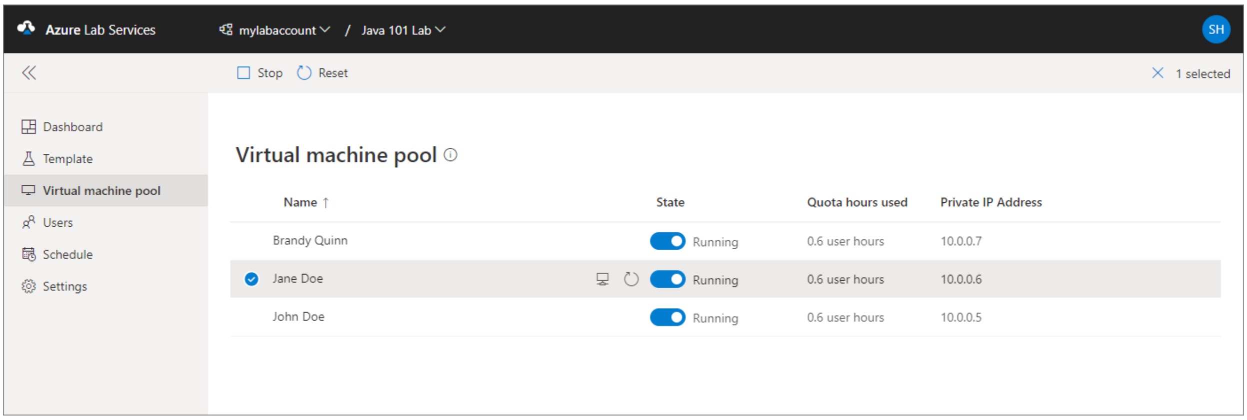
View the usage of VMs

1. Select **Virtual machines** on menu to the left.
2. Confirm that you see the status of VMs and the number of hours the VMs have been running. The time that a lab owner spends on a student VM doesn't count against the usage time shown in the last column.



Manage student VMs

On this page, you can start, stop, or reset student VMs by using controls in the **State** column or on the toolbar.



For more information about managing virtual machine pool for the lab, see Set up and manage virtual machine pool.

> **NOTE**
>
> When an educator turns on a student VM, quota for the student isn't affected. Quota for a user specifies the number of lab hours available to the user outside of the scheduled class time. For more information on quotas, see Set quotas for users.

Next steps

To learn more about labs, see articles under How-to guides.

Introduction to labs

8/26/2021 • 2 minutes to read • Edit Online

Azure Lab Services enables you to quickly set up a classroom lab environment in the cloud. An educator creates a classroom lab, provisions Windows, or Linux virtual machines, installs the necessary software and tools labs in the class, and makes them available to students. The students in the class connect to virtual machines (VMs) in the lab, and use them for their projects, assignments, classroom exercises.

The labs are managed lab types that are managed by Azure. The service itself handles all the infrastructure management for a managed lab type, from spinning up virtual machines (VMs) to handling errors, and scaling the infrastructure. You specify what kind of infrastructure you need and install any tools or software that's required for the class.

Automatic management of Azure infrastructure and scale

Azure Lab Services is a managed service, which means that provisioning and management of a lab's underlying infrastructure is handled automatically by the service. You can just focus on preparing the right lab experience for your users. Let the service handle the rest and roll out your lab's virtual machines to your audience. Scale your lab to hundreds of virtual machines with a single click.

Simple experience for your lab users

Users who are invited to your lab get immediate access to the resources you give them inside your labs. They just need to sign in to see the full list of virtual machines they have access to across multiple labs. They can click on a single button to connect to the virtual machines and start working. Users don't need Azure subscriptions to use the service.

Cost optimization and tracking

Keep your budget in check by controlling exactly how many hours your lab users can use the virtual machines. Set up schedules in the lab to allow users to use the virtual machines only during designated time slots or set up reoccurring auto-shutdown and start times. Keep track of individual users' usage and set limits.

Example class types

You can set up labs for several types of classes with Azure Lab Services. See the Example class types on Azure Lab Services article for a few example types of classes for which you can set up labs with Azure Lab Services.

Next steps

Get started with setting up a lab account that's required to create a classroom lab using Azure Lab Services:

- Set up a lab account

Labs concepts

3/5/2021 • 2 minutes to read • Edit Online

The following list contains key Lab Services concepts and definitions:

Quota

Quota is the time limit (in hours) that an educator can set for a student to use a lab VM. It can be set to 0, or a specific number of hours. If the quota is set to 0, a student can only use the virtual machine when a schedule is running or when an educator manually turns on the virtual machine for the student.

Quota hours are counted when the student starts the lab VM themselves. If an educator manually starts the lab VM for a student, quota hours aren't used for that student.

Schedules

Schedules are the time slots that an educator can create for the class so the student VMs are available for class time. Schedules can be one-time or recurring. Quota hours aren't used when a schedule is running.

There are three types of schedules: Standard, Start only and Stop only.

- **Standard**. This schedule will start all student VMs at the specified start time and shutdown all student VMs at the specified stop time.
- **Start only**. This schedule will start all student VMs at the specified time. Student VMs won't be stop until a student stops the their VM through the Azure Lab Services portal or a stop only schedule occurs.
- **Stop only**. This schedule will stop all student VMs at the specified time.

Template virtual machine

A template virtual machine in a lab is a base virtual machine image from which all users' virtual machines are created. Trainers/lab creators set up the template virtual machine and configure it with the software that they want to provide to training attendees to do labs. When you publish a template VM, Azure Lab Services creates or updates lab VMs based on the template VM.

User profiles

This article describes different user profiles in Azure Lab Services.

Lab account owner

Typically, an IT administrator of organization's cloud resources, who owns the Azure subscription acts as a lab account owner and does the following tasks:

- Sets up a lab account for your organization.
- Manages and configures policies across all labs.
- Gives permissions to people in the organization to create a lab under the lab account.

Educator

Typically, users such as a teacher or an online trainer creates labs under a lab account. An educator does the following tasks:

- Creates a classroom lab.

- Creates virtual machines in the lab.
- Installs the appropriate software on virtual machines.
- Specifies who can access the lab.
- Provides registration link to the lab to students.

Student

A student does the following tasks:

- Uses the registration link that the lab user receives from a lab creator to register with the lab.
- Connects to a virtual machine in the lab and use it for doing class work, assignments, and projects.

Next steps

Get started with setting up a lab account that's required to create a classroom lab using Azure Lab Services:

- Set up a lab account

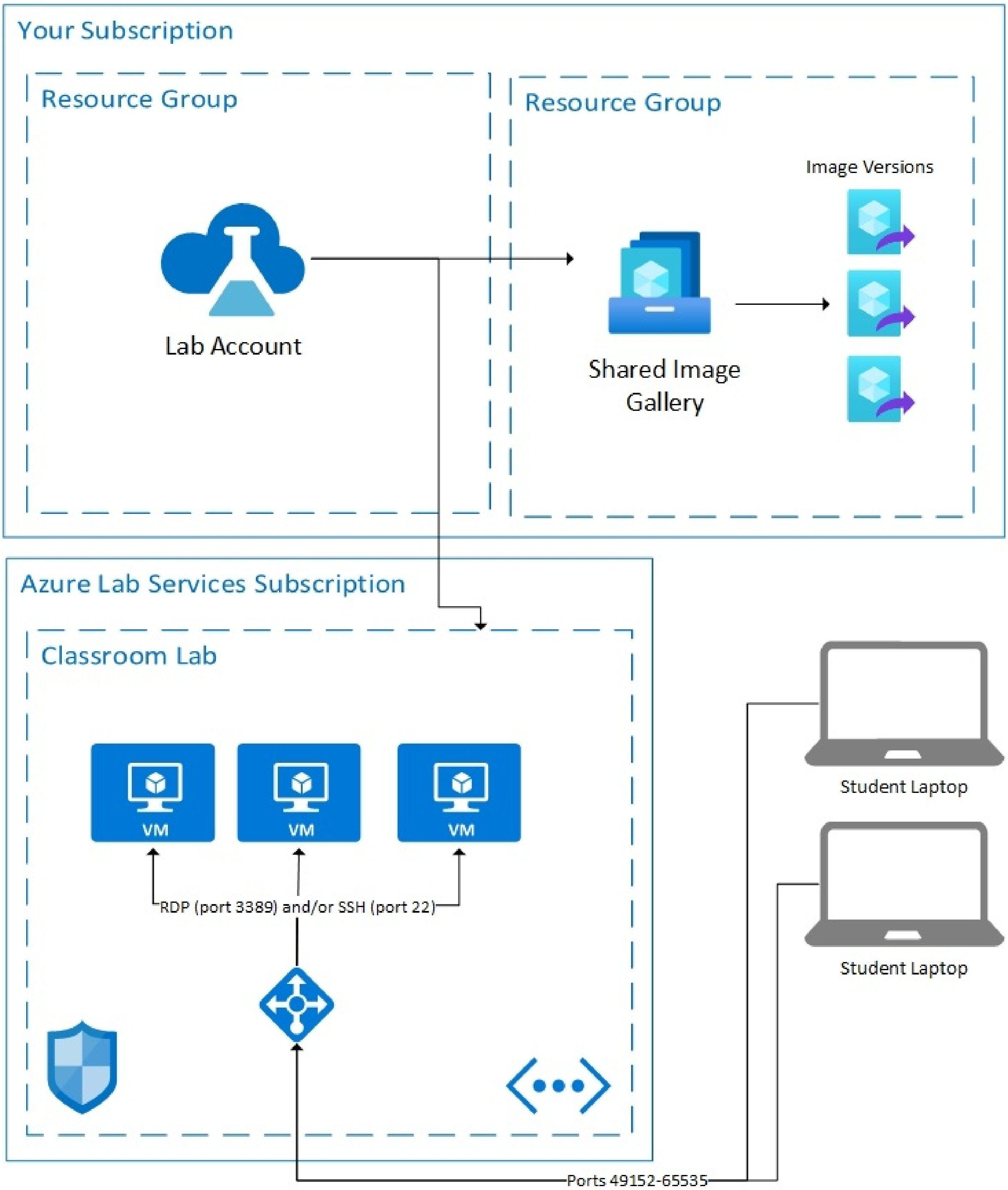
Architecture Fundamentals in Azure Lab Services

3/5/2021 • 2 minutes to read • Edit Online

Azure Lab Services is a SaaS (software as a service) solution, which means that the resources needed by Lab Services are handled for you. This article will cover the fundamental resources used by Lab Services and basic architecture of a lab.

Azure Lab Services does provide a couple of areas that allow you to use your own resources in conjunction with Lab Services. For more information about using VMs on your own network, see how to peer a virtual network. To reuse images from a Shared Image Gallery, see how to attach a Shared Image Gallery.

Below is the basic architecture of a classroom lab. The lab account is hosted in your subscription. The student VMs, along with the resources needed to support the VMs are hosted in a subscription owned by Lab Services. Let's talk about what is in Lab Service's subscriptions in more detail.



Hosted Resources

The resources required to run a classroom lab are hosted in one of the Microsoft-managed Azure subscriptions. Resources include a template virtual machine for the instructor, virtual machine for each student, and network-related items such as a load balancer, virtual network, and network security group. These subscriptions are monitored for suspicious activity. It is important to note that this monitoring is done externally to the virtual machines through VM extension or network pattern monitoring. If shutdown on disconnect is enabled, a diagnostic extension is enabled on the virtual machine. The extension allows Lab Services to be informed of the remote desktop protocol (RDP) session disconnect event.

Virtual Network

Each lab is isolated by its own virtual network. If the lab has a peered virtual network, then each lab is isolated by its own subnet. Students connect to their virtual machine through a load balancer. No student virtual machines have a public IP address; they only have a private ip address. The connection string for the student will be the public IP address of the load balancer and a random port between 49152 and 65535. Inbound rules on the load balancer forward the connection, depending on the operating system, to either port 22 (SSH) or port 3389 (RDP) of the appropriate virtual machine. An NSG prevents outside traffic on any other ports.

Access control to the virtual machines

Lab Services handles the student's ability to perform actions like start and stop on their virtual machines. It also controls access to their VM connection information.

Lab Services also handles the registration of students to the service. There are currently two different access settings: restricted and nonrestricted. For more information, see the manage lab users article. Restricted access means Lab Services verifies that the students are added as user before allowing access. Nonrestricted means any user can register as long as they have the registration link and there is capacity in the lab. Nonrestricted can be useful for hackathon events.

Student VMs that are hosted in the classroom lab have a username and password set by the creator of the lab. Alternately, the creator of the lab can allow registered students to choose their own password on first sign-in.

Next steps

To learn more about features available in Lab Services, see Azure Lab Services concepts and Azure Lab Services overview.

Capacity limits in Azure Lab Services

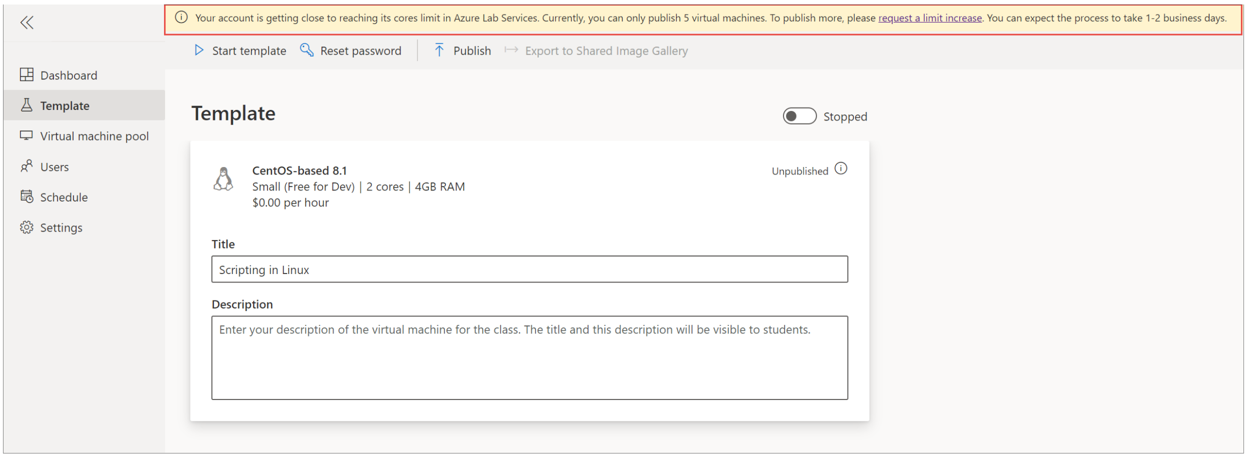
6/16/2021 • 2 minutes to read • Edit Online

Azure Lab Services has default capacity limits on Azure subscriptions to adhere to Azure Compute quota limitations and to mitigate fraud. All Azure subscriptions will have an initial capacity limit, which can vary based on subscription type, number of standard compute cores, and GPU cores available inside Azure Lab Services. It restricts how many virtual machines you can create inside your labs before you need to request for a limit increase.

If you are close to or have reached your subscription's virtual machine cores limit, you will see messages from Azure Lab Services when you try to perform actions that create additional virtual machines. For example:

- Create a lab
- Publish a lab
- Adjust lab capacity to add more virtual machines to an existing lab

These actions may also be disabled if you have already reached the cores limit.



Subscriptions with default limit of zero cores

Some rare subscription types that are more commonly used for fraud can have a default limit of 0 standard cores and 0 GPU cores. If you are using one of these subscription types, the admin who creates your lab account will need to request a limit increase before you can use Azure Lab Services.

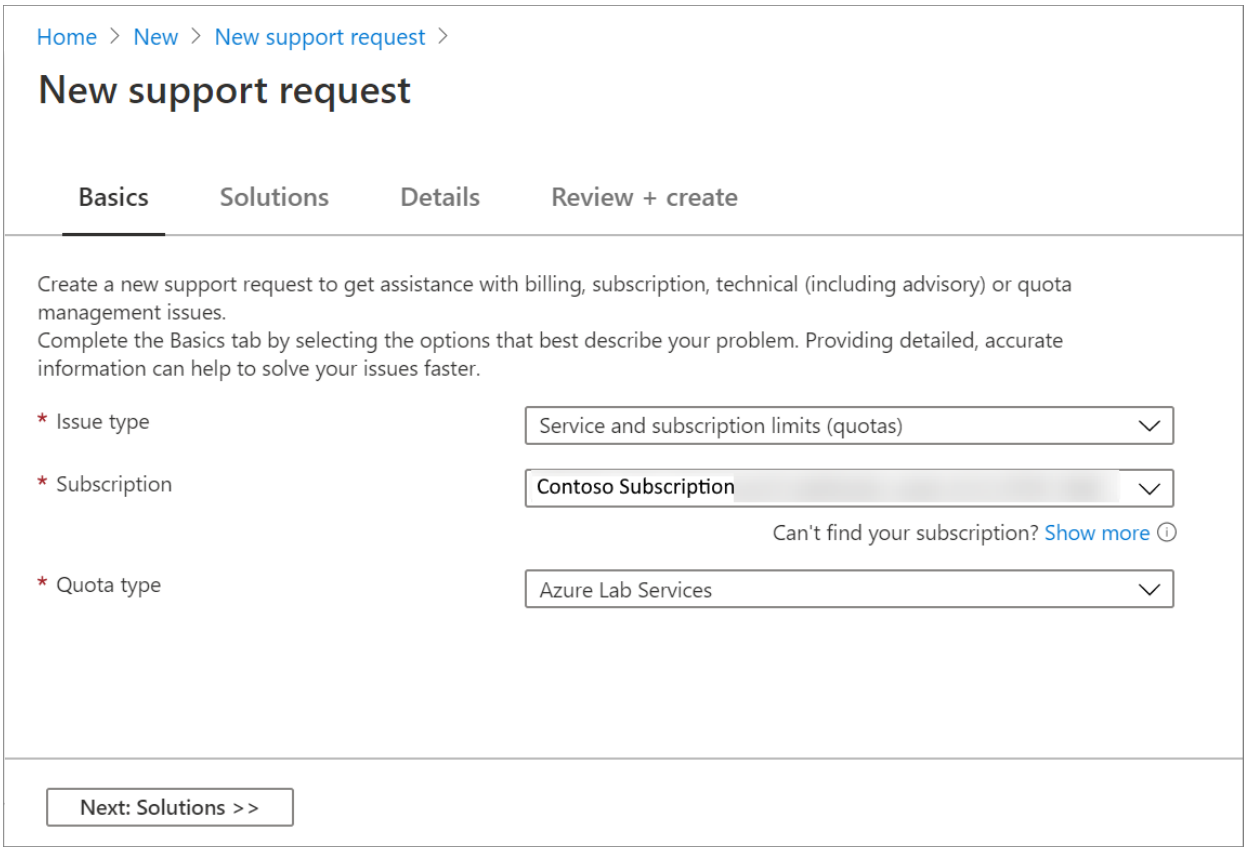
The admin can follow these steps to request a limit increase:

1. In your subscription, create a lab account.
2. On the **Overview** page of the lab account, click **Request limit increase** button at the top.
3. Follow the steps in the form to submit a support request to increase the limit.

Request a limit increase

If you reach the cores limit, you can request a limit increase to continue using Azure Lab Services. The request process is a checkpoint to ensure your subscription isn't involved in any cases of fraud or unintentional, sudden large-scale deployments.

The messages about the virtual machine cores limit in the Azure Lab Services portal includes a link to request a limit increase. The link opens a new browser tab where you can create a new support request. The issue type, subscription, and quota type information will be automatically filled out for you as shown in the following image:



Then, you will be prompted to provide more information about the limit increase. In the **Description** field, provide the following details:

- What you are trying to do (for example, creating a lab to teach a Computer Science class, run a hackathon, and so on.)
- Virtual machine size you are using for this lab
- Number of virtual machines you need

Once you submit the support request, we will review the request. If necessary, we will contact you to get additional details.

Next steps

See the following article:

- Administrator Guide - VM sizing.
- Frequently asked questions.

Cost management for Azure Lab Services

3/5/2021 • 7 minutes to read • Edit Online

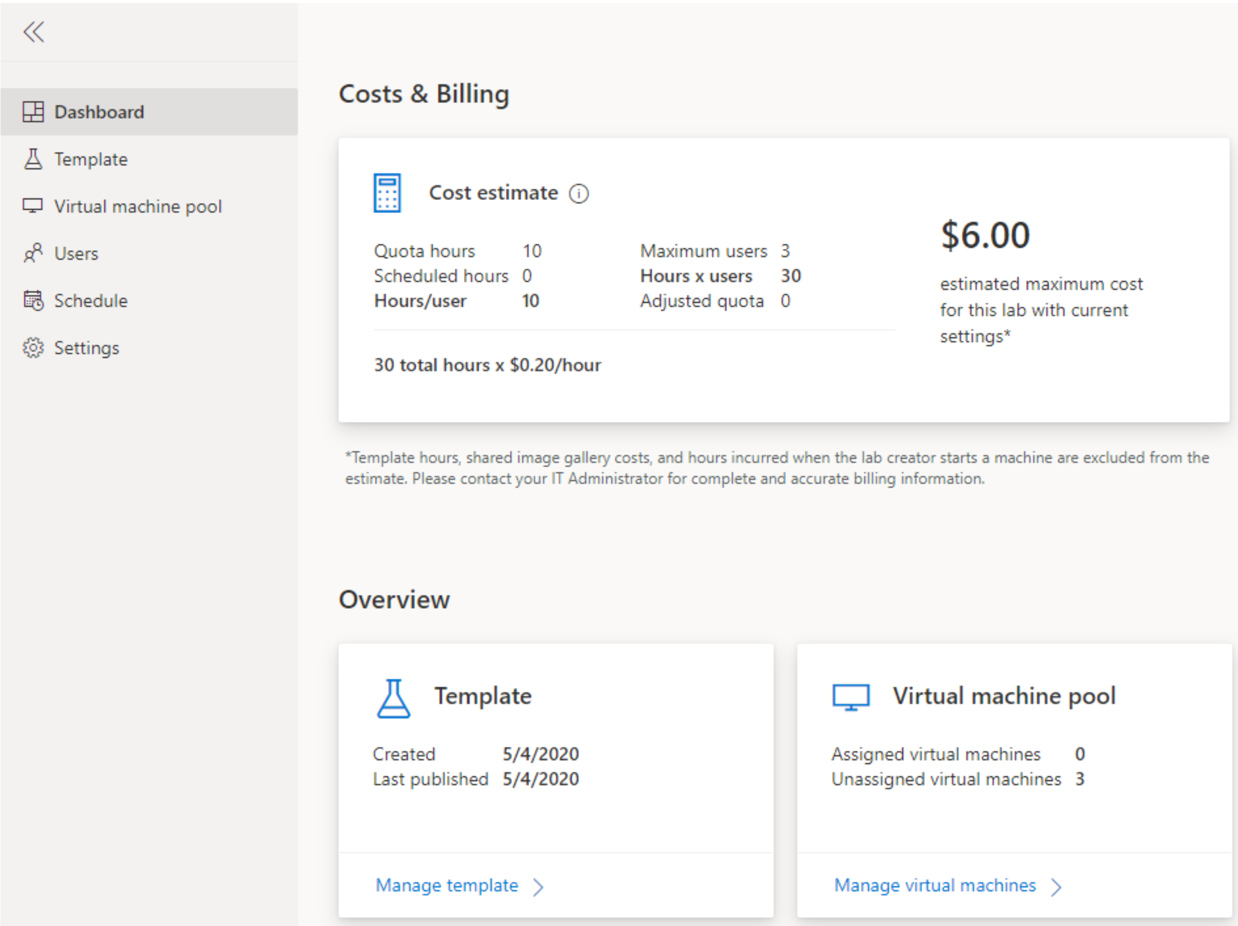
For Azure Lab Services, cost management can be broken down into two distinct areas: cost estimation and cost analysis. Cost estimation occurs when you're setting up the lab to make sure that the initial structure of the lab will fit within the expected budget. Cost analysis usually occurs at the end of the month to determine the necessary actions for the next month.

Estimate the lab costs

Each lab dashboard has a **Costs & Billing** section that lays out a rough estimate of what the lab will cost for the month. The cost estimate summarizes the hour usage with the maximum number of users by the estimated cost per hour. To get the most accurate estimate, set up the lab, including the schedule. The dashboard will reflect the estimated cost.

This estimate might not show all the possible costs. A few resources aren't included:

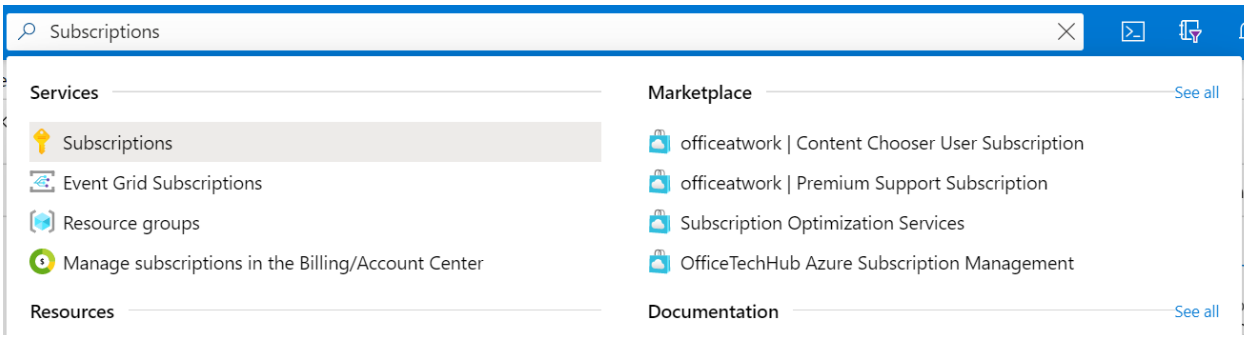
- The template preparation cost. It can vary significantly in the amount of time needed to create the template. The cost to run the template is the same as the overall lab cost per hour.
- Any shared image gallery costs, because a gallery can be shared among multiple labs.
- Hours incurred when the lab creator starts a virtual machine (VM).



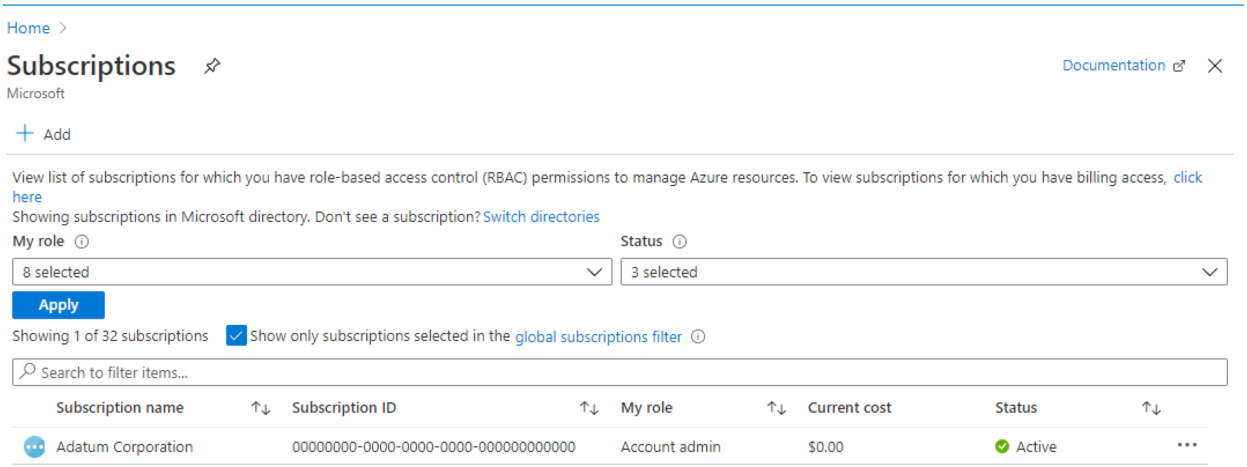
Analyze the previous month's usage

The cost analysis is for reviewing the previous month's usage to help you determine any adjustments for the lab.

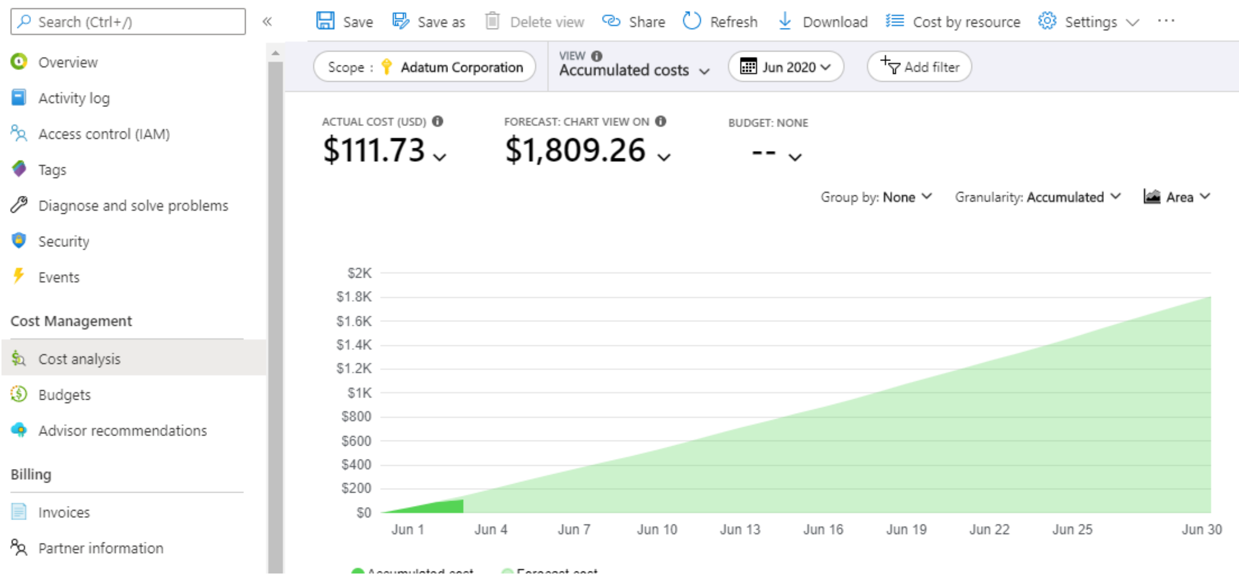
You can find the breakdown of past costs in the subscription cost analysis. In the Azure portal, you can enter **Subscriptions** in the search box and then select the **Subscriptions** option.



Select the specific subscription that you want to review.



Select **Cost Analysis** in the left pane under **Cost Management**.

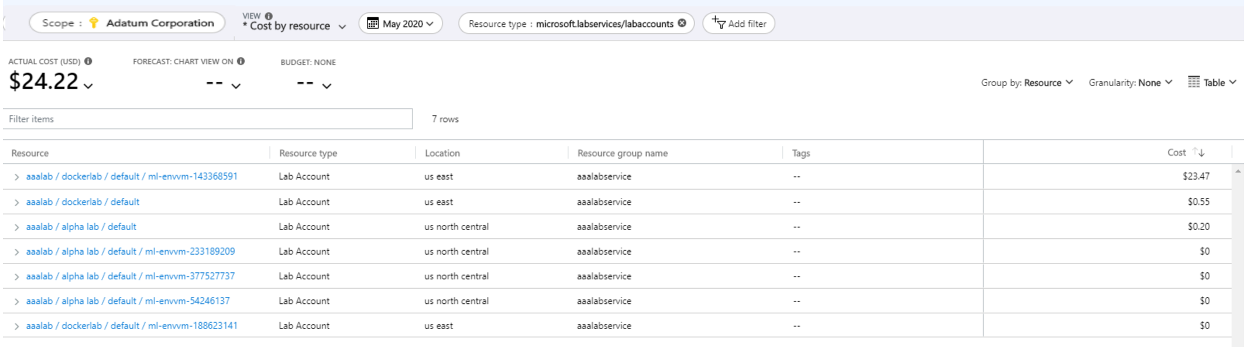


This dashboard allows in-depth cost analysis, including the ability to export to different file types on a schedule. For more information, see Cost Management + Billing overview.

You can filter by resource type. Using `microsoft.labservices/labaccounts` will show only the cost associated with Lab Services.

Understand the usage

The following screenshot is an example of a cost analysis.



By default, there are six columns: **Resource**, **Resource type**, **Location**, **Resource group name**, **Tags**, and **Cost**. The **Resource** column contains the information about the lab account, lab name, and VM. The rows that show the lab account, lab name, and default (second and third rows) are the cost for the lab. The used VMs have a cost that you can see for the rows that show the lab account, lab name, default, and VM name.

In this example, adding the first and second rows (both start with **aaalab / dockerlab**) will give you the total cost for the lab "dockerlab" in the "aaalab" lab account.

To get the overall cost for the image gallery, change the resource type to `Microsoft.Compute/Galleries`. A shared image gallery might not show up in the costs, depending on where the gallery is stored.

> **NOTE**
>
> A shared image gallery is connected to the lab account. That means multiple labs can use the same image.

Separate the costs

Some universities have used the lab account and the resource group as ways to separate the classes. Each class has its own lab account and resource group.

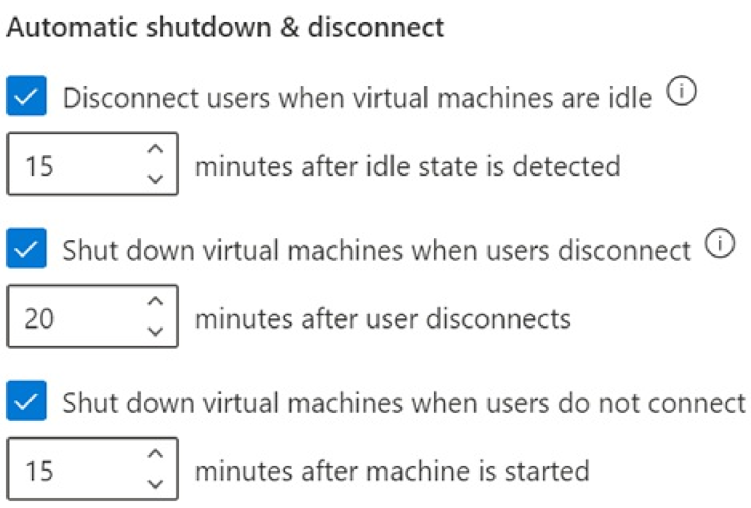
In the cost analysis pane, add a filter based on the resource group name with the appropriate resource group name for the class. Then, only the costs for that class will be visible. This allows a clearer delineation between the classes when you're viewing the costs. You can use the scheduled export feature of the cost analysis to download the costs of each class in separate files.

Manage costs

Depending on the type of class, there are ways to manage costs to reduce instances of VMs that are running without a student using them.

Automatic shutdown settings for cost control

Automatic shutdown features enable you to prevent wasted VM usage hours in the labs. The following settings catch most of the cases where users accidentally leave their virtual machines running:



You can configure these settings at both the lab account level and the lab level. If you enable them at the lab account level, they're applied to all labs within the lab account. For all new lab accounts, these settings are turned on by default.

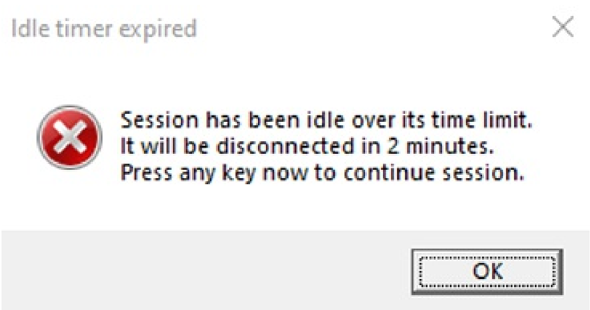
Automatically disconnect users from virtual machines that the OS deems idle

> **NOTE**
>
> This setting is available only for Windows virtual machines.

When the **Disconnect users when virtual machines are idle** setting is turned on, the user is disconnected from any machines in the lab when the Windows OS deems the session to be idle (including the template virtual machines). The Windows OS definition of idle uses two criteria:

- User absence: no keyboard or mouse input.
- Lack of resource consumption: All the processors and all the disks were idle for a certain percentage of time.

Users will see a message like this in the VM before they're disconnected:



The virtual machine is still running when the user is disconnected. If the user reconnects to the virtual machine by signing in, windows or files that were open or work that was unsaved before the disconnect will still be there. In this state, because the virtual machine is running, it still counts as active and accrues cost.

To automatically shut down idle Windows virtual machines that are disconnected, use the combination of **Disconnect users when virtual machines are idle** and **Shut down virtual machines when users disconnect** settings.

For example, if you configure the settings as follows:

- **Disconnect users when virtual machines are idle**: 15 minutes after the idle state is detected.
- **Shut down virtual machines when users disconnect**: 5 minutes after the user disconnects.

The Windows virtual machines will automatically shut down 20 minutes after the user stops using them.

Automatically shut down virtual machines when users disconnect

The **Shut down virtual machines when users disconnect** setting supports both Windows and Linux virtual machines. When this setting is on, automatic shutdown will occur when:

- For Windows, a Remote Desktop (RDP) connection is disconnected.
- For Linux, an SSH connection is disconnected.

> **NOTE**
>
> Only specific distributions and versions of Linux are supported.

You can specify how long the virtual machines should wait for the user to reconnect before automatically shutting down.

Automatically shut down virtual machines that are started but users don't connect

In a lab, a user might start a virtual machine but never connect to it. For example:

- A schedule in the lab starts all virtual machines for a class session, but some students don't show up and don't connect to their machines.
- A user starts a virtual machine but forgets to connect.

The **Shut down virtual machines when users do not connect** setting will catch these cases and automatically shut down the virtual machines.

For information on how to configure and enable automatic shutdown of VMs on disconnect, see these articles:

- Configure automatic shutdown of VMs for a lab account
- Configure automatic shutdown of VMs for a lab

Scheduled time vs. quota time

Understanding scheduled time and quota time will help you to configure a lab to better fit the needs of the professor and the students.

Scheduled time is a set time where all the student VMs have been started and are available for connection. Scheduled time is commonly used when all the students have their own VMs and are following the professor's directions at a set time during the day (like class hours). The downside is that all the student VMs are started and are accruing costs, even if a student doesn't log in to a VM.

Quota time is time allocated to each student for use at their discretion and is often used for independent studying. The VMs aren't started until the student starts the VM.

A lab can use either quota time or scheduled time, or a combination of both. If a class doesn't need scheduled time, then use only quota time for the most effective use of the VMs.

Scheduled event: stop only

In the schedule, you can add a stop-only event type that will stop all machines at a specific time. Some lab owners have set a stop-only event for every day at midnight to reduce the cost and quota usage when a student forgets to shut down the VM they're using. The downside to this type of event is that all VMs will be shut down, even if a student is using a VM.

Other costs related to labs

Some costs aren't rolled into Lab Services but can be tied to a lab service. You can connect a shared image gallery to a lab, but it won't show under the Lab Services costs and does have costs. To help keep overall costs down, you should remove any unused images from the gallery because the images have an inherent storage cost.

Labs can have connections to other Azure resources through a virtual network. When a lab is removed, you should remove the virtual network and the other resources.

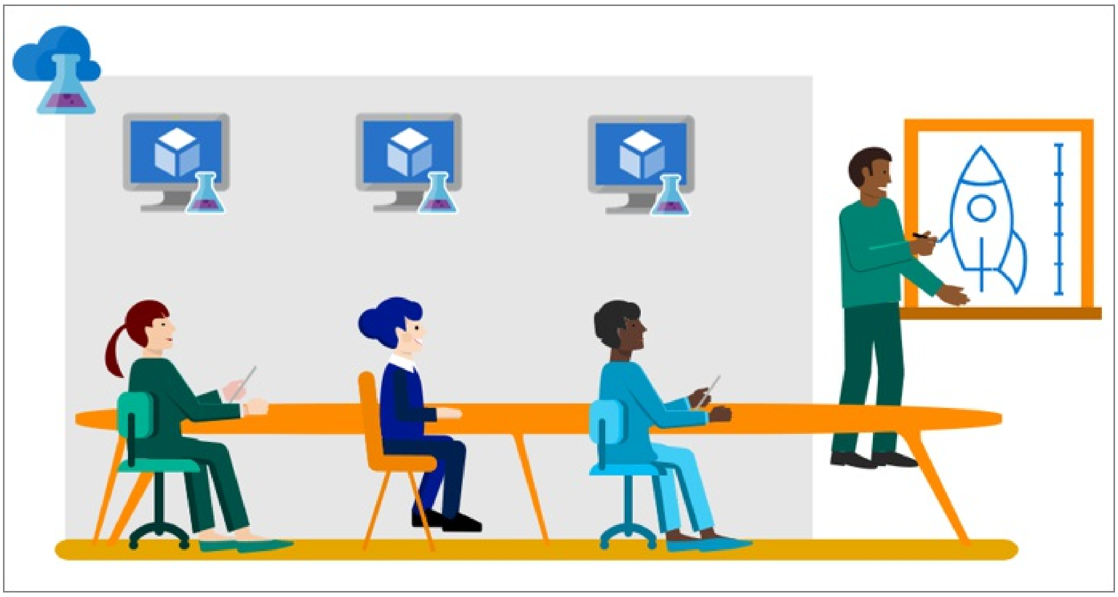
Conclusion

Hopefully, the information in this article has given you a better understanding of the tools that can help you reduce usage costs.

Use labs for trainings

3/5/2021 • 3 minutes to read • Edit Online

Azure Labs Services allows educators (teachers, professors, trainers, or teaching assistants, etc.) to quickly and easily create an online lab to provision pre-configured learning environments for the trainees. Each trainee would be able use identical and isolated environments for the training. Policies can be applied to ensure that the training environments are available to each trainee only when they need them and contain enough resources - such as virtual machines - required for the training.



labs meets the following requirements that are required to conduct training in any virtual environment:

- Trainees can quickly provision their training environments
- Every training machine should be identical
- Trainees cannot see VMs created by other trainees
- Control cost by ensuring that trainees cannot get more VMs than they need for the training and also shutdown VMs when they are not using them
- Easily share the training lab with each trainee
- Reuse the training lab again and again

In this article, you learn about various Azure Lab Services features that can be used to meet the previously described training requirements and detailed steps that you can follow to set up a lab for training.

Create the lab account as a lab account administrator

The first step in using Azure Lab Services is to create a lab account in the Azure portal. After a lab account administrator creates the lab account, the admin adds users who want to create labs to the `Lab Creator` role. The educators create labs with virtual machines for students to do exercises for the course they are teaching. For details, see Create and manage lab account.

Create and manage labs

An educator, who is a member of the Lab Creator role in a lab account, can create one or more labs in the lab account. You create and configure a template VM with all the required software for doing exercises in your course. You pick a ready-made image from the available images for creating a classroom lab and then customize it by installing the software required for the lab. For details, see Create and manage labs.

Configure usage settings and policies

The lab creator can add or remove users to the lab, get registration link to send to lab users, set up policies such as setting individual quotas per user, update the number of VMs available in the lab, and more. For details, see Configure usage settings and policies.

Create and manage schedules

Schedules allow you to configure a classroom lab such that VMs in the lab automatically start and shut down at a specified time. You can define a one-time schedule or a recurring schedule. For details, see Create and manage schedules for labs.

Set up and publish a template VM

A template in a lab is a base virtual machine image from which all users' virtual machines are created. Set up the template VM so that it is configured with exactly what you want to provide to the training attendees. You can provide a name and description of the template that the lab users see. Then, you publish the template to make instances of the template VM available to your lab users. When you publish a template, Azure Lab Services creates VMs in the lab by using the template. The number of VMs created in this process is same as the maximum number of users allowed into the lab, which you can set in the usage policy of the lab. All virtual machines have the same configuration as the template. For details, see Set up and publish template virtual machines.

Use VMs in the classroom lab

A student or training attendee registers to the lab, and connects to the VM to do exercises for the course. For details, see How to access a classroom lab.

Next steps

Start with creating a lab account in labs by following instructions in the article: Tutorial: Setup a lab account with Azure Lab Services.

Use Azure Lab Services for your next hackathon

3/8/2021 • 6 minutes to read • Edit Online

Azure Lab Services is designed to be lightweight and easy to use so that you can quickly spin up a new lab of virtual machines (VMs) for your hackathon. Use the following checklist to ensure that your hackathon goes as smoothly as possible. This checklist should be completed by your IT department or faculty who are responsible for creating and managing your hackathon lab.

To use Lab Services for your hackathon, ensure that both lab account and your lab are created at least a few days before the start of your hackathon. Also, follow the guidance below:

Guidance

- **Create the lab in a region or location that's closest to participants.**

  To reduce latency, create your lab in a region that's closest to your hackathon participants. If your participants are located all over the world, you need to use your best judgment to create a lab that is centrally located. Or, split the hackathon to use multiple labs based on the locations where your participants are located.

- **Choose a compute size best suited for usage needs.**

  Generally, the larger the compute size, the faster the virtual machine will perform. However, to limit costs, you'll need to select the appropriate compute size based on your participants' needs. See VM sizing information in the administrator guide for details on the available compute sizes.

- **Configure RDP\SSH for remote desktop connection to Linux VMs.**

  If your hackathon uses Linux VMs, ensure that remote desktop is enabled so that your participants can use either RDP (remote desktop protocol) or SSH (secure shell) to connect to their VMs. This step is only required for Linux VMs and must be enabled when creating the lab. Also, for RDP, you may need to install and configure the RDP server and GUI packages on the template VM before publishing. For more information, see the how-to guide on enabling remote desktop for Linux.

- **Install and stop Windows updates.**

  If you're using a Windows image, we recommend that you install the latest Windows updates on the lab's template VM before you publish it to create labs' VMs. It's for security purposes and to prevent participants from being disrupted during the hackathon to install updates, which can also cause their VMs to restart. You might also consider turning off Windows updates to prevent any future interruptions. See the how-to guide on installing and configuring Windows updates.

- **Decide how students will back up their work.**

  Students are each assigned a virtual machine for the lifetime of the hackathon. They can save their work directly to the machine, but it's recommended that students back up their work so that they have access to it after the hackathon is over. For example, they should save to an external location, such as OneDrive, GitHub, and so on. To use OneDrive, you may choose to configure it automatically for students on their lab virtual machines. See the how-to guide to install and configure OneDrive.

- **Set VM capacity according to number of participants.**

  Ensure that your lab's virtual machine capacity is set based on the number of participants you expect at your hackathon. When you publish the template virtual machine, it can take several hours to create all of the machines in the lab. That's why we recommend that you do it well in advance to the start of the hackathon. For more information, see the how-to guide on updating lab capacity.

- **Decide whether to restrict lab access**.

  When adding users to the lab, there is a restrict access option that's enabled by default. This feature requires you to add all of your hackathon participants' emails to the list before they can register and access the lab using the registration link. If you have a hackathon where you don't know who the participants will be before the event, you can choose to disable the restrict access option, which allows anyone to register to the lab using the registration link. For more information, see the how-to guide on adding users.

- **Verify schedule, quota, and autoshutdown settings**.

  Lab Services provides several cost controls to limit usage of VMs. However, if these settings are misconfigured, they can cause your lab's virtual machines to unexpectedly shut down. To ensure that these settings are configured appropriately for your hackathon, verify the following settings:

  Schedule: A schedule allows you to automatically control when your labs' machines are started and shut down. By default, no schedule is configured when you create a new lab. However, you should ensure that your lab's schedule is set according to what makes sense for your hackathon. As an example, if your hackathon starts on Saturday at 8:00 AM and ends on Sunday at 5:00 PM – you could create a schedule that automatically starts the machine at 7:30 AM on Saturday (about 30 minutes before the start of the hackathon) and shuts it down at 5:00 PM on Sunday. Instead, you may also decide not to use a schedule at all.

  Quota: The quota controls the number of hours that participants will have access to a virtual machine outside of the scheduled hours. If the quota is reached while a participant is using it, the machine is automatically shut down and the participant won't be able to restart it unless the quota is increased. By default, when you create a lab, the quota is set to 10 hours. Again, you should be sure to set the quota so that it allows enough time for the hackathon, which is especially important if you haven't created a schedule.

  Autoshutdown: When enabled, the autoshutdown setting causes Windows virtual machines to automatically shut down after a certain period of time once a student has disconnected from their RDP session. By default, this setting is disabled.

- **Configure firewall settings to allow connections to lab VMs**.

  Ensure that your school's or organization's firewall settings allow connecting to lab VMs using RDP\SSH. For more information, see the how-to guide on configuring your network's firewall settings.

- **Install RDP\SSH client on participants' tablets, Macs, PCs, and so on**.

  Hackathon participants must have an RDP and/or SSH client installed on their tablets or laptops that they'll use to connect to lab VMs. You may choose from different RDP or SSH clients, such as:

  - Microsoft's **Remote Desktop Connection** app for RDP connections. The Remote Desktop Connection app is supported on different kinds of platforms, including Chromebooks and Mac.
  - Putty for using SSH to connect to a Linux VM.

- **Verify lab virtual machines**.

  Once you've published lab VMs, you should verify they're configured properly. You only need to do this verification for one of the participant's lab virtual machines:

  1. Connect using RDP and\or SSH.
  2. Open each additional application and tool that you installed to customize the base virtual machine image.

3. Walk through a few basic scenarios that are representative of the activities that participants will do to ensure VM performance is adequate based on the selected compute size.

On the day of hackathon

This section outlines the steps to complete the day of your hackathon.

1. **Start lab VMs**.

   Depending on your OS, your lab machine may take up to 30 minutes to start. As a result, it's important to start machines before the hackathon starts so that your participants don't have to wait. If you're using a schedule, ensure that the VMs are automatically started at least 30 minutes earlier as well.

2. **Invite students to register and access their lab virtual machine**.

   Provide your participants with the following information so that participants can access their lab VMs.

   - The lab's registration link.
   - Credentials that should be used to connect to the machine. This step applies only if your lab uses a Windows-based image and you configured all VMs to use the same password.
   - Instructions on how participants SSH and\or RDP to their machines.

     For more information, See how-to guide on sending invitations to users and connecting to Linux VMs.

Next steps

Start with creating a lab account in labs by following instructions in the article: Tutorial: Setup a lab account with Azure Lab Services.

Class types overview - Azure Lab Services

9/7/2021 • 8 minutes to read • Edit Online

Azure Lab Services enables you to quickly set up classroom lab environments in the cloud. Articles in this section provide guidance on how to set up several types of labs using Azure Lab Services.

Adobe Creative Cloud

The Adobe Creative Cloud collection of applications are commonly used in digital arts and media classes.

For detailed information on how to set up this type of lab, see Setup a lab for Adobe Creative Cloud.

ArcGIS

ArcGIS is a type of geographic information system (GIS). You can set up a lab that uses ArcGIS Desktop's various applications, such as ArcMap to make, edit, and analyze 2D maps.

For detailed information on how to set up this type of lab, see Setup a lab for ArcMap\ArcGIS Desktop.

Autodesk

Autodesk offers software solutions in architecture, engineering, construction, design, manufacturing, and more. These solutions are commonly used in engineering classes and in the Project Lead the Way curriculum.

For detailed information on how to set up this type of lab, see Autodesk.

Big data analytics

You can set up a GPU lab to teach a big data analytics class. With this type of class, students learn how to handle large volumes of data, and apply machine and statistical learning algorithms to derive data insights. A key objective for students is to learn to use data analytics tools, such as Apache Hadoop's open-source software package that provides tools for storing, managing, and processing big data.

For detailed information on how to set up this type of lab, see Set up a lab for big data analytics using Docker deployment of HortonWorks Data Platform.

Database management

Databases concepts are one of the introductory courses taught in most of the Computer Science departments in college. You can set up a lab for a basic databases management class in Azure Lab Services. For example, you can set up a virtual machine template in a lab with a MySQL Database Server or a SQL Server 2019 server.

For detailed information on how to set up this type of lab, see Set up a lab to teach database management for relational databases.

Deep learning in natural language processing

You can set up a lab focused on deep learning in natural language processing (NLP) using Azure Lab Services. Natural language processing (NLP) is a form of artificial intelligence (AI) that enables computers with translation, speech recognition, and other language understanding capabilities. Students taking an NLP class get a Linux virtual machine (VM) to learn how to apply neural network algorithms to develop deep learning models that are used for analyzing written human language.

For detailed information on how to set up this type of lab, see Set up a lab focused on deep learning in natural language processing using Azure Lab Services.

Ethical hacking with Hyper-V

You can set up a lab for a class that focuses on forensics side of ethical hacking. Penetration testing, a practice used by the ethical hacking community, occurs when someone attempts to gain access to the system or network to demonstrate vulnerabilities that a malicious attacker may exploit.

In an ethical hacking class, students can learn modern techniques for defending against vulnerabilities. Each student gets a Windows Server host virtual machine that has two nested virtual machines – one virtual machine with Metasploitable3 image and another machine with Kali Linux image. The Metasploitable virtual machine is used for exploiting purposes. The Kali Linux virtual machine provides access to the tools needed to execute forensic tasks.

For detailed information on how to set up this type of lab, see Set up a lab to teach ethical hacking class.

Ethical hacking with VirtualBox

You can set up a lab for a class that focuses on forensics side of ethical hacking. Penetration testing, a practice used by the ethical hacking community, occurs when someone attempts to gain access to the system or network to demonstrate vulnerabilities that a malicious attacker may exploit.

In an ethical hacking class, students can learn modern techniques for defending against vulnerabilities. Each student gets a Windows Server host virtual machine that has two nested virtual machines – one virtual machine with SEED Labs image and another machine with Kali Linux image. The SEED virtual machine is used for exploiting purposes. The Kali Linux virtual machine provides access to the tools needed to execute forensic tasks.

For detailed information on how to set up this type of lab, see Set up a lab to teach ethical hacking class.

MATLAB

MATLAB, which stands for Matrix laboratory, is programming platform from MathWorks. It combines computational power and visualization making it popular tool in the fields of math, engineering, physics, and chemistry.

For detailed information on how to set up this type of lab, see Setup a lab to teach MATLAB.

Networking with GNS3

You can set up a lab for a class that focuses on allowing students to emulate, configure, test, and troubleshoot virtual and real networks using GNS3 software.

For detailed information on how to set up this type of lab, see Setup a lab to teach a networking class.

Project Lead the Way (PLTW)

Project Lead the Way (PLTW) is a nonprofit organization that provides PreK-12 curriculum across the United States in computer science, engineering, and biomedical science. In each PLTW class, students use various software applications as part of their hands-on learning experience.

For detailed information on how to set up these types of labs, see Set up labs for Project Lead the Way classes.

Python and Jupyter notebooks

You can set up a template machine in Azure Lab Services with the tools needed to teach students how to use

Jupyter Notebooks. Jupyter Notebooks is an open-source project that lets you easily combine rich text and executable Python source code on a single canvas called a notebook. Running a notebook results in a linear record of inputs and outputs. Those outputs can include text, tables of information, scatter plots, and more.

For detailed information on how to set up this type of lab, see Set up a lab to teach data science with Python and Jupyter Notebooks.

React

React is a popular JavaScript library for building user interfaces (UI). React is a declarative way to create reusable components for your website. There are many popular libraries for JavaScript-based front-end development. We'll use a few of these libraries while creating our lab. Redux is a library that provides predictable state container for JavaScript apps and is often used in compliment with React. JSX is a library syntax extension to JavaScript often used with React to describe what the UI should look like. NodeJS is a convenient way to run a webserver for your React application.

For detailed information on how to set up this type of lab on Linux using Visual Studio Code for your development environment, see Set up lab for React on Windows. For detailed information on how to set up this type of lab on Windows using Visual Studio 2019 for your development environment, see Set up lab for React on Windows.

RStudio

R is an open-source language used for statistical computing and graphics. It's used in the statistical analysis of genetics, natural language processing, analyzing financial data, and more. R provides an interactive command line experience. RStudio is an interactive development environment (IDE) available for the R language. The free version provides code editing tools, an integrated debugging experience, and package development tools. This class type will focus on solely RStudio and R as a building block for a class that requires the use of statistical computing.

For detailed information on how to set up this type of lab, see Set up a lab to teach R on Linux or Set up a lab to teach R on Windows.

Shell scripting on Linux

You can set up a lab to teach shell scripting on Linux. Scripting is a useful part of system administration that allows administrators to avoid repetitive tasks. In this sample scenario, the class covers traditional bash scripts and enhanced scripts. Enhanced scripts are scripts that combine bash commands and Ruby. This approach allows Ruby to pass data around and bash commands to interact with the shell.

Students taking these scripting classes get a Linux virtual machine to learn the basics of Linux, and also get familiar with the bash shell scripting. The Linux virtual machine comes with remote desktop access enabled and with Gedit and Visual Studio Code text editors installed.

For detailed information on how to set up this type of lab, see Shell scripting on Linux.

SolidWorks computer-aided design (CAD)

You can set up a GPU lab that gives engineering students access to SolidWorks. SolidWorks provides a 3D CAD environment for modeling solid objects. With SolidWorks, engineers can easily create, visualize, simulate and document their designs.

For detailed information on how to set up this type of lab, see Set up a lab for engineering classes using SolidWorks.

SQL database and management

Structured Query Language (SQL) is the standard language for relational database management including adding, accessing, and managing content in a database. You can set up a lab to teach database concepts using both MySQL Database server and SQL Server 2019 server.

For detailed information on how to set up this type of lab, see Set up a lab to teach database management for relational databases.

Next steps

See the following articles:

- Set up a lab focused on deep learning in natural language processing using Azure Lab Services
- Set up a lab to teach a networking class
- Set up a lab to teach ethical hacking class with Hyper-V
- Set up a lab to teach ethical hacking class with VirtualBox

Set up a lab for Adobe Creative Cloud

6/10/2021 • 5 minutes to read • Edit Online

Adobe Creative Cloud is a collection of desktop applications and web services used for photography, design, video, web, user experience (UX), and more. Universities and K-12 schools use Creative Cloud in digital arts and media classes. Some of Creative Cloud's media processes may require more computational and visualization (GPU) power than a typical tablet, laptop, or workstation support. With Azure Lab Services, you have flexibility to choose from various virtual machine (VM) sizes, including GPU sizes.

In this article, we'll show how to set up a class that uses Creative Cloud.

Licensing

To use Creative Cloud on a lab VM, you must use Named User Licensing, which is the only type of licensing that supports deployment on a virtual machine. Each lab VM has internet access so that your students can activate Creative Cloud apps by signing into the software. Once a student signs in, their authentication token is cached in the user profile so that they don't have to sign in again on their VM. Read Adobe's article on licensing for more details.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to set up a lab account. You can also use an existing lab account.

Lab account settings

Enable the settings described in the table below for the lab account. For more information about how to enable marketplace images, see the article on how to specify Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the Windows 10 image for use within your lab account.

Lab settings

The size of VM that you need to use for your lab depends on the types of projects that your students will create. Most Creative Cloud apps support GPU-based acceleration and require a GPU for features to work properly. To ensure that you select the appropriate VM size, we recommend that you test the projects that your students will create to ensure adequate performance. The below table shows the recommended VM size to use with Creative Cloud.

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual Machine Size	**Small GPU (Visualization)**. This VM is best suited for remote visualization, streaming, gaming, encoding using frameworks such as OpenGL and DirectX.
Virtual Machine Image	Windows 10

NOTE

The **Small GPU (Visualization)** virtual machine size is configured to enable a high-performing graphics experience and meets Adobe's system requirements for each application. Make sure to choose Small GPU (Visualization) not Small GPU (Compute). For more information about this virtual machine size, see the article on how to set up a lab with GPUs.

Template virtual machine configuration

Creative Cloud deployment package

Installing Creative Cloud requires the use of a deployment package. Typically, the deployment package is created by your IT department using Adobe's Admin Console. When IT creates the deployment package, they also have the option to enable self-service. There are a few ways to enable self-service for the deployment package:

- Create a self-service package.
- Create a managed package with self-service elevated privileges turned on.

With self-service enabled, you don't install the entire Creative Cloud collection of apps. Instead, students can install apps themselves using the Creative Cloud desktop app. Here are some key benefits with this approach:

- The entire Creative Cloud install is about 25 GB. If students install only the apps they need on-demand, this helps optimize disk space. Lab VMs have a disk size of 128 GB.
- You can choose to install a subset of the apps on the template VM before publishing. This way the student VMs will have some apps installed by default and students can add more apps on their own as needed.
- You can avoid republishing the template VM because students can install additional apps on their VM at any point during the lifetime of the lab. Otherwise, either IT or the teacher would need to install additional apps on the template VM and republish. Republishing causes the students' VMs to be reset and any work that isn't saved externally is lost.

If you use a managed deployment package with self-service disabled, students won't have the ability to install their own apps. In this case, IT must specify the Creative Cloud apps that will be installed.

Read Adobe's steps to create a package for more information.

Install Creative Cloud

After the template machine is created, follow the steps below to set up your lab's template virtual machine (VM) with Creative Cloud.

1. Start the template VM and connect using RDP.
2. To install Creative Cloud, download the deployment package given to you by IT or directly from Adobe's Admin Console.
3. Run the deployment package file. Depending on whether self-service is enabled or disabled, this will install Creative Cloud desktop app and\or the specified Creative Cloud apps. Read Adobe's deployment steps for more information.
4. Once the template VM is set up, publish the template VM's image which is used to create all of the students' VMs in the lab.

Storage

As mentioned earlier, Azure Lab VMs have a disk size of 128 GB. If your students need additional storage for saving large media assets or they need to access shared media assets, you should consider using external file storage. For more information, read the following articles:

- Using external file storage in Lab Services
- Install and configure OneDrive

Save template VM image

Consider saving your template VM for future use. To save the template VM, see save an image to Shared Image Gallery.

- When self-service is *enabled*, the template VM's image will have Creative Cloud desktop installed. Teachers can then reuse this image to create labs and to choose which Creative Cloud apps to install. This helps reduce IT overhead since teachers can independently set up labs and have full control over installing the Creative Cloud apps required for their classes.
- When self-service is *disabled*, the template VM's image will already have the specified Creative Cloud apps installed. Teachers can reuse this image to create labs; however, they won't be able to install additional Creative Cloud apps.

Cost

In this section, we'll look at a possible cost estimate for this class. We'll use a class of 25 students with 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The virtual machine size we chose was **Small GPU (Visualization)**, which is 160 lab units.

25 students * (20 scheduled hours + 10 quota hours) * 160 Lab Units * 0.01 USD per hour = 1200.00 USD

> **IMPORTANT**
>
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Next steps

Next steps are common to setting up any lab.

- Create and manage a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up a lab for ArcMap\ArcGIS Desktop

3/5/2021 • 4 minutes to read • Edit Online

ArcGIS is a type of geographic information system (GIS). ArcGIS is used to make\analyze maps and work with geographic data that is provided by the Environmental Systems Research Institute (ESRI). Although ArcGIS Desktop includes several applications, this article shows how to set up labs for using ArcMap. ArcMap is used to make, edit, and analyze 2D maps.

Lab configuration

To begin setting up a lab for using ArcMap, you need an Azure subscription and lab account. If you don't have an Azure subscription, create a free account before you begin.

After you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see Set up a lab account. You can also use an existing lab account.

Lab account settings

Enable your lab account settings as described in the following table. For more information about how to enable Azure Marketplace images, see Specify the Azure Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the Windows 10 Pro or Windows 10 Pro N image for use within your lab account.

Licensing server

One type of licensing that ArcGIS Desktop offers is concurrent use licenses. This requires that you install ArcGIS License Manager on your license server. The License Manager keeps track of the number of copies of software that can be run at the same time. For more information on how to set up the License Manager on your server, see the License Manager Guide.

The license server is typically located in either your on-premises network or hosted on an Azure virtual machine within an Azure virtual network. After your license server is set up, you'll need to peer the virtual network with your lab account. You need to do the network peering before you create the lab so that your lab VMs can access the license server and vice versa.

For more information, see Set up a license server as a shared resource.

Lab settings

The size of the virtual machine (VM) that we recommend using for ArcGIS Desktop depends on the applications, extensions, and the specific versions that students will use. The VM size also depends on the workloads that students are expected to perform. Refer to ArcGIS Desktop system requirements to help with identifying the VM size. Once you've identified the potential VM size, we recommend that you test your students' workloads to ensure adequate performance.

In this article, we recommend using the **Medium** VM size for version 10.7.1 of ArcMap, assuming that no other ArcGIS Desktop extensions are used. However, depending on the needs of your class, you may require a **Large** or even a **Small\Medium GPU (Visualization)** VM size. For example, the Spatial Analyst extension that is included with ArcGIS Desktop supports a GPU for enhanced performance, but doesn't require using a GPU.

LAB SETTING	VALUE AND DESCRIPTION
Virtual Machine Size	**Medium**. Best suited for relational databases, in-memory caching, and analytics.

Template machine

The steps in this section show how to set up the template VM:

1. Start the template VM and connect to the machine using RDP.
2. Download and install the ArcGIS Desktop components using instructions from by ESRI. These steps include assigning the license manager for concurrent use licensing:
   - Introduction to installing and configuring ArcGIS Desktop
3. Set up external backup storage for students. Students can save files directly to their assigned VM since all changes that they make are saved across sessions. However, we recommend that students back up their work to storage that is external from their VM for a few reasons:
   - To enable students to access their work after the class and lab ends.
   - In case the student gets their VM into a bad state and their image needs to be reset.

   With ArcGIS, each student should back up the following files at the end of each work session:
   - mxd file, which stores the layout information for a project.
   - File geodatabases, which store all data produced by ArcGIS.
   - Any other data that the student may be using such as raster files, shapefiles, GeoTIFF, etc.

   We recommend using OneDrive for backup storage. To set up OneDrive on the template VM, follow the steps in the article Install and configure OneDrive.
4. Finally, publish the template VM to create the students' VM.

Auto-shutdown and disconnect settings

A lab's auto-shutdown and disconnect settings help make sure that a student's VM is shut down when it's not being used. These settings should be set according to the types of workloads that your students will perform so that their VM doesn't shut down in the middle of their work. For example, the **Disconnect users when virtual machines are idle** setting disconnects the student from their RDP session after no mouse or keyboard inputs have been detected for a specified amount of time. This setting must allow sufficient time for workloads where the student isn't actively using the mouse or keyboard, such as to run long queries or wait for rendering.

For ArcGIS, we recommend the following values for these settings:

- Disconnect users when virtual machines are idle
  - 30 minutes after idle state is detected
- Shut down virtual machines when users disconnect
  - 15 minutes after user disconnects

Cost

Let's cover a possible cost estimate for this class. This estimate doesn't include the cost of running the license server. We'll use a class of 25 students. There are 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The virtual machine size we chose was **Medium**, which is 42 lab units.

25 students * (20 scheduled hours + 10 quota hours) * 42 Lab Units * 0.01 USD per hour = 315.00 USD

> **IMPORTANT**
>
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Next steps

Next steps are common to setting up any lab.

- Create and manage a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up labs for Autodesk

4/29/2021 • 3 minutes to read • Edit Online

This article describes how to set up Autodesk Inventor and Autodesk Revit software for engineering classes:

- Inventor computer-aided design (CAD) and computer-aided manufacturing (CAM) provide 3D modeling and are used in engineering design.
- Revit is used in architecture design for 3D building information modeling (BIM).

Autodesk is commonly used in both universities and K-12 schools. For example, in K-12, AutoDesk is included in the Project Lead the Way (PLTW) curriculum.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable your lab account settings as described in the following table. For more information about how to enable Azure Marketplace images, see Specify the Azure Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the Windows 10 image for use within your lab account.

Lab settings

The size of the virtual machine (VM) that we recommend depends on the types of workloads that your students need to do. We recommend using the Small GPU (Visualization) size.

LAB SETTING	VALUE AND DESCRIPTION
Virtual Machine Size	**Small GPU (Visualization)** Best suited for remote visualization, streaming, gaming, and encoding with frameworks such as OpenGL and DirectX.

> **NOTE**
> The **Small GPU (Visualization)** virtual machine size is configured to enable a high-performing graphics experience. For more information about this virtual machine size, see the article on how to set up a lab with GPUs.

License server

You will need to access a license server if you plan to use the Autodesk network licensing model. Read Autodesk's article on Network License Administration for more information.

To use network licensing with Autodesk software, AutoDesk provides detailed steps to install Autodesk Network License Manager on your license server. This license server is ordinarily located in either your on-premises network or hosted on an Azure virtual machine (VM) within in Azure virtual network.

After your license server is set up, you'll need to peer the virtual network with your lab account. You need to do the network peering *before* you create the lab so that your lab VMs can access the license server and vice versa.

Autodesk-generated license files embed the MAC address of the license server. If you decide to host your license server by using an Azure VM, it's important to make sure that your license server's MAC address doesn't change. If the MAC address changes, you'll need to regenerate your licensing files. To prevent your MAC address from changing, do the following:

- Set a static private IP and MAC address for the Azure VM that hosts your license server.
- Be sure to set up both your lab account and the license server's virtual network in a region or location that has sufficient VM capacity so that you don't have to move these resources to a new region or location later.

For more information, see Set up a license server as a shared resource.

> **WARNING**
> Don't forget to peer the virtual network for the lab account to the virtual network for the license server **before** creating the lab.

Template machine

The steps in this section show how to set up the template VM:

1. Start the template VM and connect to the machine.
2. Download and install Inventor and Revit using instructions from AutoDesk. When prompted, specify the computer name of your license server.
3. Finally, publish the template VM to create the students' VMs.

Cost

Let's cover an example cost estimate for this class. This estimate doesn't include the cost of running a license server. Suppose you have a class of 25 students, each of whom has 20 hours of scheduled class time. Each student also has an additional 10 quota hours for homework or assignments outside of scheduled class time. The virtual machine size we chose was **Small GPU (Visualization)**, which is 160 lab units.

- 25 students × (20 scheduled hours + 10 quota hours) × 160 Lab Units × USD0.01 per hour = USD1200.00

> **IMPORTANT**
> The cost estimate is for example purposes only. For current pricing information, see Azure Lab Services pricing.

Next steps

As you set up your lab, see the following articles:

- Add users
- Set quotas
- Set a schedule
- Email registration links to students

Set up a lab for big data analytics using Docker deployment of HortonWorks Data Platform

3/5/2021 • 5 minutes to read • Edit Online

This article shows you how to set up a lab to teach a big data analytics class. With this type of class, students learn how to handle large volumes of data and apply machine and statistical learning algorithms to derive data insights. A key objective for students is to learn to use data analytics tools, such as Apache Hadoop's open-source software package which provides tools for storing, managing, and processing big data.

In this lab, students will use a popular commercial version of Hadoop provided by Cloudera, called Hortonworks Data Platform (HDP). Specifically, students will use HDP Sandbox 3.0.1 which is a simplified, easy-to-use version of the platform that is free of cost and intended for learning and experimentation. Although this class may use either Windows or Linux virtual machines (VM) with HDP Sandbox deployed, this article will show how to use Windows.

Another interesting aspect of this lab, is that we will deploy HDP Sandbox on the lab VMs using Docker containers. Each Docker container provides its own isolated environment for software applications to run inside. Conceptually, Docker containers are like nested VMs and can be used to easily deploy and run a wide variety of software applications based on container images provided on Docker Hub. Cloudera's deployment script for HDP Sandbox automatically pulls the HDP Sandbox 3.0.1 Docker image from Docker Hub and runs two Docker containers:

- sandbox-hdp
- sandbox-proxy

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see Tutorial to Setup a Lab Account. You can also use an existing lab account.

Lab account settings

Enable the settings described in the table below for the lab account. For more information about how to enable marketplace images, see Specify Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the Windows 10 Pro image for use within your lab account.

Lab settings

Use the settings in the table below when setting up a classroom lab. For more information how to create a classroom lab, see set up a classroom lab tutorial.

LAB SETTINGS	VALUE/INSTRUCTIONS

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual Machine Size	Medium (Nested Virtualization). This VM size is best suited for relational databases, in-memory caching, and analytics. This size also supports nested virtualization.
Virtual Machine Image	Windows 10 Pro

NOTE

We need to use Medium (Nested Virtualization) since deploying HDP Sandbox using Docker requires:

- Windows Hyper-V with nested virtualization
- At least 10 GB of RAM

Template machine configuration

To set up the template machine, we will:

- Install Docker
- Deploy HDP Sandbox
- Use PowerShell and Windows Task Scheduler to automatically start the Docker containers

Install Docker

The steps in this section are based on Cloudera's instructions for deploying with Docker containers.

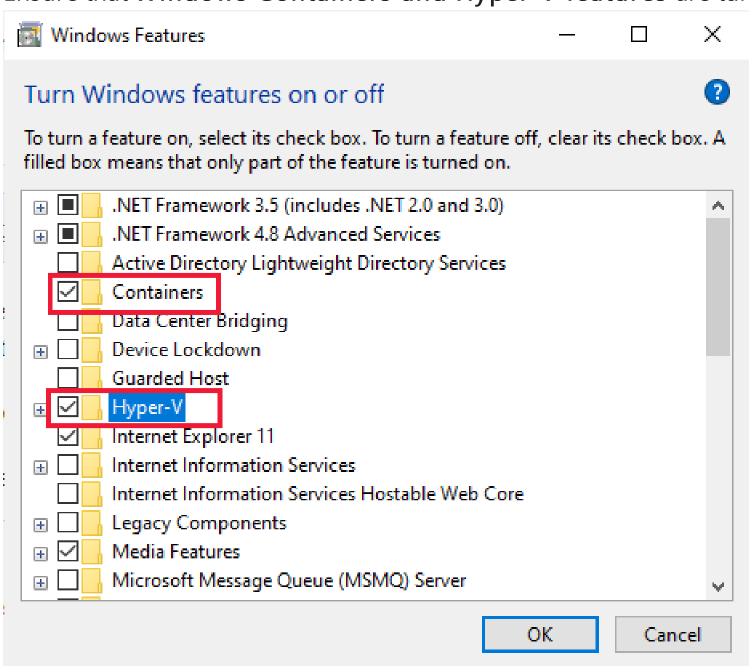
To use Docker containers, you must first install Docker Desktop on the template VM:

1. Follow the steps in the Prerequisites section to install Docker for Windows.

   IMPORTANT

   Ensure that the **Use Windows containers instead of Linux containers** configuration option is left unchecked.

2. Ensure that **Windows Containers and Hyper-V features** are turned on.



3. Follow the steps in the Memory for Windows section to configure Docker's memory configuration.

> **WARNING**
>
> If you inadvertently check the **Use Windows containers instead of Linux containers** option when installing Docker, you won't see the memory configuration settings. To fix this, you can switch to using Linux containers by clicking on the Docker icon in Windows System tray; when the Docker Desktop menu opens, select **Switch to Linux containers**.

Deploy HDP Sandbox

In this section, you will deploy HDP Sandbox and then also access HDP Sandbox using the browser.

1. Ensure that you have installed Git Bash as listed in the Prerequisites section of the guide since this is recommended for completing the next steps.

2. Using Cloudera's Deployment and Install Guide for Docker, complete the steps in the following sections:

   - Deploy HDP Sandbox
   - Verify HDP Sandbox

> **WARNING**
>
> When you download the latest .zip file for HDP, ensure that you do *not* save the .zip file in a directory path that includes whitespace.

> **NOTE**
>
> If you receive an exception during deployment stating **Drive has not been shared**, you need to share your C drive with Docker so that HDP's Linux containers can access local Windows files. To fix this, click on the Docker icon in Windows System tray to open the Docker Desktop menu and select **Settings**. When **Docker's Settings** dialog opens, select **Resources** > **File Sharing** and check the **C** drive. You can then repeat the steps to deploy HDP Sandbox.

3. Once the Docker containers for HDP Sandbox are deployed and running, you can access the environment by launching your browser and following Cloudera's instructions for opening the Sandbox Welcome Page and launching the HDP Dashboard.

> **NOTE**
>
> These instructions assume that you have first mapped the local IP address of the sandbox environment to the sandbox-hdp.hortonworks.com in the host file on your template VM. If you do **not** do this mapping, you can access the Sandbox Welcome page by navigating to `http://localhost:8080`.

Automatically start Docker containers when students log in

To provide an easy to use experience for students, we'll use a PowerShell script that automatically:

- Starts the HDP Sandbox Docker containers when a student starts and connects to their lab VM.
- Launches the browser and navigates to the Sandbox Welcome Page. We'll also use Windows Task Scheduler to automatically run this script when a student logs into their VM. To set this up, follow these steps: Big Data Analytics scripting.

Cost estimate

If you would like to estimate the cost of this lab, you can use the following example.

For a class of 25 students with 20 hours of scheduled class time and 10 hours of quota for homework or assignments, the price for the lab would be:

- 25 students * (20 + 10) hours * 55 Lab Units * 0.01 USD per hour = 412.50 USD

Further more details on pricing, see Azure Lab Services Pricing.

Conclusion

This article walked you through the steps necessary to create a lab for a big data analytics class that using Hortonworks Data Platform deployed with Docker. The setup for this class type may be used for similar data analytics classes. This setup may also be applicable to other types of classes that use Docker for deployment.

Next steps

Next steps are common to setting up any lab.

- Create and manage a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up a lab to teach database management for relational databases

3/5/2021 • 2 minutes to read • Edit Online

This article describes how to set up a lab for a basic databases management class in Azure Lab Services. Databases concepts are one of the introductory courses taught in most of the Computer Science departments in college. Structured Query Language (SQL) is an international standard. SQL is the standard language for relation database management including adding, accessing, and managing content in a database. It is most noted for its quick processing, proven reliability, ease, and flexibility of use.

In this article, we'll show how to set up a virtual machine template in a lab with both MySQL Database Server and SQL Server 2019 server. MySQL is a freely available open source Relational Database Management System (RDBMS). SQL Server 2019 is the latest version of Microsoft's RDBMS.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see Tutorial to Setup a Lab Account. You can also use an existing lab account.

Lab account settings

Enable the settings described in the table below for the lab account. For more information about how to enable marketplace images, see Specify Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the 'SQL Server 2019 Standard on Windows Server 2019' image for use within your lab account.

Lab settings

Use the settings in the table below when setting up a classroom lab. For more information how to create a classroom lab, see set up a classroom lab tutorial.

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual Machine Size	Medium. This size is best suited for relational databases, in-memory caching, and analytics.
Virtual Machine Image	SQL Server 2019 Standard on Windows Server 2019

Template machine configuration

To install MySQL on Windows Server 2019, you can follow the steps mentioned in Install and Run MySQL Community Server on a Virtual Machine.

SQL Server 2019 is pre-installed in the virtual machine image we chose when creating the new lab.

Cost estimate

Let's cover a possible cost estimate for this class. We'll use a class of 25 students. There are 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The virtual machine size we chose was medium, which is 42 lab units.

Here is an example of a possible cost estimate for this class:

25 students * (20 scheduled hours + 10 quota hours) * 0.42 USD per hour = 315.00 USD

Further more details on pricing, see Azure Lab Services Pricing.

Conclusion

This article walked you through the steps necessary to create a lab for basic database management concepts using both MySQL and SQL Server. You can use a similar setup for other databases classes.

Next steps

Next steps are common to setting up any lab.

- Create and manage a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up a lab focused on deep learning in natural language processing using Azure Lab Services

11/2/2020 • 2 minutes to read • Edit Online

This article shows you how to set up a lab focused on deep learning in natural language processing (NLP) using Azure Lab Services. Natural language processing (NLP) is a form of artificial intelligence (AI) that enables computers with translation, speech recognition, and other language understanding capabilities.

Students taking an NLP class get a Linux virtual machine (VM) to learn how to apply neural network algorithms to develop deep learning models that are used for analyzing written human language.

Lab configuration

To set up this lab, you need an Azure subscription to get started. If you don't have an Azure subscription, create a free account before you begin. Once you have an Azure subscription, you can either create a new lab account in Azure Lab Services or use an existing lab account. See the following tutorial for creating a new lab account: Tutorial to Setup a Lab Account.

After you create the lab account, enable following settings in the lab account:

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace images	Enable the Data Science Virtual Machine for Linux (Ubuntu) image for use within your lab account. See the following article for instructions: Specify marketplace images available to lab creators.

Follow this tutorial to create a new lab and apply the following settings:

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual machine (VM) size	**Small GPU (Compute)**. This size is best suited for compute-intensive and network-intensive applications like Artificial Intelligence and Deep Learning.
VM image	Data Science Virtual Machine for Linux (Ubuntu). This image provides deep learning frameworks and tools for machine learning and data science. To view the full list of installed tools on this image, see the following article: What's included on the DSVM?.
Enable remote desktop connection	The Data Science image is already configured to use X2Go so that teachers and students can connect using a GUI remote desktop. X2Go does *not* require the **Enable remote desktop connection** setting to be enabled. This setting only needs to be enabled if you choose to instead use RDP.

> **Important**: Although we recommend using X2Go with the Data Science image, if you choose to instead use RDP, you will need to connect to the Linux VM using SSH the first time and install the RDP and GUI packages. Then, you/students can connect to the Linux VM using RDP later. For more information, see Enable graphical

> remote desktop for Linux VMs.

The Data Science Virtual Machine for Linux image provides the necessary deep learning frameworks and tools required for this type of class. As a result, after the template machine creation, you don't need to customize it further. It can be published for students to use. Select the **Publish** button on template page to publish the template to the lab.

Cost

If you would like to estimate the cost of this lab, you can use the following example:

For a class of 25 students with 20 hours of scheduled class time and 10 hours of quota for homework or assignments, the price for the lab would be - 25 students * (20 + 10) hours * 139 Lab Units * 0.01 USD per hour = 1042.5 USD

Further more details on pricing, see Azure Lab Services Pricing.

Conclusion

This article walked you through the steps to create a lab for natural language processing class. You can use a similar setup for other deep learning classes.

Next steps

Next steps are common to setting up any lab:

- Add users
- Set quota
- Set a schedule
- Email registration links to students.

Set up a lab to teach ethical hacking class

3/5/2021 • 8 minutes to read • Edit Online

This article shows you how to set up a class that focuses on forensics side of ethical hacking. Penetration testing, a practice used by the ethical hacking community, occurs when someone attempts to gain access to the system or network to demonstrate vulnerabilities that a malicious attacker may exploit.

In an ethical hacking class, students can learn modern techniques for defending against vulnerabilities. Each student gets a Windows Server host virtual machine that has two nested virtual machines – one virtual machine with Metasploitable3 image and another machine with Kali Linux image. The Metasploitable virtual machine is used for exploiting purposes and Kali virtual machine provides access to the tools needed to execute forensic tasks.

This article has two main sections. The first section covers how to create the classroom lab. The second section covers how to create the template machine with nested virtualization enabled and with the tools and images needed. In this case, a Metasploitable image and a Kali Linux image on a machine that has Hyper-V enabled to host the images.

Lab configuration

To set up this lab, you need an Azure subscription to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can either create a new lab account in Azure Lab Services or use an existing account. See the following tutorial for creating a new lab account: Tutorial to setup a lab account.

Follow this tutorial to create a new lab and then apply the following settings:

VIRTUAL MACHINE SIZE	IMAGE
Medium (Nested Virtualization)	Windows Server 2019 Datacenter

Template machine

After the template machine is created, start the machine and connect to it to complete the following three major tasks.

1. Set up the machine for nested virtualization. It enables all the appropriate windows features, like Hyper-V, and sets up the networking for the Hyper-V images to be able to communicate with each other and the internet.
2. Set up the Kali Linux image. Kali is a Linux distribution that includes tools for penetration testing and security auditing.
3. Set up the Metasploitable image. For this example, the Metasploitable3 image will be used. This image is created to purposely have security vulnerabilities.

The rest of this article will cover the manual steps to completing the tasks above. Alternatively, you can run the Lab Services Hyper-V Scripts and Lab Services Ethical Hacking Scripts.

Prepare template machine for nested virtualization

Follow instructions to enable nested virtualization to prepare your template virtual machine for nested virtualization.

Set up a nested virtual machine with Kali Linux Image

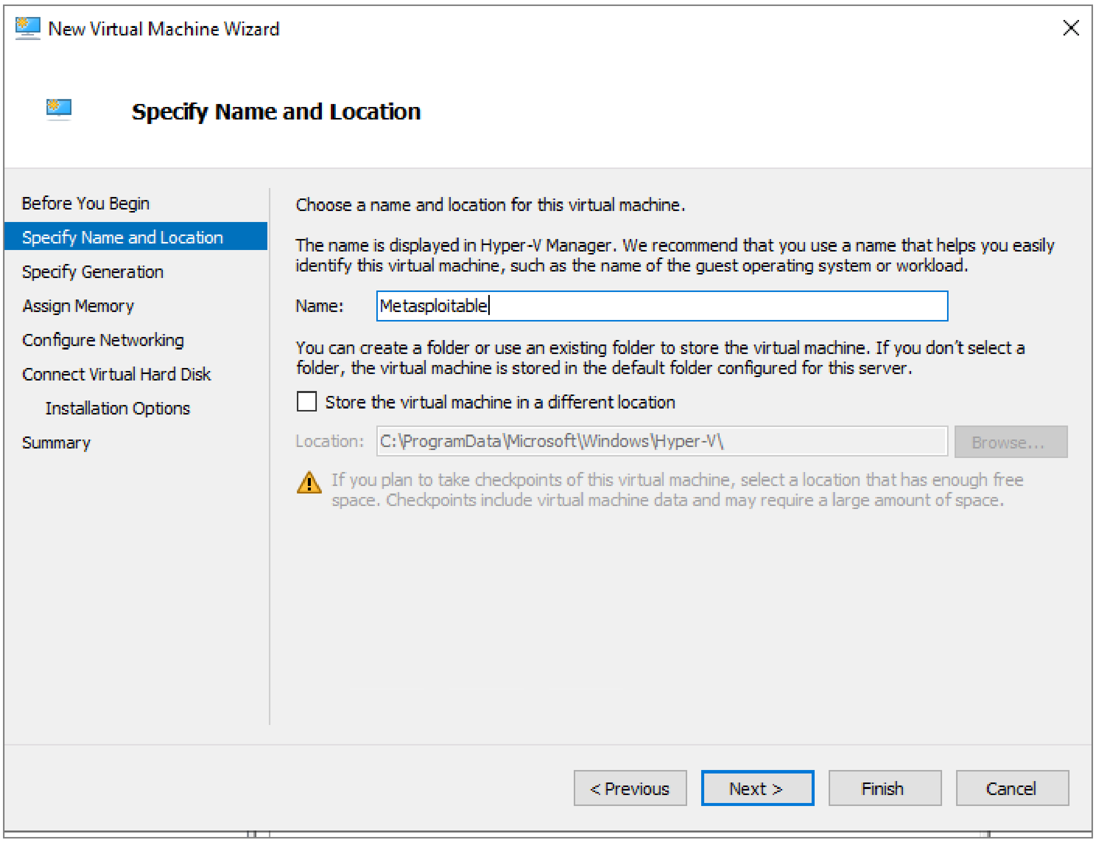
Kali is a Linux distribution that includes tools for penetration testing and security auditing.

1. Download image from Offensive Security Kali Linux VM images. Remember the default username and password noted on the download page.
   a. Download the **Kali Linux VMware 64-Bit (7z)** image for VMware.
   b. Extract the .7z file. If you don't already have 7 zip, download it from https://www.7-zip.org/download.html. Remember the location of the extracted folder as you'll need it later.
2. Convert the extracted vmdk file to a vhdx file so that you can use the vhdx file with Hyper-V. There are several tools available to convert VMware images to Hyper-V images. We'll be using the StarWind V2V Converter. To download, see StarWind V2V Converter download page.
   a. Start **StarWind V2V Converter**.
   b. On the **Select location of image to convert** page, choose **Local file**. Select **Next**.
   c. On the **Source image** page, navigate to and select the Kali Linux vmdk file extracted in the previous step for the **File name** setting. The file will be in the format Kali-Linux-{version}-vmware-amd64.vmdk. Select **Next**.
   d. On the **Select location of destination image**, choose **Local file**. Select **Next**.
   e. On the **Select destination image format** page, choose **VHD/VHDX**. Select **Next**.
   f. On the **Select option for VHD/VHDX image format** page, choose **VHDX growable image**. Select **Next**.
   g. On the **Select destination file name** page, accept the default file name. Select **Convert**.
   h. On the **Converting** page, wait for the image to be converted. This may take several minutes. Select **Finish** when the conversion is completed.
3. Create a new Hyper-V virtual machine.
   a. Open **Hyper-V Manager**.
   b. Choose **Action** -> **New** -> **Virtual Machine**.
   c. On the **Before You Begin** page of the **New Virtual Machine Wizard**, select **Next**.
   d. On the **Specify Name and Location** page, enter **Kali-Linux** for the **name**, and select **Next**.
   e. On the **Specify Generation** page, accept the defaults, and select **Next**.
   f. On the **Assign Memory** page, enter **2048 MB** for the **startup memory**, and select **Next**.
   g. On the **Configure Networking** page, leave the connection as **Not Connected**. You'll set up the network adapter later.
   h. On the **Connect Virtual Hard Disk** page, select **Use an existing virtual hard disk**. Browse to the location for the **Kali-Linux-{version}-vmware-amd64.vmdk** file created in the previous step, and select **Next**.
   i. On the **Completing the New Virtual Machine Wizard** page, and select **Finish**.
   j. Once the virtual machine is created, select it in the Hyper-V Manager. Don't turn on the machine yet.
   k. Choose **Action** -> **Settings**.
   l. On the **Settings for Kali-Linux** dialog for, select **Add Hardware**.
   m. Select **Legacy Network Adapter**, and select **Add**.
   n. On the **Legacy Network Adapter** page, select **LabServicesSwitch** for the **Virtual Switch** setting, and select **OK**. LabServicesSwitch was created when preparing the template machine for Hyper-V in the **Prepare Template for Nested Virtualization** section.
   o. The Kali-Linux image is now ready for use. From **Hyper-V Manager**, choose **Action** -> **Start**, then choose **Action** -> **Connect** to connect to the virtual machine. The default username is **kali** and the password is **kali**.

Set up a nested VM with Metasploitable Image

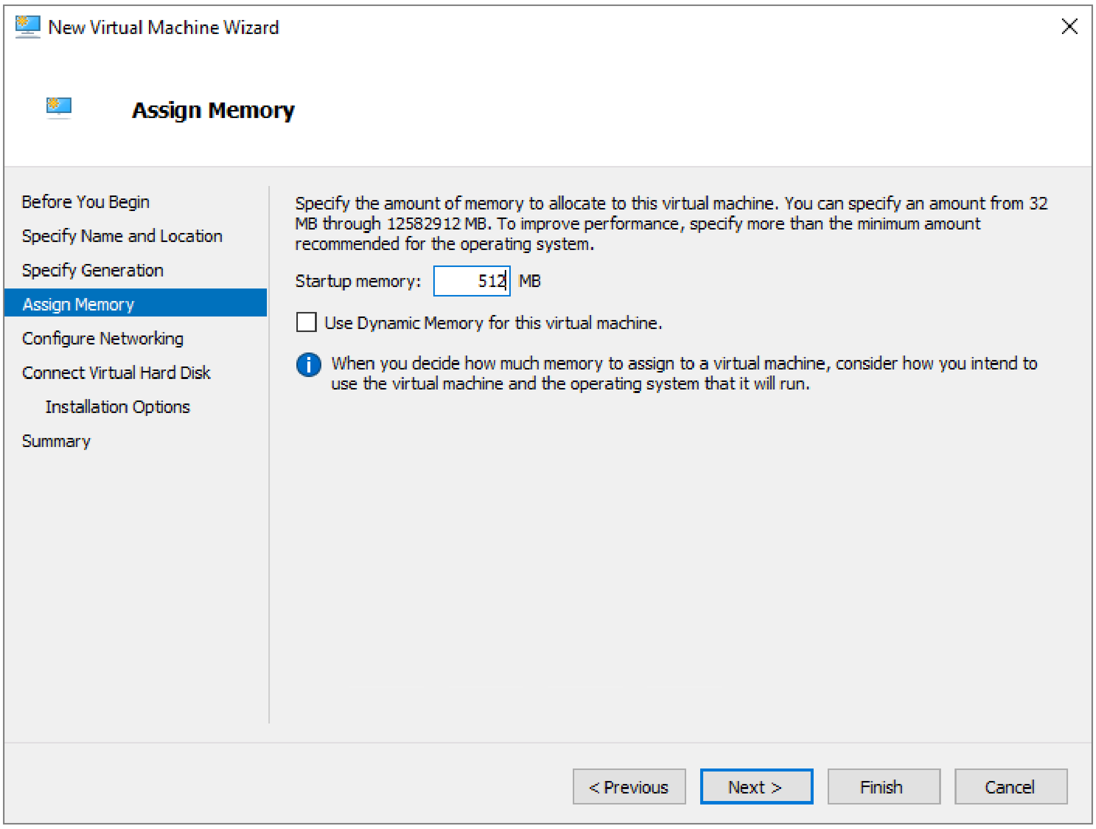
The Rapid7 Metasploitable image is an image purposely configured with security vulnerabilities. You'll use this image to test and find issues. The following instructions show you how to use a pre-created Metasploitable image. However, if a newer version of the Metasploitable image is needed, see https://github.com/rapid7/metasploitable3.

1. Download the Metasploitable image.
   a. Navigate to https://information.rapid7.com/download-metasploitable-2017.html. Fill out the form to download the image and select the **Submit** button.
   b. Select the **Download Metasploitable Now** button.
   c. When the zip file is downloaded, extract the zip file, and remember the location of the Metasploitable.vmdk file.
2. Convert the extracted vmdk file to a vhdx file so that you can use the vhdx file with Hyper-V. There are several tools available to convert VMware images to Hyper-V images. We'll be using the StarWind V2V Converter again. To download, see StarWind V2V Converter download page.
   a. Start **StarWind V2V Converter**.
   b. On the **Select location of image to convert** page, choose **Local file**. Select **Next**.
   c. On the **Source image** page, navigate to and select the Metasploitable.vmdk extracted in the previous step for the **File name** setting. Select **Next**.
   d. On the **Select location of destination image**, choose **Local file**. Select **Next**.
   e. On the **Select destination image format** page, choose **VHD/VHDX**. Select **Next**.
   f. On the **Select option for VHD/VHDX image format** page, choose **VHDX growable image**. Select **Next**.
   g. On the **Select destination file name** page, accept the default file name. Select **Convert**.
   h. On the **Converting** page, wait for the image to be converted. This may take several minutes. Select **Finish** when the conversion is completed.
3. Create a new Hyper-V virtual machine.
   a. Open **Hyper-V Manager**.
   b. Choose **Action** -> **New** -> **Virtual Machine**.
   c. On the **Before You Begin** page of the **New Virtual Machine Wizard**, select **Next**.
   d. On the **Specify Name and Location** page, enter **Metasploitable** for the **name**, and select **Next**.



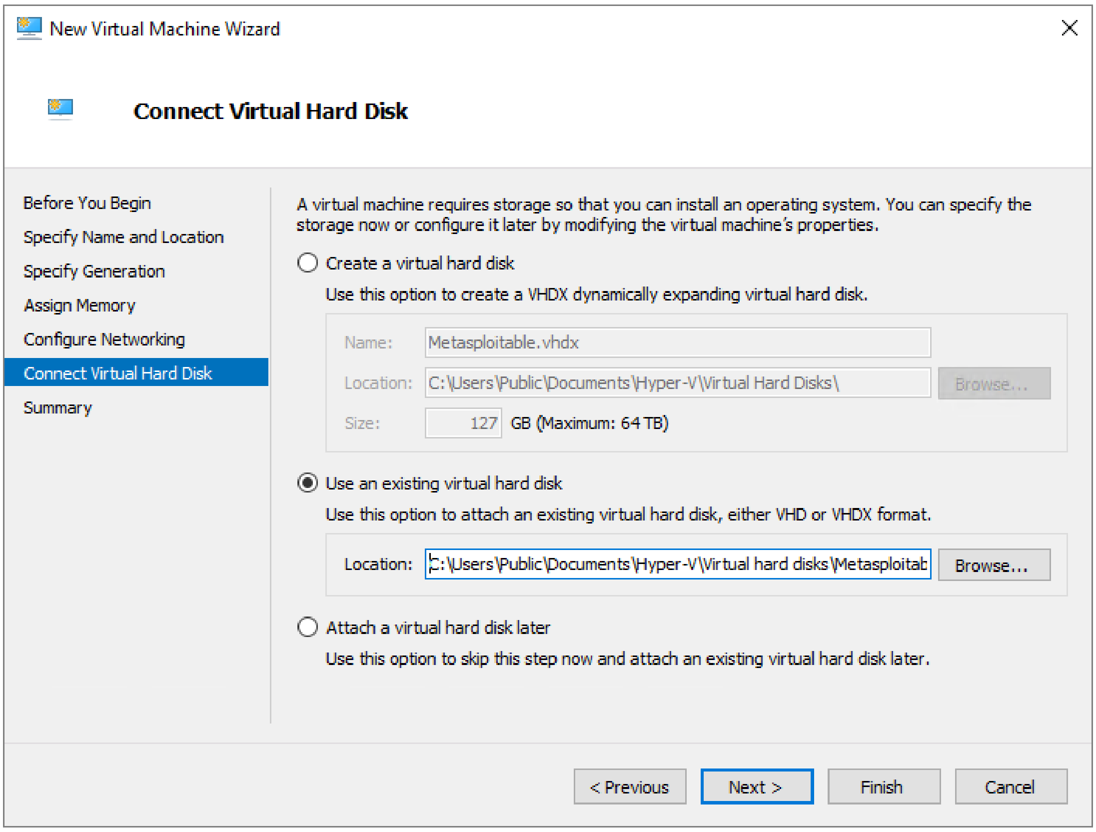
e. On the **Specify Generation** page, accept the defaults, and select **Next**.

f. On the **Assign Memory** page, enter **512 MB** for the **startup memory**, and select **Next**.



g. On the **Configure Networking** page, leave the connection as **Not Connected**. You'll set up the network adapter later.

h. On the **Connect Virtual Hard Disk** page, select **Use an existing virtual hard disk**. Browse to the location for the **metasploitable.vhdx** file created in the previous step, and select **Next**.



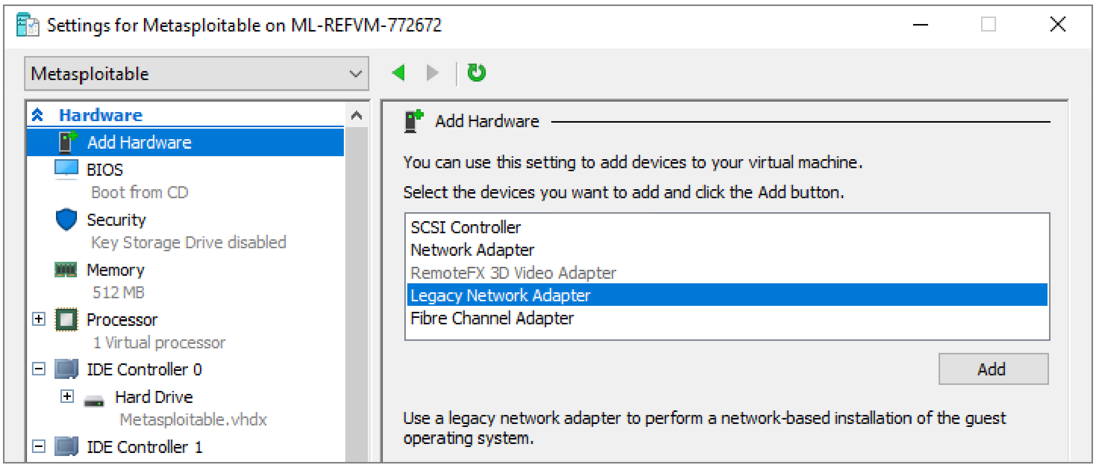
i. On the **Completing the New Virtual Machine Wizard** page, and select **Finish**.

j. Once the virtual machine is created, select it in the Hyper-V Manager. Don't turn on the machine yet.

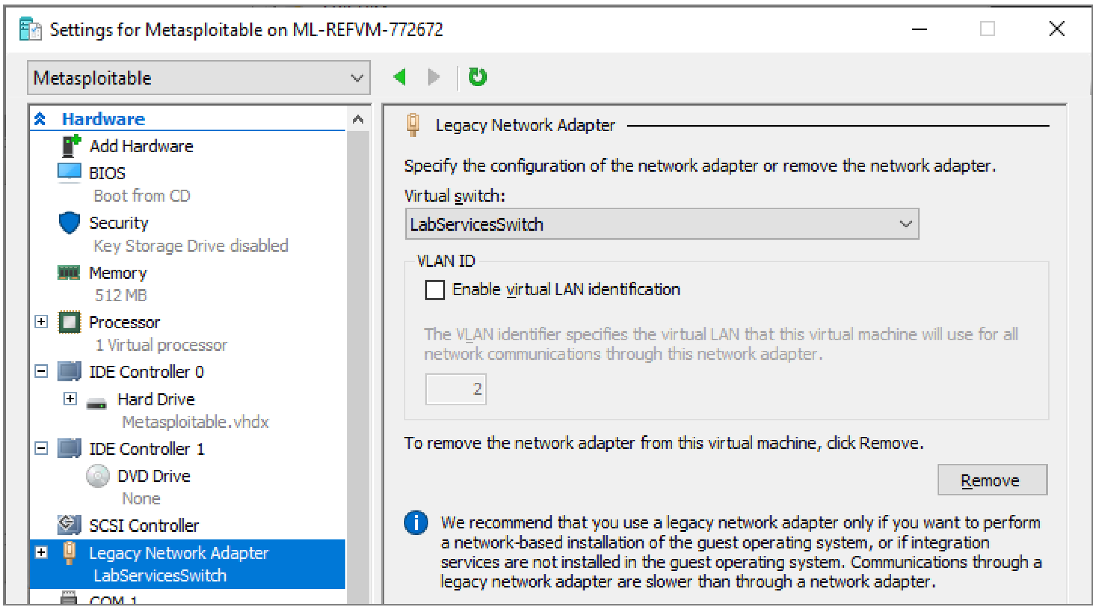
k. Choose **Action** -> **Settings**.

l. On the **Settings for Metasploitable** dialog for, select **Add Hardware**.

m. Select **Legacy Network Adapter**, and select **Add**.



n. On the **Legacy Network Adapter** page, select **LabServicesSwitch** for the **Virtual Switch** setting, and select **OK**. LabServicesSwitch was created when preparing the template machine for Hyper-V in the **Prepare Template for Nested Virtualization** section.



o. The Metasploitable image is now ready for use. From **Hyper-V Manager**, choose **Action** -> **Start**, then choose **Action** -> **Connect** to connect to the virtual machine. The default username is **msfadmin** and the password is **msfadmin**.

The template is now updated and has images needed for an ethical hacking penetration testing class, an image with tools to do the penetration testing and another image with security vulnerabilities to discover. The template image can now be published to the class. Select the **Publish** button on template page to publish the template to the lab.

Cost

If you would like to estimate the cost of this lab, you can use the following example:

For a class of 25 students with 20 hours of scheduled class time and 10 hours of quota for homework or assignments, the price for the lab would be:

25 students * (20 + 10) hours * 55 Lab Units * 0.01 USD per hour = 412.50 USD

> **IMPORTANT**
>
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Conclusion

This article walked you through the steps to create a lab for ethical hacking class. It includes steps to set up nested virtualization for creating two virtual machines inside the host virtual machine for penetrating testing.

Next steps

Next steps are common to setting up any lab:

- Add users
- Set quota
- Set a schedule
- Email registration links to students.

Set up a lab to teach ethical hacking class with VirtualBox

6/29/2021 • 3 minutes to read • Edit Online

This article shows you how to set up a class that focuses on forensics side of ethical hacking. Penetration testing, a practice used by the ethical hacking community, occurs when someone attempts to gain access to the system or network to demonstrate vulnerabilities that a malicious attacker may exploit.

In an ethical hacking class, students can learn modern techniques for defending against vulnerabilities. Each student gets a host virtual machine that has three nested virtual machines – two virtual machine with Seed image and another machine with Kali Linux image. The Seed virtual machine is used for exploiting purposes and Kali virtual machine provides access to the tools needed to execute forensic tasks.

This article has two main sections. The first section covers how to create the classroom lab. The second section covers how to create the template machine with nested virtualization enabled and with the tools and images needed. In this case, two Seed images and a Kali Linux image on a machine that has VirtualBox enabled to host the images.

Lab configuration

To set up this lab, you need an Azure subscription to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can either create a new lab account in Azure Lab Services or use an existing account. See the following tutorial for creating a new lab account: Tutorial to setup a lab account.

Follow this tutorial to create a new lab and then apply the following settings:

VIRTUAL MACHINE SIZE	IMAGE
Medium (Nested Virtualization)	Windows Server 2019 Datacenter
Medium (Nested Virtualization)	Windows 10

Template machine

After the template machine is created, start the machine and connect to it to complete the following three major tasks.

1. Set up the machine to use VirtualBox for nested virtual machines.
2. Set up the Kali Linux image. Kali is a Linux distribution that includes tools for penetration testing and security auditing.
3. Set up the Seed image. For this example, the Seed image will be used. This image is created specifically for security training.

The rest of this article will cover the manual steps to completing the tasks above.

Installing VirtualBox

1. Download the VirtualBox platform packages by selecting the Windows hosts option.
2. Run the installation executable, and use the default options to complete the installation.

Set up a nested virtual machine with Kali Linux Image

Kali is a Linux distribution that includes tools for penetration testing and security auditing.

1. Download the ova image from Kali Linux VM VirtualBox images. We recommend the 32bit version, the 64bit version loads with errors. Remember the default username and password noted on the download page.
2. Open VirtualBox Manager and import the .ova image.. The Kali licensing agreement will need to be reviewed and accepted to continue.

> **NOTE**
> - The VirtualBox default Ram for the Kali VM is 2 gig (2048), We recommend increasing the Ram to at least 4 gig (4096) or more depending on your needs. This can be changed by the students on their VMs. Changing the RAM size within VirtualBox does not change the Lab's VM size.
> - By default the Hard disk is set to an 80 gig limit, but is dynamically allocated. Lab Service machines are limited to 128 gigs of hard drive space, so be careful not to exceed this disk size.
> - The Kali image has USB 2.0 enable which requires Oracle VM VirtualBox Extension Pack or set the USB controller to 1.0 under the USB tab.

Setup Seed lab images

1. Download and extract the SEED Labs VM image..
2. Follow the directions to create a VM in VirtualBox. If you need multiple SEED VMs make copies of the .iso for each machine, using the same .iso for different machines will not work properly.

> **IMPORTANT**
> Make sure that all the nested virtual machines are powered off before publishing the template. Leaving them powered on has had unexpected side effects, including damaging the virtual machines.

Cost

If you would like to estimate the cost of this lab, you can use the following example:

For a class of 25 students with 20 hours of scheduled class time and 10 hours of quota for homework or assignments, the price for the lab would be:

25 students * (20 + 10) hours * 55 Lab Units * 0.01 USD per hour = 412.50 USD

> **IMPORTANT**
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Conclusion

This article walked you through the steps to create a lab for ethical hacking class. It includes steps to set up nested virtualization for creating two virtual machines inside the host virtual machine for penetrating testing.

Next steps

Next steps are common to setting up any lab:

- Add users
- Set quota
- Set a schedule

- Email registration links to students.

Setup a lab to teach MATLAB

11/2/2020 • 5 minutes to read • Edit Online

MATLAB, which stands for Matrix laboratory, is programming platform from MathWorks. It combines computational power and visualization making it popular tool in the fields of math, engineering, physics, and chemistry.

If you're using a campus-wide license, see directions at download MATLAB installation files to download the MATLAB installer files on the template machine.

In this article, we'll show how to set up a class that uses MATLAB client software with a license server.

License server

Before modifying the template machine for your lab, you'll need to set up the server to run the Network License Manager software. These instructions are only applicable for institutions that choose the networking licensing option for MATLAB, which allows users to share a pool of license keys. You'll also need to save the license file and file installation key for later. For detailed instructions on how to download a license file, see the first step in the install Network License Manager with internet connection article.

Detailed instructions to covering how to install a licensing server are available at install Network License Manager with Internet Connection. To enable borrowing, see the Borrow License article.

Assuming the license server is located in an on-premise network or a private network within Azure, don't forget to peer the virtual network to your lab account. The network peering must be done before creating the lab so the lab virtual machines can access the license server.

Lab configuration

To set up this lab, you need an Azure subscription to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can either create a new lab account in Azure Lab Services or use an existing account. To create a new lab account, see the setup a lab account tutorial.

To create a new lab, follow set up a classroom lab tutorial. Apply the following settings:

VIRTUAL MACHINE SIZE	IMAGE
Medium	Windows 10

MATLAB is supported on more operating systems. See MATLAB system requirements for details.

> **WARNING**
> Don't forget to peer the virtual network for the lab account to the virtual network for the license server before creating the lab.

Template machine

After the template machine is created, start the machine and connect to it to complete the following major tasks.

1. Download the installation files for the MATLAB client software.

2. Install MATLAB using the file installation key.

Installing MATLAB will be a multi-part process. The first part will download the files for MATLAB and any other products you want installed. Using a file installation key requires that all the installation files for products to be installed are pre-downloaded. The second part will be installing the MATLAB software on the template VM and activating the software. If the template VM is configured to activate using the license server, the student VMs will do the same.

Download installation files

You must be a license administrator to download the installation files as well as get the license file and file installation key. Steps to download the installation files are below.

1. Log into your account for https://www.mathworks.com.
2. Choose **My Account**.
3. Under the **My Software** section of the account page, click on the license attached to the Network License Manager setup for the lab.
4. On the license detail page, click **Download Products**.
5. Wait for the installer to self-extract.
6. Start the installer.
7. On the **Sign in to your MathWorks Account** page, enter your MathWorks account.
8. On the **MathWorks License Agreement** page, accept the term and click the **Next** button.
9. Click the **Advanced Options** drop-down and choose **I want to download without installing**.
10. On the **Select destination folder**, click **Next**.
11. Select **Windows** as the platform of the computer you're going to be installing MATLAB.
12. On the **Select product** page, make sure MATLAB is selected along with any other MathWorks products you would like to install.
13. On the **Confirm Selections and Download** page, click **Begin Download**.
14. Wait for the selected products to download. Click **Finish**.

You can also download an ISO image from the MathWorks website.

1. Log into your account for https://www.mathworks.com.
2. Go to https://www.mathworks.com/downloads.
3. Select the release of MATLAB you wish to install.
4. Click the “Get {version}.iso image” link beneath the Related links where {version} is something like R2020a.
5. Click the blue **download release** link for Windows.

Run installer

Once the files are downloaded, the second step is to run the installer. Once again, you must be a license administrator to complete this step. Only license administrators can install MATLAB with a file installation key.

1. Check the downloaded license file and verify the SERVER line lists the license server correctly. For information regarding how the license file should be formatted, see update network license, license borrowing, and find host ID articles.
2. Launch the MATLAB Installer.
3. On the **Sign in to your MathWorks Account** page, enter your MathWorks account.
4. On the **MathWorks License Agreement** page, accept the term and click the **Next** button.
5. Click the **Advanced Options** drop-down and choose **I have a File Installation Key**.
6. On the **Install using File Installation Key** page, enter the file installation key for the license server. Click **Next**.
7. On the **Select License File** page, navigate to the license file saved when downloading the installation files earlier.

8. On the **Select Destination Folder** page, click **Next**.
9. On the **Select Products** page, click **Next**.
10. On the **Select Options** page, click **Next**.
11. On the **Confirm Selections and Install** page, click **Begin Install**.
12. On the **Installation Complete** page, verify **Activate MATLAB** is checked. Click **Finish**.

Cost estimate

Let's cover a possible cost estimate for this class. This estimate does not include the cost of running the license server. We'll use a class of 25 students. There are 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The virtual machine size we chose was medium, which is 55 lab units.

Here is an example of a possible cost estimate for this class:

25 students * (20 scheduled hours + 10 quota hours) * 55 lab units * 0.01 USD per hour = 412.50 USD

> **IMPORTANT**
>
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Next steps

Next steps are common to setting up any lab.

- Create, manage, and publish a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up a lab to teach a networking class

3/5/2021 • 3 minutes to read • Edit Online

This article shows you how to set up a class that focuses on allowing students to emulate, configure, test, and troubleshoot virtual and real networks using GNS3 software.

This article has two main sections. The first section covers how to create the classroom lab. The second section covers how to create the template machine with nested virtualization enabled and with GNS3 installed and configured.

Lab configuration

To set up this lab, you need an Azure subscription to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can either create a new lab account in Azure Lab Services or use an existing account. See the following tutorial for creating a new lab account: Tutorial to setup a lab account.

Follow this tutorial to create a new lab and then apply the following settings:

VIRTUAL MACHINE SIZE	IMAGE
Large (Nested Virtualization)	Windows 10 Pro, Version 1909

Template machine

After the template machine is created, start the machine and connect to it to complete the following three major tasks.

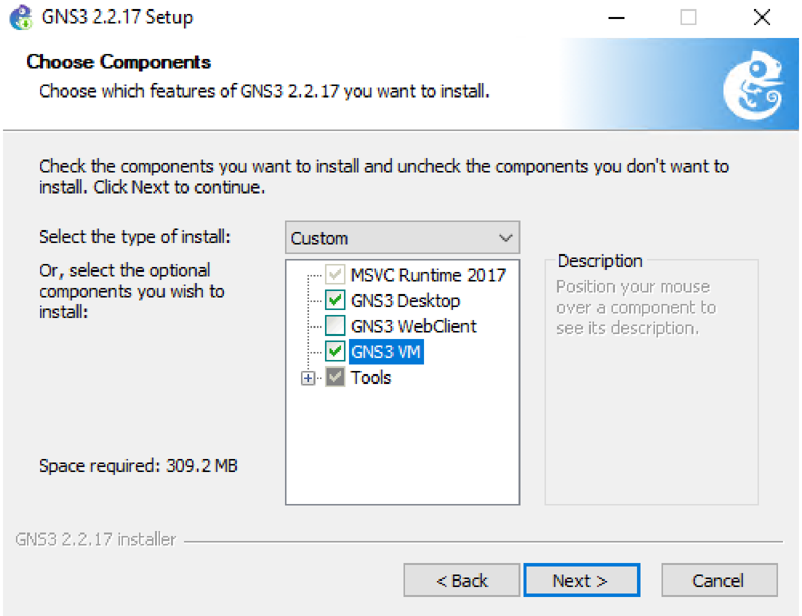
1. Prepare the template machine for nested virtualization.
2. Install GNS3.
3. Create nested GNS3 VM in Hyper-V.
4. Configure GNS3 to use Windows Hyper-V VM.
5. Add appropriate appliances.
6. Publish template.

Prepare template machine for nested virtualization

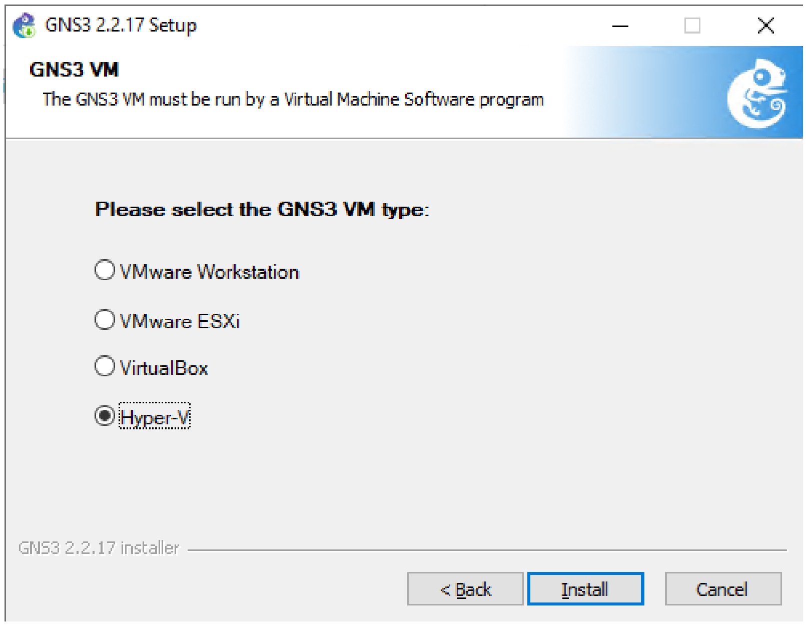
- Follow instructions in this article to prepare your template virtual machine for nested virtualization.

Install GNS3

- Follow the instructions for installing GNS3 on Windows. Make sure to include installing the **GNS3 VM** in the component dialog, see below.



Eventually you'll reach the GNS3 VM selection. Make sure to select the **Hyper-V** option.

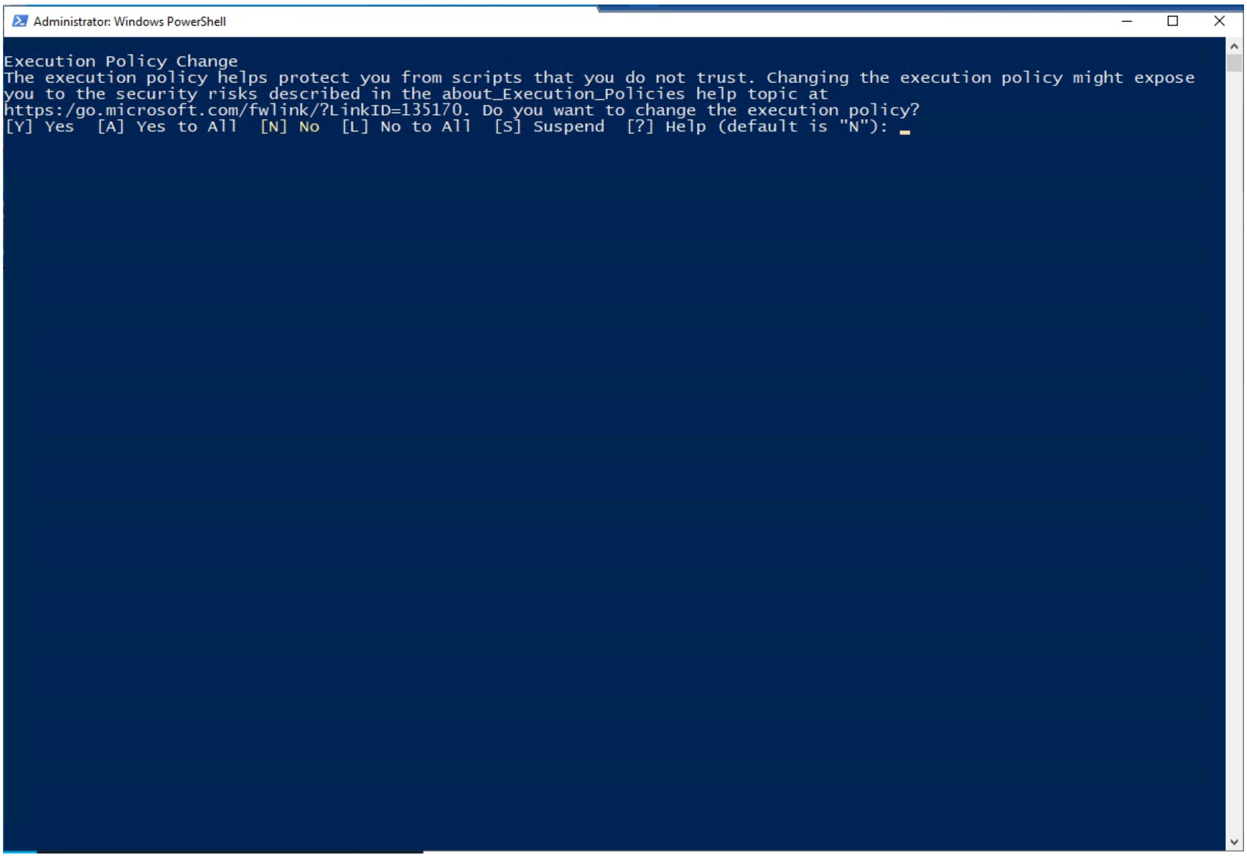


This option will download the PowerShell script and VHD files to create the GNS3 VM in the Hyper-V manager. Continue installation using the default values. **Once the setup is complete, don't start GNS3**.

Create GNS3 VM

Once the setup has completed, a zip file **"GNS3.VM.Hyper-V.2.2.17.zip"** is downloaded to the same folder as the installation file, containing the drives and the PowerShell script to create the Hyper-V vm.

- **Extract all** on the GNS3.VM.Hyper-V.2.2.17.zip. This action will extract out the drives and the PowerShell script to create the VM.
- **Run with PowerShell** on the "create-vm.ps1" PowerShell script by right clicking on the file.
- An Execution Policy Change request may show up. Enter "Y" to execute the script.

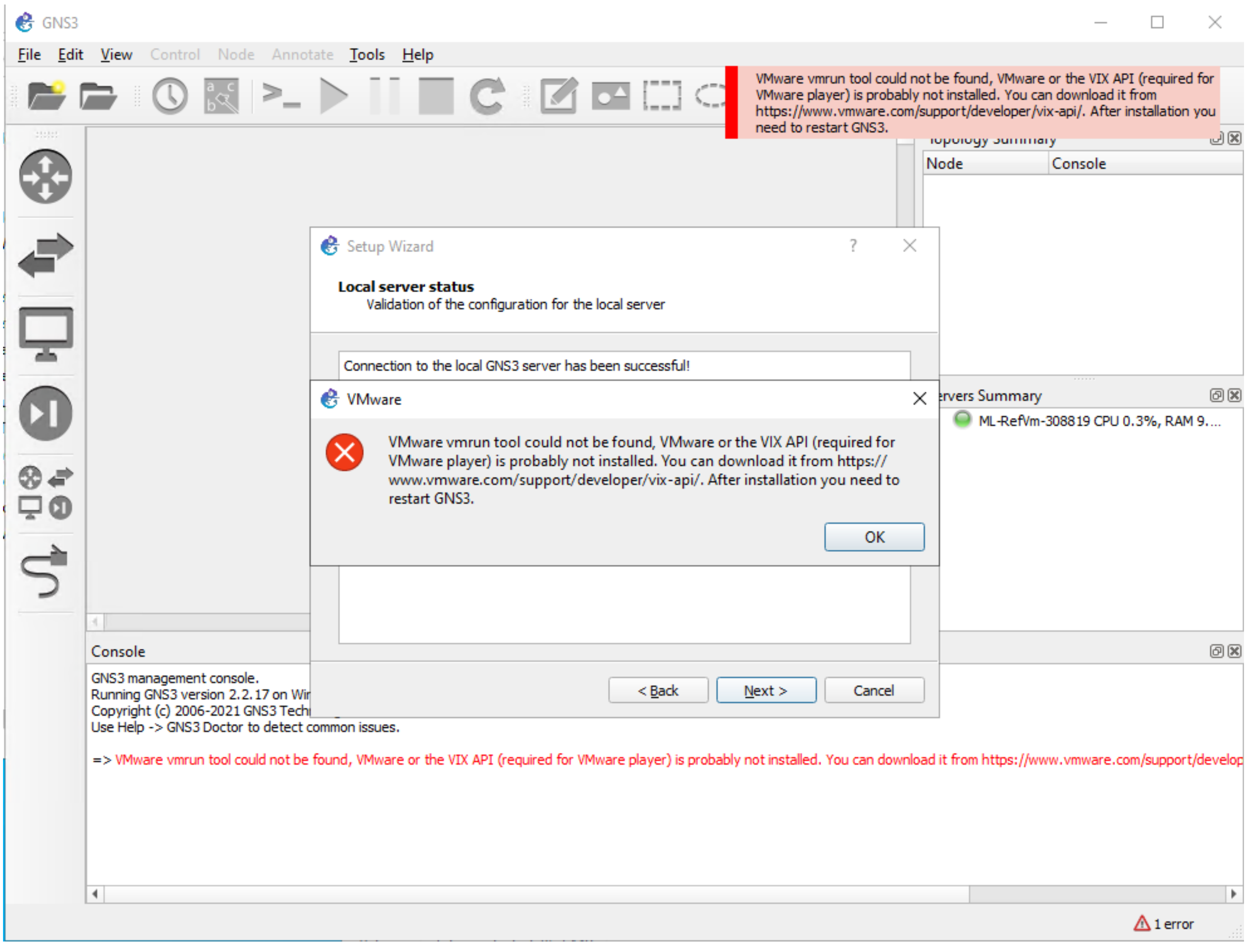


- Once the script has completed, you can confirm the VM "GNS3 VM" has been created in the Hyper-V Manager.

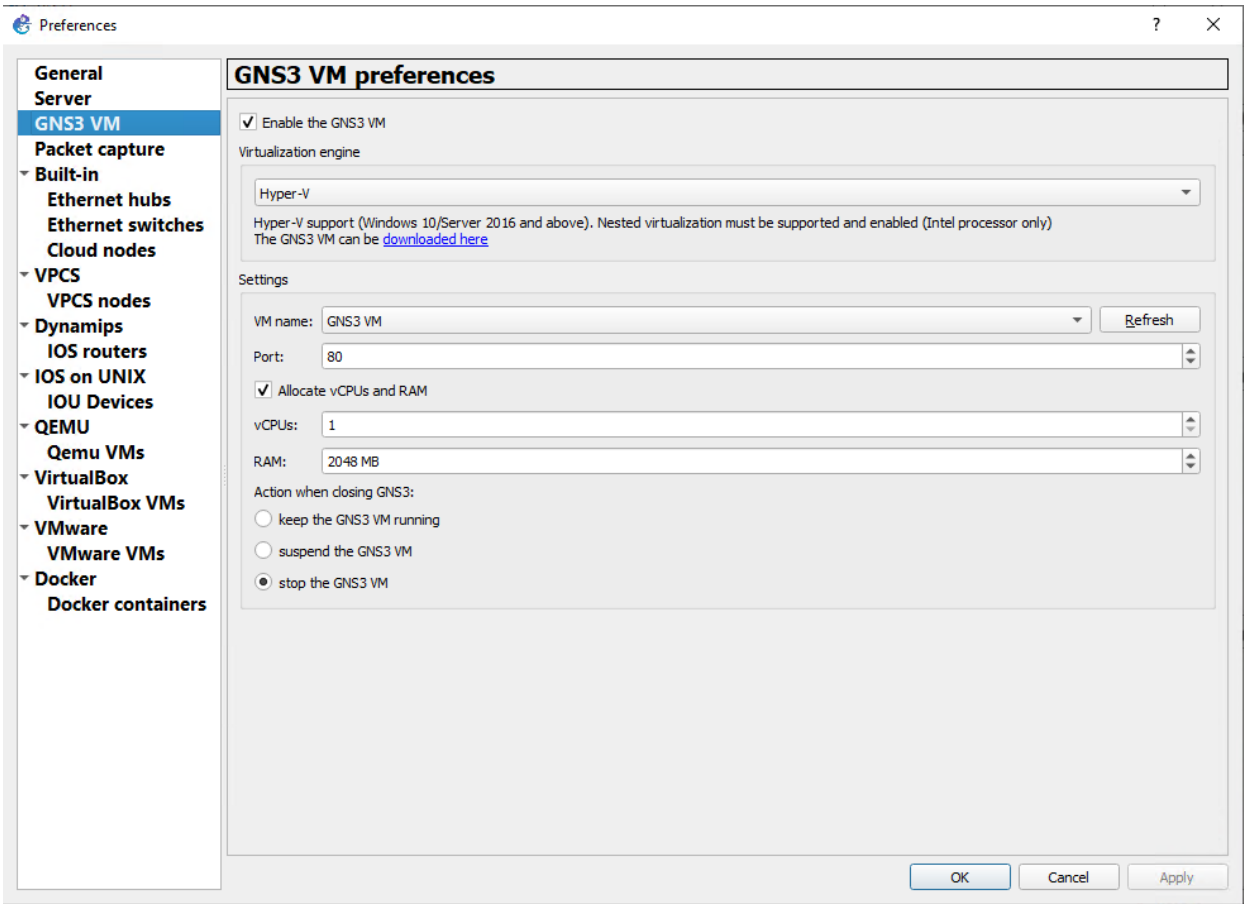
Configure GNS3 to use Hyper-V VM

Now that GNS3 is installed and the GNS3 VM is added, start up GNS3 to link the two together. The GNS3 Setup wizard will start automatically..

- Use the **Run appliances from virtual machine.** option. Use the defaults for the rest of the wizard until you hit the **VMware vmrun tool cannot be found.** error.



- Choose **Ok**, and **Cancel** out of the wizard.
- To complete the connection to the Hyper-V vm, open the **Edit -> Preferences -> GNS3 VM** and select **Enable the GNS3 VM** and select the **Hyper-V** option.



Add appropriate appliances

At this point, you'll want to add the appropriate appliances for the class.

Publish template

Now that the template VM is set up properly, and ready for publishing there are a few key points to check.

- Make sure that the GNS3 VM is shut down or turned off. Publishing while the VM is still running will corrupt the VM.
- Close down GNS3, publishing while and running can lead to unintended side effects.
- Clean up any installation files or other unnecessary files.

Cost

If you would like to estimate the cost of this lab, you can use the following example:

For a class of 25 students with 20 hours of scheduled class time and 10 hours of quota for homework or assignments, the price for the lab would be:

25 students * (20 + 10) hours * 84 Lab Units * 0.01 USD per hour = 630 USD.

Important: Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Conclusion

This article walked you through the steps to create a lab for network configuration using GNS3.

Next steps

Next steps are common to setting up any lab:

- Add users
- Set quota
- Set a schedule
- Email registration links to students.

Set up labs for Project Lead The Way classes

8/5/2021 • 7 minutes to read • Edit Online

Project Lead The Way (PLTW) is a nonprofit organization that provides PreK–12 curriculum across the United States in computer science, engineering, and biomedical science. In each PLTW class, students use a variety of software applications as part of their hands-on learning experience. Many of the software applications require either a fast CPU or, in some cases, a GPU. This article shows you how to set up labs for the following PLTW classes, which are typically offered to students in grades 9–12:

- **Introduction to Engineering Design**

  Students are introduced to the process of engineering design, which includes using Autodesk Inventor computer-aided design (CAD) software for 3D modeling.

- **Principles of Engineering**

  Students learn about engineering mechanisms, structural and material strength, and automation. This class uses software such as MD Solids, West Point Bridge Designer, and America's Army simulation.

- **Civil Engineering and Architecture**

  Students learn building and site design and development by using Autodesk Revit architecture design software for 3D building information modeling (BIM).

- **Computer Integrated Manufacturing**

  Students explore modern manufacturing processes that involve robotics and automation. In this class, students use Autodesk Inventor CAD and Autodesk Inventor computer-aided manufacturing (CAM) software.

- **Digital Electronics**

  Students study electronic logic circuits and devices by using National Instrument Multisim simulation and circuit design software.

- **Engineering Design and Development**

  Students contribute to an end-to-end solution by combining research, design, and testing that they present to a panel of engineers. In this class, students use Autodesk Inventor CAD software.

- **Computer Science Essentials**

  Students are introduced to computational concepts and tools. They start with block-based programming and then move to text-based coding by using coding environments such as VEXcode V5 blocks.

- **Computer Science Principles**

  Students grow their programming expertise with Python by using the Microsoft Visual Studio Code development environment.

- **Computer Science A**

  Students expand their programming competence in this class by learning mobile app development. In this class, they learn Java by using the Microsoft Visual Studio Code development environment. Students also use an emulator that allows them to run and test their mobile app code. For information about how to set up an emulator in Azure Lab Services, contact us via the Azure Lab Services' forums for more information.

For a full list of class software, go to the PLTW site for each class.

Lab configuration

To begin setting up labs for PLTW, you need an Azure subscription and lab account. If you don't have an Azure subscription, create a free account before you begin.

After you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see Set up a lab account. You can also use an existing lab account.

After you've set up a lab account, you should create a separate lab for each PLTW class session that your school offers. We also recommend that you create separate images for each type of PLTW class. For more information about how to structure your labs and images, see the blog post Moving from a Physical Lab to Azure Lab Services.

Lab account settings

Enable your lab account settings as described in the following table. For more information about how to enable Azure Marketplace images, see Specify the Azure Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the Windows 10 Pro image for use within your lab account.

Lab settings

The size of the virtual machine (VM) that we recommend using for PLTW classes depends on the types of workloads that your students are doing in the class. For the earlier-listed classes, we recommend using Small GPU (Visualization) and Large VM sizes. As you set up labs for your PLTW classes, refer to the guidance in the following table:

LAB SETTING	VALUE AND DESCRIPTION	CLASS RECOMMENDATION
Virtual Machine Size	**Small GPU (Visualization)** Best suited for remote visualization, streaming, gaming, and encoding with frameworks such as OpenGL and DirectX.	We recommend using this size for the following PLTW classes: Civil Engineering and Architecture, Digital Electronics, Computer Integrated Manufacturing, Engineering Design and Development, and Introduction to Engineering Design.
Virtual Machine Size	**Large** Best suited for applications that need faster CPUs, better local disk performance, large databases, and large memory caches.	We recommend using this size for the following PLTW classes: Principles of Engineering, Computer Science Essentials, Computer Science Principles, and Computer Science A.

License server

Most of the software that's used in the earlier-mentioned PLTW classes do *not* require access to a license server. However, you'll need to access a license server if you plan to use the Autodesk network licensing model for the following software:

- Revit
- Inventor CAD
- Inventor CAM

To use network licensing with Autodesk software, PLTW provides detailed steps to install Autodesk Network License Manager on your license server. This license server is ordinarily located in either your on-premises network or hosted on an Azure virtual machine (VM) within in Azure virtual network.

After your license server is set up, you'll need to peer the virtual network with your lab account. You need to do the network peering *before* you create the lab so that your lab VMs can access the license server and vice versa.

Autodesk-generated license files embed the MAC address of the license server. If you decide to host your license server by using an Azure VM, it's important to make sure that your license server's MAC address doesn't change. If the MAC address changes, you'll need to regenerate your licensing files. To prevent your MAC address from changing, do the following:

- Set a static private IP and MAC address for the Azure VM that hosts your license server.
- Be sure to set up both your lab account and the license server's virtual network in a region or location that has sufficient VM capacity so that you don't have to move these resources to a new region or location later.

For more information, see Set up a license server as a shared resource.

Template machine

Some of the installation files that you need for PLTW are large. When you download the files to a lab template VM, they might take a long time to copy.

Instead of downloading installation files to the template machine and installing everything there, we recommend creating your PLTW images in your physical environment. You can then import the custom images into a shared image gallery so that you can use them to create your labs. For more information, see Recommended approaches for creating custom images.

As you follow this recommendation, note the major tasks for setting up a lab:

1. In your physical environment, create the image for the class.

   a. Use PLTW's detailed steps for downloading the installation files and installing the required software.

   > **NOTE**
   >
   > When you install the Autodesk applications, the computer that you're installing them on needs to be able to communicate with your license server. The Autodesk installation wizard will prompt you to specify the computer name of the machine that the license server is hosted on. If you're hosting your license server on an Azure VM, you might need to wait to install Autodesk on the lab template VM so that the installation wizard can access your license server.

   b. Install and configure OneDrive or other backup options that your school might use.

   c. Install and configure Windows updates.

2. Upload the custom image to the shared image gallery that's attached to your lab account.

3. Create a lab, and then select the custom image that you uploaded in the preceding step.

4. After the lab is created, start and connect to the template VM to validate that the image works as expected.

5. Finally, publish the template VM to create the students' VMs.

NOTE

If your school needs to perform content filtering, such as for compliance with the Children's Internet Protection Act (CIPA), you will need to use 3rd party software. For more information, read guidance on content filtering with Lab Services.

Student devices

Students can connect to their lab VMs from Windows computers, Mac, and Chromebook. For instructions, see:

- Connect from Windows
- Connect from Mac
- Connect from Chromebook

Cost

Let's cover an example cost estimate for the PLTW classes. This estimate doesn't include the cost of running a license server or using a shared image gallery. Suppose you have a class of 25 students, each of whom has 20 hours of scheduled class time. Each student also has an additional 10 quota hours for homework or assignments outside of scheduled class time. Here are the estimated costs:

- **Large VM**

  25 students × (20 scheduled hours + 10 quota hours) × 70 Lab Units × USD0.01 per hour = USD525.00

- **Small GPU (visualization)**

  25 students × (20 scheduled hours + 10 quota hours) × 160 Lab Units × USD0.01 per hour = USD1200.00

IMPORTANT

The cost estimate is for example purposes only. For current pricing information, see Azure Lab Services pricing.

NOTE

Many of the PLTW classes use applications that are accessed via a browser, such as MIT App Inventor. These browser-based applications don't require a fast CPU or GPU, and you can access them from any device that has an internet connection. When students are using these types of applications, we recommend that they use the browser on their physical device instead the browser on their lab VM. Students can help keep costs down by using their lab VM only for applications that require a fast CPU or GPU.

Next steps

As you set up your lab, see the following articles:

- Add users
- Set quotas
- Set a schedule
- Email registration links to students

Set up a lab to teach data science with Python and Jupyter Notebooks

5/10/2021 • 10 minutes to read • Edit Online

This article outlines how to set up a template virtual machine (VM) in Lab Services with the tools that are needed to teach students how to use Jupyter Notebooks, and how students can connect to their notebooks on their virtual machines (VMs).

Jupyter Notebooks is an open-source project that lets you easily combine rich text and executable Python source code on a single canvas called a notebook. Running a notebook results in a linear record of inputs and outputs. Those outputs can include text, tables of information, scatter plots, and more.

Set up the lab

Lab configuration

To set up this lab, you need access to an Azure subscription and a lab account. Discuss with your organization's admin to see if you can get access to an existing Azure subscription. If you don't have an Azure subscription, create a free account before you begin.

Once you have an Azure subscription, create a new lab account in Azure Lab Services by following instructions in the tutorial: Setup a lab account. You can also use an existing lab account.

Lab account settings

Enable settings described in the table below for the lab account. For more information on enabling marketplace images, see specify Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Inside your lab account, enable one of the Azure Marketplace images based on your operating system needs: • Data Science Virtual Machine – Windows Server 2019 • Data Science Virtual Machine – Ubuntu 18.04

> **NOTE**
> This article uses the Data Science virtual machine images available on the Azure marketplace because they are preconfigured with Jupyter Notebook. These images, however, also include many other development and modeling tools for data science. If you don't want those extra tools and want a lightweight setup with just Jupyter notebooks, create a custom VM image. For an example, Installing JupyperHub on Azure. Once the custom image is created, you can upload it to a shared image gallery to use the image inside Azure Lab Services. Learn more about using Shared Image Gallery in Azure Lab Services.

Lab settings

Configure **Virtual machine size** and **Virtual machine image** settings as shown in the following table when setting up a classroom lab. For instructions on creating a classroom lab, see Set up a classroom lab.

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual machine size	The size you pick here depends on the workload you want to run: • Small or Medium – good for a basic setup of accessing Jupyter Notebooks • Small GPU (Compute) – best suited for compute-intensive and network-intensive applications like Artificial Intelligence and Deep Learning
Virtual machine image	Choose one of the following images based on your operating system needs: • Data Science Virtual Machine – Windows Server 2019 • Data Science Virtual Machine – Ubuntu 18.04

When you create a lab with the **Small GPU (Compute)** size, you have the option to Install GPU drivers. This option installs recent NVIDIA drivers and Compute Unified Device Architecture (CUDA) toolkit which are required to enable high-performance computing with the GPU. For more information, see the article Set up a lab with GPU virtual machines.

Template virtual machine

Once you create a lab, a template VM will be created based on the virtual machine size and image you chose. You configure the template VM with everything you want to provide to your students for this class. To learn more, see how to manage the template virtual machine.

The Data Science VM images by default come with many of data science frameworks and tools required for this type of class. For example, the images include:

- Jupyter Notebooks: A web application that allows data scientists to take raw data, run computations, and see the results all in the same environment. It will run locally in the template VM.
- Visual Studio Code: An integrated development environment (IDE) that provides a rich interactive experience when writing and testing a notebook. For more information, see Working with Jupyter Notebooks in Visual Studio Code.

If you are using the **Small GPU (Compute)** size, we recommend that you verify that the Data Science frameworks and libraries are properly set up with the GPU. To properly set up the frameworks and libraries, you may need to install a different version of the NVIDIA Drivers and CUDA toolkit. For example, to validate that the GPU is configured for TensorFlow, you can connect to the template VM and run the following Python-TensorFlow code in Jupyter Notebooks:

```
import tensorflow as tf
from tensorflow.python.client import device_lib

print(device_lib.list_local_devices())
```

If the output from the above code looks like the following, this means that the GPU isn't configured for TensorFlow:

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 15833696144144374634
]
```

To properly configure the GPU, you should consult the framework's or library's documentation. Continuing with the above example, TensorFlow provides the following guidance:

- TensorFlow GPU Support

Their guidance covers the required version of the NVIDIA drivers and CUDA Toolkit. Their guidance also includes installing the NVIDIA CUDA Deep Neural Network library (cudDNN).

After you've followed TensorFlow's steps to configure the GPU, when you rerun the above code, you should see output similar to the following:

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 15833696144144374634
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 11154792128
locality {
  bus_id: 1
  links {
  }
}
incarnation: 2659412736190423786
physical_device_desc: "device: 0, name: NVIDIA Tesla K80, pci bus id: 0001:00:00.0, compute capability: 3.7"
]
```

Provide notebooks for the class

The next task is to provide students with notebooks that you want them to use. To provide your own notebooks, you can save notebooks locally on the template VM.

If you want to use sample notebooks from Azure Machine Learning, see how to configure an environment with Jupyter Notebooks.

Optional: enable graphical desktop for Linux

The **Data Science Virtual Machine – Ubuntu** image is already provisioned with X2GO server and is ready to accept client connections. No further steps are required when setting up the template VM.

Publish the template machine

When you publish the template, each student registered to your lab will get a copy of the template VM with all the local tools and notebooks you've set up on it.

How students connect to Jupyter Notebooks?

Once you publish the template, each student will have access to a VM that comes with everything you've preconfigured for the class, including the Jupyter Notebooks. The following sections show different ways for students to connect to Jupyter Notebooks.

For Windows VMs

If you've provided students with Windows VMs, they need to connect to their VMs and use Jupyter Notebooks that are available locally on them.

To connect to a Windows VM, a student can use a remote desktop connection (RDP). For detailed steps, see how to access a classroom lab.

A student using a Mac or Chromebook can follow instructions from following articles to connect to the Data Science Windows VM.

- Connect to a VM using RDP on a Mac
- Connect to a VM using RDP on a Chromebook

For Linux VMs

If you've provided students with Linux VMs, there are several options students can use to connect to their Jupyter Notebooks in the VMs:

- Access Jupyter Notebooks locally after connecting to the VM
  - SSH to the VM for terminal sessions
  - X2Go connection to the VM for graphical sessions
- Use SSH tunneling to connect from the student's local computer directly to the Jupyter Server on the VM.

The following sections provide details about these ways to connect to Jupyter notebooks.

SSH to virtual machine

Students can connect via SSH to their Linux VMs from a terminal session. For detailed steps, see how to access a classroom lab. If they are using a Windows client machine, they will need to enable an SSH client by downloading PuTTY or enabling OpenSSH in Windows to SSH from the command prompt.

1. Start the VM.
2. Once the VM is running, click **Connect**, which will pop up a dialog box that provides the SSH command string, which will look like the following sample:

```
ssh -p 12345 student@ml-lab-00000000-0000-0000-0000-000000000000.eastus2.cloudapp.azure.com
```

3. Go to your command prompt or terminal, and paste in this command, and then press **ENTER**.
4. Enter the password to sign in to the VM.

Once students are connected to VMs, they can access and run Jupyter Notebooks locally.

X2Go to virtual machine

The **Data Science Virtual Machine – Ubuntu** image is already provisioned with X2GO Server and is ready to accept client connections. To connect to the graphical desktop of the Linux machine, students need to follow these one-time steps to set up X2Go on their client machines:

1. Download and install the X2Go client for your client platform.
2. In the Azure Lab Services portal, make sure that the Linux VM you want to connect to is started.
3. Once the VM is running, click **Connect**, which will pop a dialog box that provides the SSH command string, which will look like the following sample:

```
ssh -p 12345 student@ml-lab-00000000-0000-0000-0000-000000000000.eastus2.cloudapp.azure.com
```

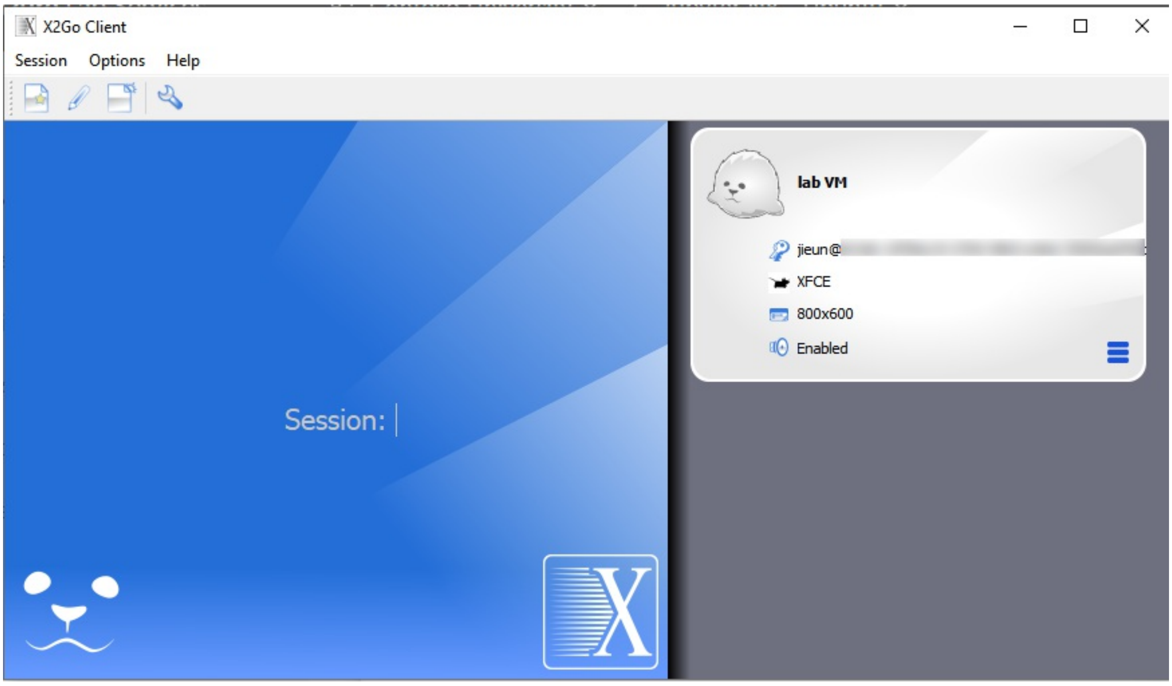
4. Once you have this info, open the X2Go client app and create a new session.

5. Fill in the following values in the **Session Preferences** pane:
   - **Session name**: It can be whatever you want, but we recommend using the name of your Lab VM.
   - **Host**: `ml-lab-00000000-0000-0000-0000-000000000000.eastus2.cloudapp.azure.com`
   - **Login**: student
   - **SSH port**: 12345
   - **Session type**: XFCE
6. Select **OK**.

> **NOTE**
>
> When creating a new X2Go session, make sure to use the SSH port, **not** the RDP port.

Now, to connect to the VM, follow these steps:

1. In the X2Go client, double-click on the VM you want to connect to.



2. Enter the password to connect to the VM. (You may have to give X2Go permission to bypass your firewall to finish connecting.)
3. You should now see the graphical interface for your Ubuntu Data Science VM.

SSH tunnel to Jupyter server on the VM

Some students may want to connect directly from their local computer directly to the Jupyter server inside their VMs. The SSH protocol enables port forwarding between the local computer and a remote server (in our case, the student's lab VM), so that an application running on a certain port on the server is **tunneled** to the mapping port on the local computer. Students should follow these steps to SSH tunnel to the Jupyter server on their lab VMs:

1. In the Azure Lab Services portal, make sure that the Linux VM that you want to connect is started.
2. Once the VM is running, click **Connect**, which will pop a dialog box that provides the SSH command string, which will look like the following string:

```
ssh -p 12345 student@ml-lab-00000000-0000-0000-0000-000000000000.eastus2.cloudapp.azure.com
```

3. On your local computer, launch a terminal or command prompt, and copy the SSH connection string to it. Then, add `-L 8888:localhost:8888` to the command string, which creates the **tunnel** between the ports. The final string should look like:

```
ssh -L 8888:localhost:8888 -p 12345 student@ml-lab-b720853e-570f-49ac-9cb2-bd0bd2aeec35.eastus.cloudapp.azure.com
```

4. Press **ENTER** to run the command.

5. When prompted, provide the password to connect to the lab VM.

6. Once you're connected to the VM, start the Jupyter server using this command:

```
jupyter notebook
```

7. Running the command will provide you with a URL in the terminal or command prompt. The URL should look like:

```
http://localhost:8888/?token=8c09ecfc93e6a8cbedf9c66dffdae19670a64acc1d37
```

8. Paste this URL into a browser on your local computer to connect and work on your Jupyter Notebook.

> **NOTE**
> Visual Studio Code also enables a great Jupyter Notebook editing experience. You can follow the instructions on how to connect to a remote Jupyter server and use the same URL from the previous step to connect from VS Code instead of from the browser.

Cost estimate

Let's cover a possible cost estimate for this class. We'll use a class of 25 students. There are 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The VM size we chose was small GPU (compute), which is 139 lab units. If you want to use the Small (20 lab units) or Medium size (42 lab units), you can replace the lab unit part in the equation below with the correct number.

Here is an example of a possible cost estimate for this class: 25 students * (20 scheduled hours + 10 quota hours) * 139 lab units * 0.01 USD per hour = 1042.5 USD

Further more details on pricing, see Azure Lab Services Pricing.

Conclusion

In this article, we walked through the steps to create a lab for a Jupyter Notebooks class. You can use a similar setup for other machine learning classes.

Next steps

Next steps are common to setting up any lab.

- Create and manage a template
- Add users
- Set quota

- Set a schedule
- Email registration links to students

Set up lab for React on Linux

6/28/2021 • 3 minutes to read • Edit Online

React is a popular JavaScript library for building user interfaces (UI). React is a declarative way to create reusable components for your website. There are many other popular libraries for JavaScript-based front-end development. We'll use a few of these libraries while creating our lab. Redux is a library that provides predictable state container for JavaScript apps and is often used in compliment with React. JSX is a library syntax extension to JavaScript often used with React to describe what the UI should look like. NodeJS is a convenient way to run a webserver for your React application.

This article will show how to install Visual Studio Code for your development environment, the tools, and libraries needed for a React web development class.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable your lab account settings as described in the following table. For more information about how to enable Azure Marketplace images, see Specify the Azure Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace images	Enable the 'Ubuntu Server 18.04 LTS' image for use within your lab account.

Lab settings

The size of the virtual machine (VM) that we recommend depends on the types of workloads that your students need to do.

LAB SETTING	VALUE AND DESCRIPTION
Virtual Machine Size	**Small**.

We recommend testing your workloads to see if a larger size is needed. For more information about each size, see VM sizing.

Template machine configuration

The steps in this section show how to complete the following to set up the template VM:

1. Install Development tools.
2. Install debugger extensions for your web browser.
3. Update firewall settings.

Install Development tools

1. Install your preferred web browser.

2. Install Node.js.

```
sudo apt install nodejs
```

3. Install Node Package Manager, which will be used for installing the React, Redux, and JSX.

```
sudo apt install npm
```

4. Install Visual Studio Code.

5. Install Reactive Native Tools extension for Visual Studio Code.

6. Optionally, install extensions for development with Redux and JSX.

Create React App is an officially supported way to create a ReactApp and requires no further configuration if using npm 5.2 and above. For instructions using Create React App, see their getting started documentation.

Other components needed for a React-based website are installed using NPM into a specific application. For example, enter the following commands to install the Redux and JSX libraries:

```
npm install react-redux
npm install react-jsx
```

Install debugger extensions

Install the React Developer Tools extensions for your browser so you can inspect React components and record performance information.

- React Developer Tools Edge add-on
- React Developer Tools Chrome extension
- React Developer Tools FireFox add-on

To run the app in development mode, use the `npm start` built-in command. The local and network urls will be listed in the command output. To use HTTPS instead of HTTP, see create React app using https in development.

Update firewall settings

Official Ubuntu builds have iptables installed and will allow all incoming traffic by default. However, if you're using a VM that has a more restrictive firewall, add an inbound rule to allow traffic to the NodeJS server. The example below uses iptables to allow traffic to port 3000.

```
sudo iptables -I INPUT -p tcp -m tcp --dport 3000 -j ACCEPT
```

> **IMPORTANT**
>
> Instructors must use the template VM or another lab VM to access a student's website.

Cost

Let’s cover an example cost estimate for this class. Suppose you have a class of 25 students. Each student has 20 hours of scheduled class time. Another 10 quota hours for homework or assignments outside of scheduled class time is given to each student. The virtual machine size we chose was **Small**, which is 20 lab units.

- 25 students × (20 scheduled hours + 10 quota hours) × 20 Lab Units × USD0.01 per hour = 150.00 USD

IMPORTANT

The cost estimate is for example purposes only. For current pricing information, see Azure Lab Services pricing.

Next steps

The template image can now be published to the lab. See publish the template VM for further instructions.

As you set up your lab, see the following articles:

- Add users
- Set quotas
- Set a schedule
- Email registration links to students

Set up lab for React on Windows

6/21/2021 • 3 minutes to read • Edit Online

React is a popular JavaScript library for building user interfaces (UI). React is a declarative way to create reusable components for your website. There are many other popular libraries for JavaScript-based front-end development. We'll use a few of these libraries while creating our lab. Redux is a library that provides predictable state container for JavaScript apps and is often used in compliment with React. JSX is a library syntax extension to JavaScript often used with React to describe what the UI should look like. NodeJS is a convenient way to run a webserver for your React application.

This article will show how to install Visual Studio 2019 for your development environment, and the tools, and libraries needed for a React web development class.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable your lab account settings as described in the following table. For more information about how to enable Azure Marketplace images, see Specify the Azure Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable 'Visual Studio 2019 Community (latest release) on Windows Server 2019 (x64)' image for use within your lab account.

Lab settings

The size of the virtual machine (VM) that we recommend depends on the types of workloads that your students need to do.

LAB SETTING	VALUE AND DESCRIPTION
Virtual Machine Size	**Medium**

We recommend testing your workloads to see if a larger size is needed. For more information about each size, see VM sizing.

Template machine configuration

The steps in this section show how to complete the following to set up the template VM:

1. Install Development tools.
2. Install debugger extensions for your web browser.
3. Update firewall settings.

Install Development tools

The 'Visual Studio 2019 Community (latest release) on Windows Server 2019 (x64)' image already has the required **Node.js development** workload installed for Visual Studio 2019.

1. Install your preferred web browser. The image has Internet Explorer installed by default.
2. Navigate to Node.js website and select the **Download** button. You can use the latest long-term service (LTS) version, current version with that latest features, or a previous release. Installing NodeJS will also install Node Package Manager, which will be used for installing the React, Redux, and JSX.
3. Update Visual Studio 2019 to the latest release, if needed.

Other components needed for a React-based website are installed using NPM into a specific application. To add NPM packages, see manage your NPM packages in Visual Studio.

For example, if using the Node.js Interactive Window in a project, enter the following commands to install the React, Redux, and JSX libraries:

```
.npm install react
.npm install react-dom
.npm install react-redux
.npm install react-jsx
```

To create your first Node.js with React app in Visual Studio, see Tutorial: Create a Node.js and React app in Visual Studio.

Install debugger extensions

Install the React Developer Tools extensions for your browser so you can inspect React components and record performance information.

- React Developer Tools Edge add-on
- React Developer Tools Chrome extension
- React Developer Tools FireFox add-on

Update firewall settings

By default, inbound traffic to your Node.js server will be blocked. If you wish to access a student's website while it's running, add an in-bound firewall rule to allow the traffic. Look at the **Application Port** project property to see which port will be used during debugging. The example below uses the New-NetFirewallRule PowerShell cmdlet to allow access to port 1337.

```
New-NetFirewallRule -DisplayName "Allow access to Port 1337" -Direction Inbound -LocalPort 1337 -Protocol TCP -Action Allow
```

> **IMPORTANT**
>
> Instructors must use the template VM or another lab VM to access a student's website.

Cost

Let's cover an example cost estimate for this class. Suppose you have a class of 25 students. Each student has 20 hours of scheduled class time. Another 10 quota hours for homework or assignments outside of scheduled class time is given to each student. The virtual machine size we chose was **Medium**, which is 55 lab units.

- 25 students × (20 scheduled hours + 10 quota hours) × 55 Lab Units × USD0.01 per hour = 412.50 USD

> **IMPORTANT**
>
> The cost estimate is for example purposes only. For current pricing information, see Azure Lab Services pricing.

Next steps

The template image can now be published to the lab. See publish the template VM for further instructions.

As you set up your lab, see the following articles:

- Add users
- Set quotas
- Set a schedule
- Email registration links to students

Set up a lab to teach R on Linux

9/10/2021 • 5 minutes to read • Edit Online

R is an open-source language used for statistical computing and graphics. It's used in the statistical analysis of genetics to natural language processing to analyzing financial data. R provides an interactive command line experience. RStudio is an interactive development environment (IDE) available for the R language. The free version provides code editing tools, an integrated debugging experience, and package development tools.

This article will focus on solely RStudio and R as a building block for a class that requires the use of statistical computing. The deep learning and Python and Jupyter Notebooks class types setup RStudio differently. Each article describes how to use the Data Science Virtual Machine for Linux (Ubuntu) marketplace image, which has many data science related tools, including RStudio, pre-installed.

Lab Account configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable your lab account settings as described in the following table. For more information about how to enable Azure Marketplace images, see Specify the Azure Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace images	Ubuntu Server 18.04 LTS
Enable peer virtual network	Enable if: • Class requires a shared R Server. • Class requires large data files that you want to store externally and not on the student VM.

> **IMPORTANT**
> If you choose to enable peer virtual network, this must be done before the lab is created.

Lab configuration

For instructions to create a new lab and apply the needed settings, see Tutorial: Set up a classroom lab. When creating the lab, apply the following settings:

LAB SETTING	VALUE AND DESCRIPTION
Virtual Machine Size	Small GPU (Compute)
VM image	Ubuntu Server 18.04 LTS

LAB SETTING	VALUE AND DESCRIPTION
Enable remote desktop connection	This setting should be enabled if you choose to use RDP. This setting isn't needed if you choose X2Go to connect to lab machines. You'll need to connect to the Linux VM using SSH the first time and install the RDP/X2Go and GUI packages. For more information, see enable graphical remote desktop for Linux VMs.

External resource configuration

Some classes require files, such as large data files, to be stored externally. See use external file storage in Azure Lab Services for options and setup instructions.

If you choose to have a shared R Server for the students, the server should be set up before the lab is created. For more information on how to set up a shared server, see how to create a lab with a shared resource in Azure Lab Services. For instructions to create an RStudio Server, see Download RStudio Server for Debian & Ubuntu and Accessing RStudio Server Open-Source.

Template configuration

After the template machine is created, start the machine, and connect to it to install R, RStudio Desktop and optionally X2Go Server.

First, let's update apt and upgrade existing packages on the machine.

```
sudo apt update
sudo apt upgrade
```

Install X2Go Server

If you choose to use X2Go, install the server. You'll first need to connect using ssh to install the server component. Once that is completed, the rest of the setup can be completed after connecting using the X2Go client.

The default installation of X2Go isn't compatible with RStudio. To work around this issue, update the x2goagent options file.

1. Edit `/etc/x2go/x2goagent.options` file. Don't forget to edit file as sudo.
   a. Uncomment the line that states: `X2GO_NXAGENT_DEFAULT_OPTIONS+=" -extension GLX"`
   b. Comment the line that states: `X2GO_NXAGENT_DEFAULT_OPTIONS+=" -extension GLX"`
2. Restart the X2Go server so the new options are used.

```
sudo systemctl restart x2goserver
```

Alternatively, you can build the required libraries by following instructions at Glx Xlib workaround for X2Go.

Install R

There are a few ways to install R on the VM. We'll install R from the Comprehensive R Archive Network (CRAN) repository. It provides the most up-to-date versions of R. Once this repository is added to our machine, we can install R and many other related packages.

We need to add the CRAN repository. Commands are modified from instructions available at Ubuntu Packages for R brief instructions.

```
#download helper packages
sudo apt install --no-install-recommends software-properties-common dirmngr
# download and add the signing key (by Michael Rutter) for these repos
sudo wget -q "https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc" -O
/etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
#add repository
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/"
```

Now we can install R, running the following command:

```
sudo apt install r-base
```

Install RStudio

Now that we have R installed locally, we can install the RStudio IDE. We'll install the free version of RStudio Desktop. For all available versions, see RStudio downloads.

1. Import the code signing key for RStudio.

   ```
   sudo gpg --keyserver keyserver.ubuntu.com  --recv-keys 3F32EE77E331692F
   ```

2. Download the Debian Linux Package file (.deb) for R Studio for Ubuntu. File will be in the format `rstudio-{version}-amd64.deb`. For example:

   ```
   export rstudiover="1.4.1717"
   wget --quiet -O rstudio.deb https://download1.rstudio.org/desktop/bionic/amd64/rstudio-$rstudiover-
   amd64.deb
   ```

3. Use gdebi to install RStudio. Make sure to use the file path to indicate to apt that were installing a local file.

   ```
   sudo apt install gdebi-core
   echo "y" | gdebi rstudio.deb –quiet
   ```

CRAN packages

Now it's time to install any CRAN packages you want. First, add the current R 4.0 or later 'c2d4u' repository.

```
sudo add-apt-repository ppa:c2d4u.team/c2d4u4.0+
```

Use the `install.packages(“package name”)` command in an R interactive session as shown in quick list of useful R packages article. Alternately, use Tools -> Install Packages menu item in RStudio.

If you need help with finding a package, see a list of packages by task or alphabetic list of packages.

Cost

Let's cover an example cost estimate for this class. Suppose you have a class of 25 students. Each student has 20 hours of scheduled class time. Another 10 quota hours for homework or assignments outside of scheduled class time is given to each student. The virtual machine size we chose was **Small GPU (Compute)**, which is 139 lab units.

25 students × (20 scheduled hours + 10 quota hours) × 139 Lab Units × 0.01 USD per hour = 1042.5 USD

> **IMPORTANT**
>
> The cost estimate is for example purposes only. For current pricing information, see Azure Lab Services pricing.

Next steps

The template image can now be published to the lab. See publish the template VM for further instructions.

As you set up your lab, see the following articles:

- Add users
- Set quotas
- Set a schedule
- Email registration links to students

Set up a lab to teach R on Windows

9/10/2021 • 5 minutes to read • Edit Online

R is an open-source language used for statistical computing and graphics. It's used in the statistical analysis of genetics to natural language processing to analyzing financial data. R provides an interactive command line experience. RStudio is an interactive development environment (IDE) available for the R language. The free version provides code editing tools, an integrated debugging experience, and package development tools.

This article will focus on solely RStudio and R as a building block for a class that requires the use of statistical computing. The deep learning and Python and Jupyter Notebooks class types setup RStudio differently. Each article describes how to use the Data Science Virtual Machine for Linux (Ubuntu) marketplace image, which has many data science related tools, including RStudio, pre-installed.

Lab Account configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable your lab account settings as described in the following table.

LAB ACCOUNT SETTING	INSTRUCTIONS
Enable peer virtual network	Enable if: • Class requires a shared R Server. • Class requires large data files that you want to store externally and not on the student VM.

> **IMPORTANT**
> If you choose to enable peer virtual network, this must be done before the lab is created.

Lab configuration

For instructions to create a new lab and apply the needed settings, see Tutorial: Set up a classroom lab. When creating the lab, apply the following settings:

LAB SETTING	VALUE AND DESCRIPTION
Virtual Machine Size	Small GPU (Compute)
VM image	Windows 10 Pro. Version 2004

External resource configuration

Some classes require files, such as large data files, to be stored externally. See use external file storage in Azure Lab Services for options and setup instructions.

If you choose to have a shared RStudio Server for the students, the server should be set up before the lab is created. For more information on how to set up a shared server, see how to create a lab with a shared resource in Azure Lab Services. For instructions to create an RStudio Server, see Download RStudio Server for Debian & Ubuntu and Accessing RStudio Server Open-Source.

Template configuration

After the template machine is created, start the machine, and connect to it to install R and RStudio Desktop.

Install R

1. Download the latest installer for R for Windows. For a full list of versions available, see the R for Windows download page.
2. Run the installer.
   a. For the **Select Setup Language** prompt, choose the language you want and select **OK**
   b. On the **Information** page of the installer, read the license agreement. Select **Next** to accept agreement and continue on.
   c. On the **Select Destination Location** page, accept the default install location and select **Next**.
   d. On the **Select Components** page, optionally uncheck **32-bit files** option. For more information about running both 32-bit and 62-bit versions of R, see Can both 32-bit and 64-bit R be installed on the same machine? frequently asked question.
   e. On the **Startup options** page, leave startup options as **No (accept defaults)**. If you want the R graphical user interface (GUI) to use separate windows (SDI) or plain text help, choose **Yes (customize startup)** radio button and change startup options in the following to pages of the wizard.
   f. On the **Select Start Menu Folder** page, select **Next**.
   g. On the **Select Additional Tasks** page, optionally select **Create a desktop shortcut**. Select **Next**.
   h. On the **Installing** page, wait for the installation to finish.
   i. On the **Completing the R for Windows** page, select **Finish**.

You can also execute the installation of R using PowerShell. The code example shows how to install R without the 32-bit component and adds a desktop icon for the latest version of R. To see a full list of command-line options for the installer, see setup command-line parameters.

```
#Avoid prompt to setup Internet Explorer if we must parse download page
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Internet Explorer\Main" -Name "DisableFirstRunCustomize" -
Value 2

$outputfile = "R-win.exe"

$result = Invoke-WebRequest "https://cran.r-project.org/bin/windows/base/release.html" -OutFile $outputfile
-PassThru

#Check if we need to parse the result ourselves, to find the latest version of R
if ($result.StatusCode -eq '200' -and $result.Headers["Content-Type"] -eq 'text/html')
{
    $metaTag = $result.ParsedHtml.Head.children | Where-Object {$_.nodeName -eq 'META'}
    if ($metaTag.content  -match "R-\d+\.\d+\.\d+-win.exe"){
        $outputfile = $Matches.0

        #Download latest version
        Invoke-WebRequest "https://cran.r-project.org/bin/windows/base/$outputfile" -OutFile $outputfile
    }else{
        Write-Error "Unable to find latest version of R installer.  Go to https://cran.r-
project.org/bin/windows/base/release.html to download manually."
    }
}

#Install Silently
$installPath = Get-Item -Path $outputfile
Start-Process -FilePath $installPath.FullName -ArgumentList "/VERYSILENT /LOG=r-install.log /NORESTART
/COMPONENTS=""main,x64,translations"" /MERGETASKS=""desktopicon"" /LANG=""en""" -NoNewWindow -Wait
```

Install RStudio

Now that we have R installed locally, we can install the RStudio IDE. We'll install the free version of RStudio Desktop. For all available versions, see RStudio downloads.

1. Download the installer for R Studio for Windows 10. The installer file will be in the format `rstudio-{version}.exe`.
2. Run the RStudio installer.
   a. On the **Welcome to RStudio Setup** page of the **RStudio Setup** wizard, select **Next**.
   b. On the **Choose Install Location** page, select **Next**.
   c. On the **Choose Start Menu Folder** page, select **Install**.
   d. On the **Installing** page, wait for the installation to finish.
   e. On the **Completing RStudio Setup** page, select **Finish**.

To execute the RStudio installation steps using PowerShell, run the following commands. See RStudio downloads to verify the RStudio version is available before executing the commands.

```
$rstudiover="1.4.1717"
$outputfile = "RStudio-$rstudiover.exe"

#Download installer executable
Invoke-WebRequest "https://download1.rstudio.org/desktop/windows/RStudio-$rstudiover.exe" -OutFile
$outputfile

#Install RStudio silently
$installPath = Get-Item -Path $outputfile
Start-Process -FilePath $installPath.FullName -ArgumentList "/S" -NoNewWindow -Wait
```

CRAN packages

Use the `install.packages(“package name”)` command in an R interactive session as shown in quick list of useful R packages article. Alternately, use Tools -> Install Packages menu item in RStudio.

If you need help with finding a package, see a list of packages by task or alphabetic list of packages.

Cost

Let's cover an example cost estimate for this class. Suppose you have a class of 25 students. Each student has 20 hours of scheduled class time. Another 10 quota hours for homework or assignments outside of scheduled class time is given to each student. The virtual machine size we chose was **Small GPU (Compute)**, which is 139 lab units.

25 students × (20 scheduled hours + 10 quota hours) × 139 Lab Units × 0.01 USD per hour = 1042.5 USD

> **IMPORTANT**
>
> The cost estimate is for example purposes only. For current pricing information, see Azure Lab Services pricing.

Next steps

The template image can now be published to the lab. See publish the template VM for further instructions.

As you set up your lab, see the following articles:

- Add users
- Set quotas
- Set a schedule
- Email registration links to students

Set up a lab to teach shell scripting on Linux

11/2/2020 • 3 minutes to read • Edit Online

This article shows you how to set up a lab to teach shell scripting on Linux. Scripting is a useful part of system administration that allows administrators to avoid repetitive tasks. In this sample scenario, the class covers traditional bash scripts and enhanced scripts. Enhanced scripts are scripts that combine bash commands and Ruby. This approach allows Ruby to pass data around and bash commands to interact with the shell.

Students taking these scripting classes get a Linux virtual machine to learn the basics of Linux, and also get familiar with the bash shell scripting. The Linux virtual machine comes with remote desktop access enabled and with gedit and Visual Studio Code text editors installed.

Lab configuration

To set up this lab, you need an Azure subscription to get started. If you don't have an Azure subscription, create a free account before you begin. Once you have an Azure subscription, you can either create a new lab account in Azure Lab Services or use an existing lab account. See the following tutorial for creating a new lab account: Tutorial to Setup a Lab Account.

After you create the lab account, enable following settings in the lab account:

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace images	Enable the Ubuntu Server 18.04 LTS image for use within your lab account. For more information, see Specify Marketplace images available to lab creators.

Follow this tutorial to create a new lab and apply the following settings:

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual machine (VM) size	Small
VM image	Ubuntu Server 18.04 LTS
Enable remote desktop connection	Enable. Enabling this setting will allow teachers and students to connect to their VMs using the remote desktop (RDP). For more information, see Enable remote desktop for Linux virtual machines in a lab in Azure Lab Services.

Install desktop and RDP

The Ubuntu Server 18.04 LTS image doesn't have the RDP remote desktop server installed by default. Follow instructions in the Install and configure Remote Desktop to connect to a Linux VM in Azure article to install the packages that are needed on the template machine to connect via remote desktop protocol (RDP).

Install Ruby

Ruby is an open-source dynamic language that can be combined with bash scripts. This section shows how to use `apt-get` to install the latest version of Ruby.

1. Install updates by running the following commands:

```
sudo apt-get update
sudo apt-get upgrade
```

2. Install Ruby. Ruby is an open-source dynamic language that can be combined with bash scripts.

```
sudo apt-get install ruby-full
```

Install development tools

This section shows you how to install a couple of text editors. Gedit is the default text editor for the gnome desktop environment. It's designed as a general-purpose text editor. Visual Studio Code is a text editor that includes support for debugging and source control integration.

> **NOTE**
> There are several different text editors available. Visual Studio Code and gedit are just two examples.

1. Install gedit.

```
sudo apt-get install gedit
```

2. Install Visual Studio Code. Visual Studio code can be installed using the Snap Store. For alternate installation options, see Visual Studio Code alternate downloads.

```
sudo snap install vscode --classic
```

The template is now updated and has both the programming language and development tools needed to complete the lab. The template image can now be published to the lab. Select the **Publish** button on template page to publish the template to the lab.

Cost

If you would like to estimate the cost of this lab, you can use the following example:

For a class of 25 students with 20 hours of scheduled class time and 10 hours of quota for homework or assignments, the price for the lab would be:

25 students * (20 + 10) hours * 20 Lab Units * 0.01 USD per hour = 150 USD

For more information on the pricing can be found in the following document: Azure Lab Services Pricing.

Conclusion

This article walked you through the steps to create a lab for scripting classes. While this article focused on setting up Ruby scripting tools on Linux machine, same setup can be used for other scripting classes like Python on Linux.

Next steps

Next steps are common to setting up any lab:

- Add users
- Set quota
- Set a schedule
- Email registration links to students.

Set up a lab for engineering classes using SOLIDWORKS

3/5/2021 • 4 minutes to read • Edit Online

SOLIDWORKS provides a 3D computer-aided design (CAD) environment for modeling solid objects and is used in different kinds of engineering fields. With SOLIDWORKS, engineers can easily create, visualize, simulate, and document their designs.

A licensing option commonly used by universities is SOLIDWORKS' Network Licensing. With this option, users share a pool of licenses that are managed by a licensing server. This type of license is sometimes called a "floating" license because you only need to have enough licenses for the number of concurrent users. When a user is done using SOLIDWORKS, their license goes back into the centrally managed license pool so that it can be reused by another user.

In this article, we'll show how to set up a class that uses SOLIDWORKS 2019 and Network Licensing.

License server

SOLIDWORKS Network Licensing requires that you have SolidNetWork License Manager installed and activated on your license server. This license server is typically located in either your on-premise network or a private network within Azure. For more information on how to set up SolidNetWork License Manager on your server, see Installing and Activating a License Manager in the SOLIDWORKS install guide. When setting this up, remember the `port number` and `serial number` that are used since they'll be needed in later steps.

After your license server is set up, you'll need to peer the virtual network (VNet) to your lab account. The network peering must be done before creating the lab so that lab virtual machines can access the license server and the other way around.

> **NOTE**
> You should verify that the appropriate ports are opened on your firewalls to allow communication between the lab virtual machines and the license server. For example, see the instructions on Modifying License Manager Computer Ports for Windows Firewall that show how to add inbound and outbound rules to the license server's firewall. You may also need to open up ports to the lab virtual machines. Follow the steps in the article on firewall settings for labs for more information on this, including how to get the lab's public IP address.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see the tutorial on how to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable the settings described in the table below for the lab account. For more information about how to enable marketplace images, see the article on how to specify Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the Windows 10 Pro image for use within your lab account.

NOTE

In addition to Windows 10, SOLIDWORKS supports other versions of Windows. See SOLIDWORKS system requirements for details.

Lab settings

Use the settings in the table below when setting up a classroom lab. For more information on how to create a classroom lab, see set up a classroom lab tutorial.

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual Machine Size	**Small GPU (Visualization)**. This VM is best suited for remote visualization, streaming, gaming, encoding using frameworks such as OpenGL and DirectX.
Virtual Machine Image	Windows 10 Pro

NOTE

The **Small GPU (Visualization)** virtual machine size is configured to enable a high-performing graphics experience. For more information about this virtual machine size, see the article on how to set up a lab with GPUs.

WARNING

Don't forget to peer the virtual network for the lab account to the virtual network for the license server **before** creating the lab.

Template virtual machine configuration

The steps in this section show how to set up your template virtual machine by downloading the SOLIDWORKS installation files and installing the client software:

1. Start the template virtual machine and connect to the machine using RDP.
2. Download the installation files for SOLIDWORKS client software. You have two options for downloading:
   - Download from SOLIDWORKS customer portal.
   - Download from a directory on a server. If you used this option, you need to ensure that the server is accessible from the template virtual machine. For example, this server may be located in the same virtual network that is peered with your lab account.

   For details, see Installation on Individual Computers in the SOLIDWORKS in SOLIDWORKS install guide.
3. Once the installation files are downloaded, install the client software using SOLIDWORKS Installation Manager. See details on Installing a License Client in SOLIDWORKS install guide.

> **NOTE**
>
> In the **Add Server** dialog box, you will be prompted for the **port number** used for your license server and the name or IP address of the license server.

Cost

Let's cover a possible cost estimate for this class. This estimate doesn't include the cost of running the license server. We'll use a class of 25 students. There are 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The virtual machine size we chose was **Small GPU (Visualization)**, which is 160 lab units.

25 students * (20 scheduled hours + 10 quota hours) * 160 Lab Units * 0.01 USD per hour = 1200.00 USD

> **IMPORTANT**
>
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Next steps

Next steps are common to setting up any lab.

- Create and manage a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up a lab to manage and develop with SQL Server

3/28/2021 • 7 minutes to read • Edit Online

This article describes how to set up a lab for a basic SQL Server management and development class in Azure Lab Services. Database concepts are one of the introductory courses taught in most of the Computer Science departments in college. Structured Query Language (SQL) is an international standard. SQL is the standard language for relation database management including adding, accessing, and managing content in a database. It is most noted for its quick processing, proven reliability, ease, and flexibility of use.

In this article, we'll show how to set up a virtual machine template in a lab with Visual Studio 2019, SQL Server Management Studio, and Azure Data Studio. For this lab, we will use one shared SQL Server Database for the entire lab. Azure SQL Database is Platform as a Service (PaaS) Database Engine offering from Azure.

Lab configuration

To set up this lab, you need an Azure subscription and lab account to get started. If you don't have an Azure subscription, create a free account before you begin. Once you get an Azure subscription, you can create a new lab account in Azure Lab Services. For more information about creating a new lab account, see tutorial to setup a lab account. You can also use an existing lab account.

Lab account settings

Enable the settings described in the table below for the lab account. For more information about how to enable marketplace images, see Specify Marketplace images available to lab creators.

LAB ACCOUNT SETTING	INSTRUCTIONS
Marketplace image	Enable the 'Visual Studio 2019 Community (latest release) on Windows 10 Enterprise N (x64)' image for use within your lab account.

Shared resource configuration

To use a shared resource in Lab Services, you first need to create the virtual network and the resources itself. To create the virtual network and connect it to the lab, follow how to create a lab with a shared resource in Azure Lab Services. Remember, any resources external to Lab Services will be billed separately and will not be included in lab cost estimates.

> **WARNING**
> Shared resources for a lab should be setup before the lab is created. If the vnet is not peered to the lab account *before* the lab is created, the lab will not have access to the shared resource.

Now that the networking side of things is handled, lets create a SQL Server Database. We are going to create a single database as it is the quickest deployment option for Azure SQL Database. For other deployment options, create an elastic pool, managed instance, or SQL virtual machine.

1. From the Azure portal menu, choose **Create new resource**.
2. Choose **SQL Database** and click the **Create** button.
3. On the **Basics** tab of the **Create SQL database** form, select the resource group for the database. We will

use *sqldb-rg*.

4. For **Database name**, enter *classlabdb*.
5. Under the **Server** setting, click **Create new** to create a new server to hold the database.
6. On the **New server** flyout, enter the Server name. We will use *classlabdbserver*. The server name must be globally unique.
7. Enter *azureuser* for the **Server admin login**.
8. Enter a memorable password. Password must be at least eight characters in length and contain special characters.
9. Choose region for the **location**. If possible, enter the same location as the lab account and peered vnet to minimize latency.
10. Click **OK** to return to the **Create SQL Database** form.
11. Click **Configure database** link under the **Compute + storage** setting.
12. Modify database settings as needed for the class. You can choose between Provisioned and Serverless options. For this example, we'll use the autoscaled Serverless option with max vCores of 4, min vCores of 1. We'll keep the autopause setting at the minimum of 1 hour. Click **Apply**.
13. Click **Next: Networking** button.
14. On the Networking tab, choose Private endpoint for the **Connectivity method**.
15. Under the **Private endpoints** section, click **Add private endpoint**.
16. On the **Create private endpoint** flyout, choose the same resource group as your virtual network peered to the lab account.
17. For **Location**, choose the same location as the virtual network.
18. For **Name**, enter *labsql-endpt*.
19. Leave the Target subresource set to SqlServer.
20. For **Virtual network**, choose the same virtual network peered to the lab account.
21. For **Subnet**, choose subnet you want the endpoint hosted in. The IP assigned to the endpoint will be from the range assigned to that subnet.
22. Set **Integrate with private DNS** to **No**. For simplicity, we'll use Azure's DNS over own private DNS zone or our own DNS servers.
23. Click **OK**.
24. Click **Next: Additional settings**.
25. For the **Use existing data** setting, choose **Sample**. The data from the AdventureWorksLT database will be used when the database is created.
26. Click **Review + create**.
27. Click **Create**.

Once the SQL Database deployment successfully completes, we can create the lab and install software on the lab template machine.

Lab settings

Use the settings in the table below when setting up a classroom lab. For more information how to create a classroom lab, see set up a classroom lab tutorial.

LAB SETTINGS	VALUE/INSTRUCTIONS
Virtual Machine Size	Medium. This size is best suited for relational databases, in-memory caching, and analytics.
Virtual Machine Image	Visual Studio 2019 Community (latest release) on Windows 10 Enterprise N (x64)

Now that our lab is created, let's modify the template machine with the software we need.

Visual Studio

The image chosen above includes Visual Studio 2019 Community. All workloads and tool sets are already installed on the image. Use the Visual Studio Installer to install any optional tools you may want. Sign in to Visual Studio to unlock the community edition.

Visual Studio includes the **Data storage and processing** tool set, which includes SQL Server Data Tools (SSDT). For more information about SSDT's capabilities, see SQL Server Data Tools overview. To verify connection to the shared SQL Server for the class will be successful, see connect to a database and browse existing objects. If prompted add the template machine IP to the list of allowed computers that can connect to your SQL Server instance.

Visual Studio supports several workloads including **Web & cloud** and **Desktop & mobile** workloads. Both of these workloads support SQL Server as a data source. For more information using ASP.NET Core to SQL Server, see build an ASP.NET Core and SQL Database app in Azure App Service tutorial. Use System.Data.SqlClient library to connect to a SQL Database from a Xamarin app.

Install Azure Data Studio

Azure Data Studio is a multi-database, cross-platform desktop environment for data professionals using the family of on-premises and cloud data platforms on Windows, macOS, and Linux.

1. Download the Azure Data Studio *system* installer for Windows. To find installers for other supported operating systems, go to the Azure Data Studio download page.
2. On the **License Agreement** page, select **I accept the agreement**. Click **Next**.
3. On the **Select Destination Location** page, click **Next**.
4. On the **Select Start Menu Folder** page, click **Next**.
5. On the **Select Additional Tasks** page, check **Create a desktop icon** if you want a desktop icon. Click **Next**.
6. On the **Ready to Install**, click **Next**.
7. Wait for the installer to run. Click **Finish**.

Now that we have Azure Data Studio installed, let's setup the connection to Azure SQL Database.

1. On the **Welcome** page for Azure Data Studio, click the **New Connection** link.
2. In the **Connection Details** box, fill in necessary information.
   - Set **Server** to *classlabdbserver.database.windows.net*
   - Set **User** name to *azureuser*
   - Set **Password** to password used to create the database.
   - Check **Remember Password**.
   - For **Database**, select *classlabdb*.
3. Click **Connect**.

Install SQL Server Management Studio

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. SSMS is a tool used by database administrators to deploy, monitor, and upgrade data infrastructure.

1. Download Sql Server Management Studio. Once downloaded, start the installer.
2. On the **Welcome** page, click **Install**.
3. On the **Setup Completed** page, click **Close**.

4. Start Sql Server Management Studio.
5. On the **Dependency Configuration process** page, click **Close**.

Not that SSMS is installed, you can connect and query a SQL Server. When setting up the connection, use the following values:

- Server type: Database Engine
- Server name: *classlabdbserver.database.windows.net*
- Authentication: SQL Server Authentication
- Login: *azureuser*
- Password: password used to create the database.

Cost estimate

Let's cover a possible cost estimate for this class. Estimate does not include the cost of running the SQL Server. See SQL Database pricing for current details on database pricing.

We'll use a class of 25 students. There are 20 hours of scheduled class time. Also, each student gets 10 hours quota for homework or assignments outside scheduled class time. The virtual machine size we chose was medium, which is 42 lab units.

Here is an example of a possible cost estimate for this class:

25 students * (20 scheduled hours + 10 quota hours) * 0.42 USD per hour = 315.00 USD

> **IMPORTANT**
> Cost estimate is for example purposes only. For current details on pricing, see Azure Lab Services Pricing.

Next steps

Next steps are common to setting up any lab.

- Create, manage, and publish a template
- Add users
- Set quota
- Set a schedule
- Email registration links to students

Azure Lab Services within Microsoft Teams

3/5/2021 • 2 minutes to read • Edit Online

Azure Lab Services can be leveraged within Microsoft Teams using **Azure Lab Services** Teams App. Any team owner with owner/contributor/creator access to the lab accounts will be able to create labs and provision virtual machines to everyone on the team.

This article outlines the benefits of using Azure Lab Services within Teams and provides links to other articles for instructions on how to create and manage labs within Teams.

> **NOTE**
>
> **Azure Lab Services** Teams App can be added only to a team, it cannot be added to individual chats or group chats.

Benefits

Azure Lab Services integration with Microsoft Teams will help educators set up a classroom environment and provide virtual lab environments within the team(class):

- Educators can set up labs so the students can access their VMs from within Teams, without leaving Teams and having to navigate to the Azure Lab Services website.
- Single Sign-on (SSO) from Teams to Azure Lab Services.
- Team and Lab owners need not maintain class rosters in two different systems - Lab user list is autopopulated from the Team membership and a sync is performed every 24 hours automatically.
- After the initial publish of the Template VM, Lab Capacity (that is, number of VMs in the lab) would be automatically adjusted based on the addition/deletion of users from the Team membership.
- Team and Lab Owners will view only the labs related to the team and students will view only the VMs, which are provisioned for the specific team.
- Users will be auto registered to the lab and VMs will be automatically assigned upon the first login after the lab is published. Educators don't need to send invitations and students don't need to register for the lab separately.

Next steps

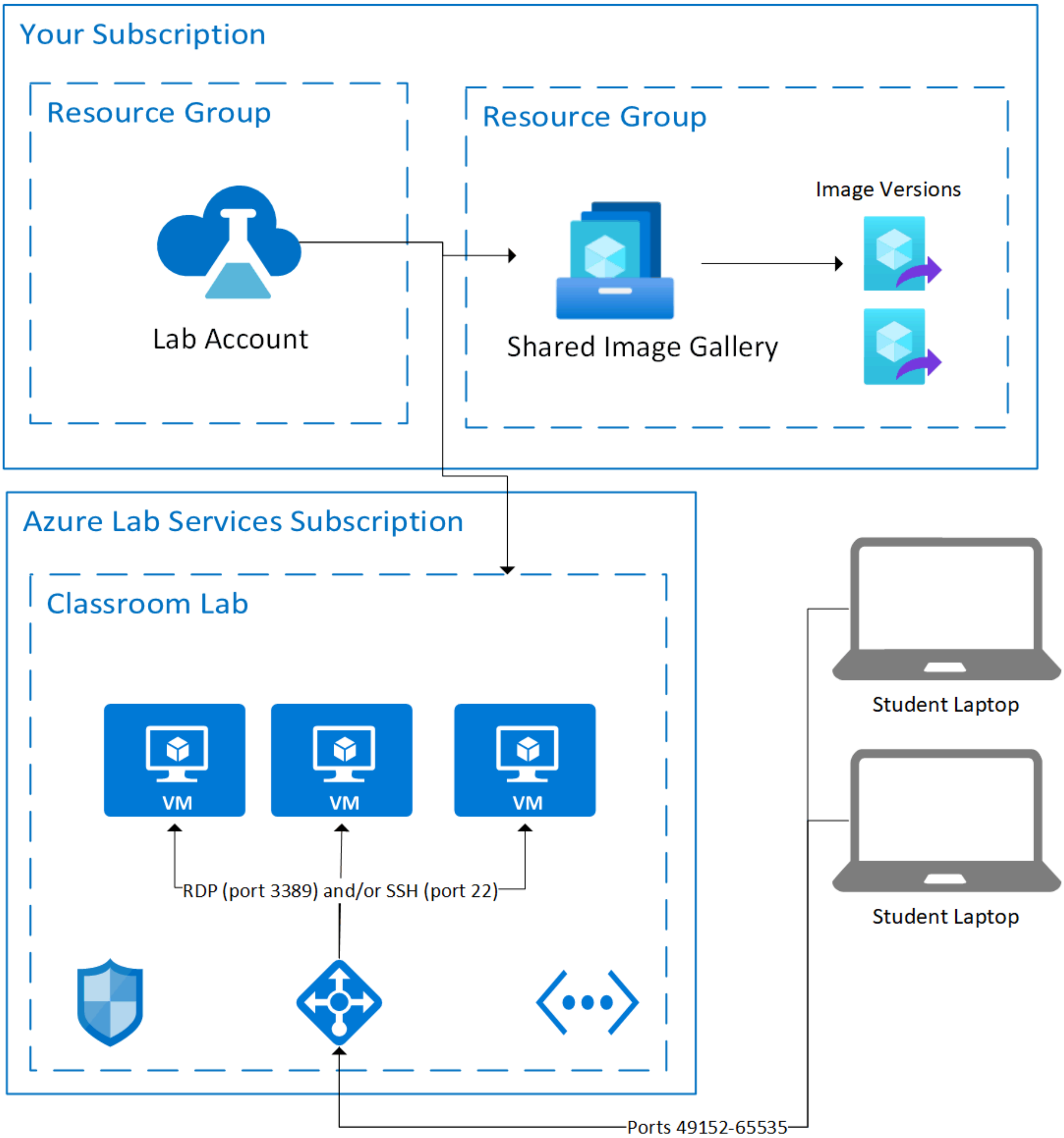
See the following articles:

- Get started and create a lab within Teams
- Manage lab user lists within Teams
- Manage lab's VM pool within Teams
- Create and manage lab schedules within Teams
- Access a VM within Teams – Student view

Azure Lab Services - Administrator guide

8/12/2021 • 19 minutes to read • Edit Online

Information technology (IT) administrators who manage a university's cloud resources are ordinarily responsible for setting up the lab account for their school. After they've set up a lab account, administrators or educators create the labs that are contained within the account. This article provides a high-level overview of the Azure resources that are involved and the guidance for creating them.



- Labs are hosted within an Azure subscription that's owned by Azure Lab Services.
- Lab accounts, a shared image gallery, and image versions are hosted within your subscription.
- You can have your lab account and the shared image gallery in the same resource group. In this diagram, they are in different resource groups.

For more information about the architecture, see Labs architecture fundamentals.

Subscription

Your university might have one or more Azure subscriptions. You use subscriptions to manage billing and security for all Azure resources and services that are used within it, including lab accounts.

The relationship between a lab account and its subscription is important because:

- Billing is reported through the subscription that contains the lab account.
- You can grant users in the subscription's Azure Active Directory (Azure AD) tenant access to Azure Lab Services. You can add a user as a lab account Owner or Contributor, or as a Lab Creator or lab Owner.

Labs and their virtual machines (VMs) are managed and hosted for you within a subscription that's owned by Azure Lab Services.

Resource group

A subscription contains one or more resource groups. Resource groups are used to create logical groupings of Azure resources that are used together within the same solution.

When you create a lab account, you must configure the resource group that contains the lab account.

A resource group is also required when you create a shared image gallery. You can place your lab account and shared image gallery in the same resource group or in two separate resource groups. You might want to take this second approach if you plan to share the image gallery across various solutions.

When you create a lab account, you can automatically create and attach a shared image gallery at the same time. This option results in the lab account and the shared image gallery being created in separate resource groups. You'll see this behavior when you follow the steps that are described in the Configure shared image gallery at the time of lab account creation tutorial. The image at the beginning of this article uses this configuration.

We recommend that you invest time up front to plan the structure of your resource groups, because it's *not* possible to change a lab account or shared image gallery resource group once it's created. If you need to change the resource group for these resources, you'll need to delete and re-create your lab account or shared image gallery.

Lab account

A lab account serves as a container for one or more labs. When you're getting started with Azure Lab Services, it's most common to have a single lab account. As your lab usage scales up, you can choose to create more lab accounts later.

The following list highlights scenarios where more than one lab account might be beneficial:

- **Manage different policy requirements across labs**

  When you set up a lab account, you set policies that apply to *all* labs under the lab account, such as:

  - The Azure virtual network with shared resources that the lab can access. For example, you might have a set of labs that need access to a shared data set within a virtual network.
  - The virtual machine images that the labs can use to create VMs. For example, you might have a set of labs that need access to the Data Science VM for Linux Azure Marketplace image.

  If each of your labs has unique policy requirements, it might be beneficial to create separate lab accounts for managing each lab separately.

- **Assign a separate budget to each lab account**

  Instead of reporting all lab costs through a single lab account, you might need a more clearly apportioned budget. For example, you can create separate lab accounts for your university's Math department, Computer Science department, and so forth, to distribute the budget across departments. You can then view the cost for each individual lab account by using Azure Cost Management.

- **Isolate pilot labs from active or production labs**

  You might have cases where you want to pilot policy changes for a lab account without potentially affecting your active or production labs. In this type of scenario, creating a separate lab account for piloting purposes allows you to isolate changes.

Lab

A lab contains VMs that are each assigned to a single student. In general, you can expect to:

- Have one lab for each class.
- Create a new set of labs for each semester, quarter, or other academic system you're using. For classes that need to use the same image, you should use a shared image gallery. This way, you can reuse images across labs and academic periods.

When you're determining how to structure your labs, consider the following points:

- **All VMs within a lab are deployed with the same image that's published**

  As a result, if you have a class that requires different lab images to be published at the same time, a separate lab must be created for each image.

- **The usage quota is set at the lab level and applies to all users within the lab**

  To set different quotas for users, you must create separate labs. However, it's possible to add more hours to specific users after you've set the quota.

- **The startup or shutdown schedule is set at the lab level and applies to all VMs within the lab**

  Similar to quota setting, if you need to set different schedules for users, you need to create a separate lab for each schedule.

By default, each lab has its own virtual network. If you have virtual network peering enabled, each lab will have its own subnet peered with the specified virtual network.

Shared image gallery

A shared image gallery is attached to a lab account and serves as a central repository for storing images. An image is saved in the gallery when an educator chooses to export it from a lab's template VM. Each time an educator makes changes to the template VM and exports it, new image definitions and\or versions are created in the gallery.

Instructors can publish an image version from the shared image gallery when they create a new lab. Although the gallery stores multiple versions of an image, educators can select only the most recent version during lab creation. The most recent version is chosen based on the highest value of MajorVersion, then MinorVersion, then Patch. For more information about versioning, see Image versions.

The shared image gallery service is an optional resource that you might not need immediately if you're starting with only a few labs. However, shared image gallery offers many benefits that are helpful as you scale up to additional labs:

- **You can save and manage versions of a template VM image**

  It's useful to create a custom image or make changes (software, configuration, and so on) to an image from the Azure Marketplace gallery. For example, it's common for educators to require different software or tooling be installed. Rather than requiring students to manually install these prerequisites on their own, different versions of the template VM image can be exported to a shared image gallery. You can then use these image versions when you create new labs.

- **You can share and reuse template VM images across labs**

  You can save and reuse an image so that you don't have to configure it from scratch each time that you create a new lab. For example, if multiple classes need to use the same image, you can create it once and export it to the shared image gallery so that it can be shared across labs.

- **You can upload your own custom images from other environments outside of labs**

  You can upload custom images other environments outside of the context of labs. For example, you can upload images from your own physical lab environment or from an Azure VM into shared image gallery. Once an image is imported into the gallery, you can then use the images to create labs.

To logically group shared images, you can do either of the following:

- Create multiple shared image galleries. Each lab account can connect to only one shared image gallery, so this option also requires you to create multiple lab accounts.
- Use a single shared image gallery that's shared by multiple lab accounts. In this case, each lab account can enable only images that are applicable to the labs in that account.

Naming

As you get started with Azure Lab Services, we recommend that you establish naming conventions for resource groups, lab accounts, labs, and the shared image gallery. Although the naming conventions that you establish will be unique to the needs of your organization, the following table provides general guidelines:

RESOURCE TYPE	ROLE	SUGGESTED PATTERN	EXAMPLES
Resource group	Contains one or more lab accounts and one or more shared image galleries	<organization short name>-<environment>-rg • **Organization short name** identifies the name of the organization that the resource group supports. • **Environment** identifies the environment for the resource, such as *pilot* or *production*. • **Rg** stands for the resource type *resource group*.	contosouniversitylabs-rg contosouniversitylabs-pilot-rg contosouniversitylabs-prod-rg

RESOURCE TYPE	ROLE	SUGGESTED PATTERN	EXAMPLES
Lab account	Contains one or more labs	<organization short name>-<environment>-la • **Organization short name** identifies the name of the organization that the resource group supports. • **Environment** identifies the environment for the resource, such as *pilot* or *production*. • **La** stands for the resource type *lab account*.	contosouniversitylabs-la mathdeptlabs-la sciencedeptlabs-pilot-la sciencedeptlabs-prod-la
Lab	Contains one or more VMs	<class name>-<timeframe>-<educator identifier> • **Class name** identifies the name of the class that the lab supports. • **Timeframe** identifies the timeframe in which the class is offered. **Educator identifier** identifies the educator who owns the lab.	CS1234-fall2019-johndoe CS1234-spring2019-johndoe
Shared image gallery	Contains one or more VM image versions	<organization short name>gallery	contosouniversitylabsgallery

For more information about naming other Azure resources, see Naming conventions for Azure resources.

Regions\locations

When you set up your Azure Lab Services resources, you're required to provide a region or location of the datacenter that will host the resources. The next sections describe how a region or location might affect each resource that's involved with setting up a lab.

Resource group

The region specifies the datacenter where information about a resource group is stored. Azure resources contained within the resource group can be in a different region from that of their parent.

Lab account

A lab account's location indicates the region that a resource exists in.

Lab

The location that a lab exists in varies, depending on the following factors:

- **The lab account is peered with a virtual network**

You can peer a lab account with a virtual network when they're in the same region. When a lab account is peered with a virtual network, labs are automatically created in the same region as both the lab account and the virtual network.

> **NOTE**
>
> When a lab account is peered with a virtual network, the **Allow lab creator to pick lab location** setting is disabled. For more information, see Allow lab creator to pick location for the lab.

- **No virtual network is peered *and* Lab Creators aren't allowed to pick the lab location**

  When *no* virtual network is peered with the lab account and Lab Creators are *not allowed* to pick the lab location, labs are automatically created in a region that has available VM capacity. Specifically, Azure Lab Services looks for availability in regions that are within the same geography as the lab account.

- **No virtual network is peered *and* Lab Creators are allowed to pick the lab location**

  When *no* virtual network is peered and Lab Creators *are allowed* to pick the lab location, the locations that can be selected by the Lab Creator depend on available capacity.

> **NOTE**
>
> To help ensure that a region has sufficient VM capacity, it's important to first request capacity through the lab account when you're creating the lab.

A general rule is to set a resource's region to one that's closest to its users. For labs, this means creating the lab that's closest to your students. For online courses whose students are located all over the world, use your best judgment to create a lab that's centrally located. Or you can split a class into multiple labs according to your students' regions.

VM sizing

When administrators or Lab Creators create a lab, they can choose from a variety of VM sizes, depending on the needs of their classroom. Remember that the size availability depends on the region that your lab account is located in.

In the following table, notice that several of the VM sizes map to more than one VM series. Depending on capacity availability, Lab Services may use any of the VM series that are listed for a VM size. For example, the *Small* VM size maps to using either the Standard_A2_v2 or the Standard_A2 VM series. When you choose *Small* as the VM size for your lab, Lab Services will first attempt to use the *Standard_A2_v2* series. However, when there isn't sufficient capacity available, Lab Services will instead use the *Standard_A2* series. The pricing is determined by the VM size and is the same regardless of which VM series Lab Services uses for that specific size. For more information on pricing for each VM size, read the Lab Services pricing guide.

SIZE	MINIMUM SPECS	SERIES	SUGGESTED USE
Small	• 2 cores • 3.5 gigabytes (GB) RAM	Standard_A2_v2, Standard_A2	Best suited for command line, opening web browser, low-traffic web servers, small to medium databases.
Medium	• 4 cores • 7 GB RAM	Standard_A4_v2, Standard_A3	Best suited for relational databases, in-memory caching, and analytics.

SIZE	MINIMUM SPECS	SERIES	SUGGESTED USE
Medium (nested virtualization)	• 4 cores • 16 GB RAM	Standard_D4s_v3	Best suited for relational databases, in-memory caching, and analytics. This size also supports nested virtualization.
Large	• 8 cores • 16 GB RAM	Standard_A8_v2, Standard_A7	Best suited for applications that need faster CPUs, better local disk performance, large databases, large memory caches.
Large (nested virtualization)	• 8 cores • 32 GB RAM	Standard_D8s_v3	Best suited for applications that need faster CPUs, better local disk performance, large databases, large memory caches. This size also supports nested virtualization.
Small GPU (visualization)	• 6 cores • 56 GB RAM	Standard_NV6	Best suited for remote visualization, streaming, gaming, and encoding using frameworks such as OpenGL and DirectX.
Small GPU (Compute)	• 6 cores • 56 GB RAM	Standard_NC6, Standard_NC6s_v3	Best suited for computer-intensive applications such as AI and deep learning.
Medium GPU (visualization)	• 12 cores • 112 GB RAM	Standard_NV12, Standard_NV12s_v3, Standard_NV12s_v2	Best suited for remote visualization, streaming, gaming, and encoding using frameworks such as OpenGL and DirectX.

Manage identity

By using Azure role-based access control (RBAC) for access to lab accounts and labs, you can assign the following roles:

- Lab account **Owner**

  An administrator who creates a lab account is automatically assigned the lab account Owner role. The Owner role can:

  - Change the lab account settings.
  - Grant other administrators access to the lab account as an Owner or Contributor.
  - Grant educators access to labs as a Creator, Owner, or Contributor.
  - Create and manage all labs in the lab account.
- Lab account **Contributor**

  An administrator who's assigned the Contributor role can:

- Change the lab account settings.
- Create and manage all labs in the lab account.

However, the Contributor *can't* grant other users access to either lab accounts or labs.

- **Lab Creator**

  To create labs within a lab account, an educator must be a member of the Lab Creator role. An educator who creates a lab is automatically added as a lab Owner. For more information, see Add a user to the Lab Creator role.

- Lab **Owner** or **Contributor**

  An educator in either a lab Owner or Contributor role can view and change a lab's settings. The person must also be a member of the lab account Reader role.

  A key difference between the lab Owner and Contributor roles is that only an Owner can grant other users access to manage a lab. A Contributor *can't* grant other users access to manage a lab.

- **Shared image gallery**

  When you attach a shared image gallery to a lab account, lab account Owners and Contributors and Lab Creators, lab Owners, and lab Contributors are automatically granted access to view and save images in the gallery.

When you're assigning roles, it helps to follow these tips:

- Ordinarily, only administrators should be members of a lab account Owner or Contributor role. The lab account might have more than one Owner or Contributor.
- To give educators the ability to create new labs and manage the labs that they create, you need only assign them the Lab Creator role.
- To give educators the ability to manage specific labs, but *not* the ability to create new labs, assign them either the Owner or Contributor role for each lab that they'll manage. For example, you might want to allow a professor and a teaching assistant to co-own a lab. For more information, see Add Owners to a lab.

Content filtering

Your school may need to do content filtering to prevent students from accessing inappropriate websites. For example, to comply with the Children's Internet Protection Act (CIPA). Lab Services doesn't offer built-in support for content filtering.

There are two approaches that schools typically consider for content filtering:

- Configure a firewall to filter content at the network level.
- Install 3rd party software directly on each computer that performs content filtering.

The first approach isn't currently supported by Lab Services. Lab Services hosts each lab's virtual network within a Microsoft-managed Azure subscription. As a result, you don't have access to the underlying virtual network to do content filtering at the network level. For more information on Lab Services' architecture, read the article Architecture Fundamentals.

Instead, we recommend the second approach which is to install 3rd party software on each lab's template VM. There are a few key points to highlight as part of this solution:

- If you plan to use the auto-shutdown settings, you will need to unblock several Azure host names with the 3rd party software. The auto-shutdown settings use a diagnostic extension that must be able to communicate back to Lab Services. Otherwise, the auto-shutdown settings will fail to enable for the lab.
- You may also want to have each student use a non-admin account on their VM so that they can't uninstall the

content filtering software. By default, Lab Services creates an admin account that each student uses to sign into their VM. It is possible to add a non-admin account using a specialized image, but there are some known limitations.

If your school needs to do content filtering, contact us via the Azure Lab Services' forums for more information.

Endpoint management

Many endpoint management tools, such as Microsoft Endpoint Manager, require Windows VMs to have unique machine security identifiers (SIDs). Using SysPrep to create a *generalized* image typically ensures that each Windows machine will have a new, unique machine SID generated when the VM boots from the image.

With Lab Services, even if you use a *generalized* image to create a lab, the template VM and student VMs will all have the same machine SID. The VMs have the same SID because the template VM's image is in a *specialized* state when it's published to create the student VMs.

For example, the Azure Marketplace images are generalized. If you create a lab from the Win 10 marketplace image and publish the template VM, all of the student VMs within a lab will have the same machine SID as the template VM. The machine SIDs can be verified by using a tool such as PsGetSid.

If you plan to use an endpoint management tool or similar software, we recommend that you test it with lab VMs to ensure that it works properly when machine SIDs are the same.

Pricing

Azure Lab Services

To learn about pricing, see Azure Lab Services pricing.

Shared Image Gallery

You also need to consider the pricing for the Shared Image Gallery service if you plan to use shared image galleries for storing and managing image versions.

Creating a shared image gallery and attaching it to your lab account is free. No cost is incurred until you save an image version to the gallery. The pricing for using a shared image gallery is ordinarily fairly negligible, but it's important to understand how it's calculated, because it isn't included in the pricing for Azure Lab Services.

Storage charges

To store image versions, a shared image gallery uses standard hard disk drive (HDD) managed disks by default. We recommend using HDD-managed disks when using shared image gallery with Lab Services. The size of the HDD-managed disk that's used depends on the size of the image version that's being stored. Lab Services supports image and disk sizes up to 128 GB. To learn about pricing, see Managed disks pricing.

Replication and network egress charges

When you save an image version by using a lab template VM, Azure Lab Services first stores it in a source region and then automatically replicates the source image version to one or more target regions.

It's important to note that Azure Lab Services automatically replicates the source image version to all target regions within the geography where the lab is located. For example, if your lab is in the US geography, an image version is replicated to each of the eight regions that exist within the US.

A network egress charge occurs when an image version is replicated from the source region to additional target regions. The amount charged is based on the size of the image version when the image's data is initially transferred outbound from the source region. For pricing details, see Bandwidth pricing details.

Egress charges might be waived for Education Solutions customers. To learn more, contact your account manager.

For more information, see "What data transfer programs exist for academic customers and how do I qualify?" in the FAQ section of the Programs for educational institutions page.

Pricing example

Let's look at an example of the cost of saving a template VM image to a shared image gallery. Assume the following scenarios:

- You have one custom VM image.
- You're saving two versions of the image.
- Your lab is in the US, which has a total of eight regions.
- Each image version is 32 GB in size; as a result, the HDD-managed disk price is $1.54 per month.

The total cost per month is estimated as:

- *Number of images × number of versions × number of replicas × managed disk price = total cost per month*

In this example, the cost is:

- 1 custom image (32 GB) × 2 versions × 8 US regions × $1.54 = $24.64 per month

> **NOTE**
>
> The preceding calculation is for example purposes only. It covers storage costs associated with using Shared Image Gallery and does *not* include egress costs. For actual pricing for storage, see Managed Disks pricing.

Cost management

It's important for lab account administrators to manage costs by routinely deleting unneeded image versions from the gallery.

Don't delete replication to specific regions as a way to reduce the costs, though this option exists in the shared image gallery. Replication changes might have adverse effects on the ability of Azure Lab Services to publish VMs from images saved within a shared image gallery.

Next steps

For more information about setting up and managing labs, see:

- Lab account setup guide
- Lab setup guide
- Cost management for labs
- Use Azure Lab Services in Teams

Lab account setup guide

3/5/2021 • 7 minutes to read • Edit Online

If you're an administrator, before you set up your Azure Lab Services environment, you first need to create a *lab account* within your Azure subscription. A lab account is a container for one or more labs, and it takes only a few minutes to set up.

This guide includes three sections:

- Prerequisites
- Plan your lab account settings
- Set up your lab account

Prerequisites

The following sections outline what you need to do before you can set up a lab account.

Access your Azure subscription

To create a lab account, you need access to an Azure subscription that's already set up for your school. Your school might have one or more subscriptions. You use a subscription to manage billing and security for all your Azure resources and services, including lab accounts. Azure subscriptions are usually managed by your IT department. For more information, see the "Subscription" section of Azure Lab Services - Administrator guide.

Estimate how many VMs and VM sizes you need

It's important to know how many virtual machines (VMs) and VM sizes your school lab requires.

For guidance on structuring your labs and images, see the blog post Moving from a physical lab to Azure Lab Services.

For additional guidance on how to structure labs, see the "Lab" section of Azure Lab Services - Administrator guide.

Understand subscription VM limits and regional VM capacity

After you've estimated the number of VMs and the VM sizes for your labs, you need to:

- Ensure that your Azure subscription's capacity limit allows for the number of VMs and the VM size that you plan to use in your labs.
- Create your lab account within a region that has sufficient available VM capacity.

For more information, see VM subscription limits and regional capacity.

Decide how many lab accounts to create

To get started quickly, create a single lab account within its own resource group. Later, you can create additional lab accounts and resource groups, as needed. For example, you might eventually have one lab account and resource group per department as a way to clearly separate costs.

For more information about lab accounts, resource groups, and separating costs, see:

- The "Resource group" section of Azure Lab Services - Administrator guide
- The "Lab account" section of Azure Lab Services - Administrator guide
- Cost management for Azure Lab Services

Plan your lab account settings

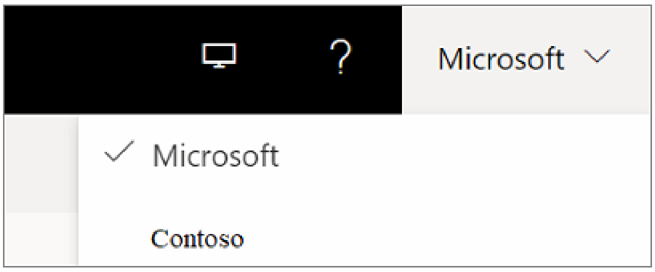
To plan your lab account settings, consider the following questions.

Who should be the Owners and Contributors of the lab account?

Your school's IT administrators ordinarily take on the Owner and Contributor roles for a lab account. These roles are responsible for managing the policies that apply to all the labs in the lab account. The person who creates the lab account is automatically an Owner. You can add additional Owners and Contributors from the Azure Active Directory (Azure AD) tenant that's associated with your subscription.

For more information about the lab account Owner and Contributor roles, see the "Manage identity" section of Azure Lab Services - Administrator guide.

If you have an Administrator or Lab Owner role in two or more Azure AD tenants, you can switch between tenants in the Azure Lab Services portal by selecting the control at the upper right, as shown in the following screenshot:



After you've chosen an Azure AD tenant, go to your lab account by selecting the control at the upper left.

Lab users see only a single list of the VMs that they have access to across Azure AD tenants in Azure Lab Services.

Who will be allowed to create labs?

You may choose to have your IT team or faculty members create labs. To create labs, you then assign these people to the Lab Creator role within the lab account. You ordinarily assign this role from the Azure AD tenant that's associated with your school subscription. Whoever creates a lab is automatically assigned as the Owner of the lab.

For more information about the Lab Creator role, see the "Manage identity" section of Azure Lab Services - Administrator guide.

Who will be allowed to own and manage labs?

You can also choose to have IT and faculty members own\manage labs *without* giving them the ability to create labs. In this case, users from your subscription's Azure AD tenant are assigned either the Owner or Contributor for existing labs.

For more information about the lab Owner and Contributor roles, see the "Manage identity" section of Azure Lab Services - Administrator guide.

Do you want to save images and share them across labs?

Shared Image Gallery is a service that you can use for saving and sharing images. For classes that need to use the same image, Lab Creators can create the image and then export it to a shared image gallery. After an image is exported to the shared image gallery, it can be used to create new labs.

You might want to create your images in your physical environment and then import them to a shared image gallery. For more information, see the blog post Import a custom image to a shared image gallery.

If you decide to use the Shared Image Gallery service, you'll need to create or attach a shared image gallery to your lab account. You can postpone this decision for now, because a shared image gallery can be attached to a lab account at any time.

For more information, see:

- The "Shared image gallery" section of Azure Lab Services - Administrator guide
- The "Pricing" section of Azure Lab Services - Administrator guide

Which images in Azure Marketplace will your labs use?

Azure Marketplace provides hundreds of images that you can enable so that Lab Creators can use them for creating their labs. Some images might include everything that a lab already needs. In other cases, you might use an image as a starting point, and then the Lab Creator can customize it by installing additional applications or tools.

If you don't know which images you need, you can come back later to enable them. The best way to see which images are available is to first create a lab account. This gives you access so that you can review the list of available images and their contents.

For more information, see Specify the Azure Marketplace images that are available to Lab Creators.

Do the lab VMs need access to other Azure or on-premises resources?

When you set up a lab account, you also can peer your lab account with a virtual network. Keep in mind that both your virtual network and the lab account must be located in the same region. To decide whether you need to peer with a virtual network, consider the following scenarios:

- **Access to a license server**

  When you use Azure Marketplace images, the cost of the operating system license is bundled into the pricing for lab services. However, you don't need to provide licenses for the operating system itself. For additional software and applications that are installed, you do need to provide a license, as appropriate. To access a license server:

  - You may choose to connect to an on-premises license server. Connecting to an on-premises license server requires additional setup.
  - Another option, which is faster to set up, is to create a license server that you host on an Azure VM. The Azure VM is located within a virtual network that you peer with your lab account.

- **Access to other on-premises resources such as a file share or database**

  You ordinarily create a virtual network to provide access to on-premises resources by using a site-to-site virtual network gateway. Setting up this type of environment will take additional time.

- **Access to other Azure resources that are located outside a virtual network**

  If you need access to Azure resources that are *not* secured within a virtual network, you can access them through the public internet, without having to do any peering.

  For more information about virtual networks, see:

  - The "Virtual network" section of Architecture fundamentals in Azure Lab Services
  - Connect your lab network with a peer virtual network in Azure Lab Services
  - Create a lab with a shared resource in Azure Lab Services

Set up your lab account

After you've finished planning, you're ready to set up your lab account. You can apply the same steps to setting up Azure Lab Services in Teams.

1. **Create your lab account**. For instructions, see Create a lab account.

For information about naming conventions, see the "Naming" section of Azure Lab Services - Administrator guide.

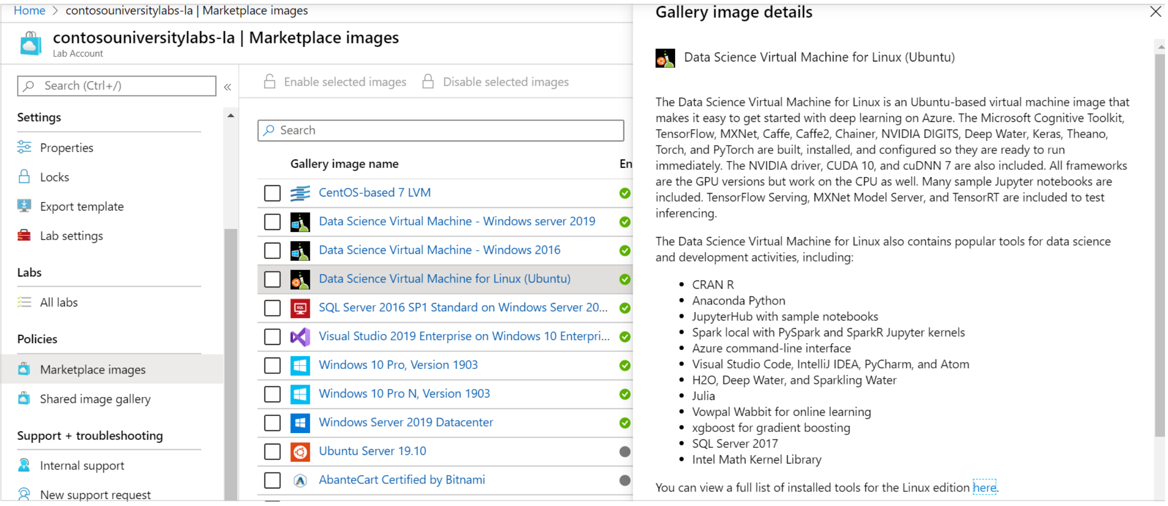
2. **Add users to the Lab Creator role**. For instructions, see Add users to the Lab Creator role.

3. **Connect to a peer virtual network**. For instructions, see Connect your lab network with a peer virtual network.

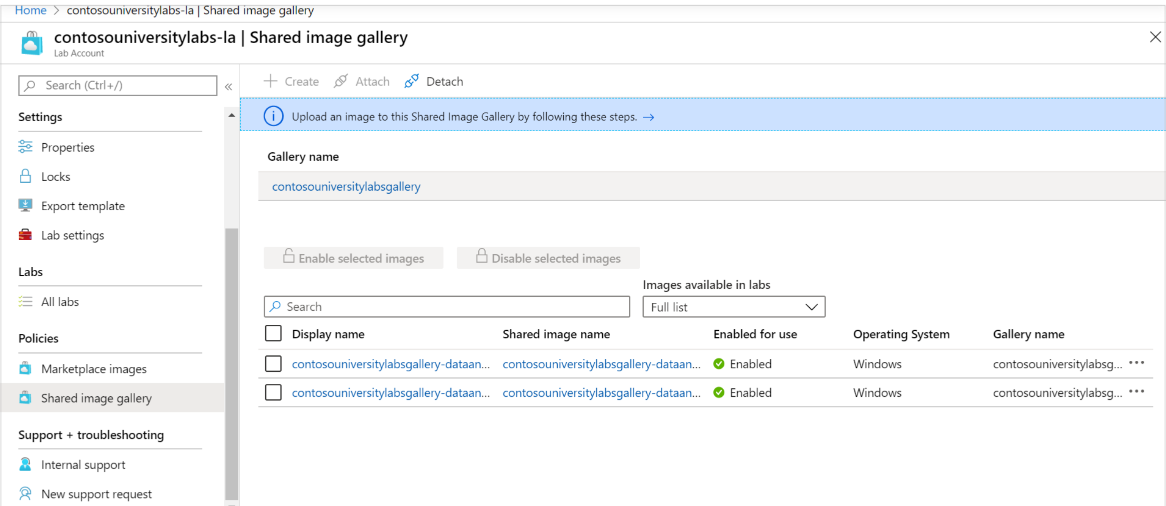
   You might also need to refer to instructions for configuring the lab VMs address range.

4. **Enable and review images**. For instructions, see Specify which Azure Marketplace images are available to Lab Creators.

   To review the contents of each Azure Marketplace image, select the image name. For example, the following screenshot shows the details of the Ubuntu Data Science VM image:



   If a shared image gallery is attached to your lab account, and you want to enable custom images to be shared by Lab Creators, complete similar steps as shown in the following screenshot:



Next steps

For more information about setting up and managing labs, see:

- Manage lab accounts
- Lab setup guide

Lab setup guide

3/5/2021 • 6 minutes to read • Edit Online

In this guide, you'll learn how to create a lab for students at your school.

The process for publishing a lab to your students can take up to several hours. The amount of setup time depends on the number of virtual machines (VMs) that you want to create in your lab. Allow at least a day to ensure that the lab is working properly and to allow enough time to publish your students' VMs.

Understand the lab requirements of your class

Before you set up a new lab, you should consider the following questions.

What software requirements does the class have?

Refer to your class's learning objectives as you decide which operating system, applications, and tools you need to install on the lab VMs. To set up lab VMs, you have three options:

- **Use an Azure Marketplace image**: Azure Marketplace provides hundreds of images that you can use when you're creating a lab. For some classes, one of these images might already contain everything that you need for your class.
- **Create a new custom image**: You can create your own custom image by using an Azure Marketplace image as a starting point. You can then customize it by installing additional software and making configuration changes.
- **Use an existing custom image**: You can reuse custom images that you previously created, or images that were created by other administrators or faculty at your school. To use custom images, your administrators need to set up a Shared Image Gallery. A Shared Image Gallery is a repository that is used for saving custom images.

> **NOTE**
>
> Your administrators are responsible for enabling Azure Marketplace images and custom images so that you can use them. Coordinate with your IT department to ensure that the images that you need are enabled. Custom images that you create are automatically enabled for use within labs that you own.

What hardware requirements does the class have?

You can choose from a variety of compute sizes:

- **Nested virtualization sizes**: Lets you give students access to a VM that can host multiple, nested VMs. For example, you might use this compute size for networking or ethical hacking classes.
- **GPU sizes**: Lets your students use computer-intensive types of applications. For example, this choice is often used with artificial intelligence and machine learning.

For guidance on selecting the appropriate VM size, see:

- VM sizing
- Move from a physical lab to Azure Lab Services

> **NOTE**
>
> Because compute size availability varies by region, fewer sizes might be available to your lab. Generally, you should select the smallest compute size that suits your needs. With Azure Lab Services, you can set up a new lab with a greater compute capacity later, if you need to.

What dependencies does the class have on external Azure or network resources?

Your lab VMs might need access to external resources, such as a database, a file share, or a licensing server. To allow your lab VMs to use external resources, coordinate with your IT administrators.

> **NOTE**
>
> You should consider whether you can reduce your lab's dependency on external resources by providing network resources directly on the VM. For example, to eliminate the need to read data from an external database, you can install the database directly on the VM.

How will you control costs?

Lab Services uses a pay-as-you-go pricing model, which means that you pay only for the time that a lab VM is running. To control costs, use any or all of the following options:

- **Schedule**: Use schedules to automatically control when your lab VMs are started and shut down.
- **Quota**: Use quotas to control the number of hours that students have access to a VM outside of the scheduled hours. When a student is using a VM and reaches a quota, the VM automatically shuts down. The student can't restart the VM unless you increase the quota.
- **Automatic shutdown**: When you enable the auto-shutdown setting, Windows VMs automatically shut down after a student has disconnected from a Remote Desktop Protocol (RDP) session. By default, this setting is disabled.

For more information about controlling costs, see:

- Estimate costs
- Manage costs

How will students save their work?

Each individual student is assigned a VM for the lifetime of the lab. Students can save their work:

- To the VM.
- To an external location, such as OneDrive or GitHub. It's possible to configure OneDrive automatically for students on their lab VMs.

> **NOTE**
>
> To ensure that your students have continued access to their saved work outside of the lab and after the class ends, we recommend that they save their work to an external repository.

How will students connect to their VMs?

For RDP connections to Windows VMs, we recommend that students use the Microsoft Remote Desktop client. The Remote Desktop client supports Mac, Chromebook, and Windows devices.

For Linux VMs, students can use either the Secure Shell (SSH) or RDP protocol. To have students connect by using RDP, you must install and configure the necessary RDP and graphical user interface (GUI) packages.

Will students also use Microsoft Teams?

Azure Lab Services integrates with Microsoft Teams so that faculty members can create and manage their labs in Teams. Similarly, students can access their labs in Teams.

For more information, see Azure Lab Services in Microsoft Teams.

Set up your lab

After you understand the requirements for your class's lab, you're ready to set it up. To learn how, follow the links in this section. Instructions are also provided for setting up labs in Teams.

1. **Create a lab**. See the following tutorials:

   - Create a classroom lab
   - Create a lab in Teams

   > **NOTE**
   >
   > If your class requires nested virtualization, see Enable nested virtualization.

2. **Customize images and publish lab VMs**. To connect to a special VM called the template VM, see:

   - Create and manage a template VM
   - Use a shared image gallery

   > **NOTE**
   >
   > If you're using Windows, also see Set up a Windows template VM. These instructions include steps for setting up OneDrive and Microsoft Office for your students.

3. **Manage VM pool and capacity**. You can easily scale up or down VM capacity, as needed by your class. Keep in mind that increasing VM capacity might take several hours because new VMs are being set up. See the following articles:

   - Set up and manage a VM pool
   - Manage a VM pool in Lab Services in Teams

4. **Add and manage lab users**. To add users to your lab, see:

   - Add users to the lab
   - Send invitations to users
   - Manage Lab Services user lists in Teams

   For information about the types of accounts that students can use, see Student accounts.

5. **Set cost controls**. To set a schedule, establish quotas, and enable automatic shutdown, see the following tutorials:

   - Set a schedule

     > **NOTE**
     >
     > Depending on the operating system you've installed, a VM might take several minutes to start. To ensure that a lab VM is ready for use during your scheduled hours, we recommend that you start it 30 minutes in advance.

   - Set quotas for users and set additional quotas for specific users

- Enable automatic shutdown on disconnect

> **NOTE**
>
> Schedules and quotas don't apply to the template VM, but the automatic shutdown settings do apply.
>
> When you create a lab, the template VM is created but not started. You can start the template VM, connect to it, install any prerequisite software for the lab, and then publish it. When you publish the template VM, it is automatically shut down for you if you haven't done so manually.
>
> Template VMs incur *cost* when they're running, so ensure that the template VM is shut down when you don't need it to be running.

- Create and manage Lab Services schedules in Teams

6. **Use the dashboard**. For instructions, see Use the classroom lab dashboard.

> **NOTE**
>
> The estimated cost shown on the dashboard is the maximum cost that you can expect to incur for student lab usage. For example, you will *not* be charged for unused quota hours by your students. The estimated costs do *not* reflect any charges for using the template VM, the shared image gallery, or when the lab creator starts a user machine.

Next steps

As part of managing your labs, see the following articles:

- Track classroom lab usage
- Access a classroom lab

Create and manage lab accounts

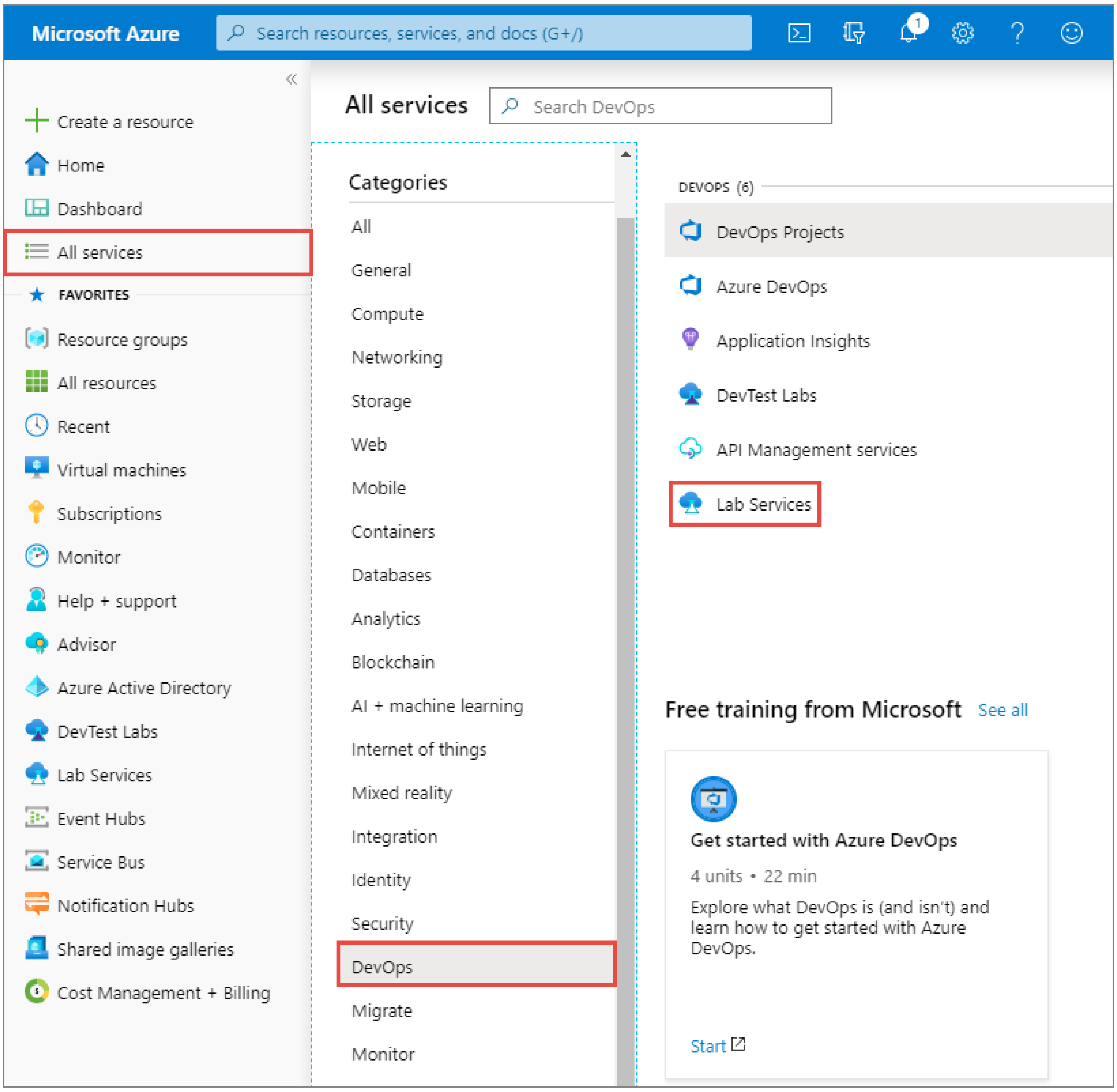
3/5/2021 • 3 minutes to read • Edit Online

In Azure Lab Services, a lab account is a container for managed lab types such as labs. An administrator sets up a lab account with Azure Lab Services and provides access to lab owners who can create labs in the account. This article describes how to create a lab account, view all lab accounts, or delete a lab account.

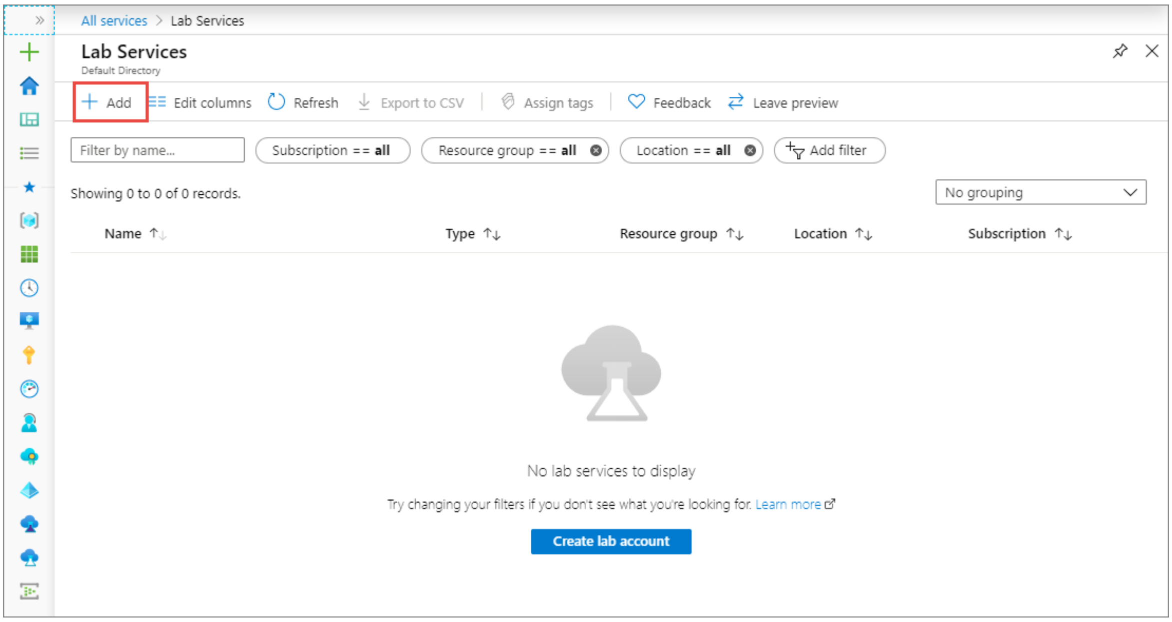
Create a lab account

The following steps illustrate how to use the Azure portal to create a lab account with Azure Lab Services.

1. Sign in to the Azure portal.
2. Select **All Services** on the left menu. Select **Lab Accounts** in the **DevOps** section. If you select star (*) next to **Lab Accounts**, it's added to the **FAVORITES** section on the left menu. From the next time onwards, you select **Lab Accounts** under **FAVORITES**.



3. On the **Lab Accounts** page, select **Add** on the toolbar or **Create lab account** on the page.



4. On the **Basics** tab of the **Create a lab account** page, do the following actions:

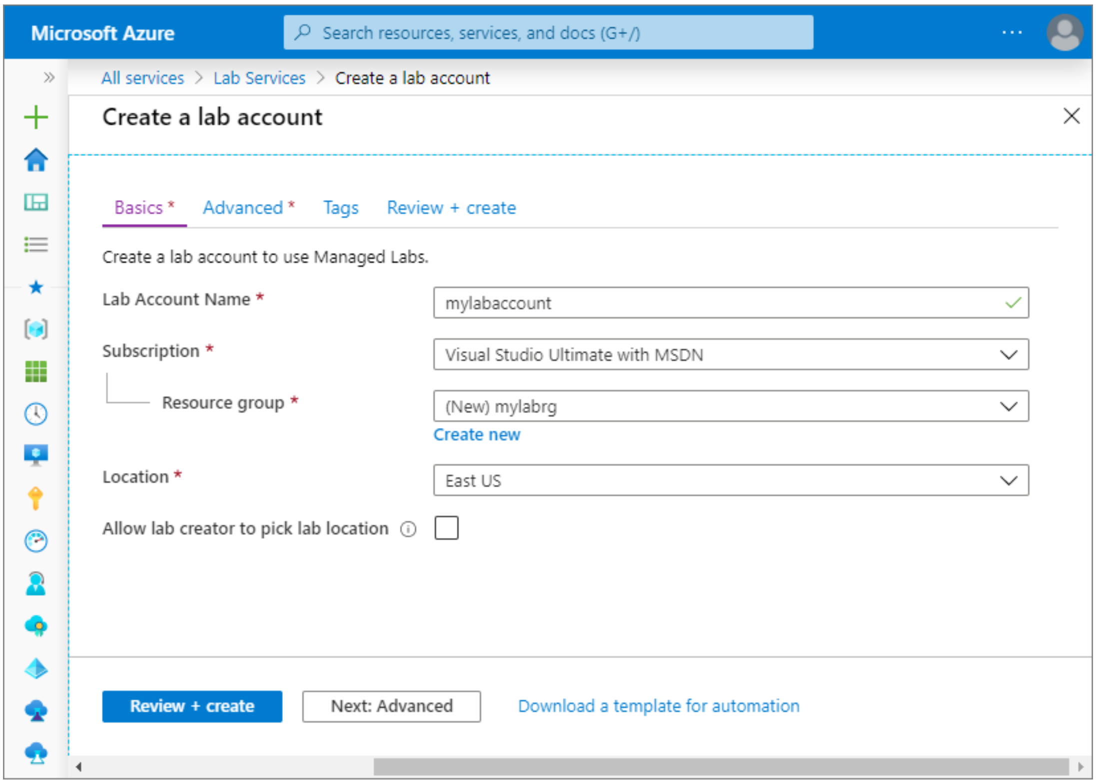
   a. For **Lab account name**, enter a name.

   b. Select the **Azure subscription** in which you want to create the lab account.

   c. For **Resource group**, select **Create new**, and enter a name for the resource group.

   d. For **Location**, select a location/region in which you want the lab account to be created.

   e. For the **Allow lab creator to pick lab location** field, specify whether you want lab creators to be able to select a location for the lab. By default, the option is disabled. When it's disabled, lab creators can't specify a location for the lab they are creating. The labs are created in the closest geographical location to lab account. When it's enabled, a lab creator can select a location at the time of creating a lab. For more information, see Allow lab creator to pick location for the lab.



5. Select **Next: Advanced** at the bottom of the page to navigate to the **Advanced** tab, and then do the following steps:

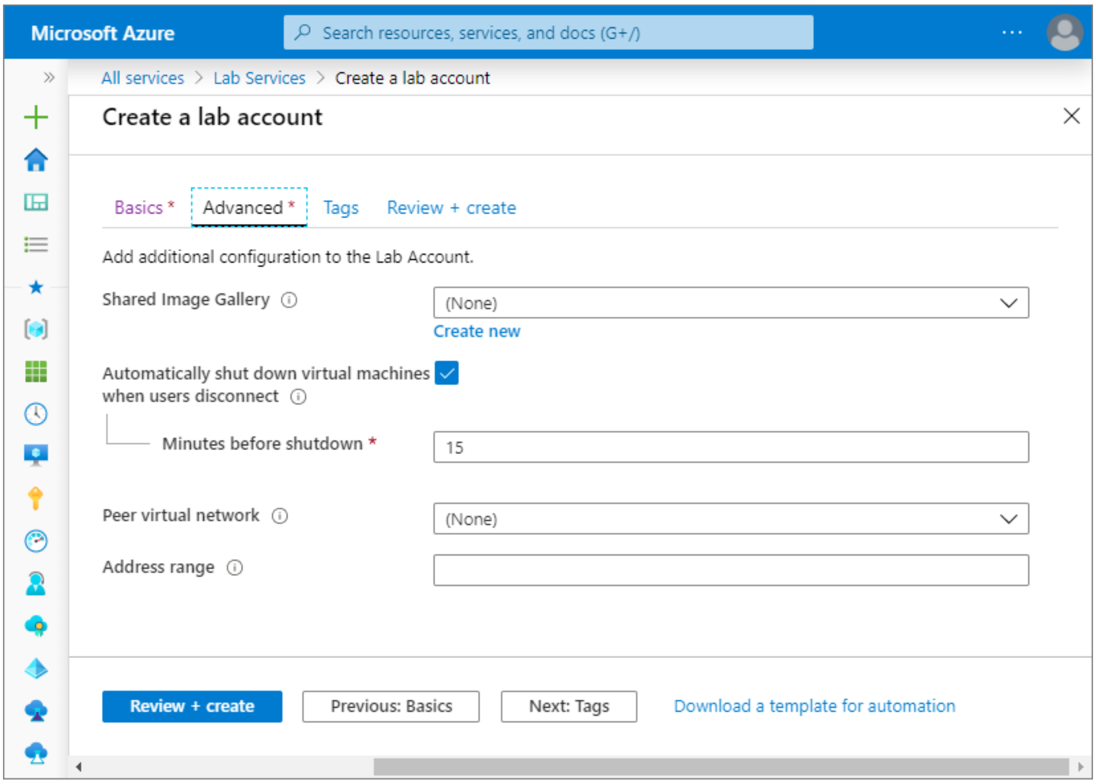
a. Select an existing **shared image gallery** or create one. You can save the template VM in the shared image gallery for it to be reused by others. For detailed information on shared image galleries, see Use a shared image gallery in Azure Lab Services.

b. Specify whether you want to **automatically shut down Windows virtual machines** when users disconnect from them. Specify how long the virtual machines should wait for the user to reconnect before automatically shutting down.

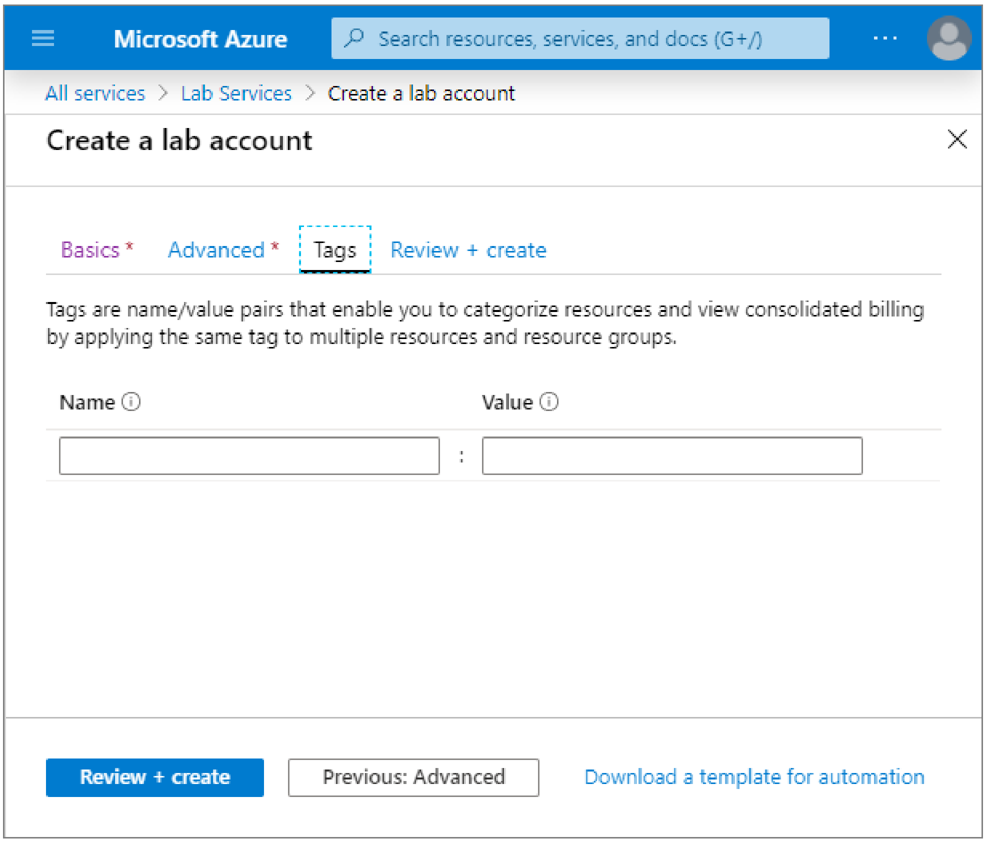
c. For **Peer virtual network**, select a peer virtual network (VNet) for the lab network. Labs created in this account are connected to the selected VNet and have access to the resources in the selected VNet. For more information, see Connect your lab's virtual network with a peer virtual network.

d. Specify an **address range** for VMs in the lab. The address range should be in the classless inter-domain routing (CIDR) notation (example: 10.20.0.0/23). Virtual machines in the lab will be created in this address range. For more information, see Specify an address range for VMs in the lab

> **NOTE**
>
> The **address range** property applies only if a **peer virtual network** is enabled for the lab.

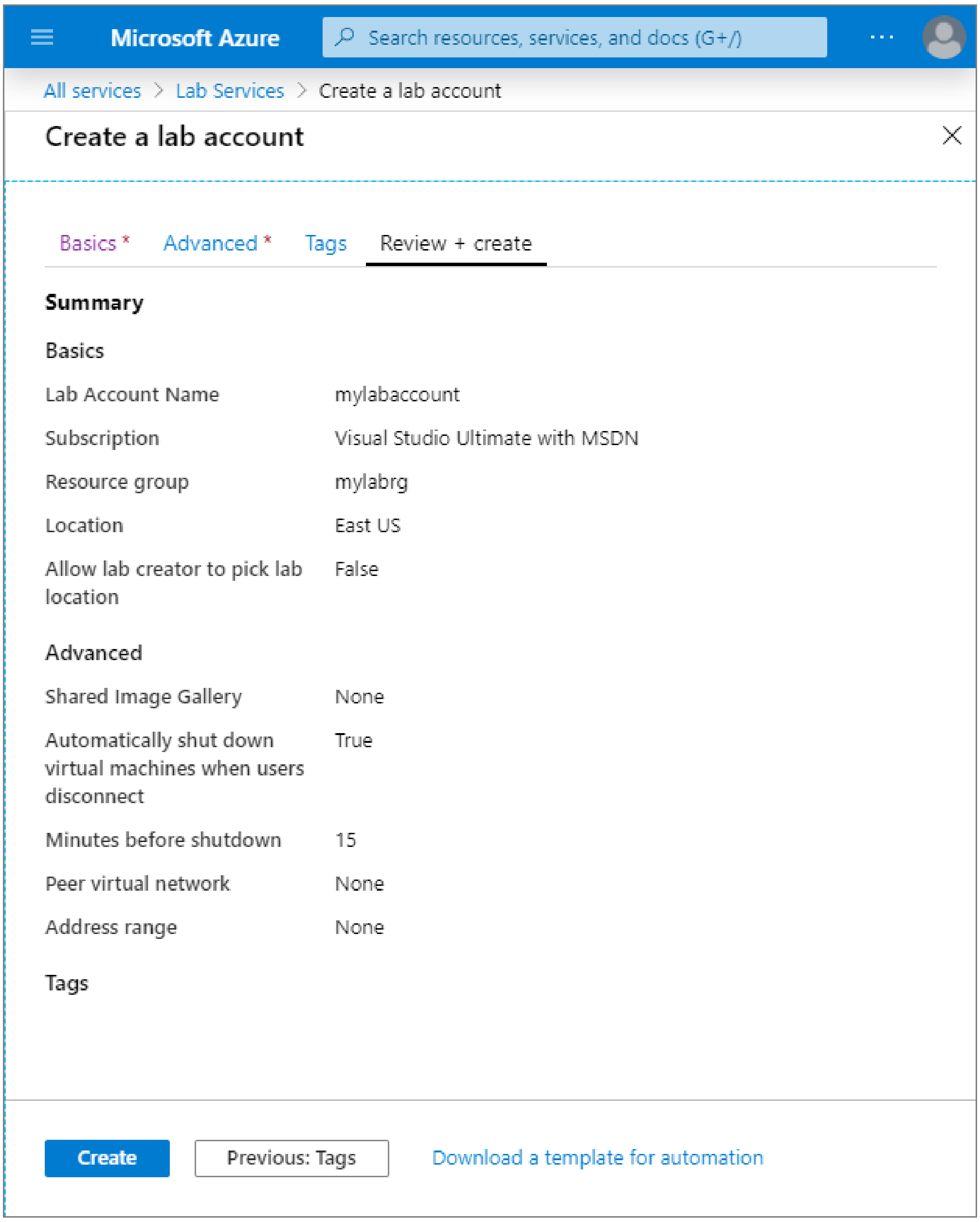


6. Select **Next: Tags** at the bottom of the page to switch to the **Tags** tab. Add any tags you want to associate with the lab account. Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. For more information, see Use tags to organize your Azure resources.



7. Select **Review + create** at the bottom of this page to switch to the **Review + create** tab.

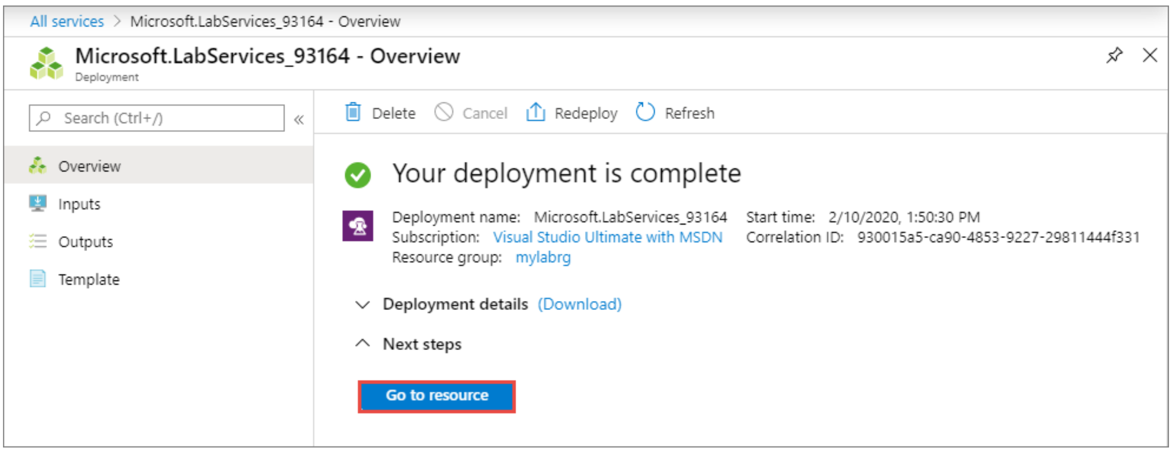
8. Review the summary information on this page, and select **Create**.



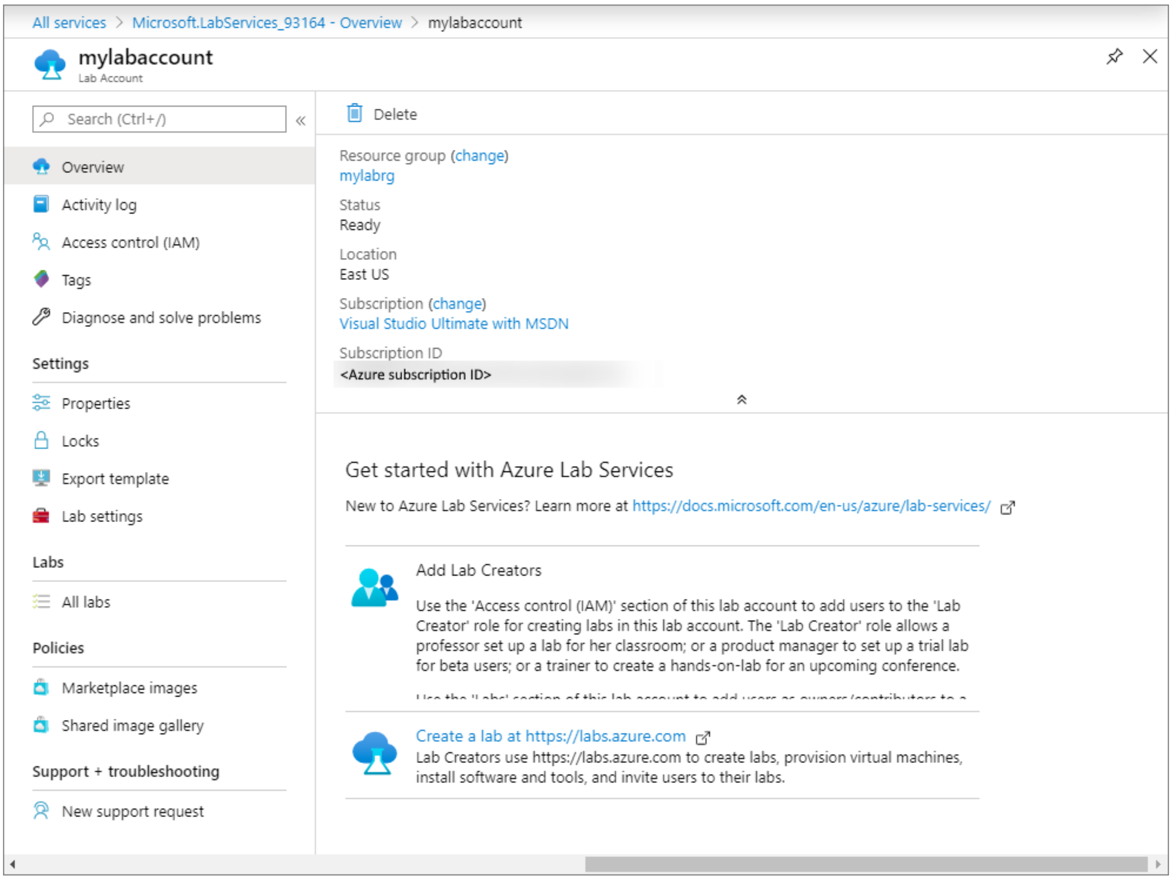
9. Wait until the deployment is complete, expand **Next steps**, and select **Go to resource** as shown in the following image:

   You can also select the **bell icon** on the toolbar (**Notifications**), confirm that the deployment succeeded, and then select **Go to resource**.

   Alternatively, select **Refresh** on the **Lab Accounts** page, and select the lab account you created.

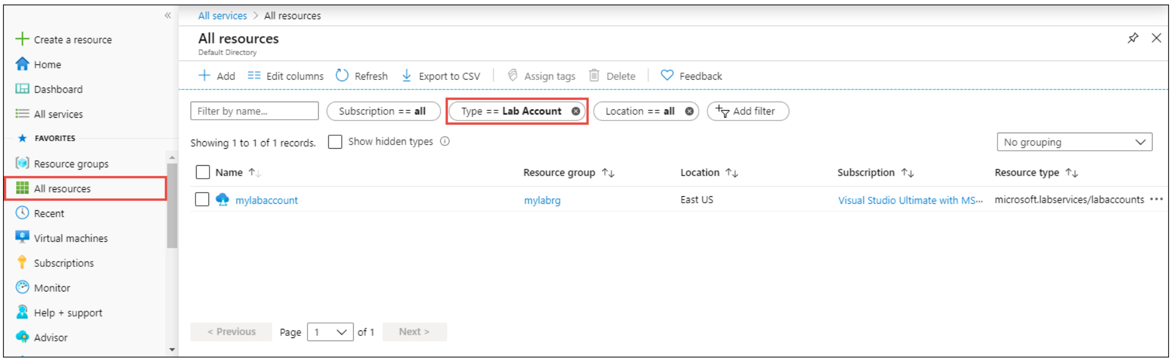


10. You see the following **lab account** page:



View lab accounts

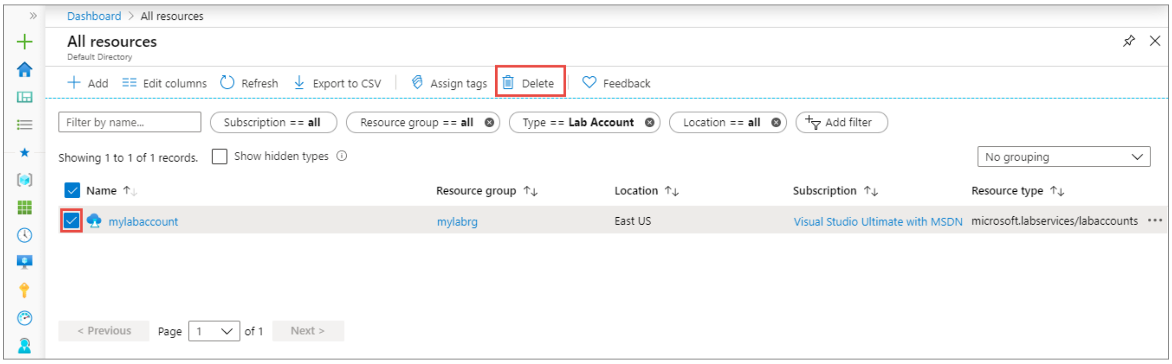
1. Sign in to the Azure portal.
2. Select **All resources** from the menu.
3. Select **Lab Accounts** for the **type**. You can also filter by subscription, resource group, locations, and tags.



Delete a lab account

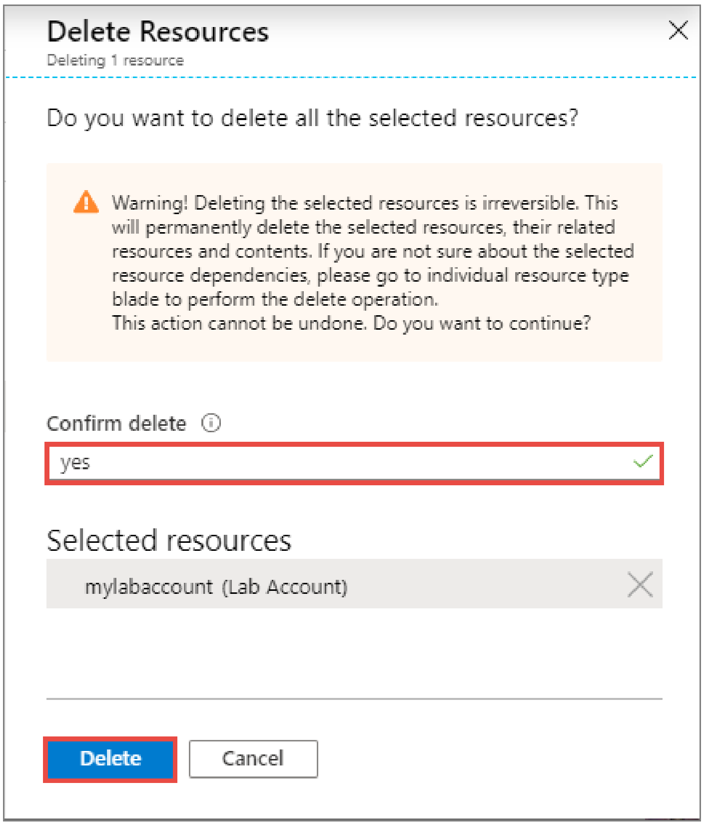
Follow instructions from the previous section that displays lab accounts in a list. Use the following instructions to delete a lab account:

1. Select the **lab account** that you want to delete.
2. Select **Delete** from the toolbar.



3. Type **Yes** for confirmation.

4. Select **Delete**.



NOTE

You can also use the Az.LabServices PowerShell module (preview) to manage lab accounts. For more information, see the Az.LabServices home page on GitHub.

Next steps

See other articles in the **How-to guides -> Create and configure lab accounts (lab account owner)** section of the table-of-content (TOC).

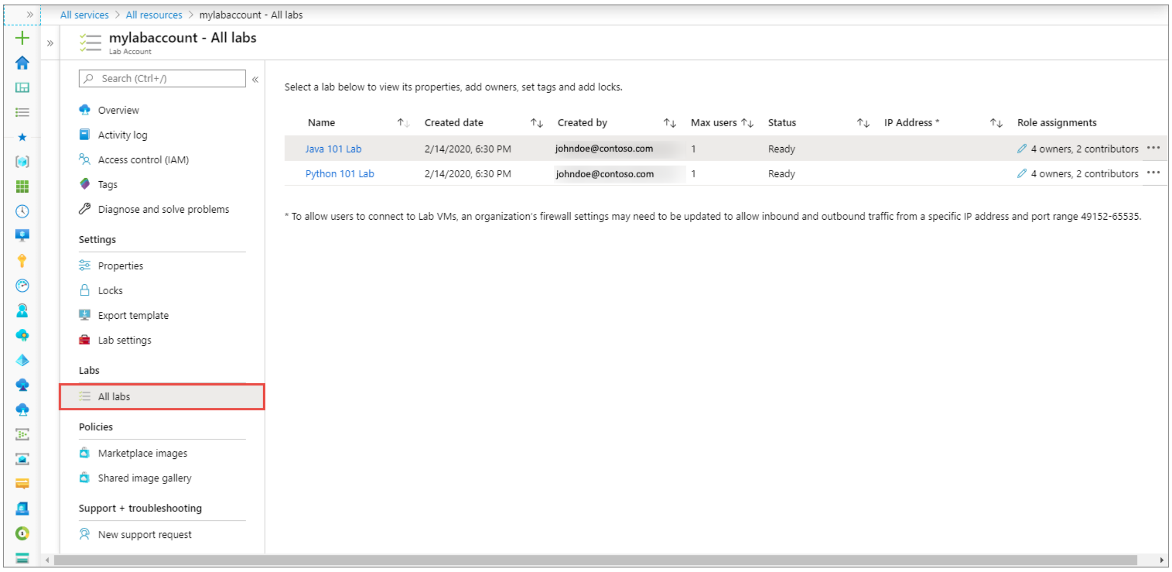
Manage labs in a lab account

11/2/2020 • 2 minutes to read • Edit Online

This article shows you how a lab account owner or administrator can view all the labs in a lab account, and delete a lab in the lab account.

View labs in a lab account

1. On the **Lab Account** page, select **All labs** on the left menu.



2. You see a **list of labs** in the account with the following information:

   a. Name of the lab.

   b. The date on which the lab was created.

   c. Email address of the user who created the lab.

   d. Maximum number of users allowed into the lab.

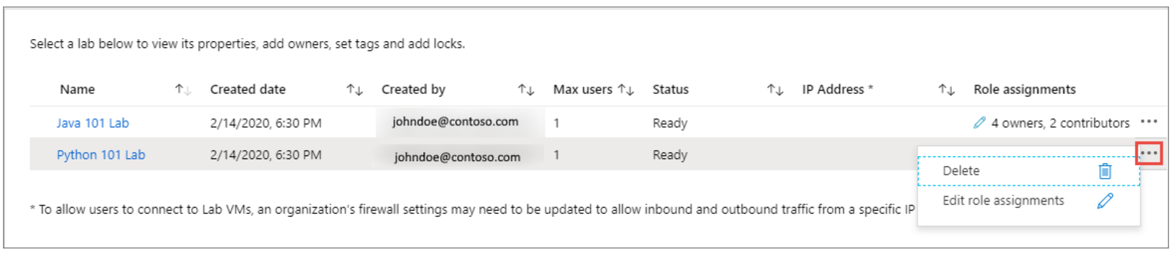
   e. Status of the lab.

   f. Role assignments.

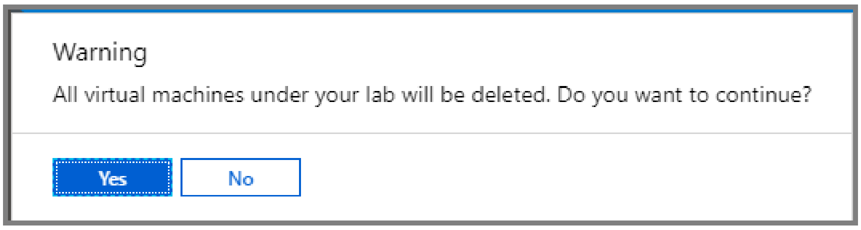
Delete a lab in a lab account

Follow instructions in the previous section to see a list of the labs in the lab account.

1. Select **... (ellipsis)**, and select **Delete**.



2. Select **Yes** on the warning message.



Next steps

See other articles in the **How-to guides** -> **Create and configure lab accounts (lab account owner)** section of the table-of-content (TOC).

Configure automatic shutdown of VMs for a lab account

11/2/2020 • 2 minutes to read • Edit Online

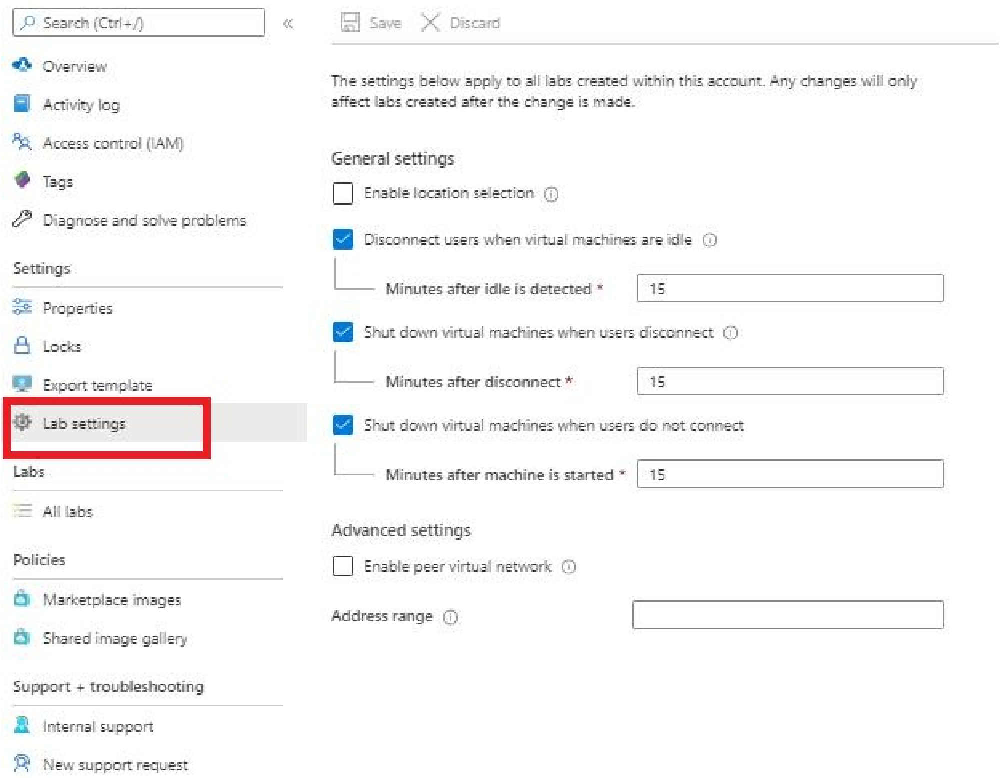
You can enable several auto-shutdown cost control features to proactively prevent additional costs when the virtual machines are not being actively used. The combination of the following three automatic shutdown and disconnect features catches most of the cases where users accidentally leave their virtual machines running:

- Automatically disconnect users from virtual machines that the OS deems idle.
- Automatically shut down virtual machines when users disconnect.
- Automatically shut down virtual machines that are started but users don't connect.

Review more details about the auto-shutdown features in the Maximize cost control with auto-shutdown settings section.

Enable automatic shutdown

1. In the Azure portal navigate to the **Lab Account** page.
2. Select **Labs settings** on the left menu.
3. Select the auto-shutdown setting(s) that is appropriate for your scenario.



The setting(s) apply to all the labs created in the lab account. A lab creator (educator) can override this setting at the lab level. The change to this setting at the lab account will only affect labs that are created after the change is made.

To disable the setting(s), uncheck the checkbox(s) on this page.

Next steps

To learn about how a lab owner can configure or override this setting at the lab level, see Configure automatic shutdown of VMs for a lab

Add lab creators to a lab account in Azure Lab Services

8/20/2021 • 3 minutes to read • Edit Online

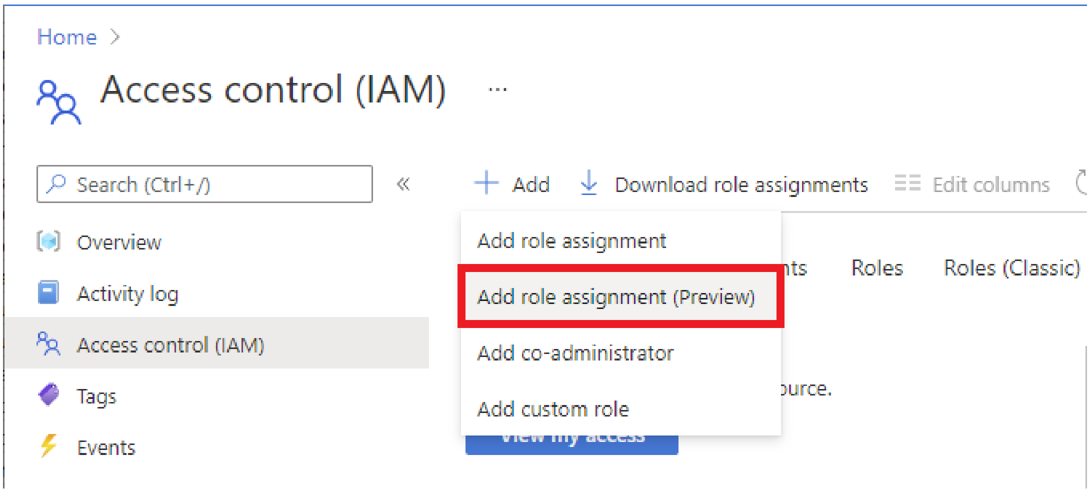
This article shows you how to add users as lab creators to a lab account in Azure Lab Services. These uses then can create labs in the lab account.

Add Microsoft user account to Lab Creator role

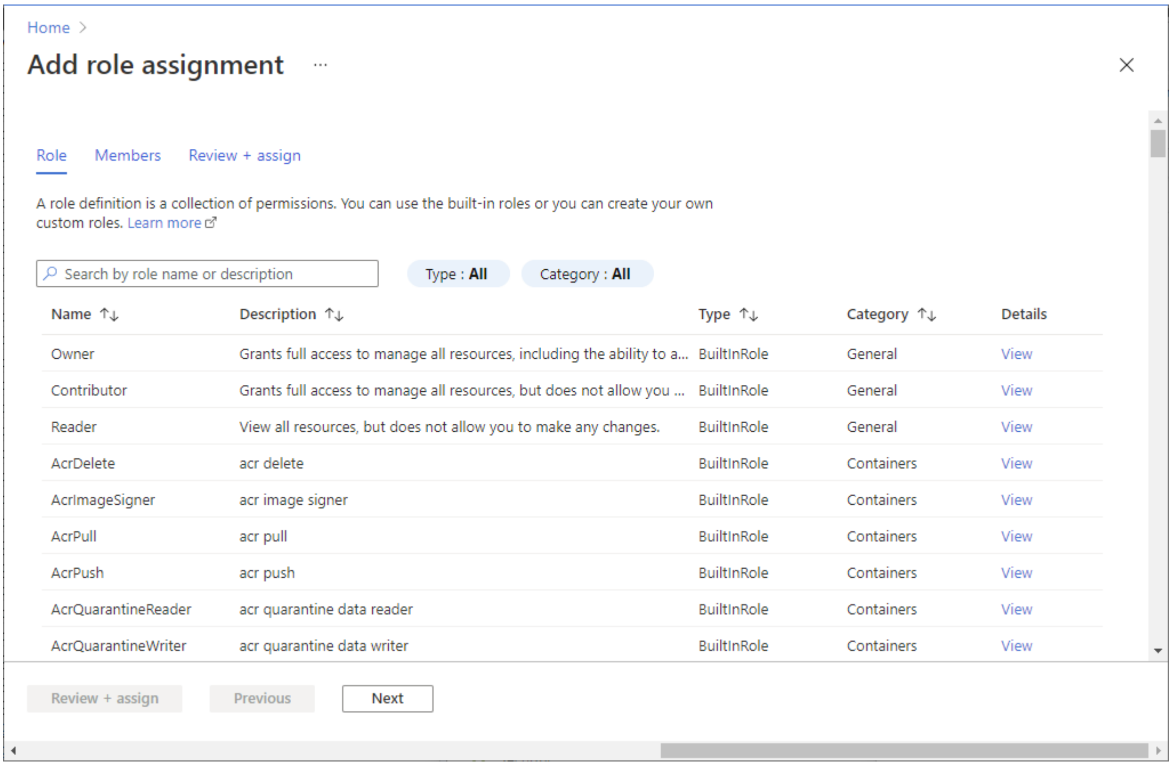
To set up a classroom lab in a lab account, the user must be a member of the **Lab Creator** role in the lab account. The account you used to create the lab account is automatically added to this role. If you are planning to use the same user account to create a classroom lab, you can skip this step. To use another user account to create a classroom lab, do the following steps:

To provide educators the permission to create labs for their classes, add them to the **Lab Creator** role: For detailed steps, see Assign Azure roles using the Azure portal.

1. On the **Lab Account** page, select **Access control (IAM)**
2. Select **Add** > **Add role assignment (Preview)**.



3. On the **Role** tab, select the **Lab Creator** role.



4. On the **Members** tab, select the user you want to add to the Lab Creators role
5. On the **Review + assign** tab, select **Review + assign** to assign the role.

> **NOTE**
>
> If you are adding a non-Microsoft account user as a lab creator, see the Add a non-Microsoft account user as a lab creator section.

Add a non-Microsoft account user as a lab creator

To add a user as a lab creator, you use their email accounts. The following types of email accounts might be used:

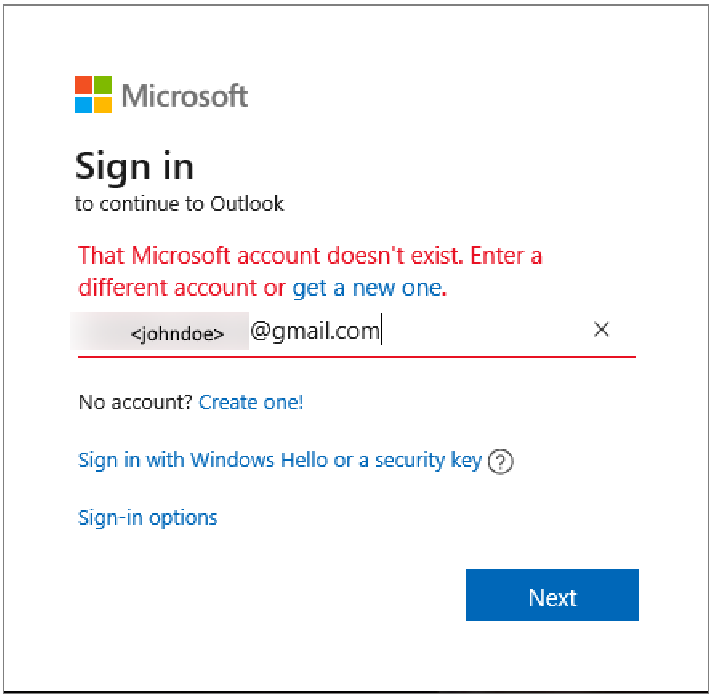
- An email account that's provided by your university's Azure Active Directory (AAD).
- A Microsoft email account, such as `@outlook.com`, `@hotmail.com`, `@msn.com`, or `@live.com`.
- A non-Microsoft email account, such as one provided by Yahoo or Google. However, these types of accounts must be linked with a Microsoft account.
- A GitHub account. This account must be linked with a Microsoft account.

Using a non-Microsoft email account

Lab creators/instructors can use non-Microsoft email accounts to register and sign in to a classroom lab. However, the sign-in to the Lab Services portal requires that instructors first create a Microsoft account that's linked to their non-Microsoft email address.

Many instructors might already have a Microsoft account linked to their non-Microsoft email addresses. For example, instructors already have a Microsoft account if they have used their email address with Microsoft's other products or services, such as Office, Skype, OneDrive, or Windows.

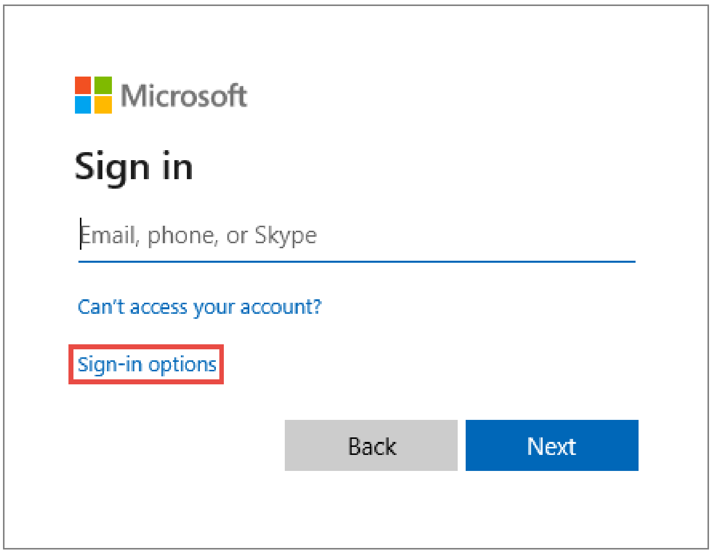
When instructors sign in to the Lab Services portal, they are prompted for their email address and password. If the instructor attempts to sign in with a non-Microsoft account that does not have a Microsoft account linked, the instructor will receive the following error message:



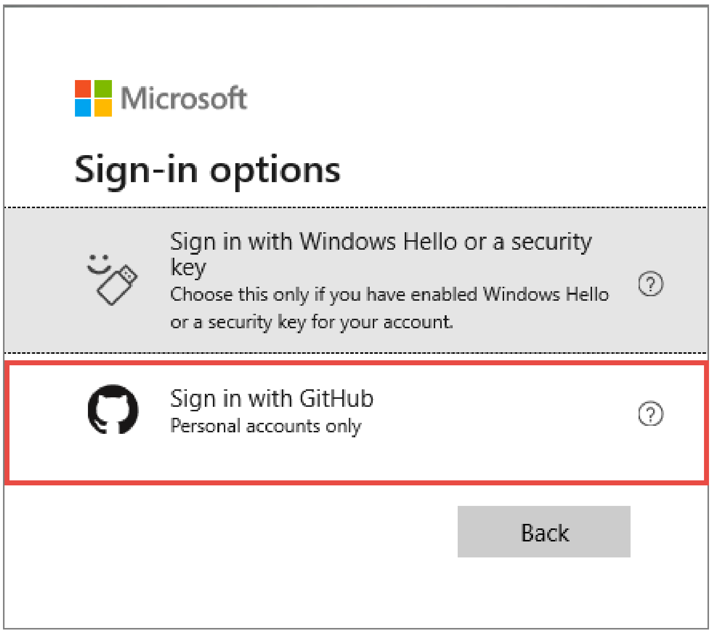
To sign up for a Microsoft account, instructors should go to http://signup.live.com.

Using a GitHub Account

Instructors can also use an existing GitHub account to register and sign in to a classroom lab. If the instructor already has a Microsoft account linked to their GitHub account, then they can sign in and provide their password as shown in the previous section. If they have not yet linked their GitHub account to a Microsoft account, they should select **Sign-in options**:



On the **Sign-in options** page, select **Sign in with GitHub**.



Finally, they are prompted to create a Microsoft account that's linked to their GitHub account. It happens automatically when the instructor selects **Next**. The instructor is then immediately signed in and connected to the classroom lab.

Next steps

See the following articles:

- As a lab owner, create and manage labs
- As a lab owner, set up and publish templates
- As a lab owner, configure and control usage of a lab
- As a lab user, access labs

Allow lab creator to pick location for the lab in Azure Lab Services

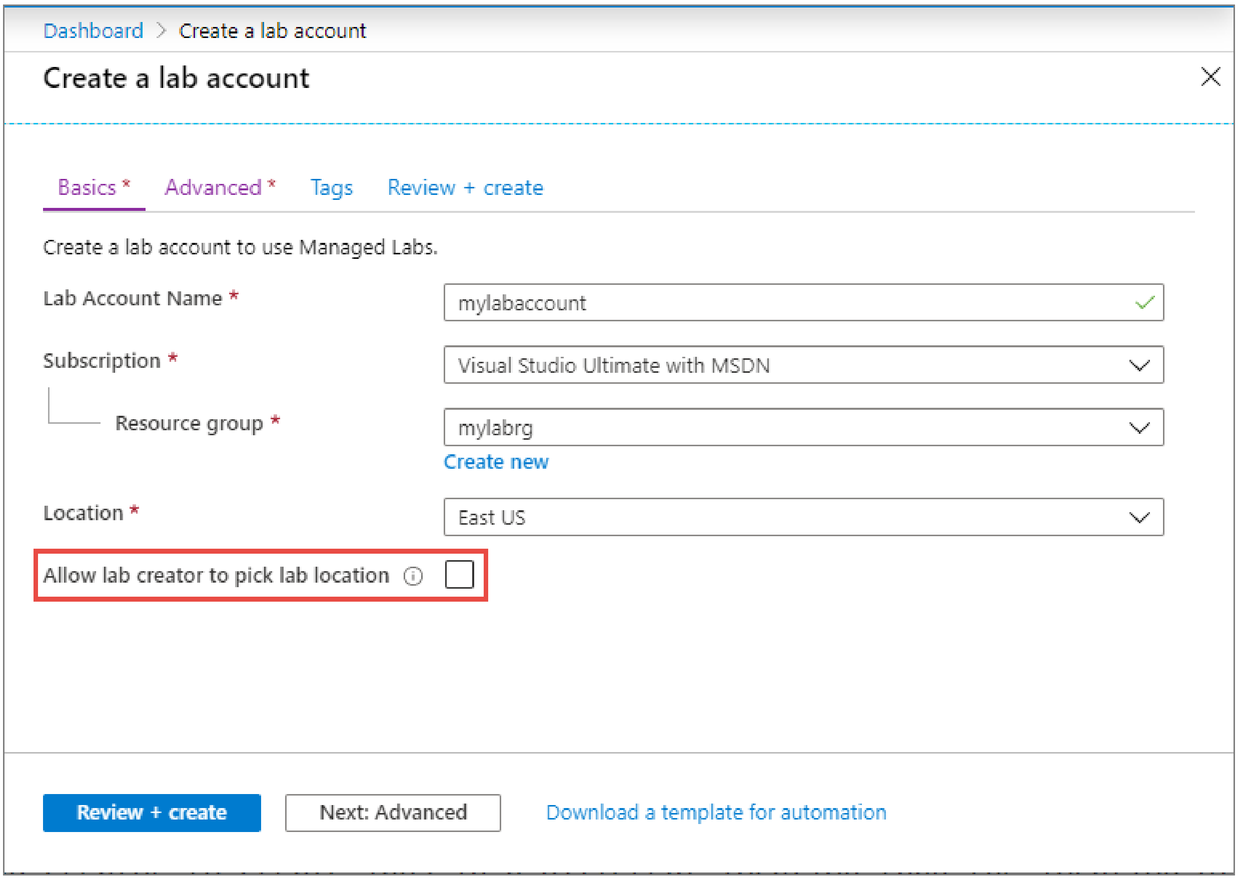
11/2/2020 • 3 minutes to read • Edit Online

In Azure Lab Services, a lab account owner can allow lab creators (educators) to pick a location for the lab they create. This location can be different from the location of the lab account. A location is a group of Azure regions. For example, United States location is a group of regions such as East US, West US, and so on.

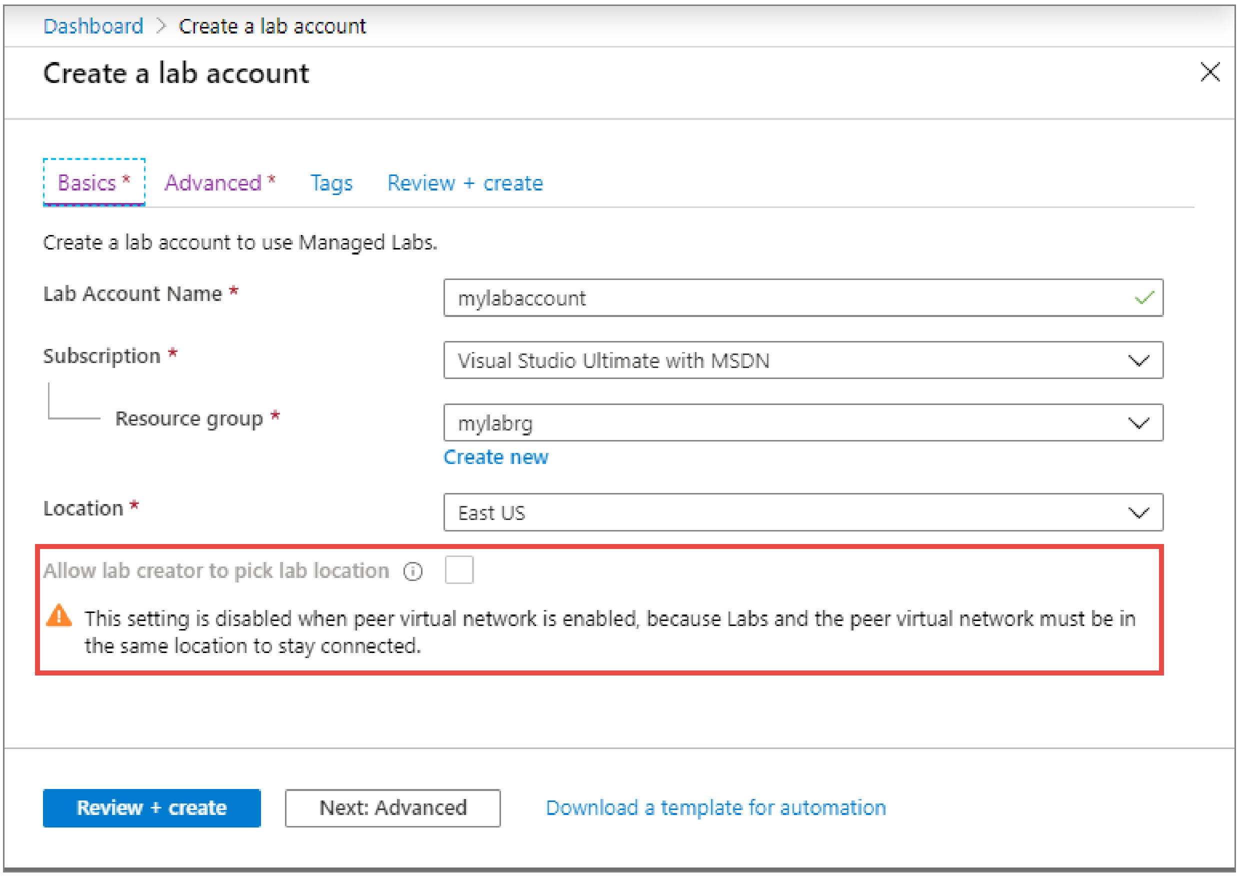
You, as a lab account owner, can select the **Allow lab creator to pick lab location** option when you create a lab account and after you create the lab account (or an existing lab account).

At the time of lab account creation

When you create a lab account, you see this option on the first screen (**Basics** tab).



This option is disabled if you select a peer virtual network for your lab account in the **Advanced** tab.

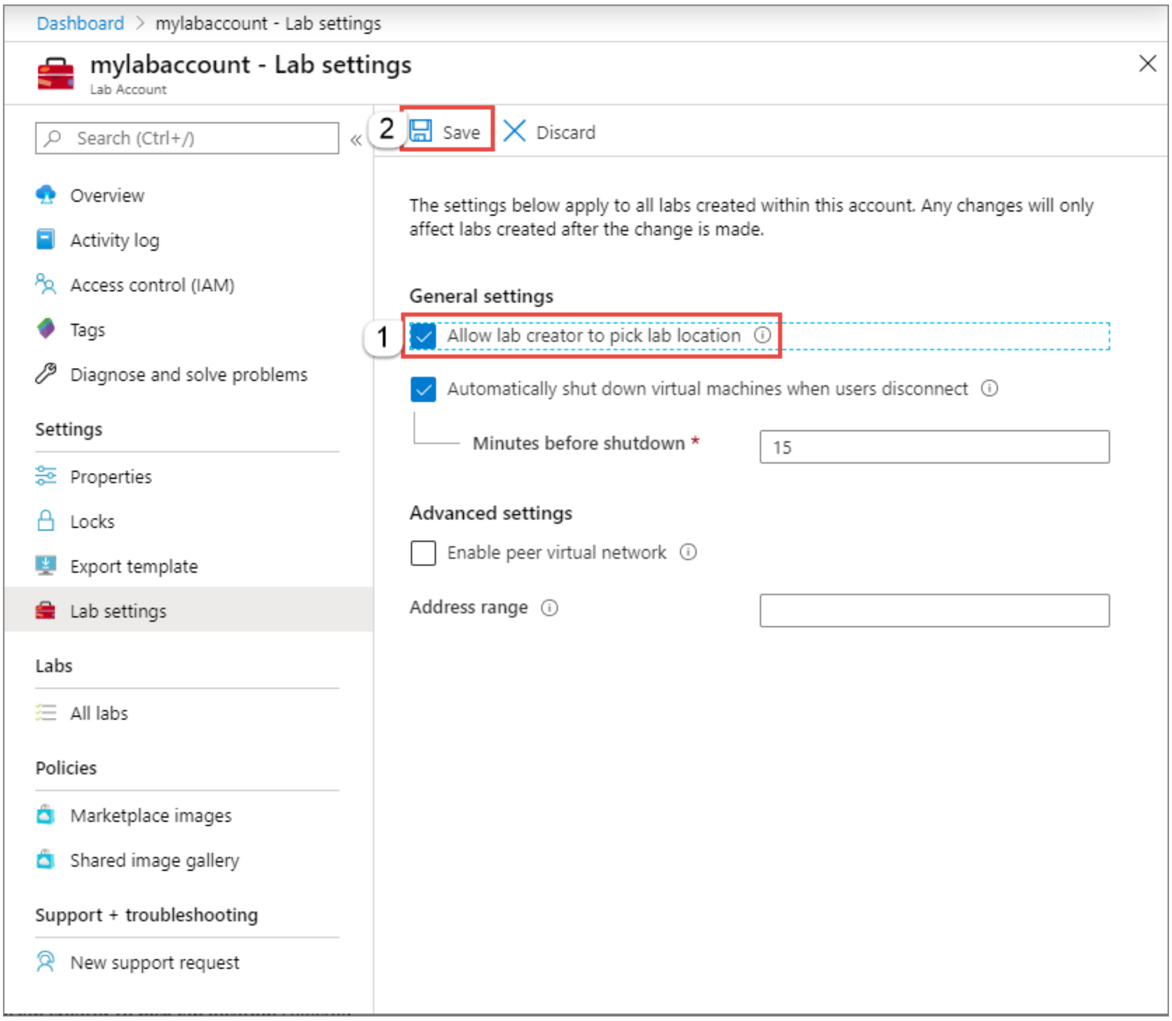


After the lab account is created

After you create the lab account, you can enable or disable this option by following these steps:

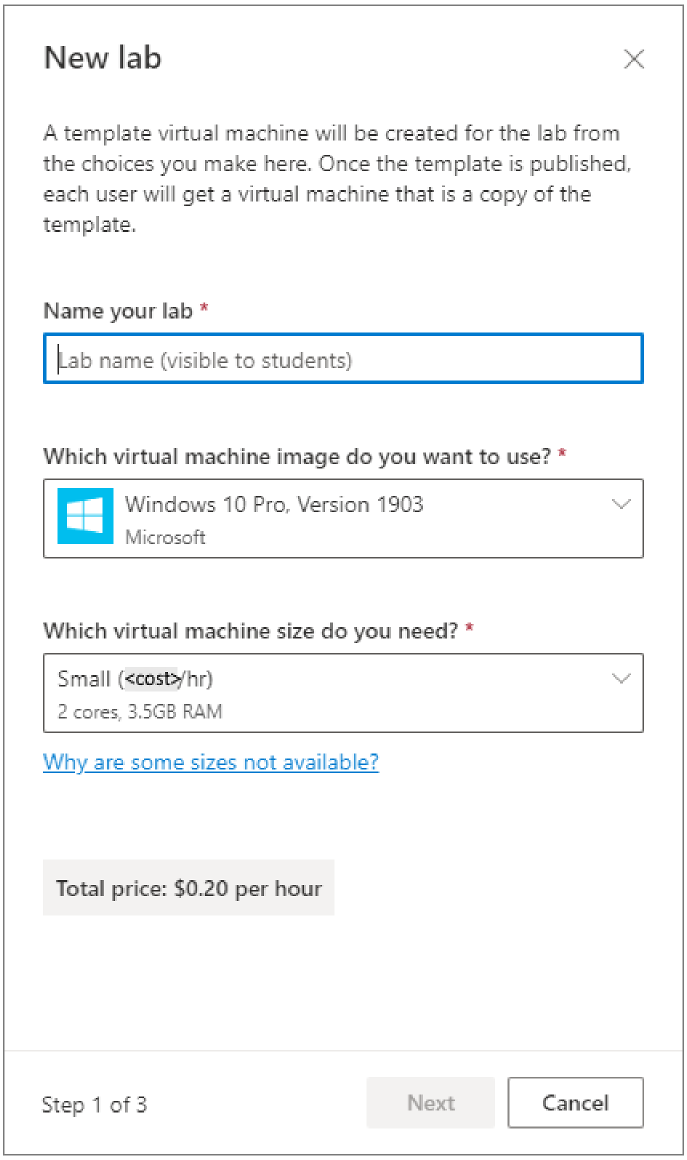
1. On the **Lab Account** page, select **Lab settings** on the left menu.
2. Select the **Allow lab creator to pick lab location** option if you want to allow the lab creator to select a location for the lab. If it's disabled, the labs are automatically created in the same location in which the lab account exists.

   This field is disabled when you select a virtual network for the **Peer virtual network** field. It's because labs in the lab account must be in the same region as the lab account for them to access resources in the peer virtual network.
3. Select **Save** on the toolbar.



No virtual network and location selection isn't allowed

In this scenario, you haven't enabled the **Allow lab creator to pick lab location** option.



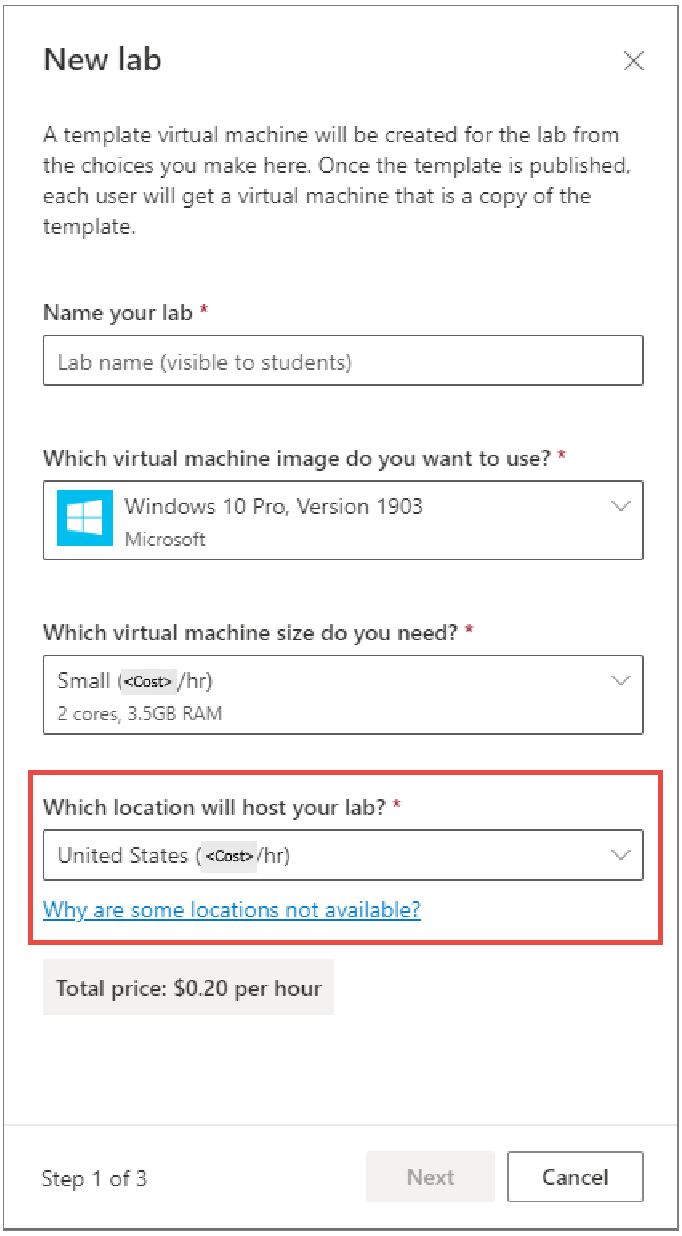
Then, lab creators (educators) don't see an option to pick a location for the lab. They will see the price per hour for every size option available to them. When they create a lab, it will be created in an Azure region that's in the same location as the Azure region that their lab account is in. For example, if the lab account is in **West US**, then the lab might be created in **South Central US** but would not be created in **Canada East**. We don't guarantee anything about the region we choose aside from that it's in the location. If a size is currently constrained, then the lab creator will see a checkbox where they can see the sizes that we normally support but are currently unavailable.

In virtual network and location selection isn't allowed

In this scenario, the **Allow lab creator to pick lab location** option is disabled because you have selected a peer virtual network for the lab account. Then, lab creators will see the same screen as with the previous option. Because all VMs have to be in the same Azure region as the virtual network, the lab will be created in the same Azure region that the virtual network is in. If that particular region is constrained for a size, the size will appear as unavailable.

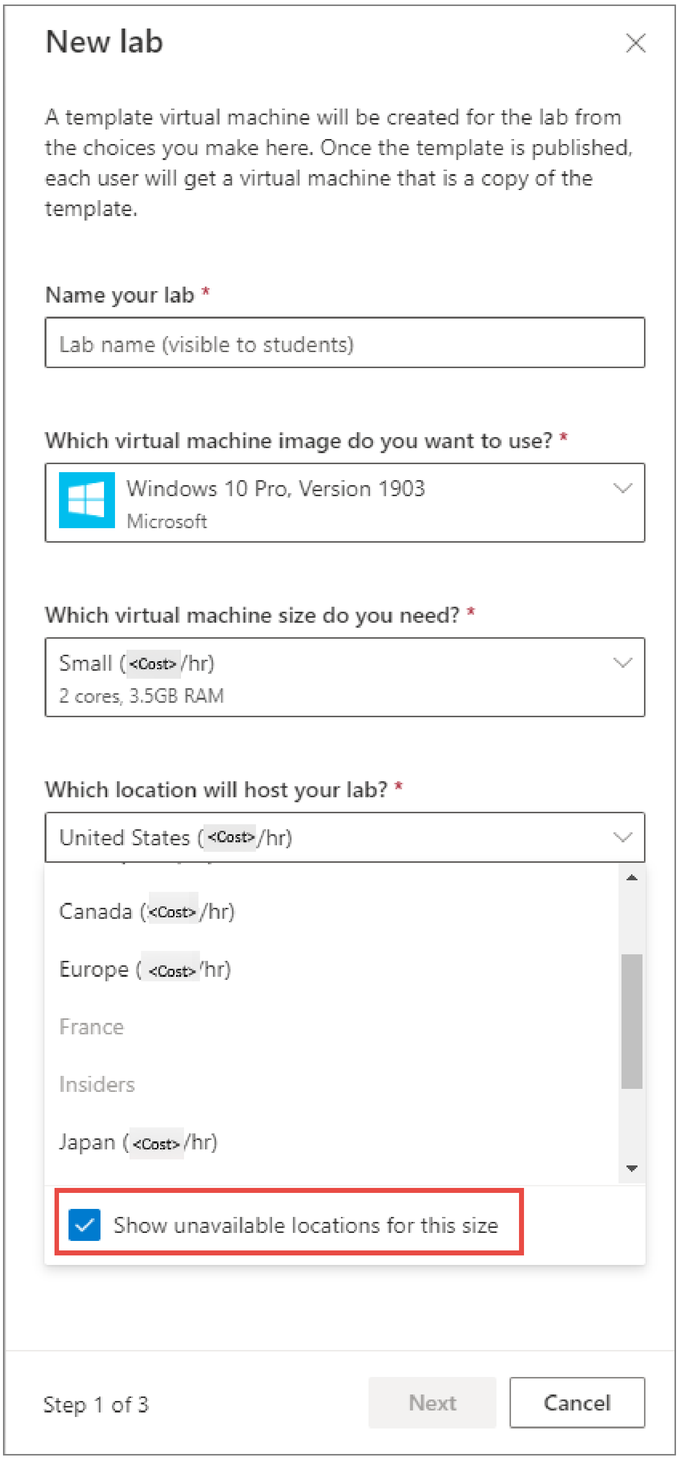
Location selection is enabled

When you select **Allow lab creator to pick lab location**, lab creators (educators) see an option to select a location when creating a lab.



Lab creators see the range of prices for all locations that size is in, and can choose a Location. The lab will be created in any Azure region that maps to that location.

If a location is constrained, it's not shown in the list by default. Expand the drop-down list, and select **Show unavailable locations for this size**.



Cost

Earlier, the pricing was based on the VM size that you choose for the lab. Now, the price is based on the combination of Operating System (OS), Size, and location.

Next steps

See the following articles:

- Connect your lab's network with a peer virtual network
- Attach a shared image gallery to a lab
- Add a user as a lab owner
- View firewall settings for a lab
- Configure other settings for a lab

Connect your lab's network with a peer virtual network in Azure Lab Services

5/5/2021 • 5 minutes to read • Edit Online

This article provides information about peering your labs network with another network.

Overview

Virtual network peering enables you to seamlessly connect Azure virtual networks. Once peered, the virtual networks appear as one, for connectivity purposes. The traffic between virtual machines in the peered virtual networks is routed through the Microsoft backbone infrastructure, much like traffic is routed between virtual machines in the same virtual network, through private IP addresses only. For more information, see Virtual network peering.

You may need to connect your lab's network with a peer virtual network in some scenarios including the following ones:

- The virtual machines in the lab have software that connects to on-premises license servers to acquire license.
- The virtual machines in the lab need access to data sets (or any other files) on university's network shares.

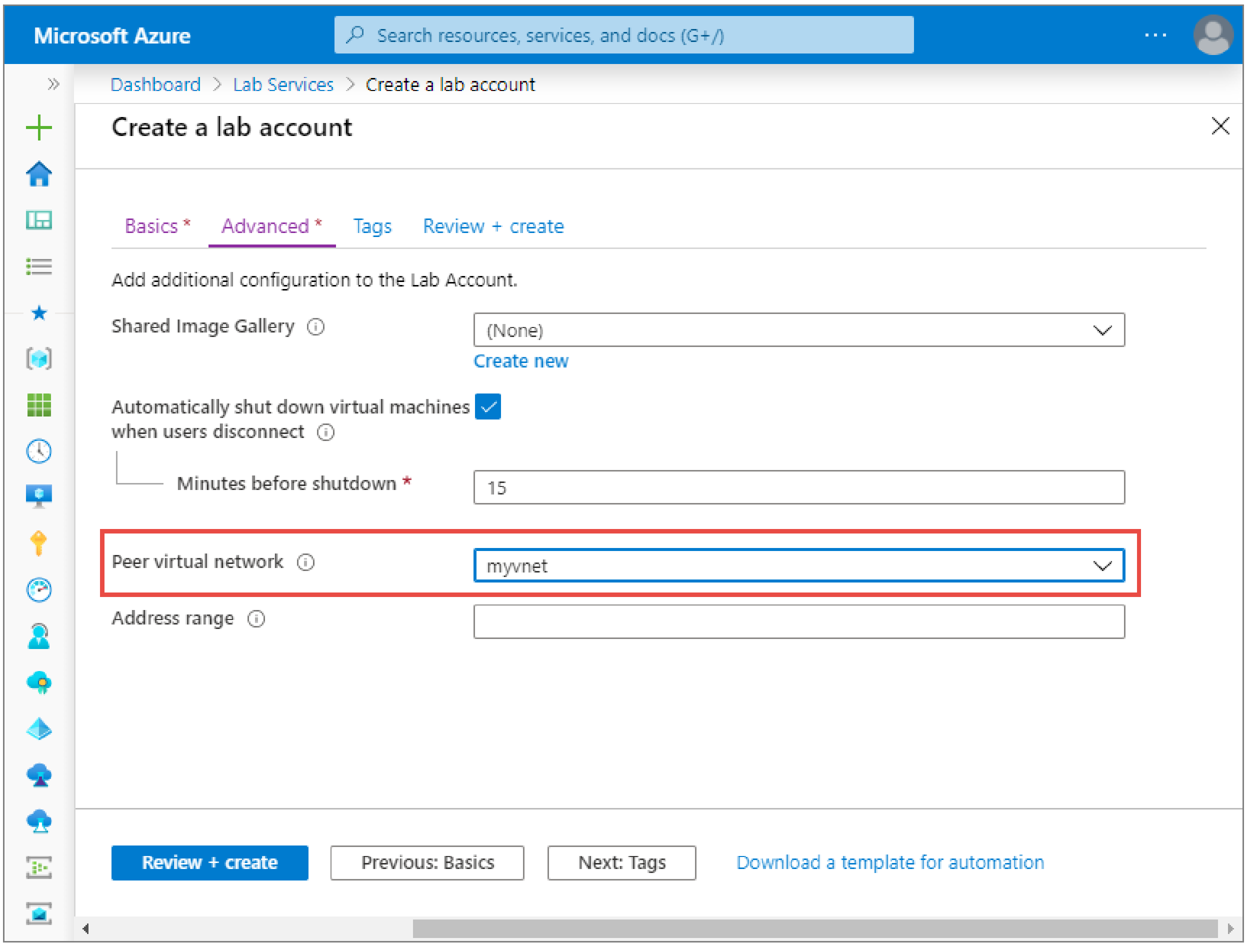
Certain on-premises networks are connected to Azure Virtual Network either through ExpressRoute or Virtual Network Gateway. These services must be set up outside of Azure Lab Services. To learn more about connecting an on-premises network to Azure using ExpressRoute, see ExpressRoute overview. For on-premises connectivity using a Virtual Network Gateway, the gateway, specified virtual network, and the lab account must all be in the same region.

> **NOTE**
>
> When creating a Azure Virtual Network that will be peered with a lab account, it's important to understand how the virtual network's region impacts where labs are created. For more information, see the administrator guide's section on regions\locations.

> **NOTE**
>
> If your school needs to perform content filtering, such as for compliance with the Children's Internet Protection Act (CIPA), you will need to use 3rd party software. For more information, read guidance on content filtering with Lab Services.

Configure at the time of lab account creation

During the new lab account creation, you can pick an existing virtual network that shows in the **Peer virtual network** dropdown list on the **Advanced** tab. The list will only show virtual networks in the same region as the lab account. The selected virtual network is connected (peered) to labs created under the lab account. All the virtual machines in labs that are created after the making this change will have access to the resources on the peered virtual network.



Address range

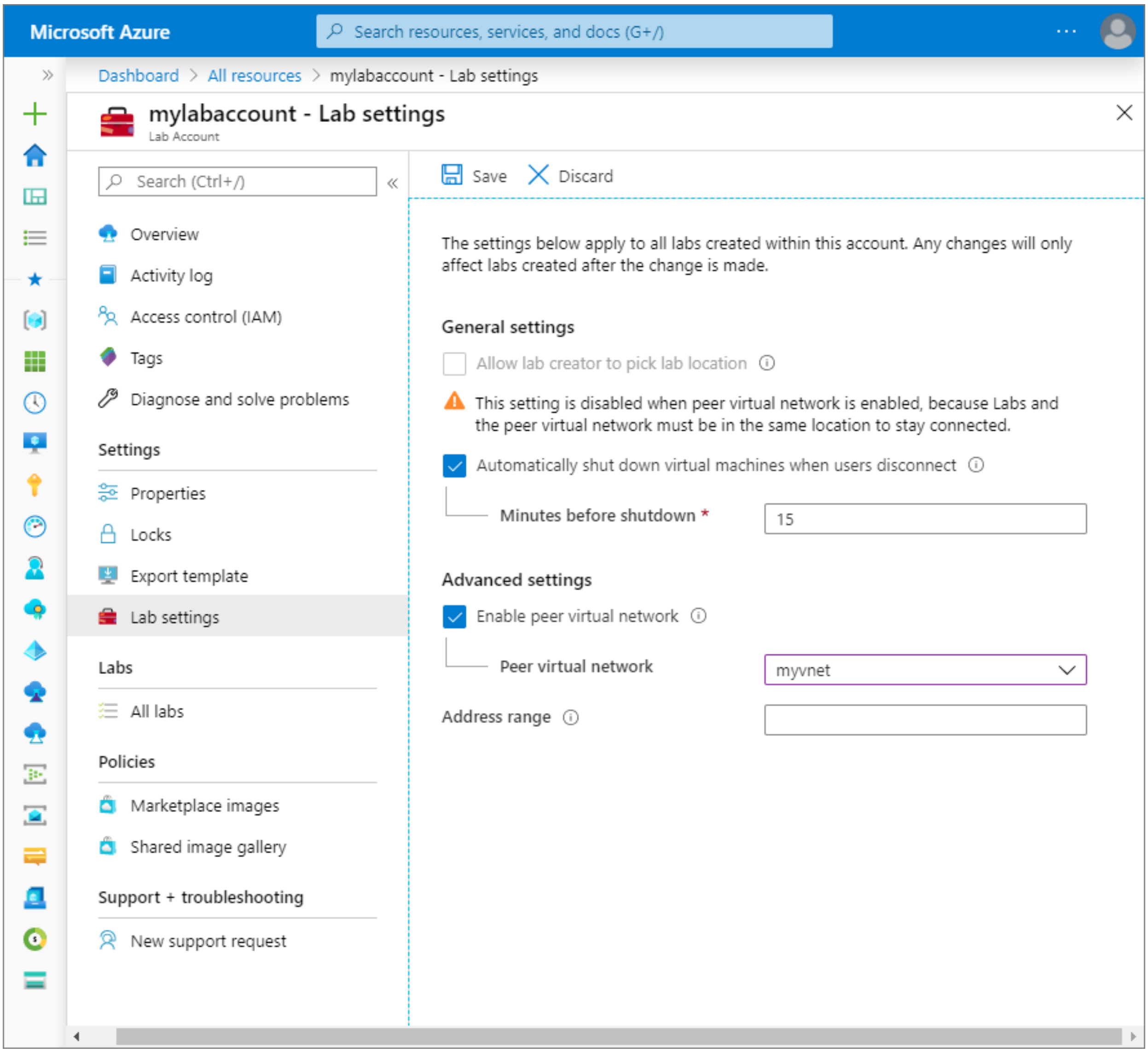
There is also an option to provide **Address range** for virtual machines for the labs. The **address range** property applies only if a **peer virtual network** is enabled for the lab. If the address range is provided, all the virtual machines in the labs under the lab account will be created in that address range. The address range should be in CIDR notation (for example, 10.20.0.0/20) and not overlap with any existing address ranges. When providing an address range, it's important to think about the number of *labs* that will be created and provide an address range to accommodate that. Lab Services assumes a maximum of 512 virtual machines per lab. For example, an ip range with '/23' can create only one lab. A range with a '/21' will allow for the creation of four labs.

If the **Address range** is not specified, Lab Services will use the default address range given to it by Azure when creating the virtual network to be peered with your virtual network. The range is often something like 10.x.0.0/16. This may lead to ip range overlap, so make sure to either specify an address range in the lab settings or check the address range of your virtual network being peered.

> **NOTE**
>
> Lab creation can fail if the lab account is peered to a virtual network but has too narrow of an IP address range. You can run out of space in the address range if there are too many labs in the lab account (each lab uses 512 addresses).
>
> If the lab creation fails, contact your lab account owner/admin and request for the address range to be increased. The admin can increase the address range using steps mentioned in the Specify an address range for VMs in a lab account section.

Configure after the lab account is created

The same property can be enabled from the **Labs configuration** tab of the **Lab Account** page if you didn't set up a peer network at the time of lab account creation. Change made to this setting applies only to the labs that are created after the change. As you can see in the image, you can enable or disable **Peer virtual network** for labs in the lab account.



When you select a virtual network for the **Peer virtual network** field, the **Allow lab creator to pick lab location** option is disabled. That's because labs in the lab account must be in the same region as the lab account for them to connect with resources in the peer virtual network.

> **IMPORTANT**
>
> The peered virtual network setting applies only to labs that are created after the change is made, not to the existing labs.

Specify an address range for VMs in the lab account

The following procedure has steps to specify an address range for VMs in the lab. If you update the range that you previously specified, the modified address range applies only to VMs that are created after the change was made.

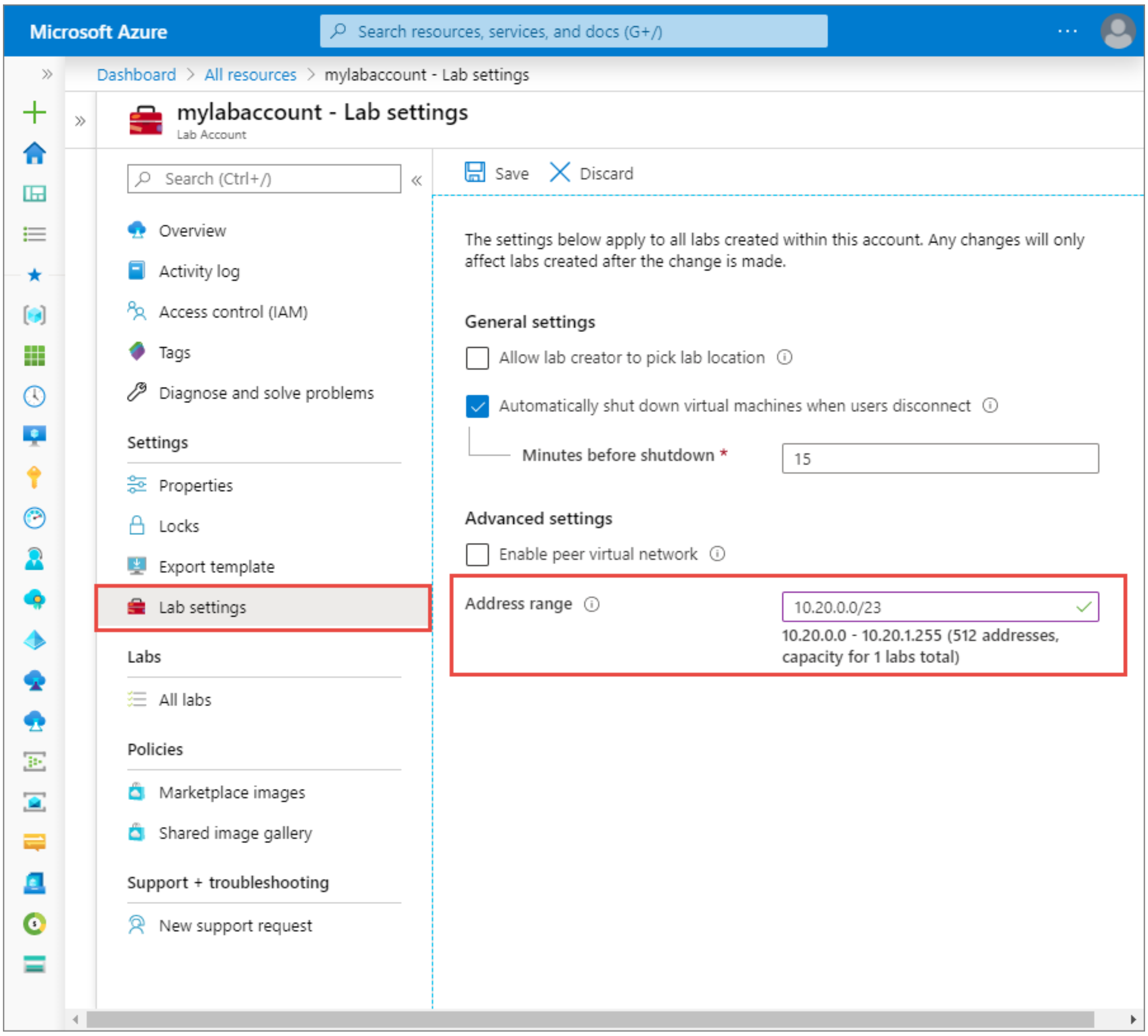
Here are some restrictions when specifying the address range that you should keep in mind.

- The prefix must be smaller than or equal to 23.
- If a virtual network is peered to the lab account, the provided address range cannot overlap with address range from peered virtual network.

1. On the **Lab Account** page, select **Labs settings** on the left menu.

2. For the **Address range** field, specify the address range for VMs that will be created in the lab. The address range should be in the classless inter-domain routing (CIDR) notation (example: 10.20.0.0/23). Virtual machines in the lab will be created in this address range.

3. Select **Save** on the toolbar.



Next steps

See the following articles:

- Allow lab creator to pick lab location
- Attach a shared image gallery to a lab
- Add a user as a lab owner
- View firewall settings for a lab
- Configure other settings for a lab

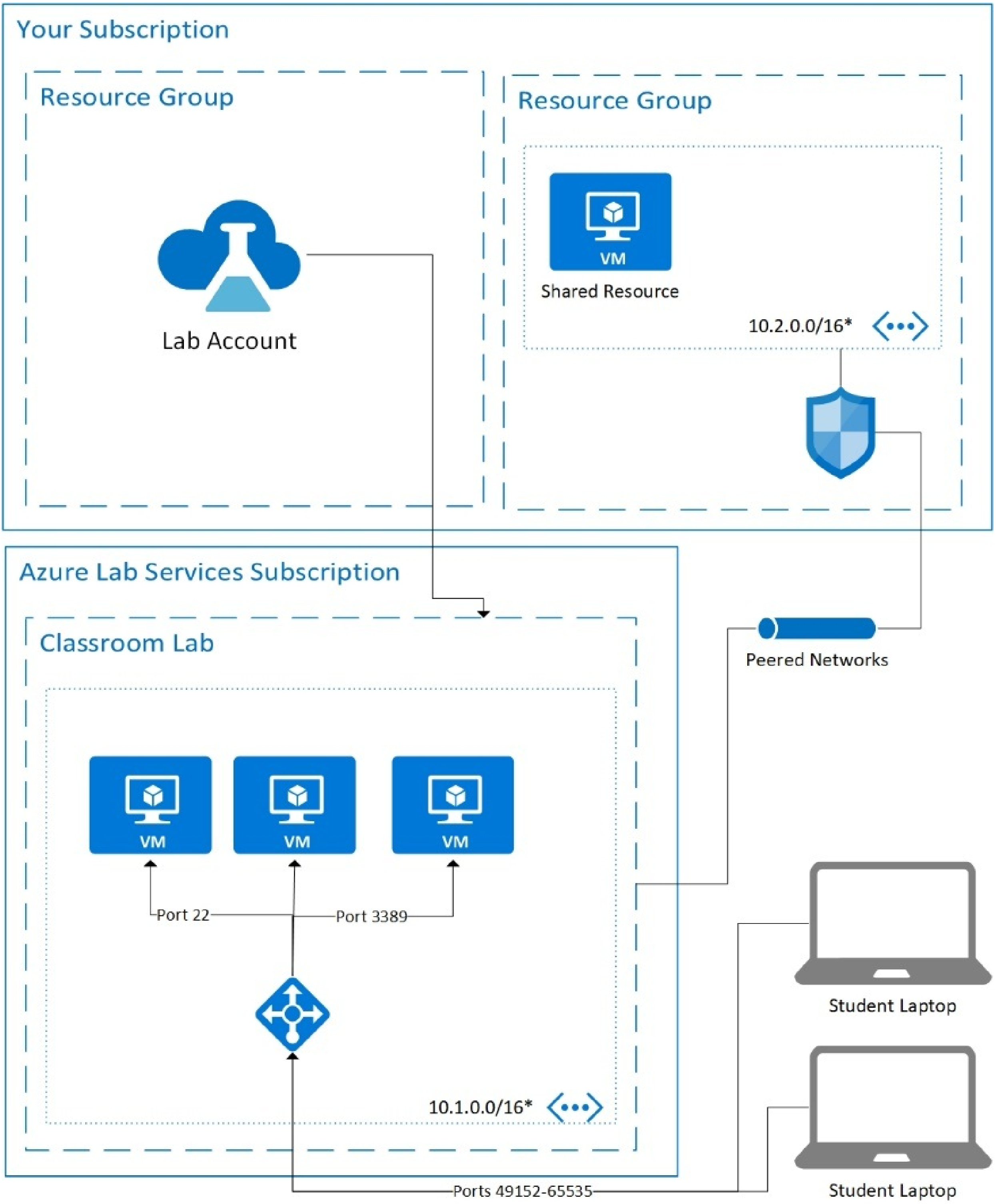
How to create a lab with a shared resource in Azure Lab Services

3/5/2021 • 3 minutes to read • Edit Online

Sometimes when creating a classroom lab, there may be some resources that need to be shared among all the students in a lab. For example, you have a licensing server or SQL Server for a database class. This article will discuss the steps to enable the shared resource for a lab. We'll also talk about how to limit access to that shared resource.

Architecture

As shown in the diagram below we'll have a lab account with a lab. The lab account will have the vnet peering settings so the virtual network for the lab is connected to the network of the shared resource. In the diagram below there are two virtual networks with non-overlapping IP ranges. These IP ranges are just example ranges. Also note that the shared resource virtual network is in the same subscription as the lab account.



Setup shared resource

The virtual network for the shared resource must be created before the lab is created. For more information on how to create a virtual network, see create a virtual network. Planning out virtual network ranges so they don't overlap with the ip address of the lab machines is important. For more information about planning your network, see the plan virtual networks article. In our example, the shared resource is in a virtual network with the range 10.2.0.0/16. If not done already, create a subnet to hold the shared resource. In the example, we use the 10.2.0.0/24 range, but your range may be different depending on the needs of your network.

The shared resource can be software running on a virtual machine or an Azure provided service. The shared resource should be available through private IP address. By making the shared resource available through private IP only, you limit access to that shared resource.

The diagram also shows a network security group (NSG) which can be used to restrict traffic coming from the student VM. For example, you can write a security rule that states traffic from the student VM's IP addresses can only access one shared resource and nothing else. For more information how to set security rules, see manage network security group. If you want to restrict access to a shared resource to a specific lab, get the IP address for the lab from the lab settings from the lab account and set an inbound rule to allow access only from that IP address. Don't forget to allow ports 49152 to 65535 for that IP address. Optionally you can find the private IP address of the student's VMs by using the virtual machine pool page.

If your shared resource is an Azure virtual machine running necessary software, you may have to modify the default firewall rules for the virtual machine.

Tips for shared resources - License server

One of the more common shared resources is a License server, here are a few tips on how to be successful with setting one up.

Server region

The License server will need to be connected to the virtual network that is peered to the lab, so the license server needs to be located in the same region as the lab account.

Static private IP and MAC address

By default virtual machines have a dynamic private ip, before you setup any software set the private ip to static. This sets the private IP and the MAC address to be static.

Control Access

Controlling access to the license server is key. Once the VM is setup access will still be needed for maintenance, troubleshooting, and updating. Here are a few different ways to do this.

- Setting up Just in Time (JIT) access within Azure Security Center.
- Setting up a Network Security Group to restrict access.
- Setup Bastion to allow secure access to the license server.

Lab Account

To use a shared resource, the lab account must be set up to use a peered virtual network. In this case, we will be peering to the virtual network that holds the shared resource.

> **WARNING**
>
> The lab for your class must be created **after** the lab account is peered to the shared resource virtual network.
> Template machine

Once your lab account is peered to the virtual network, the template machine should now have access to the shared resource. You may have to update the firewall rules, depending on the shared resource being accessed.

Attach or detach a shared image gallery in Azure Lab Services

8/5/2021 • 4 minutes to read • Edit Online

This article shows you how to attach or detach a shared image gallery to a lab account.

> **NOTE**
>
> When you save a template image of a lab in Azure Lab Services to a shared image gallery, the image is uploaded to the gallery as a specialized image. Specialized images keep machine-specific information and user profiles. You can still directly upload a generalized image to the gallery outside of Azure Lab Services.
>
> A lab creator can create a template VM based on both generalized and specialized images in Azure Lab Services.

Scenarios

Here are the couple of scenarios supported by this feature:

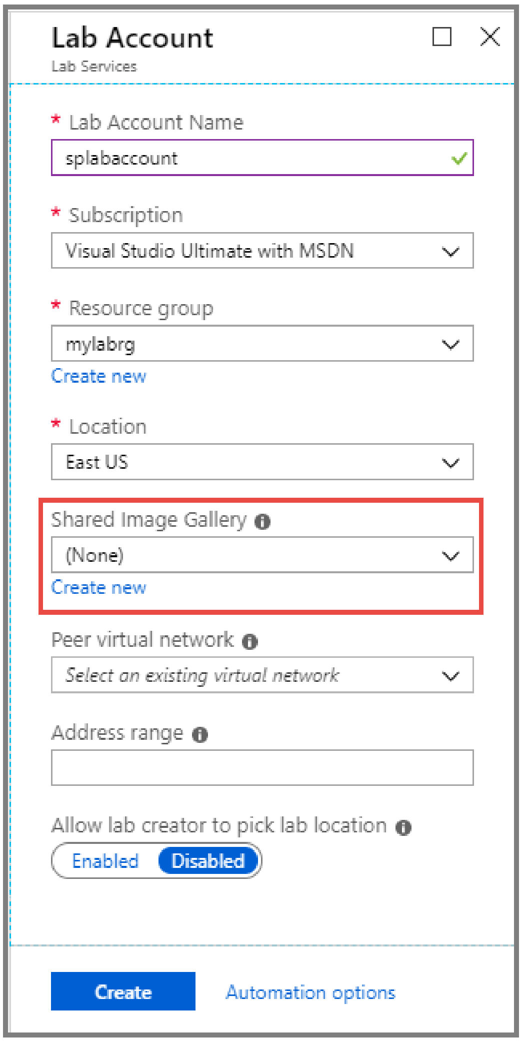
- A lab account admin attaches a shared image gallery to the lab account, and uploads an image to the shared image gallery outside the context of a lab. Then, lab creators can use that image from the shared image gallery to create labs.
- A lab account admin attaches a shared image gallery to the lab account. A lab creator (instructor) saves the customized image of his/her lab to the shared image gallery. Then, other lab creators can select this image from the shared image gallery to create a template for their labs.

  When an image is saved to a shared image gallery, Azure Lab Services replicates the saved image to other regions available in the same geography. It ensures that the image is available for labs created in other regions in the same geography. Saving images to a shared image gallery incurs an additional cost, which includes cost for all replicated images. This cost is separate from the Azure Lab Services usage cost. For more information about Shared Image Gallery pricing, see Shared Image Gallery – Billing.

> **IMPORTANT**
>
> While using a Shared Image Gallery, Azure Lab Services supports only images with less than 128 GB of OS Disk Space. Images with more than 128 GB of disk space or multiple disks will not be shown in the list of virtual machine images during lab creation.

Configure at the time of lab account creation

When you are creating a lab account, you can attach a shared image gallery to the lab account. You can either select an existing shared image gallery from the drop-down list or create a new one. To create and attach a shared image gallery to the lab account, select **Create new**, enter a name for the gallery, and enter **OK**.



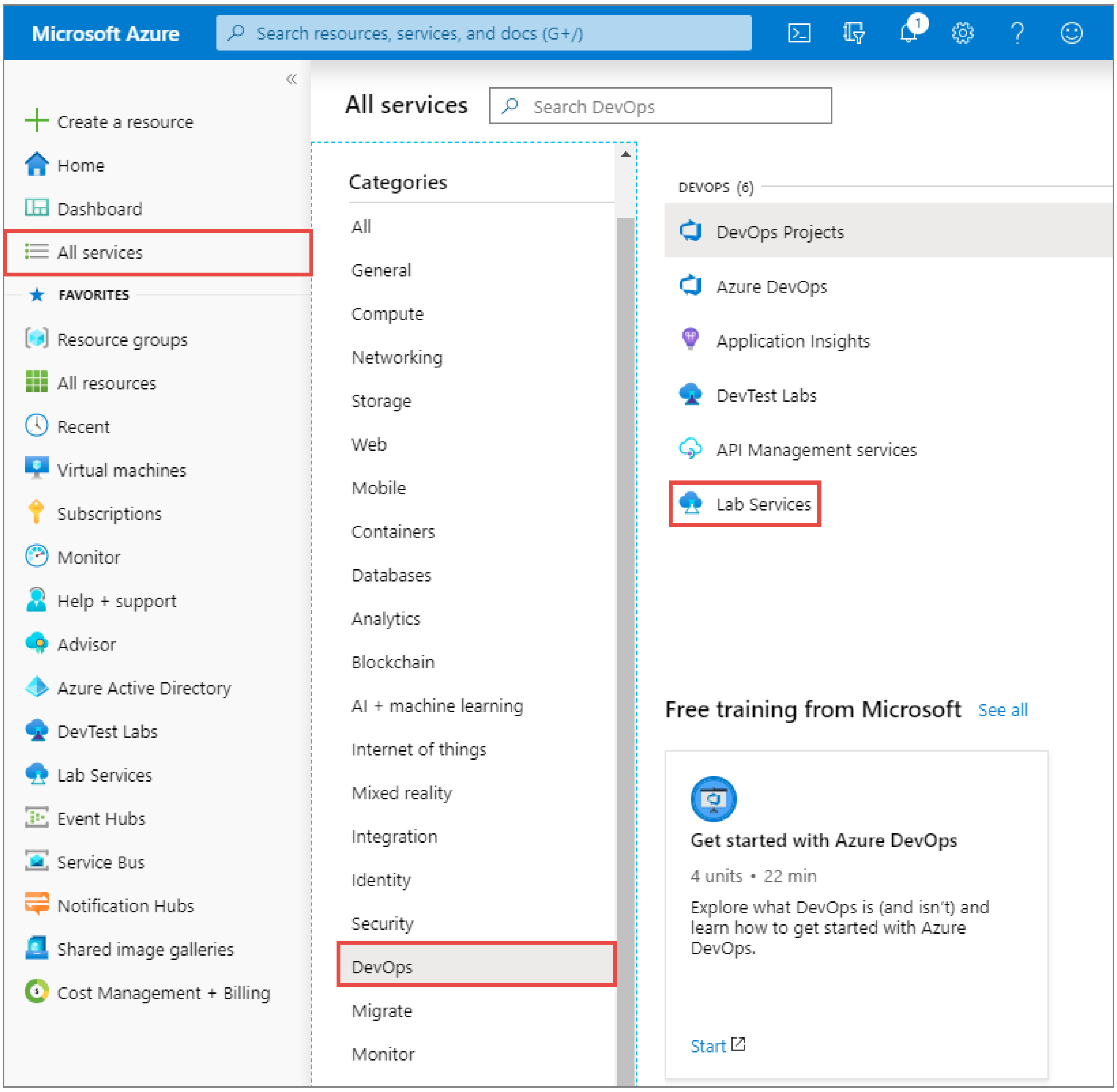
Configure after the lab account is created

After the lab account is created, you can do the following tasks:

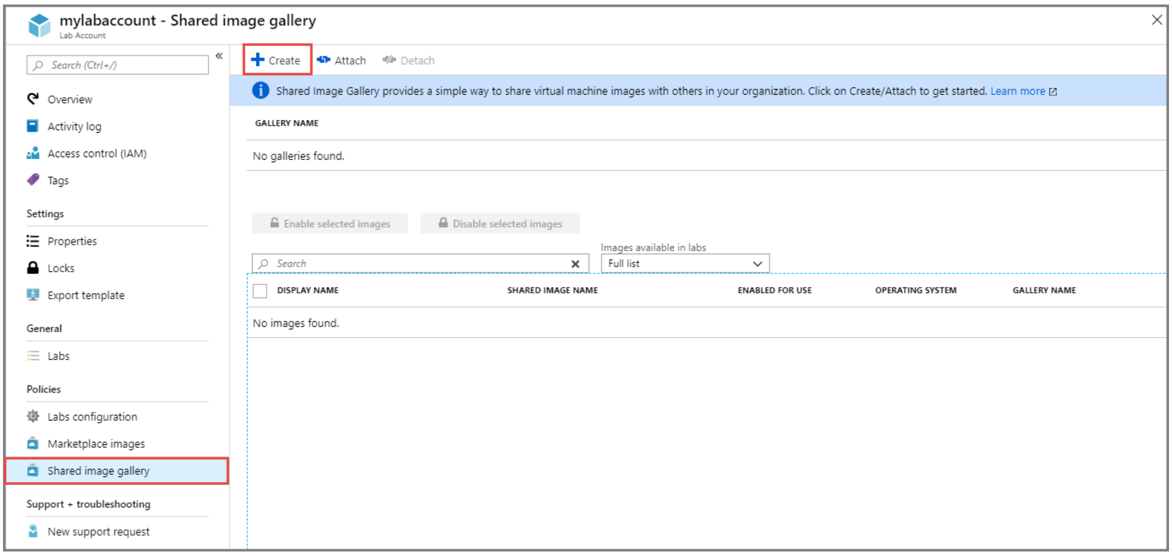
- Create and attach a shared image gallery
- Attach a shared image gallery to the lab account
- Detach a shared image gallery from the lab account

Create and attach a shared image gallery

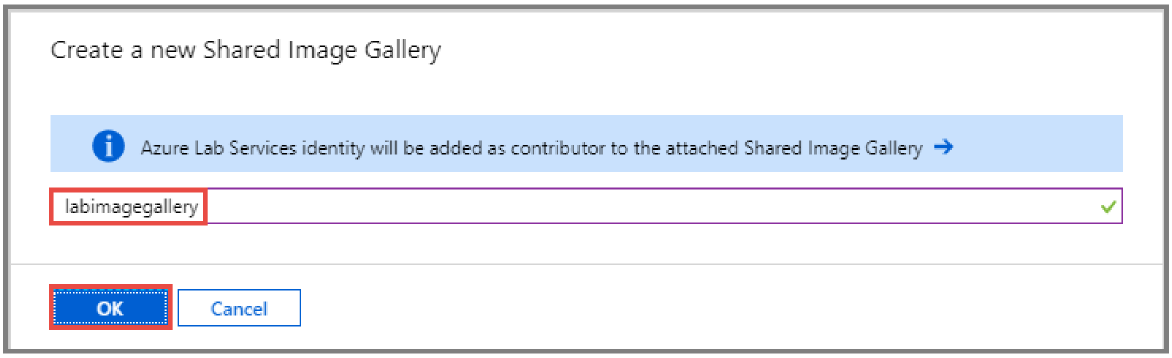
1. Sign in to the Azure portal.
2. Select **All Services** on the left menu. Select **Lab Services** in the **DEVOPS** section. If you select star (`*`) next to **Lab Services**, it's added to the **FAVORITES** section on the left menu. From the next time onwards, you select **Lab Services** under **FAVORITES**.



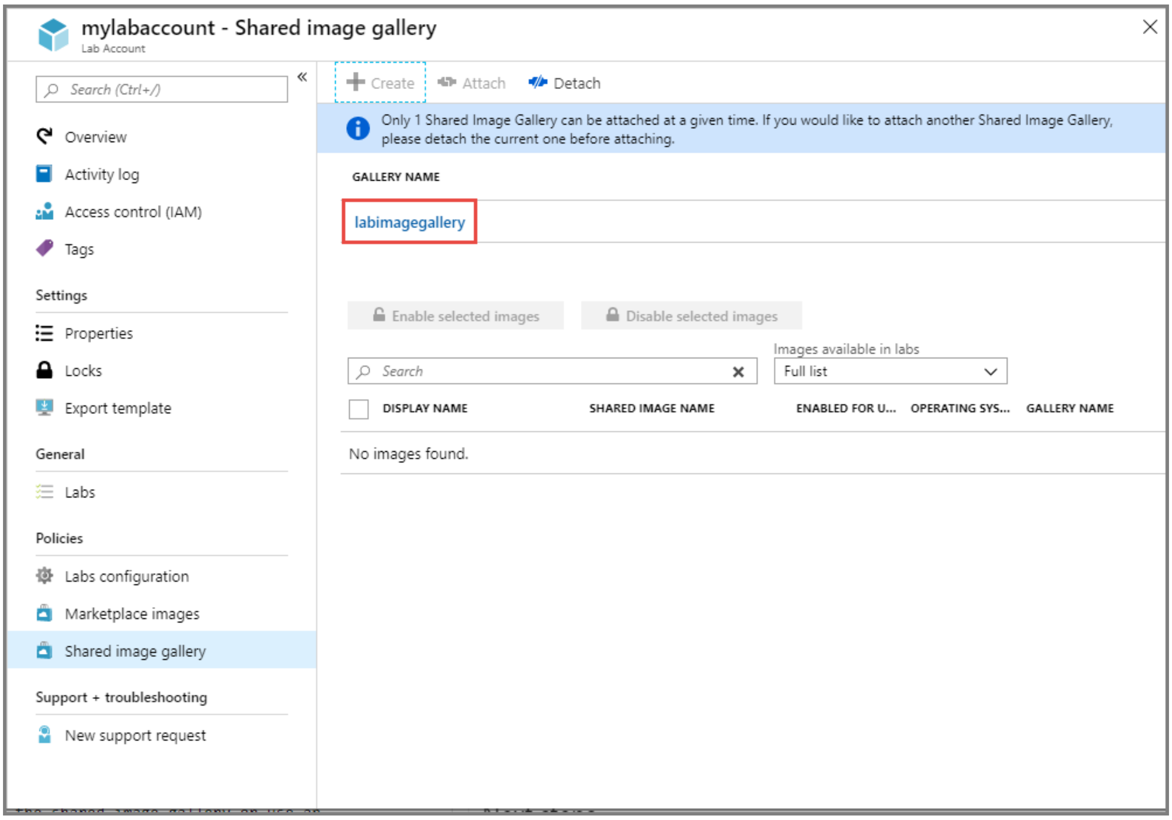
3. Select your lab account to see the **Lab Account** page.
4. Select **Shared image gallery** on the left menu, and select **+ Create** on the toolbar.



5. In the **Create shared image gallery** window, enter a **name** for the gallery, and enter **OK**.



Azure Lab Services creates the shared image gallery and attached it to the lab account. All labs created in this lab account have access to the attached shared image gallery.



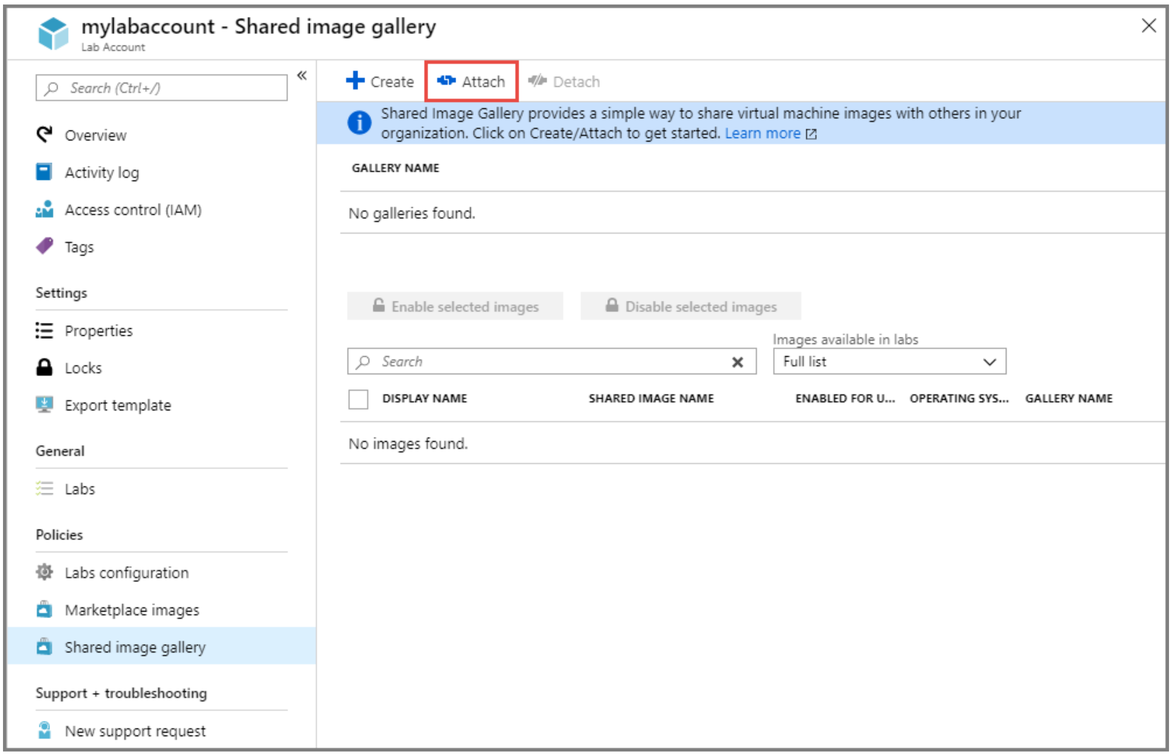
In the bottom pane, you see images in the shared image gallery. In this new gallery, there are no images. When you upload images to the gallery, you see them on this page.

All images in the attached shared image gallery are enabled by default. You can enable or disable selected images by selecting them in the list and using the **Enable selected images** or **Disable selected images** button.

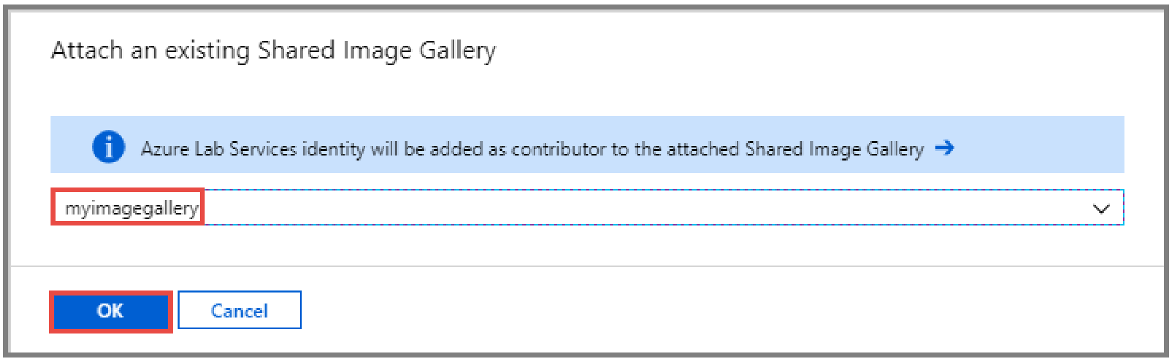
Attach an existing shared image gallery

The following procedure shows you how to attach an existing shared image gallery to a lab account.

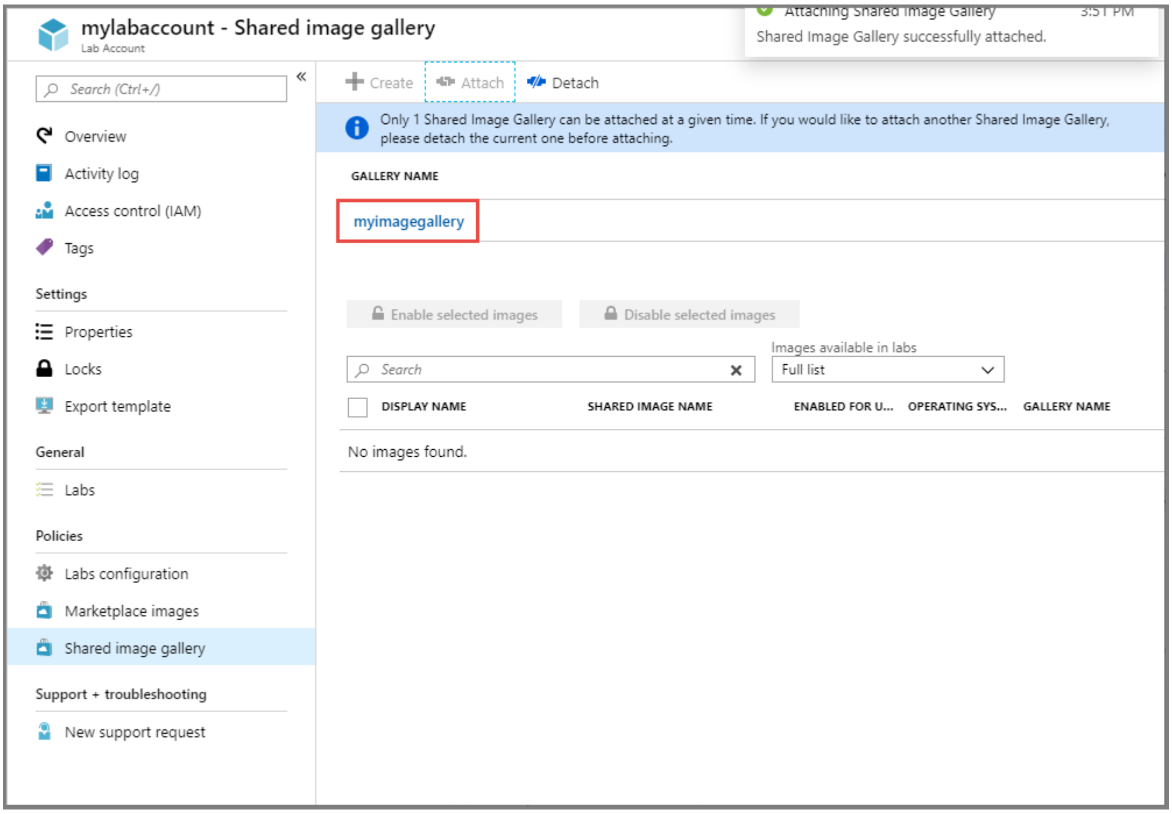
1. On the **Lab Account** page, select **Shared image gallery** on the left menu, and select **Attach** on the toolbar.



2. On the **Attach an existing Shared Image Gallery** page, select your shared image gallery, and select **OK**.



3. You see the following screen:



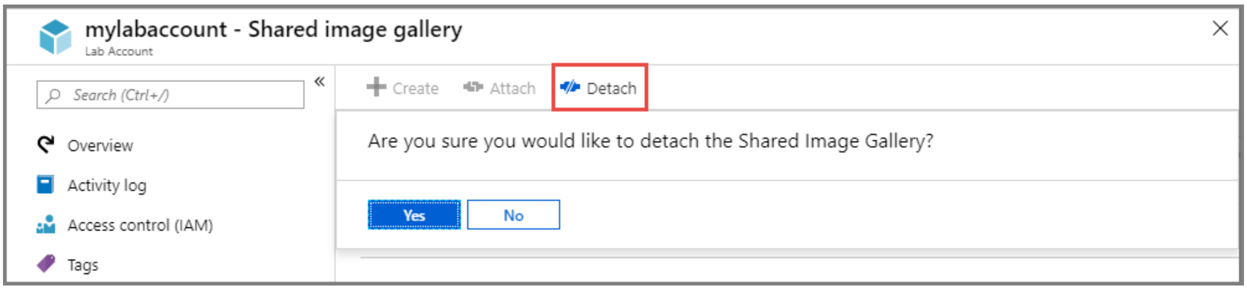
In this example, there are no images in the shared image gallery yet.

Azure Lab Services identity is added as a contributor to the shared image gallery that is attached to the lab. It allows educators/IT admins to save virtual machine images to the shared image gallery. All labs created in this lab account have access to the attached shared image gallery.

All images in the attached shared image gallery are enabled by default. You can enable or disable selected images by selecting them in the list and using the **Enable selected images** or **Disable selected images** button.

Detach a shared image gallery

Only one shared image gallery can be attached to a lab. If you would like to attach another shared image gallery, detach the current one before attaching the new one. To detach a shared image gallery from your lab, select **Detach** on the toolbar, and confirm the detach operation.



Next steps

To learn about how to save a lab image to the shared image gallery or use an image from the shared image gallery to create a VM, see How to use shared image gallery.

To explore other options for bringing custom images to shared image gallery outside of the context of a lab, see Recommended approaches for creating custom images.

For more information about shared image galleries in general, see shared image gallery.

Recommended approaches for creating custom images

9/3/2021 • 5 minutes to read • Edit Online

This article describes the following recommended approaches for creating a custom image:

- Create and save a custom image from a lab's template virtual machine (VM).
- Bring a custom image from outside of the context of a lab by using:
  - An Azure VM.
  - A VHD in your physical lab environment.

Save a custom image from a lab's template VM

Using a lab's template VM to create and save a custom image is the simplest way to create an image because it's supported by using the Azure Lab Services portal. As a result, both IT departments and educators can create custom images by using a lab's template VM.

For example, you can start with one of the Azure Marketplace images and then install the software applications and tooling that are needed for a class. After you've finished setting up the image, you can save it in the connected shared image gallery so that you and other educators can use the image to create new labs.

There are a few key points to be aware of with this approach:

- Lab Services automatically saves a *specialized* image when you export the image from the template VM. In most cases, specialized images are well suited for creating new labs because the image retains machine-specific information and user profiles. Using a specialized image helps to ensure that the installed software will run the same when you use the image to create new labs. If you need to create a *generalized* image, you must use one of the other recommended approaches in this article to create a custom image.

  You can create labs based on both generalized and specialized images in Azure Lab Services. For more information about the differences, see Generalized and specialized images.

- For more advanced scenarios with setting up your image, you might find it helpful to instead create an image outside of labs by using either an Azure VM or a VHD from your physical lab environment. Read the next sections for more information.

Use a lab's template VM to save a custom image

You can use a lab's template VM to create either Windows or Linux custom images. For more information, see Save the image to a shared image gallery.

Bring a custom image from an Azure VM

Another approach is to use an Azure VM to set up a custom image. After you've finished setting up the image, you can save it to a shared image gallery so that you and your colleagues can use the image to create new labs.

Using an Azure VM gives you more flexibility:

- You can create either generalized or specialized images. Otherwise, if you use a lab's template VM to export an image the image is always specialized.
- You have access to more advanced features of an Azure VM that might be helpful for setting up an image.

For example, you can use extensions to do post-deployment configuration and automation. Also, you can access the VM's boot diagnostics and serial console.

Setting up an image by using an Azure VM is more complex. As a result, IT departments are typically responsible for creating custom images on Azure VMs.

Use an Azure VM to set up a custom image

Here are the high-level steps to bring a custom image from an Azure VM:

1. Create an Azure VM by using a Windows or Linux Marketplace image.
2. Connect to the Azure VM and install more software. You can also make other customizations that are needed for your lab.
3. When you've finished setting up the image, save the VM's image to a shared image gallery. As part of this step, you'll also need to create the image's definition and version.
4. After the custom image is saved in the gallery, you can use your image to create new labs.

The steps vary depending on if you're creating a custom Windows or Linux image. Read the following articles for the detailed steps:

- Bring a custom Windows image from an Azure VM
- Bring a custom Linux image from an Azure VM

Bring a custom image from a VHD in your physical lab environment

The third approach to consider is to bring a custom image from a VHD in your physical lab environment to a shared image gallery. After the image is in a shared image gallery, you and other educators can use the image to create new labs.

Here are a few reasons why you might want to use this approach:

- You can create either generalized or specialized images to use in your labs. Otherwise, if you use a lab's template VM to export an image, the image is always specialized.
- You can access resources that exist within your on-premises environment. For example, you might have large installation files in your on-premises environment that are too time consuming to copy to a lab's template VM.
- You can upload images created by using other tools, such as Microsoft Endpoint Configuration Manager, so that you don't have to manually set up an image by using a lab's template VM.

Bringing a custom image from a VHD is the most advanced approach because you must ensure that the image is set up properly so that it works within Azure. As a result, IT departments are typically responsible for creating custom images from VHDs.

Bring a custom image from a VHD

Here are the high-level steps to bring a custom image from a VHD:

1. Use Windows Hyper-V on your on-premises machine to create a Windows or Linux VHD.
2. Connect to the Hyper-V VM and install more software. You can also make other customizations that are needed for your lab.
3. When you've finished setting up the image, upload the VHD to create a managed disk in Azure.
4. From the managed disk, create the image's definition and version in a shared image gallery.
5. After the custom image is saved in the gallery, you can use the image to create new labs.

The steps vary depending on if you're creating a custom Windows or Linux image. Read the following articles for the detailed steps:

- Bring a custom Windows image from a VHD
- Bring a custom Linux image from a VHD

Next steps

- Shared image gallery overview
- Attach or detach a shared image gallery
- Use a shared image gallery

Bring a Windows custom image from an Azure virtual machine

9/3/2021 • 2 minutes to read • Edit Online

The steps in this article show how to import a custom image that starts from an Azure virtual machine (VM). With this approach, you set up an image on an Azure VM and import the image into a shared image gallery so that it can be used within Azure Lab Services. Before you use this approach for creating a custom image, read Recommended approaches for creating custom images to decide the best approach for your scenario.

Prerequisites

You'll need permission to create an Azure VM in your school's Azure subscription to complete the steps in this article.

Prepare a custom image on an Azure VM

1. Create an Azure VM by using the Azure portal, PowerShell, the Azure CLI, or an Azure Resource Manager template.
   - When you specify the disk settings, ensure the disk's size is *not* greater than 128 GB.
2. Install software and make any necessary configuration changes to the Azure VM's image.
3. Optionally, you can generalize the image. Run SysPrep if you need to create a generalized image. Otherwise, if you're creating a specialized image, you can skip to the next step.

   Create a specialized image if you want to maintain machine-specific information and user profiles. For more information about the differences between generalized and specialized images, see Generalized and specialized images.

Import the custom image into a shared image gallery

1. In a shared image gallery, create an image definition or choose an existing image definition.
   - Choose **Gen 1** for the **VM generation**.
   - Choose whether you're creating a **specialized** or **generalized** image for the **Operating system state**.

   For more information about the values you can specify for an image definition, see Image definitions.

   You can also choose to use an existing image definition and create a new version for your custom image.
2. Create an image version.
   - The **Version number** property uses the following format: *MajorVersion.MinorVersion.Patch*.
   - For the **Source**, select **Disks and/or snapshots** from the dropdown list.
   - For the **OS disk** property, choose your Azure VM's disk that you created in previous steps.

   You can also import your custom image from an Azure VM to a shared image gallery by using PowerShell. For more information, see the script and ReadMe in Bring image to shared image gallery script.

Create a lab

Create the lab in Lab Services, and select the custom image from the shared image gallery.

Next steps

- Shared image gallery overview
- Attach or detach a shard image gallery
- Use a shared image gallery

Bring a Windows custom image from a physical lab environment

9/7/2021 • 4 minutes to read • Edit Online

The steps in this article show how to import a custom image that starts from your physical lab environment. With this approach, you create a VHD from your physical environment and import the VHD into a shared image gallery so that it can be used within Lab Services. Before you use this approach for creating a custom image, read the article Recommended approaches for creating custom images to decide the best approach for your scenario.

Prerequisites

You will need permission to create an Azure managed disk in your school's Azure subscription to complete the steps in this article.

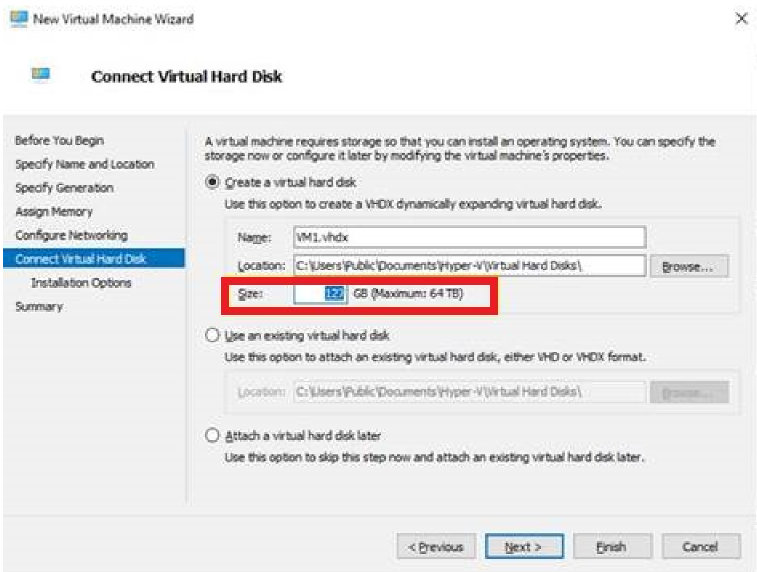
When moving images from a physical lab environment to Lab Services, you should restructure each image so that it only includes software needed for a lab's class. For more information, read the Moving from a Physical Lab to Azure Lab Services blog post.

Prepare a custom image using Hyper-V Manager

The following steps show how to create a Windows image from a Windows Hyper-V virtual machine (VM) using Hyper-V Manager:

1. Start with a Hyper-V VM in your physical lab environment that has been created from your image. Read the article on how to create a virtual machine in Hyper-V for more information.

   - The VM must be created as a **Generation 1** VM.
   - Use the **Default Switch** network configuration option to allow the VM to connect to the internet.
   - The VM's virtual disk must be a fixed size VHD. The disk size must *not* be greater than 128 GB. When you create the VM, enter the size of the disk as shown in the below image.

   Images with disk size greater than 128 GB are *not* supported by Lab Services.

2. Connect to the Hyper-V VM and prepare it for Azure by following these steps:

   a. Set Windows configurations for Azure.

   b. Check the Windows Services that are needed to ensure VM connectivity.

   c. Update remote desktop registry settings.

   d. Configure Windows Firewall rules.

   e. Install Windows Updates.

   f. Install Azure VM Agent and additional configuration as shown here

   You can upload either specialized or generalized images to a shared image gallery and use them to create labs. The steps above will create a specialized image. If you need to instead create a generalized image, you also will need to run SysPrep.

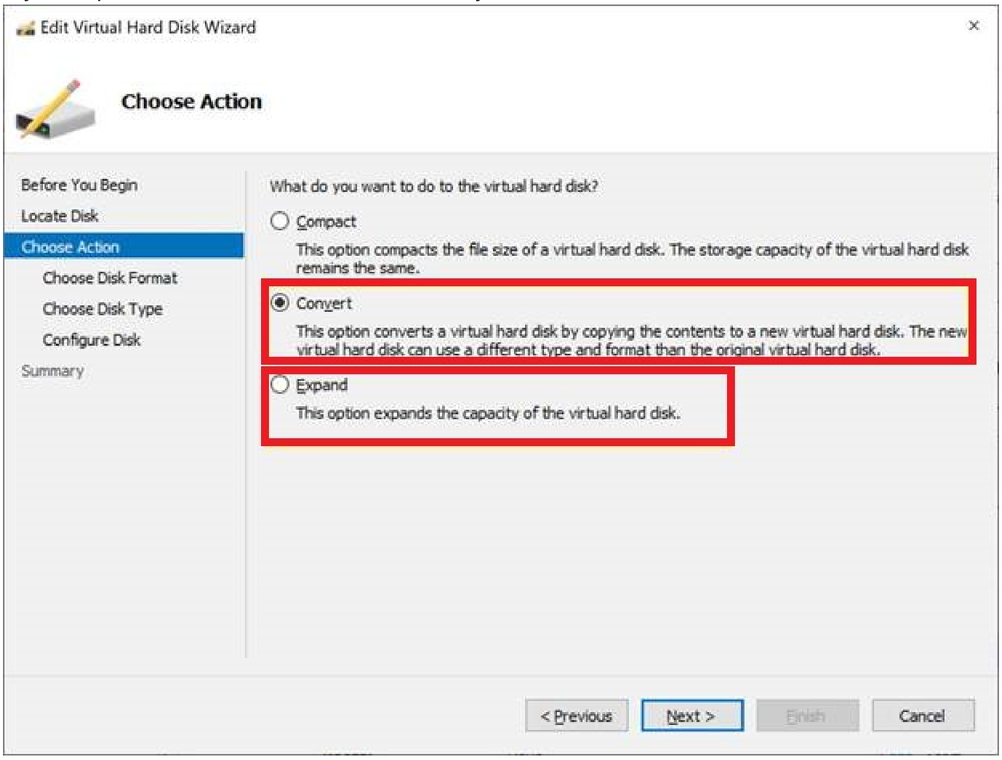
   You should create a specialized image if you want to maintain machine-specific information and user profiles. For more information about the differences between generalized and specialized images, see Generalized and specialized images.

3. Since **Hyper-V** creates a **VHDX** file by default, you need to convert this to a VHD file.

   a. Navigate to **Hyper-V Manager** -> **Action** -> **Edit Disk**.

   b. Next, **Convert** the disk from a VHDX to a VHD.

   - If you expand the disk size, make sure that you do *not* exceed 128 GB.



   For more information, read the article that shows how to convert the virtual disk to a fixed size VHD.

To help with resizing the VHD and converting to a VHDX, you can also use the following PowerShell cmdlets:

- Resize-VHD
- Convert-VHD

Upload the custom image to a shared image gallery

1. Upload the VHD to Azure to create a managed disk.

   a. You can use either Storage Explorer or AzCopy from the command line, as shown in Upload a VHD to Azure or copy a managed disk to another region.

b. After you've uploaded the VHD, you should now have a managed disk that you can see in the Azure portal. If your machine goes to sleep or locks, the upload process may get interrupted and fail. Also, make sure after AzCopy completes, that you revoke the SAS access to the disk. Otherwise, when you attempt to create an image from the disk, you will see an error: **Operation 'Create Image' is not supported with disk 'your disk name' in state 'Active Upload'. Error Code: OperationNotAllowed**

The Azure portal's **Size+Performance** tab for the managed disk allows you to change your disk size. As mentioned before, the size must *not* be greater than 128 GB.

2. In a shared image gallery, create an image definition and version:

   a. Create an image definition.

   - Choose **Gen 1** for the **VM generation**.
   - Choose whether you are creating a **specialized** or **generalized** image for the **Operating system state**.

   For more information about the values you can specify for an image definition, see Image definitions.

   You can also choose to use an existing image definition and create a new version for your custom image.

3. Create an image version.

   - The **Version number** property uses the following format: *MajorVersion.MinorVersion.Patch*. When you use Lab Services to create a lab and choose a custom image, the most recent version of the image is automatically used. The most recent version is chosen based on the highest value of MajorVersion, then MinorVersion, then Patch.
   - For the **Source**, choose **Disks and/or snapshots** from the drop-down list.
   - For the **OS disk** property, choose the disk that you created in previous steps.

   For more information about the values you can specify for an image definition, see Image versions.

Create a lab

1. Create the lab in Lab Services and select the custom image from the shared image gallery.

   If you expanded the disk *after* the OS was installed on the original Hyper-V VM, you may also need to extend the C drive in Windows to use the unallocated disk space:

   - Log into the lab's template VM and follow steps similar to what is shown in Extend a basic volume.

Next steps

- Shared image gallery overview
- Attach or detach a shared image gallery
- How to use a shared image gallery

Bring a Linux custom image from an Azure virtual machine

9/3/2021 • 2 minutes to read • Edit Online

The steps in this article show how to import a custom image that starts from an Azure virtual machine (VM). With this approach, you set up an image on an Azure VM and import the image into a shared image gallery so that it can be used within Azure Lab Services. Before you use this approach for creating a custom image, read Recommended approaches for creating custom images to decide the best approach for your scenario.

Prerequisites

You'll need permission to create an Azure VM in your school's Azure subscription to complete the steps in this article.

Prepare a custom image on an Azure VM

1. Create an Azure VM by using the Azure portal, PowerShell, the Azure CLI, or an Azure Resource Manager template.
   - When you specify the disk settings, ensure the disk's size is *not* greater than 128 GB.
2. Install software and make any necessary configuration changes to the Azure VM's image.
3. Optionally, you can generalize the image. If you decide to create a generalized image, follow the steps outlined in Step 1: Deprovision the VM. When you use the **-deprovision+user** command, it generalizes the image. But it doesn't guarantee that the image is cleared of all sensitive information or that it's suitable for redistribution.

   Otherwise, if you decide to create a specialized image, you can skip to the next step.

   Create a specialized image if you want to maintain machine-specific information and user profiles. For more information about the differences between generalized and specialized images, see Generalized and specialized images.

Import the custom image into a shared image gallery

1. In a shared image gallery, create an image definition or choose an existing image definition.
   - Choose **Gen 1** for the **VM generation**.
   - Choose whether you're creating a **specialized** or **generalized** image for the **Operating system state**.

   For more information about the values you can specify for an image definition, see Image definitions.

   You can also choose to use an existing image definition and create a new version for your custom image.
2. Create an image version.
   - The **Version number** property uses the following format: *MajorVersion.MinorVersion.Patch*.
   - For the **Source**, select **Disks and/or snapshots** from the dropdown list.
   - For the **OS disk** property, choose your Azure VM's disk that you created in previous steps.

You can also automate the preceding steps by using PowerShell. For more information, see the script and

ReadMe in Bring image to a shared image gallery script.

Create a lab

Create the lab in Lab Services, and select the custom image from the shared image gallery.

Next steps

- Shared image gallery overview
- Attach or detach a shard image gallery
- Use a shared image gallery

Bring a Linux custom image from your physical lab environment

9/7/2021 • 5 minutes to read • Edit Online

The steps in this article show how to import a Linux custom image that starts from your physical lab environment. With this approach, you create a VHD from your physical environment and import the VHD into a shared image gallery so that it can be used within Azure Lab Services. Before you use this approach for creating a custom image, read Recommended approaches for creating custom images to decide which approach is best for your scenario.

Azure endorses a variety of distributions and versions. The steps to bring a custom Linux image from a VHD varies for each distribution. Every distribution is different because each one has unique prerequisites that must be set up to run on Azure.

In this article, we'll show the steps to bring a custom Ubuntu 16.04\18.04\20.04 image from a VHD. For information on using a VHD to create custom images for other distributions, see Generic steps for Linux distributions.

Prerequisites

You'll need permission to create an Azure managed disk in your school's Azure subscription to complete the steps in this article.

When you move images from a physical lab environment to Lab Services, restructure each image so that it only includes software needed for a lab's class. For more information, read the Moving from a Physical Lab to Azure Lab Services blog post.

Prepare a custom image by using Hyper-V Manager

The following steps show how to create an Ubuntu 16.04\18.04\20.04 image from a Hyper-V virtual machine (VM) by using Windows Hyper-V Manager.

1. Download the official Linux Ubuntu Server image to your Windows host machine that you'll use to set up the custom image on a Hyper-V VM.

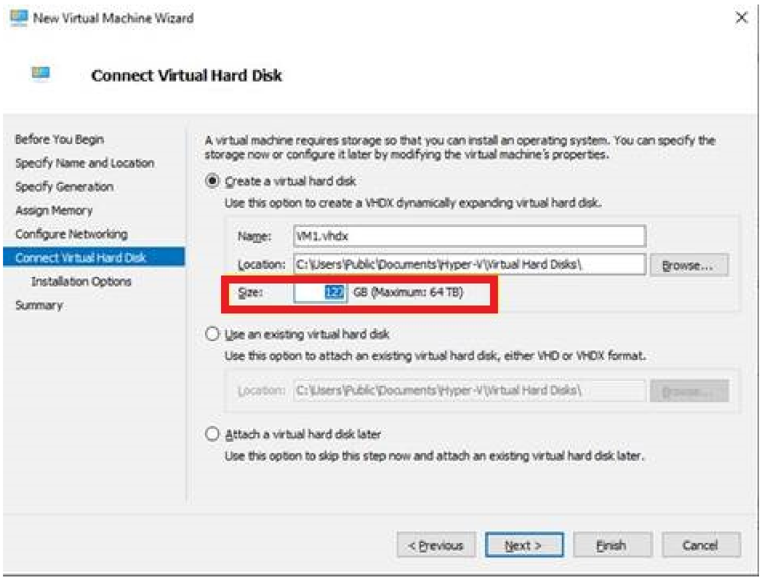
   We recommend using an Ubuntu image that does *not* have the GNOME GUI desktop installed. GNOME currently has a conflict with the Azure Linux Agent, which is needed for the image to work properly in Lab Services. For example, use the Ubuntu Server image and install a different GUI desktop, such as XFCE or MATE.

   Ubuntu also publishes prebuilt Azure VHDs for download. These VHDs are intended for creating custom images from a Linux host machine and hypervisor, such as KVM. These VHDs require that you first set the default user password, which can only be done by using Linux tooling, such as qemu, which isn't available for Windows. As a result, when you create a custom image by using Windows Hyper-V, you won't be able to connect to these VHDs to make image customizations. For more information about the prebuilt Azure VHDs, read Ubuntu's documentation.

2. Start with a Hyper-V VM in your physical lab environment that was created from your image. For more information, read the article on how to create a virtual machine in Hyper-V. Set the settings as shown here:

   - The VM must be created as a **Generation 1** VM.

- Use the **Default Switch** network configuration option to allow the VM to connect to the internet.
- In the **Connect Virtual Hard Disk** settings, the disk's **Size** must *not* be greater than 128 GB, as shown in the following image.



- In the **Installation Options** settings, select the **.iso** file that you previously downloaded from Ubuntu.

Images with a disk size greater than 128 GB are *not* supported by Lab Services.

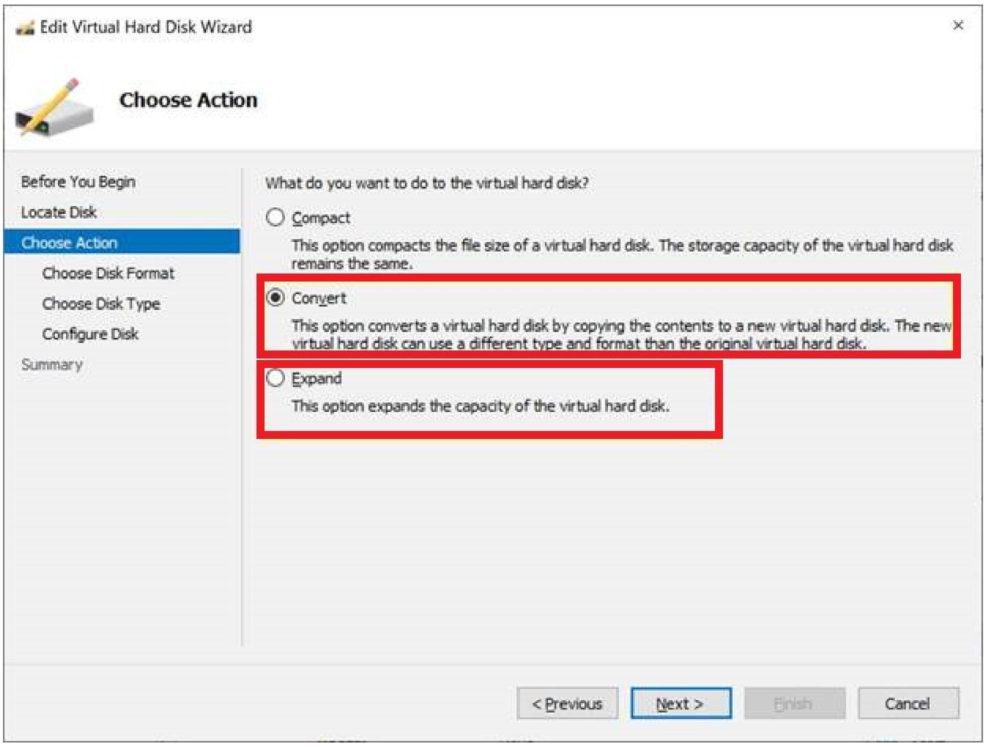
3. Connect to the Hyper-V VM and prepare it for Azure by following the steps in Manual steps to create and upload an Ubuntu VHD.

   The steps to prepare a Linux image for Azure vary based on the distribution. For more information and specific steps for each distribution, see distributions and versions.

   When you follow the preceding steps, there are a few important points to highlight:

   - The steps create a generalized image when you run the **deprovision+user** command. But it doesn't guarantee that the image is cleared of all sensitive information or that it's suitable for redistribution.
   - The final step is to convert the **VHDX** file to a **VHD** file. Here are equivalent steps that show how to do it with **Hyper-V Manager**:

     a. Go to **Hyper-V Manager** > **Action** > **Edit Disk**.
     b. Next, **Convert** the disk from a VHDX to a VHD.
     c. For the **Disk Type**, select **Fixed size**.
        - If you also choose to expand the disk size at this point, make sure that you do *not* exceed 128 GB.



To help with resizing the VHD and converting to a VHDX, you can also use the following PowerShell cmdlets:

- Resize-VHD
- Convert-VHD

Upload the custom image to a shared image gallery

1. Upload the VHD to Azure to create a managed disk.

   a. You can use either Storage Explorer or AzCopy from the command line, as shown in Upload a VHD to Azure or copy a managed disk to another region.

   b. After you've uploaded the VHD, you should now have a managed disk that you can see in the Azure portal.

   If your machine goes to sleep or locks, the upload process might get interrupted and fail. Also, make sure that when AzCopy completes, that you revoke SAS access to the disk. Otherwise, when you attempt to create an image from the disk, you'll see the error "Operation 'Create Image' is not supported with disk 'your disk name' in state 'Active Upload'. Error Code: OperationNotAllowed*."

   Use the Azure portal's **Size+Performance** tab for the managed disk to change your disk size. As mentioned before, the size must *not* be greater than 128 GB.

2. In a shared image gallery, create an image definition and version:

   a. Create an image definition:

   - Choose **Gen 1** for the **VM generation**.
   - Choose **Linux** for the **Operating system**.
   - Choose **generalized** for the **Operating system state**.

   For more information about the values you can specify for an image definition, see Image definitions.

   You can also choose to use an existing image definition and create a new version for your custom image.

3. Create an image version:

   - The **Version number** property uses the following format: *MajorVersion.MinorVersion.Patch*. When you use Lab Services to create a lab and choose a custom image, the most recent version of the image is automatically used. The most recent version is chosen based on the highest value of MajorVersion, then MinorVersion, and then Patch.

- For the **Source**, select **Disks and/or snapshots** from the dropdown list.
- For the **OS disk** property, choose the disk that you created in previous steps.

For more information about the values you can specify for an image definition, see Image versions.

Create a lab

Create the lab in Lab Services and select the custom image from the shared image gallery.

If you expanded the disk *after* the OS was installed on the original Hyper-V VM, you might also need to extend the partition in Linux's filesystem to use the unallocated disk space. Log in to the lab's template VM and follow steps similar to what is shown in Expand a disk partition and filesystem.

The OS disk typically exists on the `/dev/sad2` partition. To view the current size of the OS disk's partition, use the command `df -h`.

Next steps

- Shared image gallery overview
- Attach or detach a shard image gallery
- Use a shared image gallery

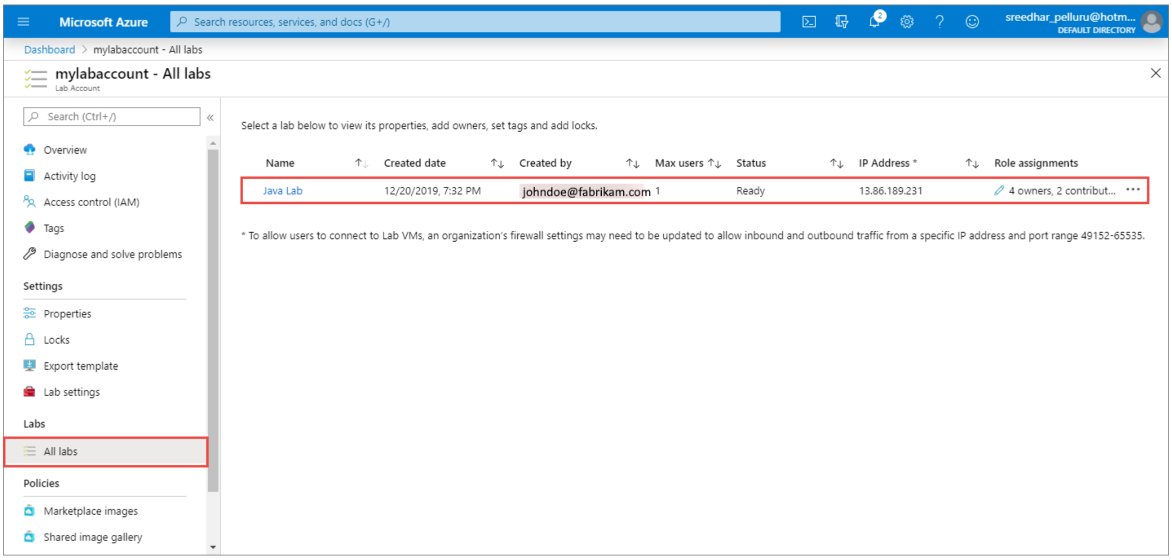
How to add additional owners to an existing lab in Azure Lab Services

8/20/2021 • 2 minutes to read • Edit Online

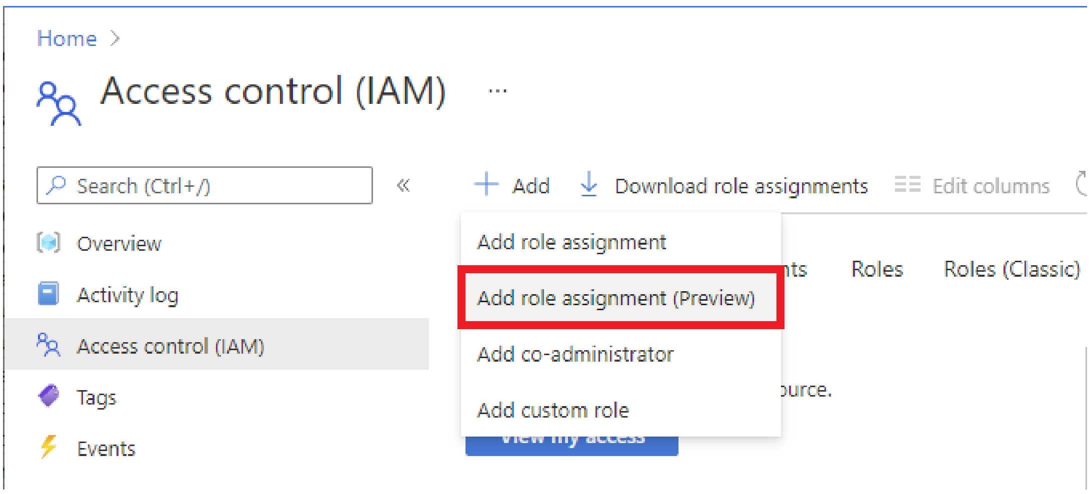
This article shows you how you, as an administrator, can add additional owners to an existing lab.

Add user to the reader role for the lab account

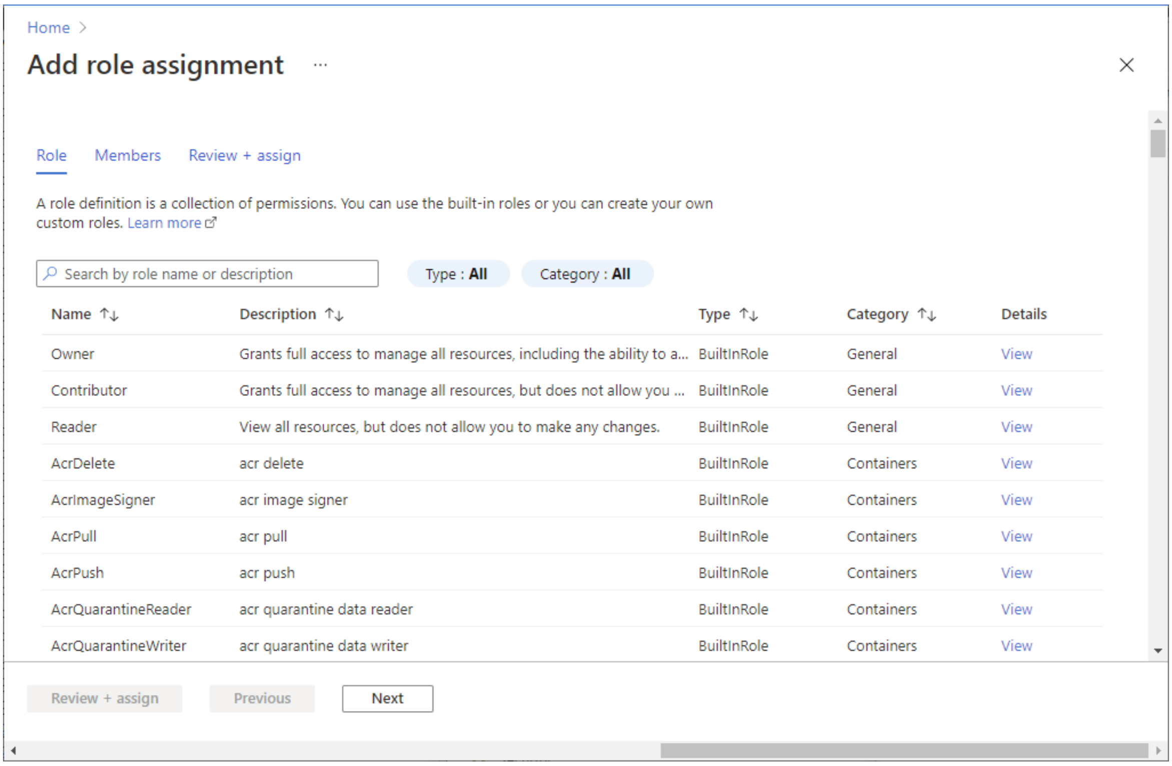
1. Back on the **Lab Account** page, select **All labs** on the left menu.
2. Select the **lab** to which you want to add user as an owner.



3. In the navigation menu, select **Access control (IAM)**.
4. Select **Add** > **Add role assignment (Preview)**.



5. On the **Role** tab, select the **Reader** role.



6. On the **Members** tab, select the user you want to add to the Reader role.

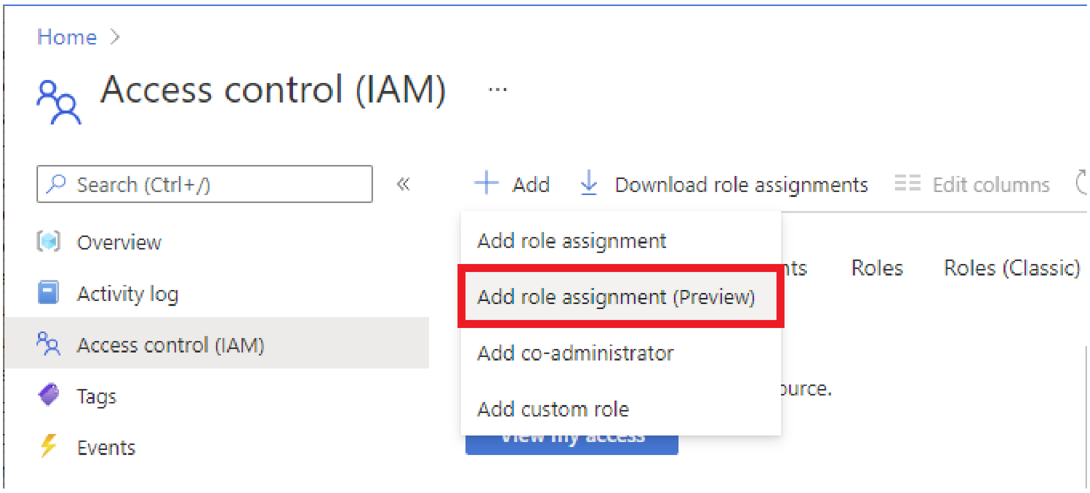
7. On the **Review + assign** tab, select **Review + assign** to assign the role.

Add user to the owner role for the lab

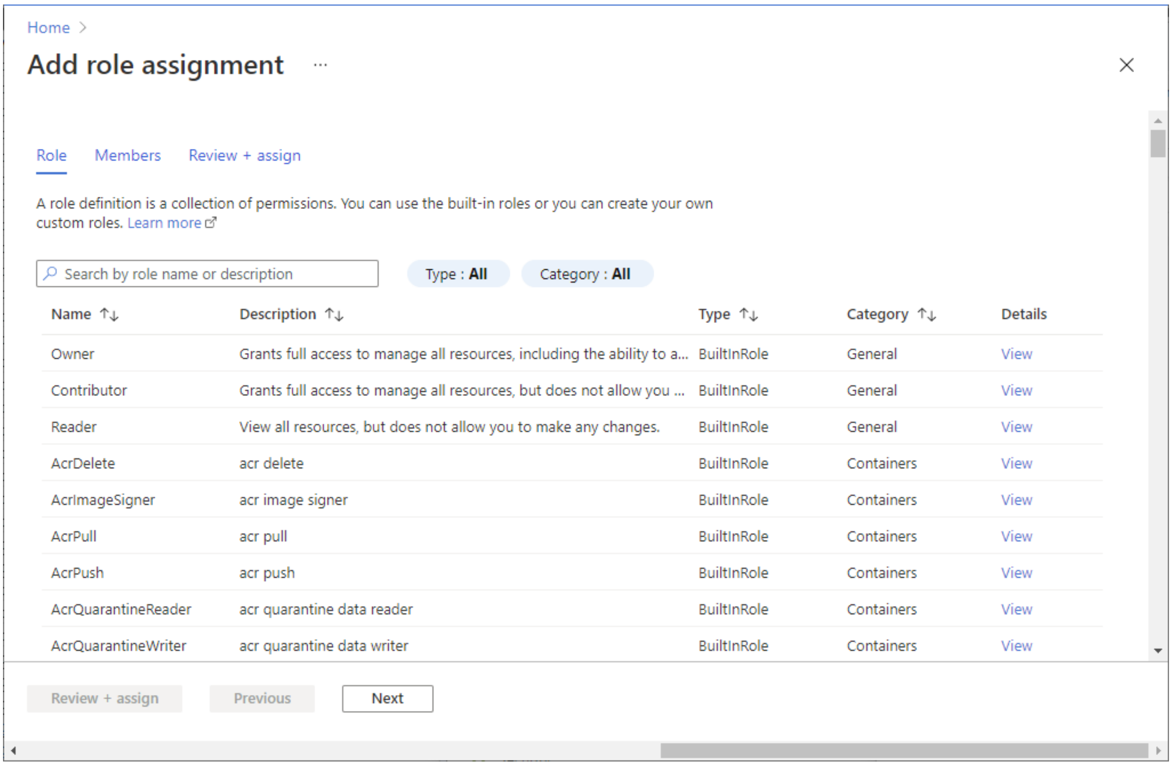
> **NOTE**
>
> If the user has only Reader access on the a lab, the lab isn't shown in labs.azure.com. For detailed steps, see Assign Azure roles using the Azure portal.

1. On the **Lab Account** page, select **Access control (IAM)**

2. Select **Add** > **Add role assignment (Preview)**.



3. On the **Role** tab, select the **Owner** role.



4. On the **Members** tab, select the user you want to add to the Owner's role
5. On the **Review + assign** tab, select **Review + assign** to assign the role.

Next steps

Confirm that the user sees the lab upon logging into the Lab Services portal.

Firewall settings for Azure Lab Services

5/5/2021 • 2 minutes to read • Edit Online

Each organization or school will set up their own network in a way that best fits their needs. Sometimes that includes setting firewall rules that block Remote Desktop Protocol (rdp) or Secure Shell (ssh) connections to machines outside their own network. Because Azure Lab Services runs in the public cloud, some extra configuration maybe needed to allow students to access their VM when connecting from the campus network.

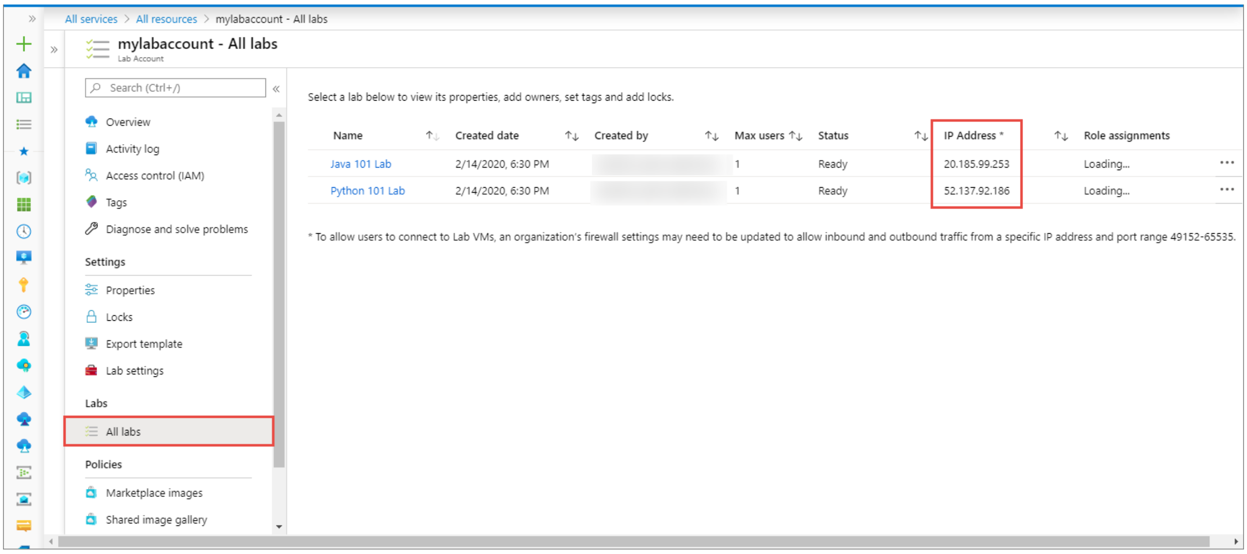
Each lab uses single public IP address and multiple ports. All VMs, both the template VM and student VMs, will use this public IP address. The public IP address will not change for the life of lab. However, each VM will have a different port number. The port numbers range from 49152 to 65535. The combination of public IP address and port number is used to connect instructor and students to the correct VM. This article will cover how to find the specific public IP address used by a lab. That information can be used to update inbound and outbound firewall rules so students can access their VMs.

> **IMPORTANT**
> Each lab will have a different public IP address.

> **NOTE**
> If your school needs to perform content filtering, such as for compliance with the Children's Internet Protection Act (CIPA), you will need to use 3rd party software. For more information, read guidance on content filtering with Lab Services.

Find public IP for a lab

The public IP addresses for each lab are listed in the **All labs** page of the Lab Services lab account. For directions how to find the **All labs** page, see View labs in a lab account.



> **NOTE**
> You won't see the public IP address if the template machine for your lab isn't published yet.

Conclusion

Now we know the public IP address for the lab. Inbound and outbound rules can be created for the organization's firewall for the public ip address and the port range 49152-65535. Once the rules are updated, students can access their VMs without the network firewall blocking access.

Next steps

See the following articles:

- Allow lab creator to pick lab location
- Connect your lab's network with a peer virtual network
- Attach a shared image gallery to a lab
- Add a user as a lab owner
- View firewall settings for a lab
- Configure other settings for a lab

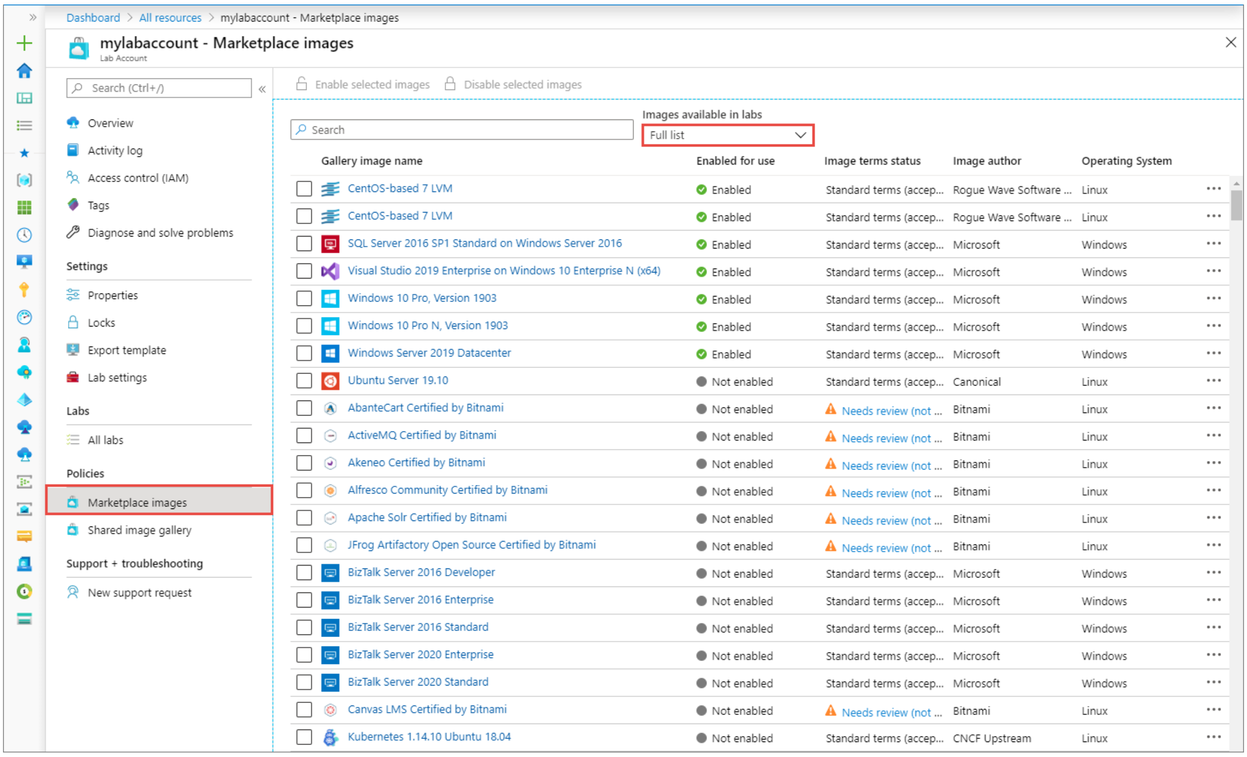
Specify Marketplace images available to lab creators

3/5/2021 • 2 minutes to read • Edit Online

As a lab account owner, you can specify the Marketplace images that lab creators can use to create labs in the lab account.

Select images available for labs

Select **Marketplace images** on the menu to the left. By default, you see the full list of images (both enabled and disabled). You can filter the list to see only enabled/disabled images by selecting the **Enabled only**/**Disabled only** option from the drop-down list at the top.

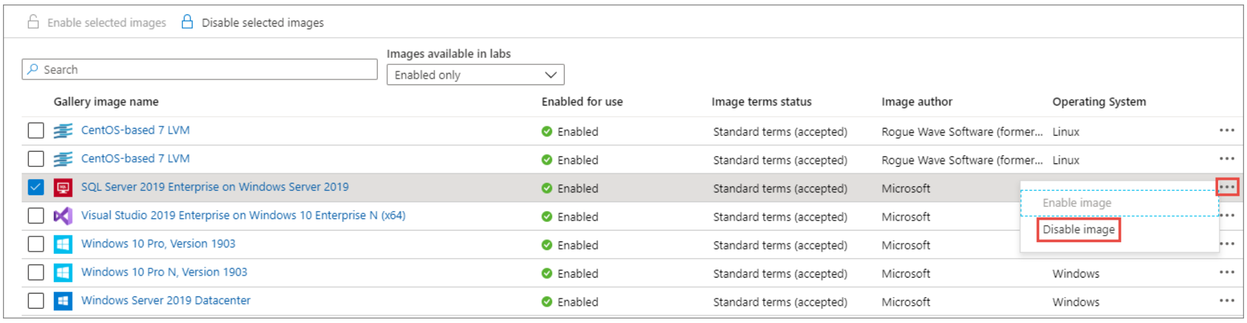


The Marketplace images that are displayed in the list are only the ones that satisfy the following conditions:

- Creates a single VM.
- Uses Azure Resource Manager to provision VMs
- Doesn't require purchasing an extra licensing plan

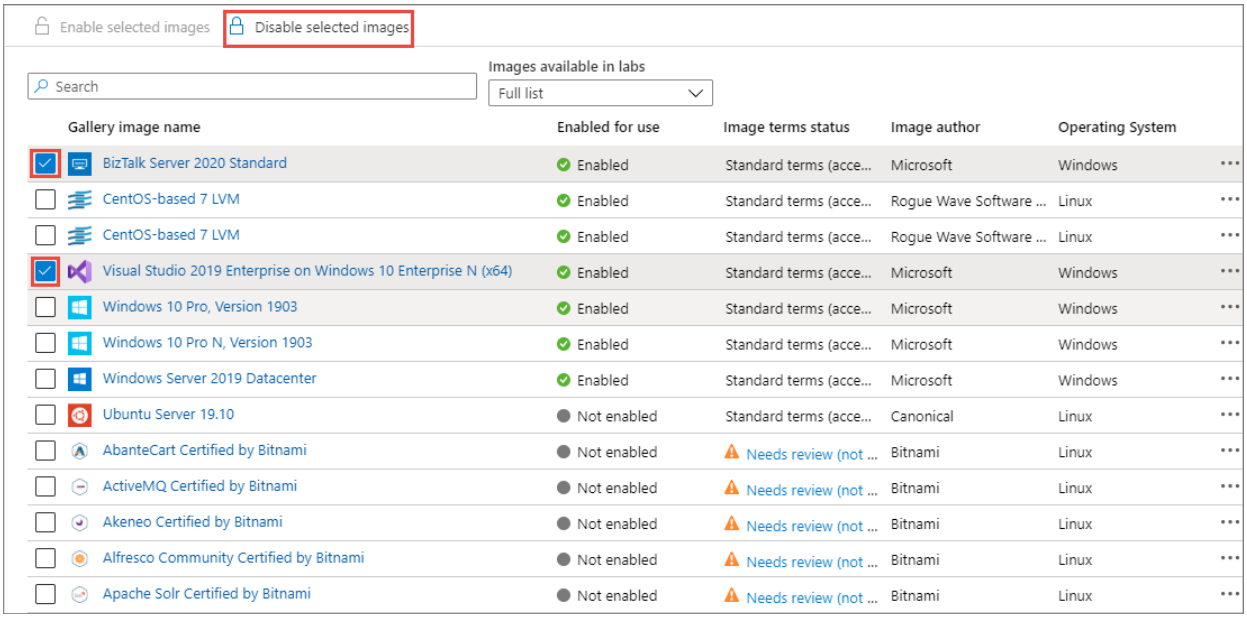
Disable images for a lab

To disable a single image for a lab, select **... (ellipsis)** in the last column, and select **Disable image**.



Alternatively, you select the checkbox before the image name, and select **Disable selected images** on the toolbar.

To disable multiple images at the same time, select checkboxes before the image names, and select **Disable selected images** on the toolbar.



Enable images for a lab

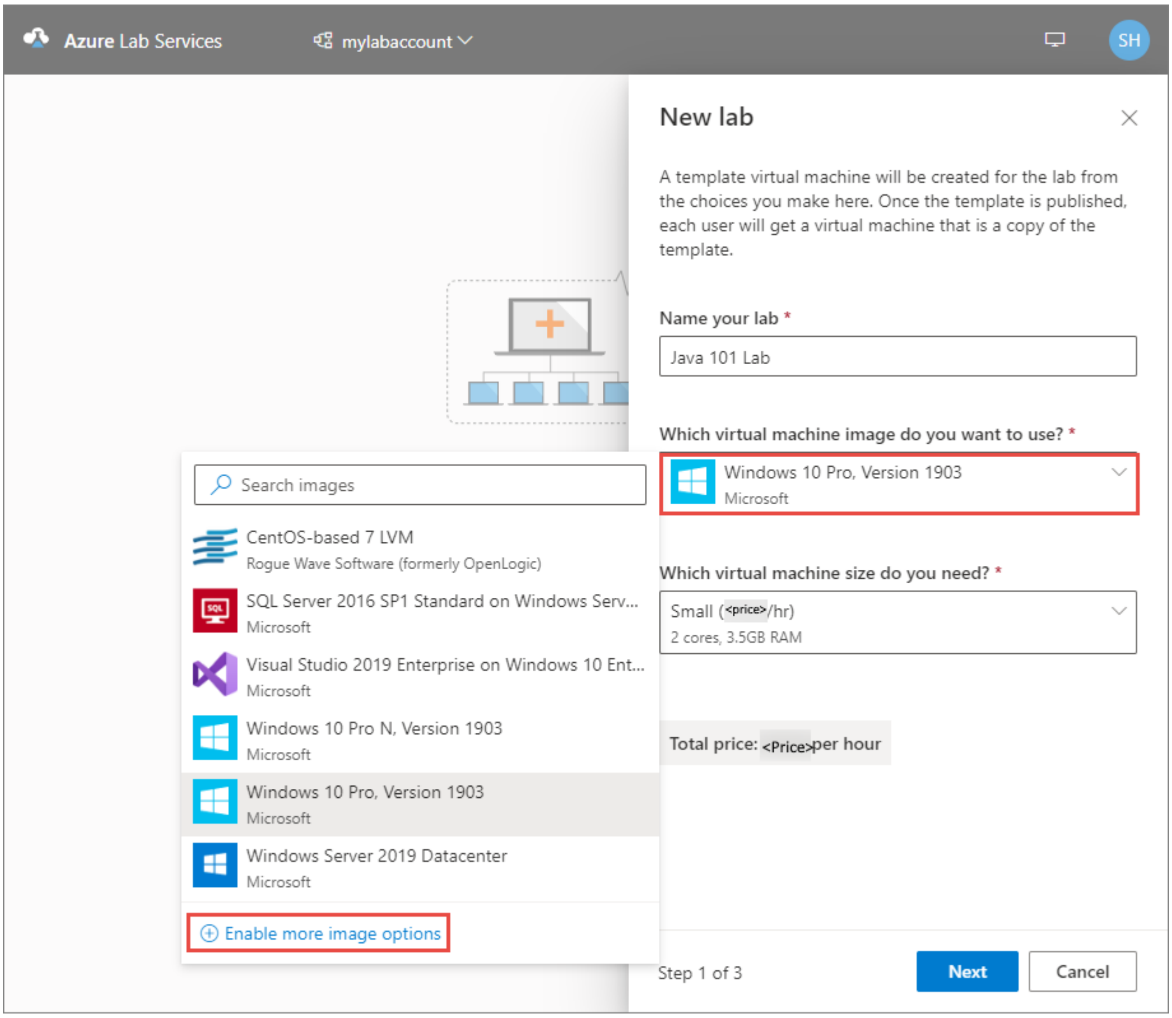
To enable a disabled image, select **... (ellipsis)** in the last column, and select **Enable image**. Alternatively, you select the checkbox before the image name, and select **Enable selected images** on the toolbar.

To disable multiple images at the same time, select checkboxes before the image names, and select **Enable selected images** on the toolbar.

Enable images at the time of lab creation

You can enable more images at the time lab creation:

1. Sign in to the Azure Lab Services website using **lab account owner** credentials
2. Select the default virtual machine image or the down arrow.
3. Select **Enable more image options**.



4. Follow instructions from the previous section to enable the images you select.

5. You may need to close the **New lab** window and reopen it to see the images you selected in the previous step.

Next steps

See the following articles:

- As a lab owner, create and manage labs
- As a lab owner, set up and publish templates
- As a lab owner, configure and control usage of a lab
- As a lab user, access labs

Set up support information (lab account owner in Azure Lab Services)

11/2/2020 • 2 minutes to read • Edit Online

This article explains how you (as a lab account owner) can specify support information that lab creators (educators) and lab users (students) can use to get help if they run into any technical issues with using the lab account or lab.

The support information includes:

- URL
- Email
- Phone
- Additional instructions

Specify support information

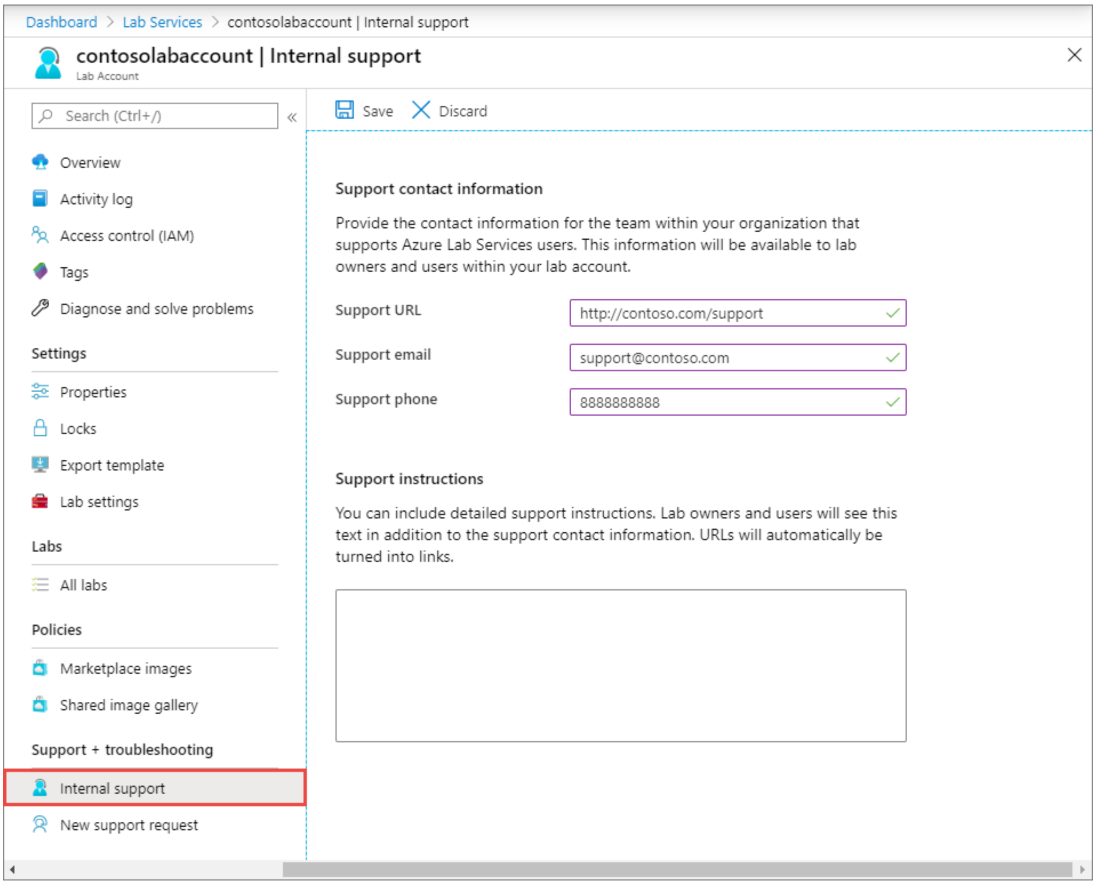
1. Sign in to Azure portal.
2. In the search bar, enter **Lab Services**, and select **Lab Services** in the search results.
3. Select your lab account from the list of lab accounts.
4. Switch to the **Internal support** page, do the following steps:
   a. Enter the **support URL**.

   b. Enter the **support email**.

   c. Enter the **support phone**.

   d. Enter detailed **support instructions** (optional). Lab owners and users will see this text along with the support contact information. URLs will be automatically turned into links.

   e. Select **Save** on the toolbar.



Next steps

See the following articles:

- View contact information (lab creator)
- View contact information (lab user)

Manage labs in Azure Lab Services

4/14/2021 • 7 minutes to read • Edit Online

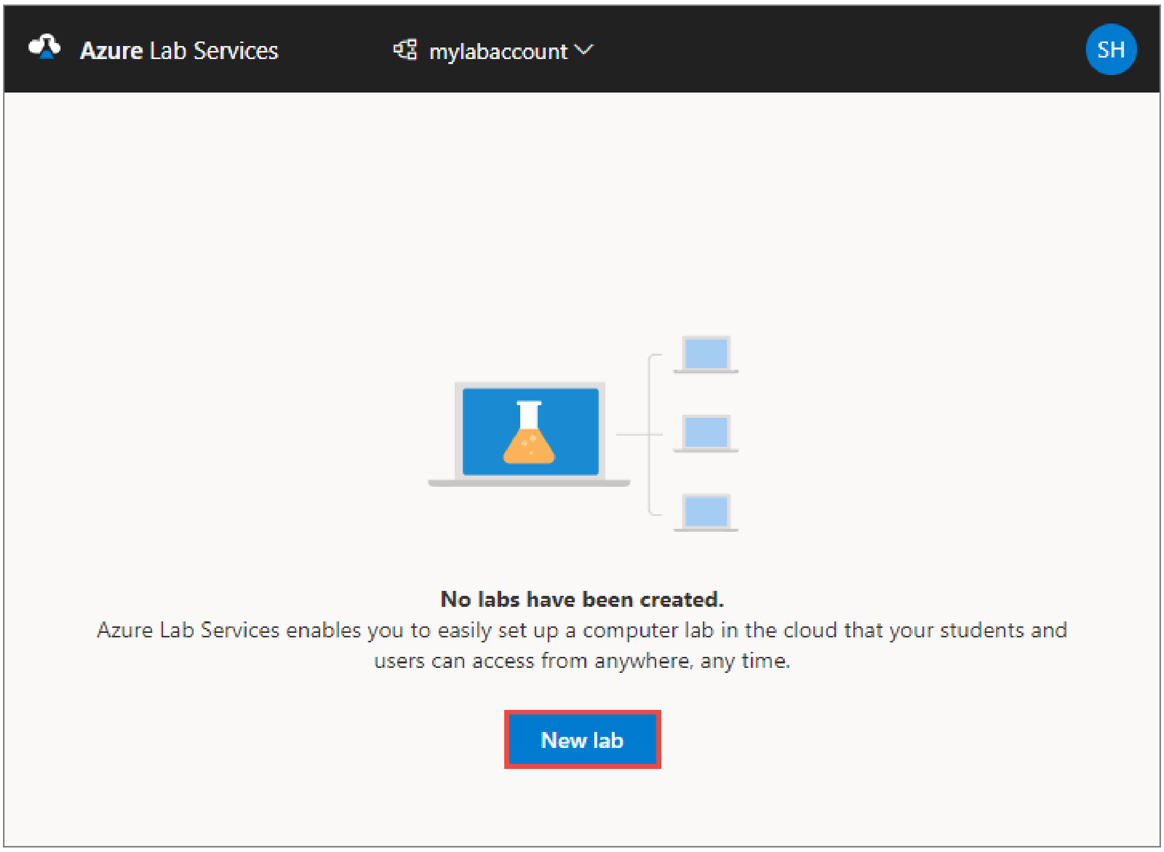
This article describes how to create and delete a classroom lab. It also shows you how to view all the labs in a lab account.

Prerequisites

To set up a classroom lab in a lab account, you must be a member of the **Lab Creator** role in the lab account. The account you used to create a lab account is automatically added to this role. A lab owner can add other users to the Lab Creator role by using steps in the following article: Add a user to the Lab Creator role.

Create a classroom lab

1. Navigate to Azure Lab Services website.
2. Select **Sign in** and enter your credentials. Select or enter a **user ID** that is a member of the **Lab Creator** role in the lab account, and enter password. Azure Lab Services supports organizational accounts and Microsoft accounts.
3. Select **New lab**.



4. In the **New Lab** window, do the following actions:

   a. Specify a **name** for your lab.

   b. Select the **size of the virtual machines** you need for the class. For the list of sizes available, see the VM Sizes section.

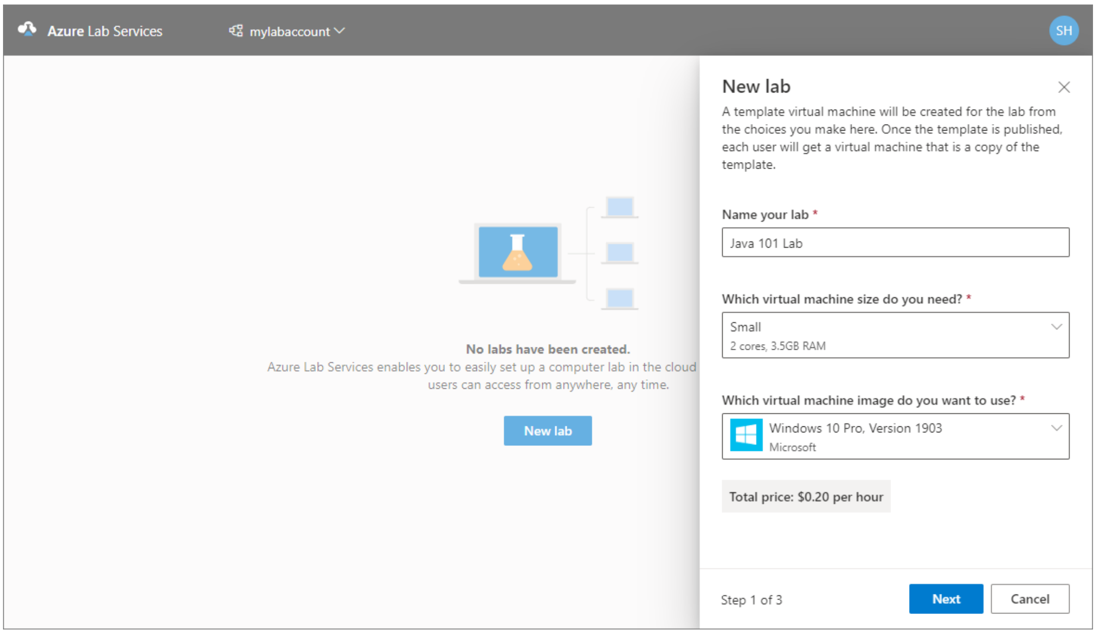
   c. Select the **virtual machine image** that you want to use for the classroom lab. If you select a

Linux image, you see an option to **enable remote desktop connection**. For details, see Enable remote desktop connection for Linux.

If you signed in using lab account owner credentials, you will see an option to enable more images for the lab. For more information, see Enable images at the time of lab creation.

d. Review the **total price per hour** displayed on the page.

e. Select **Save**.



> **NOTE**
>
> You see an option to select a location for your lab if the lab account was configured to allow lab creator to pick lab location option.

5. On the **Virtual machine credentials** page, specify default credentials for all VMs in the lab.

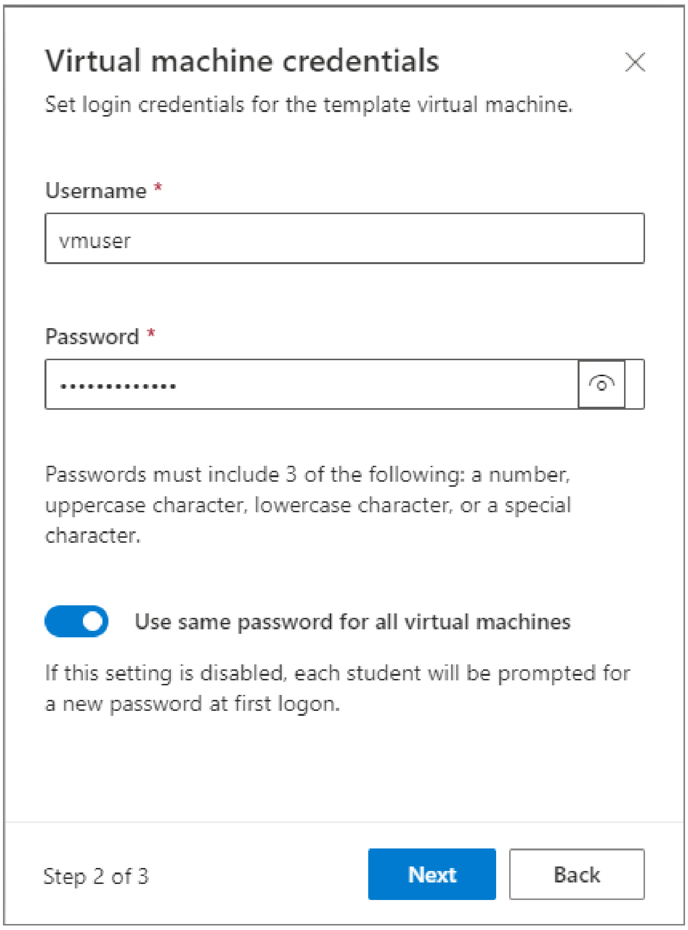
a. Specify the **name of the user** for all VMs in the lab.

b. Specify the **password** for the user.

> **IMPORTANT**
>
> Make a note of user name and password. They won't be shown again.

c. Disable **Use same password for all virtual machines** option if you want students to set their own passwords. This step is **optional**.

An educator can choose to use the same password for all the VMs in the lab, or allow students to set passwords for their VMs. By default, this setting is enabled for all Windows and Linux images except for Ubuntu. When you select **Ubuntu** VM, this setting is disabled, so the students will be prompted to set a password when they sign in for the first time.



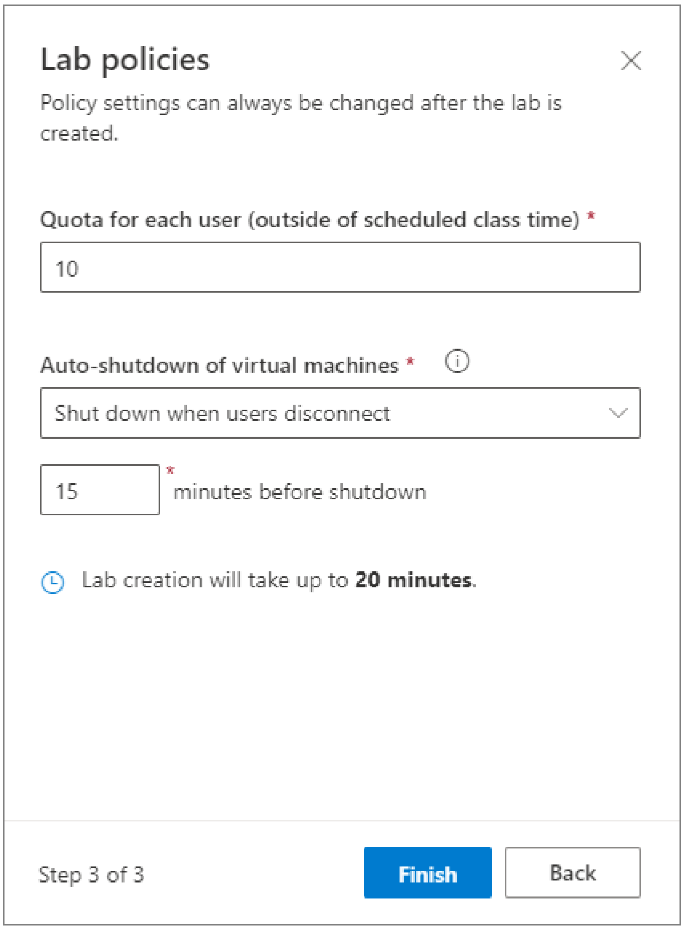
   d. Then, select **Next** on the **Virtual machine credentials** page.

6. On the **Lab policies** page, do the following steps:

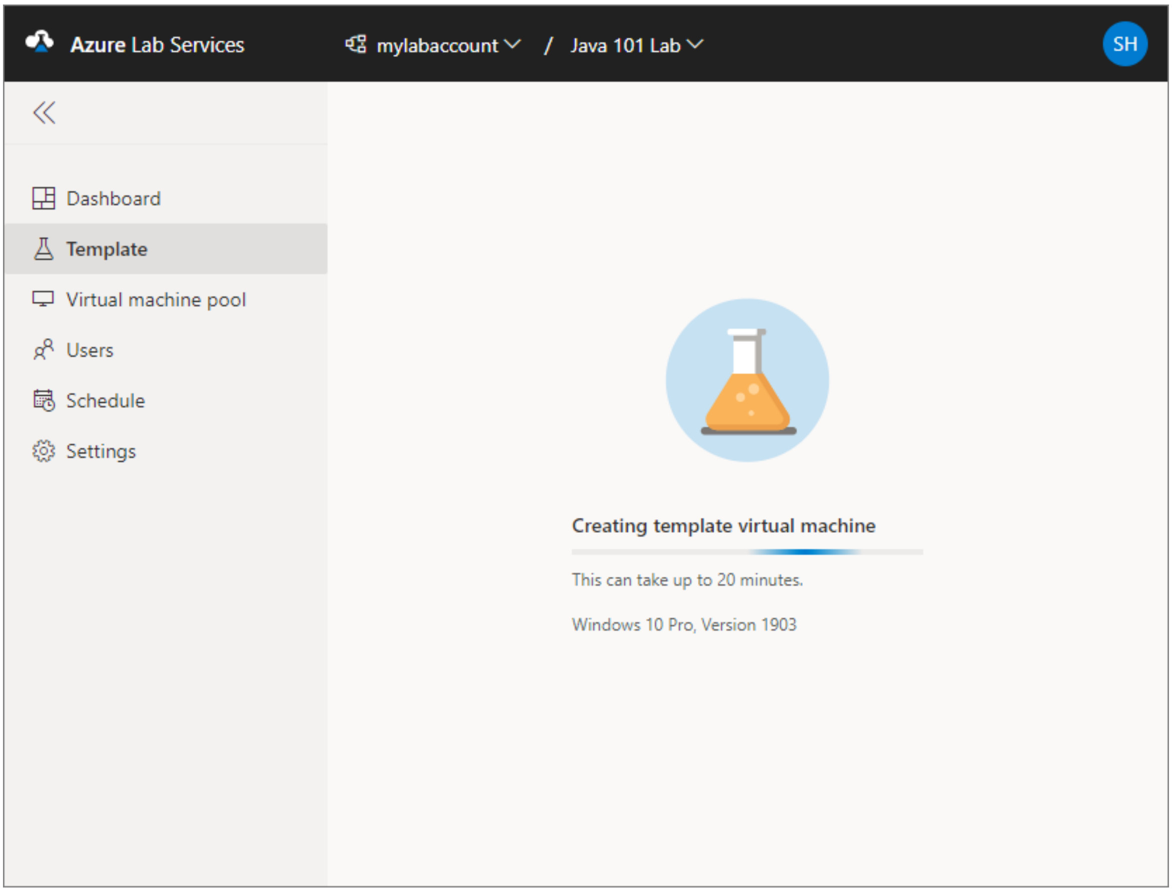
   a. Enter the number of hours allotted for each user (**quota for each user**) outside the scheduled time for the lab.

   b. For the **Auto-shutdown of virtual machines** option, specify whether you want the VM to be automatically shutdown when user disconnects. You can also specify how long the VM should wait for the user to reconnect before automatically shutting down.. For more information, see Enable automatic shutdown of VMs on disconnect.

   c. Then, select **Finish**.



7. You should see the following screen that shows the status of the template VM creation. The creation of the template in the lab takes up to 20 minutes.

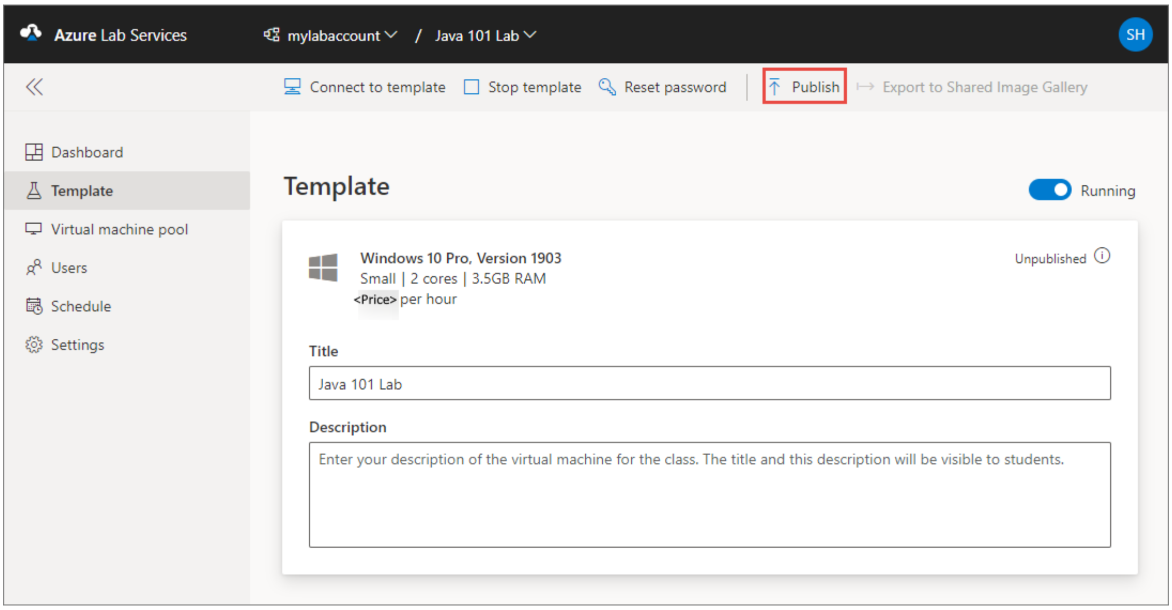


8. On the **Template** page, do the following steps: These steps are **optional** for the tutorial.

   a. Connect to the template VM by selecting **Connect**. If it's a Linux template VM, you choose whether you want to connect using SSH or a GUI remote desktop. Additional setup is required to use a GUI remote desktop. See Enable graphical remote desktop for Linux virtual machines for more information.

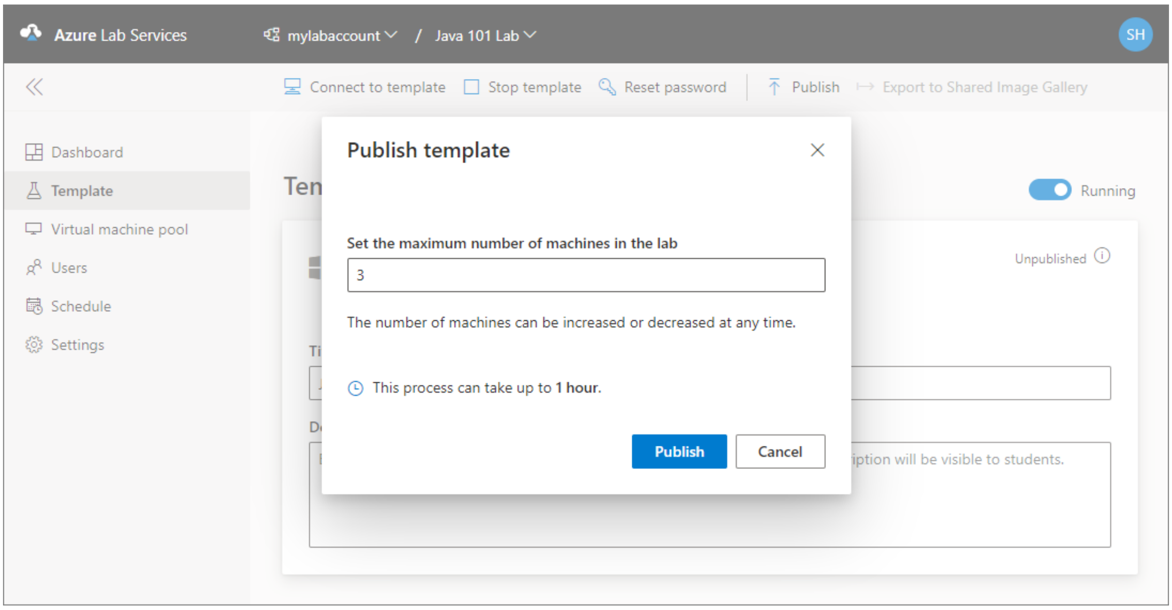
b. Select **Reset password** to reset the password for the VM.
c. Install and configure software on your template VM.
d. **Stop** the VM.
e. Enter a **description** for the template

9. On **Template** page, select **Publish** on the toolbar.

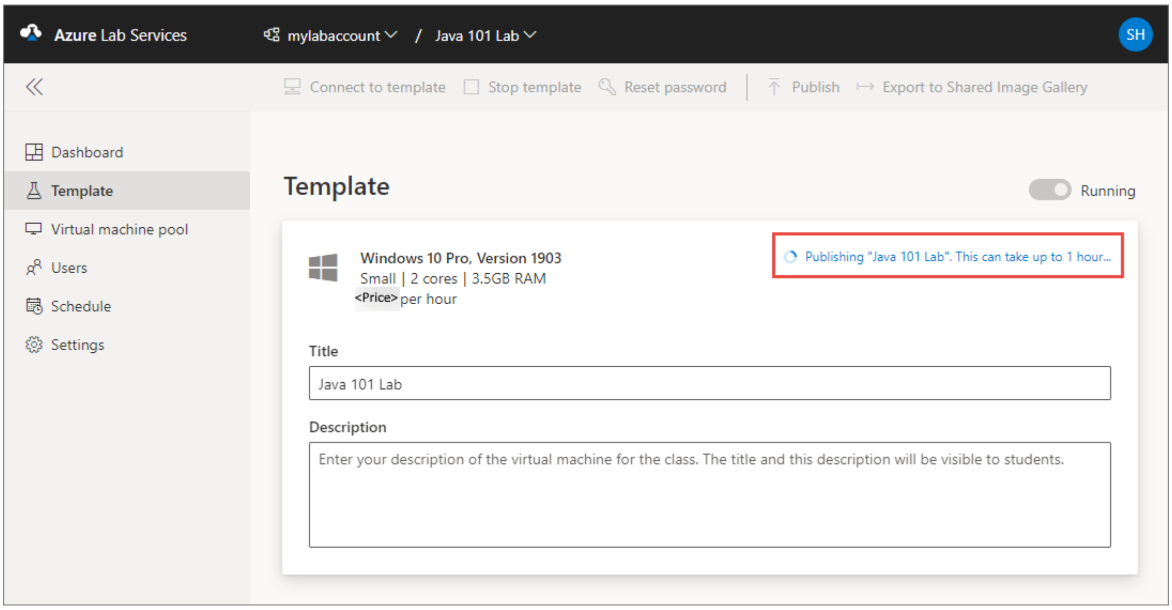


> **WARNING**
> Once you publish, you can't unpublish.

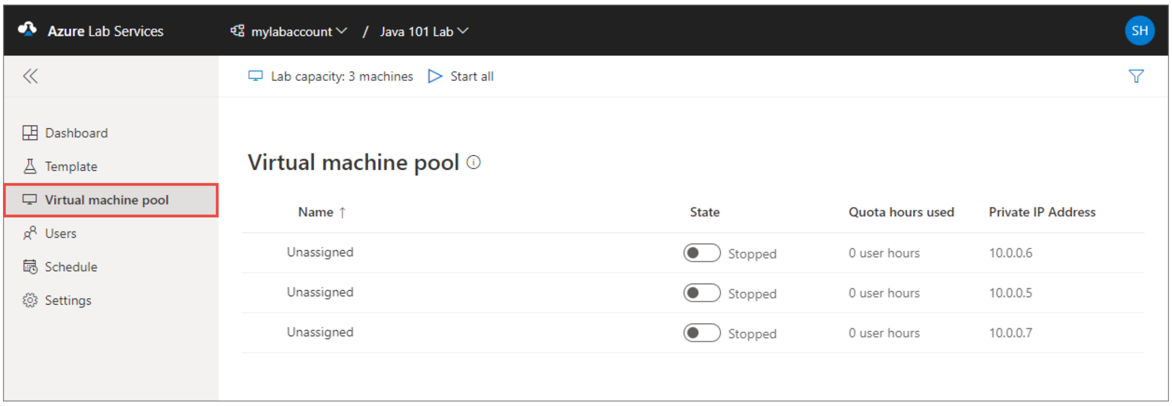
10. On the **Publish template** page, enter the number of virtual machines you want to create in the lab, and then select **Publish**.



11. You see the **status of publishing** the template on page. This process can take up to an hour.



12. Switch to the **Virtual machines pool** page by selecting Virtual machines on the left menu or by selecting Virtual machines tile. Confirm that you see virtual machines that are in **Unassigned** state. These VMs are not assigned to students yet. They should be in **Stopped** state. You can start a student VM, connect to the VM, stop the VM, and delete the VM on this page. You can start them in this page or let your students start the VMs.



You do the following tasks on this page (don't do these steps for the tutorial. These steps are for your information only.):

a. To change the lab capacity (number of VMs in the lab), select **Lab capacity** on the toolbar.
b. To start all the VMs at once, select **Start all** on the toolbar.
c. To start a specific VM, select the down arrow in the **Status**, and then select **Start**. You can also start a VM by selecting a VM in the first column, and then by selecting **Start** on the toolbar.

VM sizes

SIZE	CORES	RAM	DESCRIPTION
Small	2	3.5 GB	This size is best suited for command line, opening web browser, low traffic web servers, small to medium databases.
Medium	4	7 GB	This size is best suited for relational databases, in-memory caching, and analytics

SIZE	CORES	RAM	DESCRIPTION
Medium (Nested virtualization)	4	16 GB	This size is best suited for relational databases, in-memory caching, and analytics. This size also supports nested virtualization. This size can be used in scenarios where each student needs multiple VMs. Educators can use nested virtualization to set up a few small-size nested virtual machines inside the virtual machine.
Small GPU (Compute)	6	56 GB	This size is best suited for compute-intensive and network-intensive applications like artificial intelligence and deep learning applications. Azure Lab Services automatically installs and configures the necessary GPU drivers for you when you create a lab with GPU images.
Small GPU (Visualization)	6	56 GB	This size is best suited for remote visualization, streaming, gaming, encoding using frameworks such as OpenGL and DirectX.
Large	8	16 GB	This size is best suited for applications that need faster CPUs, better local disk performance, large databases, large memory caches.
Large (Nested virtualization)	8	32 GB	This size is best suited for applications that need faster CPUs, better local disk performance, large databases, large memory caches. This size also supports nested virtualization.

SIZE	CORES	RAM	DESCRIPTION
Medium GPU (Visualization)	12	112 GB	This size is best suited for remote visualization, streaming, gaming, encoding using frameworks such as OpenGL and DirectX.

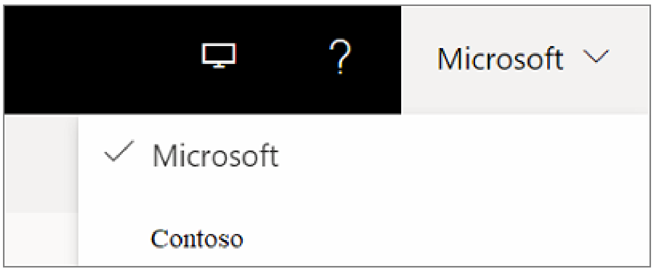
NOTE

You may not see some of these VM sizes in the list when creating a classroom lab. The list is populated based on the current capacity of the lab's location. If the lab account creator allows lab creators to pick a location for the lab, you may try choosing a different location for the lab and see if the VM size is available.

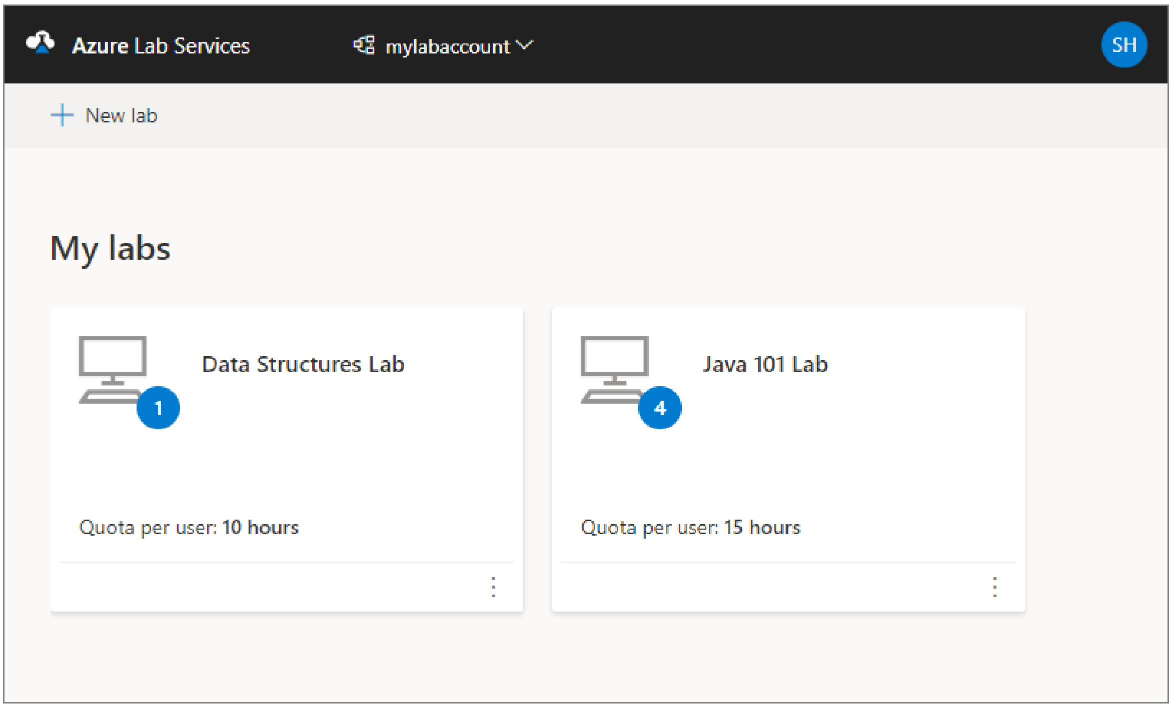
View all labs

1. Navigate to Azure Lab Services portal.
2. Select **Sign in**. Select or enter a **user ID** that is a member of the **Lab Creator** role in the lab account, and enter password. Azure Lab Services supports organizational accounts and Microsoft accounts.

   If you have an Administrator or Lab Owner role in two or more Azure AD tenants, you can switch between tenants in the Azure Lab Services portal by selecting the control at the upper right, as shown in the following screenshot:

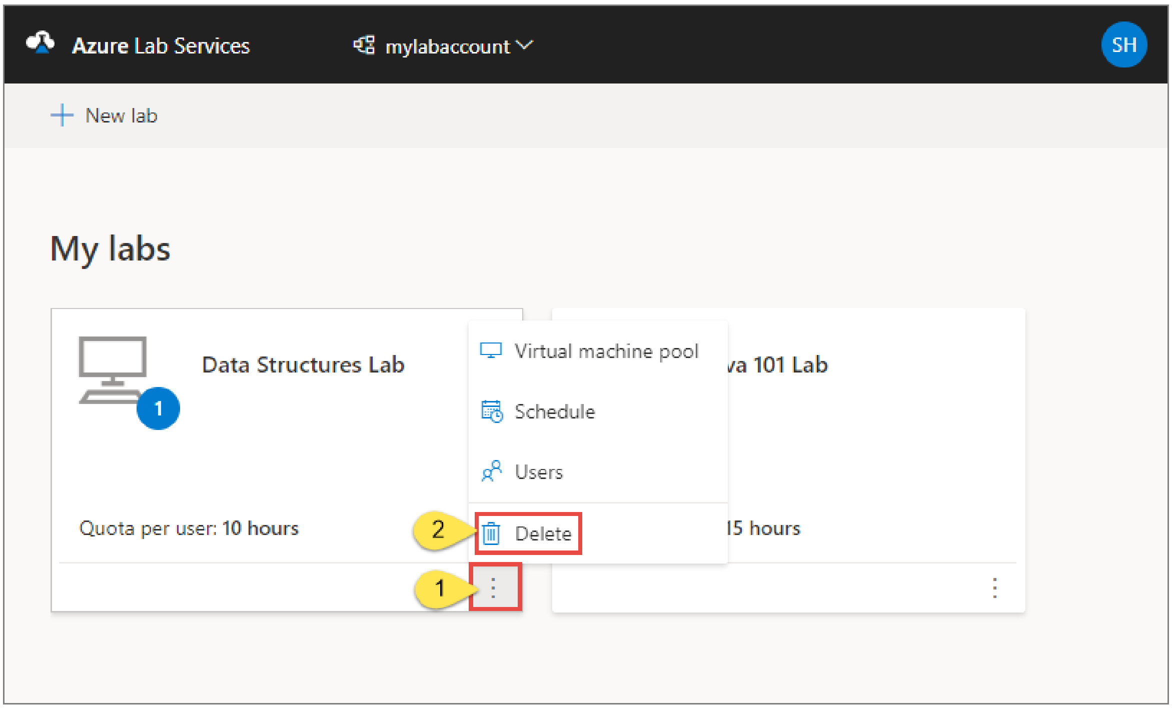
   After you've chosen an Azure AD tenant, go to your lab account by selecting the control at the upper left.
3. Confirm that you see all the labs in the selected lab account. On the lab's tile, you see the number of virtual machines in the lab and the quota for each user (outside the scheduled time).



4. Use the drop-down list at the top to select a different lab account. You see labs in the selected lab account.

Delete a classroom lab

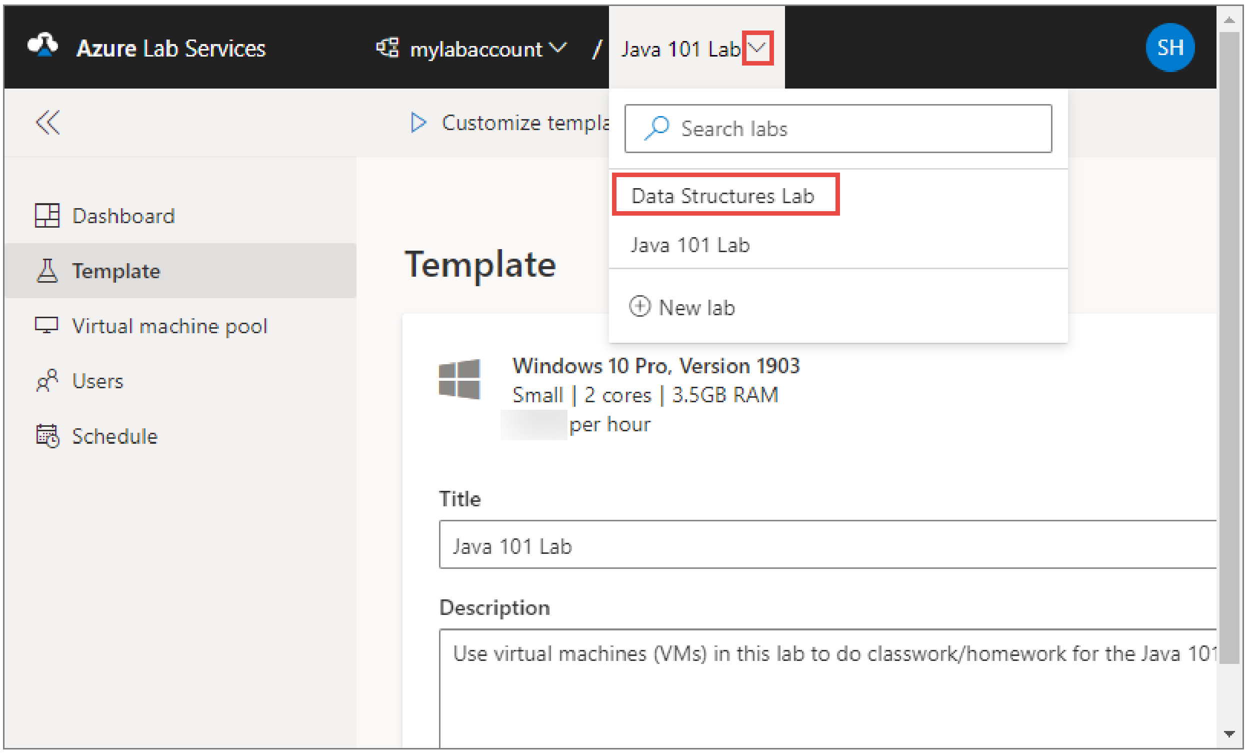
1. On the tile for the lab, select three dots (...) in the corner, and then select **Delete**.



2. On the **Delete lab** dialog box, select **Delete** to continue with the deletion.

Switch to another classroom lab

To switch to another classroom lab from the current, select the drop-down list of labs in the lab account at the top.



You can also create a new lab using the **New lab** in this drop-down list.

> **NOTE**
>
> You can also use the Az.LabServices PowerShell module (preview) to manage labs. For more information, see the Az.LabServices home page on GitHub.

To switch to a different lab account, select the drop-down next to the lab account and select the other lab account.

Next steps

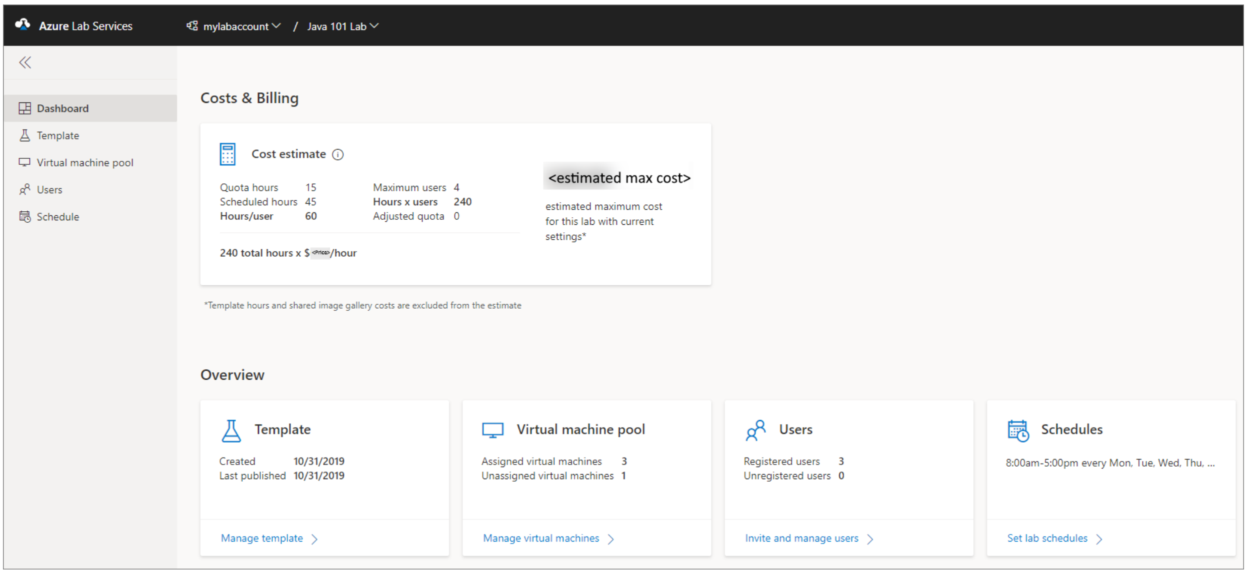
See the following articles:

- As a lab owner, set up and publish templates
- As a lab owner, configure and control usage of a lab
- As a lab user, access labs

Dashboard for labs

3/5/2021 • 2 minutes to read • Edit Online

This article describes the dashboard view of a classroom lab in Azure Lab Services.



Costs and billing tile

This tile provides the following cost estimate details:

SETTING	VALUE
Quota hours	The maximum number of hours a user can use the VM outside the scheduled hours.
Scheduled hours	Hours that will be incurred based on the schedule set in the lab. This value is only available if there is a from/to date set on all the schedule events.
Hours/user	The sum of quota hours and scheduled hours.
Maximum users	Maximum number of users in the lab based on all virtual machines to be claimed.
Hours x users	Hours/user multiplied by the number of users.
Adjusted quota	The sum of the quota hours added to specific users.
Total hours * $/hour	The cost per hour based on the VM size selected. This is based on the regular pay as you go price.
Total estimated cost	This is the maximum price for this lab based on current settings.

Template tile

You see the following information on this tile:

- The date on which the template was created
- The date on which the template was last published

It also has a link to navigate to the **Template** page where you can manage the template VM for the class.

Virtual machine pool tile

You see the following information on this tile:

- Number of virtual machines that are assigned to students (users)
- Number of virtual machines that haven't been assigned to students yet

It also has a link to navigate to the **Virtual machine pool** page where you can manage the pool of virtual machines in the lab.

Users tile

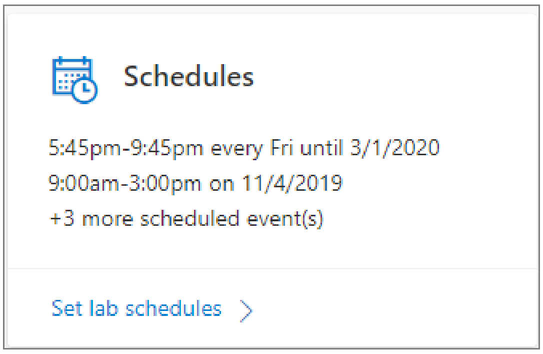
You see the following information on this tile:

- Number of users registered to the class
- Number of users who are added to the lab but not registered to the class

It also has a link to navigate to the **Users** page where you can manage users for the lab.

Schedules tile

You see the current scheduled events for the lab on the tile. It also has a link to navigate to the **Schedule** page where you can create and manage schedules. The tile shows you details for only two scheduled events and the number of remaining scheduled events for the lab.



Create and manage a classroom template in Azure Lab Services

3/5/2021 • 3 minutes to read • Edit Online

A template in a lab is a base virtual machine image from which all users' virtual machines are created. Set up the template virtual machine so that it is configured with exactly what you want to provide to the lab users. You can provide a name and description of the template that the lab users see. Then, you publish the template to make instances of the template VM available to your lab users. When you publish a template, Azure Lab Services creates VMs in the lab by using the template. The number of VMs created in this process is same as the maximum number of users allowed into the lab, which you can set in the usage policy of the lab. All virtual machines have the same configuration as the template.

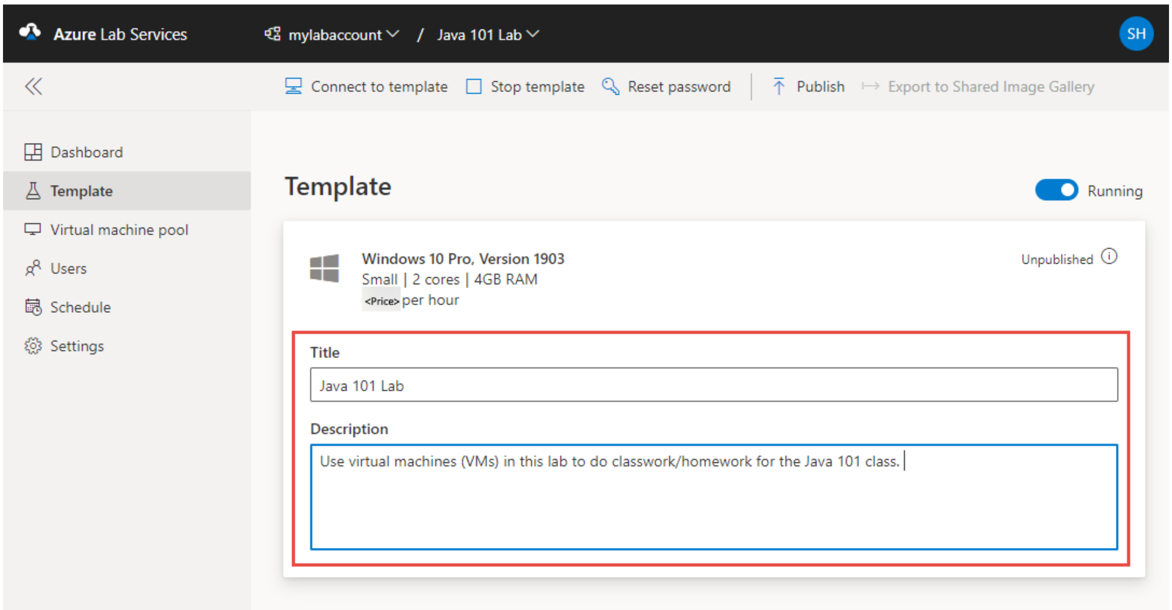
This article describes how to create and manage a template virtual machine in a classroom lab of Azure Lab Services.

> **NOTE**
>
> When you create a lab, the template VM is created but it's not started. You can start it, connect to it, and install any pre-requisite software for the lab, and then publish it. When you publish the template VM, it's is automatically shut down for you if you haven't done so.
>
> Template VMs incur **cost** when running, so ensure that the template VM is shutdown when you don't need it to be running.

Set or update template title and description

Use the following steps to set title and description for the first time, and update them later.

1. On the **Template** page, enter the new **title** for the lab.
2. Enter the new **description** for the template. When you move the focus out of the text box, it's automatically saved.

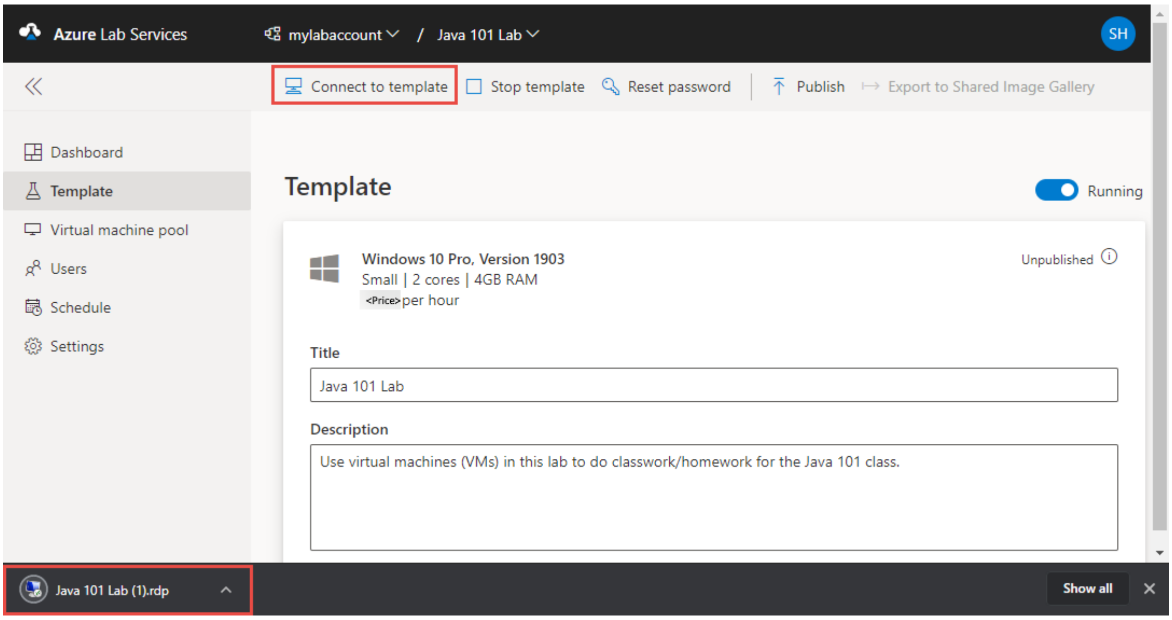


Update a template VM

Use the following steps to update a template VM.

1. Wait until the template VM is started, and then select **Connect to template** on the toolbar to connect to the template VM, and follow instructions. If it's a Windows machine, you will see an option to download the RDP file.

2. Once you connect to the template and make changes, it will no longer have the same setup as the virtual machines last published to your users. Template changes will not be reflected on your users' existing virtual machines until after you publish again.



3. Install any software that's required for students to do the lab (for example, Visual Studio, Azure Storage Explorer, etc.).

4. Disconnect (close your remote desktop session) from the template VM.

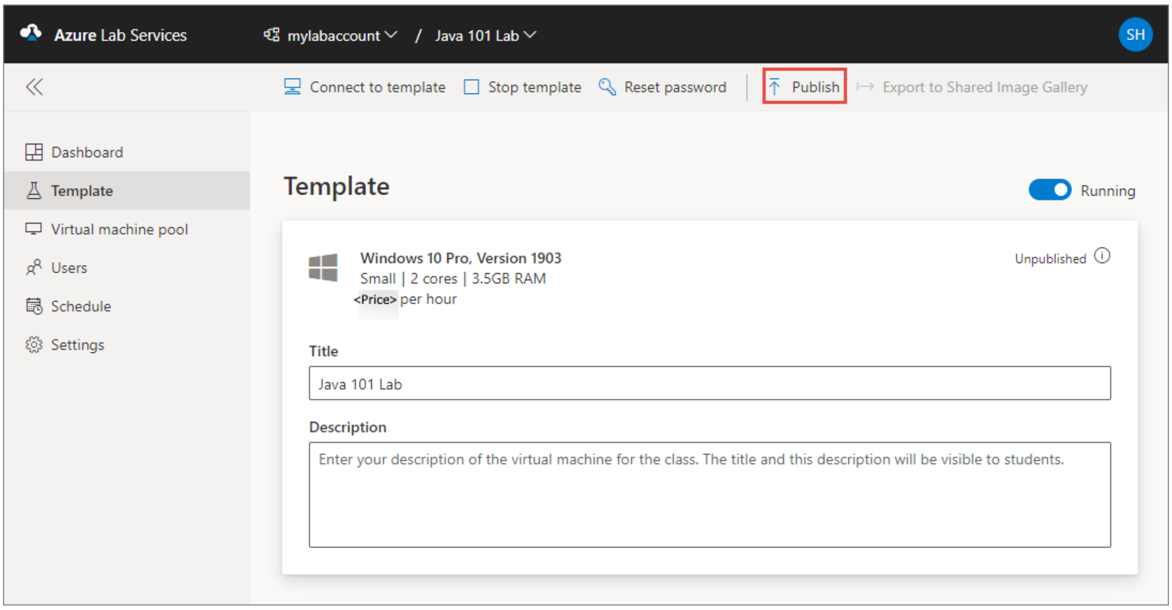
5. **Stop** the template VM by selecting **Stop template**.

6. Follow steps in the next section to **Publish** the updated template VM.

Publish the template VM

In this step, you publish the template VM. When you publish the template VM, Azure Lab Services creates VMs in the lab by using the template. All virtual machines have the same configuration as the template.

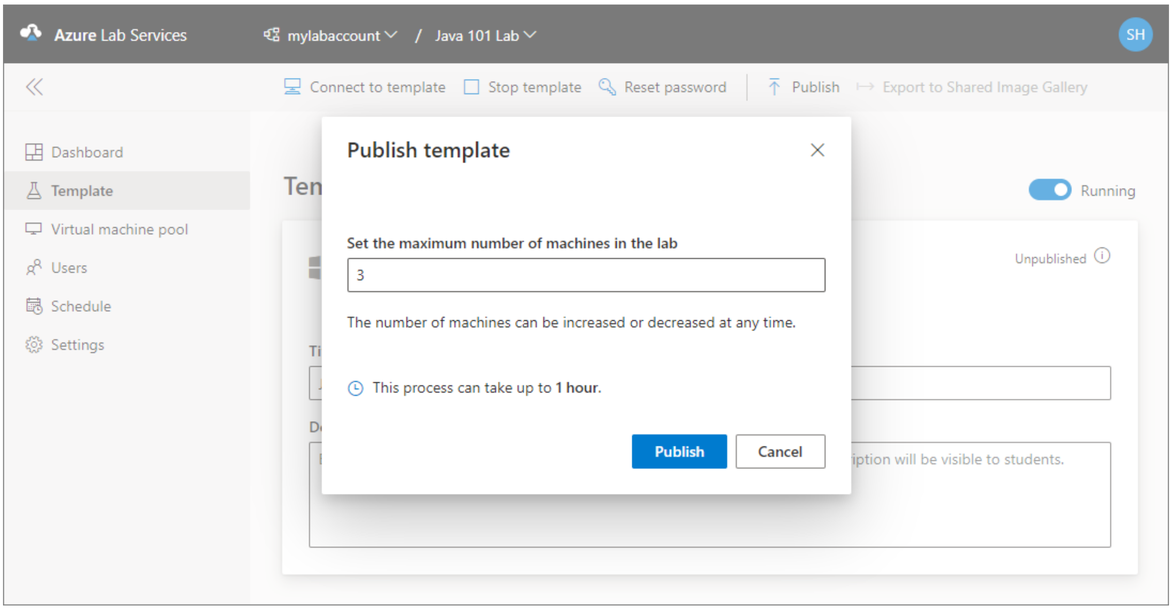
1. On the **Template** page, select **Publish** on the toolbar.



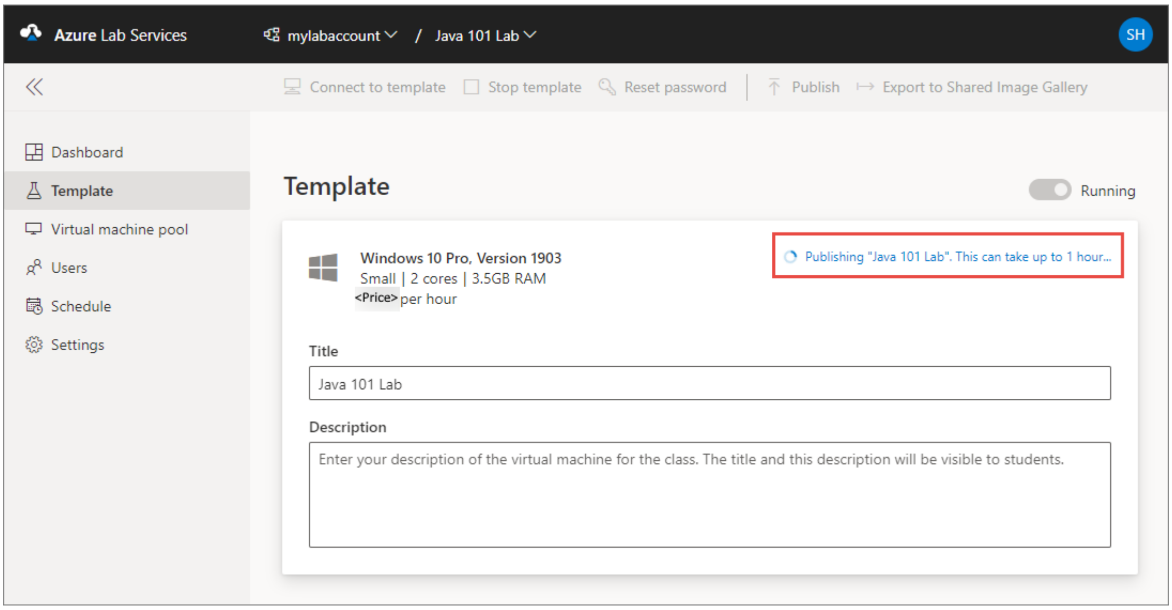
WARNING

Once you publish, you can't unpublish.

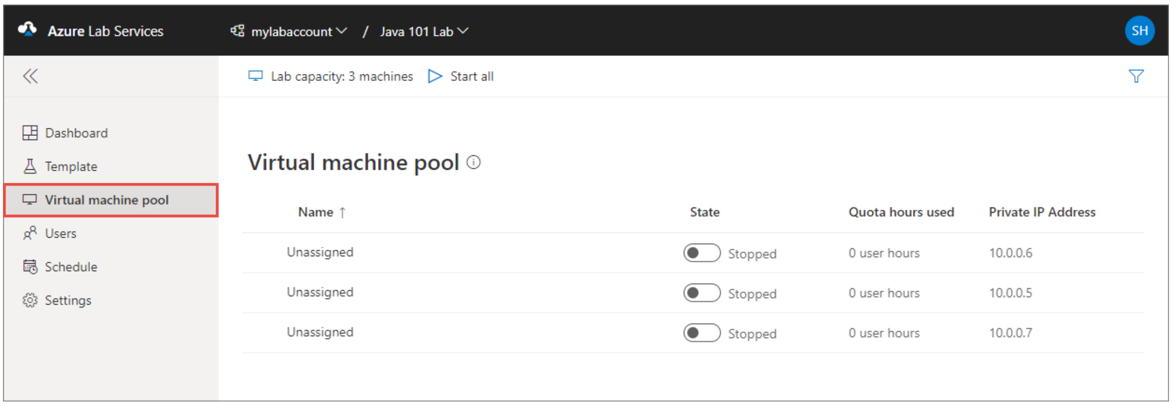
2. On the **Publish template** page, enter the number of virtual machines you want to create in the lab, and then select **Publish**.



3. You see the **status of publishing** the template on page. This process can take up to an hour.



4. Wait until the publishing is complete and then switch to the **Virtual machines pool** page by selecting **Virtual machines** on the left menu or by selecting **Virtual machines** tile. Confirm that you see virtual machines that are in **Unassigned** state. These VMs are not assigned to students yet. They should be in **Stopped** state. You can start a student VM, connect to the VM, stop the VM, and delete the VM on this page. You can start them in this page or let your students start the VMs.



Next steps

See the following articles:

- As an admin, create and manage lab accounts
- As a lab owner, create and manage labs
- As a lab owner, configure and control usage of a lab
- As a lab user, access labs

Set up and manage virtual machine pool

3/5/2021 • 3 minutes to read • Edit Online

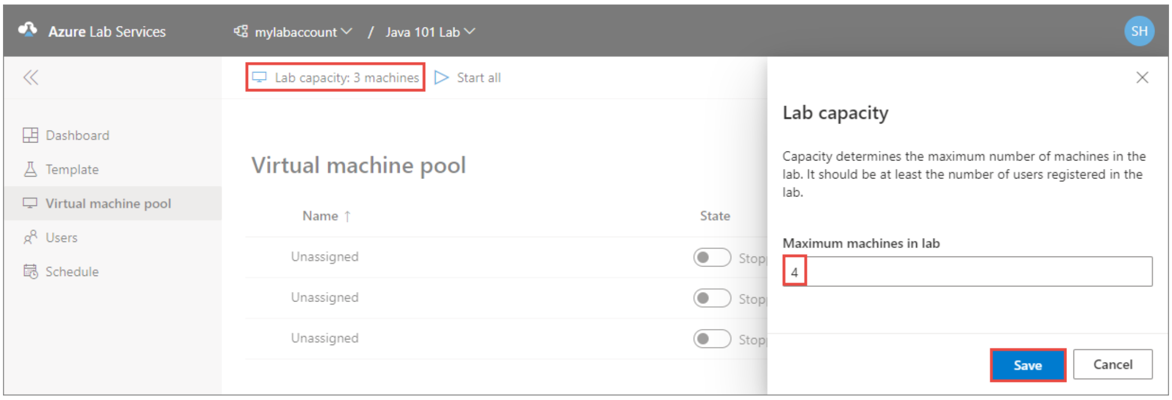
This article shows you how to do the following tasks:

- Increase the number of virtual machines (VMs) in the lab
- Start all VMs or selected VMs
- Reset VMs

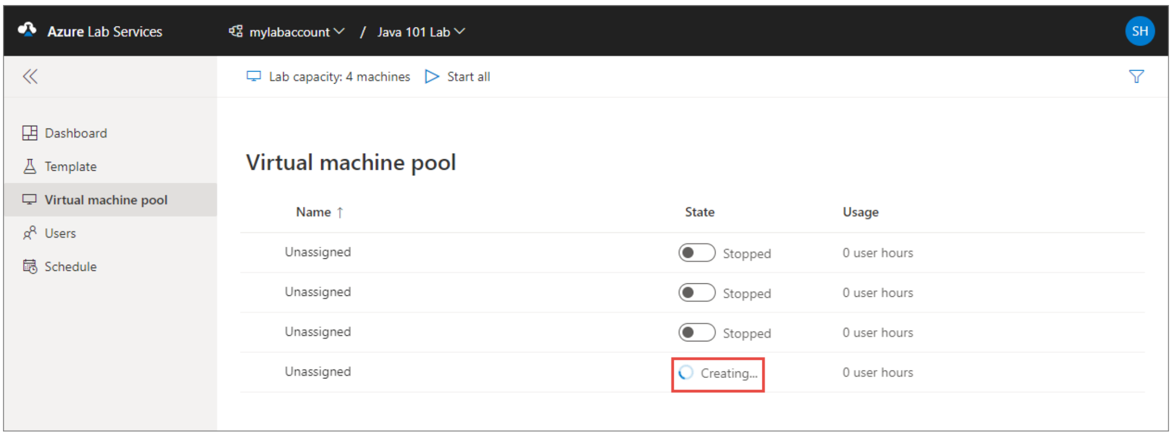
Update the lab capacity

To increase or decrease the lab capacity (number of virtual machines in a lab), do the following steps:

1. On the **Virtual machine pool** page, select **Lab capacity: <number> machines**.
2. Enter the new **number of VMs** you want in the lab. This number must be greater than or equal to the number of users registered in the lab.
3. Then, select **Save**.



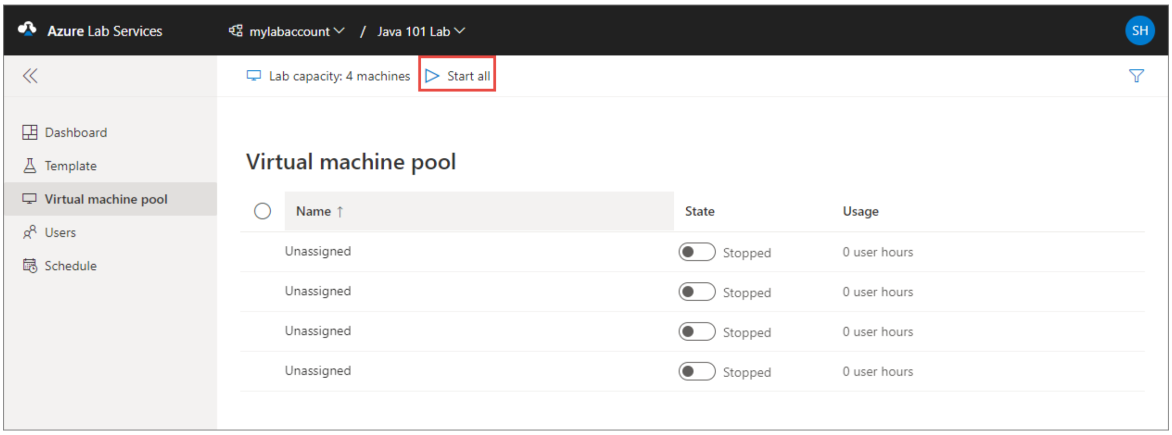
4. If you increased the capacity, you can see the VM or VMs being created. If you don't see the new VM in the list, refresh the page.



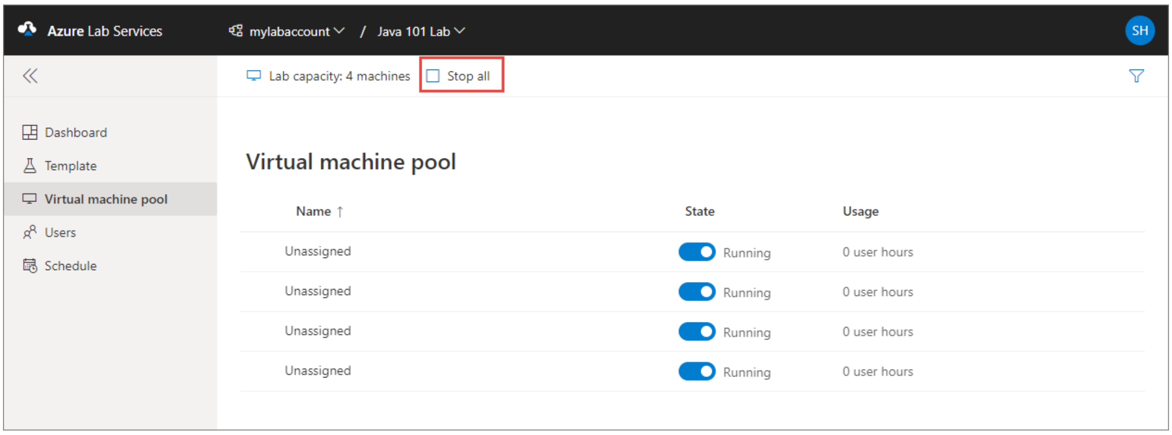
Start VMs

Start ot stop all VMs

1. Switch to the **Virtual machine pool** page.
2. Select **Start all** from the toolbar.



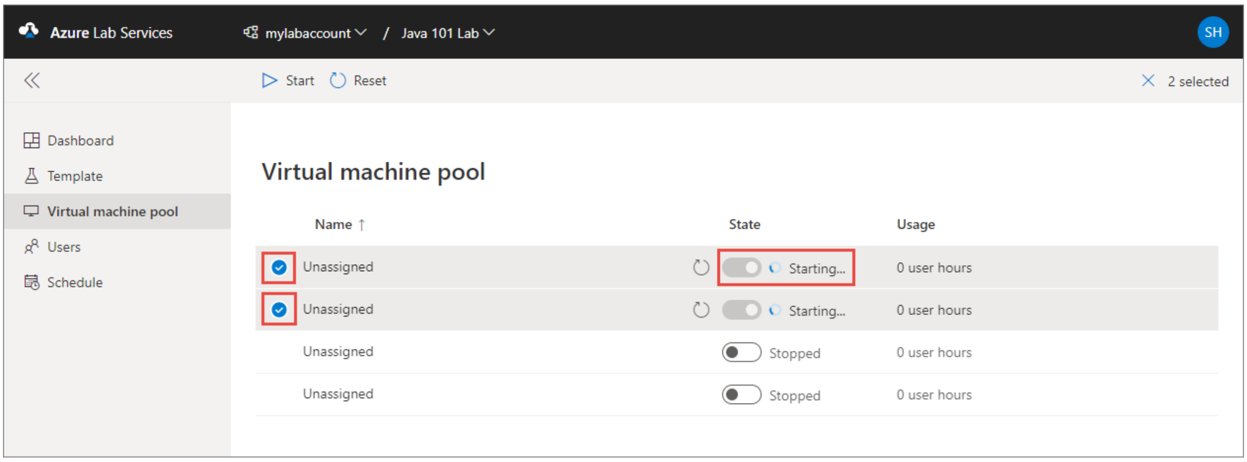
3. After all the VMs are started, you can stop all VMs by selecting the **Stop all** button on the toolbar.



Start selected VMs

There are two ways to start selected VMs (one or more). First way is to select the VM or VMs in the list, and then select **Start** on the toolbar.

The second way is to select one or more VMs in the list, and toggle the button in the **State** column.

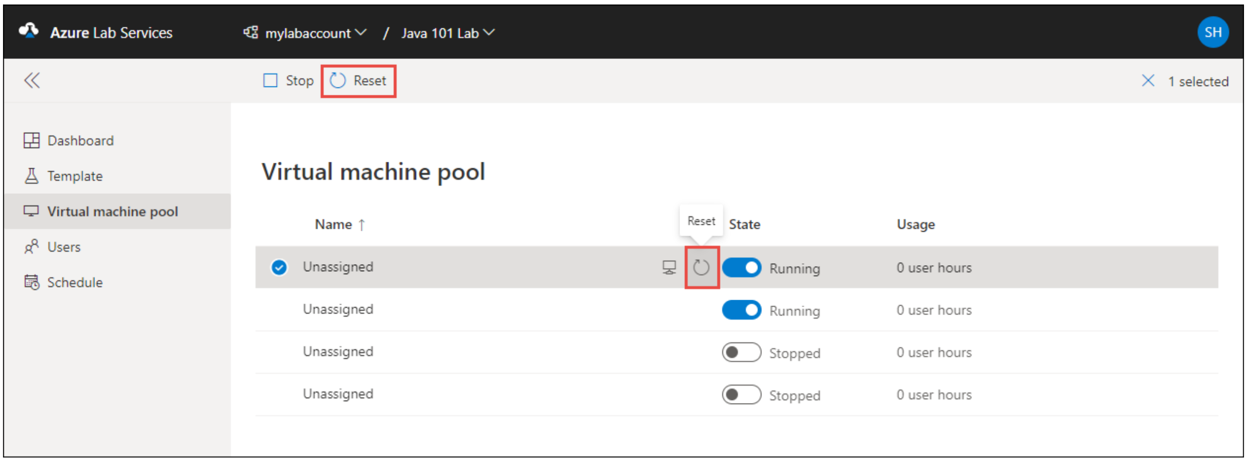


Similarly, you can stop one or more VMs by toggling the button in the **State** column or selecting **Stop** on the toolbar.

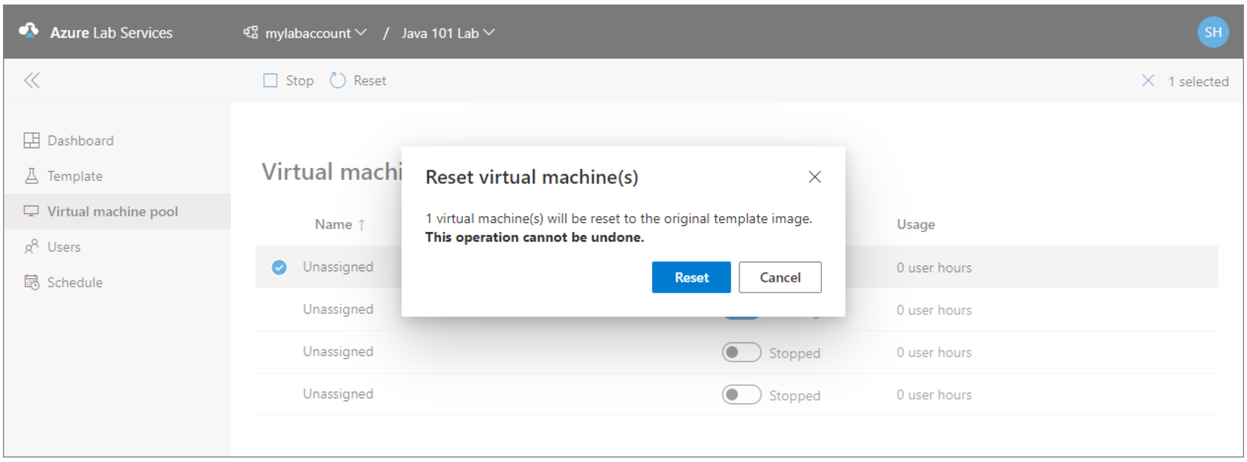
> **NOTE**
>
> When an educator turns on a student VM, quota for the student isn't affected. Quota for a user specifies the number of lab hours available to the user outside of the scheduled class time. For more information on quotas, see Set quotas for users.

Reset VMs

To reset one or more VMs, select them in the list, and then select **Reset** on the toolbar.



On the **Reset virtual machine(s)** dialog box, select **Reset**.

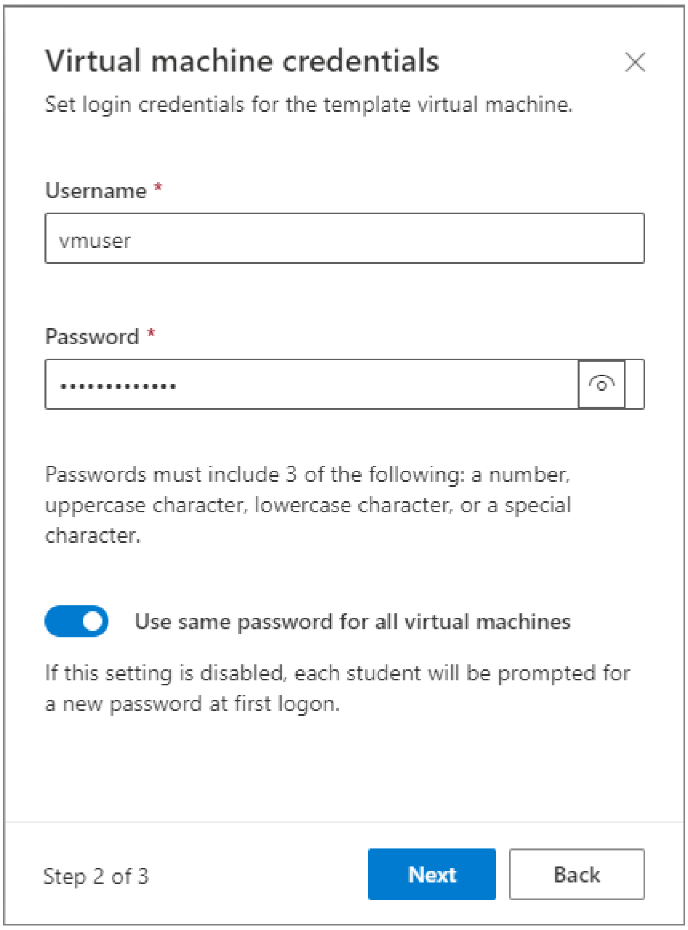


Set password for VMs

A lab owner (educator) can set/reset the password for VMs at the time of creating the lab (lab creation wizard) or after creating the lab on the **Template** page.

Set password at the time of lab creation

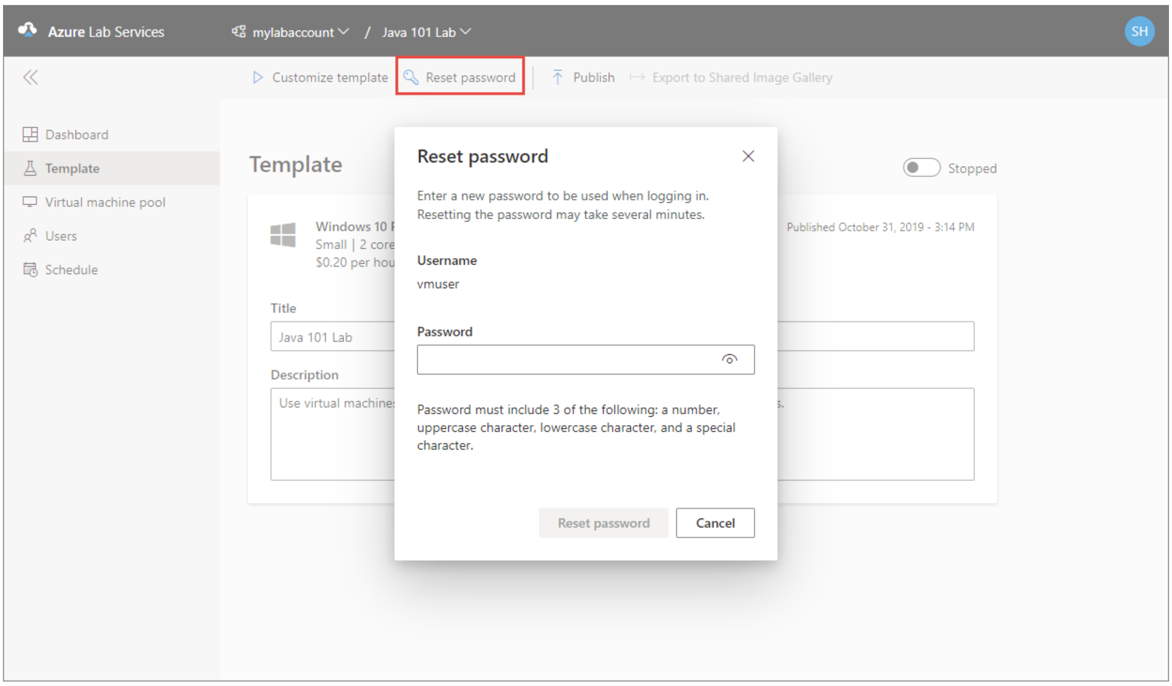
A lab owner (educator) can set a password for VMs in the lab on the **Virtual machine credentials** page of the lab creation wizard.



By enabling/disabling the **Use same password for all virtual machines** option on this page, an educator can choose to use same password for all VMs in the lab or allow students to set passwords for their VMs. By default, this setting is enabled for all Windows and Linux operating system images except Ubuntu. When this setting is disabled, students will be prompted to set a password when they try to connect to the VM for the first time.

Reset password later

1. On the **Template** page of the lab, select **Reset password** on the toolbar.
2. On the **Reset password** dialog box, enter a password, and select **Reset password**.

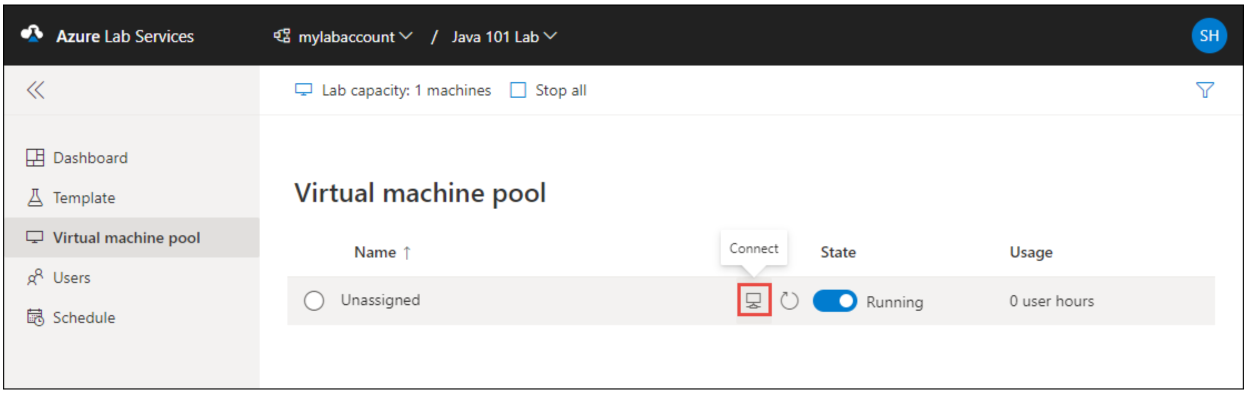


Connect to student VMs

The lab creator (educator) can connect to a student VM if the following conditions are met:

- The **Use same password for all virtual machines** option was selected when creating the lab
- The VM is running

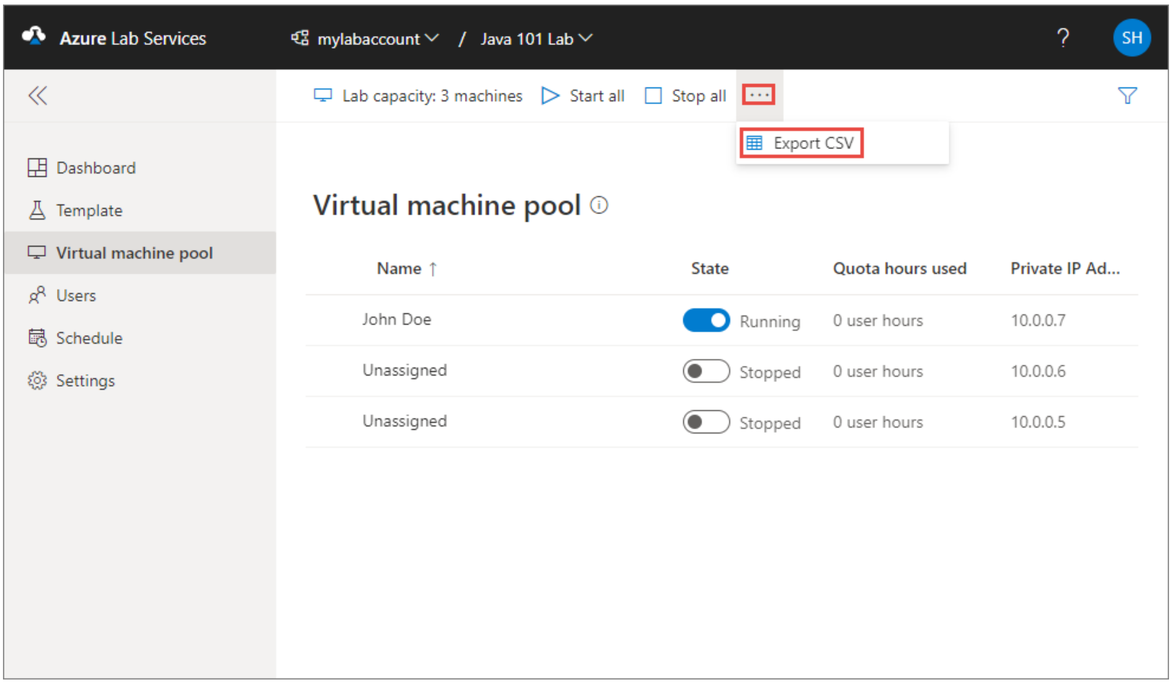
To connect to the student VM, hover the mouse on the VM in the list and select the computer button.



> **NOTE**
> When the educator starts the VM and connects to it, the student quota is unaffected.

Export list of virtual machines to a CSV file

1. Switch to the **Virtual machine pool** tab.
2. Select ... (ellipsis) on the toolbar and then select **Export CSV**.



Next steps

To learn about other student usage options you (as a lab owner) can configure, see the following article: Configure student usage.

To learn about how students can reset passwords for their VMs, see Set or reset password for virtual machines in labs (students).

Add and manage lab users

5/19/2021 • 10 minutes to read • Edit Online

This article describes how to add student users to a lab, register them with the lab, control the number of additional hours they can use the virtual machine (VM), and more.

When you add users, by default, the **Restrict access** option is turned on and, unless they're in the list of users, students can't register with the lab even if they have a registration link. Only listed users can register with the lab by using the registration link you send. You can turn off **Restrict access**, which allows students to register with the lab as long as they have the registration link.

This article shows how to add users to a lab.

Add users from an Azure AD group

Overview

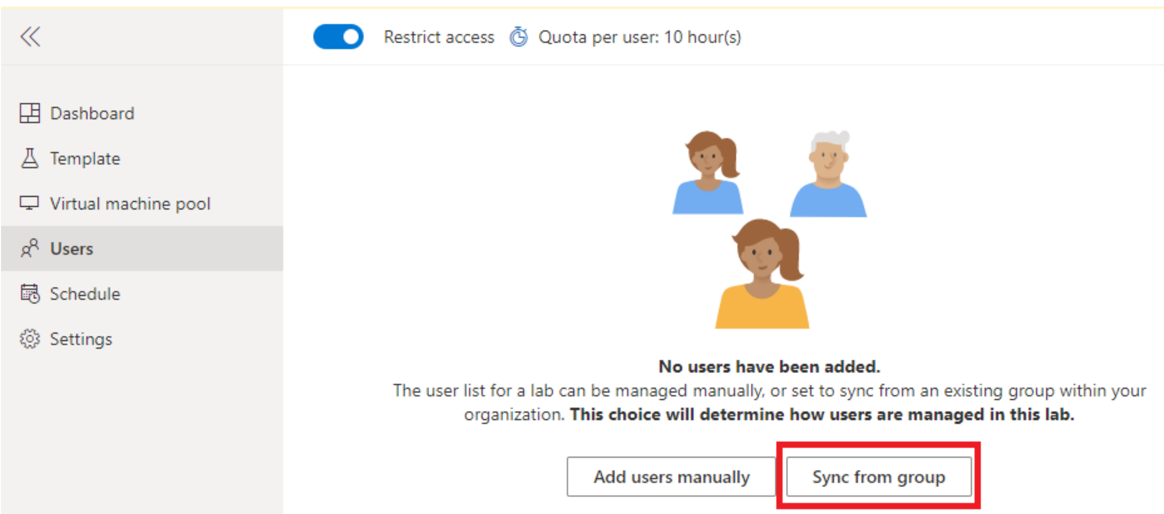
You can now sync a lab user list to an existing Azure Active Directory (Azure AD) group so that you do not have to manually add or delete users.

An Azure AD group can be created within your organization's Azure Active Directory to manage access to organizational resources and cloud-based apps. To learn more, see Azure AD groups. If your organization uses Microsoft Office 365 or Azure services, your organization will already have admins who manage your Azure Active Directory.

Sync users with Azure AD group

> **IMPORTANT**
> Make sure the user list is empty. If there are existing users inside a lab that you added manually or through importing a CSV file, the option to sync the lab to an existing group will not appear.

1. Sign in to the Azure Lab Services website.
2. Select the lab you want to work with.
3. In the left pane, select **Users**.
4. Click **Sync from group**.



5. You will be prompted to pick an existing Azure AD group to sync your lab to.

   If you don't see an Azure AD group in the list, could be because of the following reasons:

   - If you are a guest user for an Azure Active Directory (usually if you're outside the organization that owns the Azure AD), and you are not able to to search for groups inside the Azure AD. In this case, you won't be able to add an Azure AD group to the lab in this case.
   - Azure AD groups created through Teams do not show up in this list. You can add the Azure Lab Services app inside Teams to create and manage labs directly from within it. See more information about managing a lab's user list from within Teams.
6. Once you picked the Azure AD group to sync your lab to, click **Add**.
7. Once a lab is synced, it will pull everyone inside the Azure AD group into the lab as users, and you will see the user list updated. Only the people in this Azure AD group will have access to your lab. The user list will refresh every 24 hours to match the latest membership of the Azure AD group. You can also click on the Sync button in the Users tab to manually sync to the latest changes in the Azure AD group.
8. Invite the users to your lab by clicking on the **Invite All** button, which will send an email to all users with the registration link to the lab.

Automatic management of virtual machines based on changes to the Azure AD group

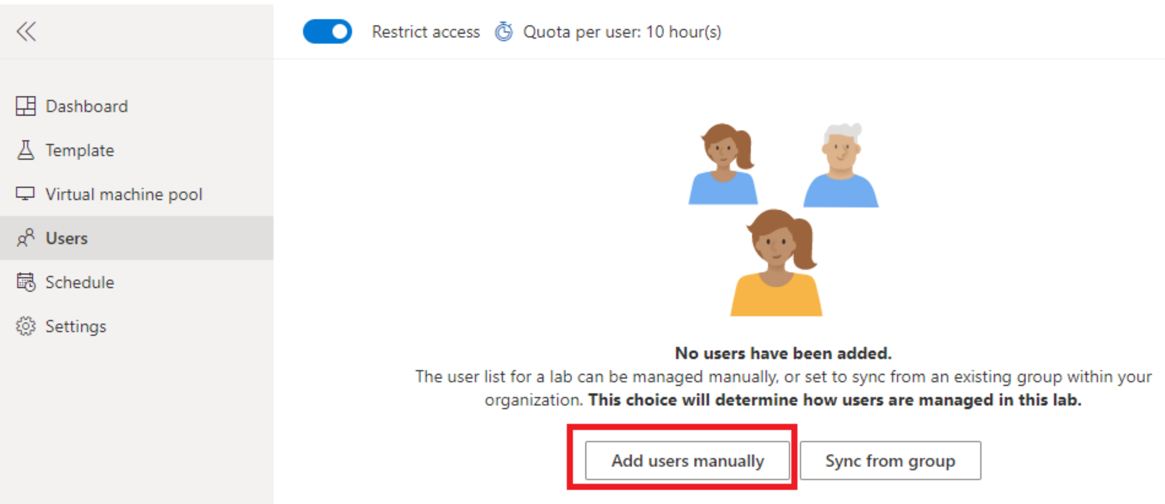
Once the lab is synced to an Azure AD group, the number of virtual machines in the lab will automatically match the number of users in the group. You will no longer be able to manually update the lab capacity. When a user is added to the Azure AD group, a lab will automatically add a virtual machine for that user. When a user is deleted from the Azure AD group, a lab will automatically delete the user's virtual machine from the lab.

Add users manually from email(s) or CSV file

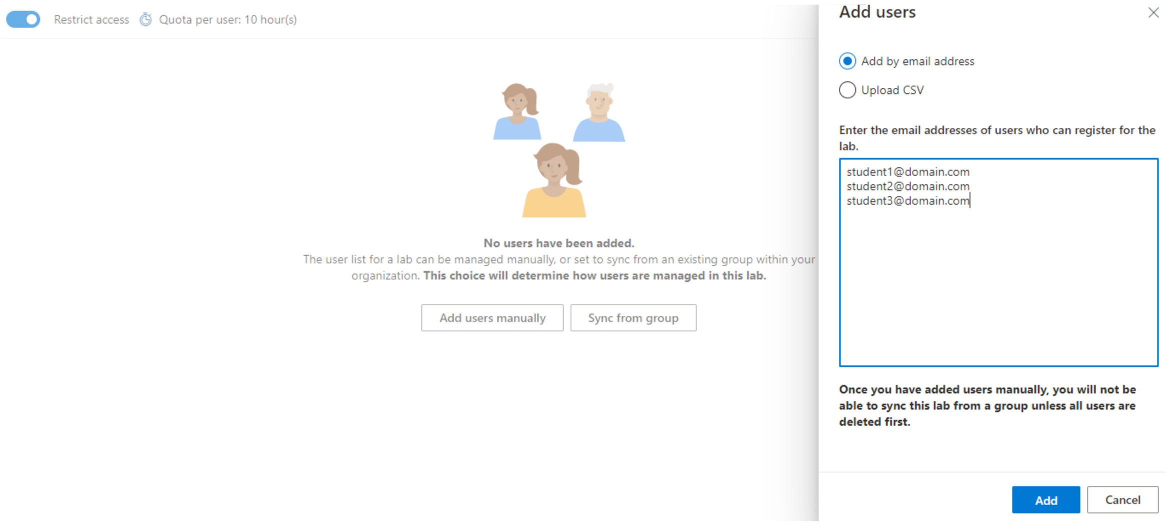
In this section, you add students manually (by email address or by uploading a CSV file).

Add users by email address

1. In the left pane, select **Users**.
2. Click **Add users manually**.

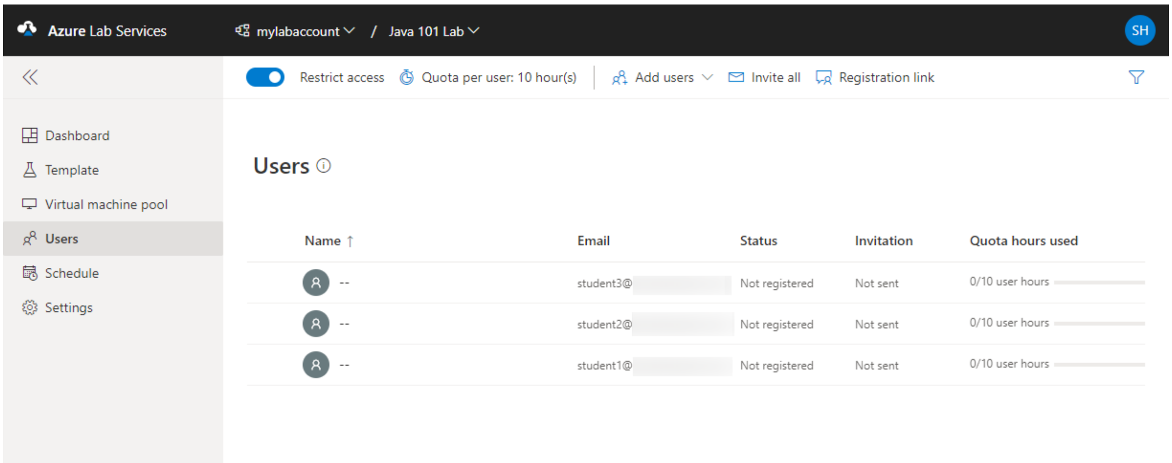


3. Select **Add by email address** (default), enter the students' email addresses on separate lines or on a single line separated by semicolons.



4. Select **Save**.

   The list displays the email addresses and statuses of the current users, whether they're registered with the lab or not.



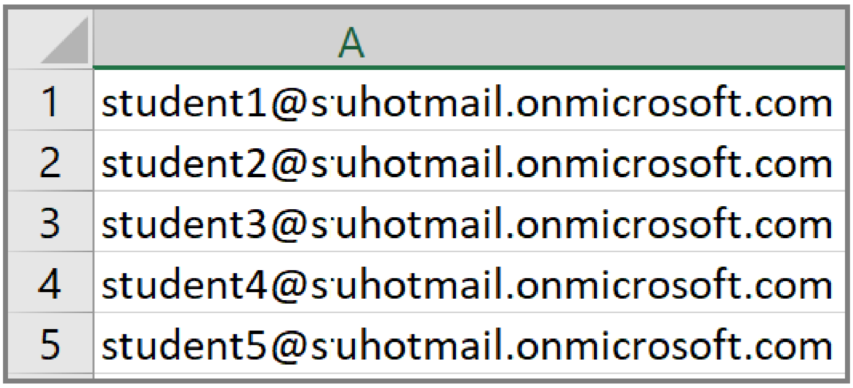
> **NOTE**
>
> After the students are registered with the lab, the list displays their names. The name that's shown in the list is constructed by using the first and last names of the students in Azure Active Directory.

Add users by uploading a CSV file

You can also add users by uploading a CSV file that contains their email addresses.

A CSV text file is used to store comma-separated (CSV) tabular data (numbers and text). Instead of storing information in columns fields (such as in spreadsheets), a CSV file stores information separated by commas. Each line in a CSV file will have the same number of comma-separated "fields." You can use Excel to easily create and edit CSV files.

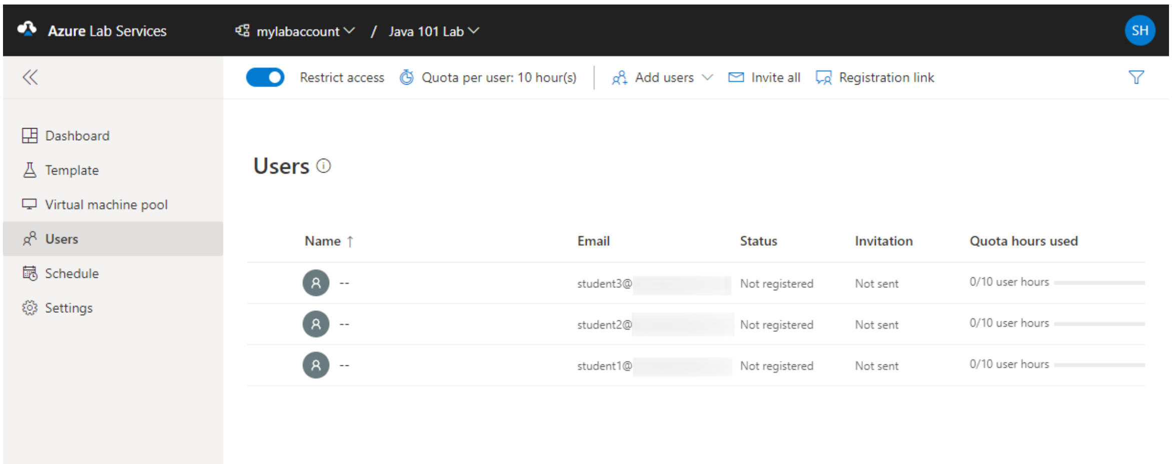
1. In Microsoft Excel, create a CSV file that lists students' email addresses in one column.



	A
1	student1@suhotmail.onmicrosoft.com
2	student2@suhotmail.onmicrosoft.com
3	student3@suhotmail.onmicrosoft.com
4	student4@suhotmail.onmicrosoft.com
5	student5@suhotmail.onmicrosoft.com

2. At the top of the **Users** pane, select **Add users**, and then select **Upload CSV**.
3. Select the CSV file that contains the students' email addresses, and then select **Open**.

   The **Add users** window displays the email address list from the CSV file.
4. Select **Save**.
5. In the **Users** pane, view the list of added students.



Send invitations to users

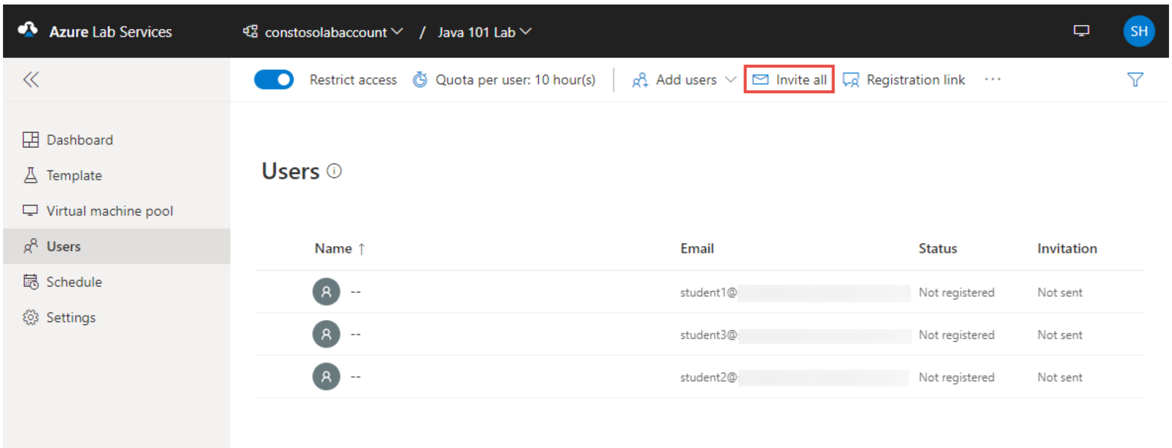
To send a registration link to new users, use one of the following methods.

If the **Restrict access** option is enabled for the lab, only listed users can use the registration link to register to the lab. This option is enabled by default.

Invite all users

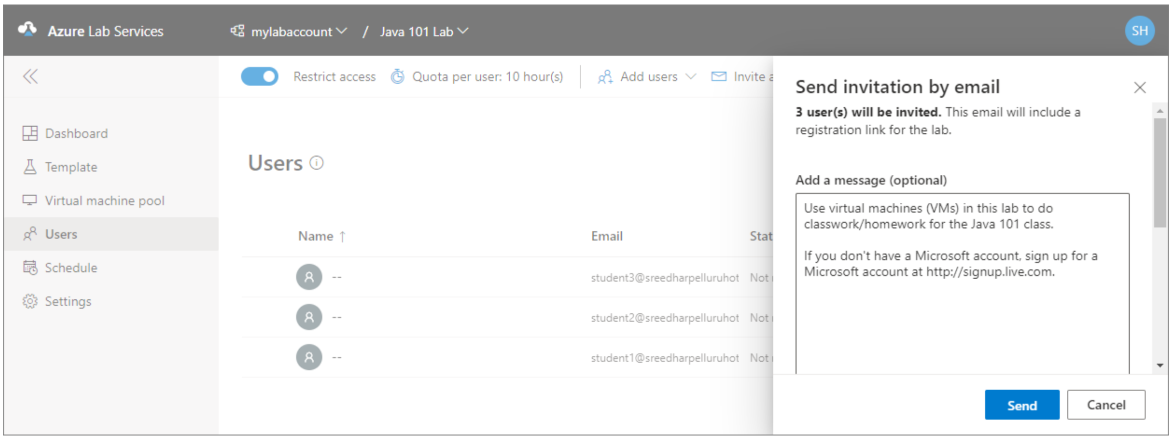
This method shows you how to send email with a registration link and an optional message to all listed students.

1. In the **Users** pane, select **Invite all**.



2. In the **Send invitation by email** window, enter an optional message, and then select **Send**.

   The email automatically includes the registration link. To get and save the registration link separately, select the ellipsis (...) at the top of the **Users** pane, and then select **Registration link**.



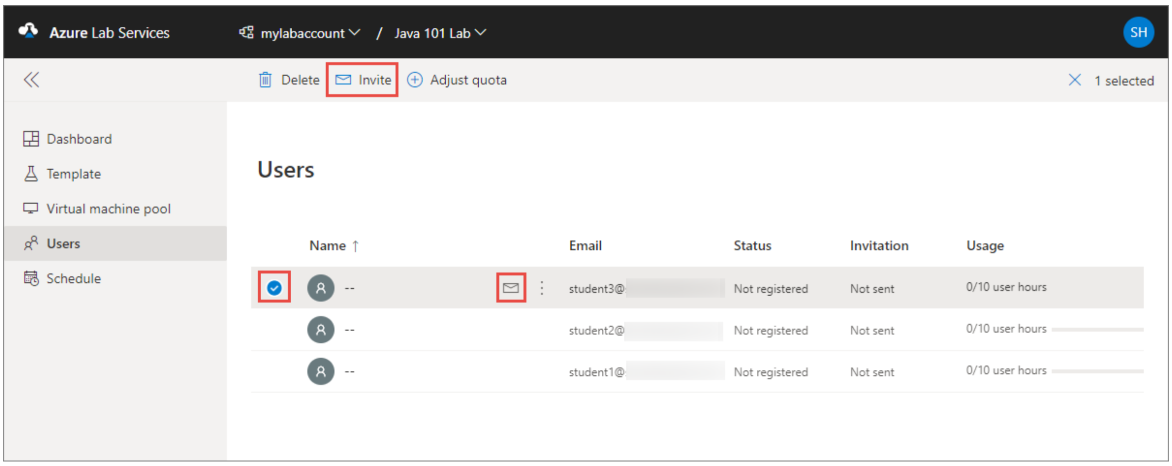
   The **Invitation** column of the **Users** list displays the invitation status for each added user. The status should change to **Sending** and then to **Sent on <date>**.

Invite selected users

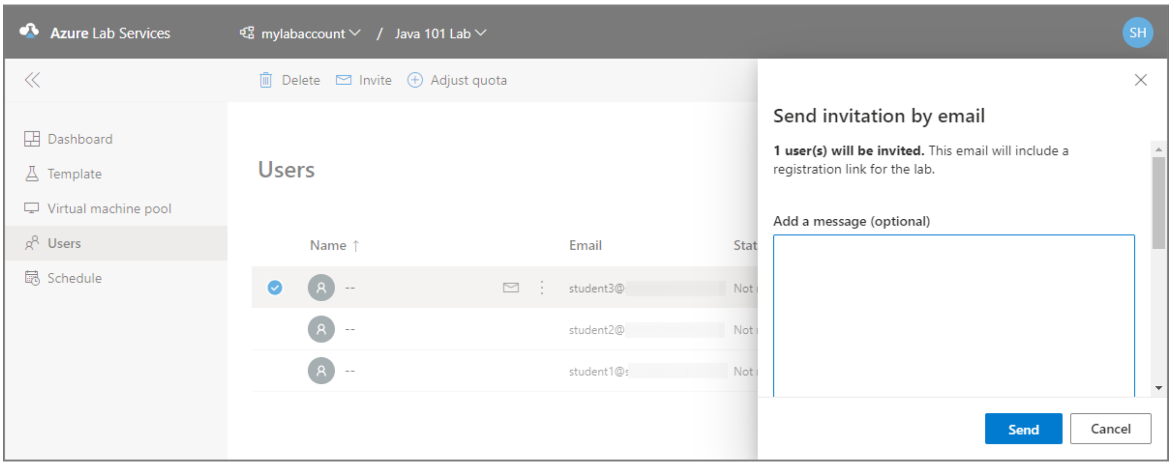
This method shows you how to invite only certain students and get a registration link that you can share with other people.

1. In the **Users** pane, select a student or multiple students in the list.

2. In the row for the student you've selected, select the **envelope** icon or, on the toolbar, select **Invite**.



3. In the **Send invitation by email** window, enter an optional **message**, and then select **Send**.

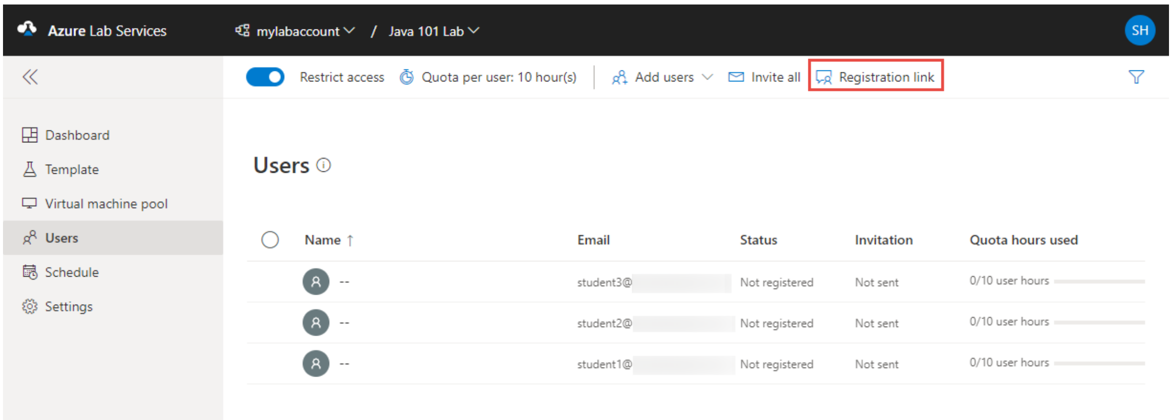


The **Users** pane displays the status of this operation in the **Invitation** column of the table. The invitation email includes the registration link that students can use to register with the lab.

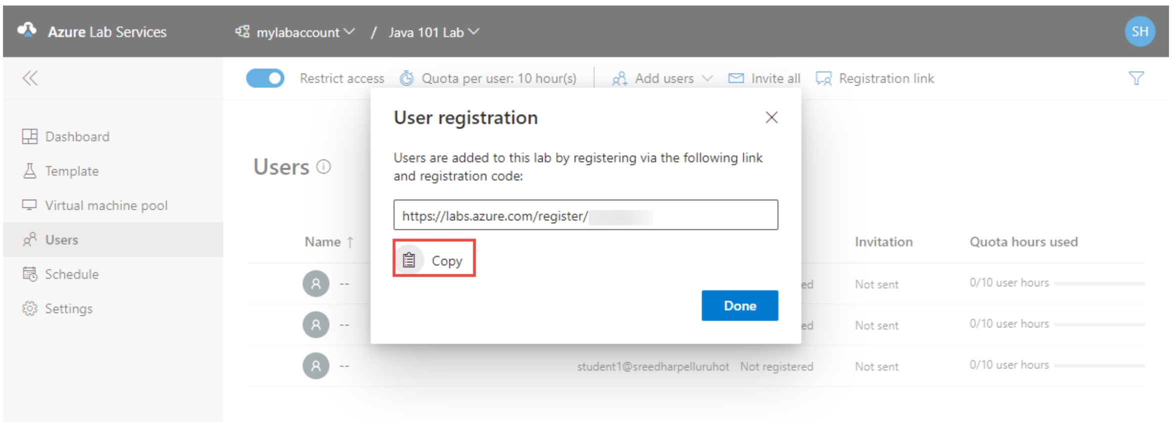
Get the registration link

In this section, you can get the registration link from the portal and send it by using your own email application.

1. In the **Users** pane, select **Registration link**.



2. In the **User registration** window, select **Copy**, and then select **Done**.



The link is copied to the clipboard.

3. In your email application, paste the registration link, and then send the email to a student so that the student can register for the class.

View registered users

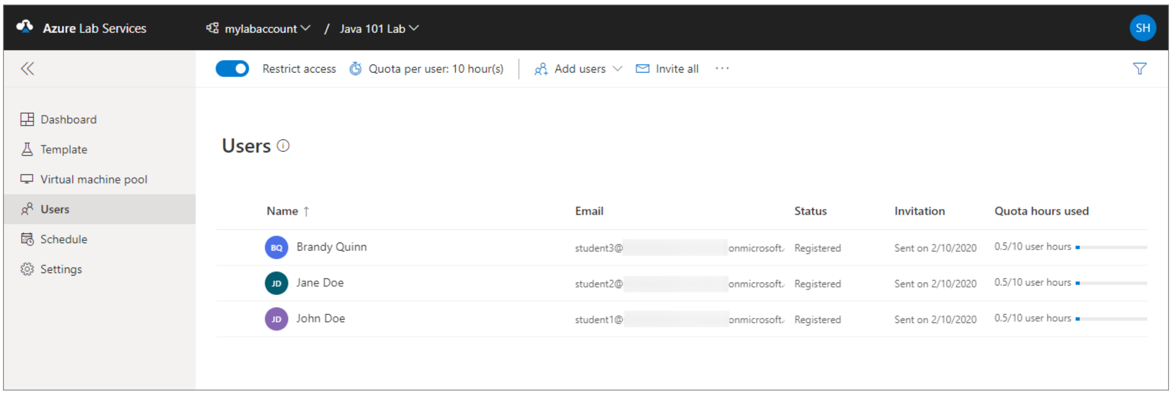
1. Go to the Azure Lab Services website.

2. Select **Sign in**, and then enter your credentials. Azure Lab Services supports organizational accounts and Microsoft accounts.

3. On the **My labs** page, select the lab whose usage you want to track.

4. In the left pane, select **Users**, or select the **Users** tile.

   The **Users** pane displays a list of students who have registered with your lab.



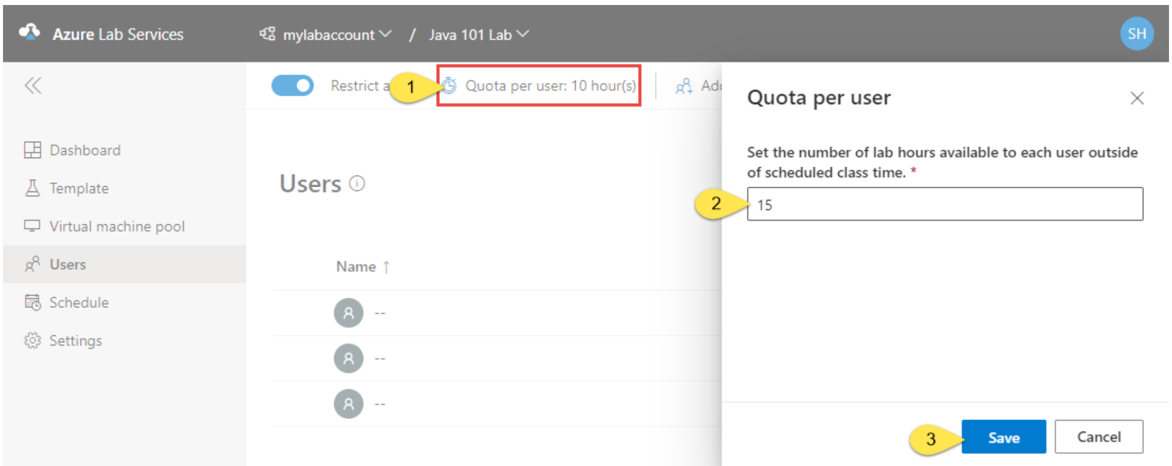
> **NOTE**
>
> If you republish a lab or reset student VMs, the students will remain registered for the labs' VMs. However, the contents of the VMs will be deleted and the VMs will be recreated with the template VM's image.

Set quotas for users

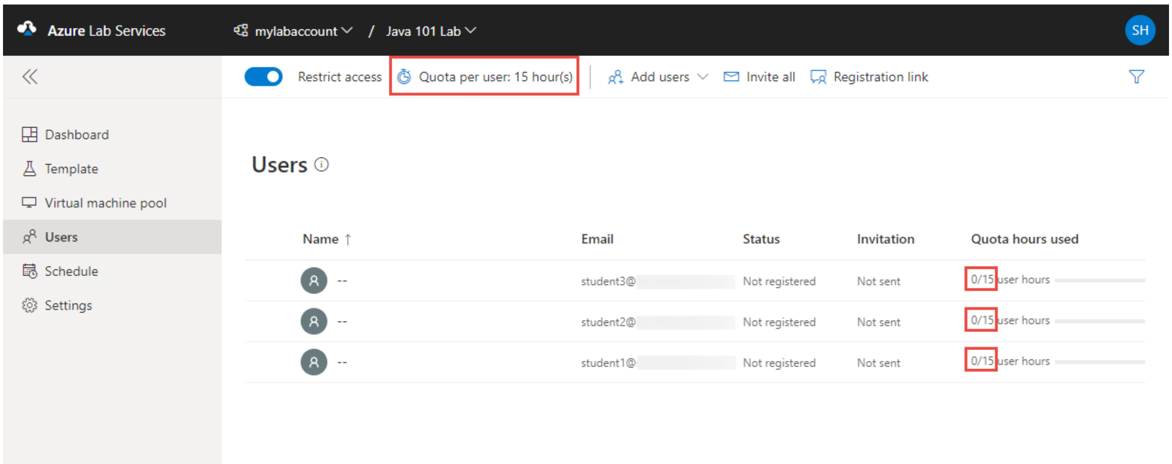
You can set an hour quota for each student by doing the following:

1. In the **Users** pane, select **Quota per user: <number> hour(s)** on the toolbar.

2. In the **Quota per user** window, specify the number of hours you want to give to each student outside the scheduled class time, and then select **Save**.



The changed values are now displayed on the **Quota per user: <number of hours>** button on the toolbar and in the users list, as shown here:



IMPORTANT

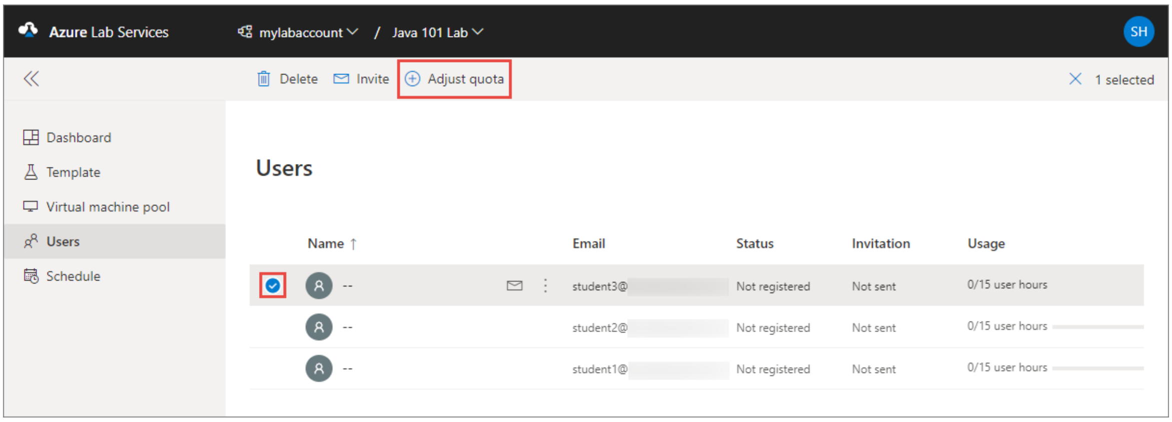
The scheduled running time of VMs does not count against the quota that's allotted to a student. The quota is for the time outside of scheduled hours that a student spends on VMs.

Set additional quotas for specific users

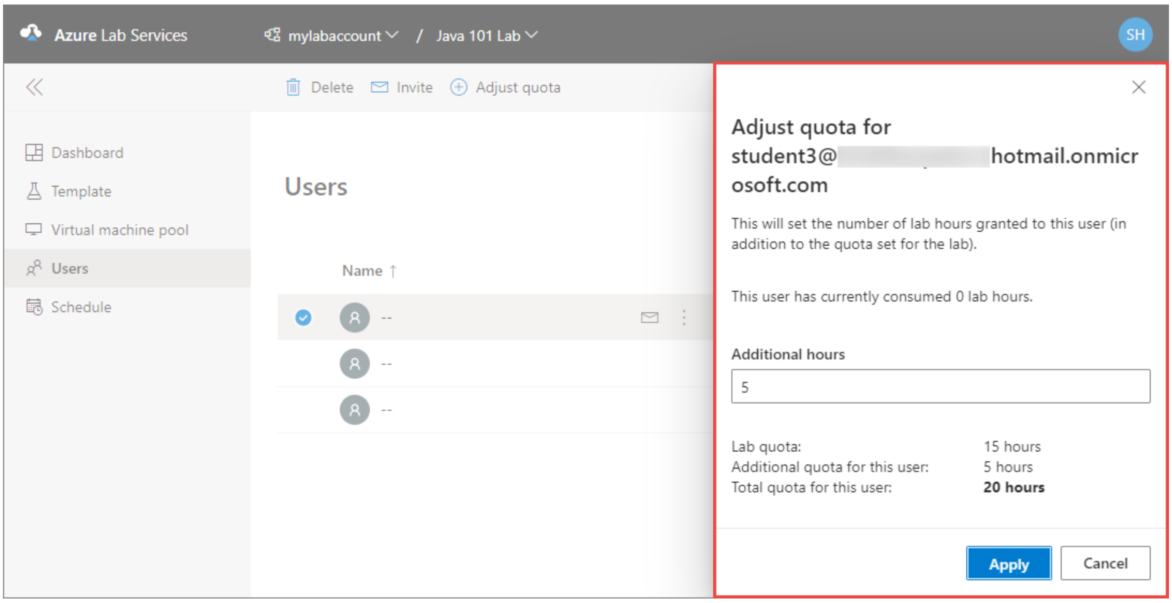
You can specify quotas for certain students beyond the common quotas that were set for all users in the preceding section. For example, if you, as an instructor, set the quota for all students to 10 hours and set an additional quota of 5 hours for a specific student, that student gets 15 (10 + 5) hours of quota. If you change the common quota later to, say, 15, the student gets 20 (15 + 5) hours of quota. Remember that this overall quota is outside the scheduled time. The time that a student spends on a lab VM during the scheduled time does not count against this quota.

To set additional quotas, do the following:

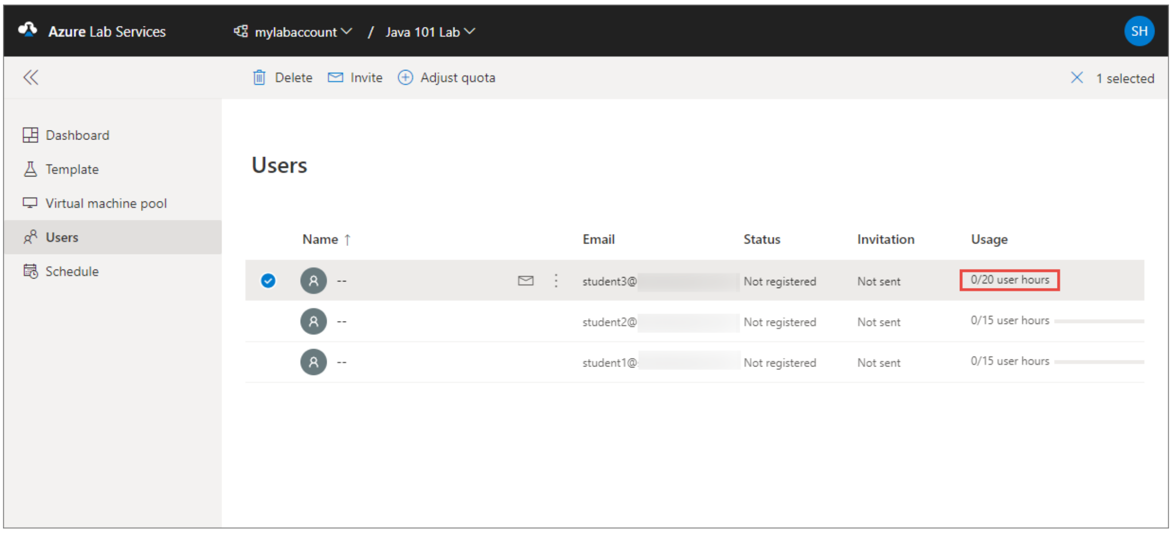
1. In the **Users** pane, select a student from the list, and then select **Adjust quota** on the toolbar.



2. In the **Adjust quota for <selected user or users email address>**, enter the number of additional lab hours you want to grant to the selected student or students, and then select **Apply**.



The **Usage** column displays the updated quota for the selected students.



Student accounts

To add students to a classroom lab, you use their email accounts. Students might have the following types of email accounts:

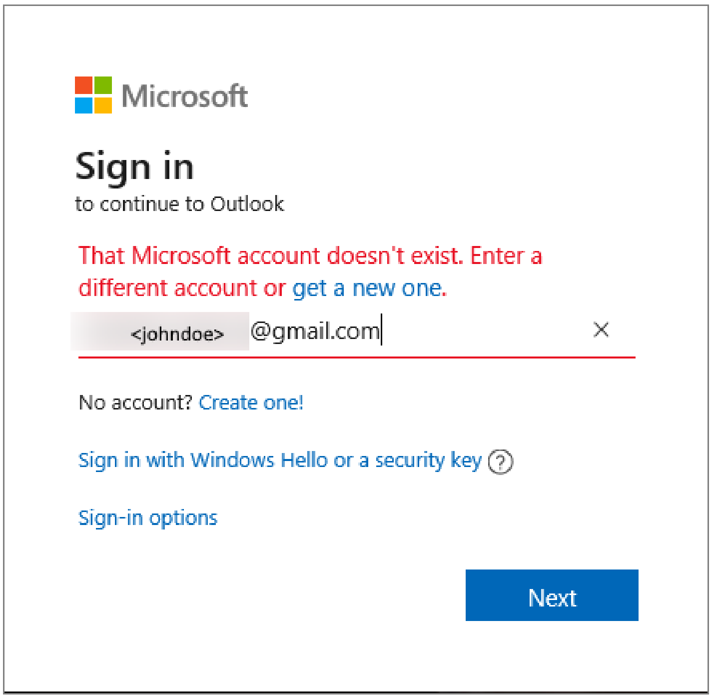
- A student email account that's provided by your university's Azure Active Directory instance.
- A Microsoft-domain email account, such as *outlook.com*, *hotmail.com*, *msn.com*, or *live.com*.
- A non-Microsoft email account, such as one provided by Yahoo! or Google. However, these types of accounts must be linked with a Microsoft account.
- A GitHub account. This account must be linked with a Microsoft account.

Use a non-Microsoft email account

Students can use non-Microsoft email accounts to register and sign in to a classroom lab. However, the registration requires that they first create a Microsoft account that's linked to their non-Microsoft email address.

Many students might already have a Microsoft account that's linked to their non-Microsoft email address. For example, students already have a Microsoft account if they've used their email address with other Microsoft products or services, such as Office, Skype, OneDrive, or Windows.

When students use the registration link to sign in to a classroom, they're prompted for their email address and password. Students who attempt to sign in with a non-Microsoft account that's not linked to a Microsoft account will receive the following error message:



Here's a link for students to sign up for a Microsoft account.

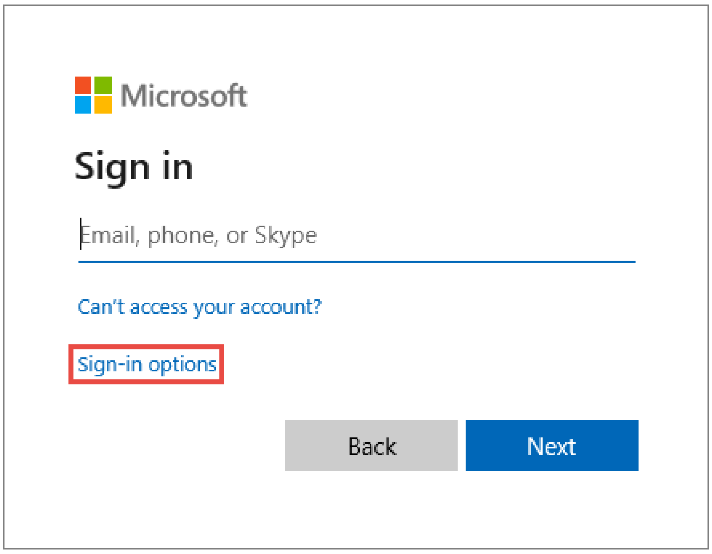
> **IMPORTANT**
>
> When students sign in to a classroom lab, they aren't given the option to create a Microsoft account. For this reason, we recommend that you include this sign-up link, http://signup.live.com, in the classroom lab registration email that you send to students who are using non-Microsoft accounts.

Use a GitHub account

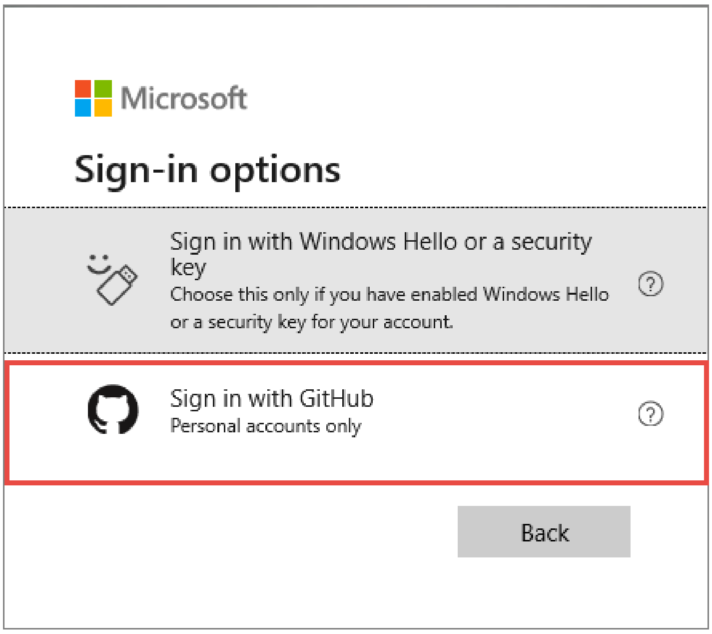
Students can also use an existing GitHub account to register and sign in to a classroom lab. If they already have a Microsoft account linked to their GitHub account, students can sign in and provide their password as shown in the preceding section.

If they haven't yet linked their GitHub account to a Microsoft account, they can do the following:

1. Select the **Sign-in options** link, as shown here:



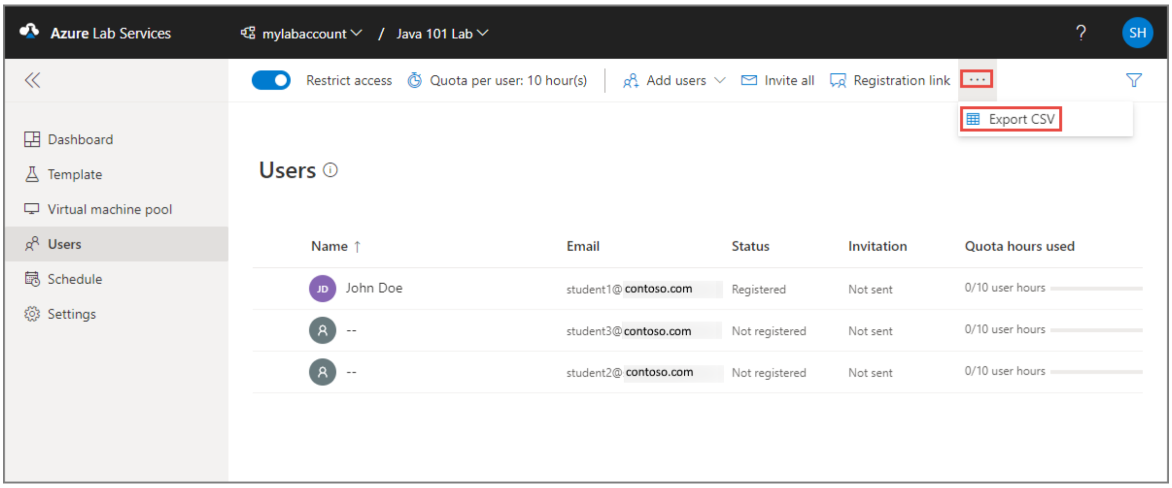
2. In the **Sign-in options** window, select **Sign in with GitHub**.



At the prompt, students then create a Microsoft account that's linked to their GitHub account. The linking happens automatically when they select **Next**. They're then immediately signed in and connected to the classroom lab.

Export a list of users to a CSV file

1. Go to the **Users** pane.
2. On the toolbar, select the ellipsis (...), and then select **Export CSV**.



Next steps

See the following articles:

- For administrators: Create and manage lab accounts
- For lab owners: Create and manage labs and Set up and publish templates
- For lab users: Access labs

Create and manage schedules for labs in Azure Lab Services

3/5/2021 • 2 minutes to read • Edit Online

Schedules allow you to configure a classroom lab such that VMs in the lab automatically start and shut down at a specified time. You can define a one-time schedule or a recurring schedule. The following procedures give you steps to create and manage schedules for a classroom lab:

IMPORTANT

The scheduled running time of VMs does not count against the quota allotted to a user. The quota is for the time outside of schedule hours that a student spends on VMs.

Set a schedule for the lab

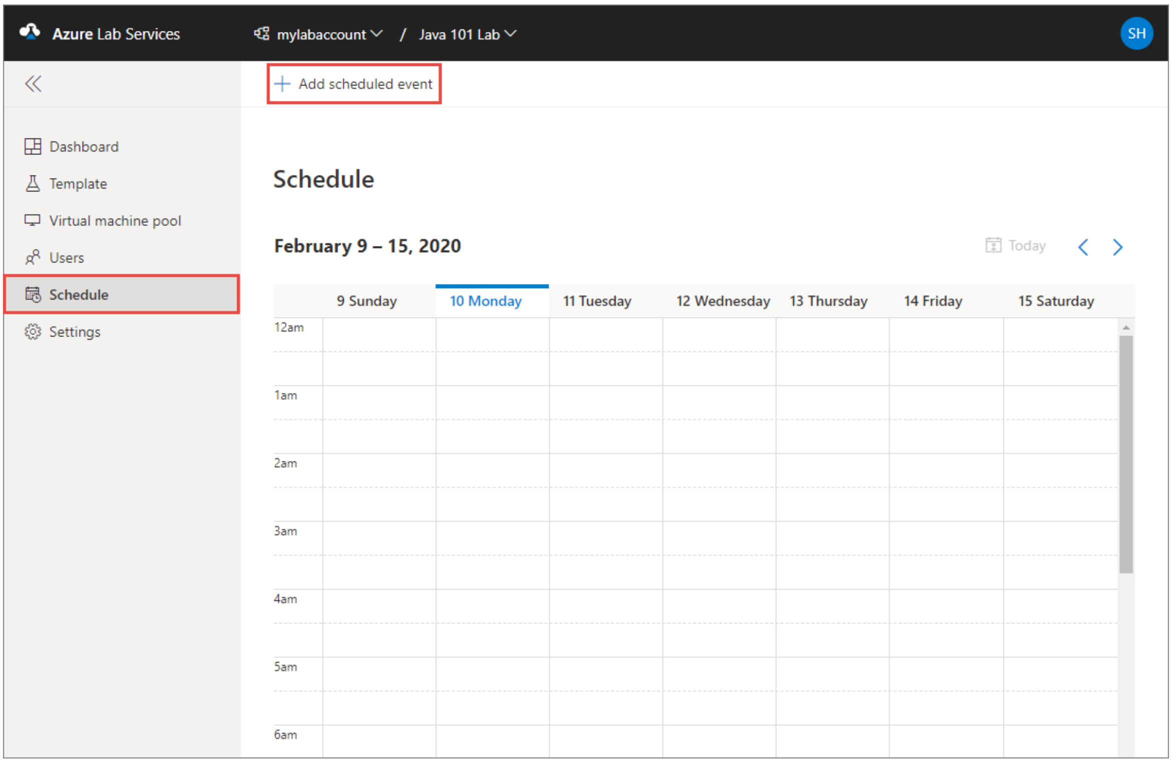
Create a scheduled event for the lab so that VMs in the lab are automatically started/stopped at specific times. The user quota you specified earlier is the additional time assigned to each user outside this scheduled time.

NOTE

Before we get started, here's how schedules affect lab virtual machines:

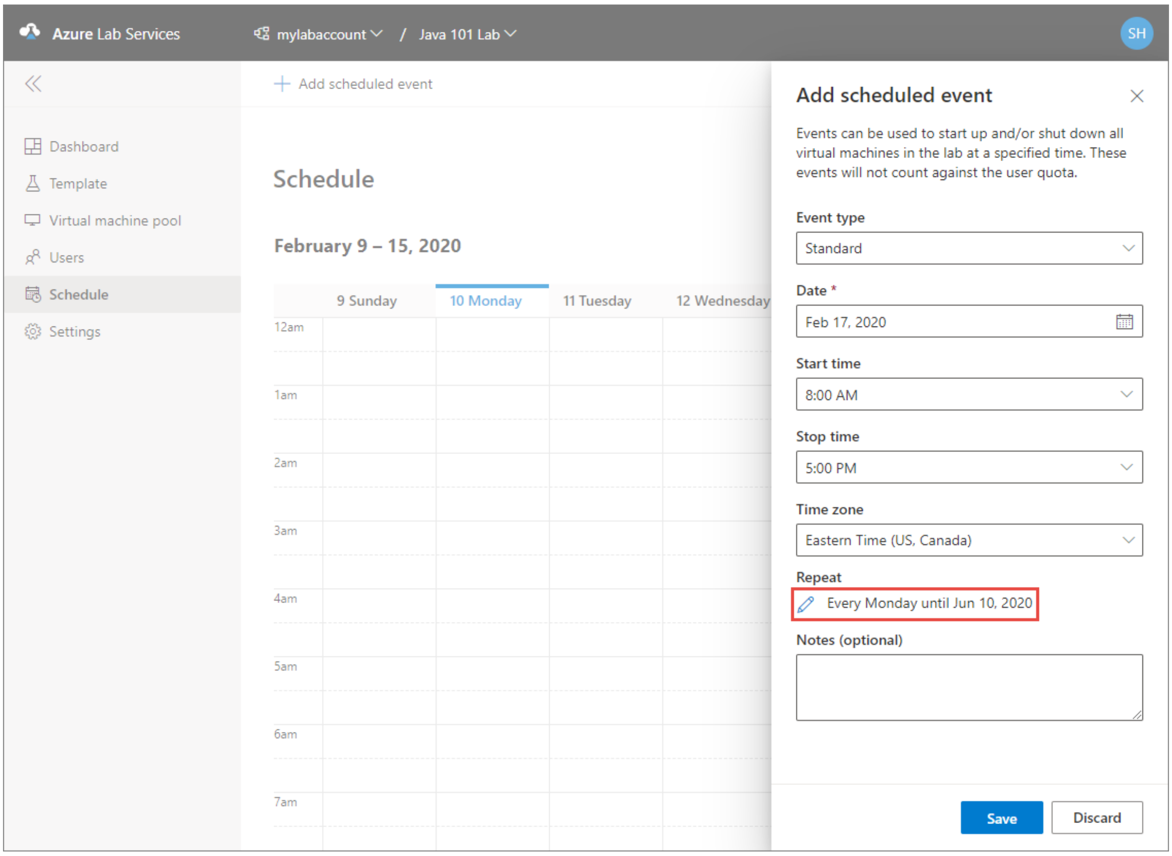
- Template virtual machine is not included in schedules.
- Only assigned virtual machines are started. This means, if a machine is not claimed by an end user (student), the the machine will not start on the scheduled hours.
- All virtual machines (whether claimed by a user or not) are stopped based on the lab schedule.

1. Switch to the **Schedules** page, and select **Add scheduled event** on the toolbar.



2. Confirm that **Standard** is selected the **Event type**. You select **Start only** to specify only the start time for the VMs. You select **Stop only** to specify only the stop time for the VMs.

3. In the **Repeat** section, select the current schedule.



4. On the **Repeat** dialog box, do the following steps:

   a. Confirm that **every week** is set for the **Repeat** field.

   b. Specify the **start date**.

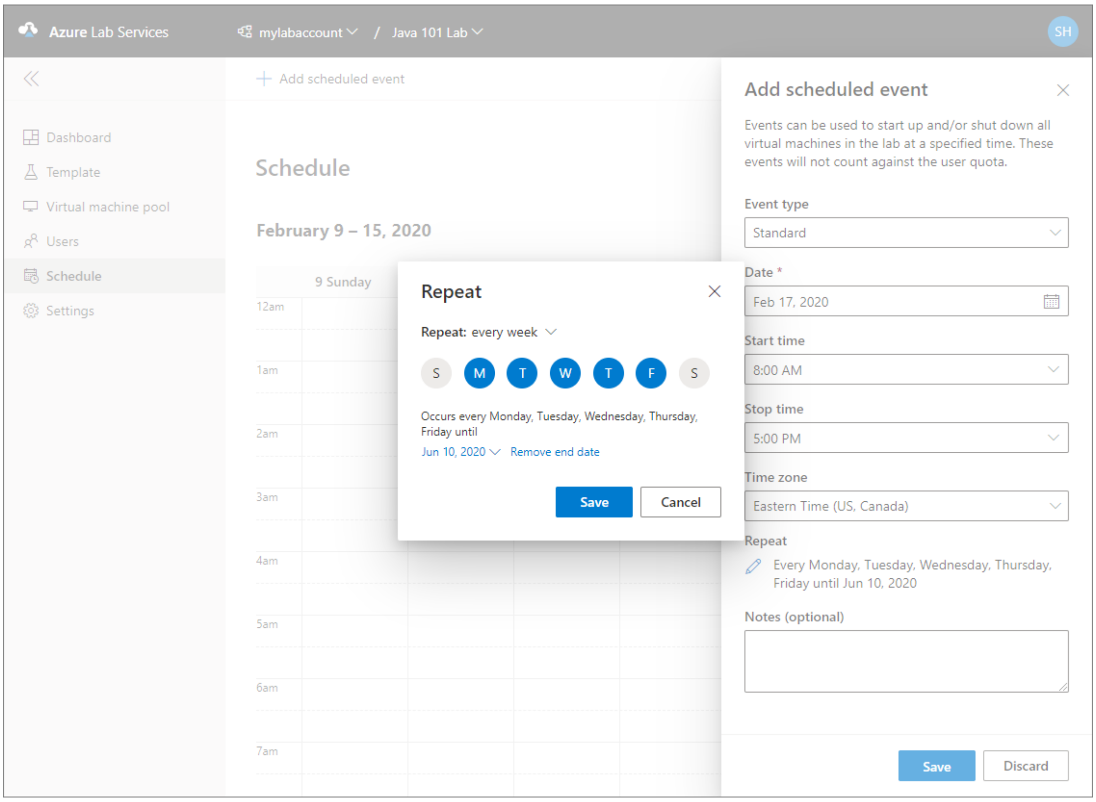
   c. Specify the **start time** at which you want the VMs to be started.

   d. Specify the **stop time** on which the VMs are to be shut down.

   e. Specify the **time zone** for the start and stop times you specified.

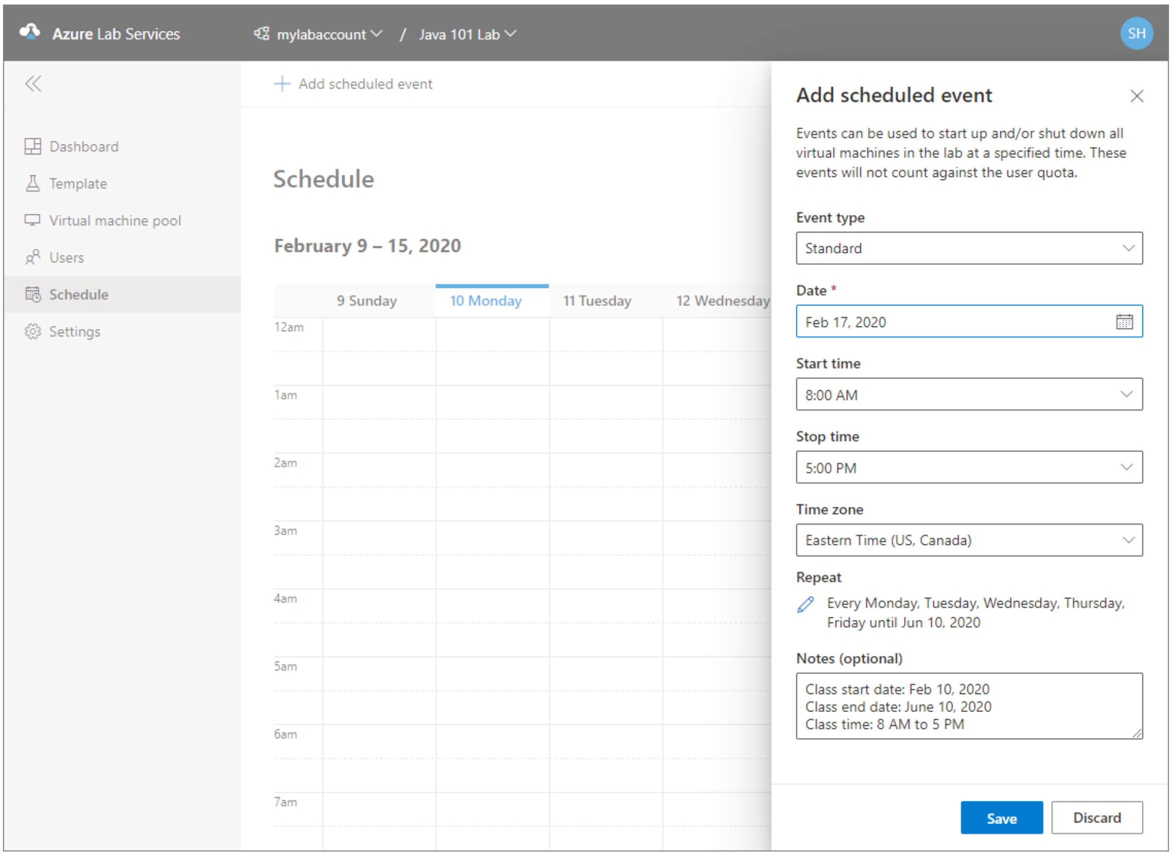
   f. Select the days on which you want the schedule to take effect. In the following example, Monday-Thursday is selected.

   g. Select **Save**.



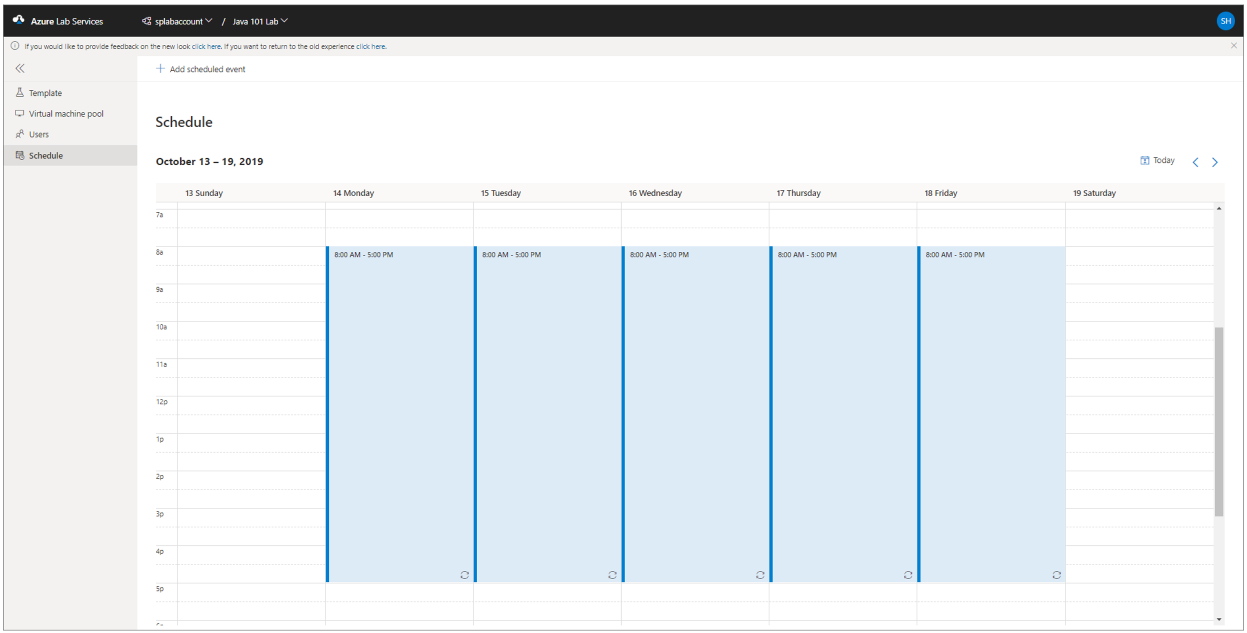
5. Now, on the **Add scheduled event** page, for **Notes (optional)**, enter any description or notes for the schedule.

6. On the **Add scheduled event** page, select **Save**.



View schedules in calendar

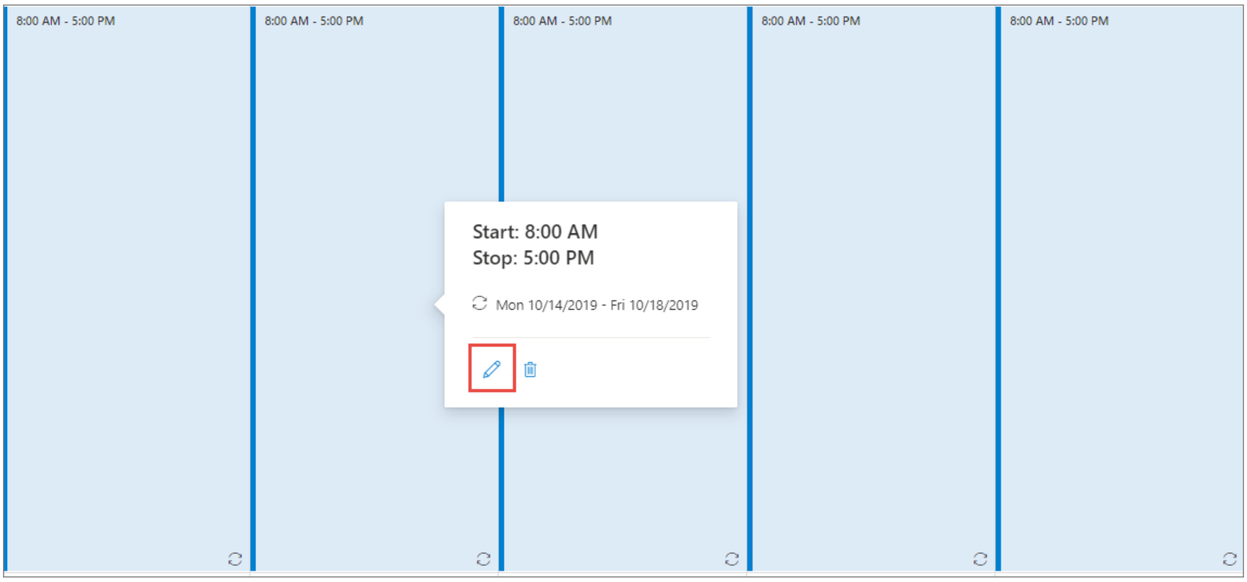
You can see the scheduled dates and times highlighted in the calendar view as shown in the following image:



Select the Today button in the top-right corner to switch to current date in the calendar. Select **left arrow** to switch to the previous week and **right arrow** to switch to the next week in the calendar.

Edit a schedule

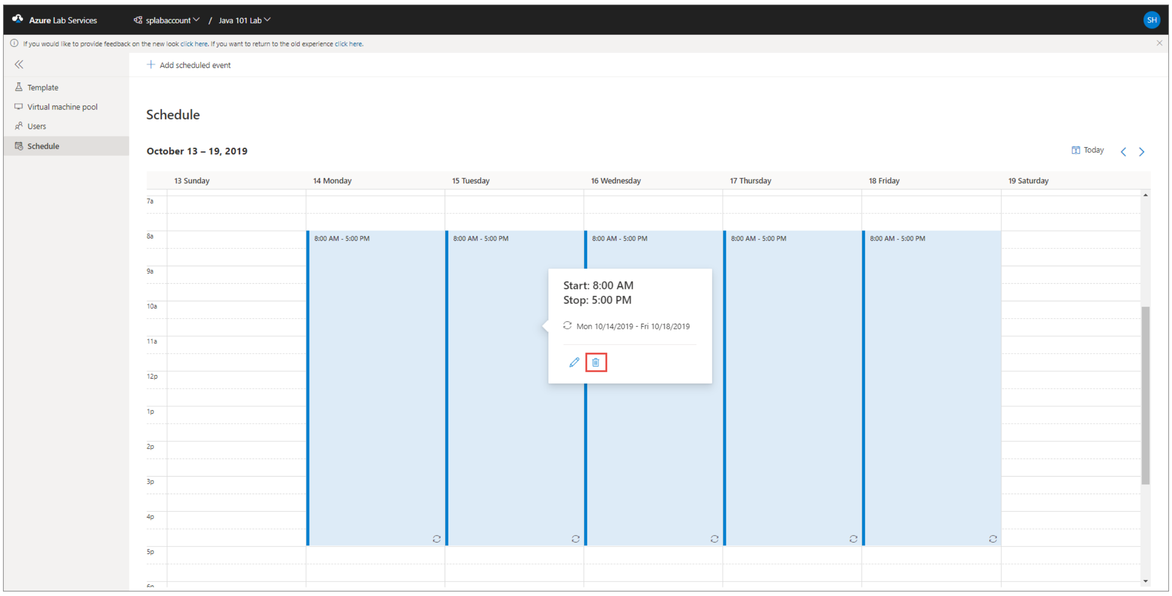
When you select a highlighted schedule in the calendar, you see buttons to **edit** or **delete** the schedule.



On the **Edit scheduled event** page, you can update the schedule, and select **Save**.

Delete a schedule

1. To delete a schedule, select a highlighted schedule in the calendar, and select the trash icon (delete) button:



2. On the **Delete scheduled event** dialog box, select **Yes** to confirm the deletion.

Next steps

See the following articles:

- As an admin, create and manage lab accounts
- As a lab owner, create and manage labs
- As a lab owner, configure and control usage of a lab
- As a lab user, access labs

Use a shared image gallery in Azure Lab Services

9/3/2021 • 4 minutes to read • Edit Online

An image contains the operating system, software applications, files, and settings that are installed on a VM. There are two types of images that you can use when you set up a new lab:

- Marketplace images that are prebuilt by Microsoft for use within Azure. These images have either Windows or Linux installed and may also include software applications. For example, the Data Science Virtual Machine image includes installed deep learning frameworks and tools.
- Custom images that are created by your institution's IT department and\or other educators. You can create both Windows and Linux custom images and have the flexibility to install Microsoft and 3rd party applications based on your unique needs. You also can add files, change application settings, and more.

This article shows how educators/lab admins can create and save a custom image from a template virtual machine to a shared image gallery so that it can be used by others to create new labs.

> **IMPORTANT**
> While using a Shared Image Gallery, Azure Lab Services supports only images with less than 128 GB of OS Disk Space. Images with more than 128 GB of disk space or multiple disks will not be shown in the list of virtual machine images during lab creation.

Scenarios

Here are the couple of scenarios supported by this feature:

- A lab account admin attaches a shared image gallery to the lab account, and uploads an image to the shared image gallery outside the context of a lab. Then, lab creators can use that image from the shared image gallery to create labs.
- A lab account admin attaches a shared image gallery to the lab account. A lab creator (instructor) saves the customized image of his/her lab to the shared image gallery. Then, other lab creators can select this image from the shared image gallery to create a template for their labs.

  When an image is saved to a shared image gallery, Azure Lab Services replicates the saved image to other regions available in the same geography. It ensures that the image is available for labs created in other regions in the same geography. Saving images to a shared image gallery incurs an additional cost, which includes cost for all replicated images. This cost is separate from the Azure Lab Services usage cost. For more information about Shared Image Gallery pricing, see Shared Image Gallery – Billing.

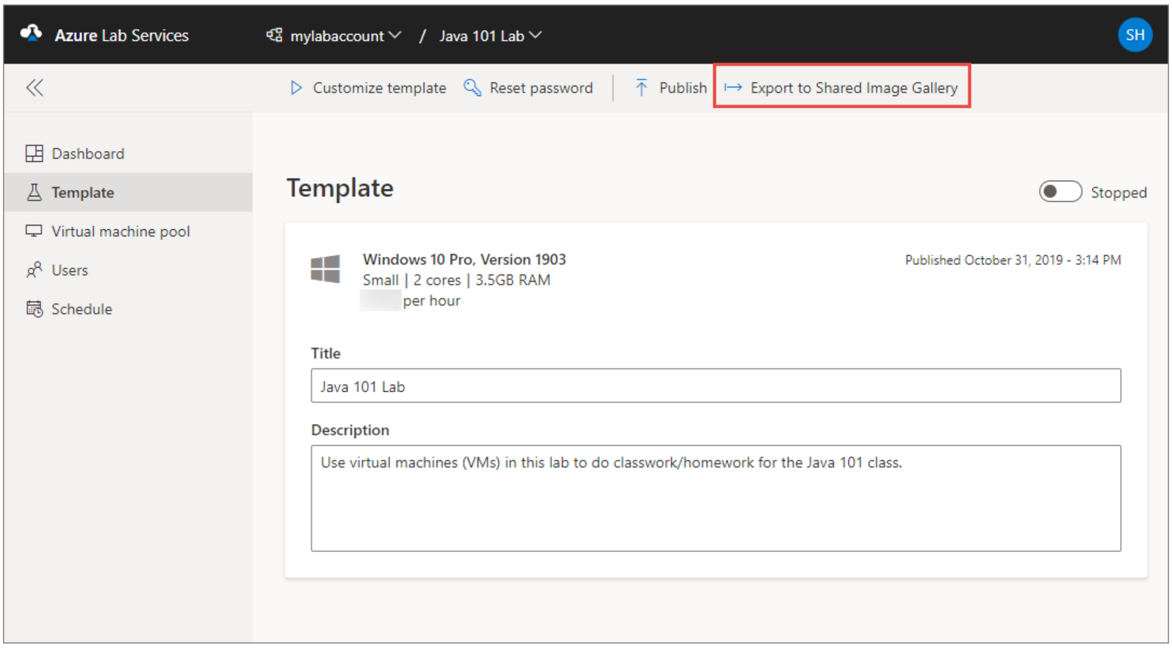
Prerequisites

- Create a shared image gallery.
- You have attached the shared image gallery to the lab account. For step-by-step instructions, see How to attach or detach shared image gallery.

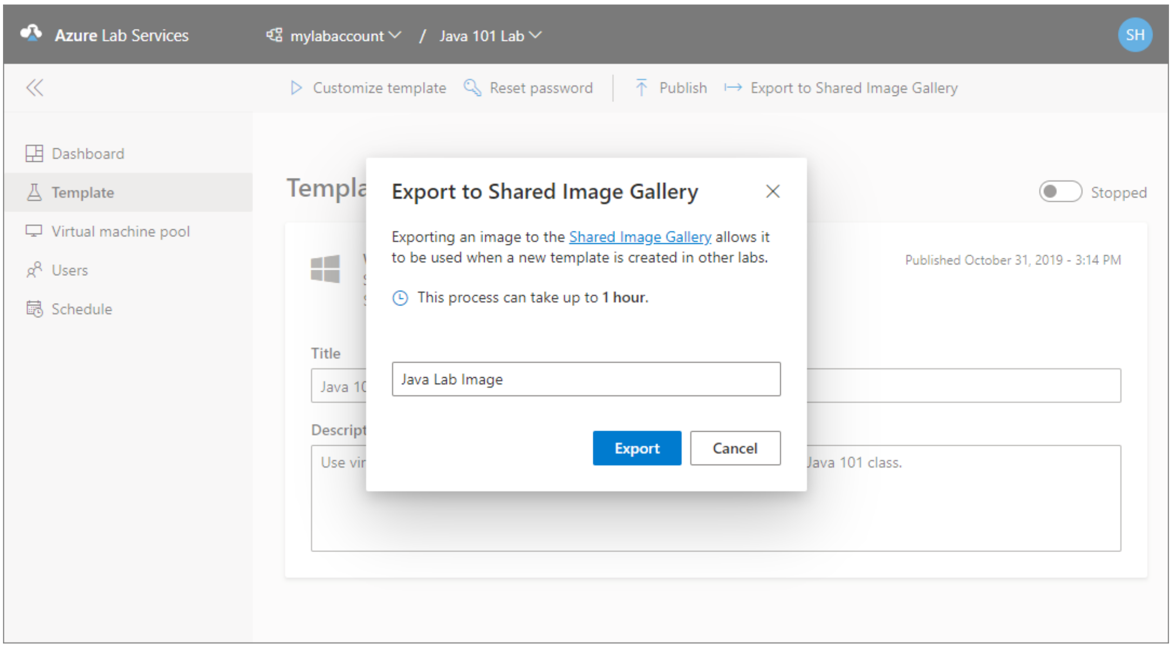
Save an image to the shared image gallery

After a shared image gallery is attached, a lab account admin or an educator can save an image to the shared image gallery so that it can be reused by other educators.

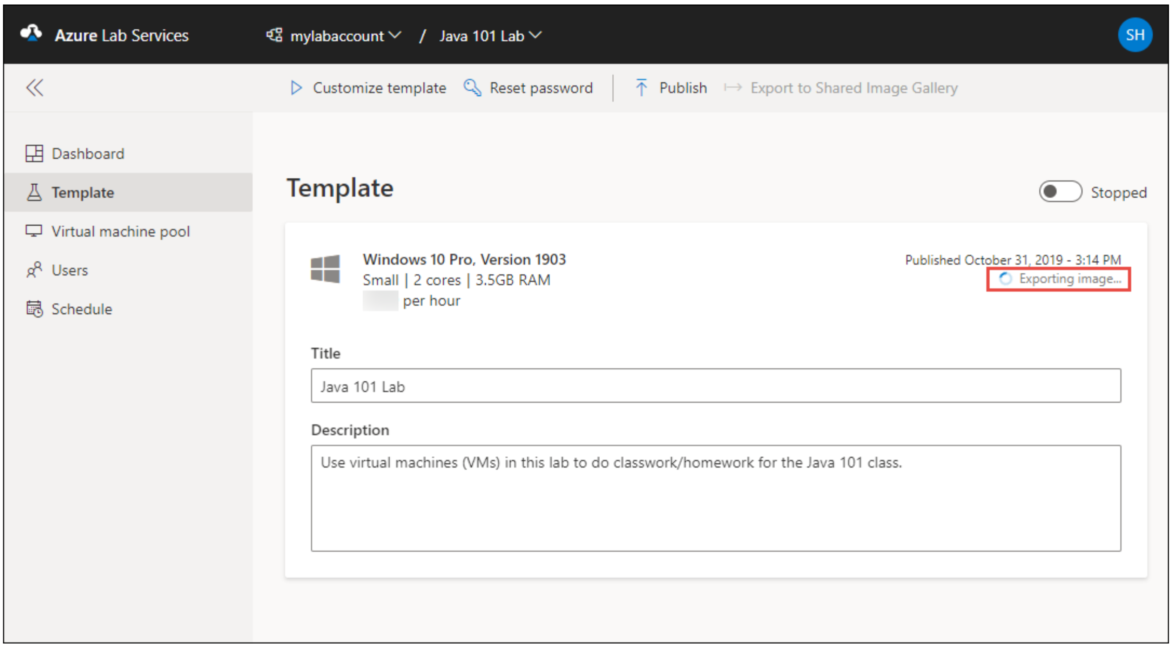
1. On the **Template** page for the lab, select **Export to Shared Image Gallery** on the toolbar.



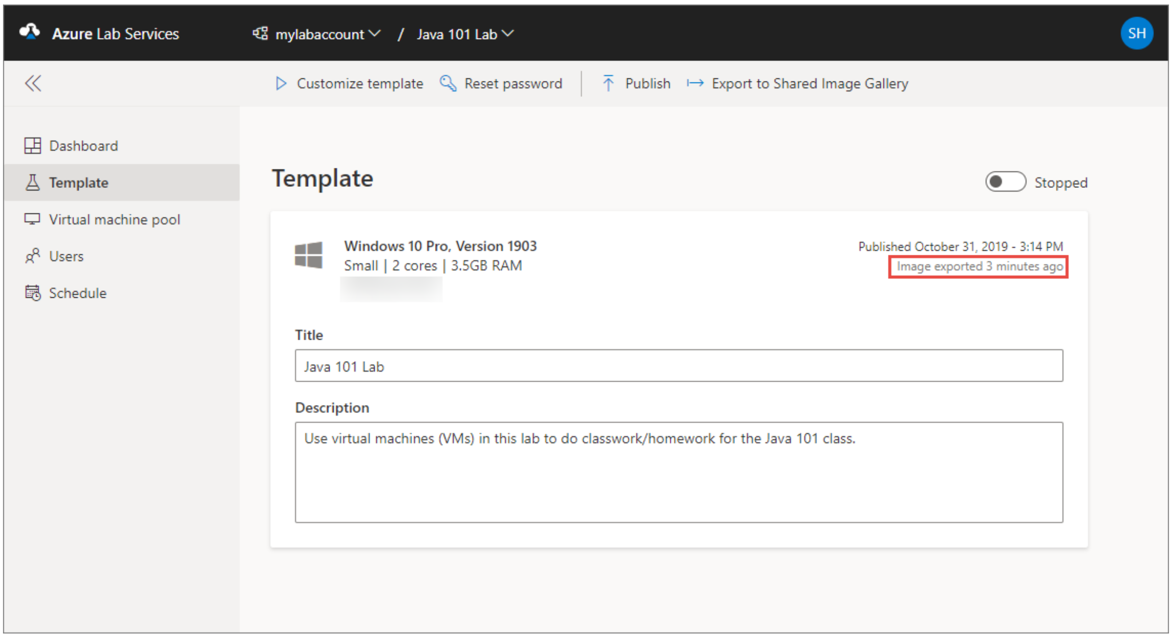
2. On the **Export to Shared Image Gallery** dialog, enter a **name for the image**, and then select **Export**.



3. You can see the progress of this operation on the **Template** page. This operation can take sometime.



4. When the export operation is successful, you see the following message:



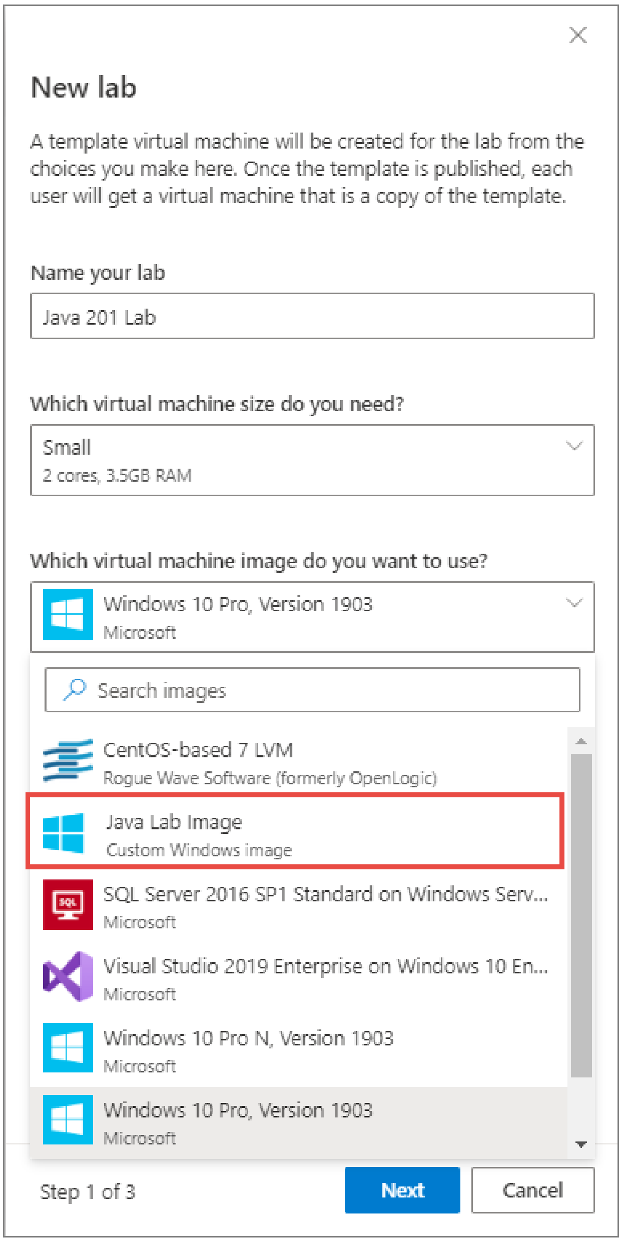
After you save the image to the shared image gallery, you can use that image from the gallery when creating another lab. You can also upload an image to the shared image gallery outside the context of a lab. For more information, see:

- Shared image gallery overview
- Recommended approaches for creating custom images

> **IMPORTANT**
>
> When you save a template image of a lab in Azure Lab Services to a shared image gallery, the image is uploaded to the gallery as a **specialized image**. Specialized images keep machine-specific information and user profiles. You can still directly upload a generalized image to the gallery outside of Azure Lab Services.

Use a custom image from the shared image gallery

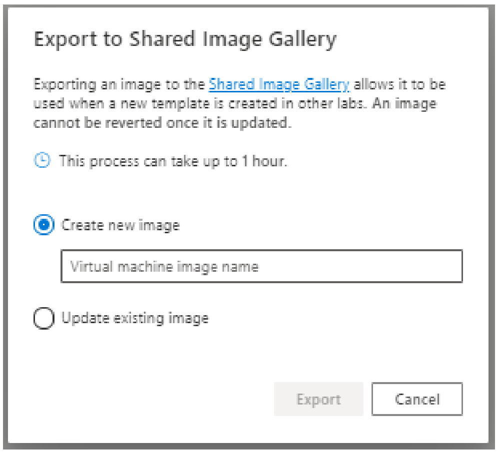
An educator can pick a custom image available in the shared image gallery for the template VM that is created when you setup a new lab.



> **NOTE**
>
> You can create a template VM based on both **generalized** and **specialized** images in Azure Lab Services.

Resave a custom image to shared image gallery

After you've created a lab from a custom image in a shared image gallery, you can make changes to the image using the template VM and reexport the image to shared image gallery. When you reexport, you have the option to either create a new image or to update the original image.



If you choose **Create new image**, a new image definition is created. This allows you to save an entirely new custom image without changing the original custom image that already exists in shared image gallery.

If instead you choose **Update existing image**, the original custom image's definition is updated with a new version. Lab Services automatically will use the most recent version the next time a lab is created using the custom image.

Next steps

To learn about how to set up shared image gallery by attaching and detaching it to a lab account, see How to attach and detach shared image gallery.

To explore other options for bringing custom images to shared image gallery outside of the context of a lab, see Recommended approaches for creating custom images.

For more information about shared image galleries in general, see shared image gallery.

Enable graphical remote desktop for Linux virtual machines in Azure Lab Services

8/5/2021 • 5 minutes to read • Edit Online

This article shows you how to do the following tasks:

- Enable graphical remote desktop sessions for a Linux VM
- How to connect to a Linux VM using RDP (Remote Desktop Protocol) or X2Go remote desktop clients

Set up graphical remote desktop solution

When a lab is created from a **Linux** image, **SSH** (Secure Shell) access is automatically configured so that the instructor can connect to the template VM from the command line using SSH. Likewise, when the template VM is published, students can also connect to their VMs using SSH.

To connect to a Linux VM using a **GUI** (graphical user interface), we recommend using either **RDP** or **X2Go**. Both of these options require the instructor to do some additional setup on the template VM:

RDP Setup

To use RDP, the instructor must:

- Enable remote desktop connection; this is specifically needed to open the VM's port for RDP.
- Install the RDP remote desktop server.
- Install a Linux graphical desktop environment (such as XFCE, MATE, and so on).

> **WARNING**
>
> We recommend using a different graphical desktop environment than GNOME. You should avoid installing GNOME on lab VMs because GNOME has a conflict with the Azure Linux Agent which is needed for the VMs to work properly in Azure Lab Services. As mentioned above, we recommend using a graphical desktop environment, such as XFCE or MATE.

X2Go Setup

To use X2Go, the instructor must:

- Install the X2Go remote desktop server.
- Install a Linux graphical desktop environment (such as XFCE, MATE, and so on).

X2Go uses the same port that is already enabled for SSH. As a result, no extra configuration required to open a port on the VM for X2Go.

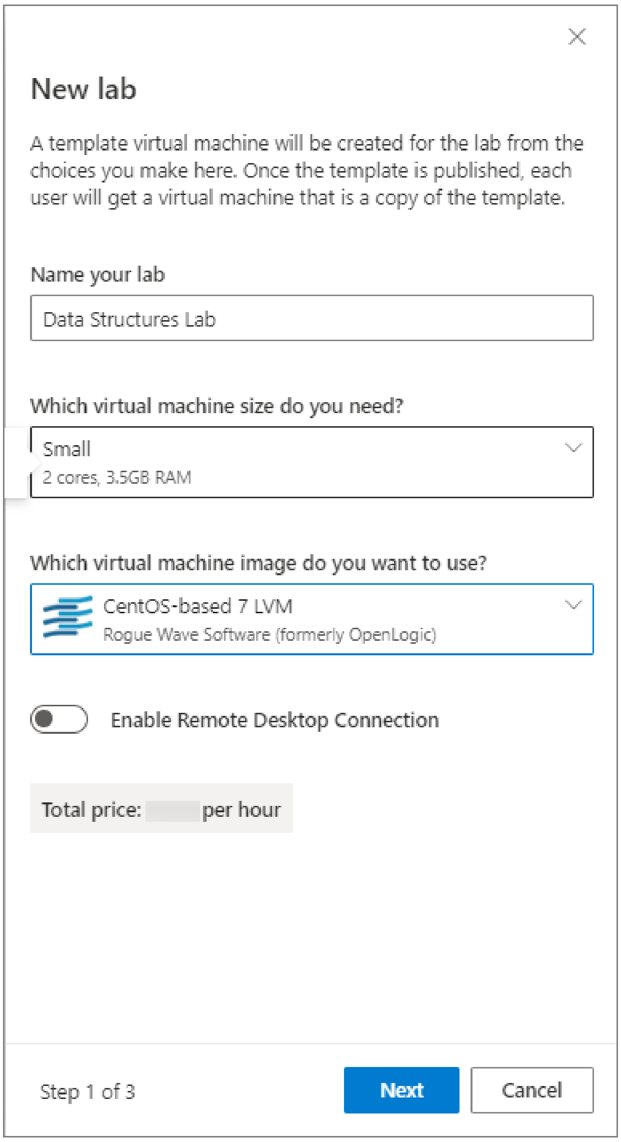
> **NOTE**
>
> In some cases, such as with Ubuntu LTS 18.04, X2Go provides better performance. If you use RDP and notice latency when interacting with the graphical desktop environment, consider trying X2Go since it may improve performance.

> **IMPORTANT**
>
> Some marketplace images already have a graphical desktop environment and remote desktop server installed. For example, the Data Science Virtual Machine for Linux (Ubuntu) already has XFCE and X2Go Server installed and configured to accept client connections.

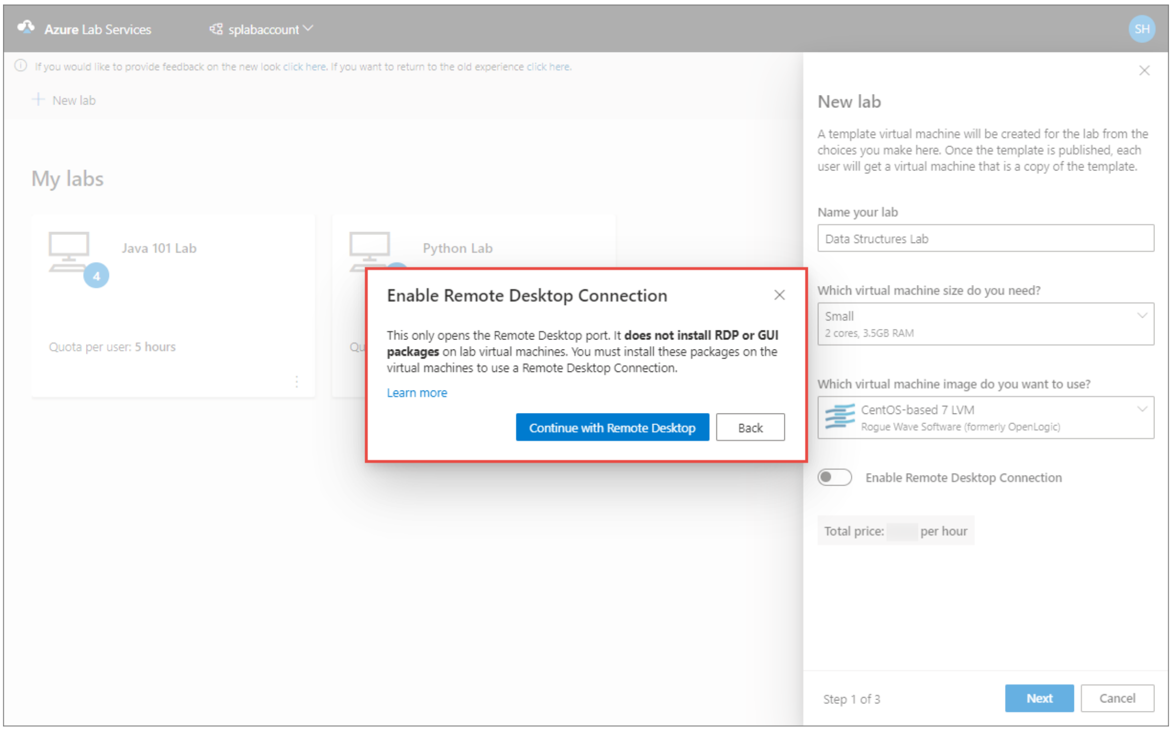
Enable remote desktop connection for RDP

This step is only needed to connect using RDP. If instead you plan to use X2Go, you can skip to the next section since X2Go uses the SSH port.

1. During lab creation, the instructor has the option to **Enable Remote Desktop Connection**. The instructor must **enable** this option to open the port on the Linux VM that is needed for an RDP remote desktop session. Otherwise, if this option is left **disabled**, only the port for SSH is opened.



2. On the **Enabling Remote Desktop Connection** message box, select **Continue with Remote Desktop**.



Install RDP or X2Go

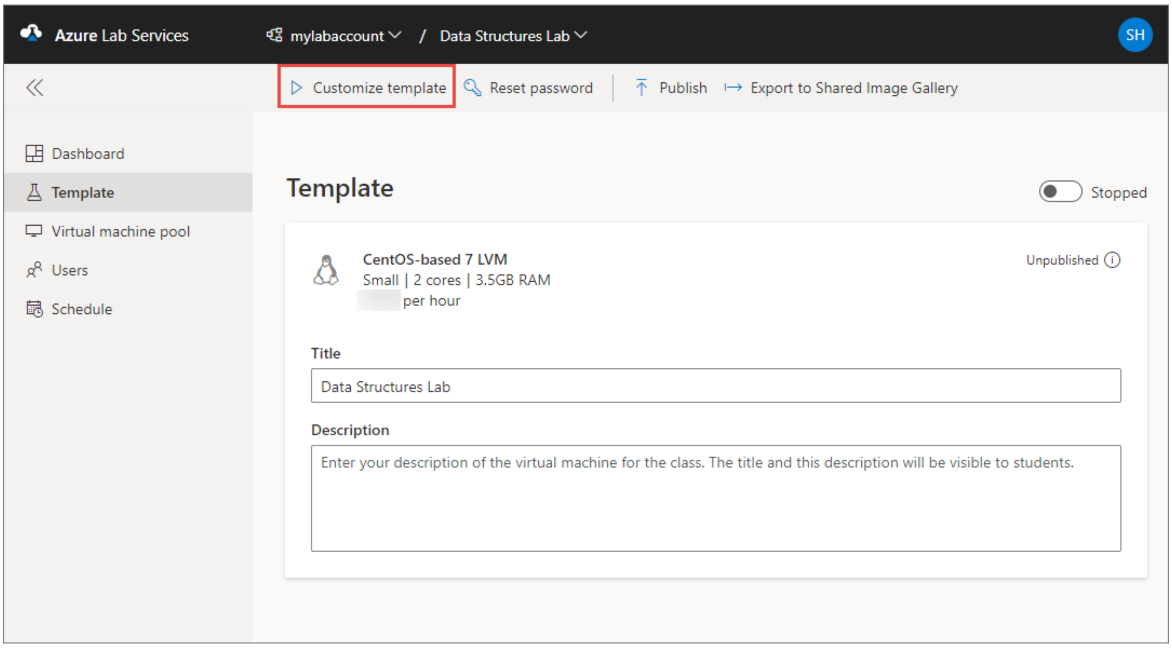
After the lab is created, the instructor needs to ensure that a graphical desktop environment and remote desktop server are installed on the template VM. Instructors must first connect to the template VM using SSH to install the packages for:

- Either the RDP or X2Go remote desktop server.
- A graphical desktop environment, such as MATE, XFCE, etc.

After this is set up, the instructor can connect to the template VM using either the **Microsoft Remote Desktop (RDP)** client or **X2Go** client.

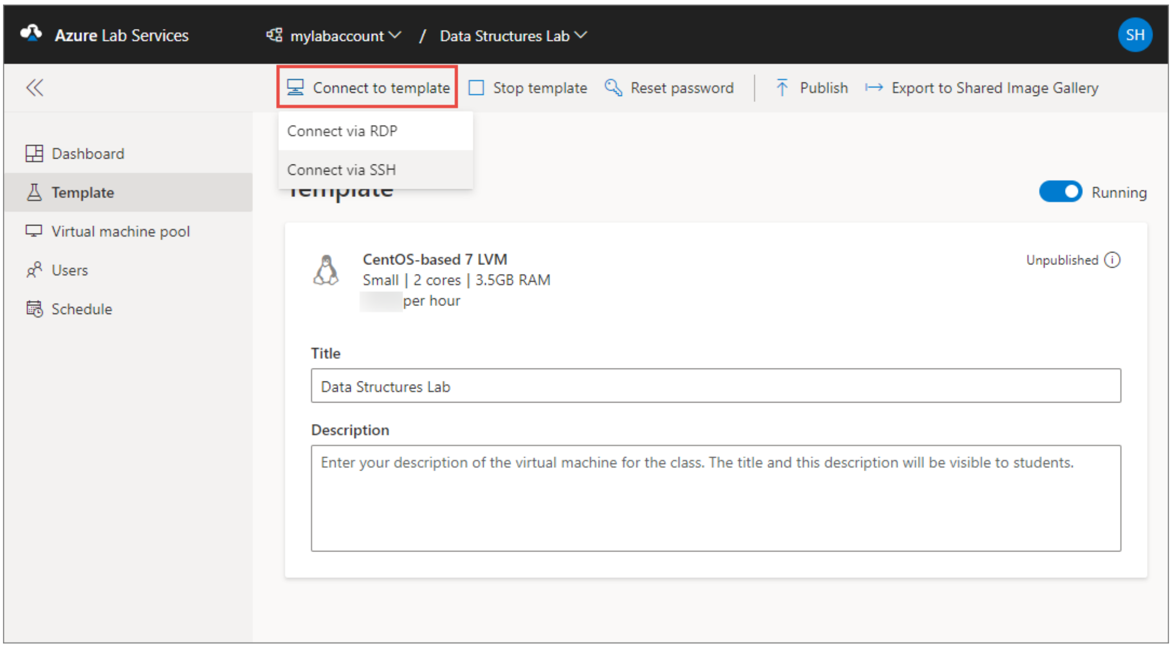
Follow the below steps to set up the template VM:

1. If you see **Customize template** on the toolbar, select it. Then, select **Continue** on the **Customize template** dialog box. This action starts the template VM.

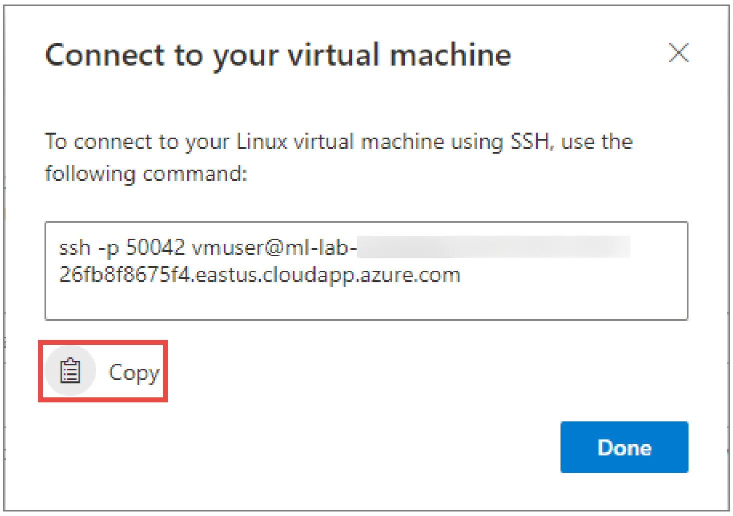


2. After the template VM is started, you can select **Connect template** and then **Connect via SSH** on the

toolbar.



3. You see the following **Connect to your virtual machine** dialog box. Select the **Copy** button next to the text box to copy it to the clipboard. Save the SSH connection information. Use this connection information from an SSH terminal (like Putty) to connect to the virtual machine.



4. Install either RDP or X2Go along with the graphical desktop environment of your choice. Refer to the following instructions:

   - Install and configure RDP
   - Install and configure X2Go

Connect to the template VM via the GUI

After the template VM is set up, the instructor can connect via the GUI using either the **Microsoft Remote Desktop (RDP)** client or **X2Go** client. The client that you use depends on if RDP or X2Go is configured as the remote desktop server on the template VM.

Microsoft Remote Desktop (RDP) client

The Microsoft Remote Desktop (RDP) client is used to connect to a template VM that has RDP configured. The Remote Desktop client can be used on Windows, Chromebooks, Macs and more. Refer to the article on Remote Desktop clients for further details.

Follow the below steps based on the type of computer used to connect to the template VM:

- Windows
  1. Click **Connect to template** on your lab's toolbar and select **Connect via RDP** to connect to the template VM.
  2. Save the RDP file and use it to connect to the template VM using the Remote Desktop client.
  3. Typically, the Remote Desktop client is already installed and configured on Windows. As a result, all you need to do is click on the RDP file to open it and start the remote session.
- Mac
  1. Click **Connect to template** on your lab's toolbar and then select **Connect via RDP** to save the RDP file.
  2. Then, refer to the how-to article Connect to a VM using RDP on a Mac.
- Chromebook
  1. Click **Connect to template** on your lab's toolbar and then select **Connect via RDP** to save the RDP file.
  2. Then, refer to the how-to article Connect to a VM using RDP on a Chromebook.

X2Go client

The X2Go client is used to connect to a template VM that has X2Go configured. Using the template VM's SSH connection information, follow the steps in the how-to article Connect to a VM using X2Go.

Next steps

After an instructor sets up either RDP or X2Go on their template VM and publishes, students can connect to their VMs via the GUI remote desktop or SSH.

For more information, see:

- Connect to a Linux VM

Configure automatic shutdown of VMs for a lab

3/5/2021 • 2 minutes to read • Edit Online

This article shows you how you can configure automatic shutdown of VMs for a lab.

You can enable several auto-shutdown cost control features to proactively prevent additional costs when the virtual machines are not being actively used. The combination of the following three automatic shutdown and disconnect features catches most of the cases where users accidentally leave their virtual machines running:

- Automatically disconnect users from virtual machines that the OS deems idle.
- Automatically shut down virtual machines when users disconnect.
- Automatically shut down virtual machines that are started but users don't connect.

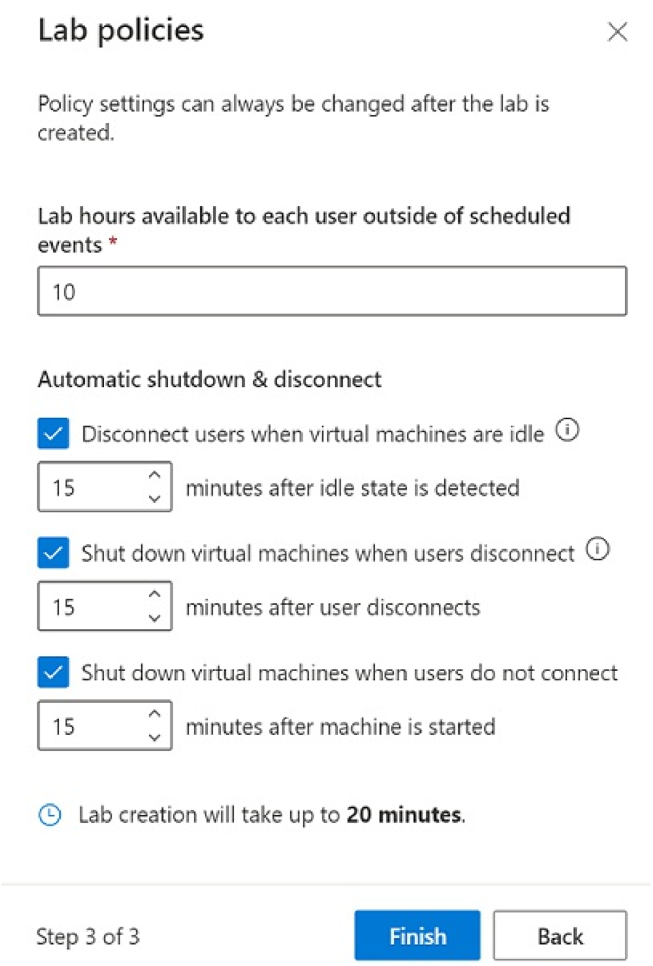
Review more details about the auto-shutdown features in the Maximize cost control with auto-shutdown settings section.

A lab account administrator can configure this setting for the lab account in which you create labs. For more information, see Configure automatic shutdown of VMs for a lab account. As a lab owner, you can override the setting when creating a lab or after the lab is created.

Configure for the lab level

You can configure the auto-shutdown setting in the Azure Lab Services.

- when creating a lab (in **Lab policies**), or
- after the lab is created (in **Settings**)



Make sure to review details about the auto-shutdown in the Maximize cost control with auto-shutdown settings section.

> **WARNING**
>
> If you shutdown the Linux or Windows operating system (OS) on a VM before disconnecting an RDP session to the VM, the autoshutdown feature will not work properly.

Next steps

Dashboard for labs

Enable nested virtualization on a template virtual machine in Azure Lab Services

4/27/2021 • 3 minutes to read • Edit Online

Currently, Azure Lab Services enables you to set up one template virtual machine in a lab and make a single copy available to each of your users. If you're an educator teaching networking, security, or IT classes, you may need to provide each of your students with an environment in which multiple virtual machines can talk to each other over a network.

Nested virtualization enables you to create a multi-VM environment inside a lab's template virtual machine. Publishing the template will provide each user in the lab with a virtual machine set up with multiple VMs within it. This article covers how to set up nested virtualization on a template machine in Azure Lab Services.

What is nested virtualization?

Nested virtualization enables you to create virtual machines within a virtual machine. Nested virtualization is done through Hyper-V, and is only available on Windows VMs.

For more information about nested virtualization, see the following articles:

- Nested Virtualization in Azure
- How to enable nested virtualization in an Azure VM

Considerations

Before setting up a lab with nested virtualization, here are a few things to take into consideration.

- When creating a new lab, select **Medium (Nested virtualization)** or **Large (Nested virtualization)** sizes for the virtual machine size. These virtual machine sizes support nested virtualization.
- Choose a size that will provide good performance for both the host and client virtual machines. Remember, when using virtualization, the size you choose must be adequate for not just one machine, but the host as well as any Hyper-V machines running concurrently.
- Client virtual machines won't have access to Azure resources, such as DNS servers, on the Azure virtual network.
- Host virtual machine requires setup to allow for the client machine to have internet connectivity.
- Client virtual machines are licensed as independent machines. See Microsoft Licensing for information about licensing for Microsoft operation systems and products. Check licensing agreements for any other software being used before setting up the template machine.

Enable nested virtualization on a template VM

This article assumes that you have created a lab account and lab. For more information about creating a new lab account, see tutorial to set up a Lab Account. For more information how to create lab, see set up a classroom lab tutorial.

> **IMPORTANT**
>
> Select **Large (nested virtualization)** or **Medium (nested virtualization)** for the virtual machine size when creating the lab. Nested virtualization will not work otherwise.

To connect to the template machine, see create and manage a classroom template.

To enable nested virtualization, there are a few tasks to accomplish.

- **Enable Hyper-V role**. Hyper-V role must be enabled for the creation and running of Hyper-V virtual machines on the Lab Services virtual machine.
- **Enable DHCP**. When the Lab Services virtual machine has the DHCP role enabled, the Hyper-V virtual machines can automatically be assigned an IP address.
- **Create NAT network for Hyper-V VMs**. The NAT network is set up to allow the Hyper-V virtual machines to have internet access. The Hyper-V virtual machines can communicate with each other.

> **NOTE**
>
> The NAT network created on the Lab Services VM will allow a Hyper-V VM to access the internet and other Hyper-V VMs on the same Lab Services VM. The Hyper-V VM won't be able to access Azure resources, such as DNS servers, on the Azure virtual network.

Accomplishing the tasks listed above can be done using a script or using Windows tools. Read the sections below for further details.

Using script to enable nested virtualization

To use the automated setup for nested virtualization with Windows Server 2016 or Windows Server 2019, see Enable nested virtualization on a template virtual machine in Azure Lab Services using a script. You will use scripts from Lab Services Hyper-V scripts to install the Hyper-V role. The scripts will also set up networking so the Hyper-V virtual machines can have internet access.

Using Windows tools to enable nested virtualization

The setup nested virtualization for Windows Server 2016 or Windows Server 2019 using Windows roles and administrative tools, see Enable nested virtualization on a template virtual machine in Azure Lab Services manually. Instructions will also cover how to set up networking so the Hyper-V virtual machines can have internet access.

Enable nested virtualization on a template virtual machine in Azure Lab Services manually

3/5/2021 • 7 minutes to read • Edit Online

Nested virtualization enables you to create a multi-VM environment inside a lab's template virtual machine. Publishing the template will provide each user in the lab with a virtual machine set up with multiple VMs within it. For more information about nested virtualization and Azure Lab Services, see Enable nested virtualization on a template virtual machine in Azure Lab Services.

This article covers how to set up nested virtualization on a template machine in Azure Lab Services using Windows roles and tools directly. There are a few things needed to enable a class to use nested virtualization. The steps below will describe how to manually set up a Lab Services machine template with Hyper-V. Steps are intended for Windows Server 2016 or Windows Server 2019.

> **IMPORTANT**
>
> Select **Large (nested virtualization)** or **Medium (nested virtualization)** for the virtual machine size when creating the lab. Nested virtualization will not work otherwise.

Enable Hyper-V role

The following steps describe actions needed to enable Hyper-V on Windows Server using either Server Manager. Once the installation is successful, Hyper-V manager will be available to add, modify, and delete client virtual machines.

1. In **Server Manager**, on the Dashboard page, click **Add Roles and Features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, keep the default selection of Role-based or feature-based installation and then click **Next**.
4. On the **Select destination server** page, select Select a server from the server pool. The current server will already be selected. Click Next.
5. On the **Select server roles** page, select **Hyper-V**.
6. The **Add Roles and Features Wizard** pop-up will appear. Select **Include management tools (if applicable)**. Click the **Add Features** button.
7. On the **Select server roles** page, click **Next**.
8. On the **Select features page**, click **Next**.
9. On the **Hyper-V** page, click **Next**.
10. On the **Create Virtual Switches** page, accept the defaults, and click **Next**.
11. On the **Virtual Machine Migration** page, accept the defaults, and click **Next**.
12. On the **Default Stores** page, accept the defaults, and click **Next**.
13. On the **Confirm installation selections** page, select **Restart the destination server automatically if required**.
14. When the **Add Roles and Features Wizard** pop-up appears, click **Yes**.
15. Click **Install**.
16. Wait for the **Installation progress** page to indicate that the Hyper-V role is complete. The machine may restart in the middle of the installation.

17. Click **Close**.

Enable DHCP role

Any Hyper-V client virtual machines created, needs an IP address in the NAT network. We'll create the NAT network later. One way to assign IP addresses is to set up the host, in this case the lab virtual machine template, as a DHCP server. Below are the steps required to enable the DHCP role.

1. In **Server Manager**, on the **Dashboard** page, click **Add Roles and Features**.

2. On the **Before you begin** page, click **Next**.

3. On the **Select installation type** page, select **Role-based or feature-based installation** and then click **Next**.

4. On the **Select destination server** page, select the current server from the server pool and then click **Next**.

5. On the **Select server roles** page, select **DHCP Server**.

6. The **Add Roles and Features Wizard** pop-up will appear. Select **Include management tools (if applicable)**. Click **Add Features**.

   > **NOTE**
   >
   > You may see a validation error stating that no static IP addresses were found. This warning can be ignored for our scenario.

7. On the **Select server roles** page, click **Next**.

8. On the **Select features** page, click **Next**.

9. On the **DHCP Server** page, click **Next**.

10. On the **Confirm installation selections** page, click **Install**.

11. Wait for the **Installation progress page** to indicate that the DHCP role is complete.

12. Click Close.

Enable Routing and Remote Access role

1. In **Server Manager**, on the **Dashboard** page, click **Add Roles and Features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation** and then click **Next**.
4. On the **Select destination server** page, select the current server from the server pool and then click **Next**.
5. On the **Select server roles** page, select **Remote Access**. Click **OK**.
6. On the **Select features** page, click **Next**.
7. On the **Remote Access** page, click **Next**.
8. On the **Role Services** page, select **Routing**.
9. The **Add Roles and Features Wizard** pop-up will appear. Select **Include management tools (if applicable)**. Click **Add Features**.
10. Click **Next**.
11. On the **Web Server Role (IIS)** page, click **Next**.
12. On the **Select role services** page, click **Next**.

13. On the **Confirm installation selections** page, click **Install**.
14. Wait for the **Installation progress** page to indicate that the Remote Access role is complete.
15. Click **Close**.

Create virtual NAT network

Now that all the necessary roles have been installed, it's time to create the NAT network. The creation process will involve creating a switch and the NAT network, itself. A NAT (network address translation) network assigns a public IP address to a group of VMs on a private network to allow connectivity to the internet. In our case, the group of private VMs will be the nested VMs. The NAT network will allow the nested VMs to communicate with one another. A switch is a network device that handles receiving and routing of traffic in a network.

Create a new virtual switch

1. Open **Hyper-V Manager** from Windows Administrative Tools.
2. Select the current server in the left-hand navigation menu.
3. Click **Virtual Switch Manager...** from the **Actions** menu on the right-hand side of the **Hyper-V Manager**.
4. On the **Virtual Switch Manager** pop-up, select **Internal** for the type of switch to create. Click **Create Virtual Switch**.
5. For the newly created virtual switch, set the name to something memorable. For this example, we'll use 'LabServicesSwitch'. Click **OK**.
6. A new network adapter will be created. The name will be similar to 'vEthernet (LabServicesSwitch)'. To verify open the **Control Panel**, click **Network and Internet**, click **View network status and tasks**. On the left, click **Change adapter settings**.

Create a NAT network

1. Open the **Routing and Remote Access** tool from Windows Administrative Tools.
2. Select the local server in the left navigation page.
3. Choose **Action** -> **Configure and Enable Routing and Remote Access**.
4. When **Routing and Remote Access Server Setup Wizard** appears, click **Next**.
5. On the **Configuration** page, select **Network address translation (NAT)** configuration. Click **Next**.

   > **WARNING**
   >
   > Do not choose the 'Virtual private network (VPN) access and NAT' option.

6. On **NAT Internet Connection** page, choose 'Ethernet'. Don't choose the 'vEthernet (LabServicesSwitch)' connection we created in Hyper-V Manager. Click **Next**.
7. Click **Finish** on the last page of the wizard.
8. When the **Start the service** dialog appears, click **Start Service**.
9. Wait until service is started.

Update network adapter settings

The network adapter will be associated with the IP used for the default gateway IP for the NAT network created earlier. In this example, we create an IP address of 192.168.0.1 with a subnet mask of 255.255.255.0. We will use the virtual switch created earlier.

1. Open the **Control Panel**, click **Network and Internet**, click **View network status and tasks**.

2. On the left, click **Change adapter settings**.

3. In the **Network Connections** window, double-click on 'vEthernet (LabServicesSwitch)' to show the **vEthernet (LabServicesSwitch) Status** details dialog.

4. Click the **Properties** button.

5. Select **Internet Protocol Version 4 (TCP/IPv4)** item and click the **Properties** button.

6. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog, select **Use the following IP address**. For the ip address, enter 192.168.0.1. For the subnet mask, enter 255.255.255.0. Leave the default gateway blank. Leave the DNS servers blank as well.

> **NOTE**
>
> Our range for our NAT network will be, in CIDR notation, 192.168.0.0/24. This creates a range of usable IP addresses from 192.168.0.1 to 192.168.0.254. By convention, gateways have the first IP address in a subnet range.

7. Click OK.

Create DHCP Scope

The following steps are instruction to add DHCP scope. In this article, our NAT network is 192.168.0.0/24 in CIDR notation. This creates a range of usable IP addresses from 192.168.0.1 to 192.168.0.254. The scope created must be in that range of usable addresses excluding the IP address already created earlier.

1. Open **Administrative Tools** and open the **DHCP** administrative tool.

2. In the **DHCP** tool, expand the node for the current server and select **IPv4**.

3. From the Action menu, choose **New Scope...**

4. When the **New Scope Wizard** appears, click **Next** on the **Welcome** page.

5. On the **Scope Name** page, enter 'LabServicesDhcpScope' or something else memorable for the name. Click **Next**.

6. On the **IP Address Range** page, enter the following values.

   - 192.168.0.100 for the Start IP address
   - 192.168.0.200 for the End IP address
   - 24 for the Length
   - 255.255.255.0 for the Subnet mask

7. Click **Next**.

8. On the **Add Exclusions and Delay** page, click **Next**.

9. On the **Lease Duration** page, click **Next**.

10. On the **Configure DHCP Options** page, select **Yes, I want to configure these options now**. Click **Next**.

11. On the **Router (Default Gateway)**

12. Add 192.168.0.1, if not done already. Click **Next**.

13. On the **Domain Name and DNS Servers** page, add 168.63.129.16 as a DNS server IP address, if not done already. 168.63.129.16 is the IP address for an Azure static DNS server. Click **Next**.

14. On the **WINS Servers** page, click **Next**.

15. One the **Activate Scope** page, select **Yes, I want to activate this scope now**. Click **Next**.

16. On the **Completing the New Scope Wizard** page, click **Finish**.

Conclusion

Now your template machine is ready to create Hyper-V virtual machines. See Create a Virtual Machine in Hyper-V for instructions about how to create Hyper-V virtual machines. Also see the Microsoft Evaluation Center to check out available operating systems and software.

Next steps

Next steps are common to setting up any lab.

- Add users
- Set quota
- Set a schedule
- Email registration links to students

Enable nested virtualization on a template virtual machine in Azure Lab Services using a script

11/2/2020 • 2 minutes to read • Edit Online

Nested virtualization enables you to create a multi-VM environment inside a lab's template virtual machine. Publishing the template will provide each user in the lab with a virtual machine set up with multiple VMs within it. For more information about nested virtualization and Azure Lab Services, see Enable nested virtualization on a template virtual machine in Azure Lab Services.

The steps in this article focus on setting up nested virtualization for Windows Server 2016, Windows Server 2019, or Windows 10. You will use a script to set up template machine with Hyper-V. The following steps will guide you through how to use the Lab Services Hyper-V scripts.

> **IMPORTANT**
>
> Select **Large (nested virtualization)** or **Medium (nested virtualization)** for the virtual machine size when creating the lab. Nested virtualization will not work otherwise.

Run script

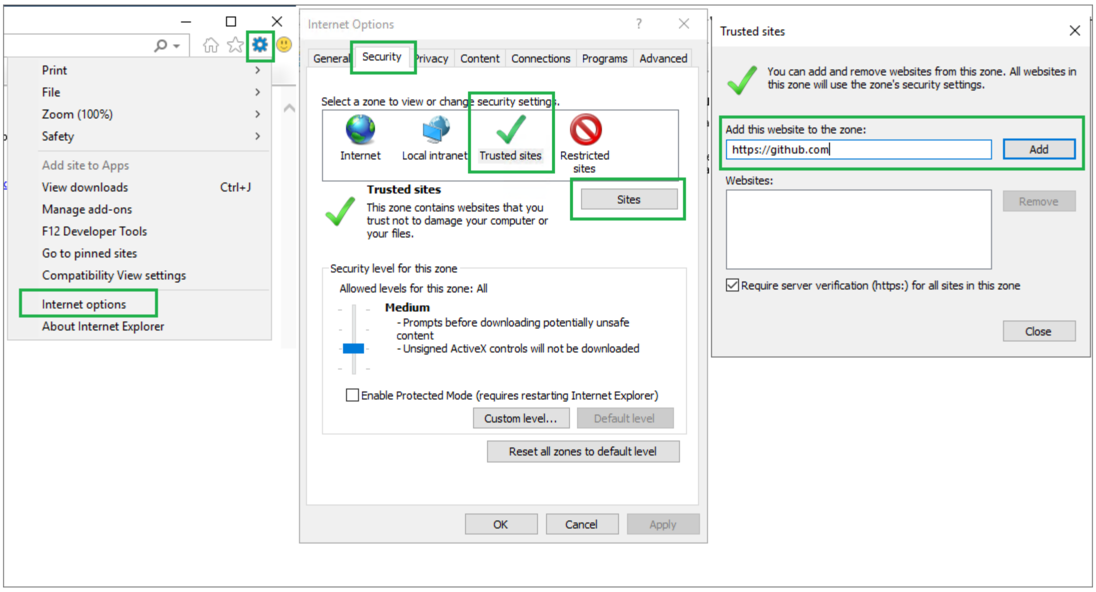
1. If you're using Internet Explorer, you may have to add `https://github.com` to the trusted sites list.

   a. Open Internet Explorer.

   b. Select the gear icon, and choose **Internet options**.

   c. When the **Internet Options** dialog appears, select **Security**, select **Trusted Sites**, click **Sites** button.

   d. When the **Trusted sites** dialog appears, add `https://github.com` to the trusted websites list, and select **Close**.



2. Download the Git repository files as outlined in the following steps.

   a. Go to https://github.com/Azure/azure-devtestlab/.

b. Click the **Clone or Download** button.

c. Click **Download ZIP**.

d. Extract the ZIP file

> **TIP**
>
> You can also clone the Git repository at https://github.com/Azure/azure-devtestlab.git.

3. Launch **PowerShell** in **Administrator** mode.

4. In the PowerShell window, navigate to the folder with the downloaded script. If you're navigating from the top folder of the repository files, the script is located at `azure-devtestlab\samples\ClassroomLabs\Scripts\HyperV\`.

5. You may have to change the execution policy to successfully run the script. Run the following command:

```
Set-ExecutionPolicy bypass -force
```

6. Run the script:

```
.\SetupForNestedVirtualization.ps1
```

> **NOTE**
>
> The script may require the machine to be restarted. Follow instructions from the script and re-run the script until `Script completed` is seen in the output.

7. Don't forget to reset the execution policy. Run the following command:

```
Set-ExecutionPolicy default -force
```

Conclusion

Now your template machine is ready to create Hyper-V virtual machines. See Create a Virtual Machine in Hyper-V for instructions on how to create Hyper-V virtual machines. Also, see Microsoft Evaluation Center to check out available operating systems and software.

Next steps

Next steps are common to setting up any lab.

- Add users
- Set quota
- Set a schedule
- Email registration links to students

Set up a lab with GPU virtual machines

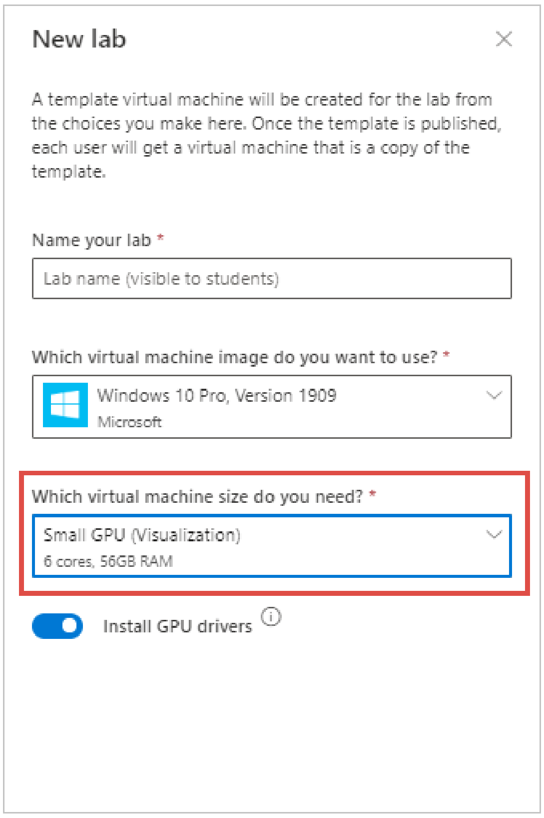
5/10/2021 • 6 minutes to read • Edit Online

This article shows you how to do the following tasks:

- Choose between *visualization* and *compute* graphics processing units (GPUs).
- Ensure that the appropriate GPU drivers are installed.

Choose between visualization and compute GPU sizes

On the first page of the lab creation wizard, in the **Which virtual machine size do you need?** drop-down list, you select the size of the VMs that are needed for your class.



In this process, you have the option of selecting either **Visualization** or **Compute** GPUs. It's important to choose the type of GPU that's based on the software that your students will use.

As described in the following table, the *compute* GPU size is intended for compute-intensive applications. For example, the Deep Learning in Natural Language Processing class type uses the **Small GPU (Compute)** size. The compute GPU is suitable for this type of class, because students use deep learning frameworks and tools that are provided by the Data Science Virtual Machine image to train deep learning models with large sets of data.

SIZE	CORES	RAM	DESCRIPTION
Small GPU (Compute)	- 6 cores - 56 GB RAM	Standard_NC6	This size is best suited for compute-intensive applications such as artificial intelligence (AI) and deep learning.

The *visualization* GPU sizes are intended for graphics-intensive applications. For example, the SOLIDWORKS engineering class type shows using the **Small GPU (Visualization)** size. The visualization GPU is suitable for this type of class, because students interact with the SOLIDWORKS 3D computer-aided design (CAD) environment for modeling and visualizing solid objects.

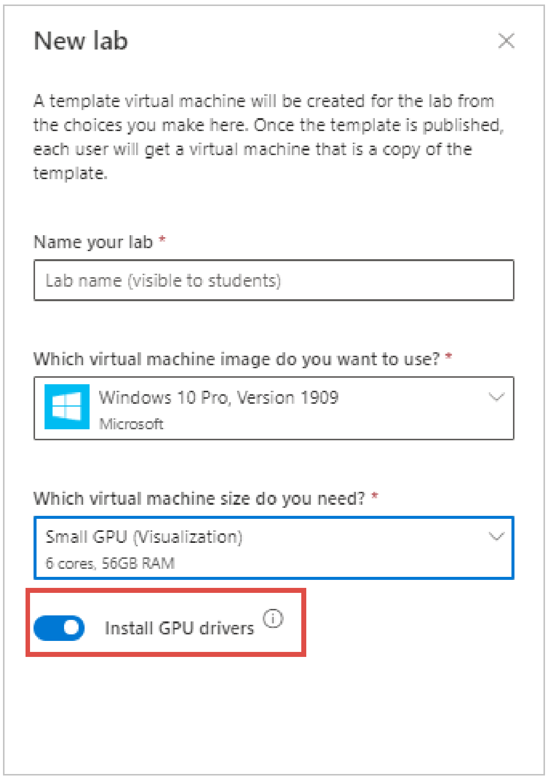
SIZE	CORES	RAM	DESCRIPTION
Small GPU (Visualization)	- 6 cores - 56 GB RAM	Standard_NV6	This size is best suited for remote visualization, streaming, gaming, and encoding that use frameworks such as OpenGL and DirectX.
Medium GPU (Visualization)	- 12 cores - 112 GB RAM	Standard_NV12	This size is best suited for remote visualization, streaming, gaming, and encoding that use frameworks such as OpenGL and DirectX.

NOTE

You may not see some of these VM sizes in the list when creating a classroom lab. The list is populated based on the current capacity of the lab's location. If the lab account creator allows lab creators to pick a location for the lab, you may try choosing a different location for the lab and see if the VM size is available. For availability of VMs, see Products available by region.

Ensure that the appropriate GPU drivers are installed

To take advantage of the GPU capabilities of your lab VMs, ensure that the appropriate GPU drivers are installed. In the lab creation wizard, when you select a GPU VM size, you can select the **Install GPU drivers** option.



As shown in the preceding image, this option is enabled by default, which ensures that recently released drivers are installed for the type of GPU and image that you selected:

- When you select a *compute* GPU size, your lab VMs are powered by the NVIDIA Tesla K80 GPU. In this case, recent Compute Unified Device Architecture (CUDA) drivers are installed, which enables high-performance computing.
- When you select a *visualization* GPU size, your lab VMs are powered by the NVIDIA Tesla M60 GPU and GRID technology. In this case, recent GRID drivers are installed, which enables the use of graphics-intensive applications.

> **IMPORTANT**
>
> The **Install GPU drivers** option only installs the drivers when they aren't present on your lab's image. For example, the GPU drivers are already installed on the Azure marketplace's Data Science image. If you create a lab using the Data Science image and choose to **Install GPU drivers**, the drivers won't be updated to a more recent version. To update the drivers, you will need to manually install them as explained in the next section.

Install the drivers manually

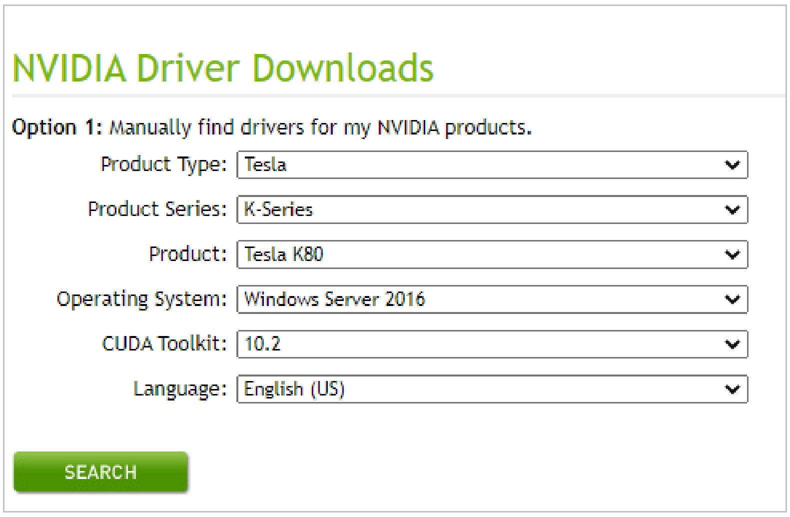
You might need to install a different version of the drivers than the version that Azure Lab Services installs for you. This section shows how to manually install the appropriate drivers, depending on whether you're using a *compute* GPU or a *visualization* GPU.

Install the compute GPU drivers

To manually install drivers for the *compute* GPU size, do the following:

1. In the lab creation wizard, when you're creating your lab, disable the **Install GPU drivers** setting.

2. After your lab is created, connect to the template VM to install the appropriate drivers.



   a. In a browser, go to the NVIDIA Driver Downloads page.
   b. Set the **Product Type** to **Tesla**.
   c. Set the **Product Series** to **K-Series**.
   d. Set the **Operating System** according to the type of base image you selected when you created your lab.
   e. Set the **CUDA Toolkit** to the version of CUDA driver that you need.
   f. Select **Search** to look for your drivers.
   g. Select **Download** to download the installer.
   h. Run the installer so that the drivers are installed on the template VM.

3. Validate that the drivers are installed correctly by following the instructions in the Validate the installed drivers section.

4. After you've installed the drivers and other software that are required for your class, select **Publish** to create your students' VMs.

> **NOTE**
>
> If you're using a Linux image, after you've downloaded the installer, install the drivers by following the instructions in Install CUDA drivers on Linux.

Install the visualization GPU drivers

To manually install drivers for the *visualization* GPU sizes, do the following:

1. In the lab creation wizard, when you're creating your lab, disable the **Install GPU drivers** setting.

2. After your lab is created, connect to the template VM to install the appropriate drivers.

3. Install the GRID drivers that are provided by Microsoft on the template VM by following the instructions for your operating system:

   - Windows NVIDIA GRID drivers
   - Linux NVIDIA GRID drivers

4. Restart the template VM.

5. Validate that the drivers are installed correctly by following the instructions in the Validate the installed drivers section.

6. After you've installed the drivers and other software that are required for your class, select **Publish** to create your students' VMs.

Validate the installed drivers

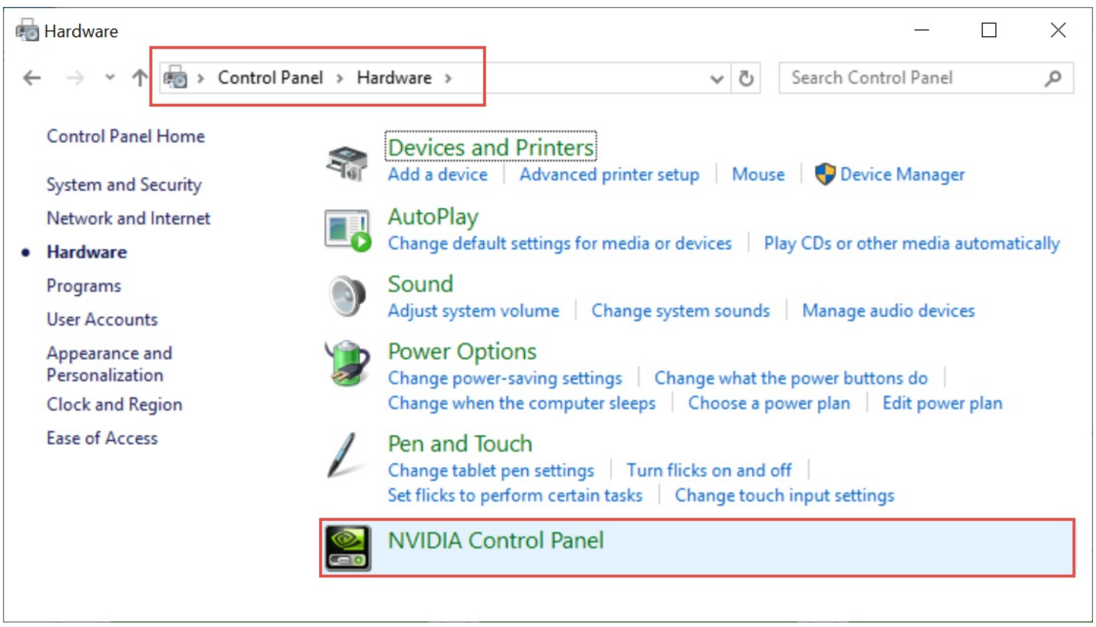
This section describes how to validate that your GPU drivers are properly installed.

Windows images

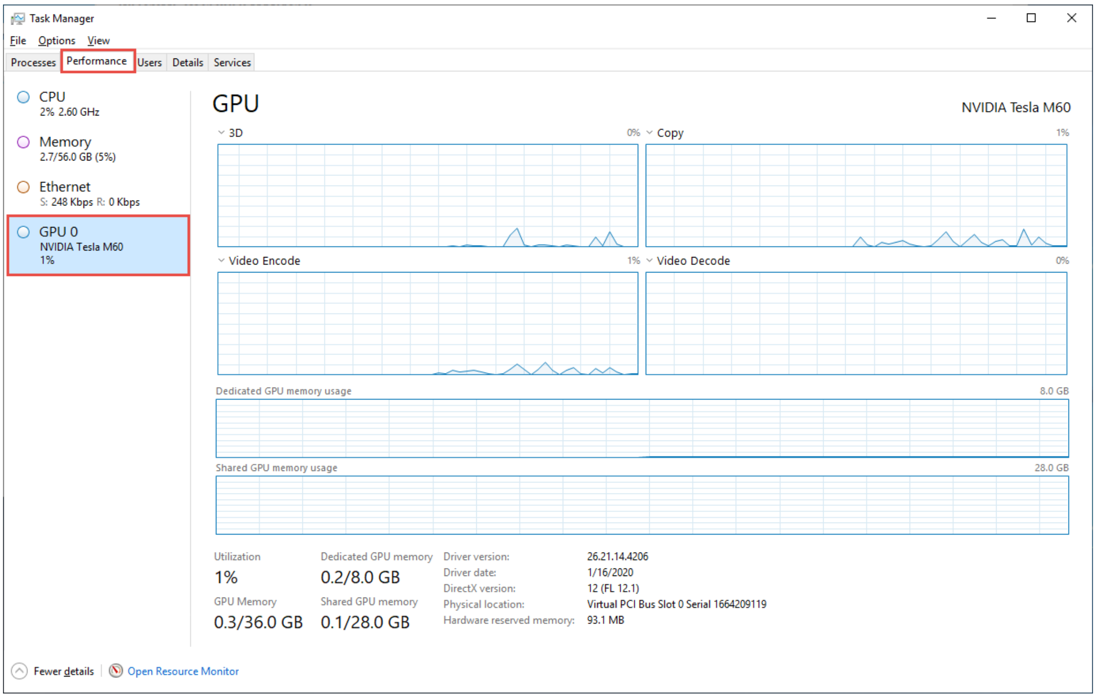
1. Follow the instructions in the "Verify driver installation" section of Install NVIDIA GPU drivers on N-series VMs running Windows.

2. If you're using a *visualization* GPU, you can also:

   - View and adjust your GPU settings in the NVIDIA Control Panel. To do so, in **Windows Control Panel**, select **Hardware**, and then select **NVIDIA Control Panel**.



- View your GPU performance by using **Task Manager**. To do so, select the **Performance** tab, and then select the **GPU** option.



> **IMPORTANT**
>
> The NVIDIA Control Panel settings can be accessed only for *visualization* GPUs. If you attempt to open the NVIDIA Control Panel for a compute GPU, you'll get the following error: "NVIDIA Display settings are not available. You are not currently using a display attached to an NVIDIA GPU." Similarly, the GPU performance information in Task Manager is provided only for visualization GPUs.

Depending on your scenario, you may also need to do additional validation to ensure the GPU is properly configured. Read the class type about Python and Jupyter Notebooks that explains an example where specific versions of drivers are needed.

Linux images

Follow the instructions in the "Verify driver installation" section of Install NVIDIA GPU drivers on N-series VMs running Linux.

Next steps

See the following articles:

- Create and manage labs
- SOLIDWORKS computer-aided design (CAD) class type
- MATLAB (matrix laboratory) class type

Guide to setting up a Windows template machine in Azure Lab Services

3/5/2021 • 7 minutes to read • Edit Online

If you're setting up a Windows 10 template machine for Azure Lab Services, here are some best practices and tips to consider. The configuration steps below are all optional. However, these preparatory steps could help make your students be more productive, minimize class time interruptions, and ensure that they're using the latest technologies.

> **IMPORTANT**
>
> This article contains PowerShell snippets to streamline the machine template modification process. For all the PowerShell scripts shown, you'll want to run them in Windows PowerShell with administrator privileges. In Windows 10, a quick way of doing that is to right-click the Start Menu and choose the "Windows PowerShell (Admin)".

Install and configure OneDrive

To protect student data from being lost if a virtual machine is reset, we recommend students back their data up to the cloud. Microsoft OneDrive can help students protect their data.

Install OneDrive

To manually download and install OneDrive, see the OneDrive or OneDrive for Business download pages.

You can also use the following PowerShell script. It will automatically download and install the latest version of OneDrive. Once the OneDrive client is installed, run the installer. In our example, we use the `/allUsers` switch to install OneDrive for all users on the machine. We also use the `/silent` switch to silently install OneDrive.

```
Write-Host "Downloading OneDrive Client..."
$DownloadPath = "$env:USERPROFILE/Downloads/OneDriveSetup.exe"
if((Test-Path $DownloadPath) -eq $False )
{
    Write-Host "Downloading OneDrive..."
    $web = new-object System.Net.WebClient
    $web.DownloadFile("https://go.microsoft.com/fwlink/p/?LinkId=248256",$DownloadPath)
} else {
    Write-Host "OneDrive installer already exists at " $DownloadPath
}

Write-Host "Installing OneDrive..."
& $env:USERPROFILE/Downloads/OneDriveSetup.exe /allUsers /silent
```

OneDrive customizations

There are many customizations that can be done to OneDrive. Let's cover some of the more common customizations.

Silently move Windows known folders to OneDrive

Folders like Documents, Downloads, and Pictures are often used to store student files. To ensure these folders are backed up into OneDrive, we recommend you move these folders to OneDrive.

If you are on a machine that is not using Active Directory, users can manually move those folders to OneDrive once they authenticate to OneDrive.

1. Open File Explorer
2. Right-click the Documents, Downloads, or Pictures folder.
3. Go to Properties > Location. Move the folder to a new folder on the OneDrive directory.

If your virtual machine is connected to Active Directory, you can set the template machine to automatically prompt your students to move the known folders to OneDrive.

You'll need to retrieve your organization ID first. For further instructions, see find your Microsoft 365 organization ID. You can also get the organization ID by using the following PowerShell.

```
Install-Module MSOnline -Confirm
Connect-MsolService
$officeTenantID = Get-MSOLCompanyInformation |
    Select-Object -expand objectID |
    Select-Object -expand Guid
```

Once you have your organization ID, set OneDrive to prompt to move known folders to OneDrive using the following PowerShell.

```
if ($officeTenantID -eq $null)
{
        Write-Error "Variable `$officeTenantId must be set to your Office Tenant Id before continuing."
}
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
New-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
    -Name "KFMSilentOptIn" -Value $officeTenantID -PropertyType STRING
```

Use OneDrive files on-demand

Students might have many files within their OneDrive accounts. To help save space on the machine and reduce download time, we recommend making all the files stored in student's OneDrive account be on-demand. On-demand files only download once a user accesses the file.

```
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive" -Force
New-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
    -Name "FilesOnDemandEnabled" -Value "00000001" -PropertyType DWORD
```

Silently sign in users to OneDrive

OneDrive can be set to automatically sign in with the Windows credentials of the logged on user. Automatic sign-in is useful for classes where the student signs in with their school credentials.

```
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
New-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
    -Name "SilentAccountConfig" -Value "00000001" -PropertyType DWORD
```

Disable the tutorial that appears at the end of OneDrive setup

This setting lets you prevent the tutorial from launching in a web browser at the end of OneDrive Setup.

```
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive" -Force
New-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
    -Name "DisableTutorial" -Value "00000001" -PropertyType DWORD -Force
```

Set the maximum size of a file that to be download automatically

This setting is used in conjunction with Silently sign in users to the OneDrive sync client with their Windows credentials on devices that don't have OneDrive Files On-Demand enabled. Any user who has a OneDrive that's larger than the specified threshold (in MB) will be prompted to choose the folders they want to sync before the OneDrive sync client (OneDrive.exe) downloads the files. In our example, "1111-2222-3333-4444" is the organization ID and 0005000 sets a threshold of 5 GB.

```
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive"
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive\DiskSpaceCheckThresholdMB"
New-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\OneDrive\DiskSpaceCheckThresholdMB"
    -Name "1111-2222-3333-4444" -Value "0005000" -PropertyType DWORD
```

Install and configure Microsoft 365

Install Microsoft 365

If your template machine needs Office, we recommend installation of Office through the Office Deployment Tool (ODT). You will need to create a reusable configuration file using the Microsoft 365 Apps Admin Center to choose which architecture, what features you'll need from Office, and how often it updates.

1. Go to Microsoft 365 Apps Admin Center and download your own configuration file.
2. Download Office Deployment Tool. Downloaded file will be `setup.exe`.
3. Run `setup.exe /download configuration.xml` to download Office components.
4. Run `setup.exe /configure configuration.xml` to install Office components.

Change the Microsoft 365 update channel

Using the Office Configuration Tool, you can set how often Office receives updates. However, if you need to modify how often Office receives updates after installation, you can change the update channel URL. Update channel URL addresses can be found at Change the Microsoft 365 Apps update channel for devices in your organization. The example below shows how to set Microsoft 365 to use the Monthly Update Channel.

```
# Update to the Microsoft 365 Monthly Channel
Set-ItemProperty
    -Path "HKLM:\SOFTWARE\Microsoft\Office\ClickToRun\Configuration\CDNBaseUrl"
    -Name "CDNBaseUrl"
    -Value "http://officecdn.microsoft.com/pr/492350f6-3a01-4f97-b9c0-c7c6ddf67d60"
```

Install and configure Updates

Install the latest Windows Updates

We recommend that you install the latest Microsoft updates on the template machine for security purposes before publishing the template VM. It also potentially avoids students from being disrupted in their work when updates run at unexpected times.

1. Launch **Settings** from the Start Menu
2. Click on **Update** & Security
3. Click **Check for updates**
4. Updates will download and install.

You can also use PowerShell to update the template machine.

```
Set-ExecutionPolicy Bypass -Scope Process -Force
Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Confirm
Install-Module PSWindowsUpdate -Confirm
Install-WindowsUpdate -MicrosoftUpdate
Set-ExecutionPolicy default -Force
```

> **NOTE**
>
> Some updates may require the machine to be restarted. You'll be prompted if a reboot is required.

Install the latest updates for Microsoft Store apps

We recommend having all Microsoft Store apps be updated to their latest versions. Here are instructions to manually update applications from the Microsoft Store.

1. Launch **Microsoft Store** application.
2. Click the ellipse (…) next to your user photo in the top corner of the application.
3. Select **Download** and updates from the drop-down menu.
4. Click **Get update** button.

You can also use PowerShell to update Microsoft Store applications that are already installed.

```
(Get-WmiObject -Namespace "root\cimv2\mdm\dmmap" -Class
"MDM_EnterpriseModernAppManagement_AppManagement01").UpdateScanMethod()
```

Stop automatic Windows Updates

After updating Windows to the latest version, you might consider stopping Windows Updates. Automatic updates could potentially interfere with scheduled class time. If your course is a longer running one, consider asking students to manually check for updates or setting automatic updates for a time outside of scheduled class hours. For more information about customization options for Windows Update, see the manage additional Windows Update settings.

Automatic Windows Updates may be stopped using the following PowerShell script.

```
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\AU"
New-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\AU"
    -Name "NoAutoUpdate" -Value "1" -PropertyType DWORD
```

Install foreign language packs

If you need additional languages installed on the virtual machine, you can add them through the Microsoft Store.

1. Launch Microsoft Store
2. Search for "language pack"
3. Choose language to install

If you are already logged on to the template VM, use "Install language pack" shortcut (`ms-settings:regionlanguage?activationSource=SMC-IA-4027670`) to go directly to the appropriate settings page.

Remove unneeded built-in apps

Windows 10 comes with many built-in applications that might not be needed for your particular class. To simplify the machine image for students, you might want to uninstall some applications from your template machine. To see a list of installed applications, use the PowerShell `Get-AppxPackage` cmdlet. The example below shows all installed applications that can be removed.

```
Get-AppxPackage | Where {$_.NonRemovable -eq $false} | select Name
```

To remove an application, use the Remove-Appx cmdlet. The example below shows how to remove everything XBox related.

```
Get-AppxPackage -Name *xbox* | foreach { if (-not $_.NonRemovable) { Remove-AppxPackage $_} }
```

Install common teaching-related applications

Install other apps commonly used for teaching through the Windows Store app. Suggestions include applications like Microsoft Whiteboard app, Microsoft Teams, and Minecraft Education Edition. These applications must be installed manually through the Windows Store or through their respective websites on the template VM.

Conclusion

This article has shown you optional steps to prepare your Windows template VM for an effective class. Steps include installing OneDrive and installing Microsoft 365, installing the updates for Windows and installing updates for Microsoft Store apps. We also discussed how to set updates to a schedule that works best for your class.

Next steps

See the article on how to control Windows shutdown behavior to help with managing costs: Guide to controlling Windows shutdown behavior

Use external file storage in Lab Services

9/7/2021 • 11 minutes to read • Edit Online

This article covers some of the options for external file storage when you use Azure Lab Services. Azure Files offers fully managed file shares in the cloud, accessible via SMB 2.1 and SMB 3.0. An Azure Files share can be connected either publicly or privately within a virtual network. You can also configure the share to use a student's Active Directory credentials for connecting to the file share. If you're on a Linux machine, you can also use Azure NetApp Files with NFS volumes for external file storage with Azure Lab Services.

Which solution to use

Each solution has different requirements and abilities. The following table lists important points to consider for each solution.

SOLUTION	IMPORTANT TO KNOW
Azure Files share with public endpoint	• Everyone has read/write access. • No virtual network peering is required. • Accessible to all VMs, not just lab VMs. • If you're using Linux, students will have access to the storage account key.
Azure Files share with private endpoint	• Everyone has read/write access. • Virtual network peering is required. • Accessible only to VMs on the same network (or a peered network) as the storage account. • If you're using Linux, students will have access to the storage account key.
Azure Files with identity-based authorization	• Either read or read/write access permissions can be set for folder or file. • Virtual network peering is required. • Storage account must be connected to Active Directory. • Lab VMs must be domain-joined. • Storage account key isn't used for students to connect to the file share.
NetApp Files with NFS volumes	• Either read or read/write access can be set for volumes. • Permissions are set by using a student VM's IP address. • Virtual network peering is required. • You might need to register to use the NetApp Files service. • Linux only.

The cost of using external storage isn't included in the cost of using Azure Lab Services. For more information about pricing, see Azure Files pricing and Azure NetApp Files pricing.

Azure Files share

Azure Files shares are accessed by using a public or private endpoint. Mount the shares by using the storage account key as the password. With this approach, everyone has read-write access to the file share.

If you're using a public endpoint to the Azure Files share, it's important to remember the following:

- The virtual network for the storage account doesn't have to be peered to the lab account. The file share can be created anytime before the template VM is published.
- The file share can be accessed from any machine if a user has the storage account key.
- Linux students can see the storage account key. Credentials for mounting an Azure Files share are stored in `{file-share-name}.cred` on Linux VMs, and are readable by sudo. Because students are given sudo access by default in Azure Lab Services VMs, they can read the storage account key. If the storage account endpoint is public, students can get access to the file share outside of their student VM. Consider rotating the storage account key after class has ended, and using private file shares.

If you're using a private endpoint to the Azure Files share, it's important to remember the following:

- Access is restricted to traffic originating from the private network, and can't be accessed through the public internet. Only VMs in the private virtual network, VMs in a network peered to the private virtual network, or machines connected to a VPN for the private network, can access the file share.
- Linux students can see the storage account key. Credentials for mounting an Azure Files share are stored in `{file-share-name}.cred` on Linux VMs, and are readable by sudo. Because students are given sudo access by default in Azure Lab Services VMs, they can read the storage account key. Consider rotating the storage account key after class has ended.
- This approach requires the file share virtual network to be peered to the lab account. The virtual network for the Azure Storage account must be peered to the virtual network for the lab account before the lab is created.

> **NOTE**
>
> By default, standard file shares can span up to 5 TiB. See Create an Azure file share for information on how to create file shares than span up to 100 TiB.

Follow these steps to create a VM connected to an Azure file share.

1. Create an Azure Storage account. On the **Connectivity method** page, choose **public endpoint** or **private endpoint**.
2. If you've chosen the private method, create a private endpoint in order for the file shares to be accessible from the virtual network.
3. Create an Azure file share. The file share is reachable by the public host name of the storage account if using a public endpoint. The file share is reachable by private IP address if using a private endpoint.
4. Mount the Azure file share in the template VM:
   - Windows
   - Linux. To avoid mounting issues on student VMs, see the use Azure Files with Linux section.
5. Publish the template VM.

> **IMPORTANT**
>
> Make sure Windows Defender Firewall isn't blocking the outgoing SMB connection through port 445. By default, SMB is allowed for Azure VMs.

Use Azure Files with Linux

If you use the default instructions to mount an Azure Files share, the file share will seem to disappear on student VMs after the template is published. The following modified script addresses this issue.

For file share with a public endpoint:

```
#!/bin/bash

# Assign variables values for your storage account and file share
storage_account_name=""
storage_account_key=""
fileshare_name=""

# Do not use 'mnt' for mount directory.
# Using ‘mnt’ will cause issues on student VMs.
mount_directory="prm-mnt"

sudo mkdir /$mount_directory/$fileshare_name
if [ ! -d "/etc/smbcredentials" ]; then
    sudo mkdir /etc/smbcredentials
fi
if [ ! -f "/etc/smbcredentials/$storage_account_name.cred" ]; then
    sudo bash -c "echo ""username=$storage_account_name"" >> /etc/smbcredentials/$storage_account_name.cred"
    sudo bash -c "echo ""password=$storage_account_key"" >> /etc/smbcredentials/$storage_account_name.cred"
fi
sudo chmod 600 /etc/smbcredentials/$storage_account_name.cred

sudo bash -c "echo ""//$storage_account_name.file.core.windows.net/$fileshare_name
/$mount_directory/$fileshare_name cifs
nofail,vers=3.0,credentials=/etc/smbcredentials/$storage_account_name.cred,dir_mode=0777,file_mode=0777,serv
erino"" >> /etc/fstab"
sudo mount -t cifs //$storage_account_name.file.core.windows.net/$fileshare_name
/$mount_directory/$fileshare_name -o
vers=3.0,credentials=/etc/smbcredentials/$storage_account_name.cred,dir_mode=0777,file_mode=0777,serverino
```

For file share with a private endpoint:

```
#!/bin/bash

# Assign variables values for your storage account and file share
storage_account_name=""
storage_account_ip=""
storage_account_key=""
fileshare_name=""

# Do not use 'mnt' for mount directory.
# Using ‘mnt’ will cause issues on student VMs.
mount_directory="prm-mnt"

sudo mkdir /$mount_directory/$fileshare_name
if [ ! -d "/etc/smbcredentials" ]; then
    sudo mkdir /etc/smbcredentials
fi
if [ ! -f "/etc/smbcredentials/$storage_account_name.cred" ]; then
    sudo bash -c "echo ""username=$storage_account_name"" >> /etc/smbcredentials/$storage_account_name.cred"
    sudo bash -c "echo ""password=$storage_account_key"" >> /etc/smbcredentials/$storage_account_name.cred"
fi
sudo chmod 600 /etc/smbcredentials/$storage_account_name.cred

sudo bash -c "echo ""//$storage_account_ip/$fileshare_name /$mount_directory/$fileshare_name cifs
nofail,vers=3.0,credentials=/etc/smbcredentials/$storage_account_name.cred,dir_mode=0777,file_mode=0777,serv
erino"" >> /etc/fstab"
sudo mount -t cifs //$storage_account_name.file.core.windows.net/$fileshare_name
/$mount_directory/$fileshare_name -o
vers=3.0,credentials=/etc/smbcredentials/$storage_account_name.cred,dir_mode=0777,file_mode=0777,serverino
```

If the template VM that mounts the Azure Files share to the `/mnt` directory is already published, the student can either:

- Move the instruction to mount `/mnt` to the top of the `/etc/fstab` file.
- Modify the instruction to mount `/mnt/{file-share-name}` to a different directory, like `/prm-mnt/{file-share-name}`.

Students should run `mount -a` to remount directories.

For more general information, see Use Azure Files with Linux.

Azure Files with identity-based authorization

Azure Files shares can also be accessed by using Active Directory authentication, if the following are both true:

- The student's VM is domain-joined.
- Active Directory authentication is enabled on the Azure Storage account that hosts the file share.

The network drive is mounted on the virtual machine by using the user's identity, not the key to the storage account. Public or private endpoints provide access to the storage account.

Keep in mind the following important points:

- You can set permissions on a directory or file level.
- You can use current user credentials to authenticate to the file share.

For a public endpoint, the virtual network for the storage account doesn't have to peer to the lab account. You can create the file share anytime before the template VM is published.

For a private endpoint:

- Access is restricted to traffic originating from the private network, and can't be accessed through the public internet. Only VMs in the private virtual network, VMs in a network peered to the private virtual network, or machines connected to a VPN for the private network, can access the file share.
- This approach requires the file share virtual network to be peered to the lab account. The virtual network for the Azure Storage account must be peered to the virtual network for the lab account before the lab is created.

To create an Azure Files share that's enabled for Active Directory authentication, and to domain-join the lab VMs, follow these steps:

1. Create an Azure Storage account.
2. If you've chosen the private method, create a private endpoint in order for the file shares to be accessible from the virtual network. Create a private DNS zone, or use an existing one. Private Azure DNS zones provide name resolution within a virtual network.
3. Create an Azure file share.
4. Follow the steps to enable identity-based authorization. If you're using Active Directory on-premises, and you're synchronizing it with Azure Active Directory (Azure AD), see On-premises Active Directory Domain Services authentication over SMB for Azure file shares. If you're using only Azure AD, see Enable Azure Active Directory Domain Services authentication on Azure Files.

   > **IMPORTANT**
   >
   > Talk to the team that manages your Active Directory instance to verify that all prerequisites listed in the instructions are met.

5. Assign SMB share permission roles in Azure. For details about permissions that are granted to each role, see

share-level permissions.

- **Storage File Data SMB Share Elevated Contributor** role must be assigned to the person or group that will set up permissions for contents of the file share.
- **Storage File Data SMB Share Contributor** role should be assigned to students who need to add or edit files on the file share.
- **Storage File Data SMB Share Reader** role should be assigned to students who only need to read the files from the file share.

6. Set up directory-level and/or file-level permissions for the file share. You must set up permissions from a domain-joined machine that has network access to the file share. To modify directory-level and/or file-level permissions, mount the file share by using the storage key, not your Azure AD credentials. To assign permissions, use the Set-Acl PowerShell command, or icacls in Windows.
7. Peer the virtual network for the storage account to the lab account.
8. Create the classroom lab.
9. Save a script on the template VM that students can run to connect to the network drive. To get example script:
   a. Open the storage account in the Azure portal.
   b. Under **File Service**, select **File Shares**.
   c. Find the share that you want to connect to, select the ellipses button on the far right, and choose **Connect**.
   d. You'll see instructions for Windows, Linux, and macOS. If you're using Windows, set **Authentication method** to **Active Directory**.
   e. Copy the code in the example, and save it on the template machine in a `.ps1` file for Windows, or an `.sh` file for Linux.
10. On the template machine, download and run the script to join student machines to the domain. The `Join-AzLabADTemplate` script publishes the template VM automatically.

    > **NOTE**
    >
    > The template machine isn't domain-joined. To view files on the share, instructors need to use a student VM for themselves.

11. Students using Windows can connect to the Azure Files share by using File Explorer with their credentials, after they've been given the path to the file share. Alternately, students can run the preceding script to connect to the network drive. For students who are using Linux, run the preceding script.

NetApp Files with NFS volumes

Azure NetApp Files is an enterprise-class, high-performance, metered file storage service.

- Access policies can be set on a per-volume basis.
- Permission policies are IP-based for each volume.
- If students need their own volume that other students don't have access to, permission policies must be assigned after the lab is published.
- In the context of Azure Lab Services, only Linux machines are supported.
- The virtual network for the Azure NetApp Files capacity pool must be peered to the virtual network for the lab account **before** the lab is created.

To use an Azure NetApp Files share in Azure Lab Services:

1. To create a NetApp Files capacity pool and one or more NFS volumes, see set up Azure NetApp Files and NFS volume. For information about service levels, see Service levels for Azure NetApp Files.
2. Peer the virtual network for the NetApp Files capacity pool to the lab account.

3. Create the classroom lab.

4. On the template VM, install the components necessary to use NFS file shares.

   - Ubuntu:

     ```
     sudo apt update
     sudo apt install nfs-common
     ```

   - CentOS:

     ```
     sudo yum install nfs-utils
     ```

5. On the template VM, save the following script as `mount_fileshare.sh` to mount the NetApp Files share. Assign the `capacity_pool_ipaddress` variable the mount target IP address for the capacity pool. Get the mount instructions for the volume to find the appropriate value. The script expects the path name of the NetApp Files volume. To ensure that users can run the script, run `chmod u+x mount_fileshare.sh`.

   ```
   #!/bin/bash

   if [ $# -eq 0 ];  then
       echo "Must provide volume name."
       exit 1
   fi

   volume_name=$1
   capacity_pool_ipaddress=0.0.0.0 # IP address of capacity pool

   # Do not use 'mnt' for mount directory.
   # Using ‘mnt’ might cause issues on student VMs.
   mount_directory="prm-mnt"

   sudo mkdir -p /$mount_directory
   sudo mkdir /$mount_directory/$folder_name

   sudo mount -t nfs -o rw,hard,rsize=65536,wsize=65536,vers=3,tcp
   $capacity_pool_ipaddress:/$volume_name /$mount_directory/$volume_name
   sudo bash -c "echo ""$capacity_pool_ipaddress:/$volume_name /$mount_directory/$volume_name nfs
   bg,rw,hard,noatime,nolock,rsize=65536,wsize=65536,vers=3,tcp,_netdev 0 0"" >> /etc/fstab"
   ```

6. If all students are sharing access to the same NetApp Files volume, you can run the `mount_fileshare.sh` script on the template machine before publishing. If students each get their own volume, save the script to be run later by the student.

7. Publish the template VM.

8. Configure the policy for the file share. The export policy can allow for a single VM or multiple VMs to have access to a volume. You can grant read-only or read/write access.

9. Students must start their VM and run the script to mount the file share. They'll only have to run the script once. The command will look like the following: `./mount_fileshare.sh myvolumename`.

Next steps

These steps are common to setting up any lab.

- Create and manage a template
- Add users

- Set quota
- Set a schedule
- Email registration links to students

Guide to controlling Windows shutdown behavior

3/5/2021 • 2 minutes to read • Edit Online

Azure Lab Services provides several cost controls to ensure that Windows virtual machines (VMs) are not running unexpectedly:

- Set a schedule
- Set quotas for users
- Enable automatic shutdown on disconnect

Even with these cost controls, there are situations where a Windows VM may unexpectedly continue to run; and as a result, deduct from the student's quota:

- **RDP window is left open**

  When a student connects to their VM using RDP, they may inadvertently leave the RDP window open. As long as the RDP window remains open, the **automatic shutdown on disconnect** setting will never take effect since it is only triggered after the RDP session is disconnected.

- **Windows shutdown command is used to turn off the VM**

  A student may use Windows shutdown command, or other shutdown mechanisms provided within Windows, to turn off the VM instead of using Azure Lab Services' stop button. When this happens, from the perspective of Azure Lab Services, the VM is still being used.

To help you prevent these situations from happening, this guide provides steps to automatically shutdown an idle Windows VM and remove the Windows shutdown command from the **Start** menu.

> **NOTE**
>
> A VM may also unexpectedly deduct from the quota when the student starts their VM, but never actually connects to it using RDP. This guide does *not* currently address this scenario. Instead, students should be reminded to immediately connect to their VM using RDP after they start it; or, they should stop the VM.

Remove Windows shutdown command from Start menu

Windows **Local Group Policy** settings also allow you to remove the shutdown command from the **Start** menu.

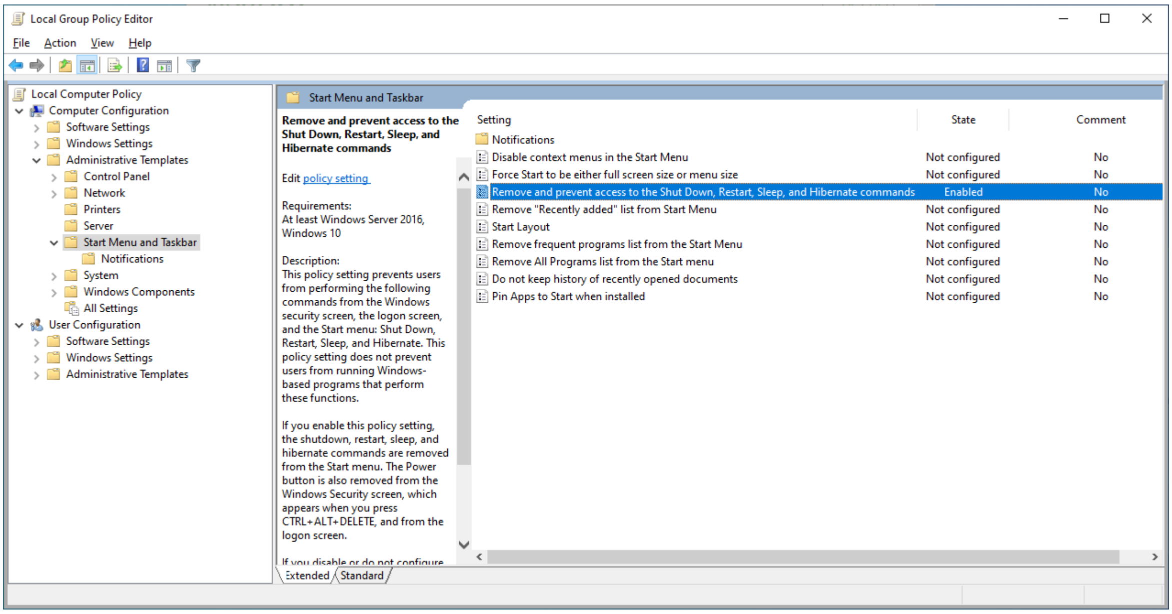
To remove the shutdown command, you can connect to the template VM and execute the below PowerShell script.

```
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name
"HidePowerOptions" -Value 1 -Force
```

Or, you can choose to follow these manual steps using the template VM:

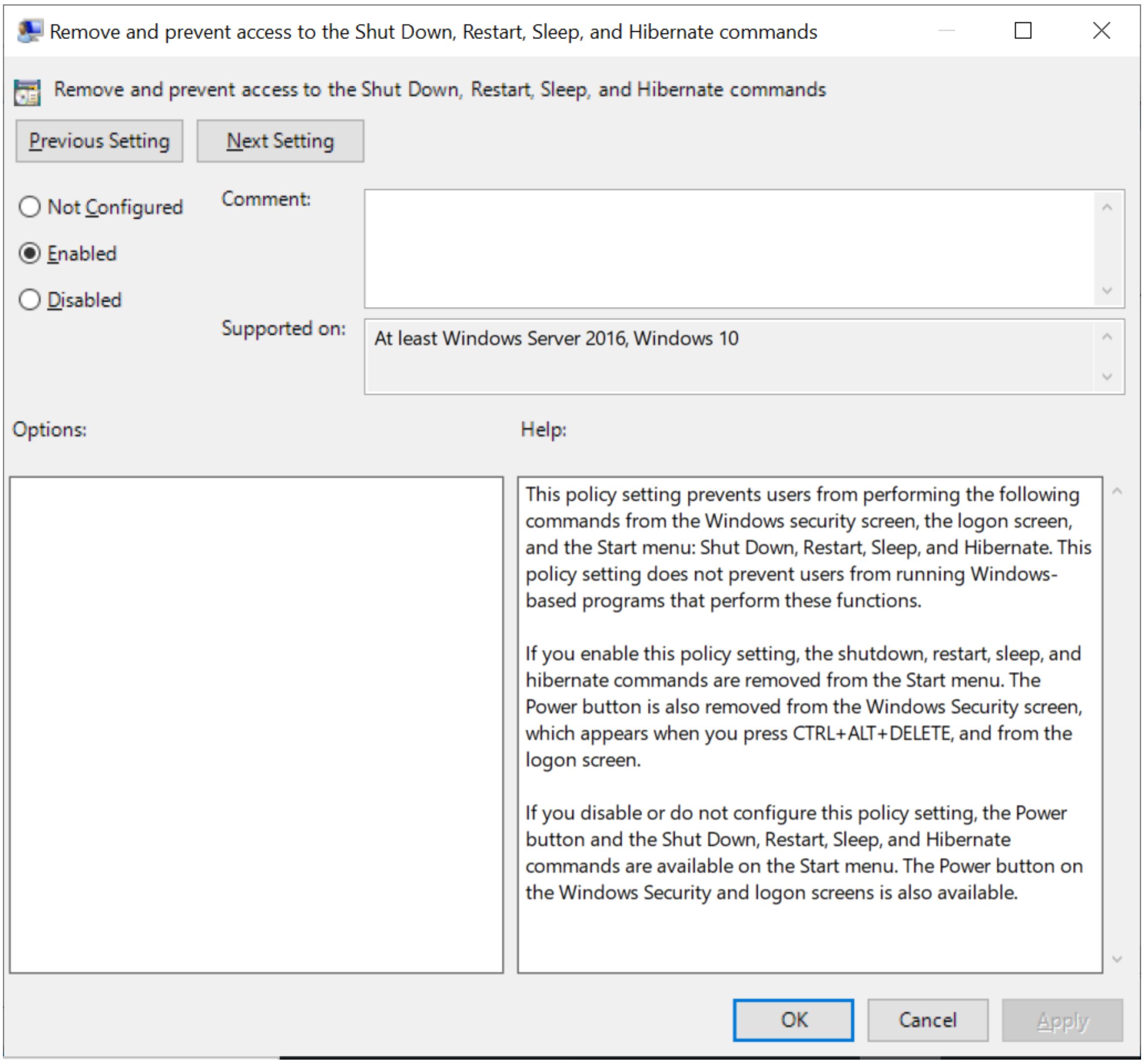
1. Press the Windows key, type **gpedit**, then select **Edit group policy (Control panel)**.

2. Go to **Computer Configuration** > **Administrative Templates** > **Start Menu and Taskbar**.

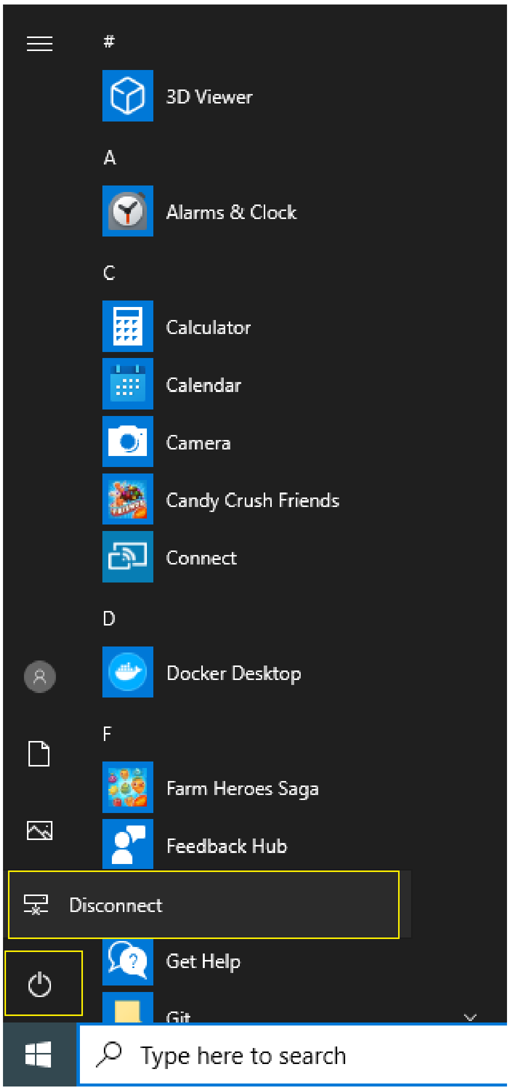


3. Right-click **Remove and prevent access to the Shut Down, Restart, Sleep, and Hibernate commands**, and click **Edit**.

4. Select the **Enabled** setting and then click **OK**:



5. Notice that the shutdown command no longer appears under Windows **Start** menu; only the **Disconnect** command appears.



Next steps

See the article on how to prepare a Windows template VM: [Guide to setting up a Windows template machine in Azure Lab Services]

View support information (lab creator in Azure Lab Services)

11/2/2020 • 2 minutes to read • Edit Online

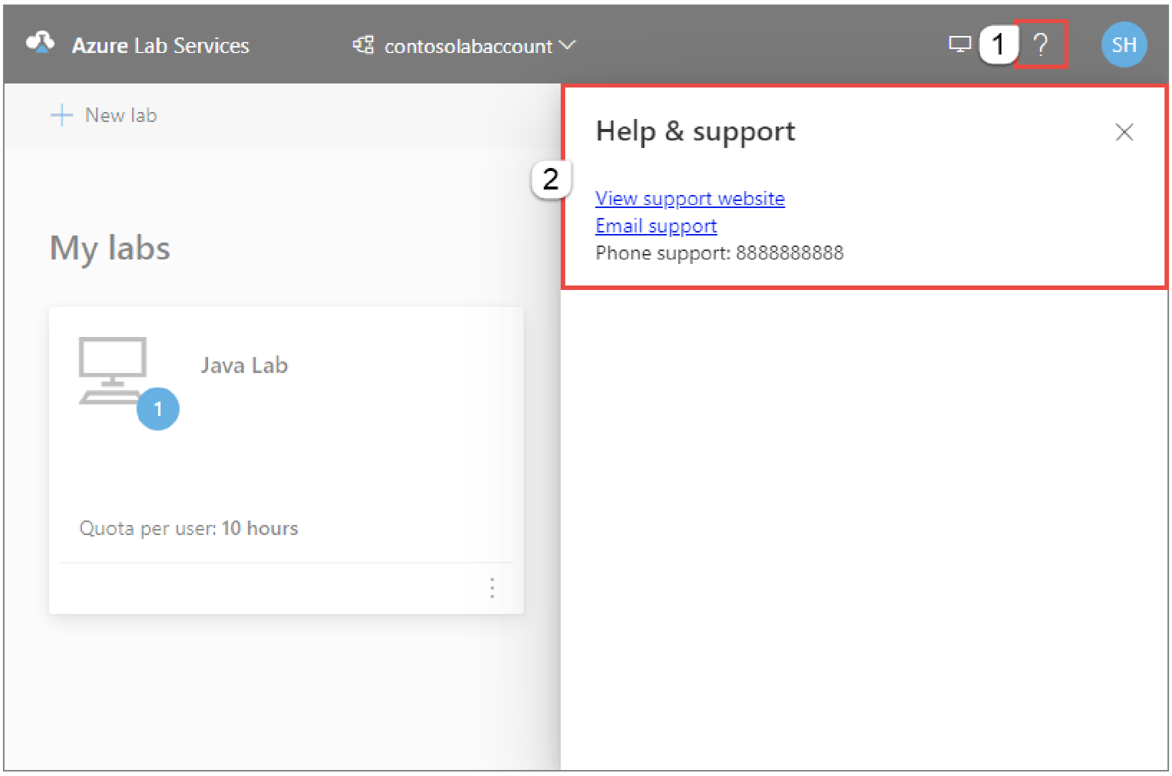
This article explains how you (as a lab creator) can view the following support information:

- URL
- Email
- Phone
- Additional instructions

You can use this information to get help when you run into any technical issues while creating a lab in a lab account.

View support information

1. Sign in to Azure Lab Services portal.
2. Select question mark (?) at the top-right corner of the page.
3. Confirm that you see links to the **view support website**, **email support**, and **support phone number**.



Next steps

See the following article to learn about how a lab user views the support contact information:

- View contact information (lab user)
- Specify contact information (lab account owner)

Access virtual machines as a student from the educator view

11/2/2020 • 2 minutes to read • Edit Online

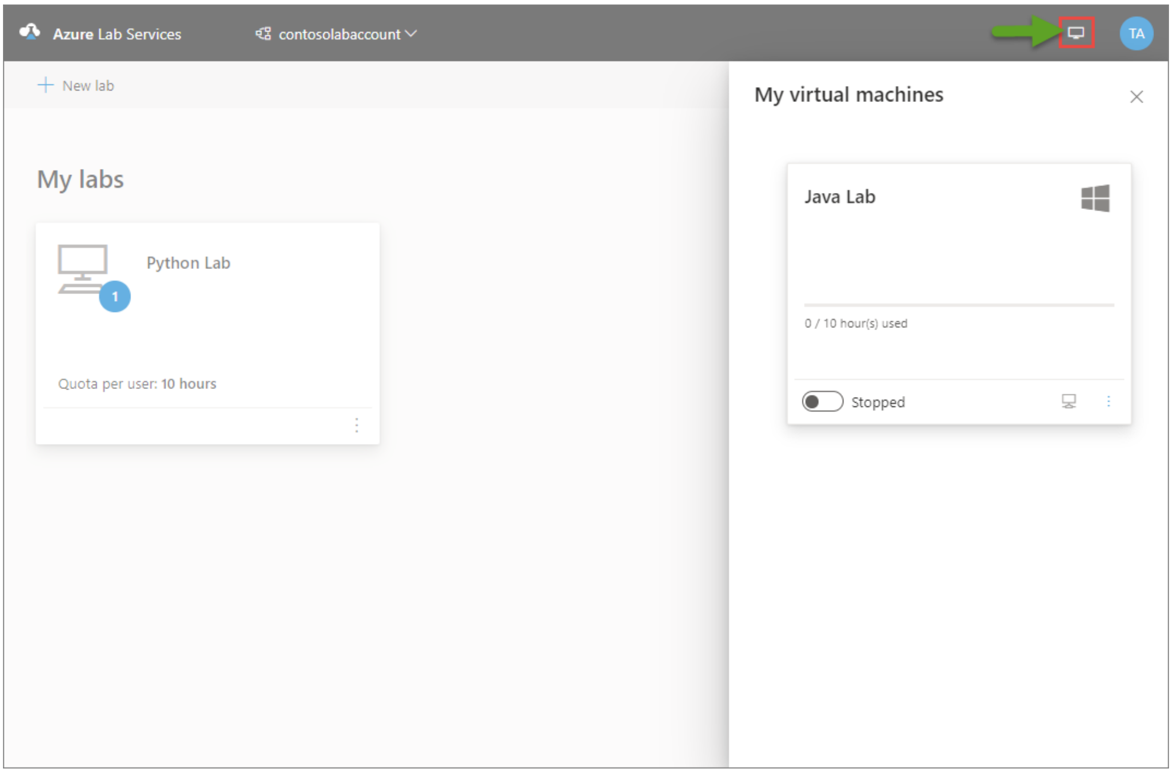
This article shows how educators can access their VMs for classes that they attend as students.

Here is a scenario where this feature will help. A teaching assistant is an educator for one class but a student in other classes. And, the teaching assistant wants to view and access the student VMs from the educator view that shows the labs they own.

Access VMs from educator view

1. Sign in to the Azure Lab Services website. You see the labs that you own. These labs may be labs you created yourself or the labs that admin assigned to you as owner. For more information, see How to add additional owners to an existing lab

2. To access VMs for classes that you attend as a student, select the computer icon in the top-right corner. Confirm that you see VMs you can access as a student. In the following example, the user is a teaching assistant for the Python lab, but a student of the Java lab. So, the user sees the VM from the Java lab in the drop-down list. The user can start the VM and connect to it.



Next steps

See the following articles:

- Connect to a VM
- Connect to a VM using RDP on a Mac
- Connect to a VM using RDP on a Chromebook
- Use remote desktop for Linux virtual machines

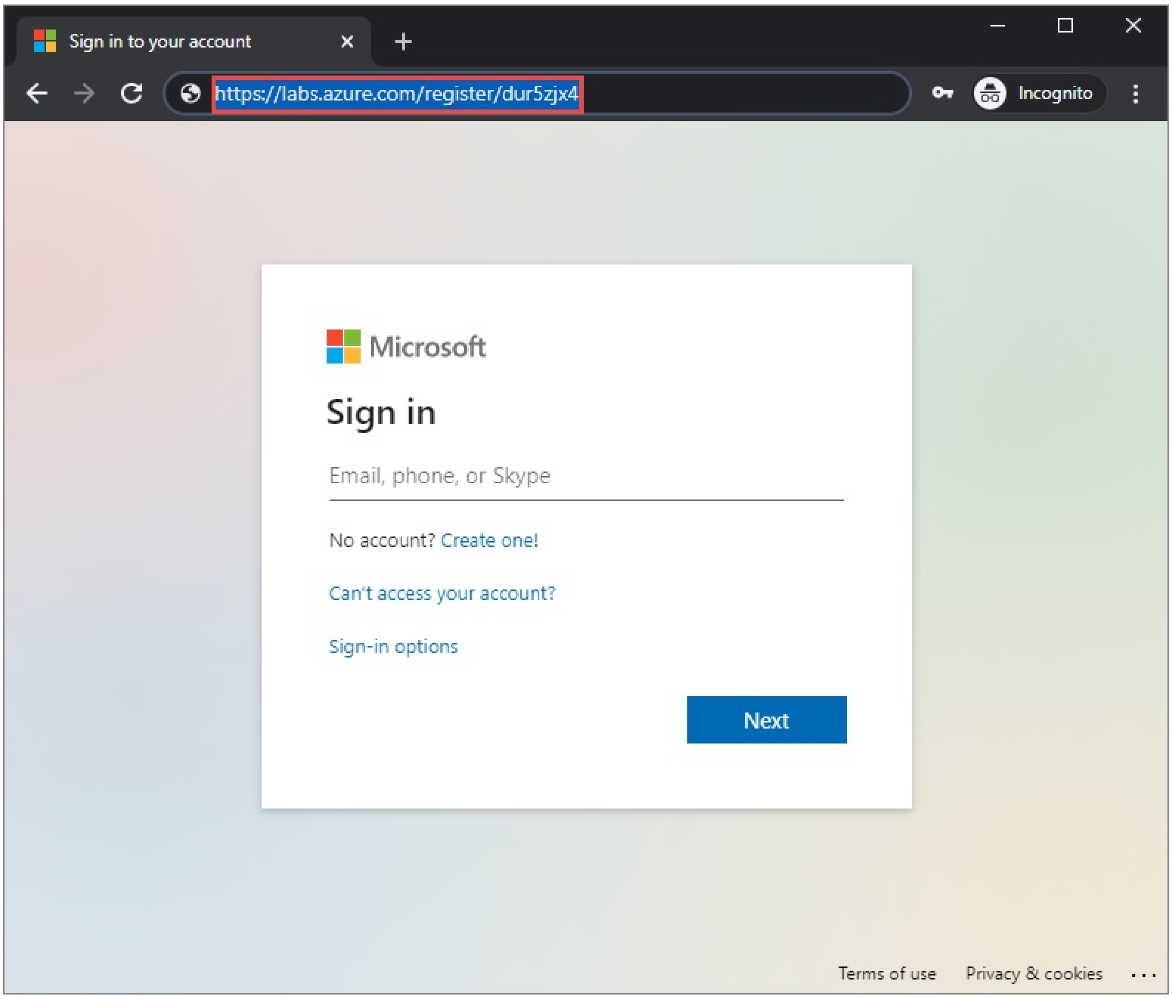
How to access a classroom lab in Azure Lab Services

3/5/2021 • 3 minutes to read • Edit Online

This article describes how to register to a classroom lab, view all the labs that you can access, start/stop a VM in the lab, and connect to the VM.

Register to the lab

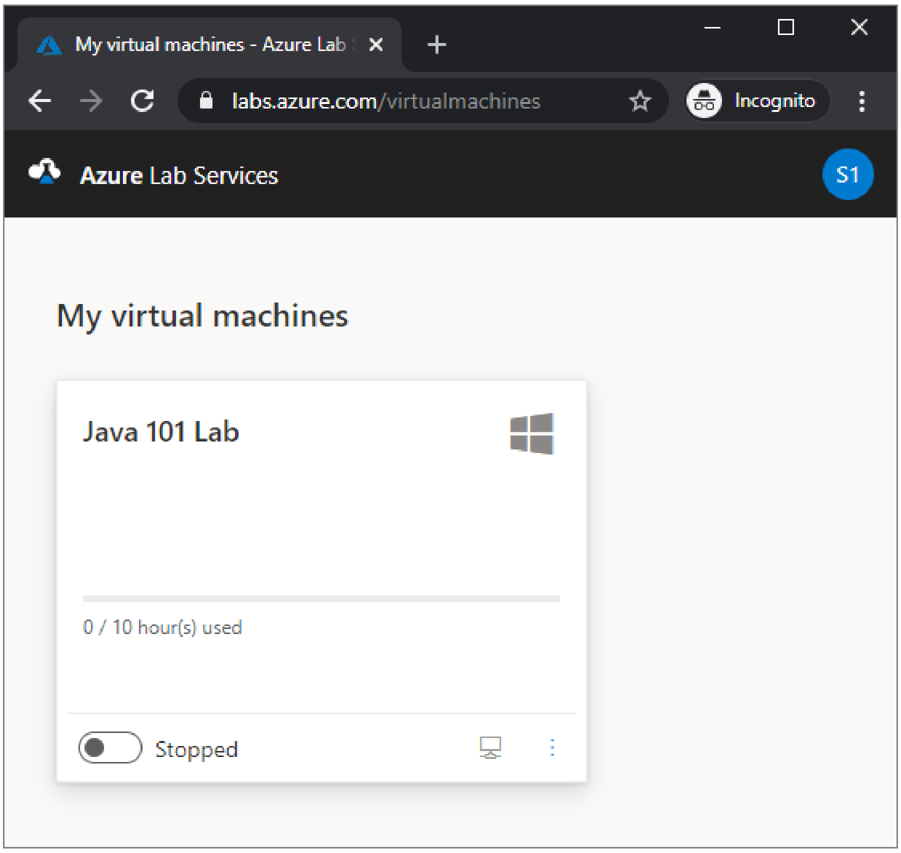
1. Navigate to the **registration URL** that you received from the educator. You don't need to use the registration URL after you complete the registration. Instead, use the URL: https://labs.azure.com. Internet Explorer 11 isn't supported yet.



2. Sign in to the service using your school account to complete the registration.

> **NOTE**
> A Microsoft account is required for using Azure Lab Services. If you are trying to use your non-Microsoft account such as Yahoo or Google accounts to sign in to the portal, follow instructions to create a Microsoft account that will be linked to your non-Microsoft account. Then, follow the steps to complete the registration process.

3. Once registered, confirm that you see the virtual machine for the lab you have access to.



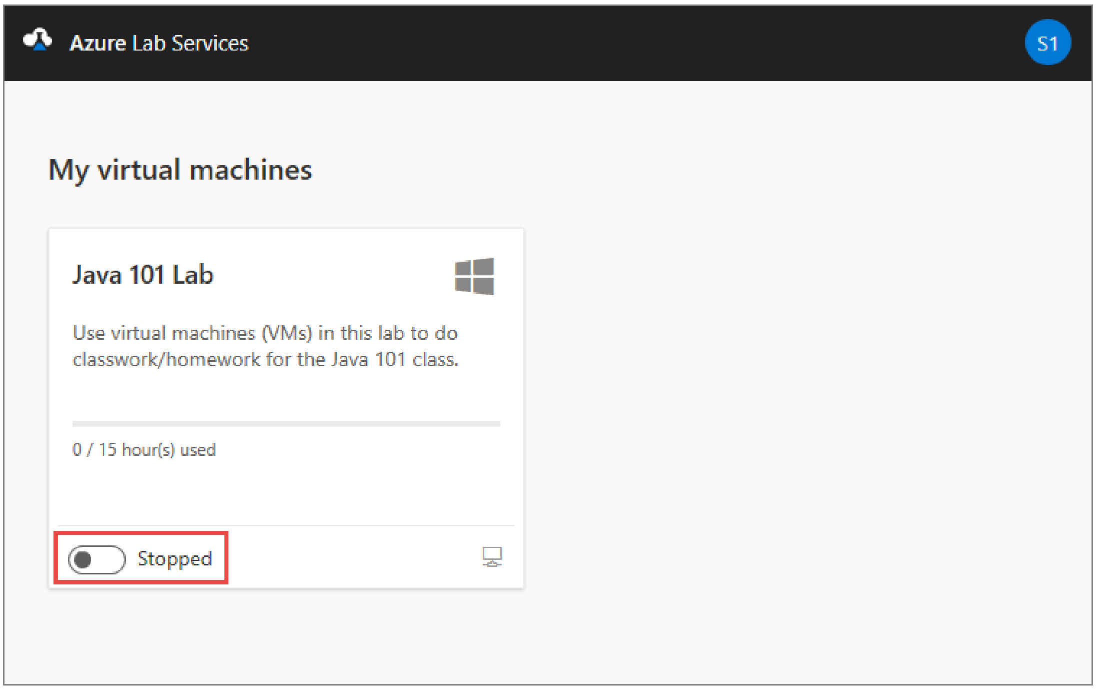
4. Wait until the virtual machine is ready. On the VM tile, notice the following fields:

   a. At the top of the tile, you see the **name of the lab**.

   b. To its right, you see the icon representing the **operating system (OS)** of the VM. In this example, it's Windows OS.

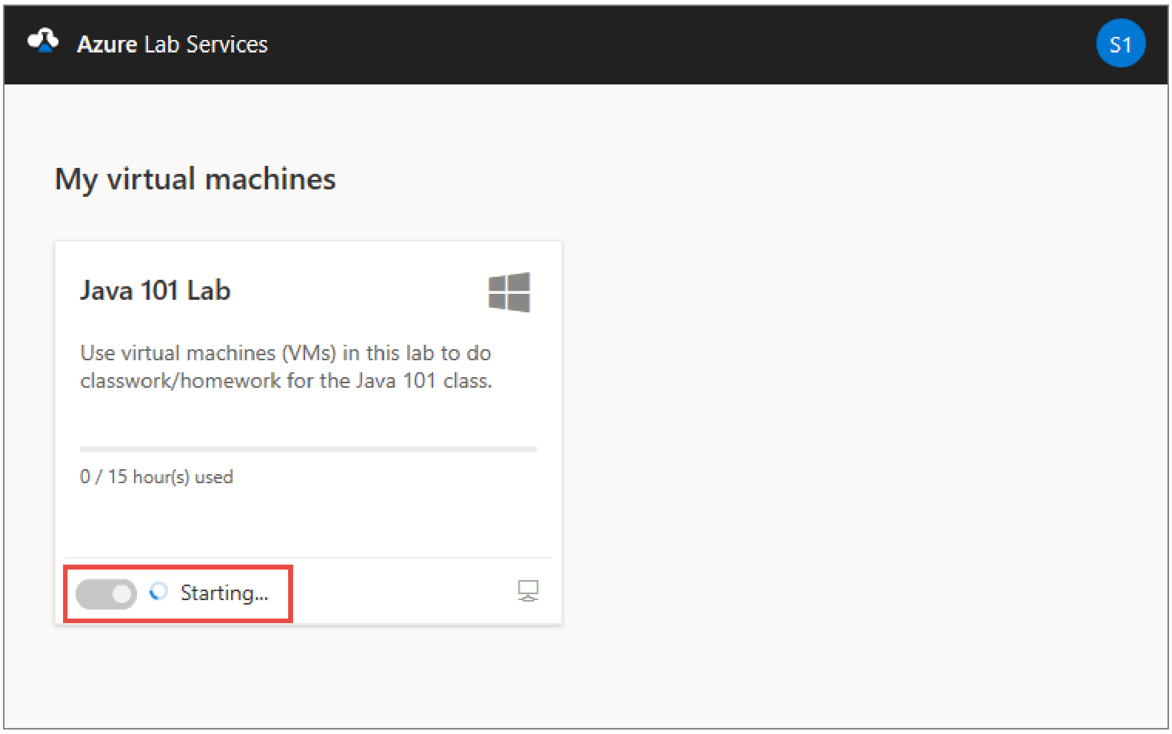
   c. You see icons/buttons at the bottom of the tile to start/stop the VM, and connect to the VM.

   d. To the right of the buttons, you see the status of the VM. Confirm that you see the status of the VM is **Stopped**.

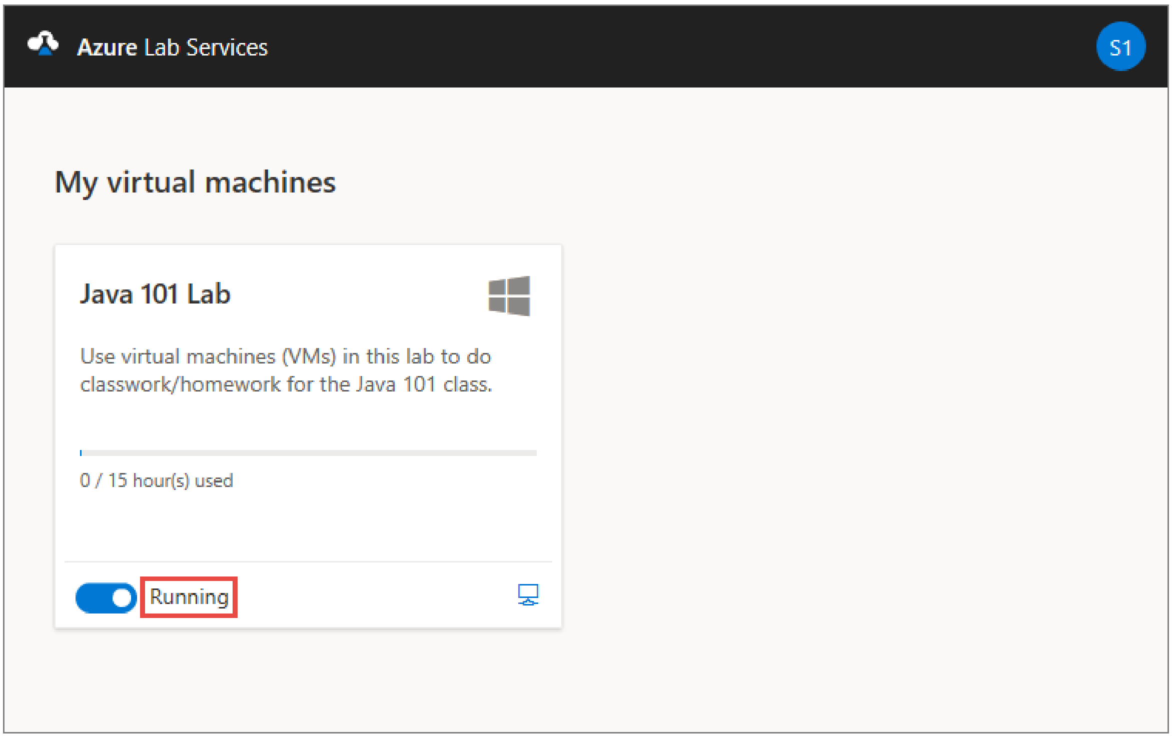


Start or stop the VM

1. **Start** the VM by selecting the first button as shown in the following image. This process takes some time.



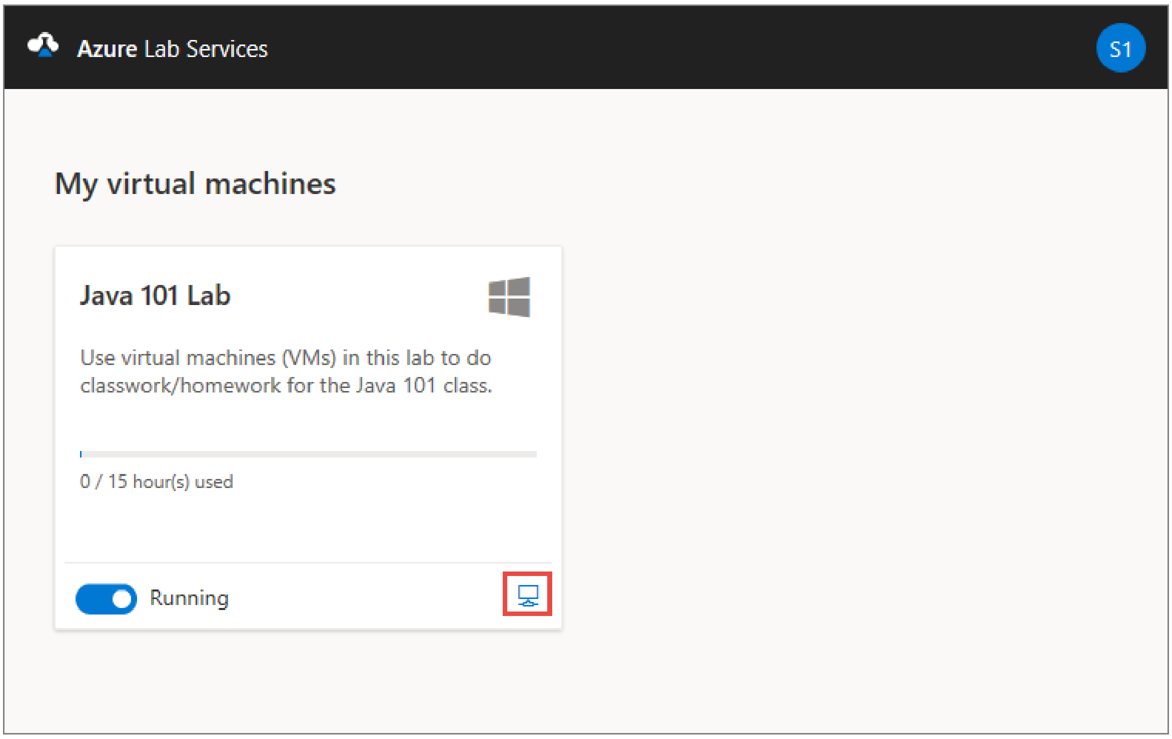
2. Confirm that the status of the VM is set to **Running**.



Notice that the icon of the first button changed to represent a **stop** operation. You can select this button to stop the VM.

Connect to the VM

1. Select the second button as shown in the following image to **connect** to the lab's VM.



2. Do one of the following steps:

   a. For **Windows** virtual machines, save the **RDP** file to the hard disk. Open the RDP file to connect to the virtual machine. Use the **user name** and **password** you get from your educator to sign in to the machine.

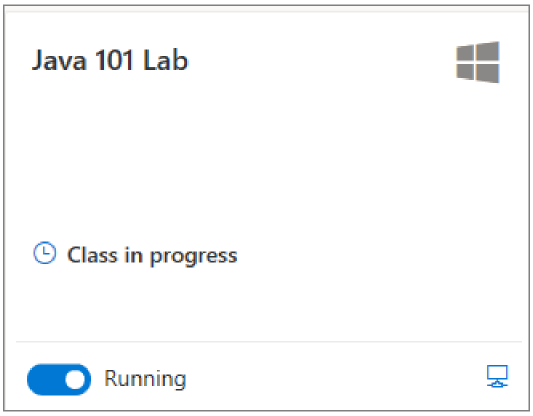
   b. For **Linux** virtual machines, you can use **SSH** or **RDP** (if it's enabled) to connect to them. For more information, see Enable remote desktop connection for Linux machines.

   c. If you are using a **Mac** to connect to the lab VM, follow instructions in the next section.

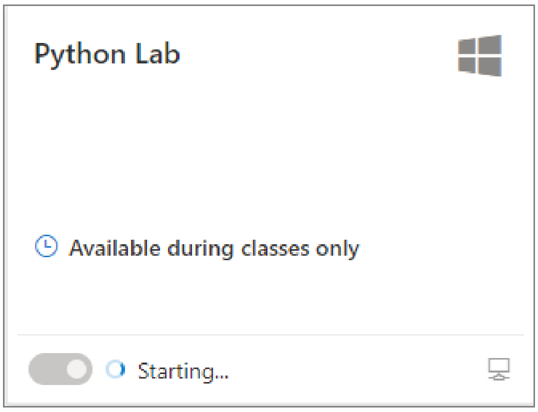
Progress bar

The progress bar on the tile shows the number of hours used against the number of quota hours assigned to you. This time is the additional time allotted to you in addition to the scheduled time for the lab. The color of the progress bar and the text under the progress bar varies as per the following scenarios:

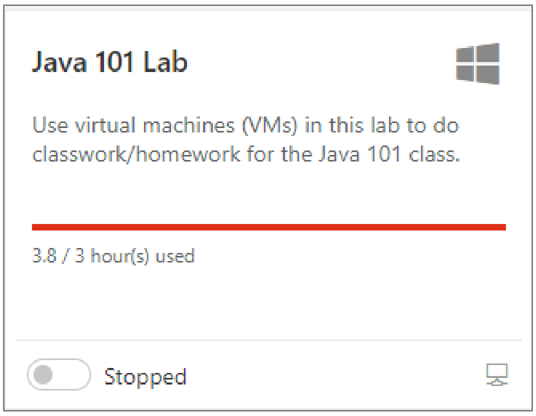
- If a class is in progress (within the schedule of the class), progress bar is grayed out to represent quota hours aren't being used.

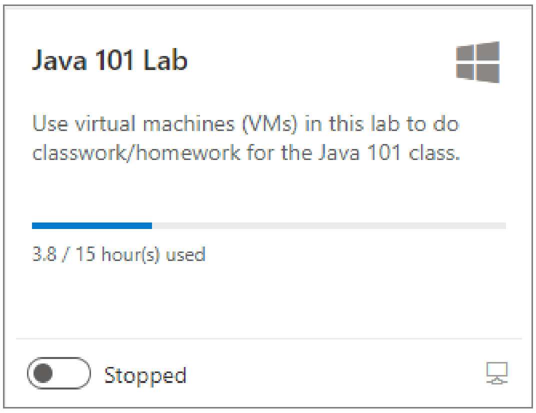
- If a quota isn't assigned (zero hours), the text **Available during classes only** is shown in place of the progress bar.



- If you ran **out of quota**, the color of the progress bar is **red**.



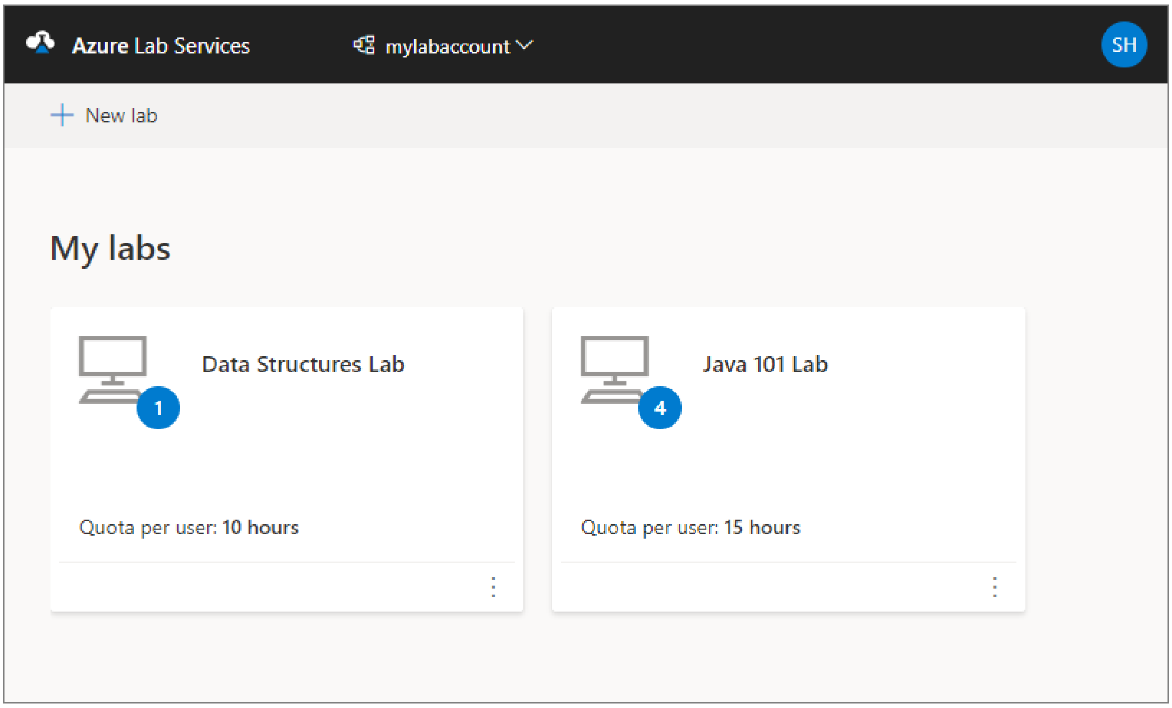
- The color of the progress bar is **blue** when it's outside the scheduled time for the lab and some of the quota time has been used.



View all the labs

After you register to the labs, you can view all the labs by taking the following steps:

1. Navigate to https://labs.azure.com. Internet Explorer 11 isn't supported yet.
2. Sign in to the service by using the user account that you used to register to the lab.
3. Confirm that you see all the labs you have access to.



Next steps

See the following articles:

- As an admin, create and manage lab accounts
- As a lab owner, create and manage labs
- As a lab owner, set up and publish templates
- As a lab owner, configure and control usage of a lab

Connect to Linux virtual machines in a classroom lab of Azure Lab Services

8/5/2021 • 4 minutes to read • Edit Online

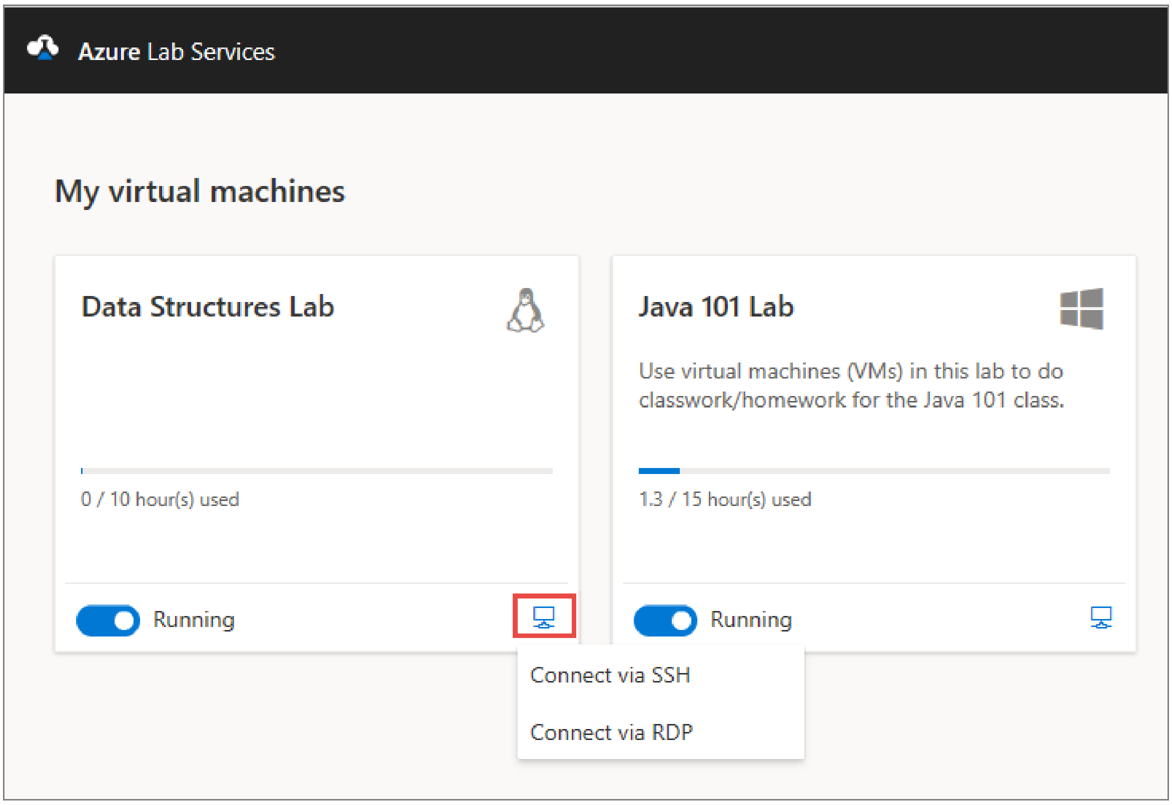
This article shows how students can connect to a Linux virtual machine (VM) in a lab using:

- SSH (secure shell protocol) terminal
- GUI (graphical user interface) remote desktop

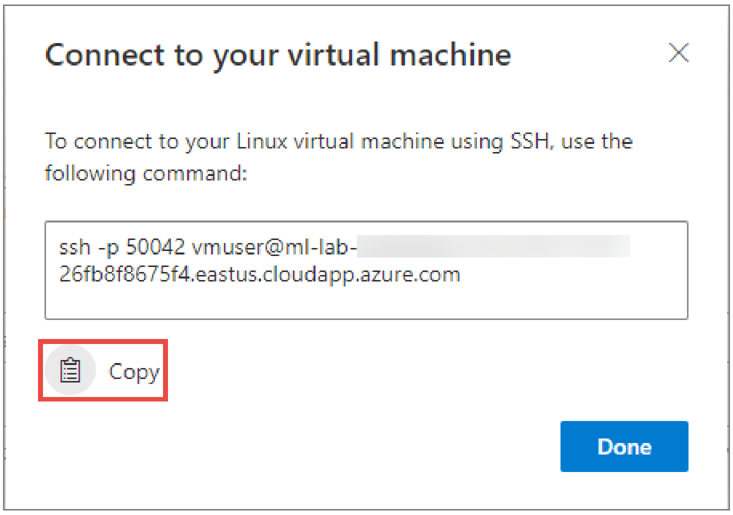
> **IMPORTANT**
> SSH is configured automatically so that both students and the instructor can SSH into Linux VMs without any additional setup. However, if students need to connect to using a GUI remote desktop, the instructor may need to do additional setup. For details, see Enable remote desktop for Linux virtual machines.

Connect to the student VM using SSH

1. When a student signs in to the Labs portal directly (`https://labs.azure.com`) or by using a registration link (`https://labs.azure.com/register/<registrationCode>`), a tile for each lab the student has access to is displayed.
2. On the tile, toggle the button to start the VM if it's in stopped state.
3. Select **Connect**. You see two options to connect to the VM: **SSH** and **RDP**.



4. Select the **SSH** option and you will see the **Connect to your virtual machine** dialog box:



5. Click the **Copy** button next to the text box to copy the SSH connection information to the clipboard.
6. Save the SSH connection information, such as in Text pad, so that you can use this connection information in the next step.
7. From an SSH terminal (like Putty), connect to your VM.

Connect to the student VM using GUI remote desktop

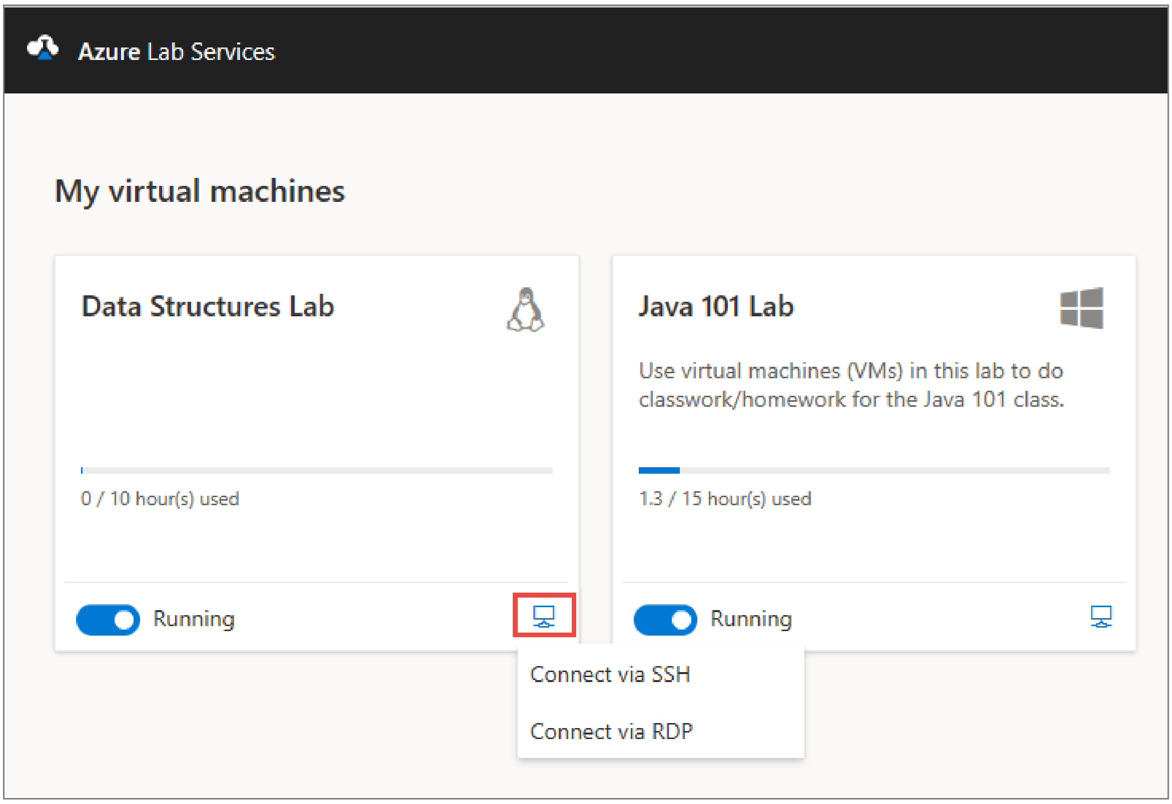
The instructor may choose to configure VMs so that students can also connect using a GUI remote desktop. In this case, students need to find out from their instructor whether to connect to their VMs using the **Microsoft Remote Desktop (RDP)** or **X2Go** client application. Both of these applications allow a student to connect remotely to their VM and display the Linux graphical desktop on their local computer.

> **WARNING**
> We recommend using a different graphical desktop environment than GNOME. You should avoid installing GNOME on lab VMs because GNOME has a conflict with the Azure Linux Agent which is needed for the VMs to work properly in Azure Lab Services. For example, we recommend using a graphical desktop environment, such as XFCE.

Connect to the student VM using Microsoft Remote Desktop (RDP)

Students can use Microsoft Remote Desktop (RDP) to connect to their Linux VMs after their instructor sets up their lab with RDP and GUI packages for a Linux graphical desktop environment (such as XFCE, MATE, and so on). Here are the steps to connect:

1. On the tile for your VM, ensure the VM is running and click **Connect**. You see two options to connect to the VM: **SSH** and **RDP**.



2. Select the **RDP** option. When the RDP file is downloaded onto your machine, save it to your VM.

3. If you are connecting from a Windows computer, typically, the Microsoft Remote Desktop (RDP) client is already installed and configured. As a result, all you need to do is click on the RDP file to open it and start the remote session.

   Instead, if you are connecting from either a Mac or Chromebook, refer to the following steps:

   - Connect to a VM using RDP on a Mac.
   - Connect to a VM using RDP on a Chromebook.

Connect to the student VM using X2Go

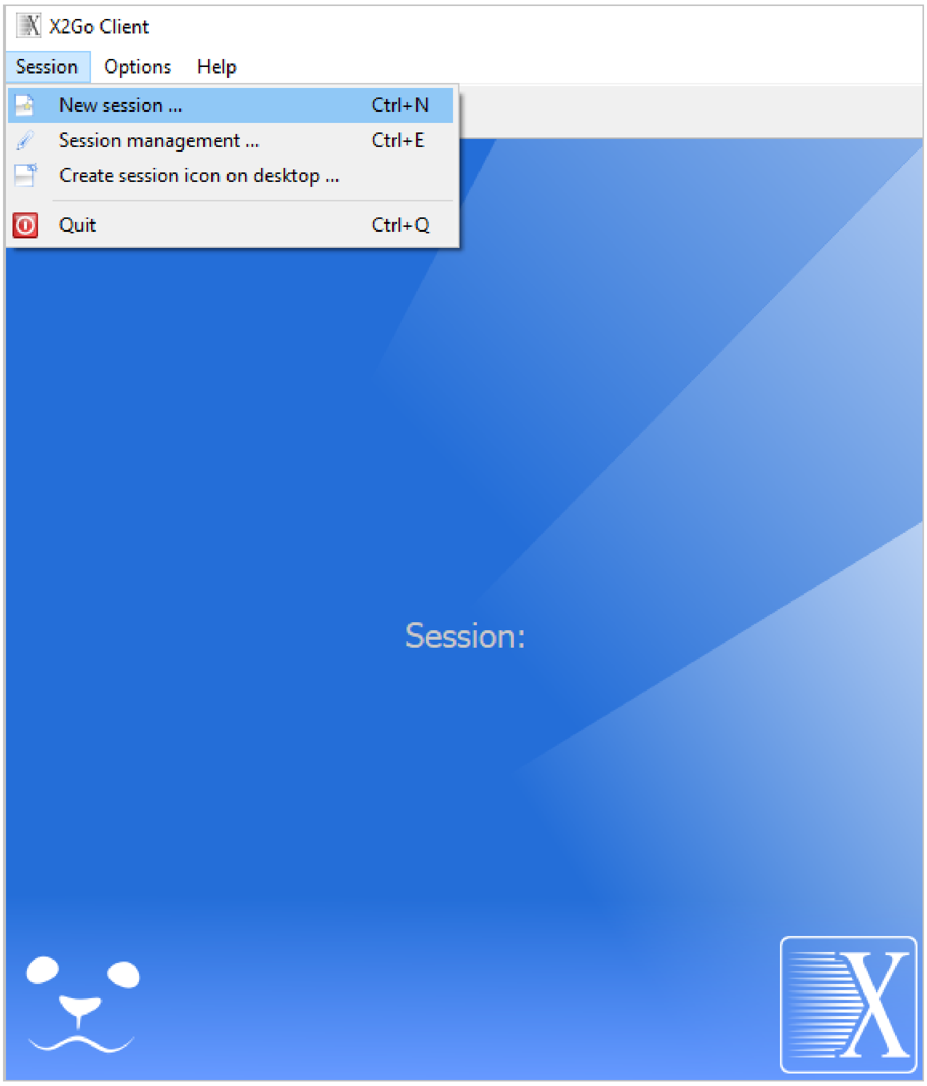
Students can use X2Go to connect to their Linux VMs after their instructor sets up their lab with X2Go and the GUI packages for a Linux graphical desktop environment (such as XFCE, MATE, and so on).

Students need to find out from their instructor which Linux graphical desktop environment their instructor has installed. This information is needed in the next steps to connect using the X2Go client.

1. Install the X2Go client on your local computer.

2. Follow the instructions in the first section to copy the SSH connection information for your VM. You need this information to connect using the X2Go client.

3. Once you have the SSH connection information, open the X2Go client and select **Session** > **New**



Session.

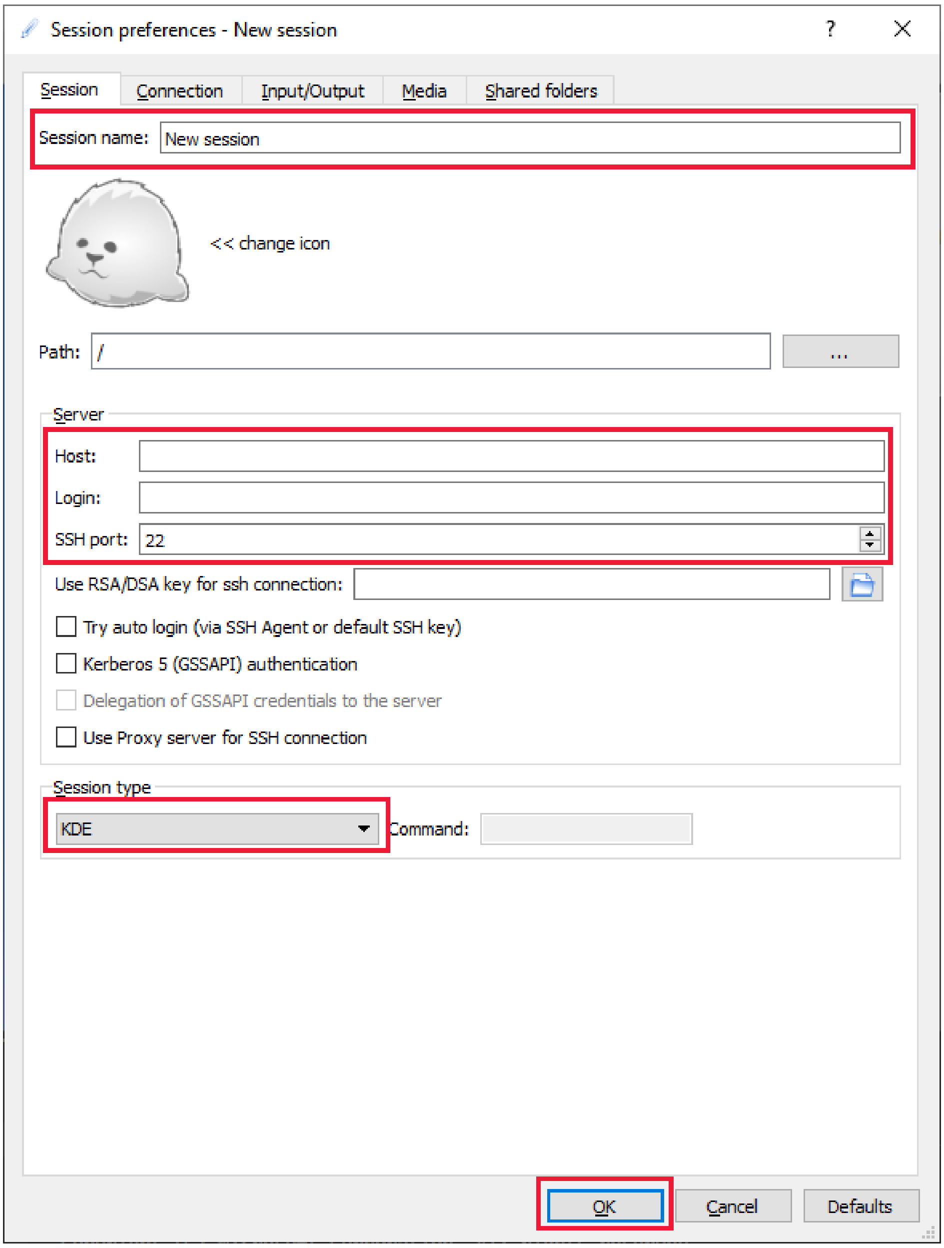
4. Enter the values in the **Session Preferences** pane based on your SSH connection information. For example, your connection information will look similar to this:

```
ssh -p 12345 student@ml-lab-00000000-0000-0000-0000-000000000000.eastus2.cloudapp.azure.com
```

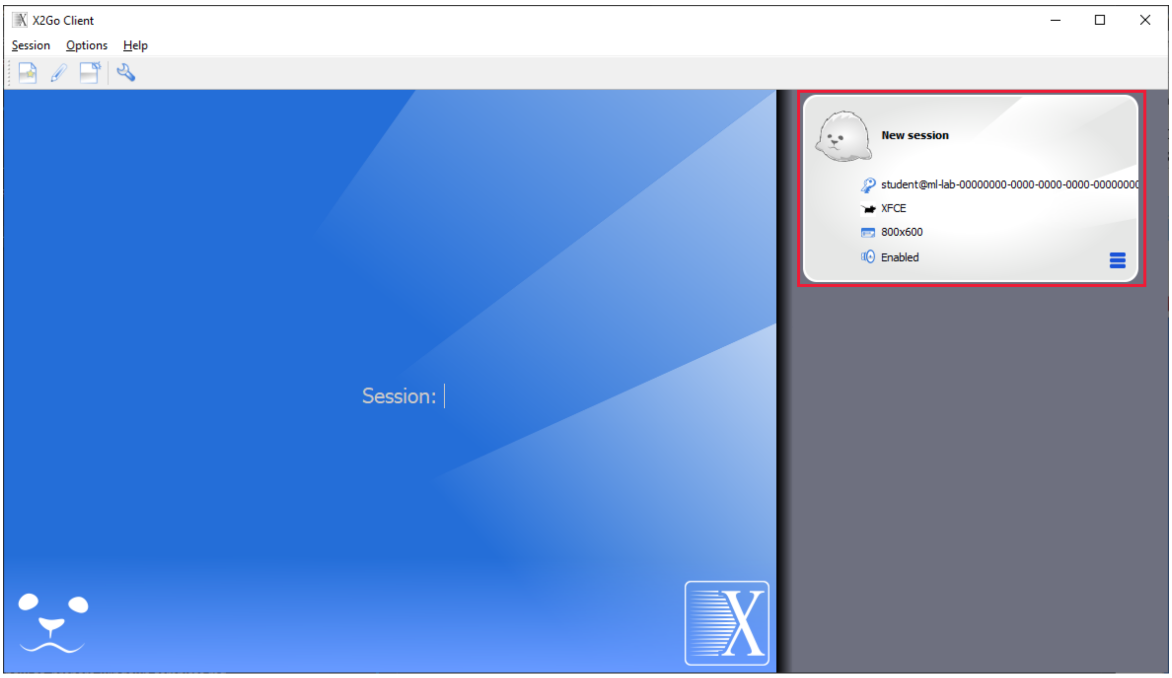
Using this example, the following values are entered:

- **Session name** - Specify a name, such as the name of your VM.
- **Host** - The ID of your VM; for example, `ml-lab-00000000-0000-0000-0000-000000000000.eastus2.cloudapp.azure.com`.
- **Login** - The username for your VM; for example, **student**.
- **SSH port** - The unique port assigned to your VM; for example, **12345**.
- **Session type** - Select the Linux graphical desktop environment that your instructor configured your VM. You need to get this information from your instructor.

Finally, click **OK** to create the session.



5. Click on your session in the right-hand pane.



> **NOTE**
>
> If you are prompted with a similar message to this, select **yes** to continue to entering your password: **The authenticity of host '[** `00000000-0000-0000-0000-000000000000.eastus2.cloudapp.eastus.cloudapp.azure.com` **]:12345' can't be established. ECDSA key fingerprint is SHA256:00000000000000000000000000000000000000000000.Are you sure you want to continue connecting (yes/no)?**

6. When prompted, enter your password and click **OK**. You will now be remotely connected to your VM's GUI desktop environment.

Next steps

To learn how to enable the remote desktop connection feature for Linux VMs in a classroom lab, see Enable remote desktop for Linux virtual machines.

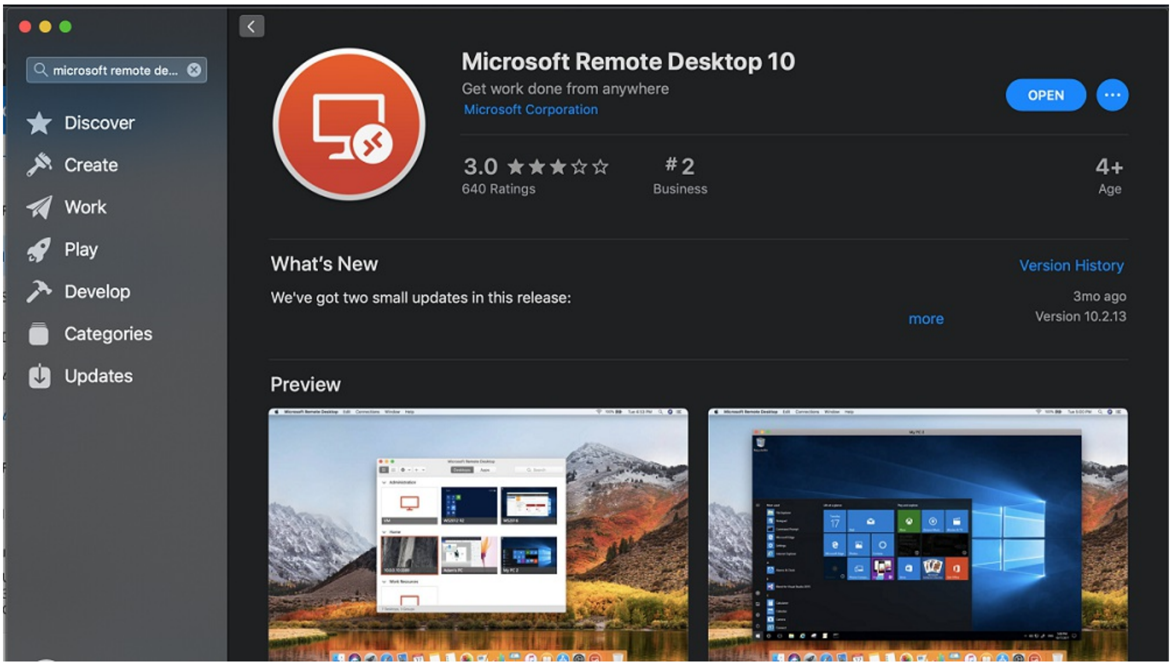
Connect to a VM using Remote Desktop Protocol on a Mac

11/2/2020 • 2 minutes to read • Edit Online

This section shows how a student can connect to a classroom lab VM from a Mac by using RDP.

Install Microsoft Remote Desktop on a Mac

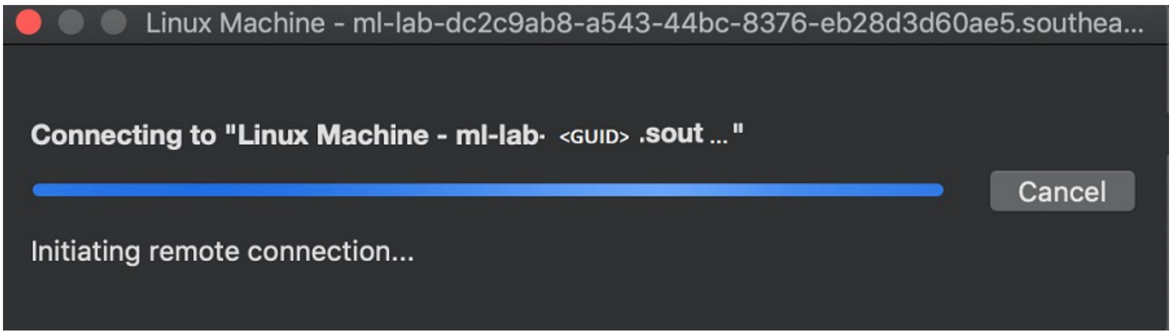
1. Open the App Store on your Mac, and search for **Microsoft Remote Desktop**.



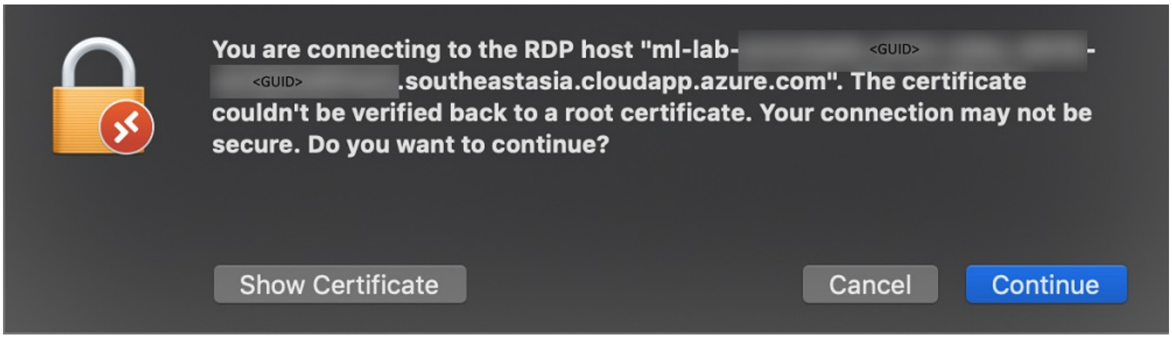
2. Install the latest version of Microsoft Remote Desktop.

Access the VM from your Mac using RDP

1. Open the **RDP** file that's downloaded on your computer with **Microsoft Remote Desktop** installed. It should start connecting to the VM.

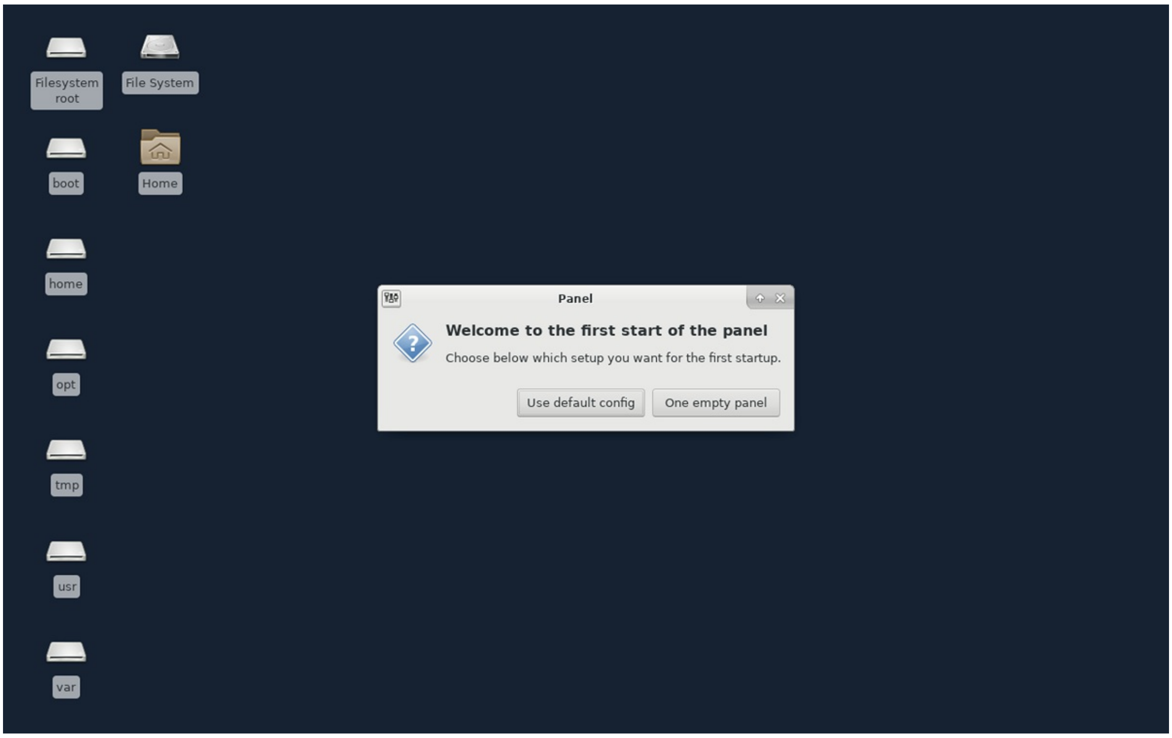


2. Select **Continue** if you receive the following warning.



3. You should see the VM.

> **NOTE**
> The following example is for a CentOS Linux VM.



Next steps

To learn how to connect to Linux VMs using RDP, see Use remote desktop for Linux virtual machines

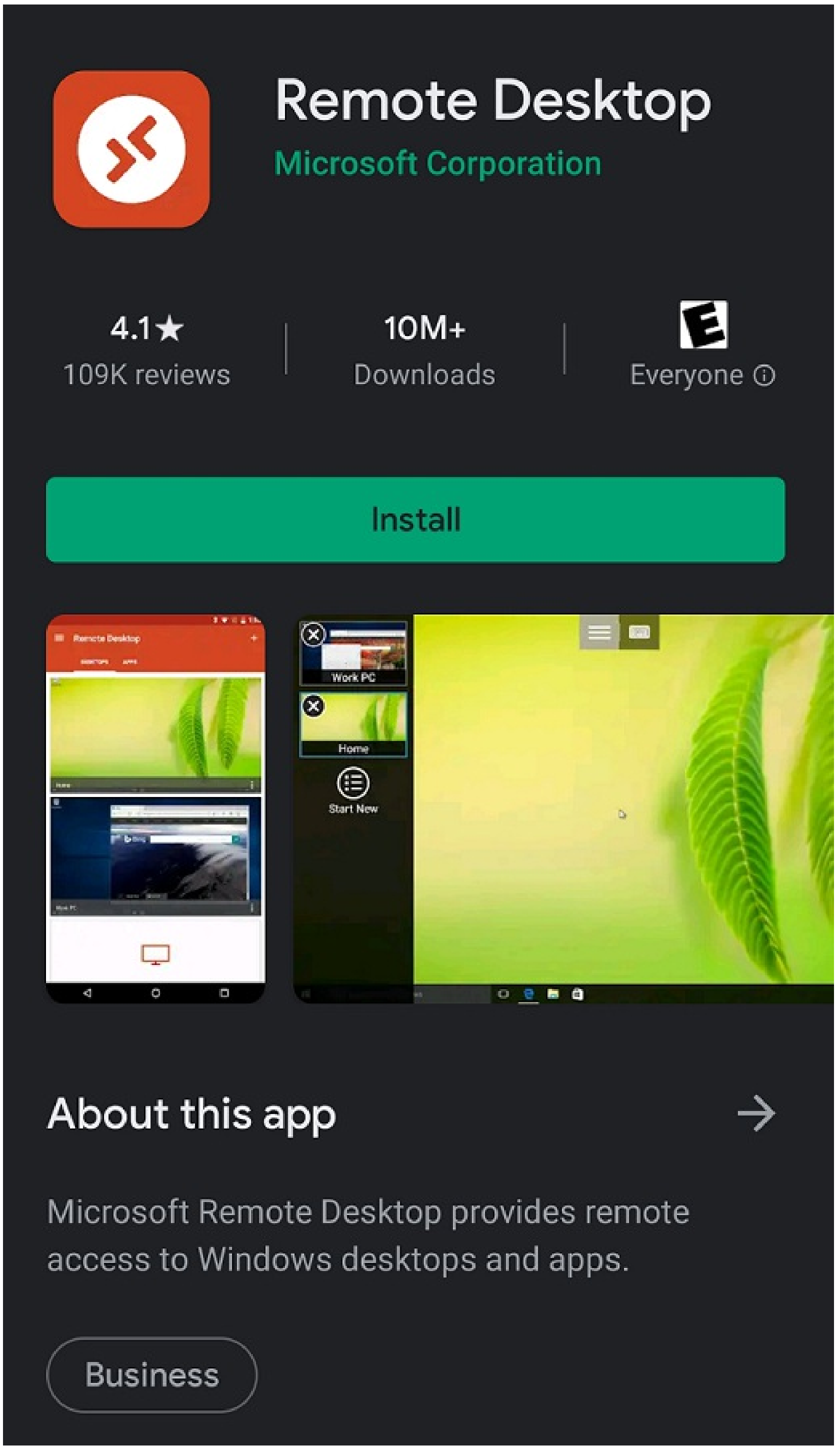
Connect to a VM using Remote Desktop Protocol on a Chromebook

3/5/2021 • 2 minutes to read • Edit Online

This section shows how a student can connect to a classroom lab VM from a Chromebook by using RDP.

Install Microsoft Remote Desktop on a Chromebook

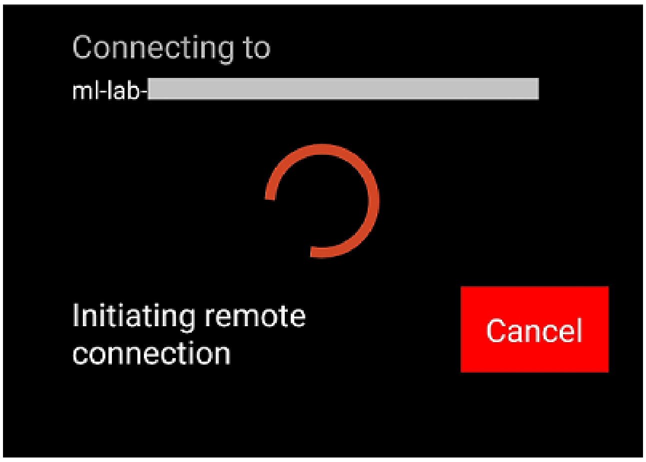
1. Open the App Store on your Chromebook, and search for **Microsoft Remote Desktop**.



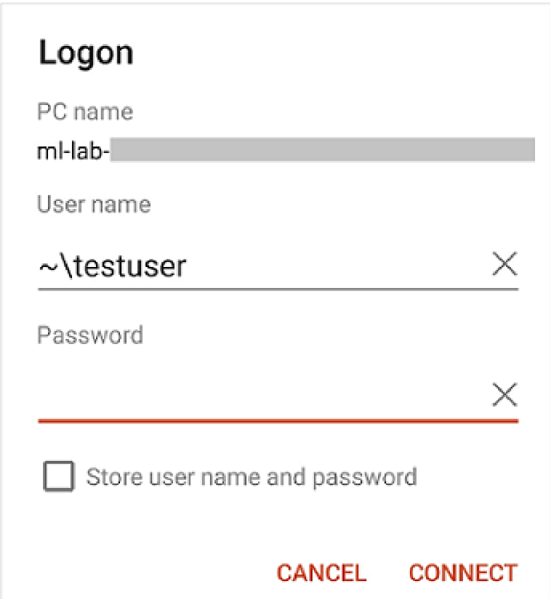
2. Install the latest version of Microsoft Remote Desktop.

Access the VM from your Chromebook using RDP

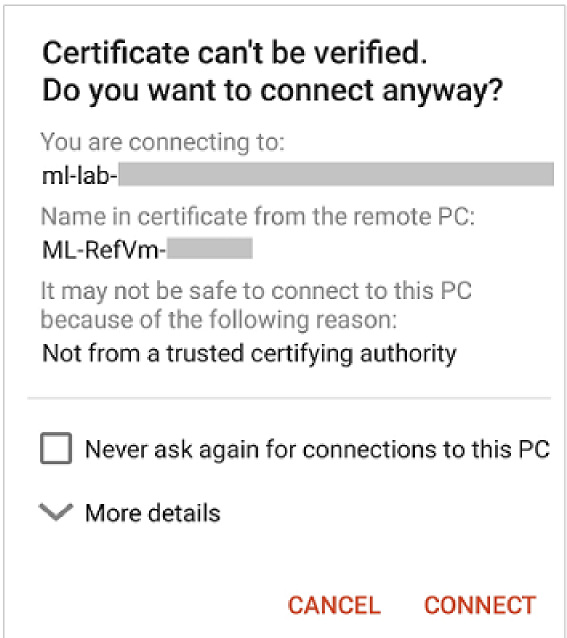
1. Open the **RDP** file that's downloaded on your computer with **Microsoft Remote Desktop** installed. It should start connecting to the VM.



2. When prompted, enter your password.



3. Select **Continue** if you receive the following warning.



4. You should see the desktop of the VM that you are connecting to.

Next steps

To learn more about connecting to Linux VMs, see Connect to Linux virtual machines

Set or reset password for virtual machines in labs (students)

3/5/2021 • 2 minutes to read • Edit Online

This article shows you how students can set/reset password for their VMs.

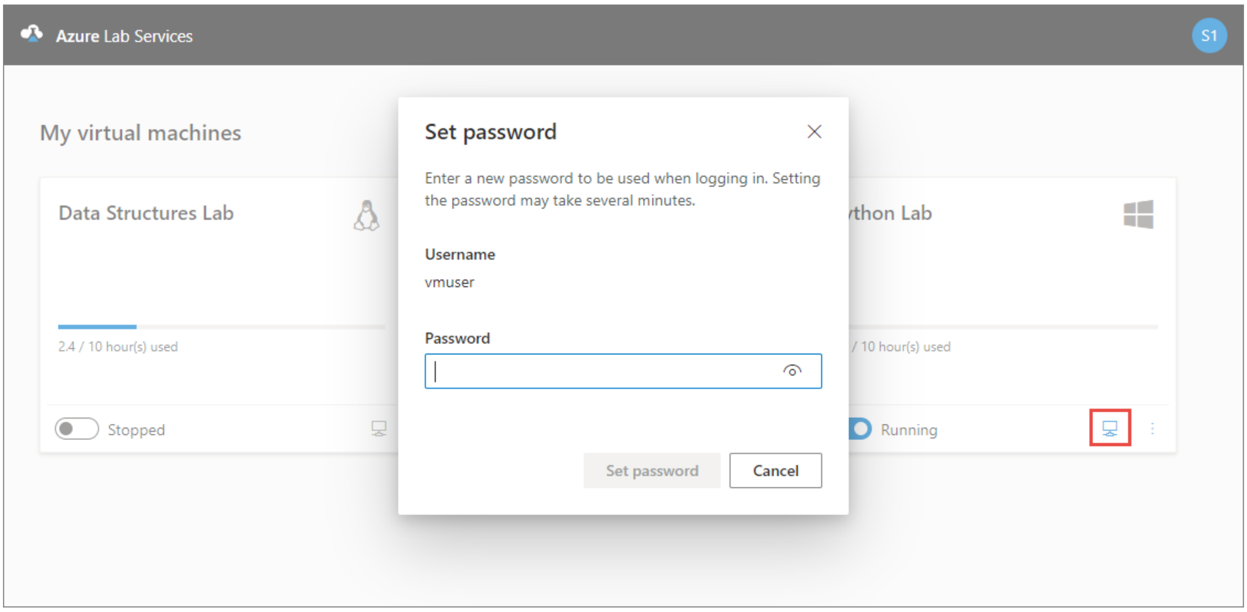
Enable resetting of passwords

At the time of creating the lab, the lab owner can enable or disable the **Use same password for all virtual machines**. If this option was enabled, students can't reset password. All the VMs in the labs will have the same password that's set by the instructor.

If this option is disabled, users will have to set a password when trying to connect to the VM for the first time. Students can also reset the password later at any time as shown in the last section of this article.

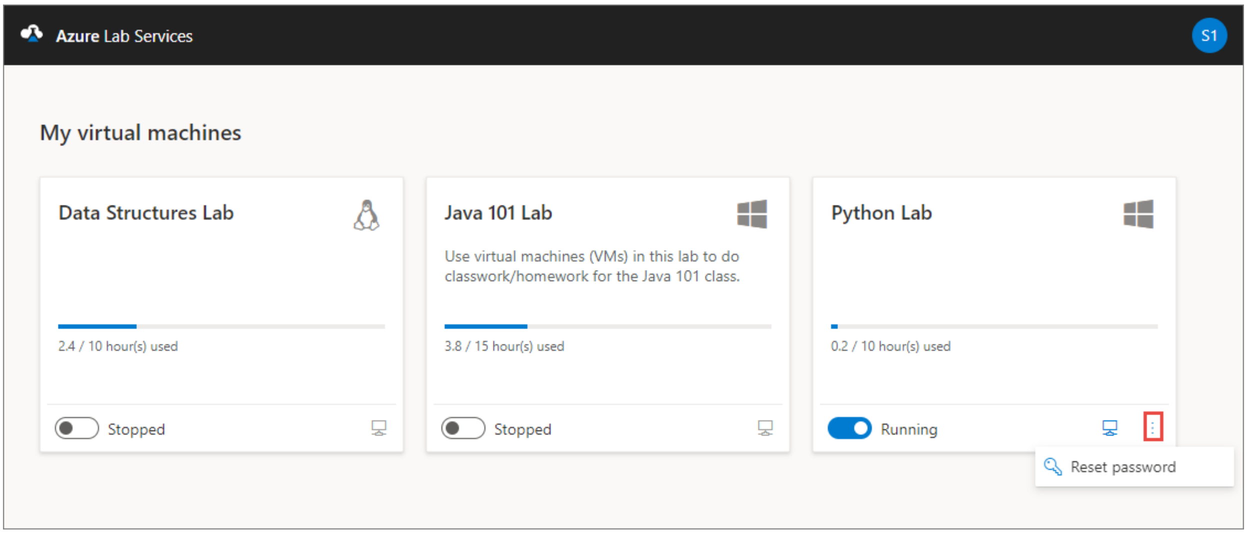
Reset password for the first time

If the **Use same password for all virtual machines** option was disabled, when users (students) select the **Connect** button on the lab tile on the **My virtual machines** page, the user sees the following dialog box to set the password for the VM:



Reset password later

Student also can set the password by clicking the overflow menu (**vertical three dots**) on the lab tile, and selecting **Reset password**.



Next steps

To learn about other student usage options that a lab owner can configure, see the following article: Configure student usage.

View support information (lab user in Azure Lab Services)

11/2/2020 • 2 minutes to read • Edit Online

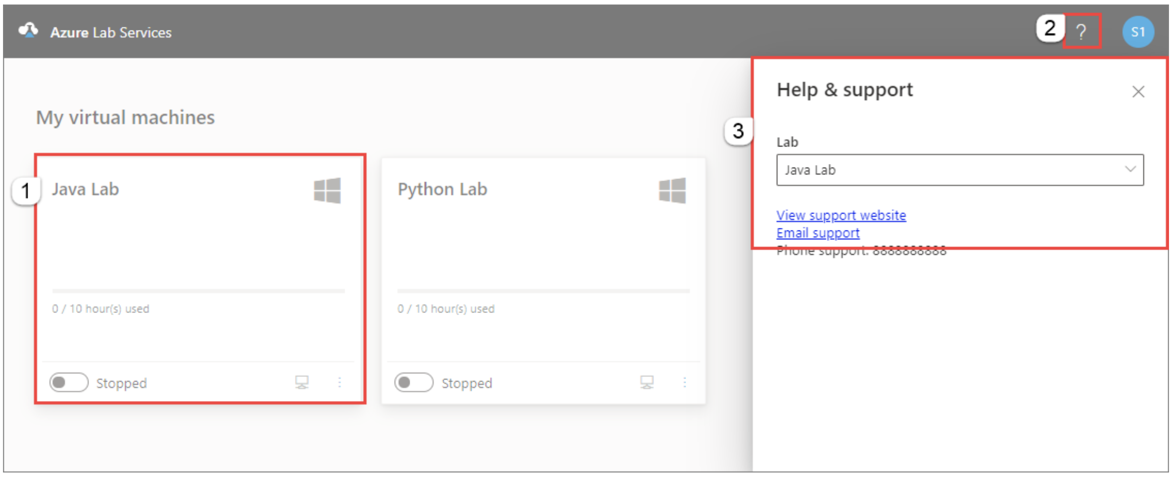
This article explains how you (as a lab user) can view the following support information:

- URL
- Email
- Phone
- Additional instructions

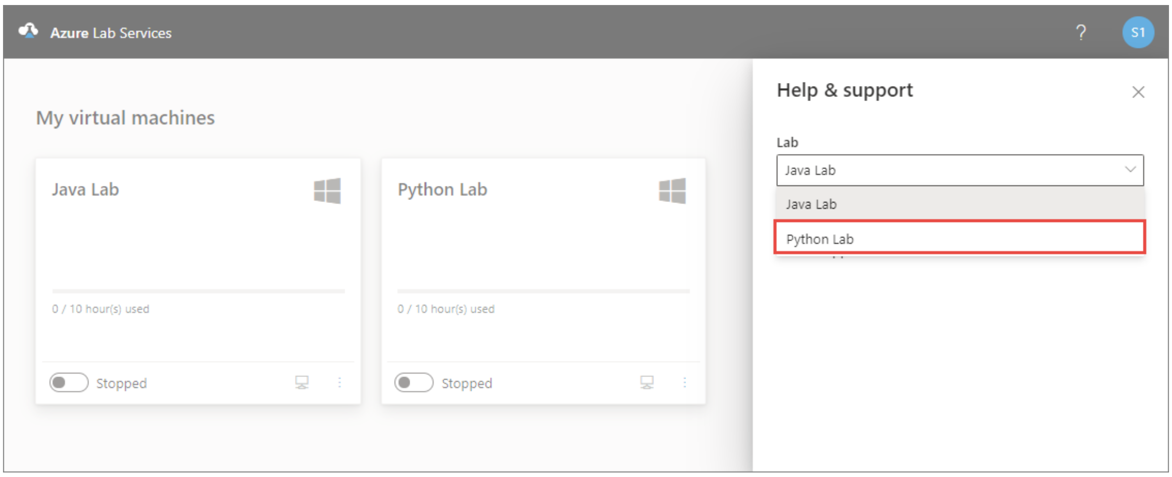
You can use this information to get help when you run into any technical issues while using a lab in a lab account.

View support information

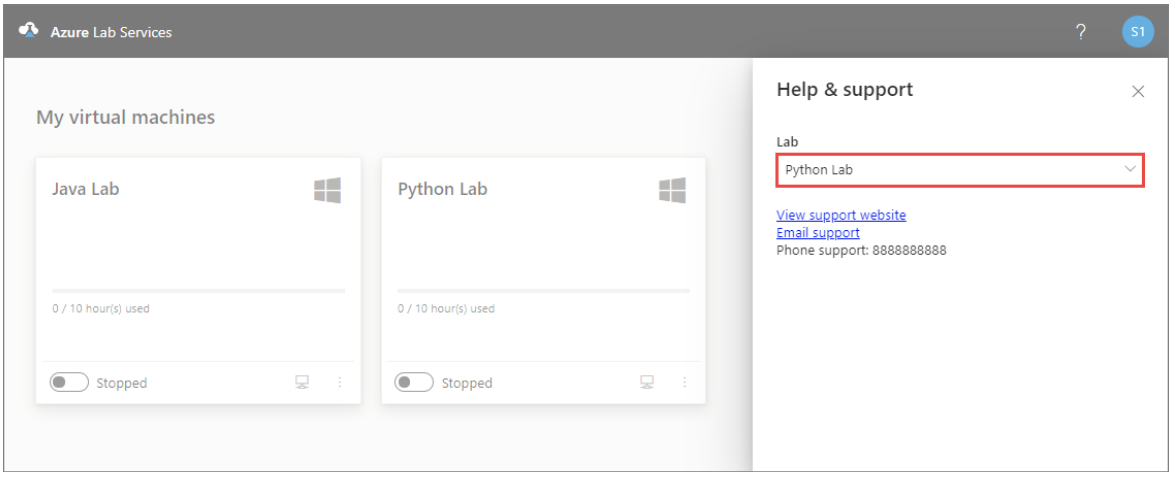
1. Sign in to Azure Lab Services portal.
2. Select the **lab or virtual machine** for which you need help, and select **?** at the top-right corner of the page.
3. Confirm that you see links to the **view support website**, **email support**, and **support phone number**.



4. You can view support contact information for another lab by switching to that lab in the drop-down list.



5. Now, you see the support contact information for the other lab.



Next steps

See the following article to learn about how a lab user views the support contact information:

- How a lab account owner can set support contact information
- How a lab creator can view support contact information

Get started and create a Lab Services lab within Teams

3/5/2021 • 3 minutes to read • Edit Online

This article shows how to add the **Azure Lab Services** app to a Team and then how to create a lab within MS Teams environment.

Prerequisites

In this tutorial you set up a lab with virtual machines for your team. To set up a lab in a lab account, you must be a member of one of these roles in the lab account: Owner, Lab Creator, or Contributor. The account you used to create a lab account is automatically added to the owner role. So, you can use the user account that you used to create a lab account to create a lab.

Here is the typical workflow when using Azure Lab Services within Teams

1. User creates a Lab Account on the Azure portal.
2. A lab account creator adds other users to the **Lab Creator** role. For example, the lab account creator/admin adds educators to the **Lab Creator** role so that they can create labs for their classes.
3. Then, the educators create labs, pre-configures the template VM and publishes the lab to provision VM's to everyone on the team.
4. Once the lab is published, a VM is assigned to everyone on the team membership list on their first login to Azure Lab Services, either by clicking on the tab containing **Azure Lab Services** App within Teams(SSO) or by accessing the labs website. Users can then use the VM to do the class work and homework.

> **IMPORTANT**
>
> Azure Lab Services can be used within Teams only if the lab accounts are created in the same tenant as Teams.

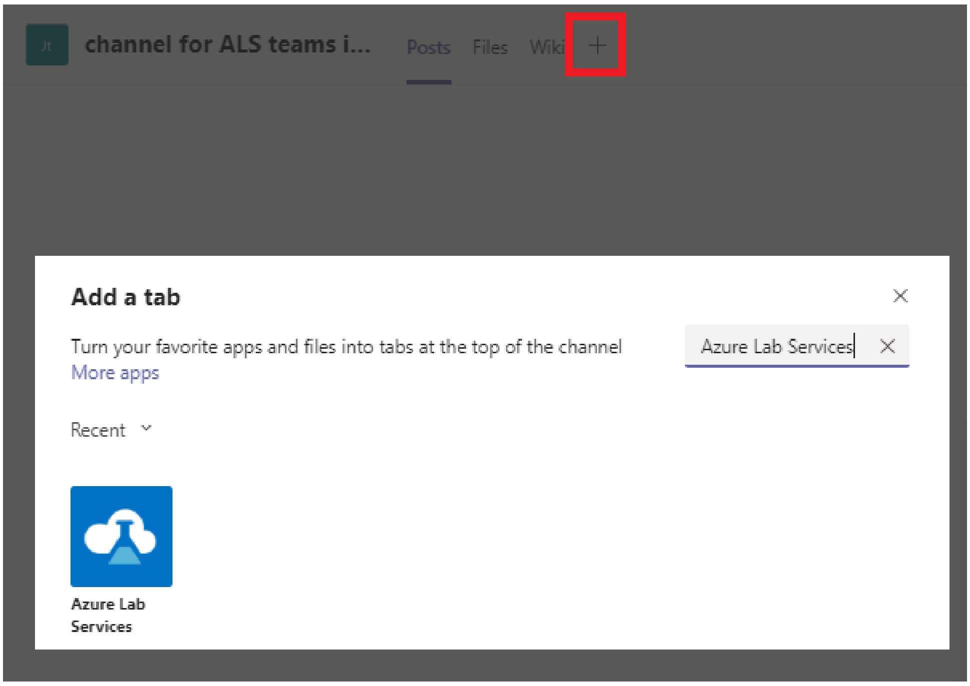
Add Azure Lab Services app as a tab to a Team

You, as a Team owner, can add **Azure Lab Services** app directly in your Teams channels, and then the app is available for everyone in the team to use. Follow the below three steps:

1. Navigate to the Teams channel where you want to add the app and select **+** to add a tab.

2. Search for **Azure Lab Services** from the tab options and add this app.

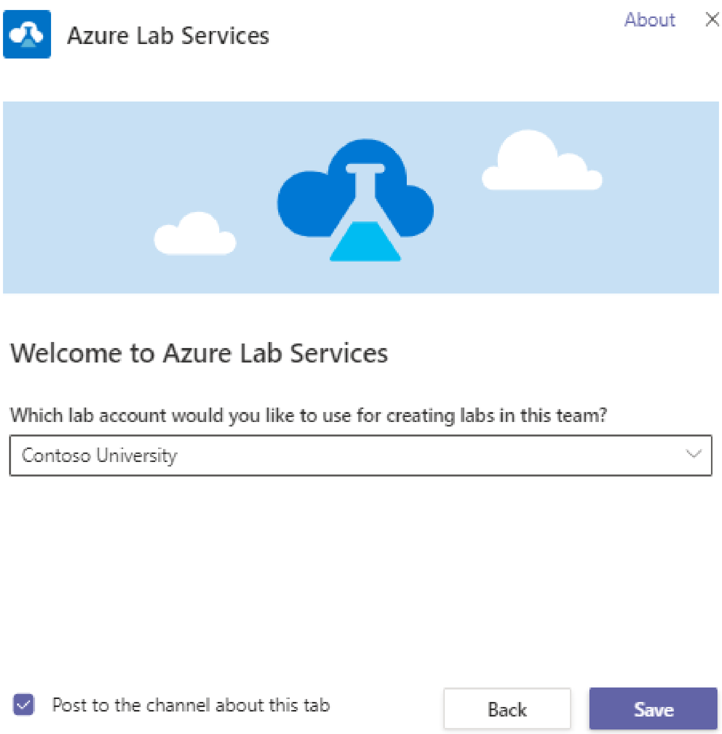
   > **NOTE**
   >
   > Only Team **Owners** will be able to create labs for the team.



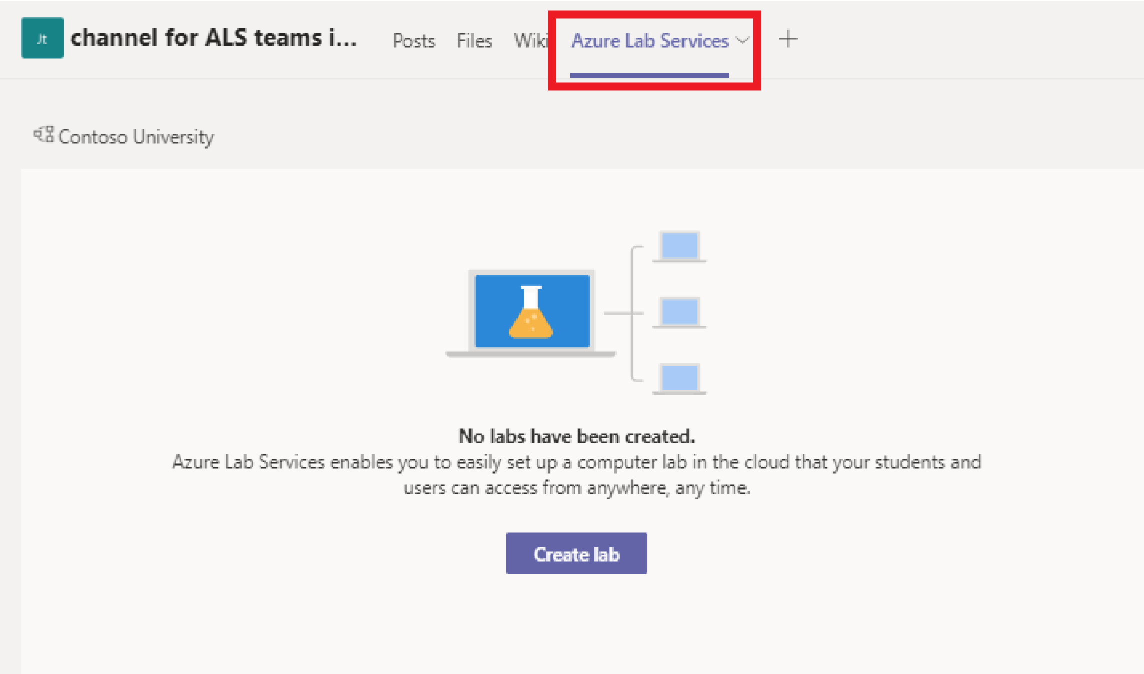
3. Select a Lab Services account, which you would like to use for creating labs in this team.

   Azure Lab Services uses single sign-on into the Azure Lab Services website and pulls all the lab accounts that you have access to.

   The accounts that are in the same tenant as Teams and for which you have **Owner**, **Contributor**, or **Creator** access are displayed.



4. Press **Save** and the tab gets added to the channel.



Now you can select the **Azure Lab Services** tab from your channel and start managing labs as described in the following articles.

After the lab account is selected, Team owners will be able to create labs for the team. The entire lab creation process and all the tasks at the lab level can be performed within Teams. Users will have the option to create multiple labs within the same team and the Team owner, with appropriate access at the lab account level, will see only the labs associated with the specific team.

Next steps

When a lab is created within Teams, the lab user list is automatically populated and synced with the team membership. Everyone on the team, including Owners, Members and Guests will be automatically added to the lab user list. Azure lab Services will maintain a sync with the team membership and an automatic sync is triggered every 24 hours. For details, see:

Manage Lab Services user lists within Teams

See also

Also see the following articles:

- Use Azure Lab Services within Teams overview
- Manage lab's VM pool within Teams
- Create and manage lab schedules within Teams
- Access a VM within Teams – Student view
- Delete labs within Teams

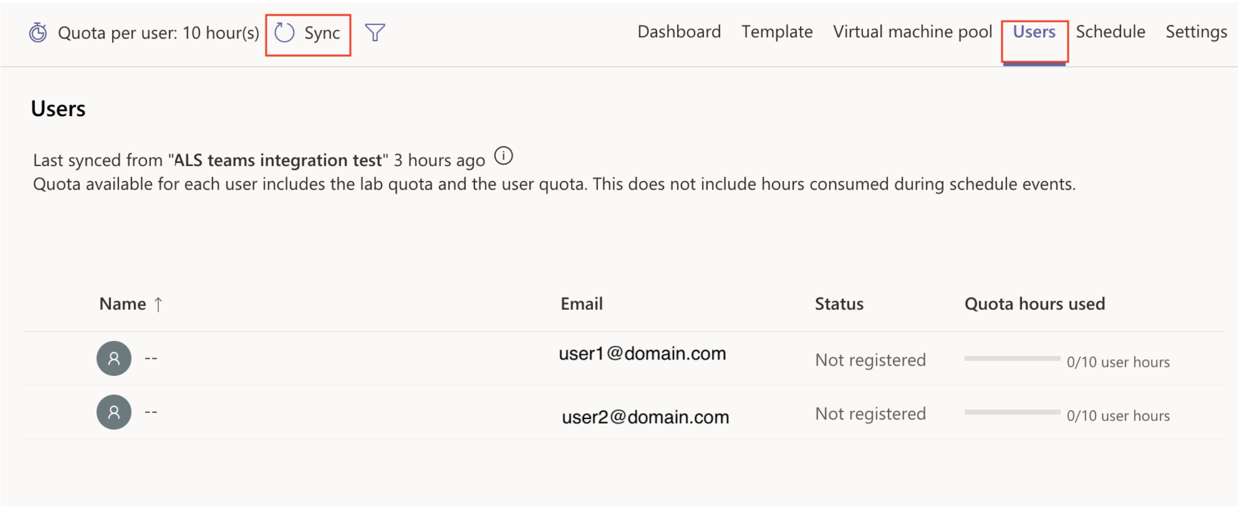
Manage Lab Services user lists from Teams

11/2/2020 • 2 minutes to read • Edit Online

When a lab is created within Teams (see Get started and create a Lab Services lab from Teams), the lab user list is automatically populated and synced with the team membership. Everyone on the team, including Owners, Members, and Guests will be automatically added to the lab user list. Azure lab Services maintains a sync with the team membership and an automatic sync is triggered every 24 hours.

Sync users

Educators can use the **Sync** button to trigger a manual sync once the team membership is updated.



Once the automatic or manual sync is complete the following is true depending on whether the lab has been published.

- If the lab has not been published at least once:
  - Users will be added or deleted from the lab user list as per changes to the team membership.
- If the lab has been published at least once, in addition to adding or deleting users, the lab capacity will be automatically updated.
  - If there are any new additions to the team, new VMs will be created.
  - If any user has been deleted from the team, the associated VM will be deleted as well.

Next steps

Once the template VM is configured and when the educator selects to publish the template, number of VMs equivalent to the number of users in the lab's user list will be created. Once the lab is published and VMs are created, Users will be automatically registered to the lab and VMs will be assigned to them on their first login to Azure Lab Services that is, when they first access the tab having **Azure Lab Services** App.

To publish the template VM, go to the Teams Lab Services window, select **Template** tab -> ... -> **Publish**.

To manage VM pools, see Manage a VM pool in Lab Services from Teams.

Also review

See the following articles:

- Use Azure Lab Services within Teams overview

- Get started and create a Lab Services lab from Teams
- Create Lab Services schedules from Teams
- Access a VM (student view) in Lab Services from Teams

Manage a VM pool in Lab Services from Teams

11/2/2020 • 2 minutes to read • Edit Online

Virtual Machine (VM) creation starts as soon as the template VM is first published. VMs equaling the number of users in the lab user list will be created. VMs are automatically assigned to students upon their first login to Azure Lab Services.

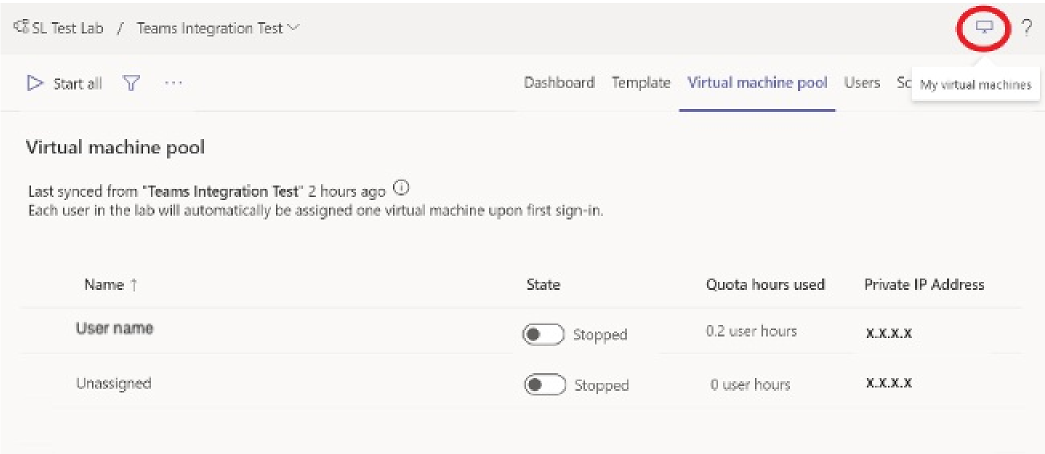
Publish a template and manage a VM pool

To publish the template, go to the Teams Lab Services window, select **Template** tab -> ... -> **Publish**.

Once the template VM is configured and when the educator selects to publish the template, number of VMs equivalent to the number of users in the lab's user list will be created. Once the lab is published and VMs are created, Users will be automatically registered to the lab and VMs will be assigned to them on their first login to Azure Lab Services that is, when they first access the tab having **Azure Lab Services** App.

When a user list sync is triggered, Lab Capacity (number of VMs in the lab) will be automatically updated based on the changes to the team membership. New VMs will be created as new users are added and VMs assigned to the users removed from the team will be deleted as well. For more information see How to manage users within Teams.

Educators can continue to access student VMs directly from the VM Pool tab. And educators can access VMs assigned to themselves either from the **Virtual machine pool** tab or by clicking on the **My Virtual Machines** button (top/right corner of the screen).



Next steps

See the following articles:

- Use Azure Lab Services within Teams overview
- Get started and create a Lab Services lab from Teams
- Manage Lab Services user lists from Teams
- Create Lab Services schedules from Teams
- Access a VM (student view) in Lab Services from Teams

Delete labs within Teams

3/5/2021 • 2 minutes to read • Edit Online

This article shows how to delete a lab from the **Azure Lab Services** app.

Prerequisites

- Create a Lab Services account in the Azure portal.
- Get started and create a Lab Services lab within Teams.

Delete labs

A lab created within Teams can be deleted in the Lab Services website by deleting the lab directly, as described in Manage labs in Azure Lab Services.

Lab deletion is also triggered when the team is deleted. If the team in which the lab is created gets deleted, lab would be automatically deleted 24 hours after the automatic user list sync is triggered.

> **IMPORTANT**
>
> Deletion of the tab or uninstalling the app will not result in deletion of the lab.

If the tab is deleted, users on the team membership list will still be able to access the VMs on the Lab Services website unless the lab deletion is explicitly triggered by deleting the lab on website or deleting the team.

Next steps

- Use Azure Lab Services within Teams overview
- Manage lab user lists within Teams
- Manage lab's VM pool within Teams
- Create and manage lab schedules within Teams
- Access a VM within Teams – Student view

Create and manage Lab Services schedules within Teams

11/2/2020 • 2 minutes to read • Edit Online

Schedules allow you to configure a classroom lab such that VMs in the lab automatically start and shut down at a specified time. You can define a one-time schedule or a recurring schedule. The following procedures give you steps to create and manage schedules for a classroom lab:

Here's how schedules affect lab virtual machines:

- Template virtual machine is not included in schedules.
- Only assigned virtual machines are started. This means, if a machine is not claimed by an end user (student), the machine will not start on the scheduled hours.
- All virtual machines (whether claimed by a user or not) are stopped based on the lab schedule.

> **IMPORTANT**
>
> The scheduled running time of VMs does not count against the quota allotted to a user. The quota is for the time outside of schedule hours that a student spends on VMs.

Users can create, edit, and delete lab schedules within Teams just as in the labs website. Refer to the article on creating and managing schedules.

Automatic shutdown and disconnect settings

You can enable several autoshutdown cost control features to proactively prevent additional costs when the virtual machines are not being actively used. The combination of the following three automatic shutdown and disconnect features catches most of the cases where users accidentally leave their virtual machines running:

- Automatically disconnect users from virtual machines that the OS deems idle.
- Automatically shut down virtual machines when users disconnect.
- Automatically shut down virtual machines that are started but users don't connect.

For more details, refer to the article on configuring auto-shutdown settings for a lab.

Next steps

See the following articles:

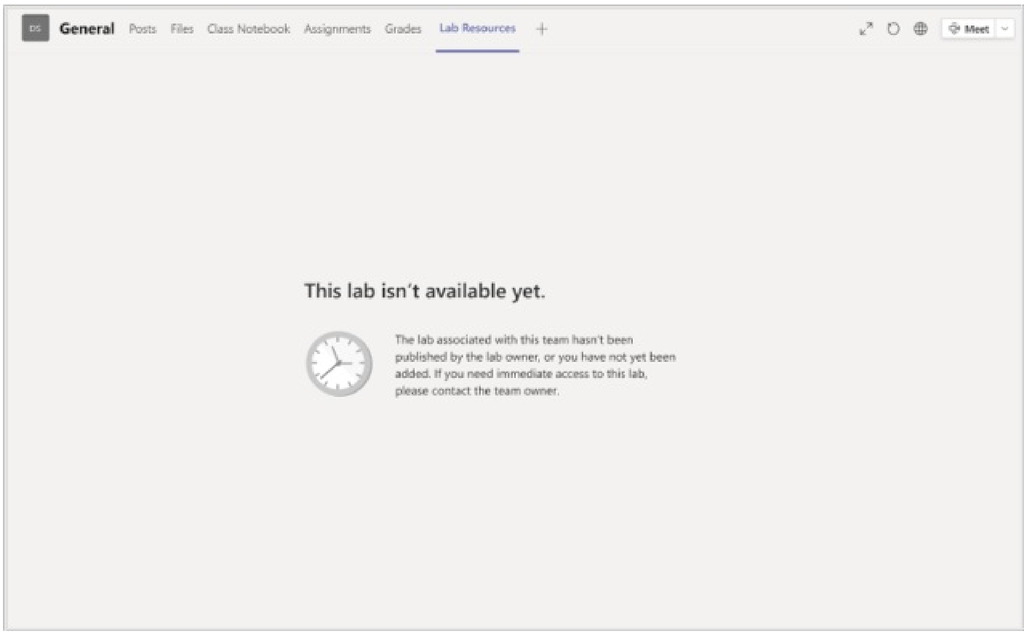
- Use Azure Lab Services within Teams overview
- Get started and create a lab within Teams
- Manage lab user lists within Teams
- Manage lab's VM pool within Teams
- Access a VM within Teams – Student view

Access a VM (student view) in Azure Lab from Teams

11/2/2020 • 2 minutes to read • Edit Online

When a lab is created within Teams, users can view and access all the VMs provisioned by the team owner. Once the lab is published and VMs are created, users will be automatically registered to the lab and assigned a VM on their first login to Azure Lab Services. Users can view and access the VM's assigned to them by selecting the tab containing **Azure Lab Services** app.

If the lab is yet to be published or a sync is yet to be triggered after they are added to the team, similar messaging will be shown to the students.



Next steps

For more information, see the following articles:

- Use Azure Lab Services within Teams overview
- Get started and create a lab within Teams
- Manage lab user lists within Teams
- Manage lab's VM pool within Teams
- Create and manage lab schedules within Teams

Az.LabServices PowerShell module (preview)

3/5/2021 • 2 minutes to read • Edit Online

Az.LabServices is a PowerShell module that simplifies the management of Azure Lab services. It provides composable functions to create, query, update and delete lab accounts, labs, VMs, and Images. For more information about this module, see the Az.LabServices home page on GitHub.

NOTE

This module is in **preview**.

Example command

Here is an example of using a PowerShell command to stop all the running VMs in all labs.

```
Get-AzLabAccount | Get-AzLab | Get-AzLabVm -Status Running | Stop-AzLabVm
```

Get started

1. Install Azure PowerShell if it doesn't exist on your machine.
2. Download Az.LabServices.psm1 to your machine.
3. Import the module:

   ```
   Import-Module .\Az.LabServices.psm1
   ```

4. Run the following command to list all the labs in your subscription.

   ```
   Get-AzLabAccount | Get-AzLab
   ```

Next steps

See the Az.LabServices home page on GitHub.