



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Introduction to CI/CD and Cloud Computing

# Content

Introduction to CI/CD

Introduction to Cloud Computing

Introduction to DigitalOcean

# Introduction to CI/CD



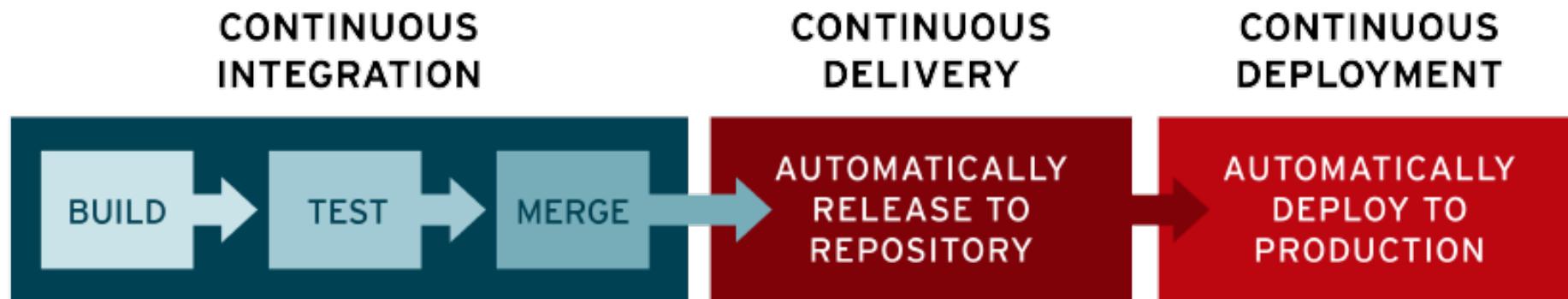
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# CI/CD Overview

- CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development.
- The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.
- CI/CD is a solution to the problems integrating new code can cause for development and operations teams (AKA "integration hell").

# CI/CD Overview (cont.)

- Specifically, CI/CD introduces ongoing automation and continuous monitoring throughout the lifecycle of apps, from integration and testing phases to delivery and deployment.
- Taken together, these connected practices are often referred to as a "CI/CD pipeline".



# CI and CD (and the other CD)?

- The acronym CI/CD has a few different meanings.
- The "CI" in CI/CD always refers to continuous integration, which is an automation process for developers.
- Successful CI means new code changes to an app are regularly built, tested, and merged to a shared repository.
- It's a solution to the problem of having too many branches of an app in development at once that might conflict with each other.

# CI and CD (and the other CD)?

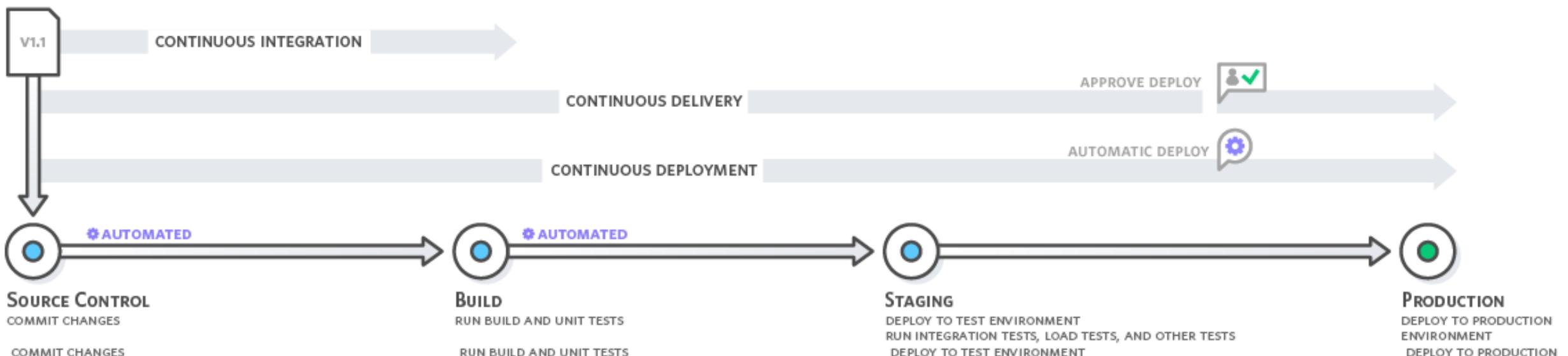
- The "CD" in CI/CD refers to continuous delivery and/or continuous deployment, which are related concepts that sometimes get used interchangeably.
- Both are about automating further stages of the pipeline, but they're sometimes used separately to illustrate just how much automation is happening.

# CI and CD (and the other CD)?

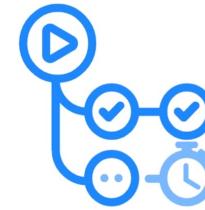
- With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment.
- The purpose of continuous delivery is to ensure that it takes minimal effort to deploy new code.

# CI and CD (and the other CD)?

- Continuous deployment (the other possible "CD") refers to automatically releasing a developer's changes from the repository to production
- The difference between continuous delivery and continuous deployment is the presence of a manual approval to update to production. With continuous deployment, production happens automatically without explicit approval.

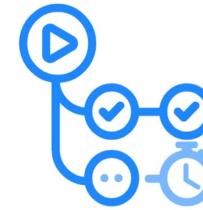


# GitHub Actions Overview



- GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.
- You can create workflows that build and test every pull request to your repository, or deploy merged pull requests to production.

# GitHub Actions Overview

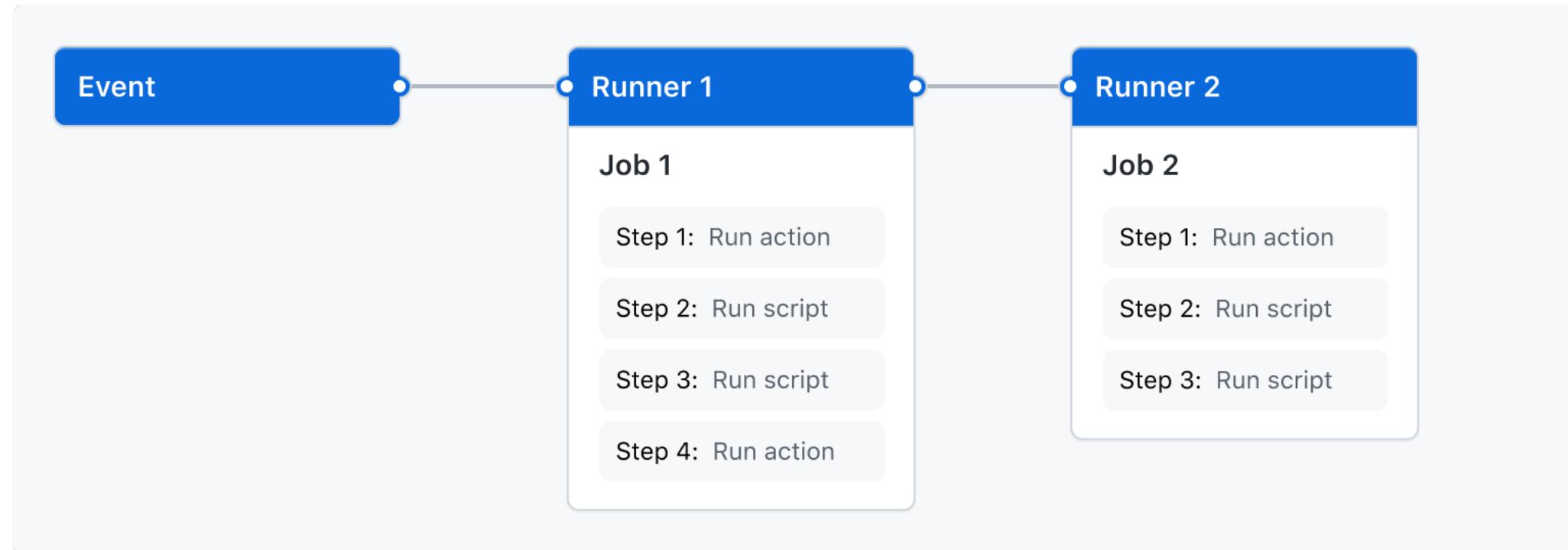


- GitHub Actions goes beyond just DevOps and lets you run workflows when other events happen in your repository.
- GitHub provides Linux, Windows, and macOS virtual machines to run your workflows, or you can host your own self-hosted runners in your own data center or cloud infrastructure.

# The components of GitHub Actions

- You can configure a GitHub Actions workflow to be triggered when an event occurs in your repository, such as a pull request being opened or an issue being created.
- Your workflow contains one or more jobs which can run in sequential order or in parallel.
  - Each job will run inside its own virtual machine runner, or inside a container, and has one or more steps that either run a script that you define or run an action, which is a reusable extension that can simplify your workflow.

# The components of GitHub Actions



# Workflows

- A workflow is a configurable automated process that will run one or more jobs.
- Workflows are defined by a YAML file checked into your repository and will run when triggered by an event in your repository, or they can be triggered manually, or at a defined schedule.
- Workflows are defined in the .github/workflows directory in a repository, and a repository can have multiple workflows, each of which can perform a different set of tasks.



# Events

- An event is a specific activity in a repository that triggers a workflow run.
- For example, activity can originate from GitHub when someone creates a pull request, opens an issue, or pushes a commit to a repository.
- You can also trigger a workflow run on a schedule, by posting to a REST API, or manually.

# Events

## Running your workflow only when a push to specific branches occurs

You can use the `branches` or `branches-ignore` filter to configure your workflow to only run when specific branches are pushed. For more information, see "[Workflow syntax for GitHub Actions](#)."

For example, this workflow will run when someone pushes to `main` or to a branch that starts with `releases/`.

```
on:  
  push:  
    branches:  
      - 'main'  
      - 'releases/**'
```

**Note:** If you use both the `branches` filter and the `paths` filter, the workflow will only run when both filters are satisfied. For example, the following workflow will only run when a push that includes a change to a JavaScript (`.js`) file is made to a branch whose name starts with `releases/`:

```
on:  
  push:  
    branches:  
      - 'releases/**'  
    paths:  
      - '**.js'
```

## Running your workflow only when a push of specific tags occurs

You can use the `tags` or `tags-ignore` filter to configure your workflow to only run when specific tags are pushed. For more information, see "[Workflow syntax for GitHub Actions](#)."

For example, this workflow will run when someone pushes a tag that starts with `v1.`.

```
on:  
  push:  
    tags:  
      - v1.**
```

## Available events

branch\_protection\_rule  
check\_run  
check\_suite  
create  
delete  
deployment  
deployment\_status  
discussion  
discussion\_comment  
fork  
gollum  
issue\_comment  
issues  
label  
merge\_group  
milestone  
page\_build  
project  
project\_card  
project\_column  
public  
pull\_request  
pull\_request\_comment (use issue\_comment)  
pull\_request\_review  
pull\_request\_review\_comment  
pull\_request\_target  
push

# Jobs

- A job is a set of steps in a workflow that execute on the same runner.
- Each step is either a shell script that will be executed, or an action that will be run.
- Steps are executed in order and are dependent on each other. Since each step is executed on the same runner, you can share data from one step to another.
- For example, you can have a step that builds your application followed by a step that tests the application that was built.

# Jobs

- You can configure a job's dependencies with other jobs;
- by default, jobs have no dependencies and run in parallel with each other.
- When a job takes a dependency on another job, it will wait for the dependent job to complete before it can run.
- For example, you may have multiple build jobs for different architectures that have no dependencies, and a packaging job that is dependent on those jobs. The build jobs will run in parallel, and when they have all completed successfully, the packaging job will run.

# Jobs

```
jobs:
  build-and-push:
    name: Build
    runs-on: ubuntu-latest
    steps:
      - name: Checkout files
        uses: actions/checkout@v2
      - name: Set up QEMU
        uses: docker/setup-qemu-action@v1
      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v1
      - name: Login to DockerHub
        uses: docker/login-action@v1
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}
      - name: Build and push
        uses: docker/build-push-action@v2
        with:
          context: .
          file: ./Dockerfile
          push: true
          tags: ${{ secrets.DOCKERHUB_USERNAME }}/${{ secrets.DOCKERHUB_REPO_NAME }}:staging

  pull-and-up:
    name: Pull new image and re run with the new version
    runs-on: ubuntu-latest
    needs: build-and-push
    steps:
      - uses: appleboy/ssh-action@master
        with:
          host: ${{ secrets.STAGING_HOST }}
          username: ${{ secrets.USERNAME }}
          key: ${{ secrets.PRIVATE_KEY }}
          script: |
            cd ${{ secrets.STAGING_PATH }}
            docker-compose pull && docker-compose down && docker-compose up -d
```



# Actions

- An action is a custom application for the GitHub Actions platform that performs a complex but frequently repeated task.
- Use an action to help reduce the amount of repetitive code that you write in your workflow files.
- An action can pull your git repository from GitHub, set up the correct toolchain for your build environment, or set up the authentication to your cloud provider.

# Actions Marketplace

Marketplace / Search results

Types

Search for apps and actions

Sort: Best Match ▾

Apps

Actions

An entirely new way to automate your development workflow.

16482 results filtered by Actions

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Learning

Localization

Mobile

Monitoring

Project management

Publishing

Recently added

Security

**Actions**

An entirely new way to automate your development workflow.

16482 results filtered by Actions

**Actions**

**First interaction**  
By actions Greet new contributors when they create their first issue or open their first pull request  
★ 490 stars

**Download a Build Artifact**  
By actions Download a build artifact that was previously uploaded in the workflow by the upload-artifact action  
★ 750 stars

**Upload a Build Artifact**  
By actions Upload a build artifact that can be used by subsequent workflow steps  
★ 1.9k stars

**Setup Node.js environment**  
By actions Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH  
★ 2.4k stars

**Setup .NET Core SDK**  
By actions Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository  
★ 608 stars

**Setup Go environment**  
By actions Setup a Go environment and add it to the PATH  
★ 925 stars

**Setup Java JDK**  
By actions Set up a specific version of the Java JDK and add the command-line tools to the PATH  
★ 952 stars

**Close Stale Issues**  
By actions Close issues and pull requests with no recent activity  
★ 858 stars



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Runners

- A runner is a server that runs your workflows when they're triggered.
- Each runner can run a single job at a time. GitHub provides Ubuntu Linux, Microsoft Windows, and macOS runners to run your workflows;
- Each workflow run executes in a fresh, newly-provisioned virtual machine.

# Runners

- GitHub also offers larger runners, which are available in larger configurations.
- If you need a different operating system or require a specific hardware configuration, you can host your own runners.

The screenshot shows the GitHub Settings interface. At the top, there are navigation links: Pull requests, Actions, Projects, Security, Insights, and Settings (which is underlined). Below the navigation, there are several sections: General, Runners, and Security. The General section includes Access, Collaborators and teams, and Code and automation (with sub-options for Branches, Tags, and Actions). The Runners section is titled "Runners" and contains the message "Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. [Learn more about self-hosted runners.](#)". It also features a button labeled "New self-hosted runner". The Security section includes Code security and analysis, Deploy keys, and Secrets. The bottom of the page has an Integrations section with GitHub apps and Email notifications.

# GitHub Actions example

- GitHub Actions uses YAML syntax to define the workflow. Each workflow is stored as a separate YAML file in your code repository, in a directory named .github/workflows.
- You can create an example workflow in your repository that automatically triggers a series of commands whenever code is pushed.
- In this workflow, GitHub Actions checks out the pushed code, installs the bats (Bash Automated Testing System) testing framework, and runs a basic command to output the bats version: bats -v.

# GitHub Actions example

1. In your repository, create the .github/workflows/ directory to store your workflow files.
2. In the .github/workflows/ directory, create a new file called learn-github-actions.yml and add the following code.
3. Commit these changes and push them to your GitHub repository.

Sample repo: <https://github.com/BKCLab/ExpressExampleProject>

# GitHub Actions example

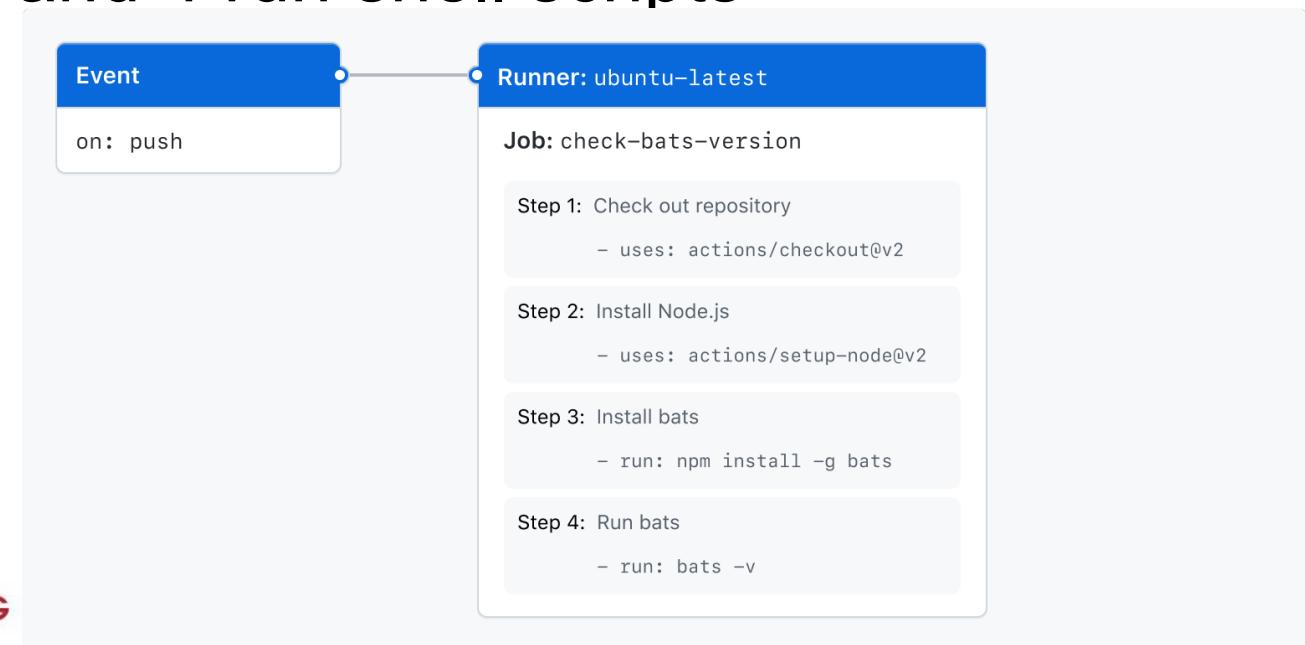
YAML



```
name: learn-github-actions
run-name: ${{ github.actor }} is learning GitHub Actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

# Visualizing the workflow file

- In this diagram, you can see the workflow file you just created and how the GitHub Actions components are organized in a hierarchy.
- Each step executes a single action or shell script. Steps 1 and 2 run actions, while steps 3 and 4 run shell scripts



# View the results of each step.

The screenshot shows a GitHub Actions job named "check-bats-version" which succeeded 8 minutes ago in 9 seconds. The job consists of several steps:

- > ✓ Set up job (3s)
- > ✓ Run actions/checkout@v2 (1s)
- > ✓ Run actions/setup-node@v1 (2s)
- > ✓ Run npm install -g bats (2s)
- > ✓ Run bats -v (0s)
  - 1 ► Run bats -v
  - 4 Bats 1.2.1
- > ✓ Post Run actions/checkout@v2 (1s)
- > ✓ Complete job (0s)

A search bar labeled "Search logs" and a gear icon for settings are visible at the top right of the interface.

# Introduction to Cloud Computing



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Cloud computing

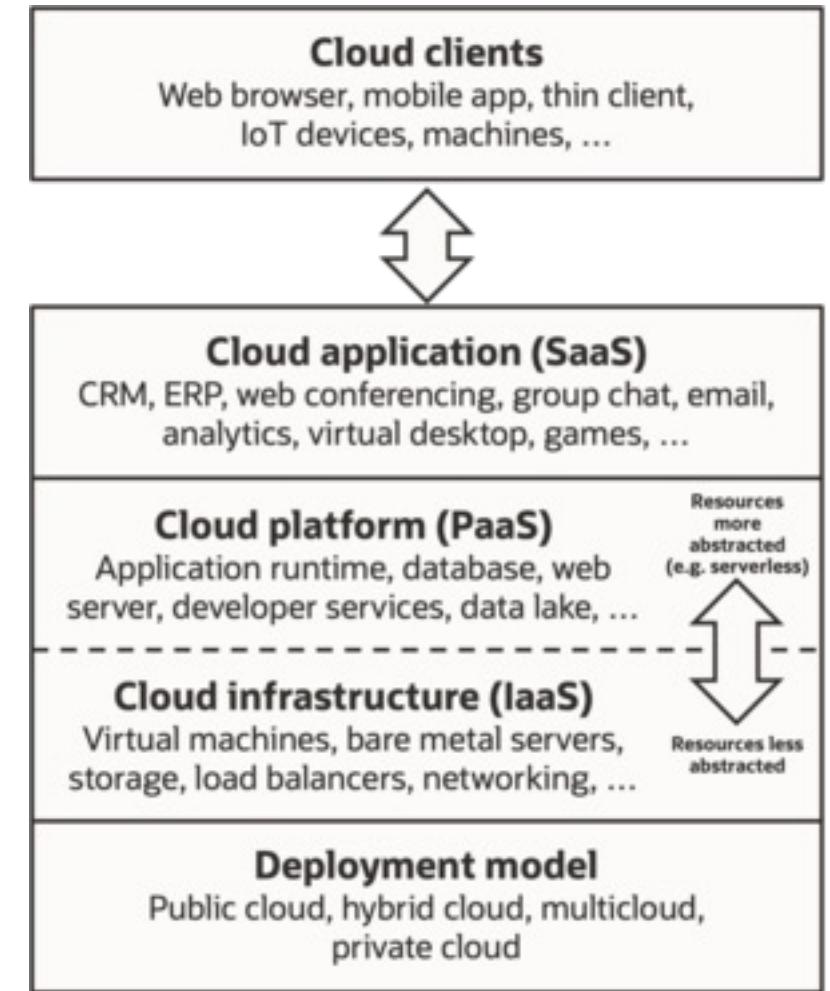
- Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user.
- Large clouds often have functions distributed over multiple locations, each of which is a data center.
- Cloud computing relies on sharing of resources to achieve coherence and typically uses a "pay as you go" model, which can help in reducing capital expenses but may also lead to unexpected operating expenses for users.

# Cloud computing: Value proposition

- Advocates of public and hybrid clouds claim that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs.
- Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand, providing burst computing capability: high computing power at certain periods of peak demand.

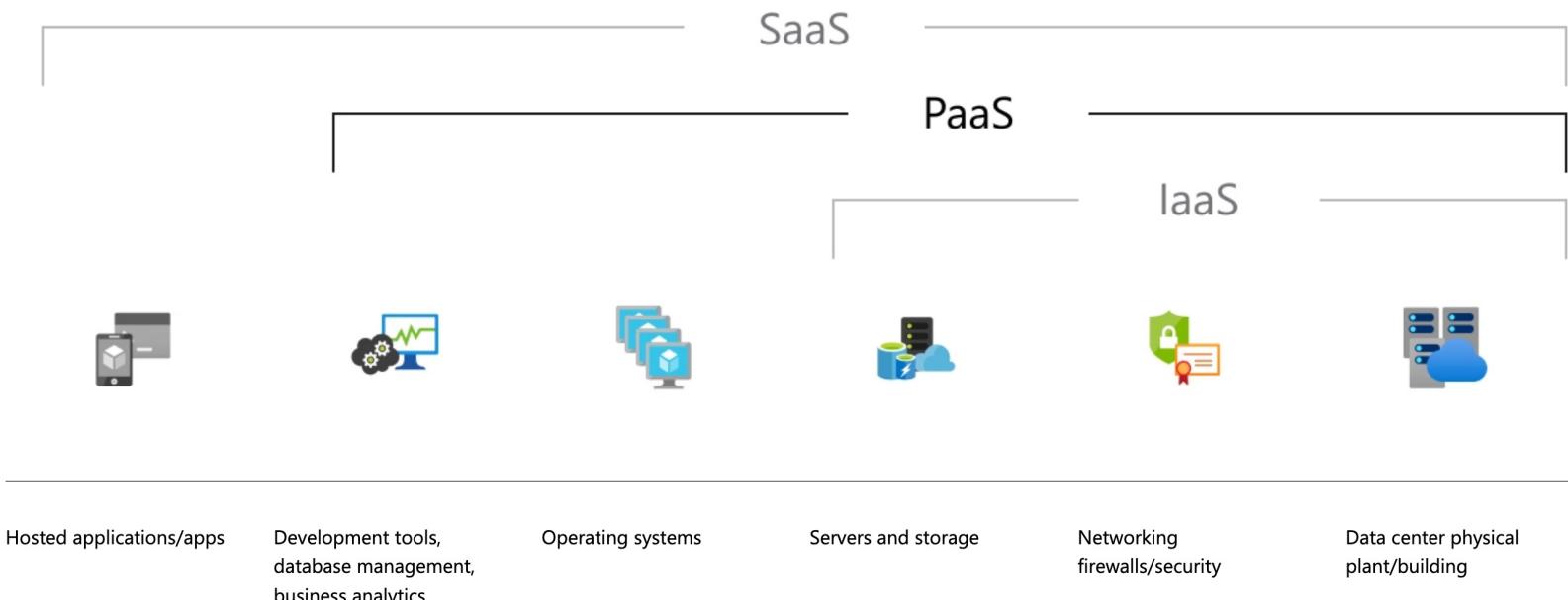
# Cloud computing: Service models

- Though service-oriented architecture advocates "Everything as a service."
- Often portrayed as layers in a stack: infrastructure-, platform- and software-as-a-service.



# Infrastructure as a service (IaaS)

- refers to online services that provide high-level APIs used to abstract various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup, etc.



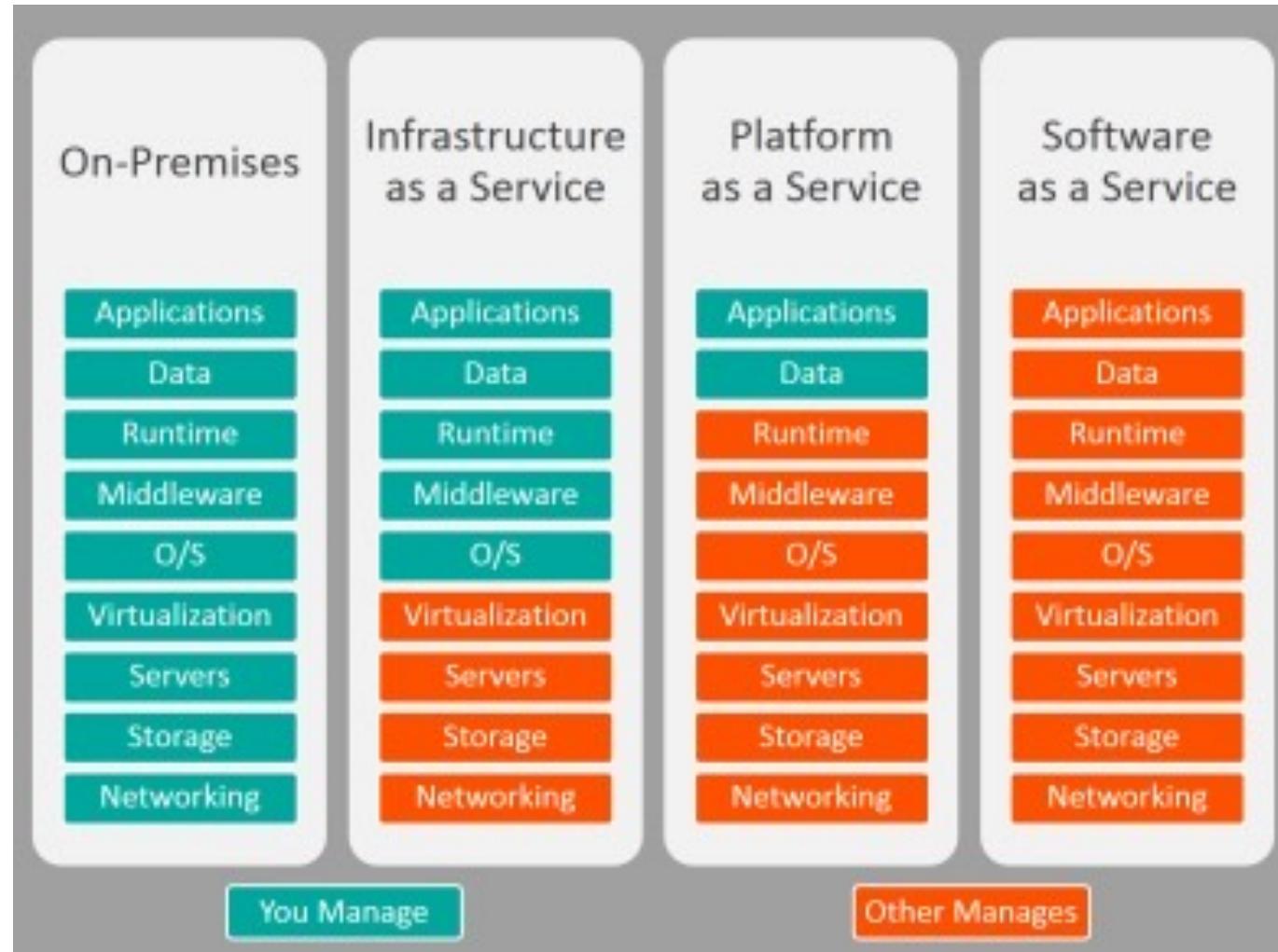
# Platform as a service (PaaS)

- PaaS provider hosts everything e.g., servers, networks, storage, operating system software, databases, development tool at their data center
- Typically customers can pay a fixed fee to provide a specified amount of resources for a specified number of users, or they can choose 'pay-as-you-go' pricing to pay only for the resources they use.

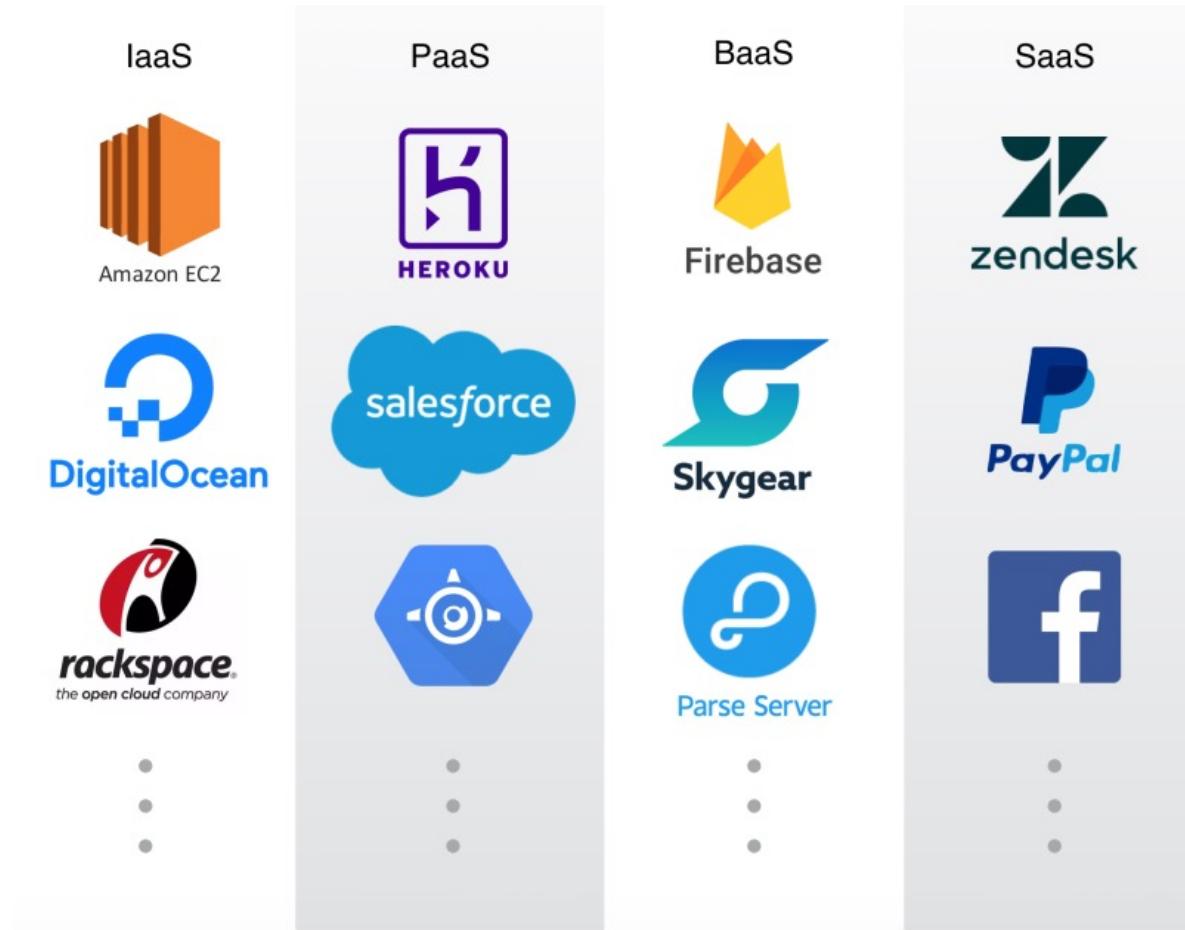
# Software as a service (SaaS)

- Users gain access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis or using a subscription fee.
- In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs.

# IaaS, PaaS, SaaS



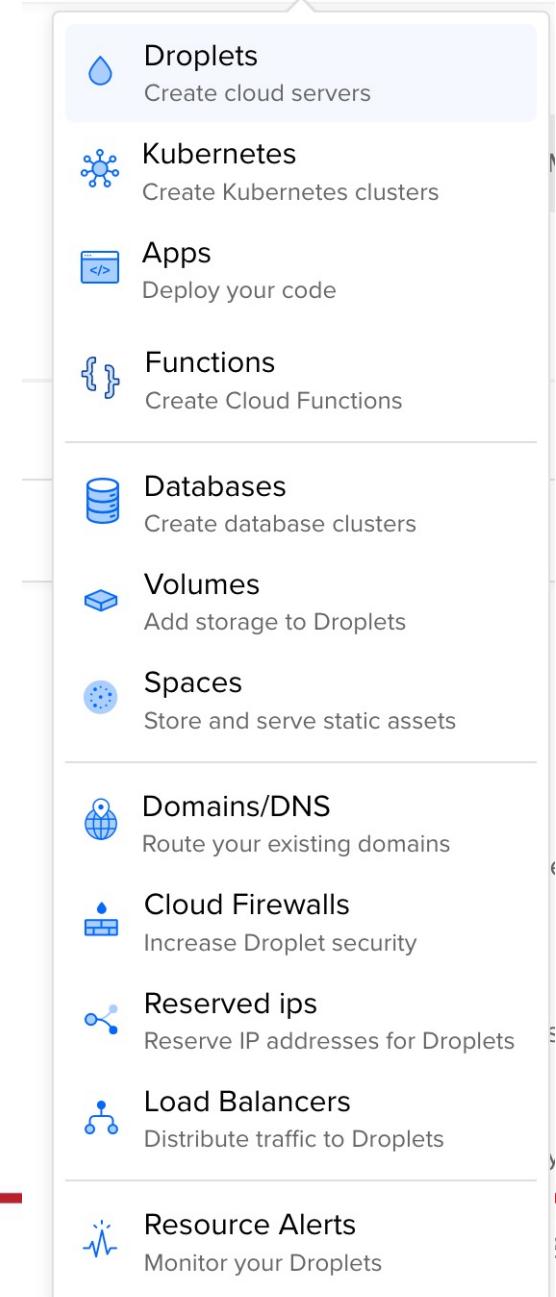
# IaaS, PaaS, SaaS



# Introduction to DigitalOcean

# D.O: Create resources:

- Computing Engine:
  - Droplets
  - k8s
  - Apps
  - Functions
- Storage:
  - Databases
  - Volumes
  - Spaces
- Networking: ...

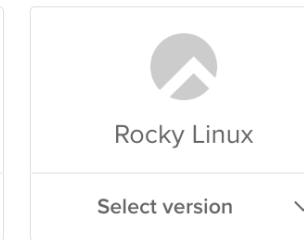
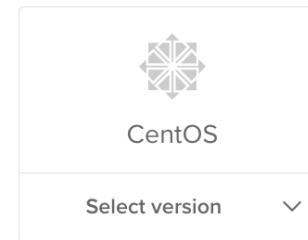
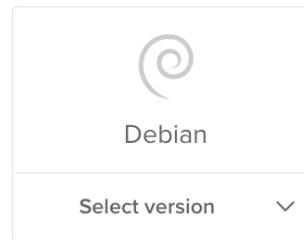
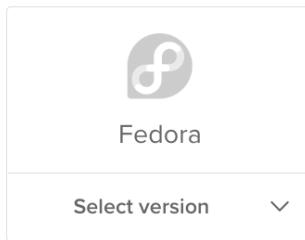


# Droplets: Choose OS

Create Droplets

Choose an image ?

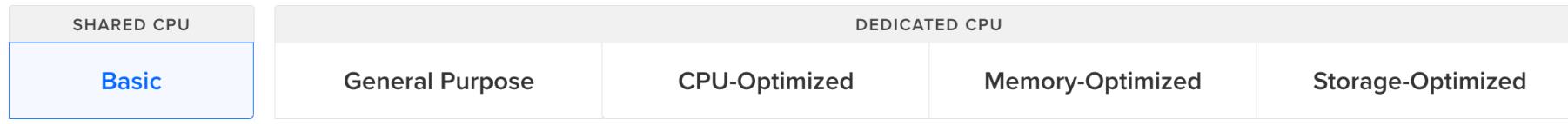
[Distributions](#) [Marketplace](#) [Snapshots](#) [Backups](#) [Custom images](#)



# Droplets: Choose Plan: CPU, RAM, Disk

Choose a plan

[Help me choose ↗](#)



Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options

<input checked="" type="radio"/> <b>Regular</b> Disk type: SSD	<input type="radio"/> <b>Premium Intel</b> Disk: NVMe SSD <small>NEW</small>	<input type="radio"/> <b>Premium AMD</b> Disk: NVMe SSD <small>NEW</small>
\$6/mo \$0.009/hour  1 GB / 1 CPU 25 GB SSD Disk 1000 GB transfer	\$12/mo \$0.018/hour  2 GB / 1 CPU 50 GB SSD Disk 2 TB transfer	\$18/mo \$0.027/hour  2 GB / 2 CPUs 60 GB SSD Disk 3 TB transfer
\$24/mo \$0.036/hour  4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer	\$48/mo \$0.071/hour  8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer	\$96/mo \$0.143/hour  16 GB / 8 CPUs 320 GB SSD Disk 6 TB transfer

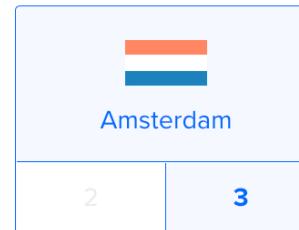
[Show all plans](#)

# D.O: Droplets: Choose region

Add block storage ?

[Add Volume](#)

Choose a datacenter region



# Droplets: Management & Monitoring

Graphs

Access

Power

Volumes

Resize

Networking

Backups

Snapshots

Kernel

History

Destroy

Tags

Recovery



ubuntu-s-2vcpu-4gb-sgp1-01

in Demo / 4 GB Memory / 80 GB Disk / SGP1 - Ubuntu 22.10 x64

ON

Graphs  
Access  
Power  
Volumes  
Resize  
Networking  
Backups  
Snapshots  
Kernel  
History  
Destroy  
Tags  
Recovery

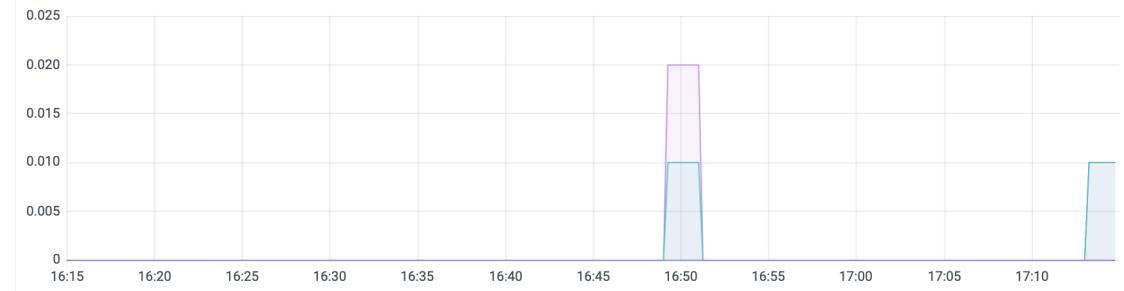
Graphs

Select period  
1 hour

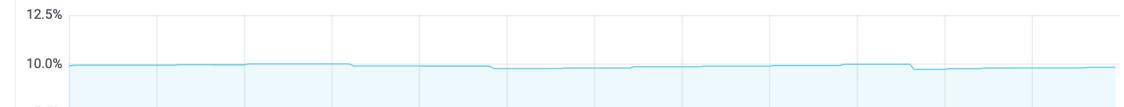
CPU %



Load (1/5/15)



Memory



# k8s: choose region & version

## Create a Kubernetes cluster

### Choose a datacenter region

Your Kubernetes cluster will be located in a single datacenter.



### VPC network

default-sfo3 DEFAULT

Each datacenter has a default VPC network [↗](#) that generates a Private IP range for you so resources in the same network can communicate securely.

Need to configure your own IP range? [Create a custom VPC ↗](#)

### Select a version

Select the Kubernetes version. The newest available version is selected by default.



**Tip:** We generally recommend the latest version unless your team has a specific need.  
[See the DigitalOcean Kubernetes release notes ↗](#)

# k8s: how many nodes

## Choose cluster capacity [?](#)

Select a plan that best suits your workload type. We can help you choose the right sizing approach [?](#) for overall availability and performance. You can add or remove nodes and node pools at any time.

### Select a scaling type

Fixed size  Autoscale [?](#)

Best for predictable workloads. Manually add and remove nodes.

Node pool name

Machine type (Droplet) [?](#)

Basic nodes  
Variable ratio of memory per shared CPU



Node plan [?](#)

\$24/month per node (\$0.036/hour)  
2.5 GB RAM usable (4 GB Total) / 2 vCPUs

Nodes

— 3 +

Cost of all nodes: \$72/month (\$0.11/hour)

Add Another Node Pool



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# k8s: creating...

← Back to Clusters



k8s-1-25-4-do-0-sgp1-1672735123935  
in Demo / SGP1 - 1.25.4-do.0

Overview

Resources

Insights

Marketplace

Settings

Actions ^

Download Config

View Nodes

Edit Settings

Getting Started

Destroy

## Getting Started with Kubernetes



Create a Kubernetes cluster

2

Connecting to Kubernetes

3

Verify connectivity

4

Deploy a workload

### Let's get you set up!

Authenticate your cluster, verify connectivity, and deploy your first workload.



Get Started

# k8s: more info

**CONFIGURATION**

CPU	Memory	Disk
<b>6 vCPUs</b>	<b>12 GB</b>	<b>240 GB</b>

VPC network  
 **default-sgp1** 10.104.0.0/20

[Download Config File](#)

You can download the configuration file to authenticate the cluster after it's done provisioning.

**TOTAL CLUSTER COST**

**\$72/month**

Cost is estimated and does not include other DigitalOcean resources added to the cluster.

**CONTROL PLANE**

High availability  
X Not running on a [high availability control plane](#)

Add [high availability](#) for \$40/month to create multiple replicas of control plane components. This [cannot be disabled](#).

**CLUSTER ID**

8eb36bf4-5937-4756-aba8-fe7c6af47401 

 **What's new**

Run your cluster on a high availability control plane  to add extra reliability to critical workloads for \$40/month.

[Add high availability](#)

**NODE POOL STATUS**

 **Provisioning**

 **web-tech-demo** 0/3 nodes provisioned

[View resource details](#)

**RECOMMENDED TOPICS**

[Settings limits and requests](#)   
[Set up CI/CD using GitHub Actions](#)   
[Using Horizontal Pod Autoscaling](#)   
[Automating Deployments](#) 



**VERSION**

Current version	Upgrade window
<b>Kubernetes 1.25.4-do.0</b>	<b>Any day after 5PM - (GMT+7)</b>
<a href="#">View changelog</a> 	<a href="#">Modify upgrade window</a>

Only required [upgrades](#)  will install automatically.

# k8s: download config file to connect

**CONFIGURATION**

CPU	Memory	Disk
<b>6 vCPUs</b>	<b>12 GB</b>	<b>240 GB</b>

VPC network

 **default-sgp1** 10.104.0.0/20

**Download Config File ↓**

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data:
    LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURKekNDQWcrZ0F3SUJBZ0lDQm5Vd0RRWUpLb1Ja
    WMzUmxjaUJEUVRBZUZ3MH1NekF4TURNdwpPRE01TlRGYUZ3MDBnekF4TURNd09ETTV0VEZhTURNeEZUQV
    BHQ1NxR1NJYjNEUUVCQVFVQUE0SUIKRHdBd2dnRutBb0lCQVFDNNF1VnpWNkFnMzJBwVBCMHLWHV6Yjh
    HK3ZkNHHhwakNBVkwra3hzY01NCktSOHBLdU1UTGtzTrNERUVvaldzWjBneGhsSU85SWZvS1JFazM0MWI5
    dEF1ZE1laU1Wckc5d05CdQpoYUQ5bW0wYS9RbmdjTVpZcnZQnZZQS93WHBTektNckU5NFRXeFBFnzRFU
    EdBMVVkRHdFQi93UUUKQXdJQmhqQVNcZ05WSFJNQkFm0EVDRFHQVFIL0FnRUFNQjBHQTfVZERnUVdCQ1
    doRDRuTkNRSLRICjhmNlR0anhDSmtwHJOSHvruLI3TmFNb3hZekh4b2VhYTFTSU0xUENoY0pleVp5eUd
    DczdqQU5FUQowRTVDRWEyZVRycHgxWFB1bjhuVWNhcElDQjh6VmvdVbTNheUgxQXhaWFY5b0FZTTBj0WN1
    RWZyNWEKYStmanBCL0ozQTgrd0d3cVVxTUhIYVVtbW1lQnlxc203YVRvNET2YWVrR1Bp0WppMVo4YUJxT
    server: https://8eb36bf4-5937-4756-aba8-fe7c6af47401.k8s.ondigitalocean.com
    name: do-sgp1-k8s-1-25-4-do-0-sgp1-1672735123935
  contexts:
  - context:
    cluster: do-sgp1-k8s-1-25-4-do-0-sgp1-1672735123935
    user: do-sgp1-k8s-1-25-4-do-0-sgp1-1672735123935-admin
    name: do-sgp1-k8s-1-25-4-do-0-sgp1-1672735123935
    current-context: do-sgp1-k8s-1-25-4-do-0-sgp1-1672735123935
    kind: Config
    preferences: {}
  users:
  - name: do-sgp1-k8s-1-25-4-do-0-sgp1-1672735123935-admin
    user:
      token: dop_v1_97a9df909418543f3613a50ed788d8227da02df16574d9441386321bdc46f732
```

# k8s: run command to setup kubectl

The screenshot shows a step-by-step guide for setting up a Kubernetes cluster. Step 1, 'Create a Kubernetes cluster', is completed with a green checkmark. Step 2, 'Connecting to Kubernetes', is currently selected and highlighted in blue. Step 3, 'Verify connectivity', and Step 4, 'Deploy a workload', are listed below. To the right, there's a section titled 'Connecting and managing this cluster' which recommends using the Kubernetes official client and doctl. It includes tabs for 'Automated (recommended)' and 'Manual'. Below this, a terminal window shows the command 'doctl kubernetes cluster kubeconfig save 8eb36bf4-5937-4756-aba8-fe7c6af47401' being run, with its output indicating successful configuration. A link for 'Expected output' is provided. At the bottom, there's a 'Having trouble installing or connecting?' link and a 'Continue' button.

1 Create a Kubernetes cluster

2 Connecting to Kubernetes

3 Verify connectivity

4 Deploy a workload

Connecting and managing this cluster

We recommend using [Kubernetes official client](#) and DigitalOcean's command-line tool, [doctl](#), to interact with and manage clusters. Next, you will need to add an authentication token or certificate to your kubectl configuration file.

Automated (recommended) Manual

This approach automatically renews your cluster's certificate. Run the command below to authenticate:

```
doctl kubernetes cluster kubeconfig save 8eb36bf4-5937-4756-aba8-fe7c6af47401
```

Expected output

```
Notice: Adding cluster credentials to kubeconfig file found in "/Users/<username>/.kube".  
Notice: Setting current-context to k8s-1-25-4-do-0-sgp1-1672735123935
```

Having trouble installing or connecting?

Continue

# k8s: verify connection

The screenshot shows a step-by-step guide for setting up a Kubernetes cluster. The left sidebar lists four steps:

- 1 Create a Kubernetes cluster (done)
- 2 Connecting to Kubernetes (done)
- 3 Verify connectivity (in progress)
- 4 Deploy a workload (not started)

The main content area is titled "Verify cluster connectivity". It includes a note: "To verify connectivity to your cluster, run the series of commands listed below." Below this are two tabs: "kubectl commands" (selected) and "doctl commands". A table lists several kubectl commands with their descriptions:

Command	Description
<code>kubectl config get-contexts</code>	Lists your cluster name, user, and namespace
<code>kubectl cluster-info</code>	Display addresses of the control plane and cluster services
<code>kubectl version</code>	Display the client and server k8s version
<code>kubectl get nodes</code>	List all nodes created in the cluster
<code>kubectl help</code>	Displays commands that help manage your cluster

Below the table, there is a note: "Check out the below resources on kubectl (which allows you to manage your workloads), and doctl (which allows you to manage your cluster configuration)."

[Kubectl's official documentation ↗](#)  
[Doctl's official documentation ↗](#)

[Continue](#)

# k8s: Ready to deploy workload

- ✓ Create a Kubernetes cluster
- ✓ Connecting to Kubernetes
- ✓ Verify connectivity
- 4 Deploy a workload

## Deploy a workload or sample app

Check out the tutorials below that are designed to help you become familiar with Kubernetes and run a production-ready cluster.

### Build your Kubernetes confidence

We created a collection of sample applications on GitHub that can be used in various projects.

[Explore sample apps ➔](#)



- [document icon] [Deploy an image using Python ➔](#)
- [document icon] [Use Helm to set up WordPress ➔](#)
- [document icon] [Build a CI/CD pipeline and deploy serverless applications ➔](#)
- [document icon] [Migrate a load balancer ➔](#)

For more tutorials, visit the [Kubernetes resource center ➔](#)

You can always access these instructions from the Actions menu.

[Great, I'm done](#)

# k8s: Dashboard



Workloads

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses
- Ingress Classes
- Services

Config and Storage

- Config Maps
- Persistent Volume Claims
- Secrets
- Storage Classes

Cluster

- Cluster Role Bindings
- Cluster Roles
- Events

[cloud.digitalocean.com/kubernetes/clusters/8eb36bf4-5937-4756-aba8-fe7c6af47401/db/0057255e0f48be6fda8b159c3ab40c6505407b45/#/overview?namespace=\\_all](https://cloud.digitalocean.com/kubernetes/clusters/8eb36bf4-5937-4756-aba8-fe7c6af47401/db/0057255e0f48be6fda8b159c3ab40c6505407b45/#/overview?namespace=_all)

**kubernetes** All namespaces Search

Overview

Workloads Workloads

Cron Jobs Daemon Sets Deployments Stateful Sets

Daemons Status

Running: 6 Running: 2 Running: 21 Running: 2

Daemon Sets Deployments Pods Replica Sets

Service Ingresses N Ingress Classes Services N

Config and Storage Config Maps N Persistent Volume Claims N Secrets N Storage Classes

Cluster Cluster Role Bindings Cluster Roles Events Namespaces Network Policies Nodes Persistent Volumes Role Bindings Roles

Daemon Sets

Name	Namespace	Images	Labels	Pods	Created ↑
do-node-agent	kube-system	docker.io/digitalocean/do-agent:3.11.0	app: do-node-agent c3.doks.digitalocean.com/component: do-node-agent c3.doks.digitalocean.com/plane: data	3 / 3	22.minutes.ago
csi-do-node	kube-system	registry.k8s.io/sig-storage/csi-node-driver-registrant:v2.6.0 docker.io/digitalocean/do-csi-plugin:v4.4.1	c3.doks.digitalocean.com/component: csi-node-service c3.doks.digitalocean.com/plane: data doks.digitalocean.com/managed: true	3 / 3	23.minutes.ago
cpc-bridge-proxy	kube-system	digitalocean/cpbridge:1.21.6	app: cpc-bridge-proxy c3.doks.digitalocean.com/component: cpc-bridge-proxy c3.doks.digitalocean.com/plane: data	3 / 3	23.minutes.ago

NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Volumes: HardDisk

X

## Volumes

Name	Droplet
 volume-sgp1-02 SGP1 / 50 GB	→  web-tech-demo-m544y 4 GB / 80 GB / SGP1

## Add a volume

### SELECT VOLUME SIZE

\$ 5/mo \$0.007/hour  Enter size in GB 50 	\$ 10/mo \$0.015/hour  100 GB	\$ 50/mo \$0.074/hour  500 GB	\$ 100/mo \$0.149/hour  1000 GB
---	--	--	--

### SELECT DROPLET TO ATTACH TO

 web-tech-demo-m544y 4 GB / 80 GB / SGP1	
--	---

### NAME VOLUME

Enter volume name volume-sgp1-02 
---

### CHOOSE CONFIGURATION OPTIONS

<b>Automatically Format &amp; Mount</b> <input checked="" type="radio"/> We will choose the appropriate default configurations. These settings can be changed later via ssh.	<b>Manually Format &amp; Mount</b> <input type="radio"/> We will still attach the volume. You can then manually format and mount the volume.
--	--

Choose a filesystem  Ext4  XFS

Block storage volumes are created in the same region and project as the Droplet they are attached to.

Create Volume

# DB: Choose DB: mongo

## Create Database Cluster

Choose a datacenter region

 Singapore • Datacenter 1 • SGP1 5 resources

Additional datacenter regions ▼

VPC network - default-sgp1

All resources created in this datacenter will be members of the same [VPC network](#). They can communicate securely over their Private IP addresses.

Choose a database engine

 MongoDB v5.0 ▾

 PostgreSQL v14 ▾

 MySQL v8

 Redis v7



## Worry-free Managed Database

Leave the complexity of database administration to us. So you can focus on building great apps.

- ✓ Point in time recovery (PITR)
- ✓ Automated Failover and Failback ?
- ✓ Logging & Metrics Dashboard
- ✓ Zero-Downtime Scaling and Forking
- ✓ Automated maintenance

[Check out all Managed Database has to offer](#)

# DB: Choose plan

## Choose a database configuration

You will be able to change the configuration at anytime after the cluster is created.

### Starter plan

- \$15/mo (\$0.022/hr)  
Basic / 1 vCPU / 1 GB RAM / 15 GB SSD

### Pro plans

- \$30/mo (\$0.045/hr)  
Basic / 1 vCPU / 2 GB RAM / 34 GB SSD
- \$60/mo (\$0.089/hr)  
Basic / 2 vCPU / 4 GB RAM / 56 GB SSD

Additional pro plans

### Add two standby nodes for zero downtime recovery

Automatically replace the primary node in the case of a failure, ensuring your data stays available.



Add two standby nodes

\$30/month  
(\$0.022/hour)

## Finalize and create

### Choose a unique database cluster name\*

Names must be in lowercase and start with a letter. They can be between 3 and 30 characters long and may contain dashes.

db-mongodb-demo-web-tech

### Select a project

Demo

### Tags

May contain letters, numbers, colons, dashes, and underscores.

Demo X

Type tags here

### Total monthly cost

Primary node \$15.00  
Basic / 1 vCPU / 1 GB RAM / 15 GB SSD

Total cost \$15.00/month  
\$0.022/hour

Create Database Cluster

# DB: get connection string

**DATABASE CLUSTER TOTAL COST**

[How does pricing work? ↗](#)

**\$15.00 monthly rate**  
This amount is prorated, and does not reflect your month to date usage.

---

**TRUSTED SOURCES**

**⚠ Warning:** Your cluster is open to all incoming connections.  
[Secure this database cluster by restricting access.](#)

---

**VPC NETWORK**

 [default-sgp1](#) 10.104.0.0/20

---

**CURRENT VERSION**

**MongoDB 5**  
Version upgrades are in place and won't disrupt your service.

---

**CONNECTION DETAILS**

[Public network ✓](#) [VPC network](#) [Connection parameters ↴](#)

```
username = doadmin
password = <replace-with-your-password>
host = mongodb+srv://db-mongodb-demo-web-tech-b25db23b.mongo.ondigitalocean.com
database = admin
```

**User:** doadmin [▼](#)    **Database:** admin [▼](#)

[Copy](#)

**SSL** **i** SSL connections are required, you will need to use a TLS compatible MongoDB client to connect. [Having difficulty connecting? ↗](#)

**[+]** [Connection troubleshooting](#) 

---

**MANAGED DATABASE RESOURCES**

 [Docs](#)

 [API Docs](#)



# Spaces: store & share files

## Spaces Object Storage

[Manage Keys ↗](#)



Store and deliver vast amounts of content

S3-compatible object storage with a built-in CDN that makes scaling easy, reliable, and affordable.

[Create a Space](#)

# Spaces: store & share files

## Create a Space

Choose a datacenter region

Singapore • Datacenter 1 • SGP1      7 resources

Additional datacenter regions

Content Delivery Network (CDN)

Deliver web assets up to 70% faster with global edge caching technology. No additional cost, standard bandwidth fees apply

Note: You can assign a custom domain in CDN settings after the Space is created

Enable CDN

Finalize and create

Choose a unique Space name\*

Names must be in lowercase. They can be between 3 and 63 characters long and may contain dashes.

Select a project

Demo



### Simple scalable object storage

Leave the complexity of Resource Name administration to us. So you can focus on building great apps.

- ✓ Predictable pricing
- ✓ Built-in CDN
- ✓ Unlimited Spaces buckets and uploads
- ✓ Use your own custom subdomain
- ✓ Easily integrate S3-compatible

[Check out all Spaces has to offer ↗](#)

### Total monthly cost

 Spaces subscription	\$5.00
<hr/>	
 Total cost	<b>\$5.00/month</b>
	\$0.007/hour

[Create a Spaces Bucket](#)

# Spaces: create a Space

## Create a Space

Choose a datacenter region

Singapore • Datacenter 1 • SGP1      7 resources

Additional datacenter regions

Content Delivery Network (CDN)

Deliver web assets up to 70% faster with global edge caching technology. No additional cost, standard bandwidth fees apply

i Note: You can assign a custom domain in CDN settings after the Space is created

Enable CDN

Finalize and create

Choose a unique Space name\*

Names must be in lowercase. They can be between 3 and 63 characters long and may contain dashes.

Select a project

Demo



### Simple scalable object storage

Leave the complexity of Resource Name administration to us. So you can focus on building great apps.

- ✓ Predictable pricing
- ✓ Built-in CDN
- ✓ Unlimited Spaces buckets and uploads
- ✓ Use your own custom subdomain
- ✓ Easily integrate S3-compatible

[Check out all Spaces has to offer ↗](#)

### Total monthly cost

Spaces subscription

\$5.00

Total cost

**\$5.00/month**

\$0.007/hour

**Create a Spaces Bucket**

# Spaces: create done

The screenshot shows the DigitalOcean Spaces web interface. At the top, there's a navigation bar with a circular icon, the space name "demo-web-tech", and a link to "Origin Endpoint" with the URL "https://demo-web-tech.sgp1.digitalocean.com". Below the navigation, there are two tabs: "Files" (which is selected) and "Settings". A sidebar on the left shows a profile picture and the space name "demo-web-tech". The main area has a purple header with the text "Use s3cmd CLI to upload your large files" and a sub-instruction about s3cmd. It features a large dashed-drag-and-drop area with a paper airplane icon and the text "Drag and drop file(s) to upload or use the ‘Upload’ button.". Below this are two buttons: "Upload" and "Create Folder". To the right of the main area is a decorative graphic of a brain.

demo-web-tech

Origin Endpoint <https://demo-web-tech.sgp1.digitalocean.com>

Files Settings

demo-web-tech

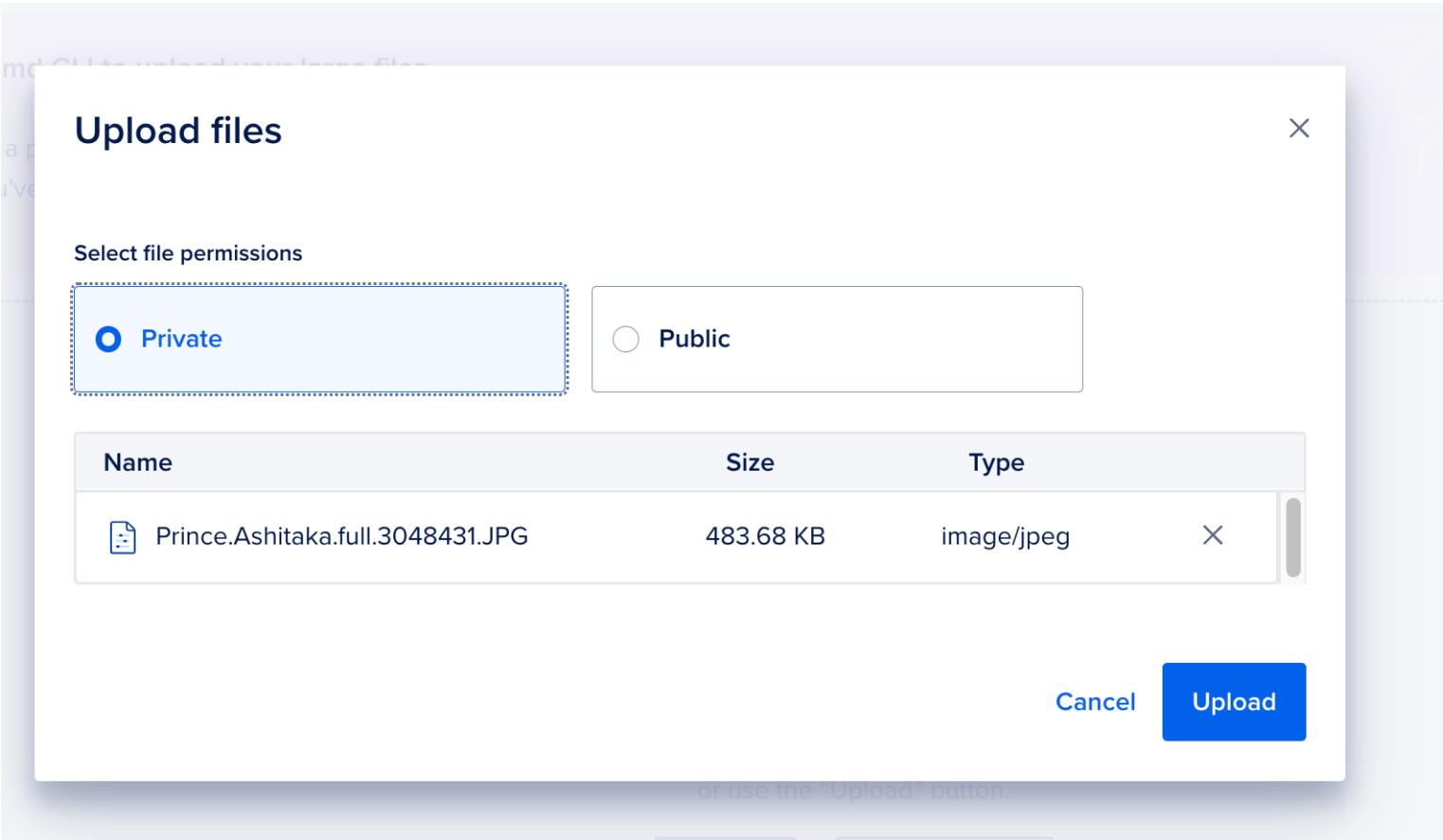
Use s3cmd CLI to upload your large files

s3cmd is a popular cross-platform command-line tool for managing S3 and S3-compatible object stores. Once you've [set up s3cmd](#) you can use it to [upload files](#) to your Space.

Drag and drop file(s) to upload  
or use the “Upload” button.

Upload Create Folder

# Spaces: upload file



# Spaces: upload done

← Go back

demo-web-tech [Origin Endpoint](#) <https://demo-web-tech.sgp1.digitalocean.com>

Upload finished (1/1 uploaded)

[View details](#)

Files Settings

demo-web-tech 1 item

Search Create Folder Upload

<input type="checkbox"/> Name	Size	Last modified	...
<input type="checkbox"/>  Prince.Ashitaka.full.3048431.JPG	483.68 KB	Just now	...



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Spaces: create folder

The screenshot shows the DigitalOcean Spaces interface. At the top, there's a navigation bar with a 'Go back' button, the space name 'demo-web-tech', its origin endpoint 'https://demo-web-tech.sgp1.digitaloceanspaces.com', and a file named 'Demo' (484 KB). A prominent message box indicates 'Upload finished (1/1 uploaded)' with a 'View details' button. Below this, there are tabs for 'Files' and 'Settings'. The 'Files' tab is selected, showing a search bar, a 'Create Folder' button, and an 'Upload' button. The file list table has columns for Name, Size, and Last modified. It contains two items: a folder named 'demo-folder' and a file named 'Prince.Ashitaka.full.3048431.JPG'.

Name	Size	Last modified
demo-folder	—	—
Prince.Ashitaka.full.3048431.JPG	483.68 KB	Just now

# Spaces: share file

← Go back

 **demo-web-tech**  
[Demo](#) • 1.1 MB • 3 items

Origin Endpoint <https://demo-web-tech.sgp1.digitalocean.com> 

✓ Upload finished (1/1 uploaded)

[View details](#)

[Files](#) [Settings](#)

 **demo-web-tech / demo-folder** 1 item

Search [Create Folder](#) [Upload](#)

<input type="checkbox"/> Name	Size	Last modified	...
 beauty.gif	642.13 KB	Just now	<a href="#">Manage Permissions</a> <a href="#">Manage Metadata</a> <a href="#">Rename</a> <a href="#">Move to...</a> <a href="#">Purge from CDN cache</a> <b>Quick Share</b> <a href="#">Copy Endpoints</a> <a href="#">Download</a> <a href="#">Delete</a>

# Spaces: share file (cont.)

