Installing GitHub Enterprise Server on AWS

In this article

To install GitHub Enterprise Server on Amazon Web Services (AWS), you must launch an Amazon Elastic Compute Cloud (EC2) instance and create and attach a separate Amazon Elastic Block Store (EBS) data volume.

Prerequisites

- You must have a GitHub Enterprise license file. For more information, see "Setting up a trial of GitHub Enterprise Server" and "About licenses for GitHub Enterprise."
- You must have an AWS account capable of launching EC2 instances and creating EBS volumes. For more information, see the Amazon Web Services website.
- Most actions needed to launch your GitHub Enterprise Server instance may also be performed using the AWS management console. However, we recommend installing the AWS command line interface (CLI) for initial setup. Examples using the AWS CLI are included below. For more information, see Amazon's guides Working with the AWS Management Console and What is the AWS Command Line Interface.

Note: At this time GitHub Enterprise Server does not support the use of the Amazon IMDSv2 Metadata API.

This guide assumes you are familiar with the following AWS concepts:

- Launching EC2 Instances
- Managing EBS Volumes
- Using Security Groups (For managing network access to your instance)
- Elastic IP Addresses (EIP) (Strongly recommended for production environments)
- EC2 and Virtual Private Cloud (If you plan to launch into a Virtual Private Cloud)
- AWS Pricing (For calculating and managing costs)

For a diagram that provides an architectural overview, see the "AWS Architecture Diagram for Deploying GitHub Enterprise Server."

This guide recommends the principle of least privilege when setting up your GitHub

Enterprise Server instance on AWS. For more information, refer to the AWS Identity and Access Management (IAM) documentation.

Hardware considerations

Minimum requirements

We recommend different hardware configurations depending on the number of user licenses for your GitHub Enterprise Server instance. If you provision more resources than the minimum requirements, your instance will perform and scale better.

User licenses	x86-64 vCPUs	Memory	Root storage	Attached (data) storage
Trial, demo, or 10 light users	4	32 GB	200 GB	150 GB
10 to 3,000	8	48 GB	200 GB	300 GB
3,000 to 5000	12	64 GB	200 GB	500 GB
5,000 to 8000	16	96 GB	200 GB	750 GB
8,000 to 10,000+	20	160 GB	200 GB	1000 GB

If you plan to enable GitHub Actions for the users of your instance, more resources are required.

For more information about these requirements, see "Getting started with GitHub Actions for GitHub Enterprise Server."

If you plan to enable Container registry for the users of your instance, more resources are required. For more information about these requirements, see "Getting started with GitHub Packages for your enterprise."

For more information about adjusting resources for an existing instance, see "Increasing storage capacity" and "Increasing CPU or memory resources."

Storage

We recommend a high-performance SSD with high input/output operations per second (IOPS) and low latency for GitHub Enterprise Server. Workloads are I/O intensive. If you use a bare metal hypervisor, we recommend directly attaching the disk or using a disk from a storage area network (SAN).

Your instance requires a persistent data disk separate from the root disk. For more information, see "System overview."

To configure GitHub Actions, you must provide external blob storage. For more information, see "Getting started with GitHub Actions for GitHub Enterprise Server."

The available space on the root filesystem will be 50% of the total disk size. You can resize your instance's root disk by building a new instance or using an existing instance. For more information, see "System overview" and "Increasing storage capacity."

CPU and memory

The CPU and memory resources that GitHub Enterprise Server requires depend on the levels of activity for users, automations, and integrations.

Any VMs you provision for your GitHub Enterprise Server instance must use the x86-64 CPU architecture. Other architectures are not supported, such as Aarch64 or arm64.

If you plan to enable GitHub Actions for the users of your GitHub Enterprise Server instance, you may need to provision additional CPU and memory resources for your instance. For more information, see "Getting started with GitHub Actions for GitHub Enterprise Server."

When you increase CPU resources, we recommend adding at least 6.5 GB of memory for each vCPU (up to 16 vCPUs) that you provision for the instance. When you use more than 16 vCPUs, you don't need to add 6.5 GB of memory for each vCPU, but you should monitor your instance to ensure it has enough memory.

> **Warning:** We recommend that users configure webhook events to notify external systems of activity on GitHub Enterprise Server. Automated checks for changes, or *polling*, will negatively impact the performance and scalability of your instance. For more information, see "About webhooks."

For more information about monitoring the capacity and performance of GitHub Enterprise Server, see "Monitoring your instance."

You can increase your instance's CPU or memory resources. For more information, see "Increasing CPU or memory resources."

Determining the instance type

Before launching your GitHub Enterprise Server instance on AWS, you'll need to determine the machine type that best fits the needs of your organization. To review the minimum requirements for GitHub Enterprise Server, see "Minimum requirements."

> **Note:** You can always scale up your CPU or memory by resizing your instance. However, because resizing your CPU or memory requires downtime for your users, we recommend over-provisioning resources to account for scale.

GitHub recommends a memory-optimized instance for GitHub Enterprise Server. For more information, see Amazon EC2 Instance Types on the Amazon EC2 website.

Selecting the GitHub Enterprise Server AMI

You can select an Amazon Machine Image (AMI) for GitHub Enterprise Server using the GitHub Enterprise Server portal or the AWS CLI.

AMIs for GitHub Enterprise Server are available in the AWS GovCloud (US-East and US-West) region. This allows US customers with specific regulatory requirements to run GitHub Enterprise Server in a federally compliant cloud environment. For more information on AWS's compliance with federal and other standards, see AWS's GovCloud (US) page and AWS's compliance page.

Using the GitHub Enterprise Server portal to select an AMI

1. Navigate to the image you want to use for your new instance.
    - Navigate to Release notes.

- In the right sidebar, click the version you want to download.
- Click **Download GitHub Enterprise Server X.X.X**.

2. Under "GitHub in the Cloud", select the "Select your platform" dropdown menu, and click **Amazon Web Services**.
3. Select the "Select your AWS region" drop-down menu, and click your desired region.
4. Take note of the AMI ID that is displayed.

Using the AWS CLI to select an AMI

1. Using the AWS CLI, get a list of GitHub Enterprise Server images published by GitHub's AWS owner IDs (`025577942450` for GovCloud, and `895557238572` for other regions). For more information, see describe-images in the AWS documentation.

```
aws ec2 describe-images \
--owners OWNER_ID \
--query 'sort_by(Images,&Name)[*].{Name:Name,ImageID:ImageId}' \
--output=text
```

2. Take note of the AMI ID for the latest GitHub Enterprise Server image.

Creating a security group

If you're setting up your AMI for the first time, you will need to create a security group and add a new security group rule for each port in the table below. For more information, see the AWS guide Using Security Groups.

1. Using the AWS CLI, create a new security group. For more information, see create-security-group in the AWS documentation.

```
aws ec2 create-security-group --group-name SECURITY_GROUP_NAME --description
"SECURITY GROUP DESCRIPTION"
```

2. Take note of the security group ID (`sg-xxxxxxxx`) of your newly created security group.
3. Create a security group rule for each of the ports in the table below. We recommend opening network ports selectively based on the network services you need to expose for administrative and user purposes. For more information, see "Network ports," and authorize-security-group-ingress in the AWS documentation.

```
aws ec2 authorize-security-group-ingress --group-id SECURITY_GROUP_ID --
protocol PROTOCOL --port PORT_NUMBER --cidr SOURCE IP RANGE
```

This table identifies what each port is used for.

Port	Service	Description
22	SSH	Git over SSH access. Clone, fetch, and push operations to public/private repositories supported.
25	SMTP	SMTP with encryption

25	SMTP	SMTP with encryption (STARTTLS) support.
80	HTTP	Web application access. *All requests are redirected to the HTTPS port when SSL is enabled.*
122	SSH	Instance shell access. *The default SSH port (22) is dedicated to application git+ssh network traffic.*
161/UDP	SNMP	Required for network monitoring protocol operation.
443	HTTPS	Web application and Git over HTTPS access.
1194/UDP	VPN	Secure replication network tunnel in high availability configuration.
8080	HTTP	Plain-text web based Management Console. *Not required unless SSL is disabled manually.*
8443	HTTPS	Secure web based Management Console. *Required for basic installation and configuration.*
9418	Git	Simple Git protocol port. Clone and fetch operations to public repositories only. *Unencrypted network communication.* If you have enabled private mode on your instance, then opening this port is only required if you also enabled anonymous Git read access. For more information, see "Enforcing repository management policies in your enterprise."

Creating the GitHub Enterprise Server instance

To create the instance, you'll need to launch an EC2 instance with your GitHub Enterprise Server AMI and attach an additional storage volume for your instance data. For more information, see "Hardware considerations."

Note: You can encrypt the data disk to gain an extra level of security and ensure that any data you write to your instance is protected. There is a slight performance impact when using encrypted disks. If you decide to encrypt your volume, we strongly recommend doing so **before** starting your instance for the first time. For more information, see the Amazon guide on EBS encryption.

Warning: If you decide to enable encryption after you've configured your instance, you will need to migrate your data to the encrypted volume, which will incur some downtime for your users.

Launching an EC2 instance

In the AWS CLI, launch an EC2 instance using your AMI and the security group you created. Attach a new block device to use as a storage volume for your instance data, and configure the size based on your user license count. For more information, see run-instances in the AWS documentation.

```
aws ec2 run-instances \
  --security-group-ids SECURITY_GROUP_ID \
  --instance-type INSTANCE_TYPE \
  --image-id AMI_ID \
  --block-device-mappings '[{"DeviceName":"/dev/xvdf","Ebs":
{"VolumeSize":SIZE,"VolumeType":"TYPE"}}]' \
  --region REGION \
  --ebs-optimized
```

Allocating an Elastic IP and associating it with the instance

If this is a production instance, we strongly recommend allocating an Elastic IP (EIP) and associating it with the instance before proceeding to GitHub Enterprise Server configuration. Otherwise, the public IP address of the instance will not be retained after instance restarts. For more information, see Allocating an Elastic IP Address and Associating an Elastic IP Address with a Running Instance in the Amazon documentation.

Both primary and replica instances should be assigned separate EIPs in production High Availability configurations. For more information, see "Configuring high availability."

Configuring the GitHub Enterprise Server instance

To configure the instance, you must upload a license file, set the root Management Console password, configure the instance's settings, and restart the instance.

Warning: To prevent an attacker from compromising the new instance, ensure that you personally set the root Management Console password and create the first user as soon as possible.

1. Copy the virtual machine's public DNS name, and paste it into a web browser.
2. At the prompt, upload your license file and set a management console password. For more information, see "Managing your license for GitHub Enterprise."
3. In the Management Console, configure and save your desired settings. For more information, see "Configuring GitHub Enterprise."
4. The instance will restart automatically.
5. Click **Visit your instance**.

Further reading

- "System overview"
- "About upgrades to new releases"