

This deployment guide shows you how to deploy and connect to a Compute Engine virtual machine (VM) that is ready for the installation of SAP MaxDB on a Windows Server operating system.

The instructions in this guide use [Cloud Deployment Manager](#) (/deployment-manager/docs/fundamentals) to deploy and configure the VM, the Windows Server operating system, and the required disk volumes for SAP MaxDB.

For details on planning your deployment, see the [SAP MaxDB Planning Guide](#) (/solutions/sap/docs/sap-maxdb-planning-guide).

If you do not already have a Google Cloud project with billing enabled, you must create one before you can deploy a VM for your SAP MaxDB installation.

To create a project:

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[Go to the project selector page](https://console.cloud.google.com/projectselector2/home/dashboard) (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](#) (/billing/docs/how-to/modify-project).

These instructions use [Cloud Shell](#) (/shell/docs/) to enter `gcloud` commands that deploy or configure your Google Cloud resources. Cloud Shell is accessed through the Cloud Console in your browser.

Cloud Shell runs on a VM that Google Cloud provisions each time you start Cloud Shell. The first time you use Cloud Shell, Google Cloud also creates a persistent `$HOME` directory for you, which is restored each time you open Cloud Shell.

The provisioned VM includes the latest [Cloud SDK](#) (`/sdk/docs/`), which provides the `gcloud` command-line interface. Therefore, the `gcloud` commands that you use in Cloud Shell are the same as those you would use in a locally installed instance of the Cloud SDK.

If you have the Cloud SDK installed, you can issue the `gcloud` commands that are used in these instructions from your local machine. However, with a locally installed Cloud SDK you must always make sure that you are using the latest version of the Cloud SDK.

Whether you use Cloud Shell or Cloud SDK, you can set and change the properties of your `gcloud` command environment and save them as a *configuration*. Configurations are collections of key-value pairs that influence the behavior of the `gcloud` commands.

Some basic actions you can take with a configuration in Cloud Shell include:

- Initialize a configuration with `gcloud init`.
- Check the settings of your current `gcloud` configuration with `gcloud config list`.
- Change the Google Cloud project you are working in with `gcloud config set project [PROJECT_ID]` where `[PROJECT_ID]` represents your Google Cloud project.
- Set a default region with `gcloud config set compute/region [REGION]` where `[REGION]` represents a Google Cloud region.
- Set a default zone with `gcloud config set compute/zone [ZONE]` where `[ZONE]` represents a Google Cloud zone.
- Create a new configuration with `gcloud config configurations create [NAME]` where `[NAME]` represents the name for the configuration.

For more information about working with configurations, see [Managing Cloud SDK configurations](#) (`/sdk/docs/configurations`).

For security purposes, create a new network. You can control who has access by adding firewall rules or by using another access control method.

If your project has a default VPC network, don't use it. Instead, create your own VPC network so that the only firewall rules in effect are those that you create explicitly.

During deployment, VM instances typically require access to the internet to download Google's monitoring agent. If you are using one of the SAP-certified Linux images that are available from Google Cloud, the VM instance also requires access to the internet in order to register the license and to access OS vendor repositories. A configuration with a NAT gateway and with VM network tags supports this access, even if the target VMs do not have external IPs.

To set up networking:

1. Go to Cloud Shell.

[Go to Cloud Shell](https://console.cloud.google.com/?cloudshell=true) (https://console.cloud.google.com/?cloudshell=true)

2. To create a new network in the custom subnetworks mode, run:

where [YOUR_NETWORK_NAME] is the name of the new network. The network name can contain only lowercase characters, digits, and the dash character (-).

Specify `--subnet-mode custom` to avoid using the default auto mode, which automatically creates a subnet in each Compute Engine region. For more information, see [Subnet creation mode](#) (/vpc/docs/vpc#subnet-ranges).

3. Create a subnetwork, and specify the region and IP range:

where:

- [YOUR_SUBNETWORK_NAME] is the new subnetwork.
- [YOUR_NETWORK_NAME] is the name of the network you created in the previous step.
- [REGION] is the region where you want the subnetwork.
- [YOUR_RANGE] is the IP address range, specified in CIDR format (https://wikipedia.org/wiki/Classless_Inter-Domain_Routing), such as 10.1.0.0/24. If you plan to add more than one subnetwork, assign non-overlapping CIDR IP ranges for each

subnetwork in the network. Note that each subnetwork and its internal IP ranges are mapped to a single region.

4. Optionally, repeat the previous step and add additional subnetworks.

If you intend to create one or more VMs that will not have public IP addresses, you must create a NAT gateway so that your VMs can access the Internet to download Google's monitoring agent.

If you intend to assign an external public IP address to your VM, you can skip this step.

Important: Do not remove the public IP addresses from your new VMs until after the installation of your SAP software is complete and validated.

To create a NAT gateway:

1. Create a VM to act as the NAT gateway in the subnet you just created:

where:

- [YOUR_VM_NAME] is the name of the VM you are creating that want to use for the NAT gateway.
- [YOUR_ZONE] is the zone where you want the VM.
- [YOUR_IMAGE_FAMILY] and [YOUR_IMAGE_PROJECT] specify the image you want to use (/compute/docs/images#os-compute-support) for the NAT gateway.
- [YOUR_MACHINE_TYPE] is any supported machine type. If you expect high network traffic, choose a machine type with that has at least eight virtual CPUs.
- [YOUR_SUBNETWORK_NAME] is the name of the subnetwork where you want the VM.

- [YOUR_VM_TAG] is a tag that is applied to the VM you are creating. If you use this VM as a bastion host, this tag is used to apply the related firewall rule only to this VM.

2. Create a route that is tagged so that traffic passes through the NAT VM instead of the default Internet gateway:

where:

- [YOUR_ROUTE_NAME] is the name of the route you are creating.
- [YOUR_NETWORK_NAME] is the network you created.
- [YOUR_VM_NAME] is the VM you are using for your NAT gateway.
- [YOUR_ZONE] is the zone where the VM is located.
- [YOUR_TAG_NAME] is the tag on the route that directs traffic through the NAT VM.

3. If you also want to use the NAT gateway VM as a bastion host, run the following command. This command creates a firewall rule that allows inbound SSH access to this instance from the Internet:

where:

- [YOUR_NETWORK_NAME] is the network you created.
- [YOUR_VM_TAG] is the tag you specified when you created the NAT gateway VM. This tag is used so this firewall rule applies only to the VM that hosts the NAT gateway, and not to all VMs in the network.

The following instructions use Deployment Manager to deploy a VM instance with Windows and all of the persistent disks that SAP MaxDB requires.

In these instructions, you define the resource options for your installation in a Deployment Manager configuration file template.

Deployment Manager treats all of the resources that are created for your SAP system as a single entity called a *deployment*. You can view and work with all of the deployments for your project on the [Deployments](https://console.cloud.google.com/dm/deployments) (https://console.cloud.google.com/dm/deployments) page in the Cloud Console.

Be aware of the following behaviors when using Deployment Manager:

- Deleting a deployment deletes all of the resources associated with the deployment, including the VMs, the persistent disks, and any SAP systems that are installed on the VM.
- By default, Deployment Manager uses the **ACQUIRE** resource creation policy. If you specify a VM name that is already in use by another VM in your project, Deployment Manager doesn't create a new VM, but instead adds the existing VM to your new deployment. If your original VM was created by a previous run of Deployment Manager, the VM is associated with two deployments.

If you then delete the new deployment, the acquired VM is deleted from the deployment that originally created it. To avoid such a scenario, either set the Deployment Manager resource policy to **CREATE**, or make sure that you use unique resource names in your new deployment.

For information about the policies you can use while creating resources with Deployment Manager and how to specify them, see the [Deployment Manager documentation](#) (/deployment-manager/docs/deployments/#create-policy).

1. Open the Cloud Shell.

[Go to the Cloud Shell](https://console.cloud.google.com/?cloudshell=true) (https://console.cloud.google.com/?cloudshell=true)

2. Download the template.yaml configuration file template to your working directory by entering the following command in the Cloud Shell:

3. Optionally, rename the `template.yaml` file to identify the configuration it defines. For example, `maxdb-win-2016.yaml`.

4. Open the `template.yaml` file in Cloud Shell code editor.

To open the Cloud Shell code editor, click the pencil icon in the upper right corner of the Cloud Shell terminal window.

5. In the `template.yaml` file, update the following property values by replacing the brackets and their contents with the values for your installation.

Property	Data type	Description
<code>instanceName</code>	String	The name of the VM instance on which SAP MaxDB will be installed. The name must be 13 characters or less and be specified in lowercase letters, numbers, or hyphens.
<code>instanceType</code>	String	The type of Compute Engine virtual machine (<code>/compute/docs/machine-types</code>) on which SAP MaxDB will be installed. Specify a machine type with two or more vCPUs. For example, <code>n1-standard-4</code> . If you are running SAP MaxDB on the same VM as SAP NetWeaver, select a machine type that includes enough CPUs and memory to support both systems. See the SAP NetWeaver Planning Guide (<code>/solutions/sap/docs/netweaver-planning-guide#machine_types</code>).
<code>zone</code>	String	The zone in which you are deploying your SAP MaxDB. It must be in the same region that you selected for your subnetwork.
<code>subnetwork</code>	String	The name of the subnetwork that you created in a previous step. If you are deploying to a shared VPC, specify this value as <code>[SHAREDVPC_PROJECT]/[SUBNETWORK]</code> . For example, <code>myproject/network1</code> .
<code>windowsImage</code>	String	The name of the Windows operating-system image or image family that you are using with SAP MaxDB. To specify an image family, add the prefix <code>family/</code> to the family name. For example, <code>family/windows-2016</code> . To specify a specific image, specify only the image name. For the list of available image families, see the Images (<code>https://console.cloud.google.com/compute/images</code>) page in the Cloud console.
<code>windowsImageProjectString</code>		The Google Cloud project that contains the image you are going to use. This project might be your own project or a Google Cloud image project, such as <code>windows-cloud</code> . For a list of GCP image projects, see the Images (<code>/compute/docs/images</code>) page in the Compute Engine documentation.
<code>maxdbRootSize</code>	Integer	The size in GB of MaxDB (D:) , which is the root directory of the database instance. The minimum and default values for <code>maxdbRootSize</code> are both 8 GB.

Property	Data type	Description
maxdbDataSize	Integer	The size of MaxDB Data (E:) , which holds the database data files. The minimum and default values for maxdbDataSize are both 30 GB.
maxdbLogSize	Integer	The size of MaxDB Log (L:) , which holds the database transaction logs. The minimum and default values for maxdbLogSize are both 8 GB.
maxdbBackupSize	Integer	The size of the Backup (X:) volume. This property is optional. If set to 0 or omitted, no disk is created.
maxdbDataSSD	boolean	Specifies whether the data drive uses an SSD persistent disk (Yes) or an HDD persistent disk (No). Yes is the default.
maxdbLogSSD	boolean	Specifies whether the log drive uses an SSD persistent disk (Yes) or an HDD persistent disk (No). Yes is the default. SSD is recommended for the log drive.
usrsapSize	Integer	Required only if you are installing SAP MaxDB to run with SAP NetWeaver on the same VM instance. In the deployed VM, this volume is labeled SAP (S:) .
swapSize	Integer	Required only if you are installing SAP MaxDB to run with SAP NetWeaver on the same VM instance. In the deployed VM, this volume is labeled Pagefile (P:) .

The following configuration file creates a VM that is configured to run both the SAP MaxDB database server and SAP NetWeaver. The configuration file directs Deployment Manager to deploy an n1-standard-16 virtual that is running a Windows Server 2016 operating system. The VM includes all of the directories that are required to run SAP MaxDB with NetWeaver.

6. Deploy the VM:

The above command invokes the Deployment Manager, which deploys the VM, all according to the specifications in your `template.yaml` file. The process might take a few minutes. To check the progress of your deployment, follow the steps in the next section.

The following steps use Stackdriver Logging, which might incur charges. For more information, see [Stackdriver pricing \(/stackdriver/pricing_v2\)](/stackdriver/pricing_v2).

1. Open Stackdriver Logging to check for errors and monitor the progress of the installation.

[Go to Stackdriver Logging \(https://console.cloud.google.com/logs/viewer\)](https://console.cloud.google.com/logs/viewer)

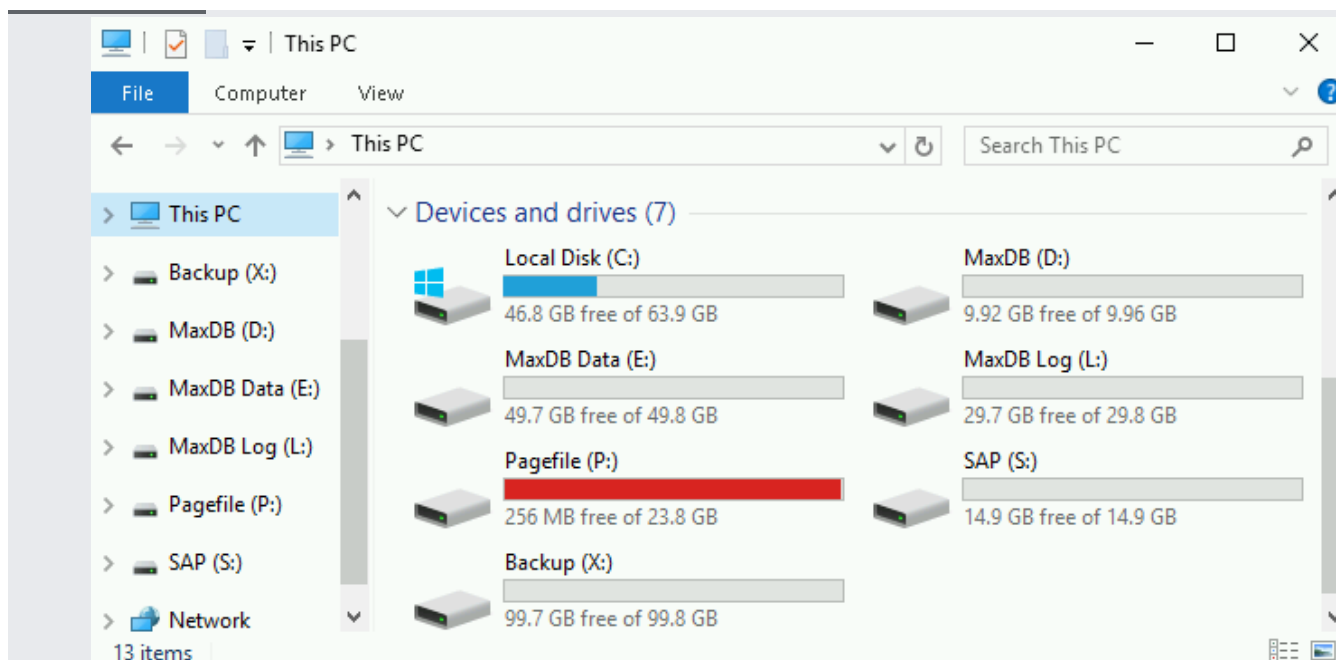
2. On the **Resources** tab, select **Global** as your logging resource. If **INSTANCE DEPLOYMENT COMPLETE** is displayed for a VM, Deployment Manager processing is complete for the VM.

```

▶ ⓘ 2018-08-08 15:52:11.854 PDT EX-VM-MDB-WIN Deployment '--- Creating D:\ (MaxDB)'
▶ ⓘ 2018-08-08 15:52:41.559 PDT EX-VM-MDB-WIN Deployment '--- Creating L:\ (MaxDB Log)'
▶ ⓘ 2018-08-08 15:52:45.703 PDT EX-VM-MDB-WIN Deployment '--- Creating E:\ (MaxDB Data)'
▶ ⓘ 2018-08-08 15:52:49.742 PDT EX-VM-MDB-WIN Deployment '--- Creating X:\ (Backup)'
▶ ⓘ 2018-08-08 15:52:56.362 PDT EX-VM-MDB-WIN Deployment '--- Creating S:\ (SAP)'
▶ ⓘ 2018-08-08 15:53:00.309 PDT EX-VM-MDB-WIN Deployment '--- Creating P:\ (Pagefile)'
▶ ⓘ 2018-08-08 15:53:04.665 PDT EX-VM-MDB-WIN Deployment 'Creating pagefile P:\pagefile.sys (24120 MB)'
▶ ⓘ 2018-08-08 15:53:05.943 PDT EX-VM-MDB-WIN Deployment 'INSTANCE DEPLOYMENT COMPLETE'

```

3. After the SAP MaxDB system deploys, connect to your VM as described in the next section, [Connecting to your VM \(#connecting_to_your_vm\)](#), to confirm that all of the required data volumes exist and that they have the expected capacities. The `Pagefile` drive should be almost full.



If any of the validation steps show that the installation failed:

1. Correct the error.
2. On the [Deployments](https://console.cloud.google.com/dm/deployments) page, delete the deployment to clean up the VMs and persistent disks from the failed installation.
3. Rerun your deployment.

To connect to a Windows-based VM, you must first generate a password for the VM. You can then connect to the VM using RDP or PowerShell.

After you create a Windows VM, you must generate a password for the VM before you can connect to it.

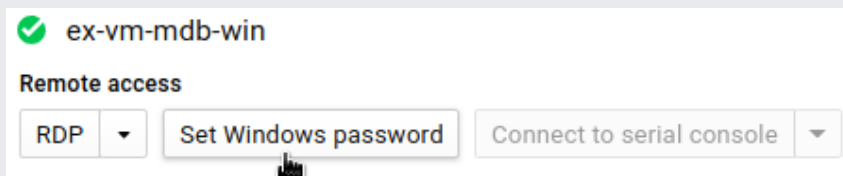
1. Go to the **VM instances** page.

[OPEN VM INSTANCES](https://console.cloud.google.com/compute/instances) (https://console.cloud.google.com/compute/instances)

2. Open the VM instance details page by clicking the name of the VM that you just created.

<input type="checkbox"/>	Name ^	Zone	Internal IP	External IP	Connect
<input type="checkbox"/>	<input checked="" type="checkbox"/> ex-vm-mdb-win	us-central1-f	10.1.0.7 (nic0)	104.155.137.195	RDP ▾ ⋮

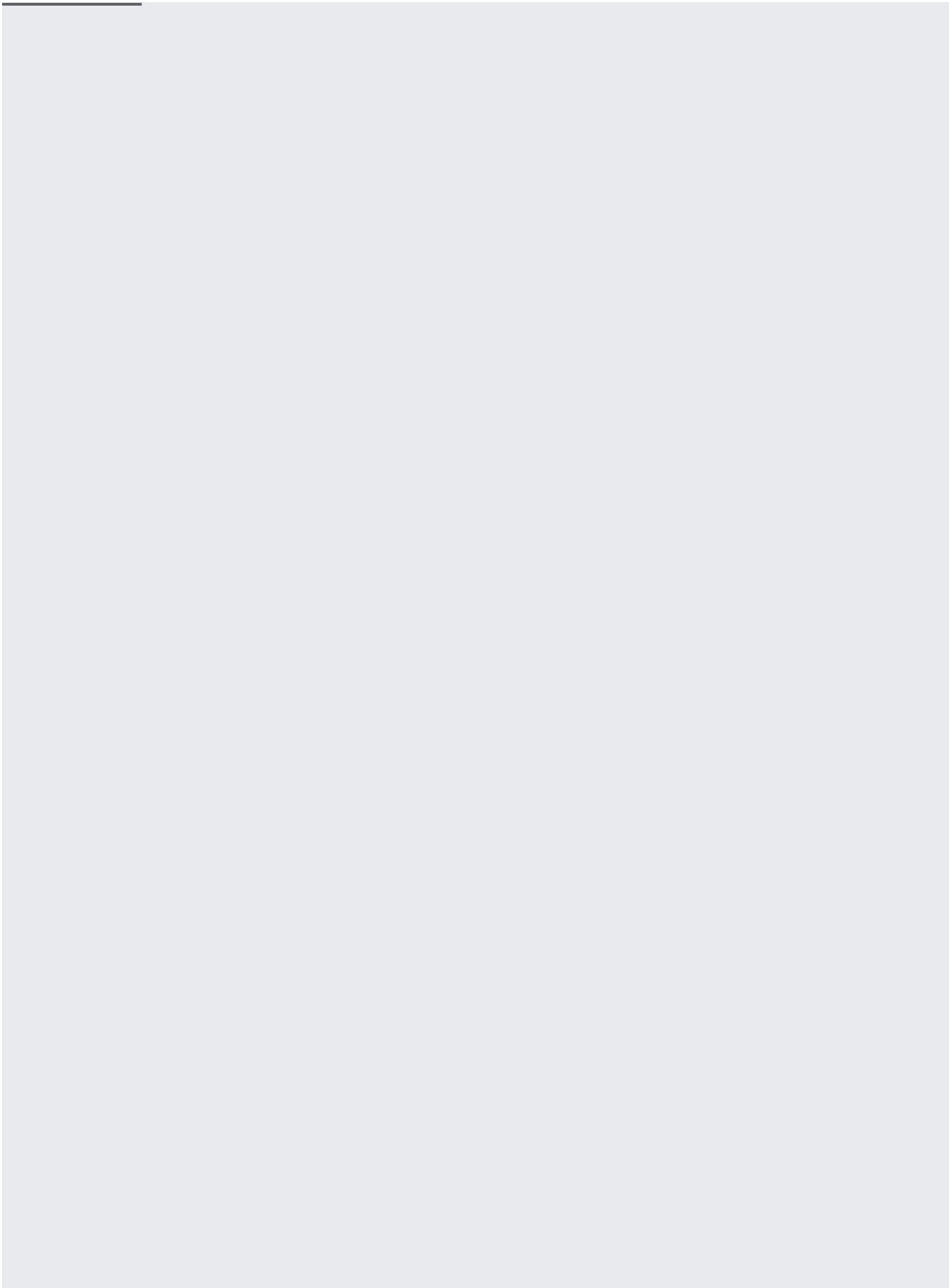
3. On the instance details page, click the **Set Windows Password** button. A password is generated for you. Copy the password and store it in a secure place.



If your firewall allows access on port 3389, you can use RDP to connect to a Windows instance and start a Remote Desktop session. To connect to your VM by using PowerShell, see [Using PowerShell \(#using_powershell\)](#).

Connect to the remote desktop on Windows instances using one of the following methods:

- Use a Chrome browser that has the [Chrome RDP for Google Cloud extension](https://chrome.google.com/webstore/detail/chrome-rdp-for-google-clo/mpbbnannobiobpfnblimoapbepghgikm/) installed and establish an RDP connection from the Cloud Console.
- Download the RDP file and manually connect through the Windows Remote Desktop Connection client or a third-party client.



1. If you have not created a username and password on the Windows VM yet, [create your Windows password](#) (#generating_passwords).
2. Add a firewall rule or edit your existing firewall rule to open port 5986 on the [GCP network](#) (/compute/docs/networking) where your Windows Server VM is located.
3. On your local workstation, open the PowerShell terminal.
4. Optionally, you can initialize a variable to hold your user credentials so you do not need to enter them each time you connect to the instance. If you skip this step, you receive a prompt for your username and password later.
5. Use the [Enter-PSSession command](#) (<https://msdn.microsoft.com/en-us/powershell/reference/5.1/microsoft.powershell.core/enter-pssession>) to start a remote PowerShell session and include the flags to use SSL and skip credentials checks.

Now that your operating system is configured, you can install your SAP MaxDB database. MaxDB is typically installed with the SAP product with which it will be used.

To install SAP MaxDB on your VM:

1. Use RDP or Windows PowerShell to connect to your Windows-based VM.
2. Download the SAP Software Provisioning Manager (SWPM), the SAP product installation media, and MaxDB installation Media according to SAP installation guides.
3. Install your SAP product and SAP MaxDB database according to the SAP installation guides for your SAP product. For additional guidance, see the [SAP MaxDB documentation](https://help.sap.com/viewer/9d62a6d2a95149718db6b65b7ffa2052/113/en-US) (<https://help.sap.com/viewer/9d62a6d2a95149718db6b65b7ffa2052/113/en-US>).

SAP provides additional installation information in [SAP Note 1020175 - FAQ: SAP MaxDB installation, upgrade or applying a patch](https://launchpad.support.sap.com/#/notes/0001020175) (<https://launchpad.support.sap.com/#/notes/0001020175>).

Before using your SAP MaxDB instance, we recommend that you perform the following post-deployment steps:

1. Update your SAP MaxDB software with the latest patches, if available.
2. Install any additional components.
3. Configure and back up your new SAP MaxDB database.

For more information, see [SAP MaxDB Database Administration](https://help.sap.com/viewer/2c2effc99b6746019aeb1af52ad59f5d/113/en-US/1a32c42a0fcf40b5b95ff7040a60aa41.html) (<https://help.sap.com/viewer/2c2effc99b6746019aeb1af52ad59f5d/113/en-US/1a32c42a0fcf40b5b95ff7040a60aa41.html>)