Education as a Service

Containerization and Orchestration of CS50 IDE

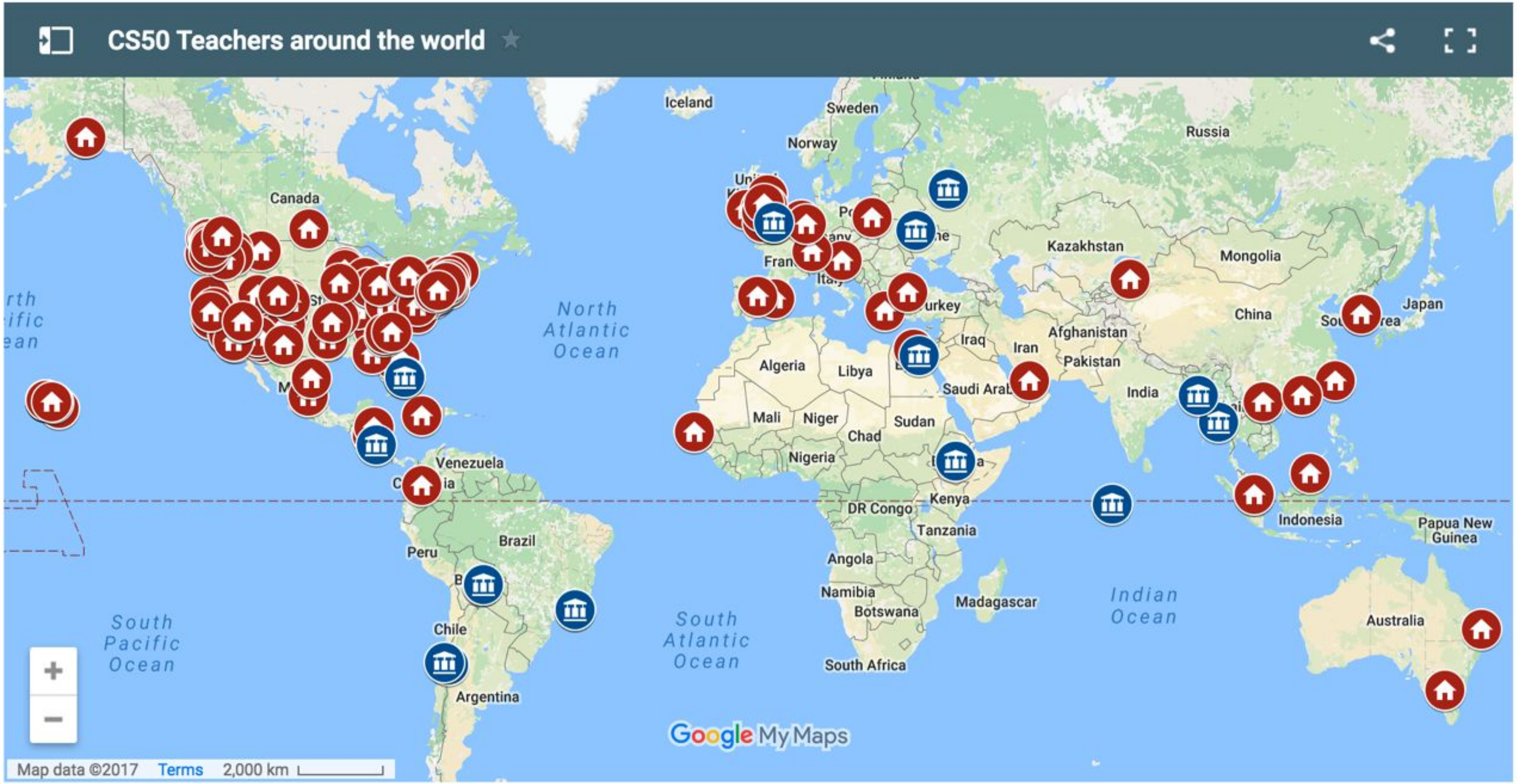
David J. Malan
malan@harvard.edu

Kareem Zidane
kzidane@cs50.harvard.edu

cs50.ly/kubecon

This is CS50

edx.org/cs50

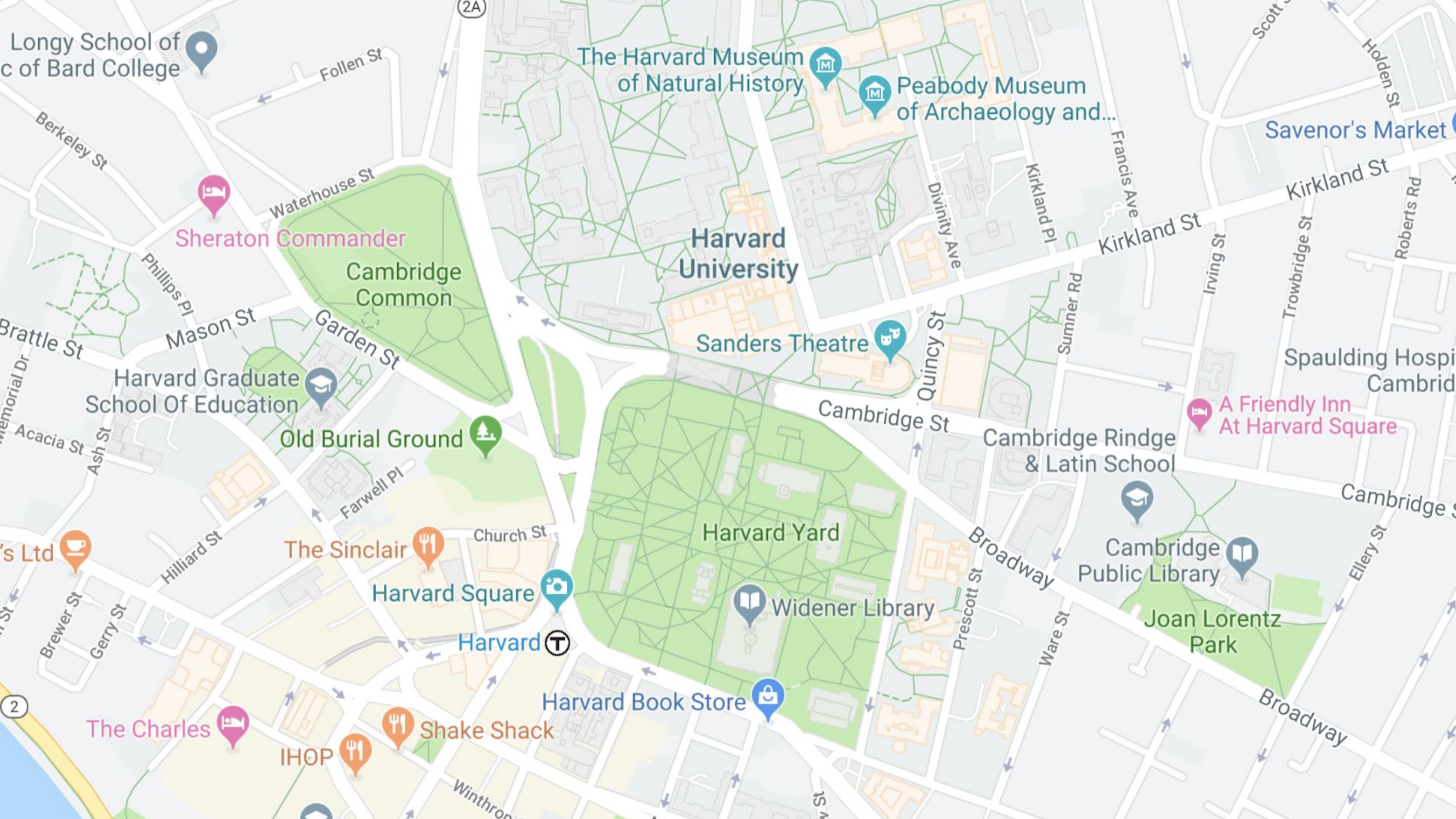

CS50 Teachers around the world
Iceland
Sweden
Norway
Russia
Canada
Kazakhstan
Mongolia
North Atlantic Ocean
Turkey
Japan
China
Iraq
Iran
Afghanistan
Pakistan
Algeria
Libya
Saudi Arabia
India
Mali
Niger
Sudan
Chad
Nigeria
Venezuela
Kenya
DR Congo
Tanzania
Indonesia
Papua New Guinea
Brazil
Peru
Angola
Namibia
Botswana
Madagascar
Indian Ocean
South Atlantic Ocean
South Pacific Ocean
Chile
Australia
South Africa
Argentina
Google My Maps
Map data ©2017
Terms
2,000 km

Scale

- 1,000 students on campus
- 1,000,000 registrants online

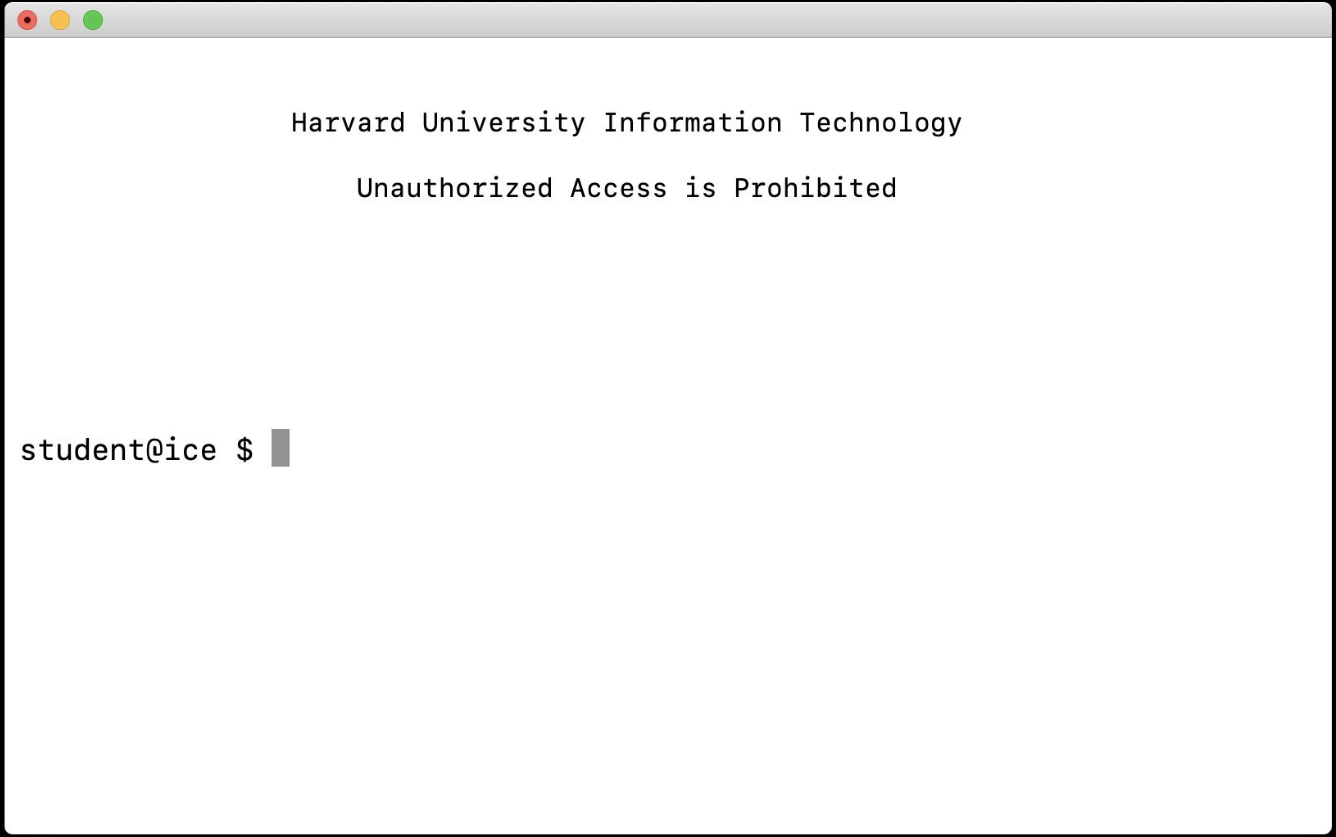
Scale

- 30,000 active per month

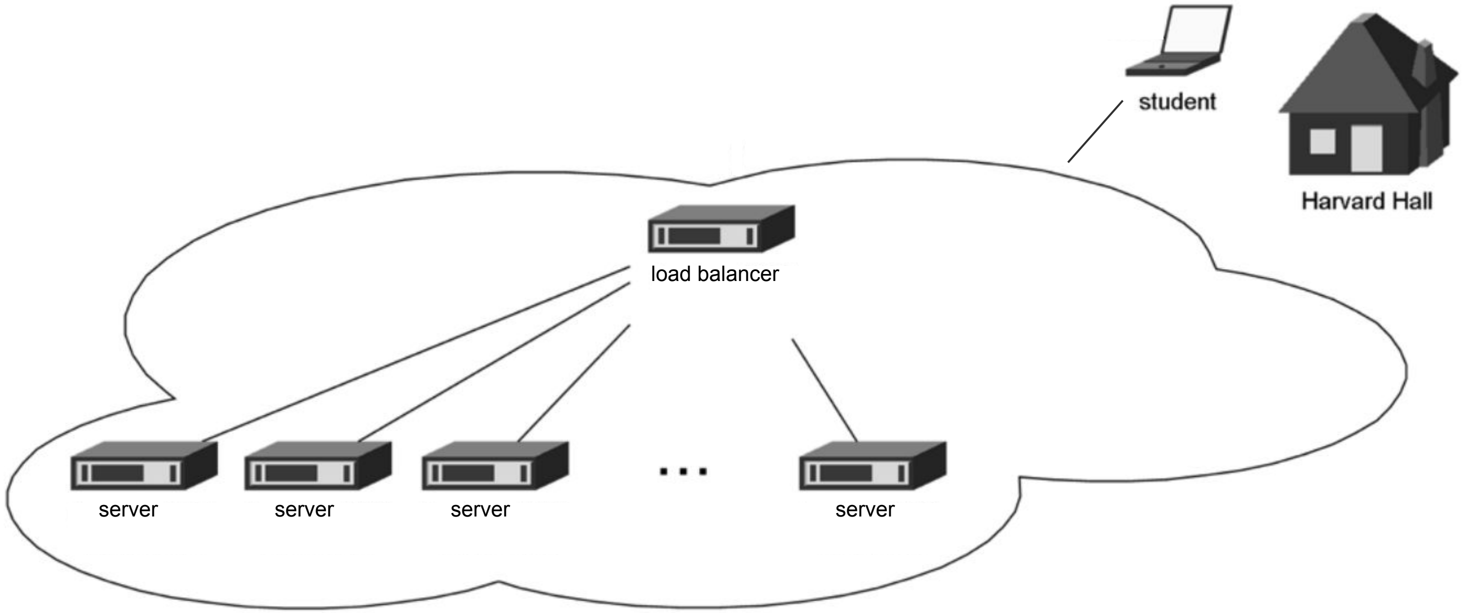

Longy School of
c of Bard College
Follen St
Berkeley St
Waterhouse St
Sheraton Commander
Cambridge Common
Phillips Pl
Mason St
Brattle St
Garden St
Harvard Graduate School Of Education
Memorial Dr
Acacia St
Ash St
Old Burial Ground
Farwell Pl
Hilliard St
Church St
The Sinclair
's Ltd
Harvard Square
Harvard
Brewer St
Gerry St
The Charles
Shake Shack
IHOP
Winthrop
Harvard Book Store
Widener Library
Harvard Yard
The Harvard Museum of Natural History
Peabody Museum of Archaeology and...
Harvard University
Sanders Theatre
Cambridge St
Quincy St
Divinity Ave
Kirkland Pl
Francis Ave
Kirkland St
Savenor's Market
Holden St
Scott St
Irving St
Trowbridge St
Roberts Rd
Sumner Rd
Spaulding Hospi
Cambrid
A Friendly Inn At Harvard Square
Cambridge Rindge & Latin School
Cambridge Public Library
Joan Lorentz Park
Broadway
Prescott St
Ware St
Ellery St
2A
2

On-Campus Cluster

1989–2007

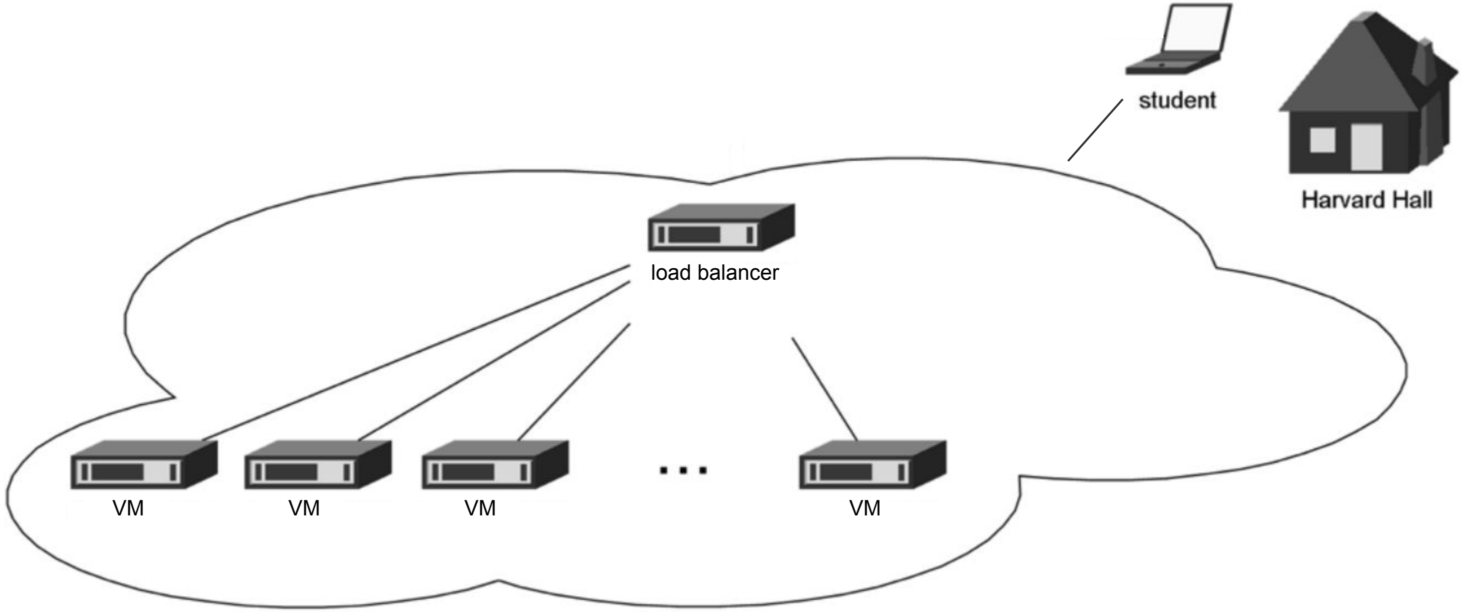


```
                Harvard University Information Technology

                    Unauthorized Access is Prohibited





student@ice $ 
```


student
Harvard Hall
load balancer
server
server
server
...
server

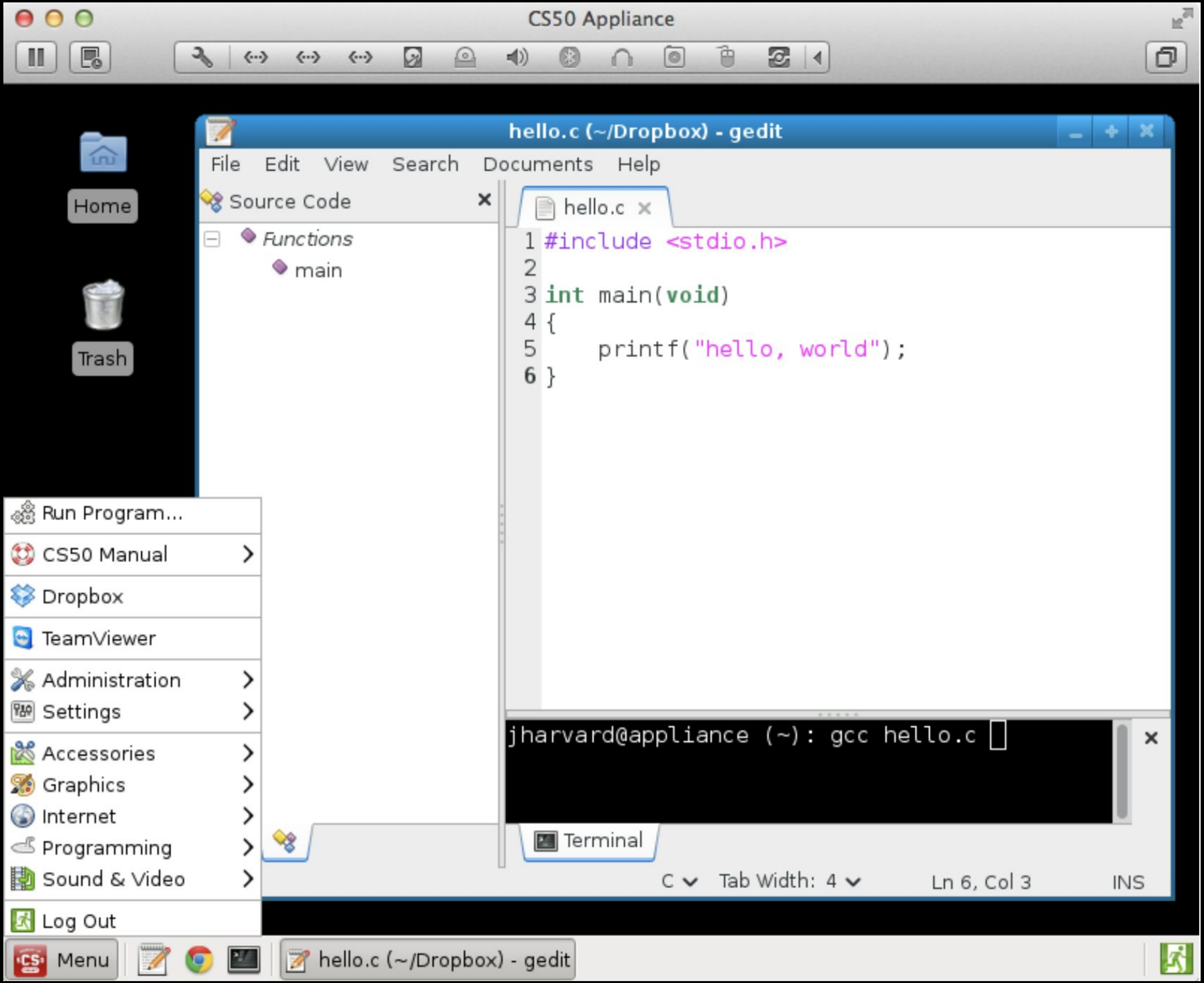
Off-Campus Cloud

2008–2010


student
Harvard Hall
load balancer
VM
VM
VM
...
VM

Client-Side Appliance

2011–2014



```c
#include <stdio.h>

int main(void)
{
    printf("hello, world");
}
```

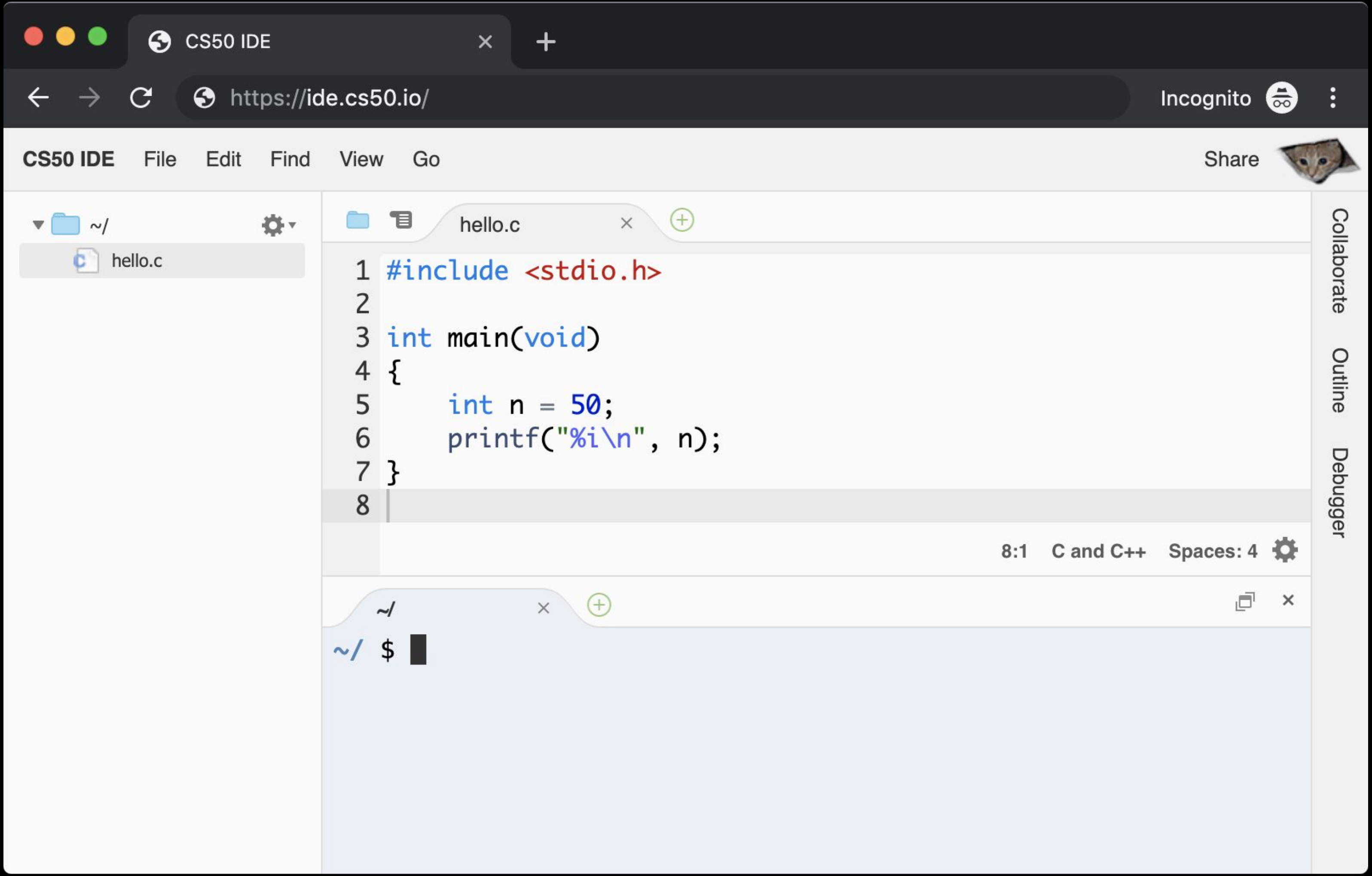
```
jharvard@appliance (~): gcc hello.c
```

Cloud-Based IDE

2015–

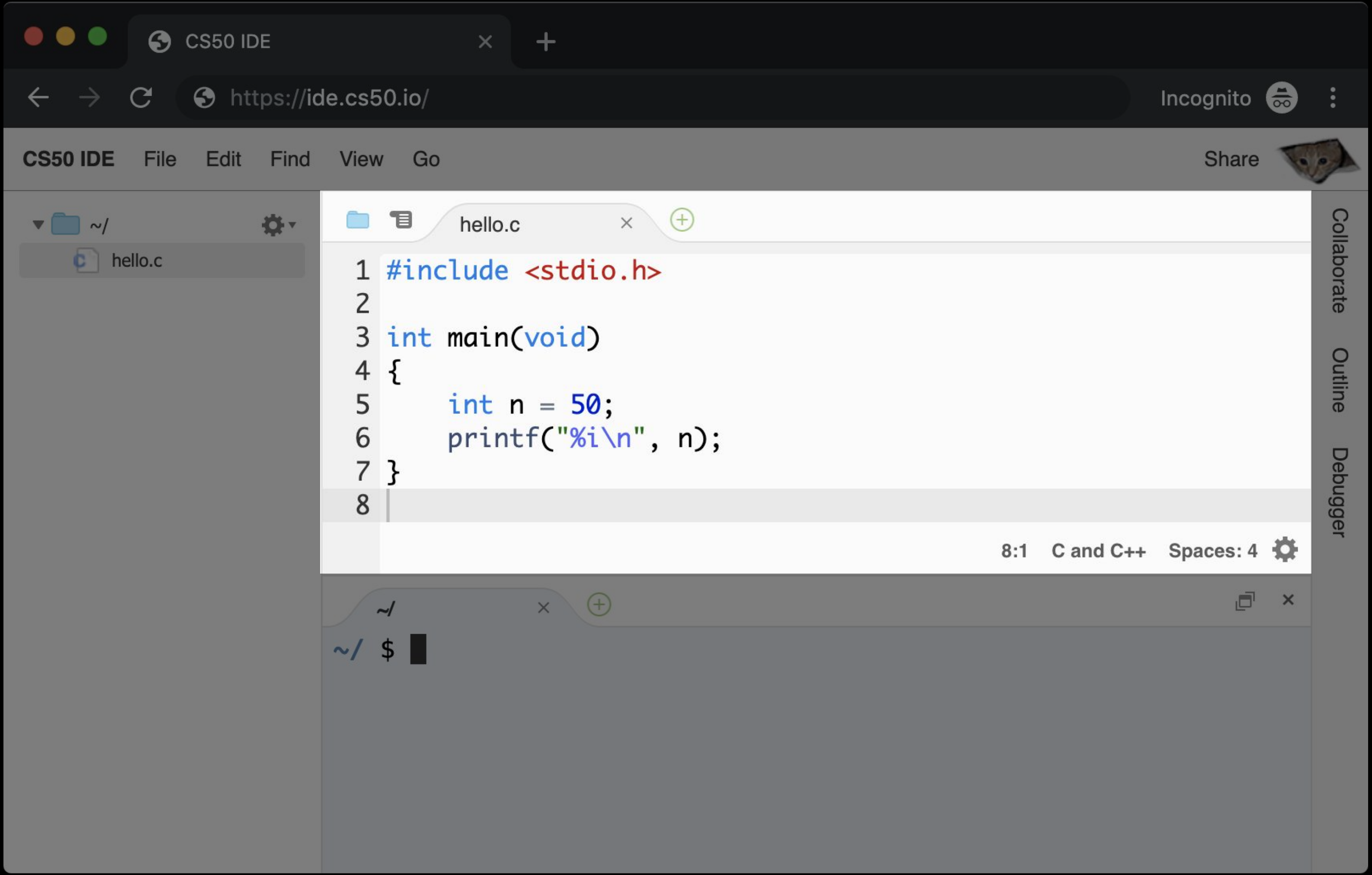
CS50 IDE

ide.cs50.io



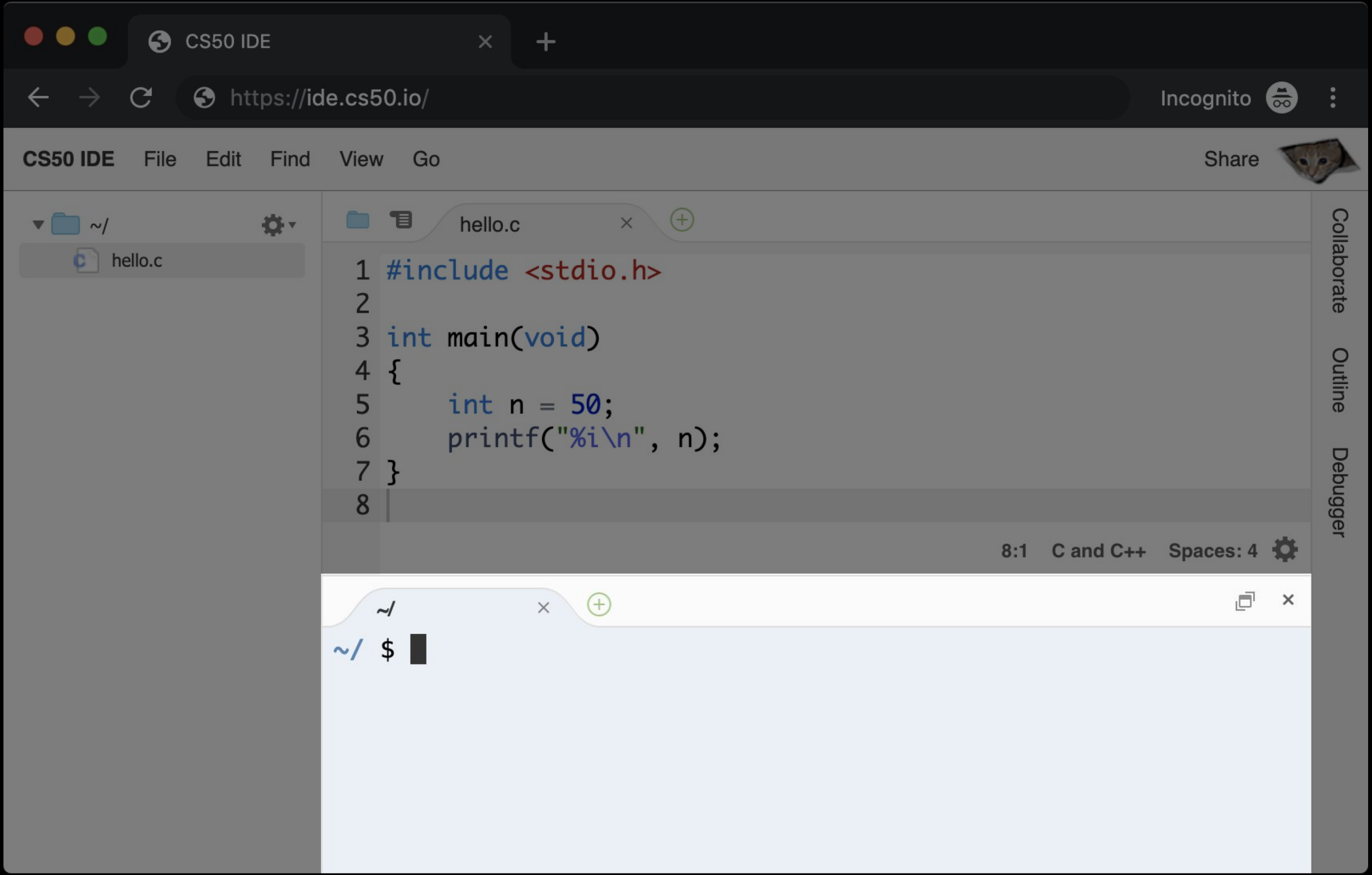
```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```



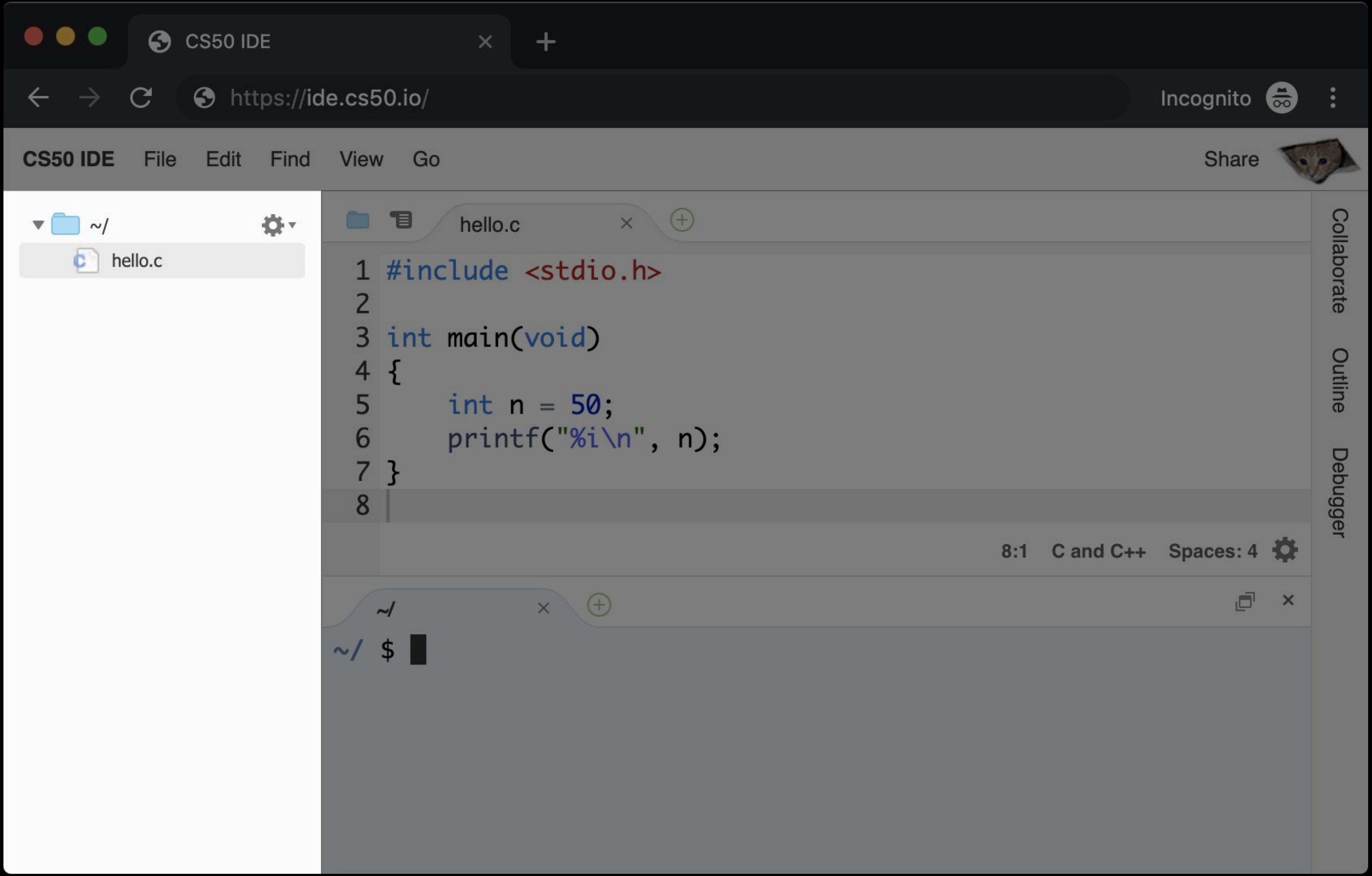
```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}
```

```
~/ $ 
```



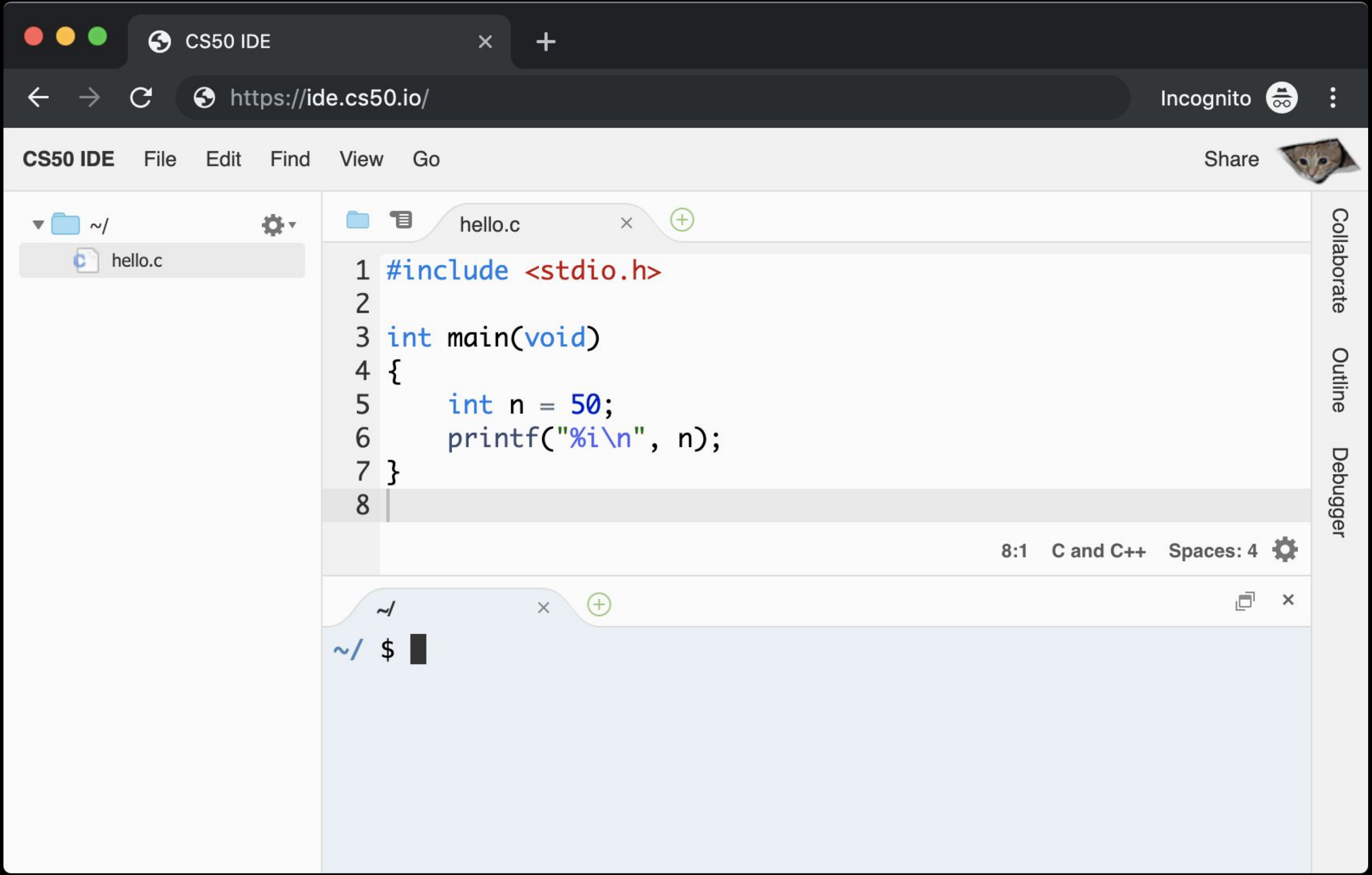
```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```



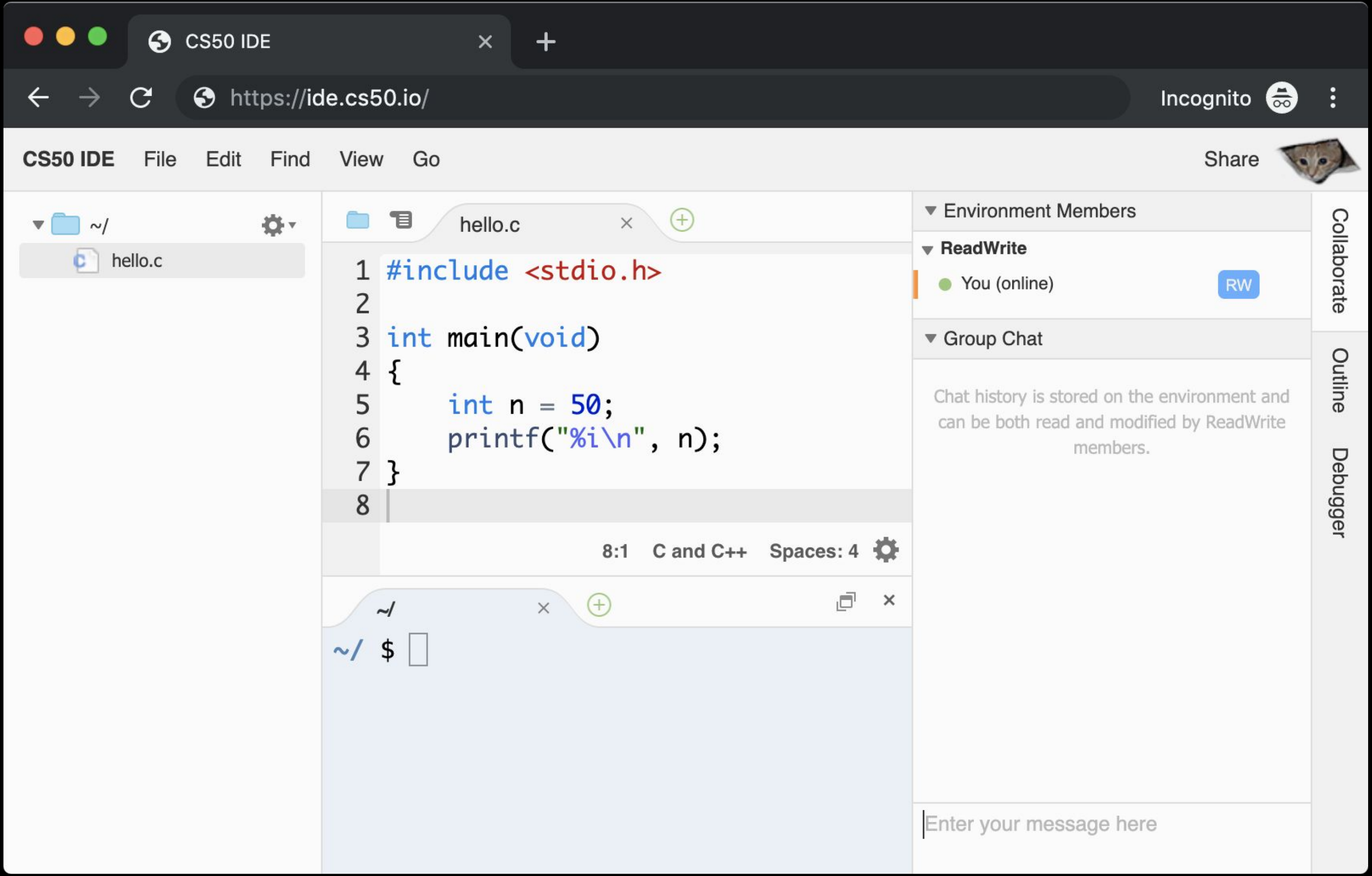
```
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```



```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}
```

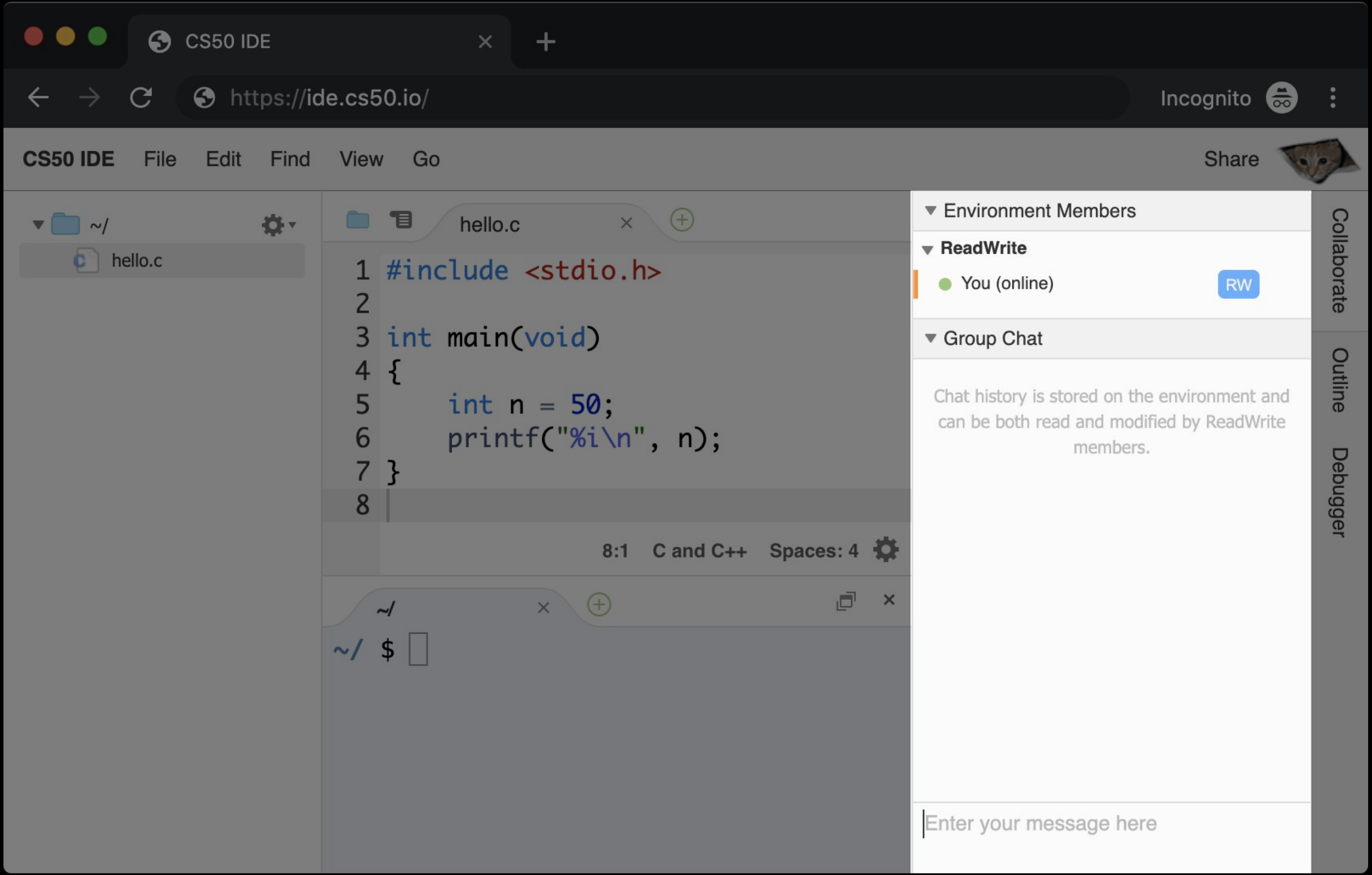
```
~/ $
```



```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}
```

Chat history is stored on the environment and can be both read and modified by ReadWrite members.

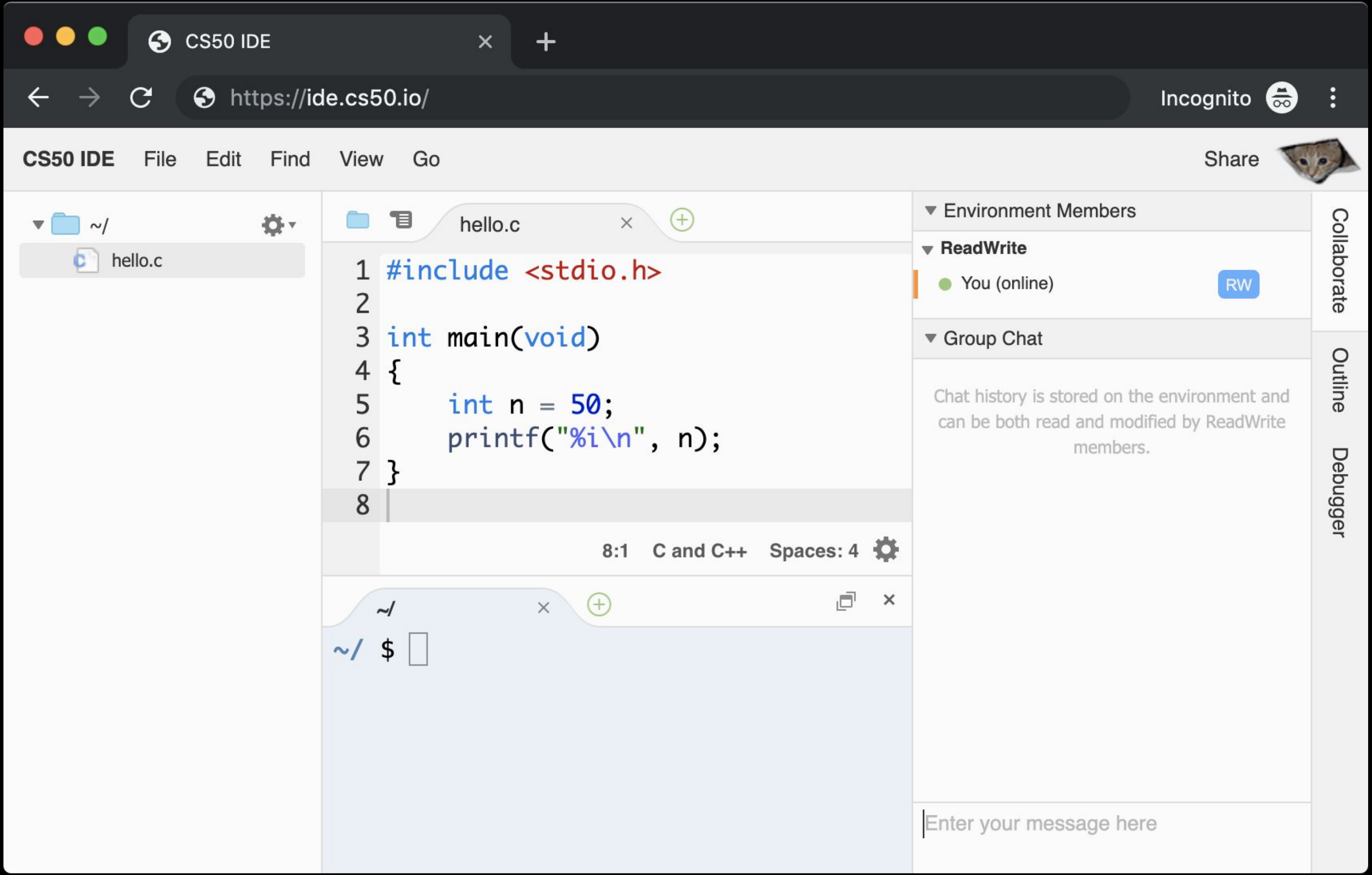
```
~/ $ 
```



```
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```

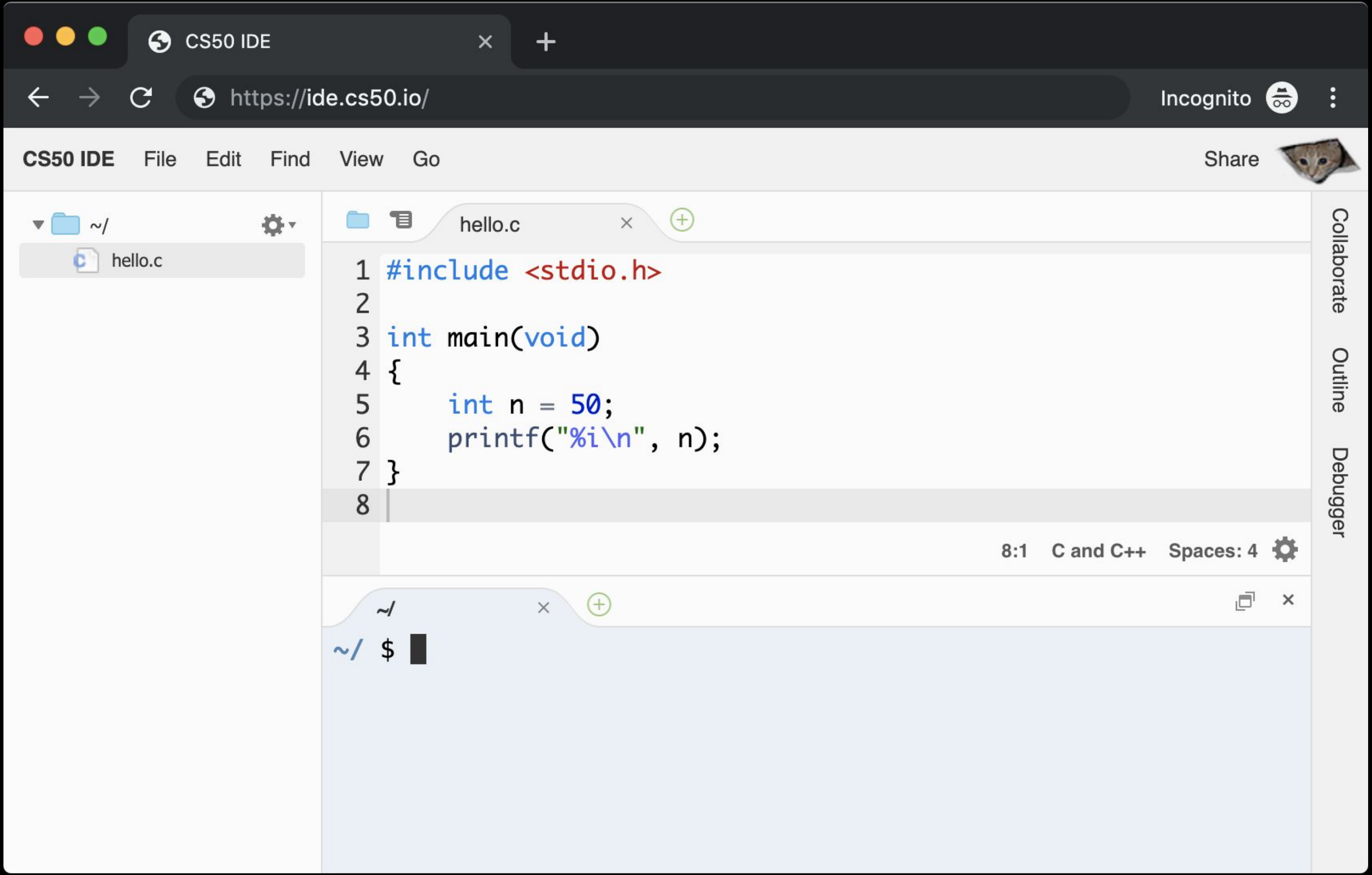
Chat history is stored on the environment and can be both read and modified by ReadWrite members.



```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}
```

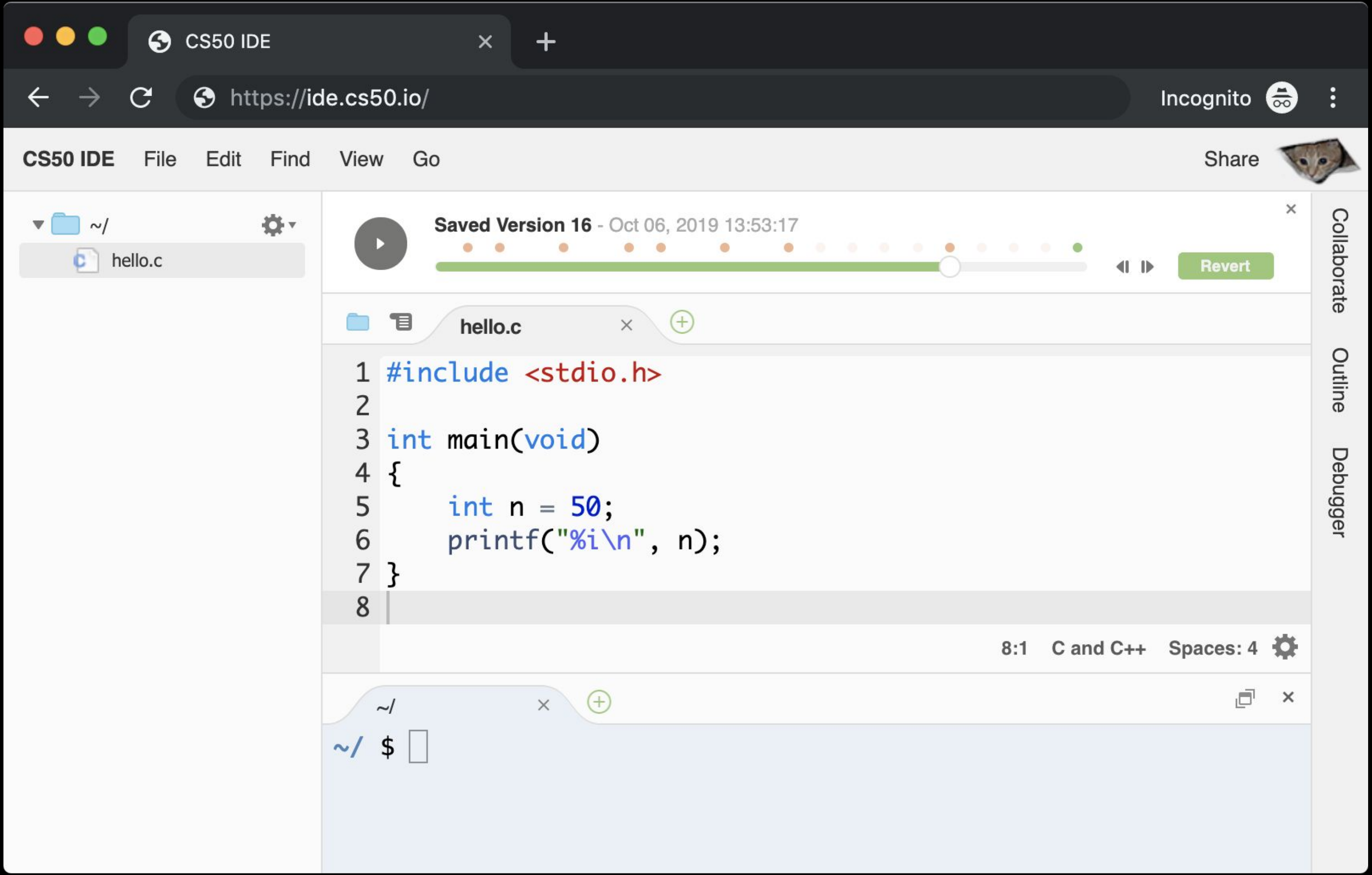
```
~/ $
```

Chat history is stored on the environment and can be both read and modified by ReadWrite members.



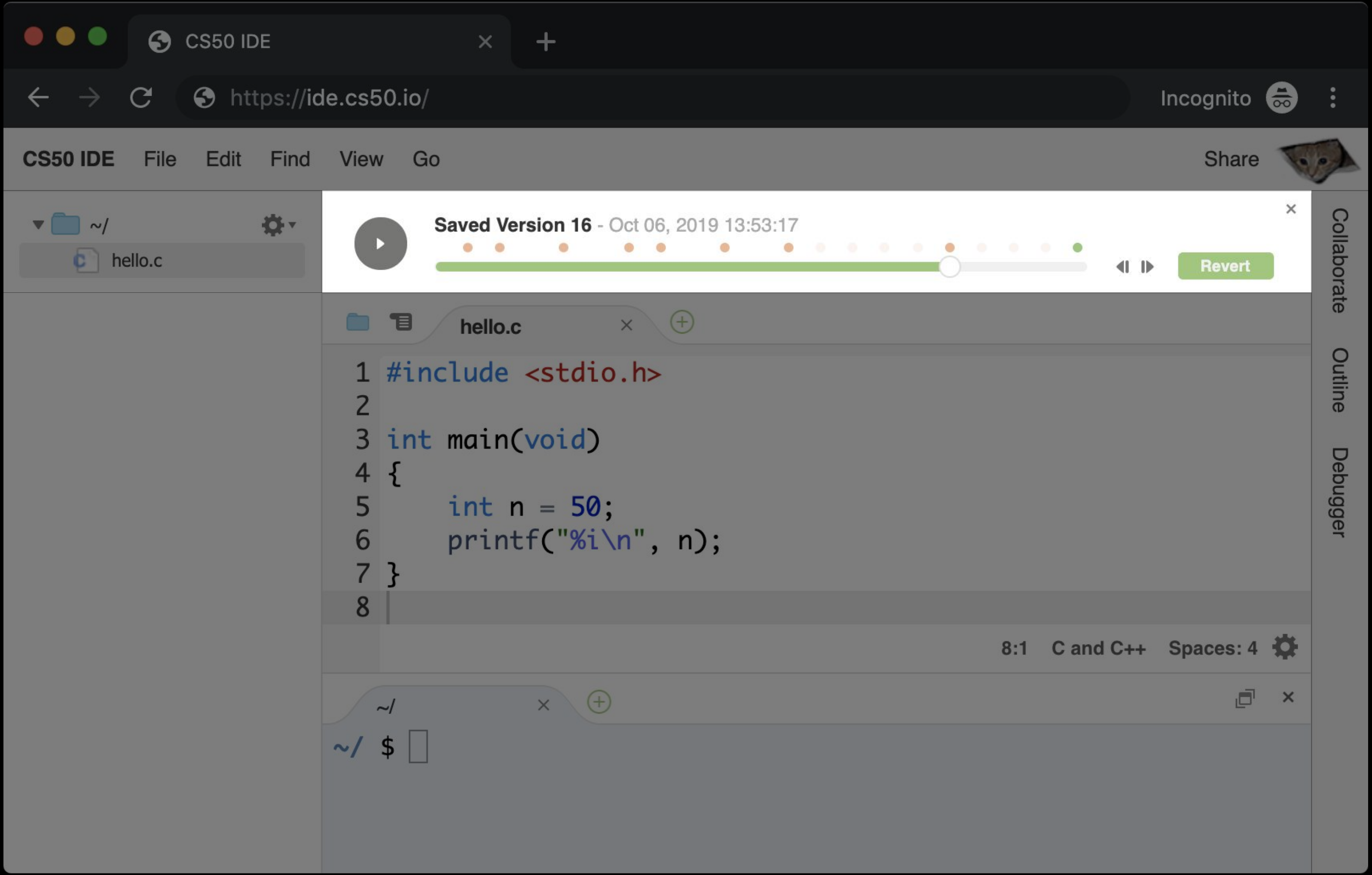
```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```



```
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

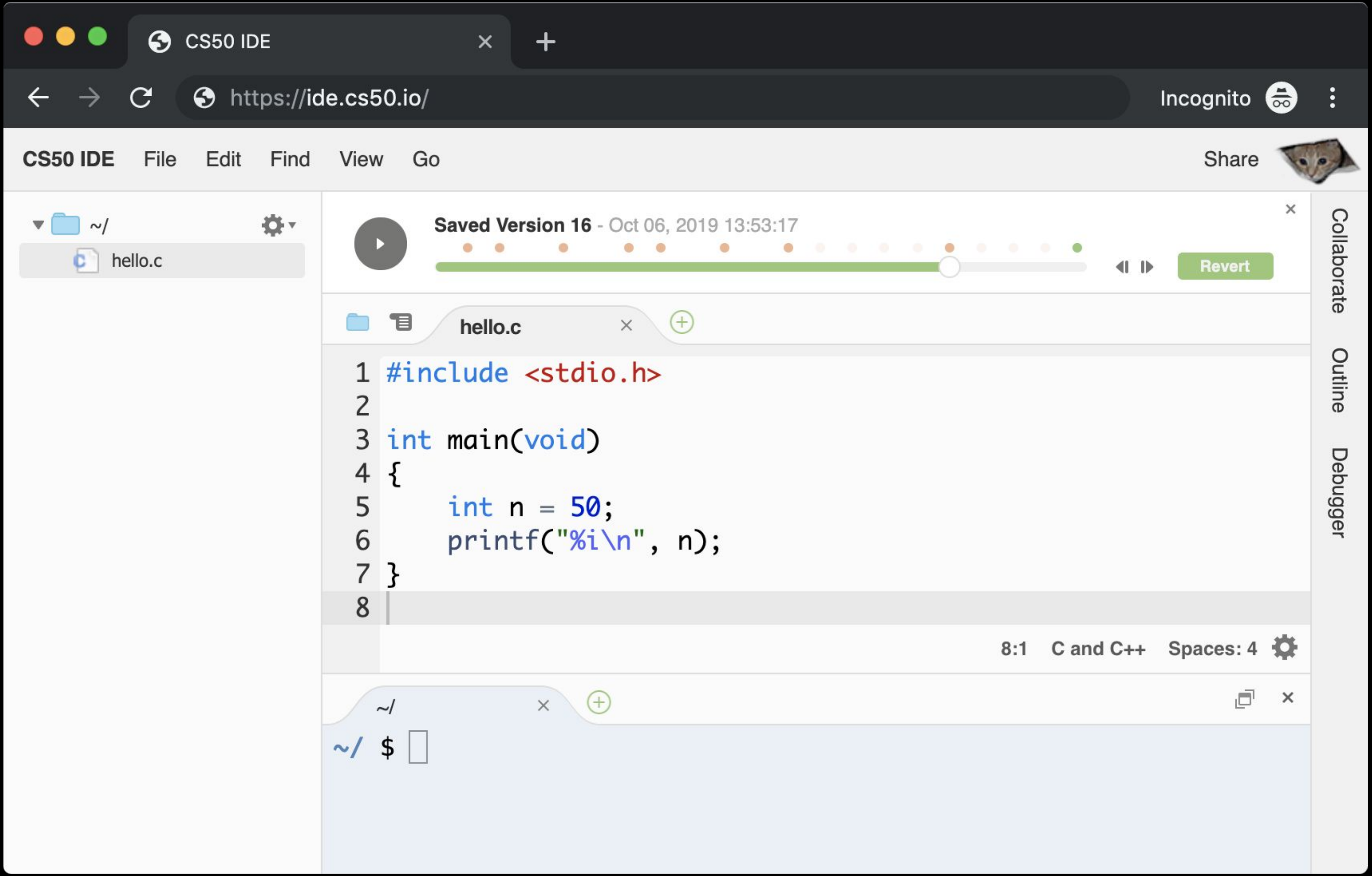
```
~/ $
```



Saved Version 16 - Oct 06, 2019 13:53:17

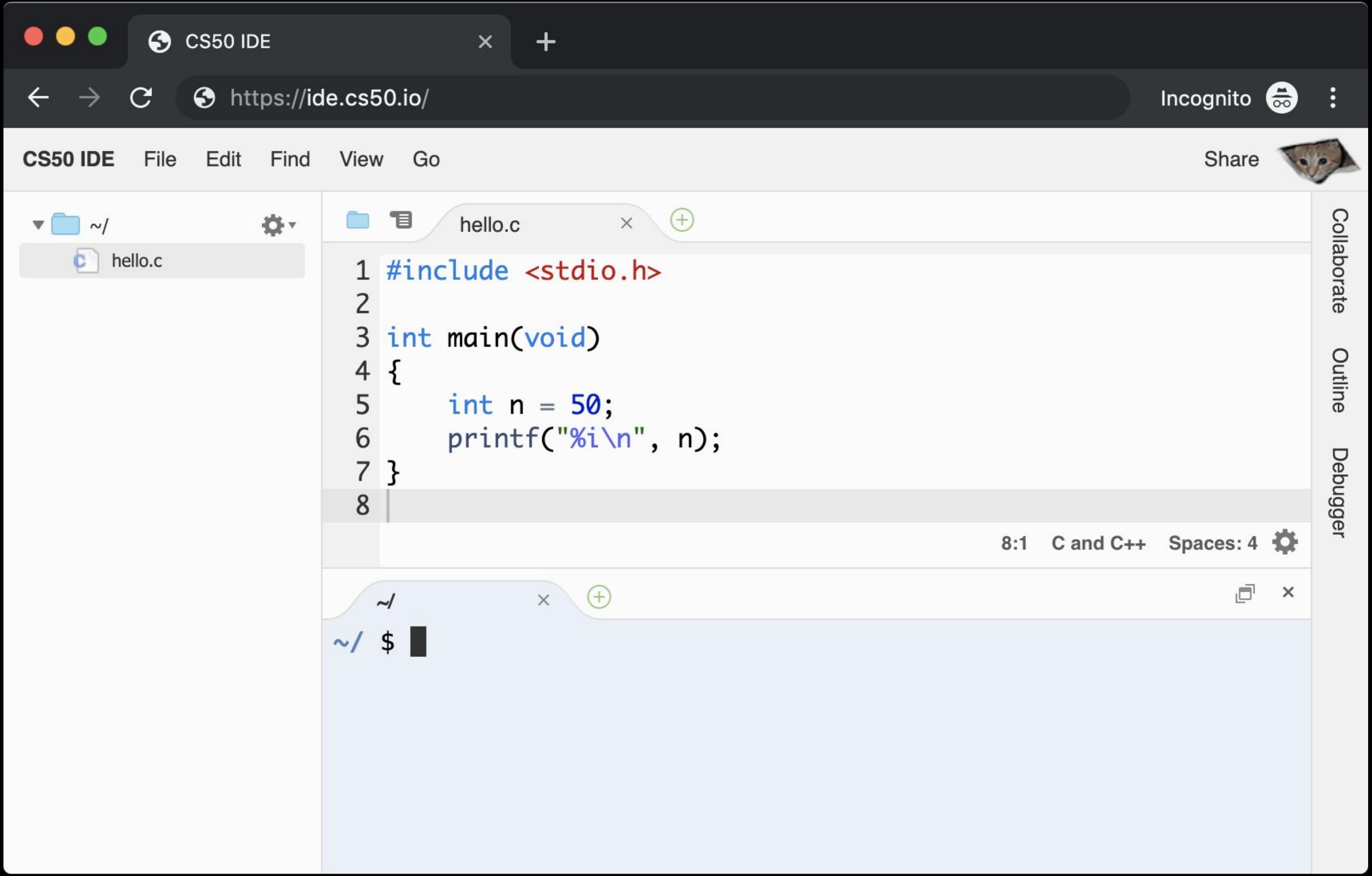
```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```



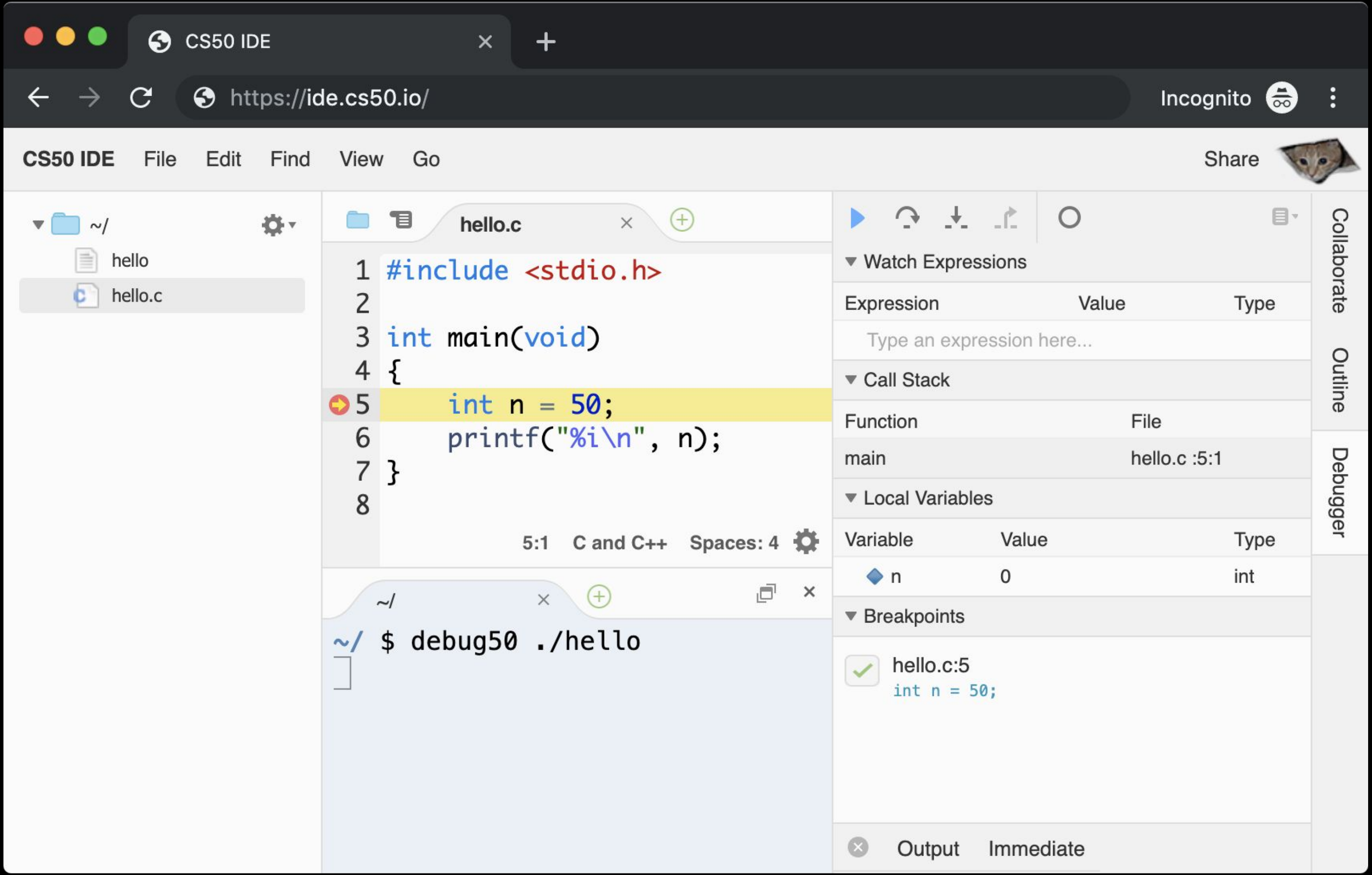
```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}
```

```
~/ $
```



```
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $
```



```
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

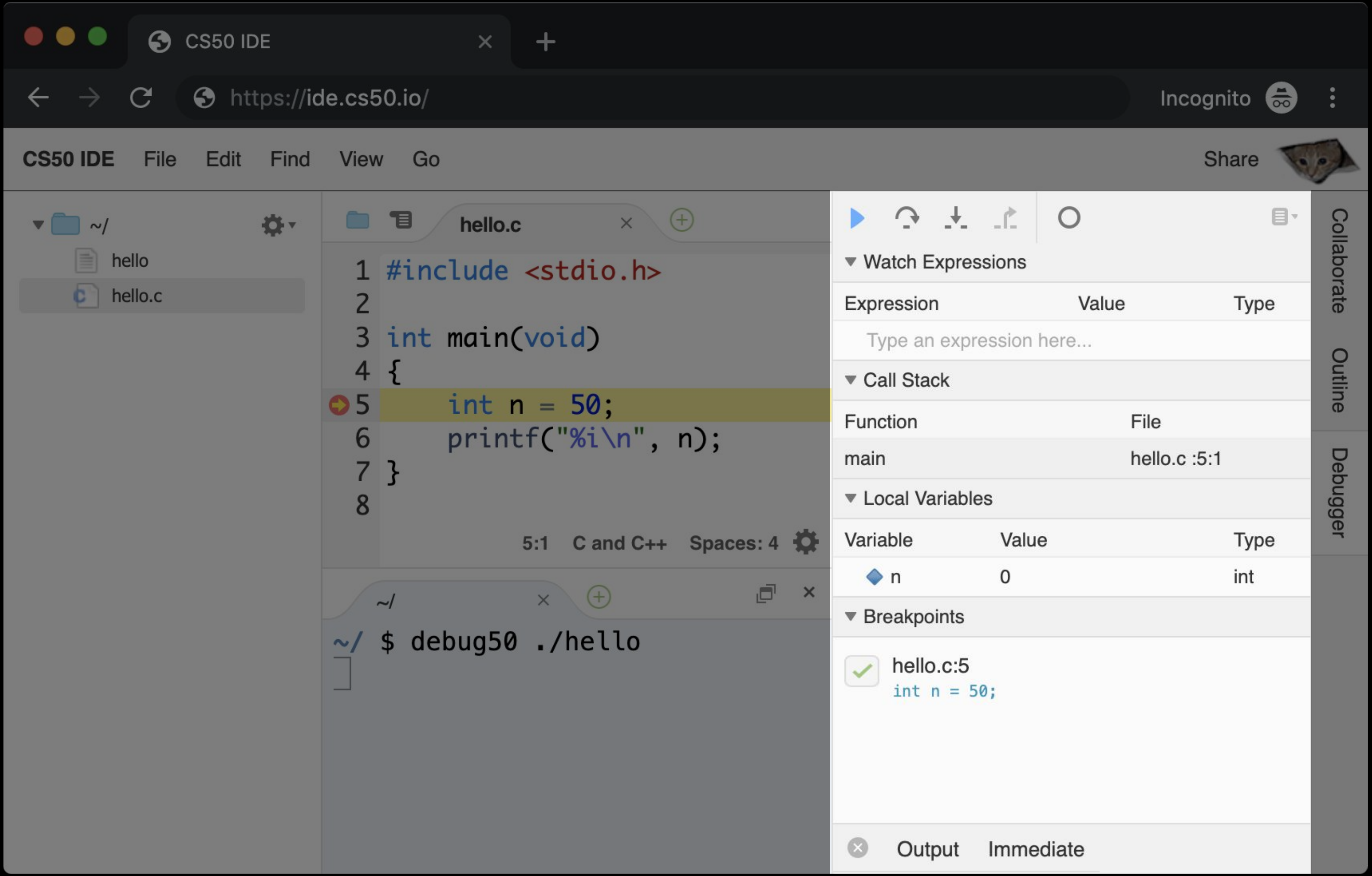
```
~/ $ debug50 ./hello
```

Expression	Value	Type
Type an expression here...		

Function	File
main	hello.c :5:1

Variable	Value	Type
n	0	int

✓ hello.c:5
`int n = 50;`



```
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}
```

```
~/ $ debug50 ./hello
```

Expression	Value	Type
Type an expression here...		

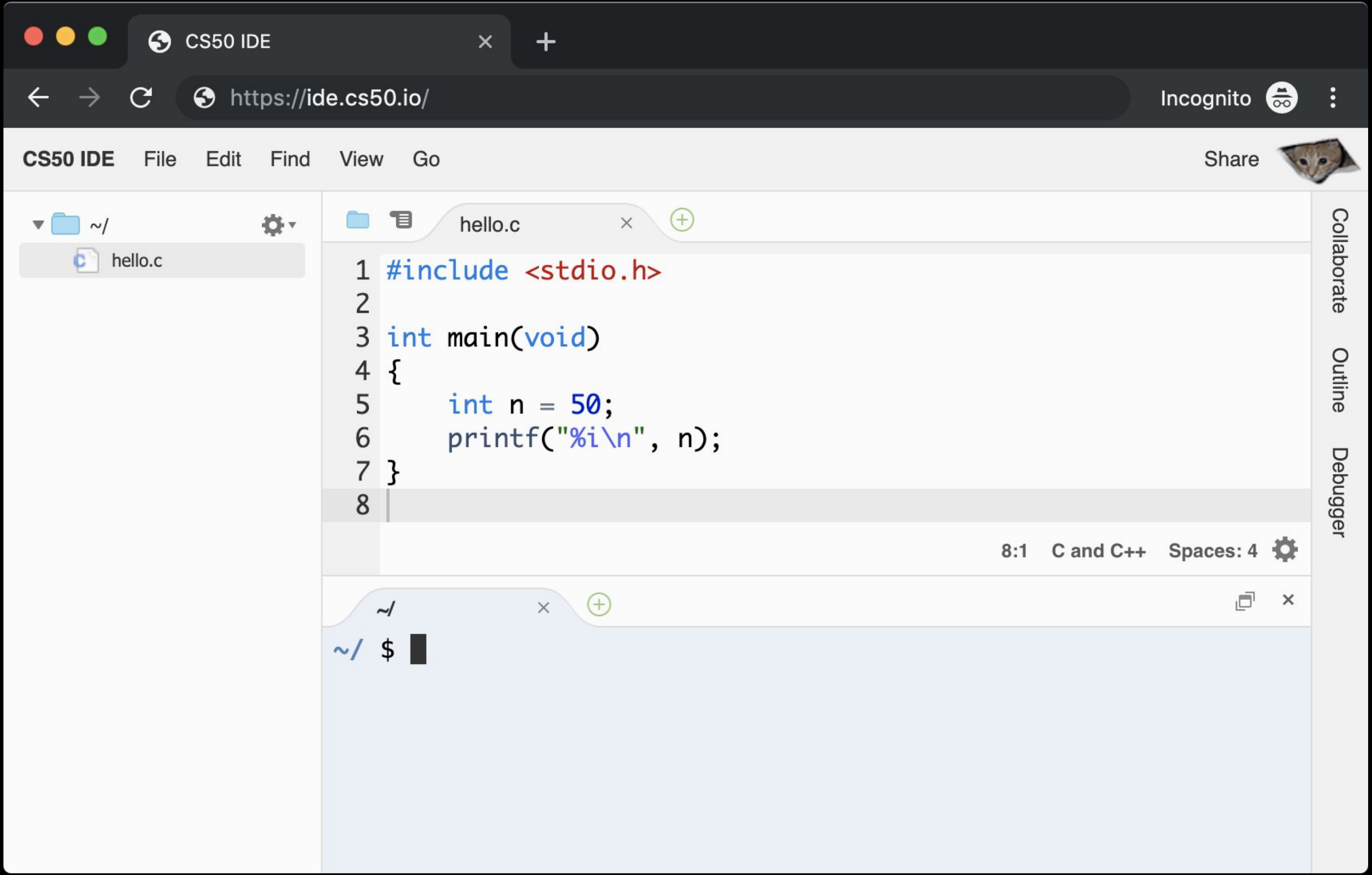
Function	File
main	hello.c :5:1

Variable	Value	Type
n	0	int

✓ hello.c:5
`int n = 50;`

AWS Cloud9

aws.amazon.com/cloud9



```c
#include <stdio.h>

int main(void)
{
    int n = 50;
    printf("%i\n", n);
}

```

```
~/ $ 
```

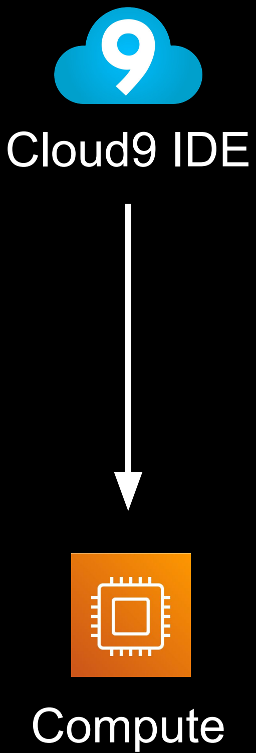
Versions

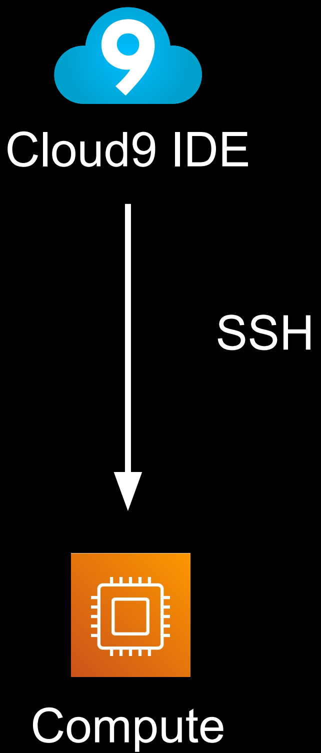
1. EC2 for Compute, EBS for Storage
2. S3 for Storage
3. Kubernetes for Orchestration

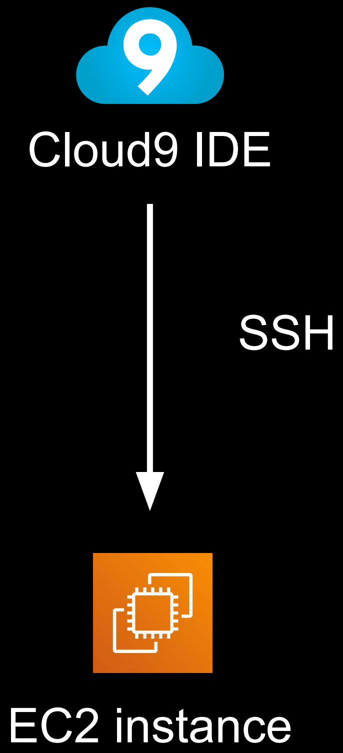
Cloud9 IDE

Cloud9 IDE

Compute

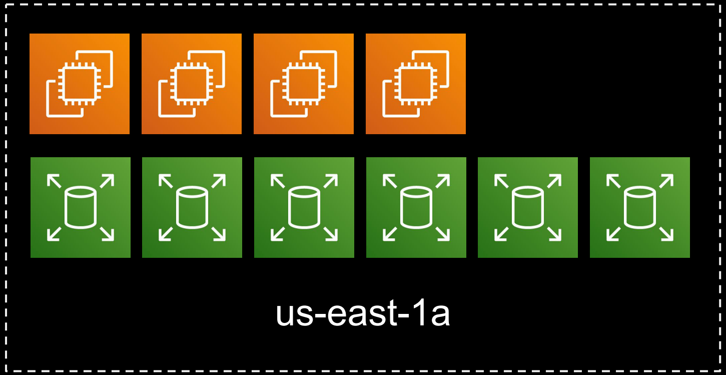

Cloud9 IDE
Compute

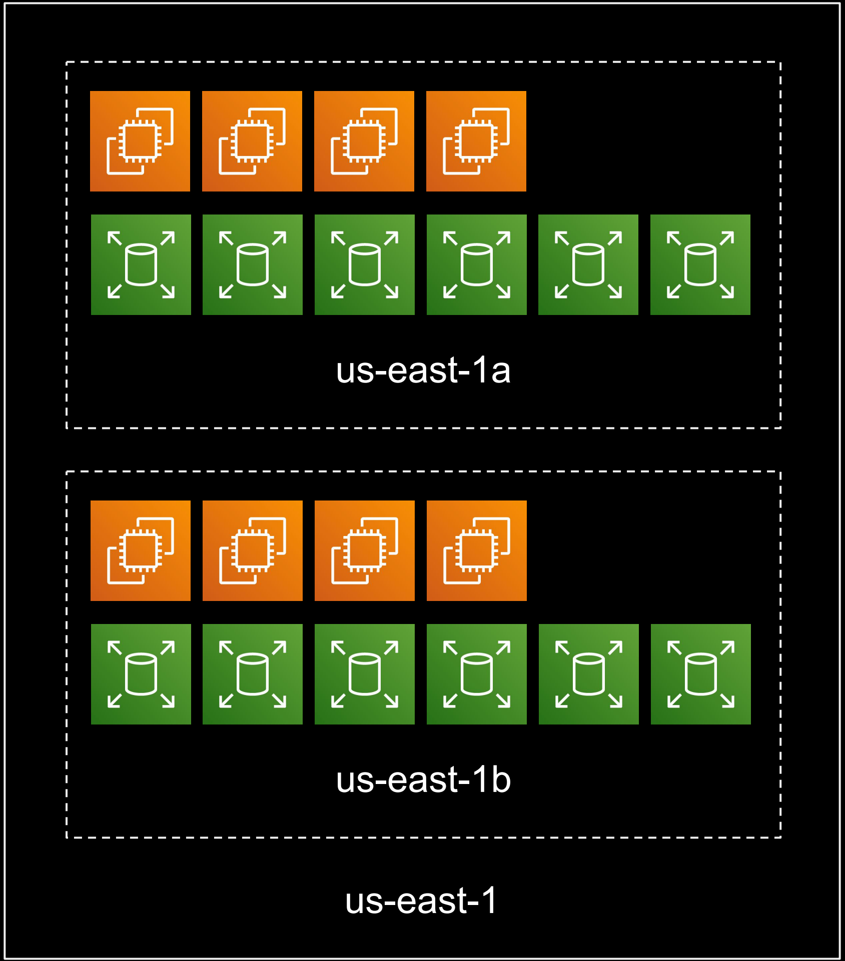

Cloud9 IDE
SSH
Compute

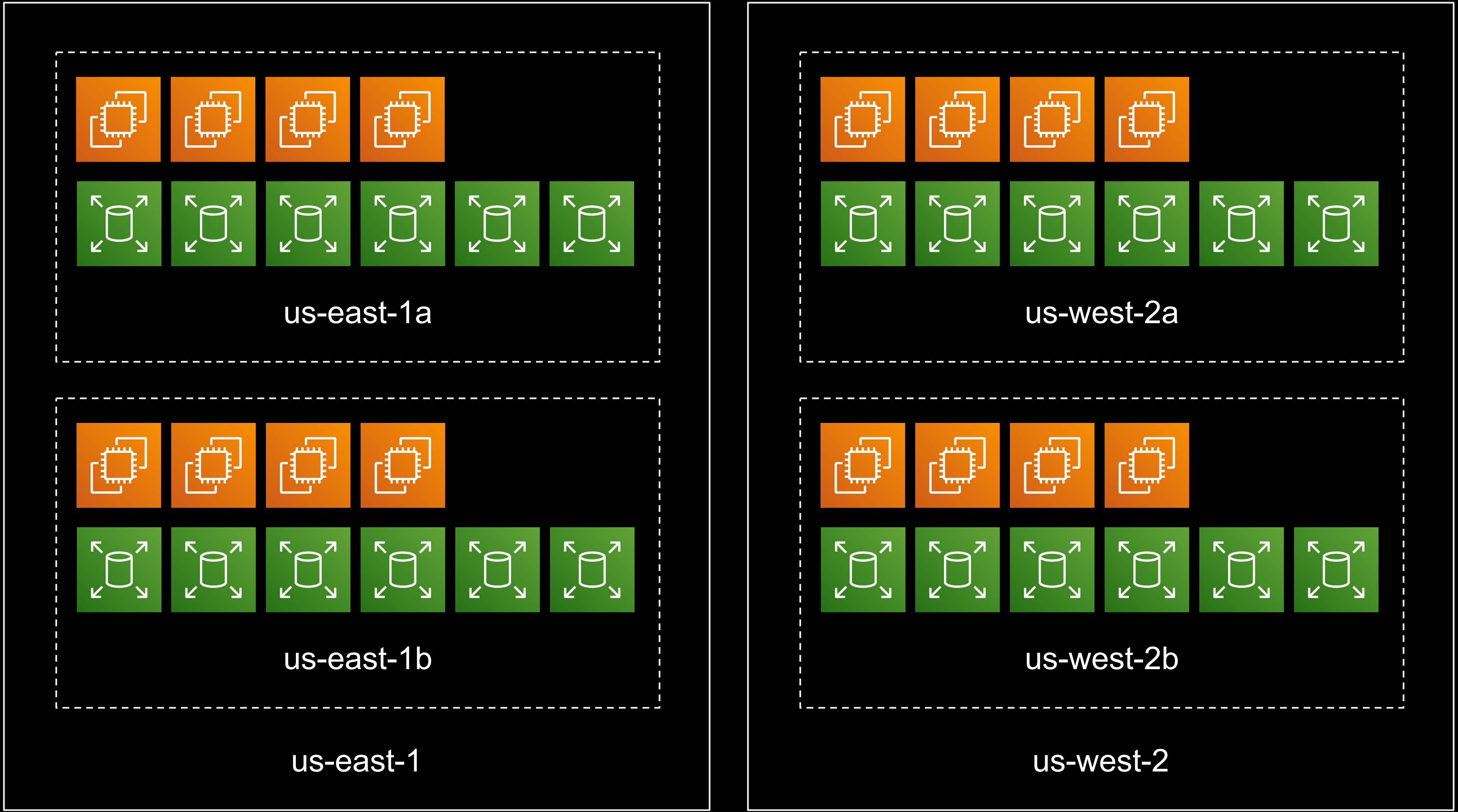

Cloud9 IDE
SSH
EC2 instance

Version 1

EC2 for Compute, EBS for Storage

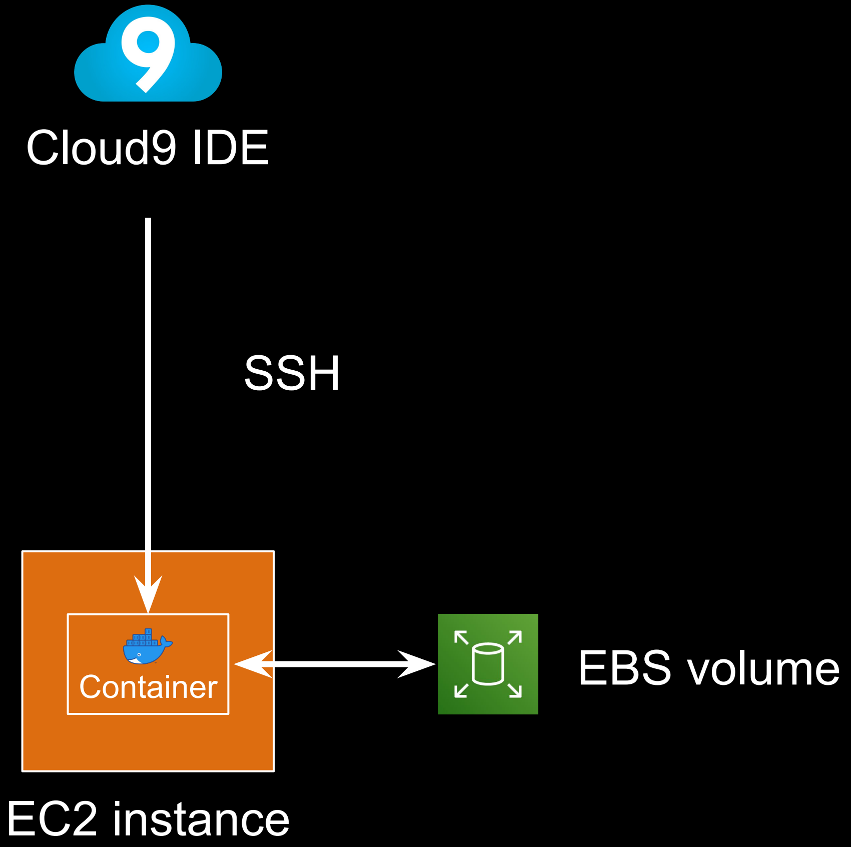

us-east-1a


us-east-1a
us-east-1b
us-east-1


us-east-1a
us-east-1b
us-east-1
us-west-2a
us-west-2b
us-west-2

Implementation Details

1. Get available EC2 instance from pool
2. Create user’s EBS volume and attach to instance
3. Format and mount volume
4. Start Docker container, mount volume, expose ports
5. Connect IDE to container using SSH
6. Redirect user to IDE


Cloud9 IDE
SSH
Container
EC2 instance
EBS volume

Challenges with EC2 for Compute

- Maintaining pools of EC2 instances

Challenges with EC2 for Compute

- Maintaining pools of EC2 instances
- Using SSM to run commands on EC2 instances

Challenges with EC2 for Compute

- Maintaining pools of EC2 instances
- Using SSM to run commands on EC2 instances
- Allocating entire EC2 instance for user

Challenges with EC2 for Compute

- Maintaining pools of EC2 instances
- Using SSM to run commands on EC2 instances
- Allocating entire EC2 instance for user
- Cleaning up after session ends
    - Terminating EC2 instance
    - Waiting for user’s EBS volume to be detached

Challenges with EC2 for Compute

- Maintaining pools of EC2 instances
- Using SSM to run commands on EC2 instances
- Allocating entire EC2 instance for user
- Cleaning up after session ends
    - Terminating EC2 instance
    - Waiting for user’s EBS volume to be detached
- Getting different hostname per session

Challenges with EC2 for Compute

- Maintaining pools of EC2 instances
- Using SSM to run commands on EC2 instances
- Allocating entire EC2 instance for user
- Cleaning up after session ends
    - Terminating EC2 instance
    - Waiting for user’s EBS volume to be detached
- Getting different hostname per session
- Removing instances temporarily to update the Docker image

Challenges with EBS for Storage

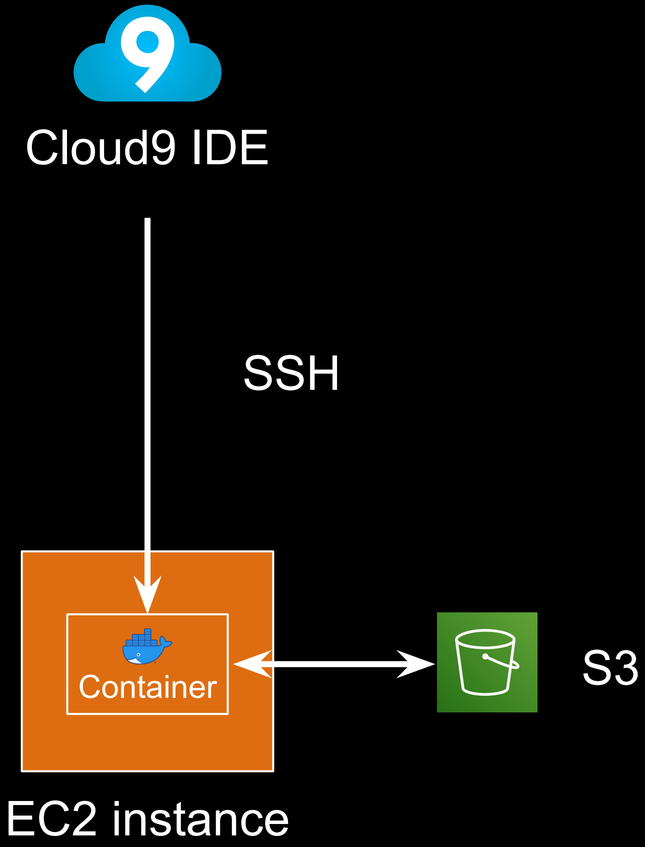
- Provisioning a volume per user wasn’t cost-effective

Challenges with EBS for Storage

- Provisioning a volume per user wasn’t cost-effective
- Assigning availability zone to each user was limiting

Version 2

S3 for Storage



Cloud9 IDE
SSH
Container
S3
EC2 instance

Challenges with S3 for Storage

- Setting up and refreshing credentials on the EC2 instance added complexity

Challenges with S3 for Storage

- Setting up and refreshing credentials on the EC2 instance added complexity
- Downloading user’s data initially was slow

Challenges with S3 for Storage

- Setting up and refreshing credentials on the EC2 instance added complexity
- Downloading user’s data initially was slow
- Uploading user’s data periodically was fragile

Challenges with S3 for Storage

- Setting up and refreshing credentials on the EC2 instance added complexity
- Downloading user’s data initially was slow
- Uploading user’s data periodically was fragile
- Limiting storage size per user wasn’t easy

Version 3?

AWS Fargate for Orchestration

Version 3?

AWS ECS for Orchestration

Version 3

Kubernetes for Orchestration

Creating an IDE per User

1. Create a namespace
2. Create a persistent volume claim (PVC)
3. Create a single-container pod, mount PVC, public SSH key
4. Connect IDE to container using SSH
5. Redirect user to their IDE

Solutions with Kubernetes

- Maintaining pools of EC2 instances

Solutions with Kubernetes

- Maintaining pools of EC2 instances
+ Managing nodes using Kubernetes

Solutions with Kubernetes

- Using SSM to run commands on EC2 instances

Solutions with Kubernetes

- - Using SSM to run commands on EC2 instances
- + Using the Kubernetes API to create the resources needed

Solutions with Kubernetes

- Allocating entire EC2 instance for user

Solutions with Kubernetes

- - Allocating entire EC2 instance for user
- + Running multiple containers on the same host

Solutions with Kubernetes

- Cleaning up after session ends
  - Terminating EC2 instance
  - Waiting for user’s EBS volume to be detached

Solutions with Kubernetes

- Cleaning up after session ends
  - Terminating EC2 instance
  - Waiting for user’s EBS volume to be detached
+ Killing the container

Solutions with Kubernetes

- Getting different hostname per session

Solutions with Kubernetes

- Getting different hostname per session
+ Using CoreDNS and a proxy to resolve hostnames to private IPs

Solutions with Kubernetes

- Removing instances temporarily to update the Docker image

Solutions with Kubernetes

- Removing instances temporarily to update the Docker image
+ Pulling images using a DaemonSet

Solutions with Portworx

- Provisioning a volume per user wasn’t cost-effective

Solutions with Portworx

- Provisioning a volume per user wasn’t cost-effective
- + Provisioning storage thinly
- + Taking snapshots to S3

Solutions with Portworx

- Assigning availability zone to each user was limiting

Solutions with Portworx

- Assigning availability zone to each user was limiting
+ Abstracting away EBS provisioning

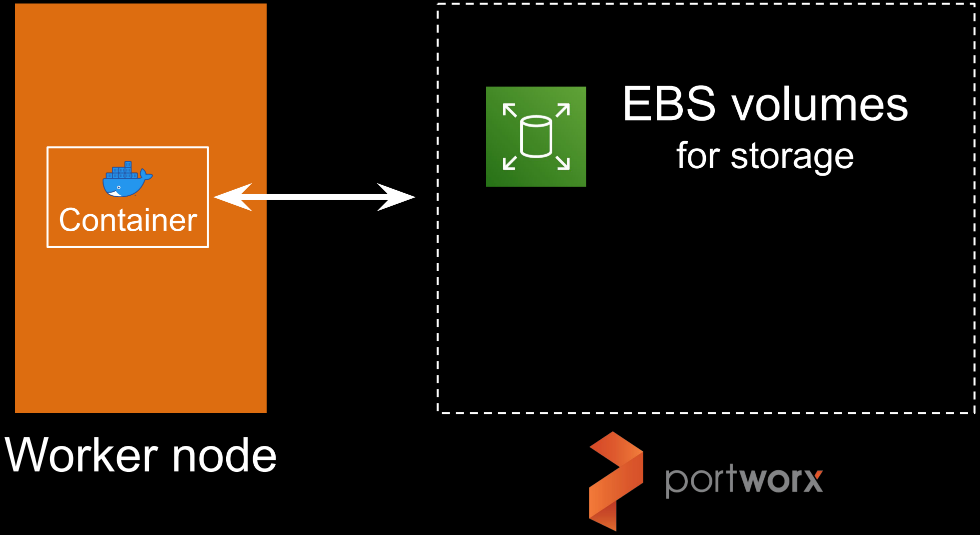
Solutions with Portworx

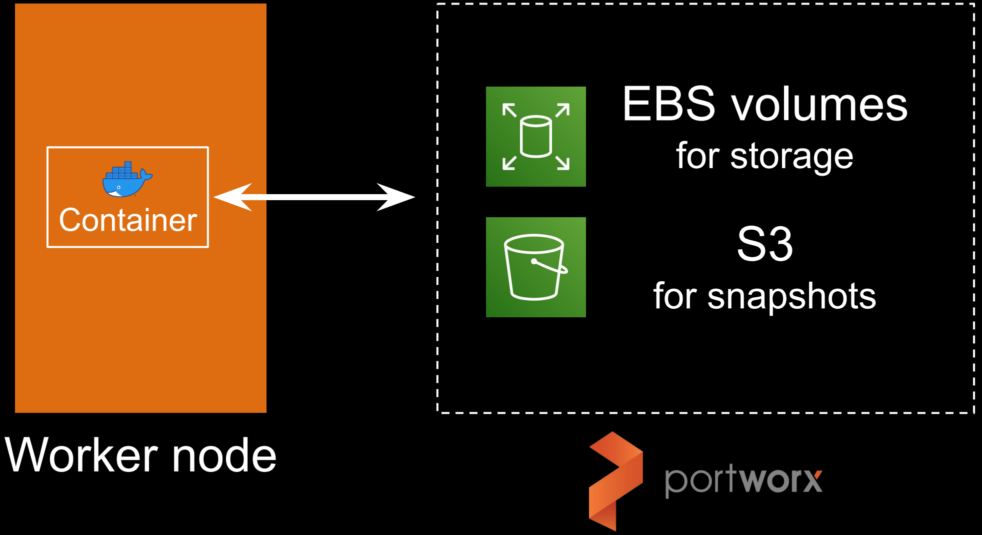
- Setting up and refreshing credentials on the EC2 instance added complexity
- Downloading user’s data initially was slow
- Uploading user’s data periodically was fragile
- Limiting storage size per user wasn’t easy

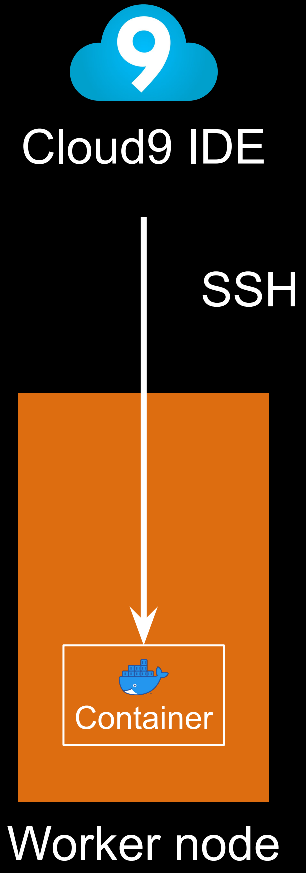
Solutions with Portworx

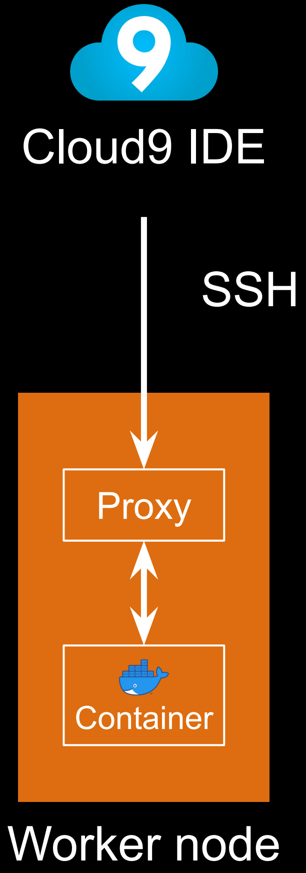
- Setting up and refreshing credentials on the EC2 instance added complexity
- Downloading user’s data initially was slow
- Uploading user’s data periodically was fragile
- Limiting storage size per user wasn’t easy

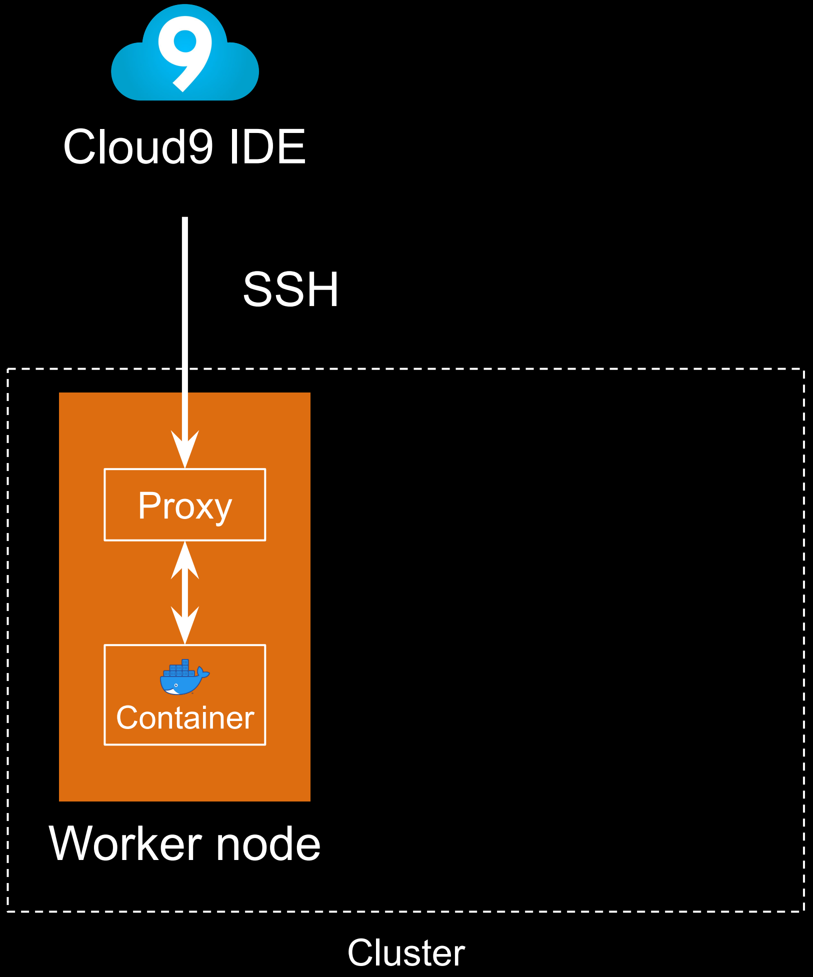
+ Using Portworx volumes

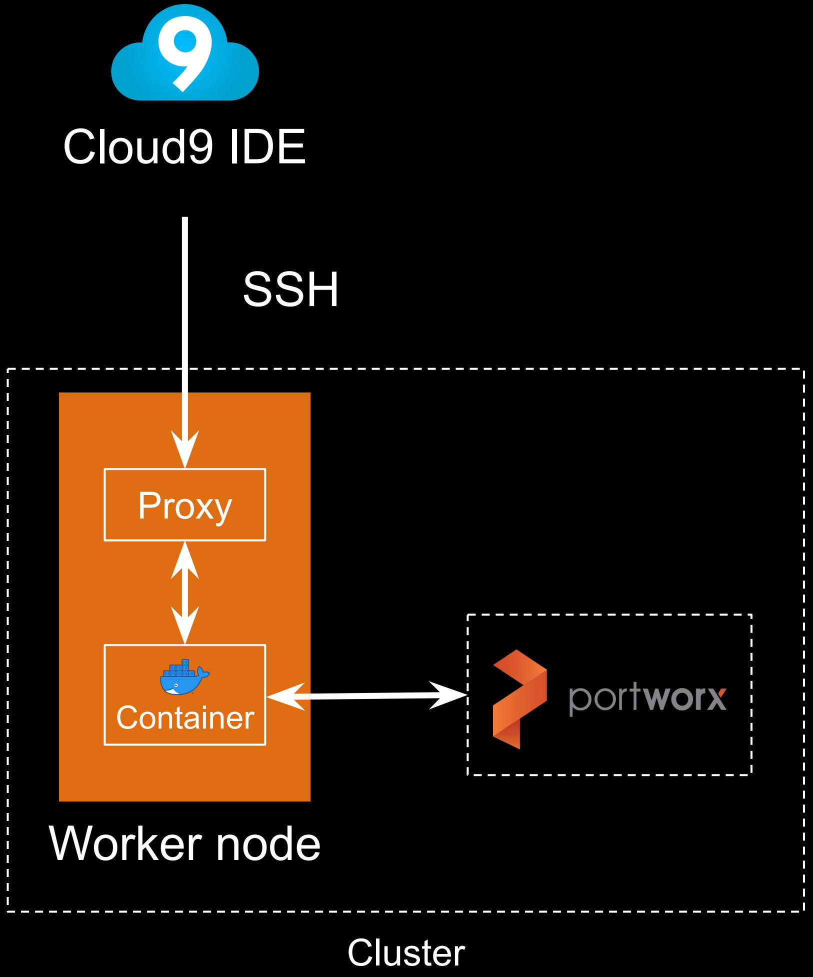

Container
Worker node
EBS volumes
for storage
portworx

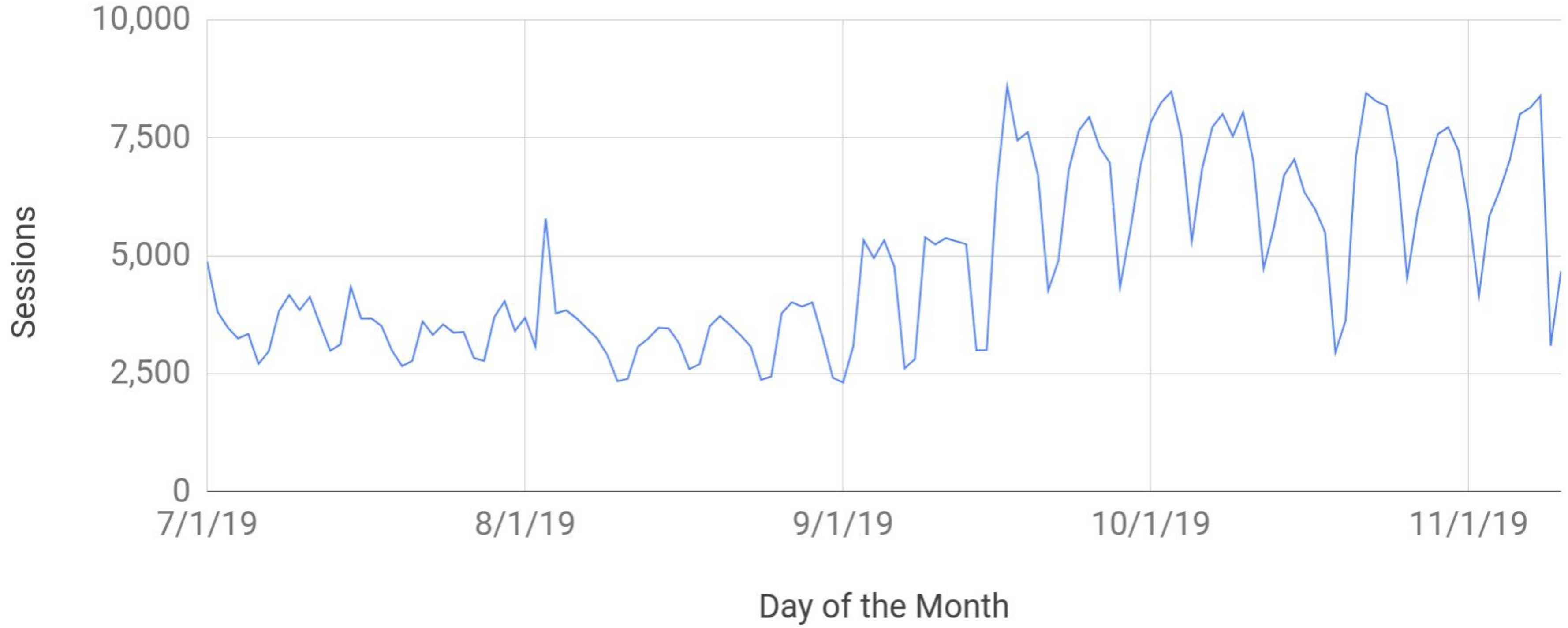

Container
Worker node
EBS volumes
for storage
S3
for snapshots
portworx


Cloud9 IDE
SSH
Container
Worker node


Cloud9 IDE
SSH
Proxy
Container
Worker node


Cloud9 IDE
SSH
Proxy
Container
Worker node
Cluster


Cloud9 IDE
SSH
Proxy
Container
portworx
Worker node
Cluster


10,000
7,500
5,000
2,500
0
Sessions
7/1/19
8/1/19
9/1/19
10/1/19
11/1/19
Day of the Month

Future Work

- Improved fraud detection and prevention
- Multiple clusters in different regions
- Multiple IDEs per user
- ...

Education as a Service

Containerization and Orchestration of CS50 IDE

David J. Malan
malan@harvard.edu

Kareem Zidane
kzidane@cs50.harvard.edu

cs50.ly/kubecon