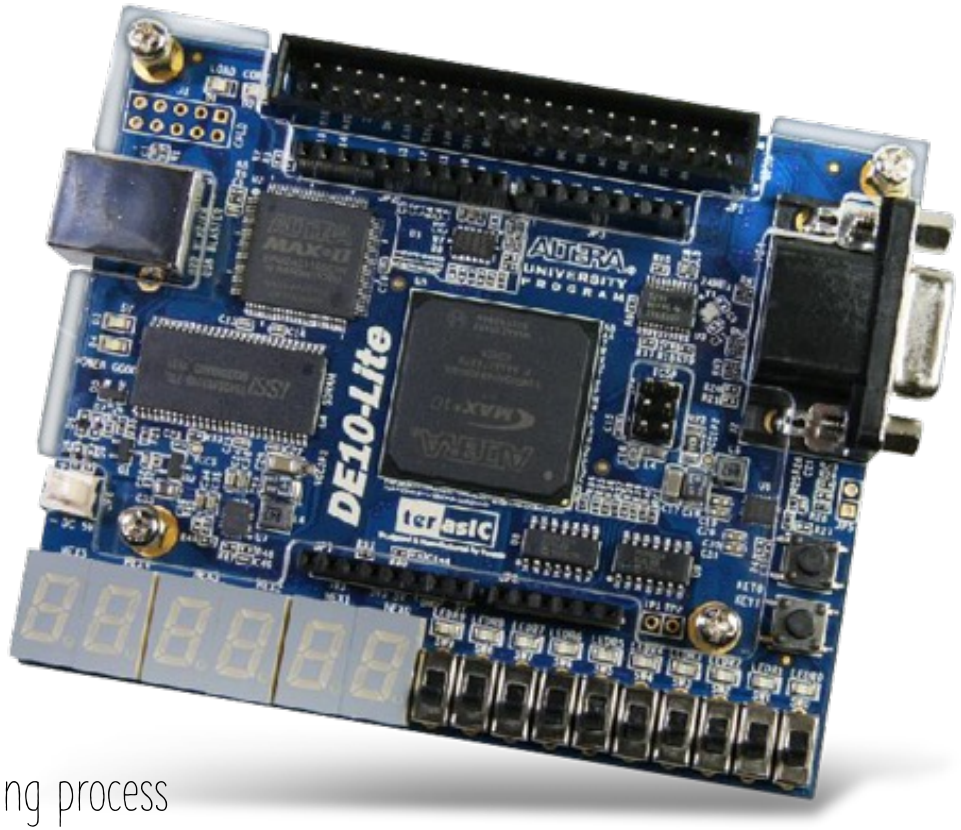
Digital Circuit Design

Li Bai

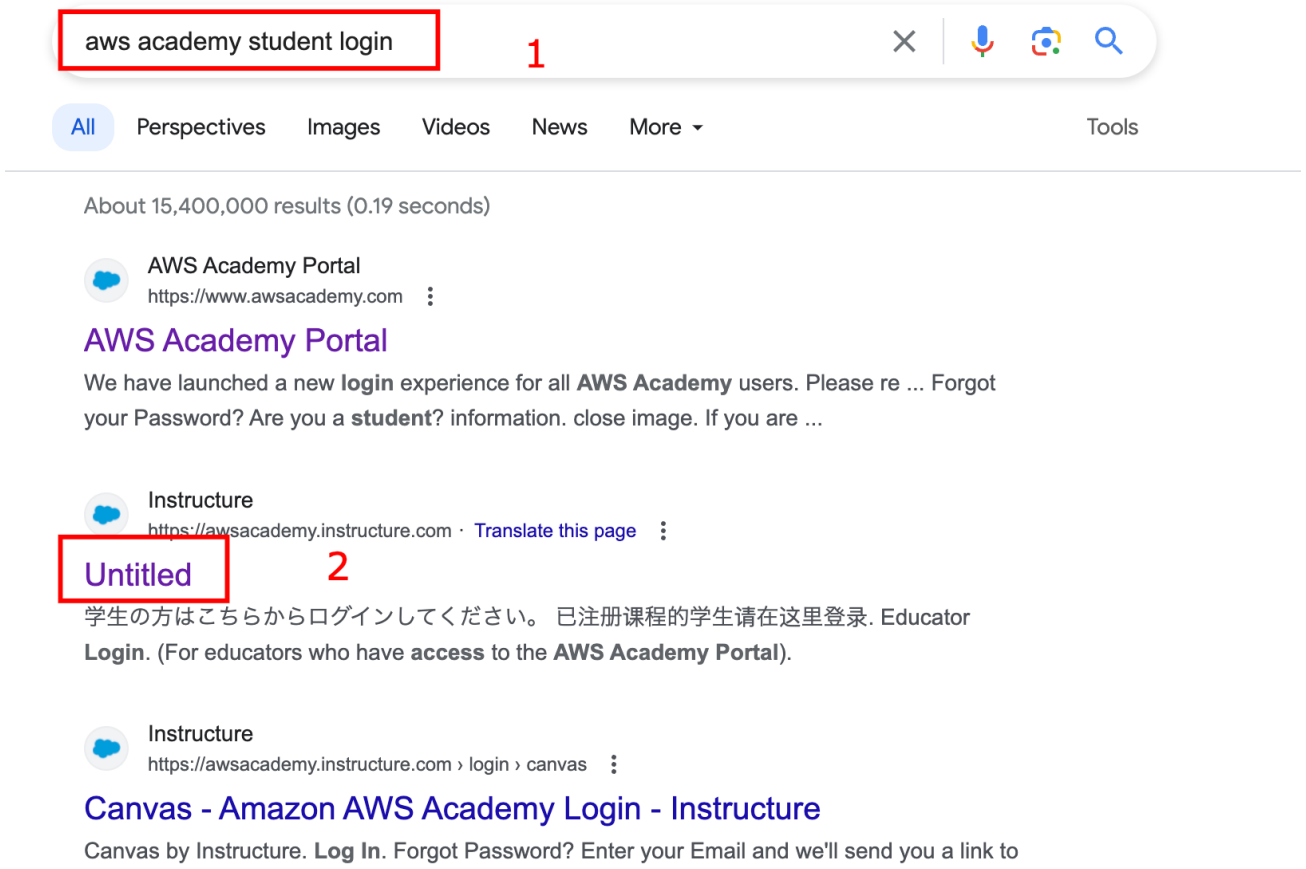
What we will cover...

- Digital (we will need to learn everything about the binary ...)
- We will need to design a process based on input/state to move into different output
- Difference between Arduino and DE10-Lite
  - Advantage of FPGA -> check out here
- Quartus and EDA Playground
  - Quartus will be on AWS Academy
  - https://edaplayground.com/

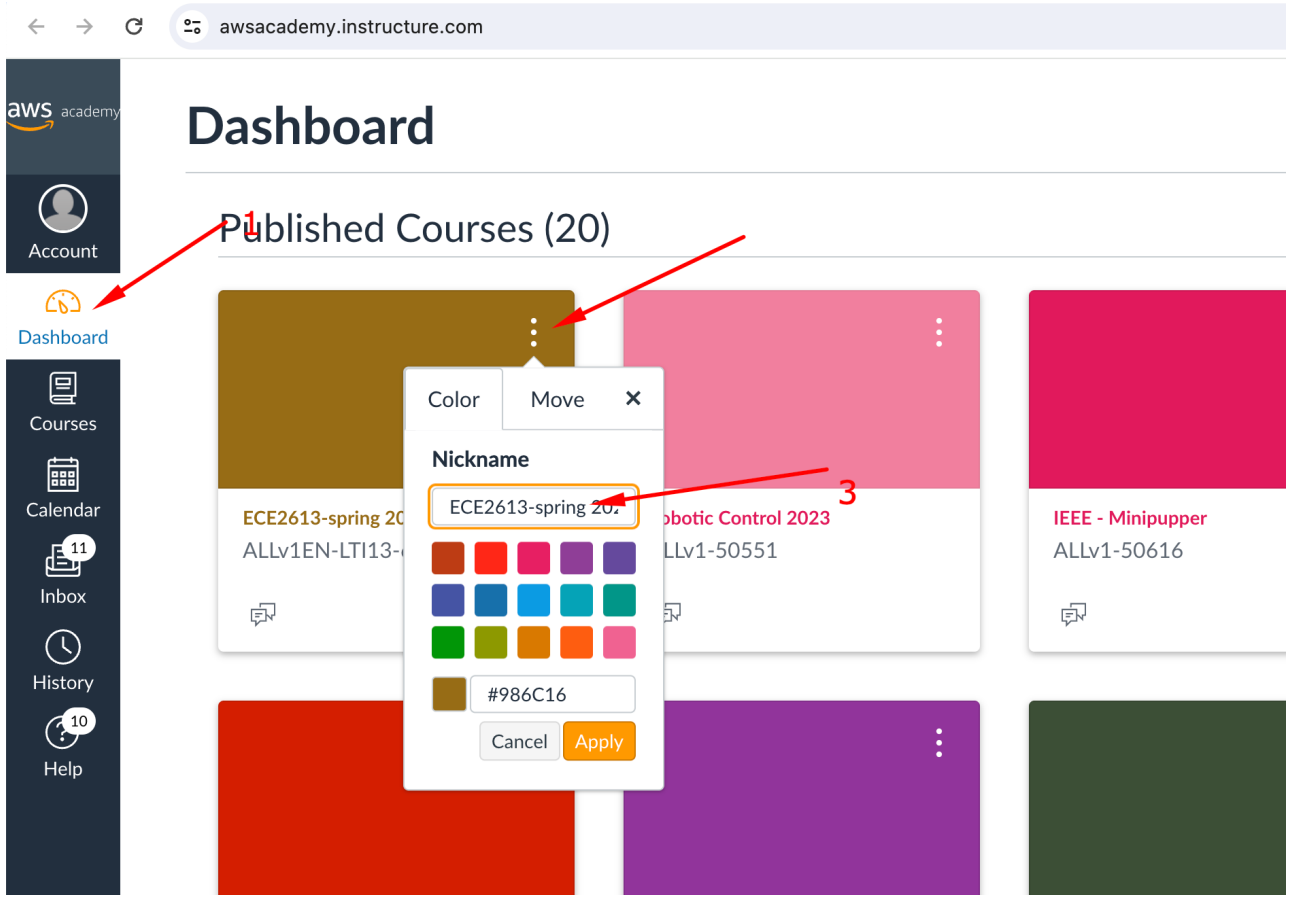


- Learn combinatory logic, sequential logic, flip-flop, block diagram, timing diagram, making process

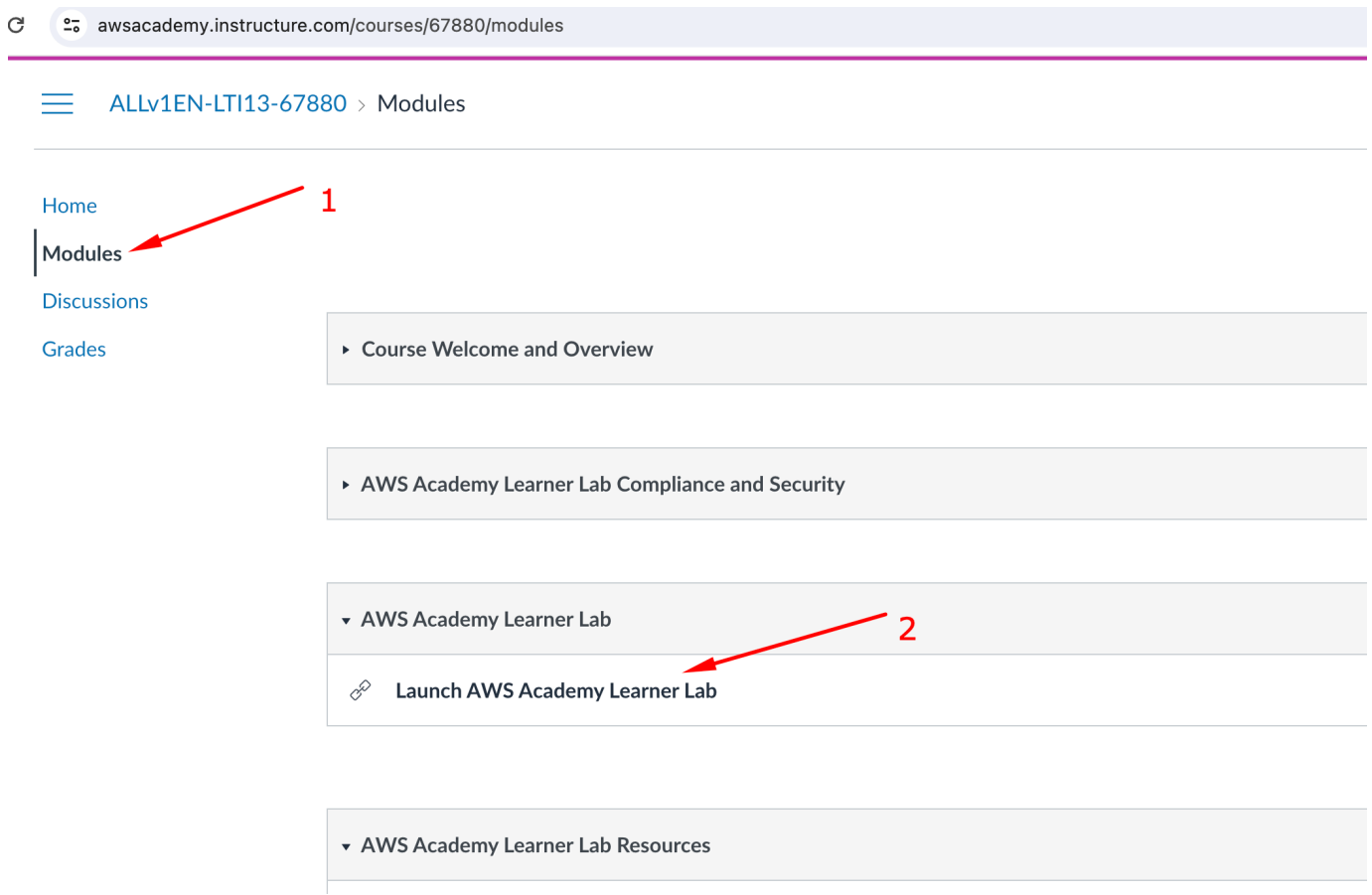
AWS Academy Student login



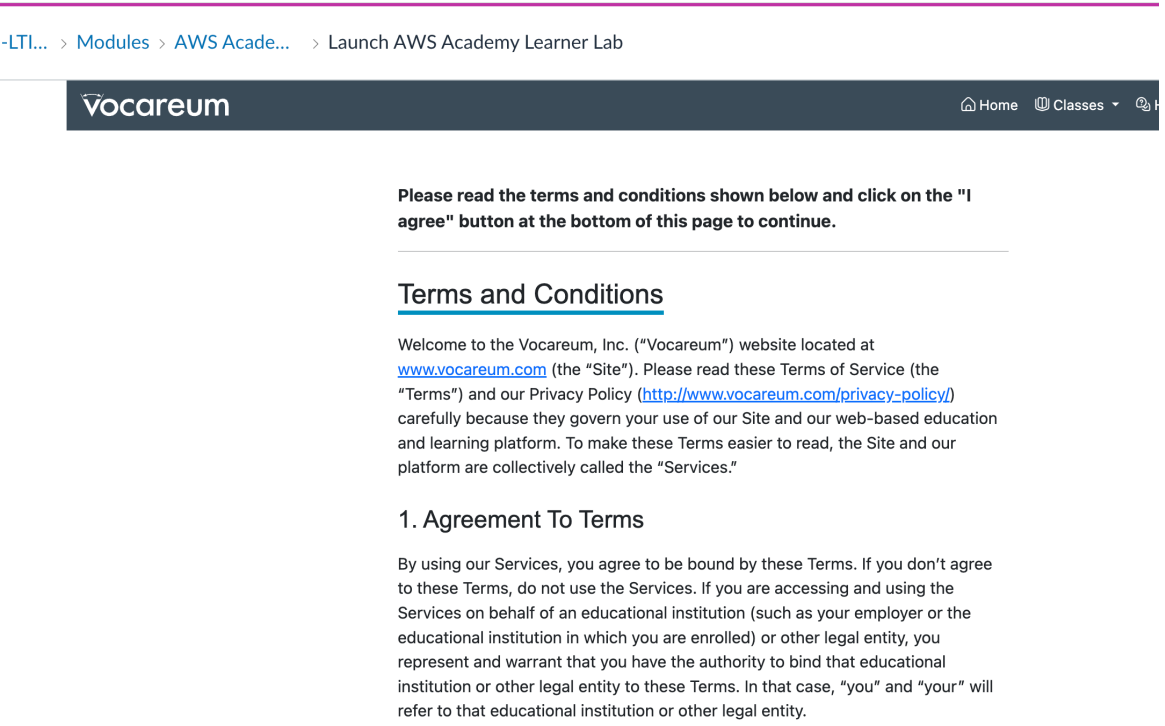
AWS Academy Canvas

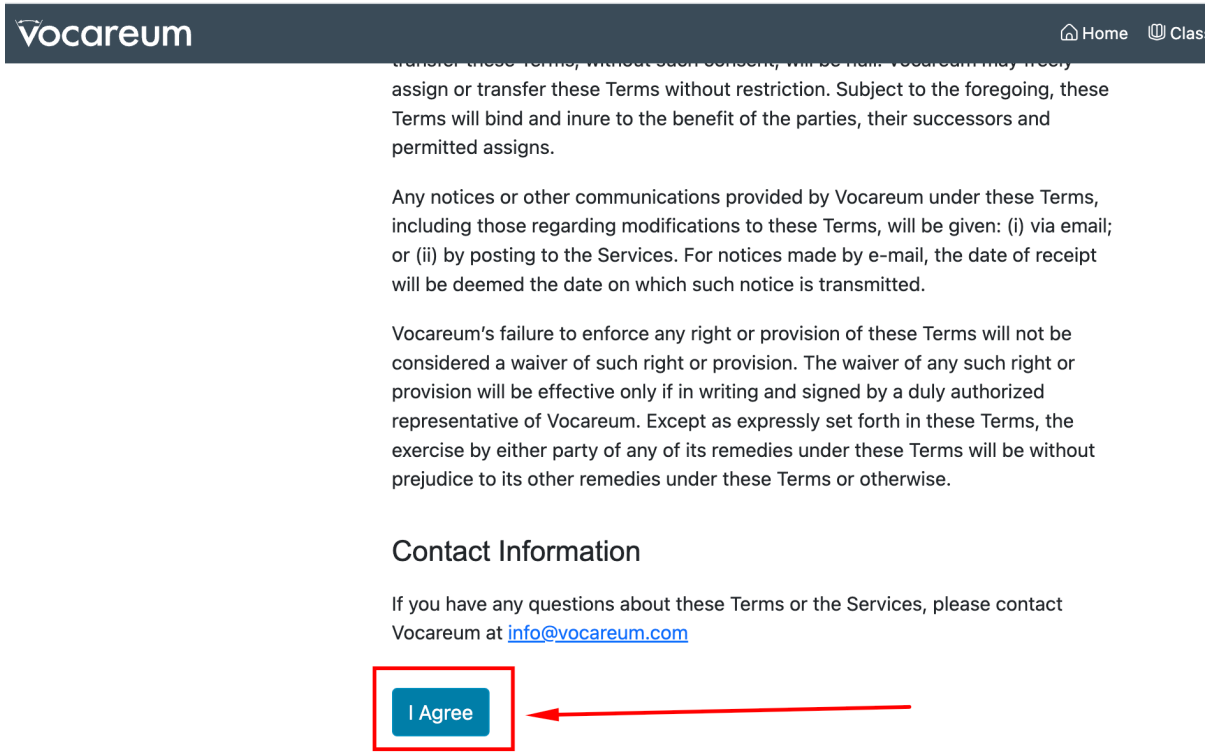


Access Learner's Lab

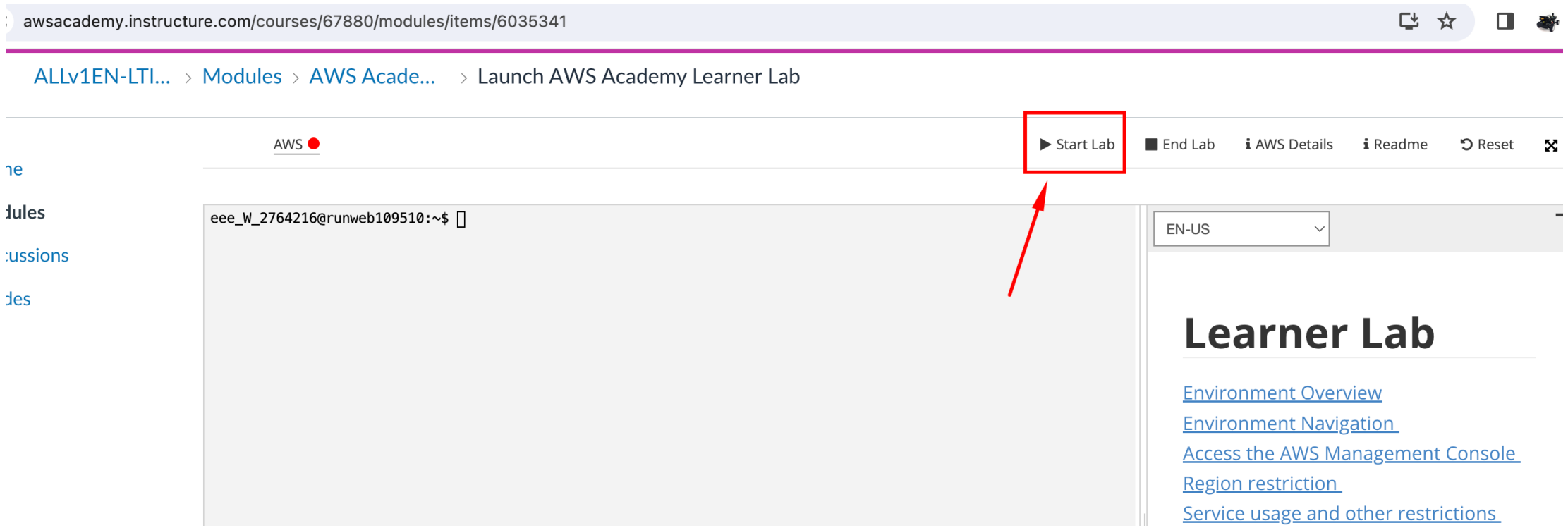


Accept terms and conditions

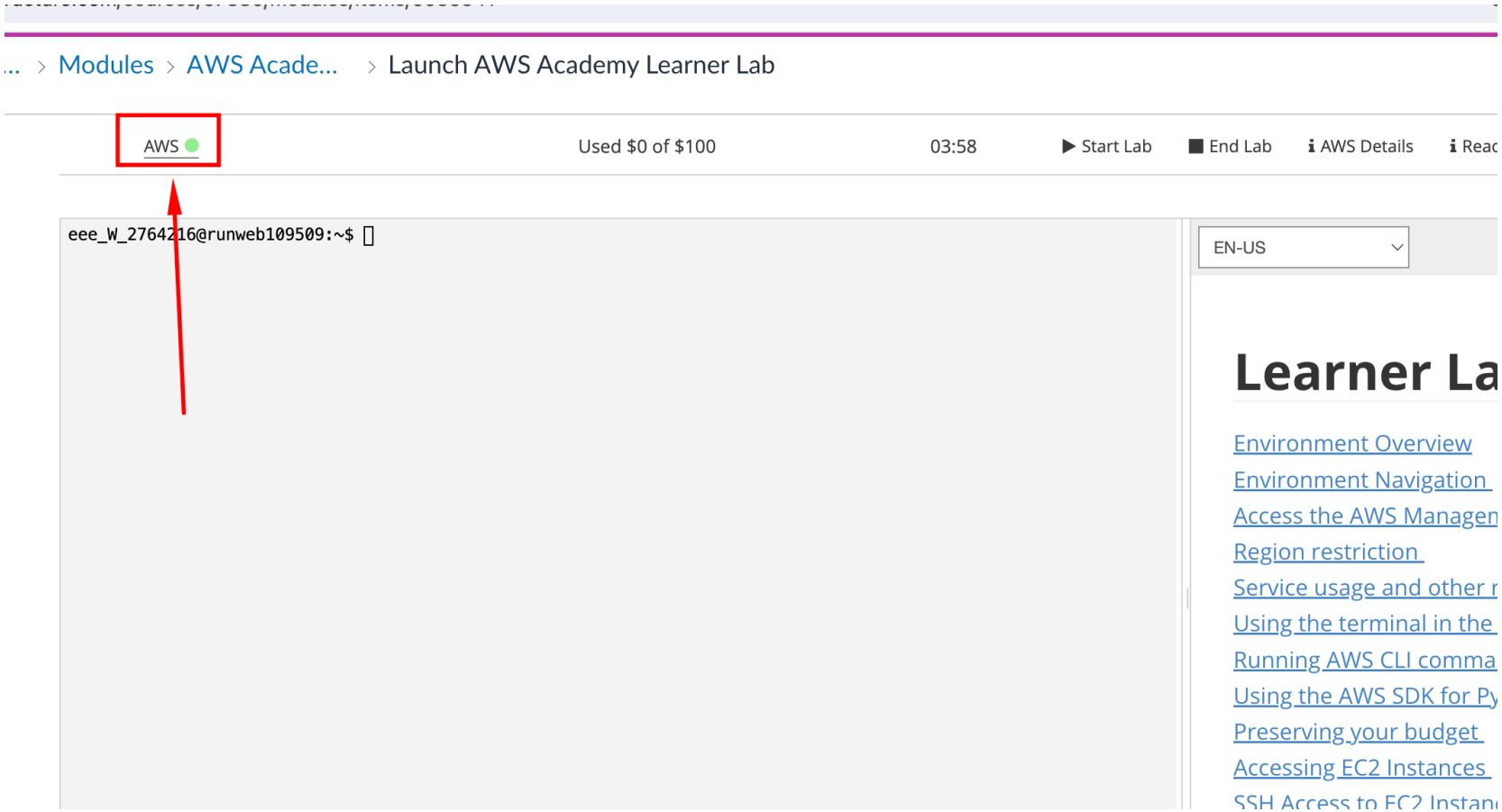




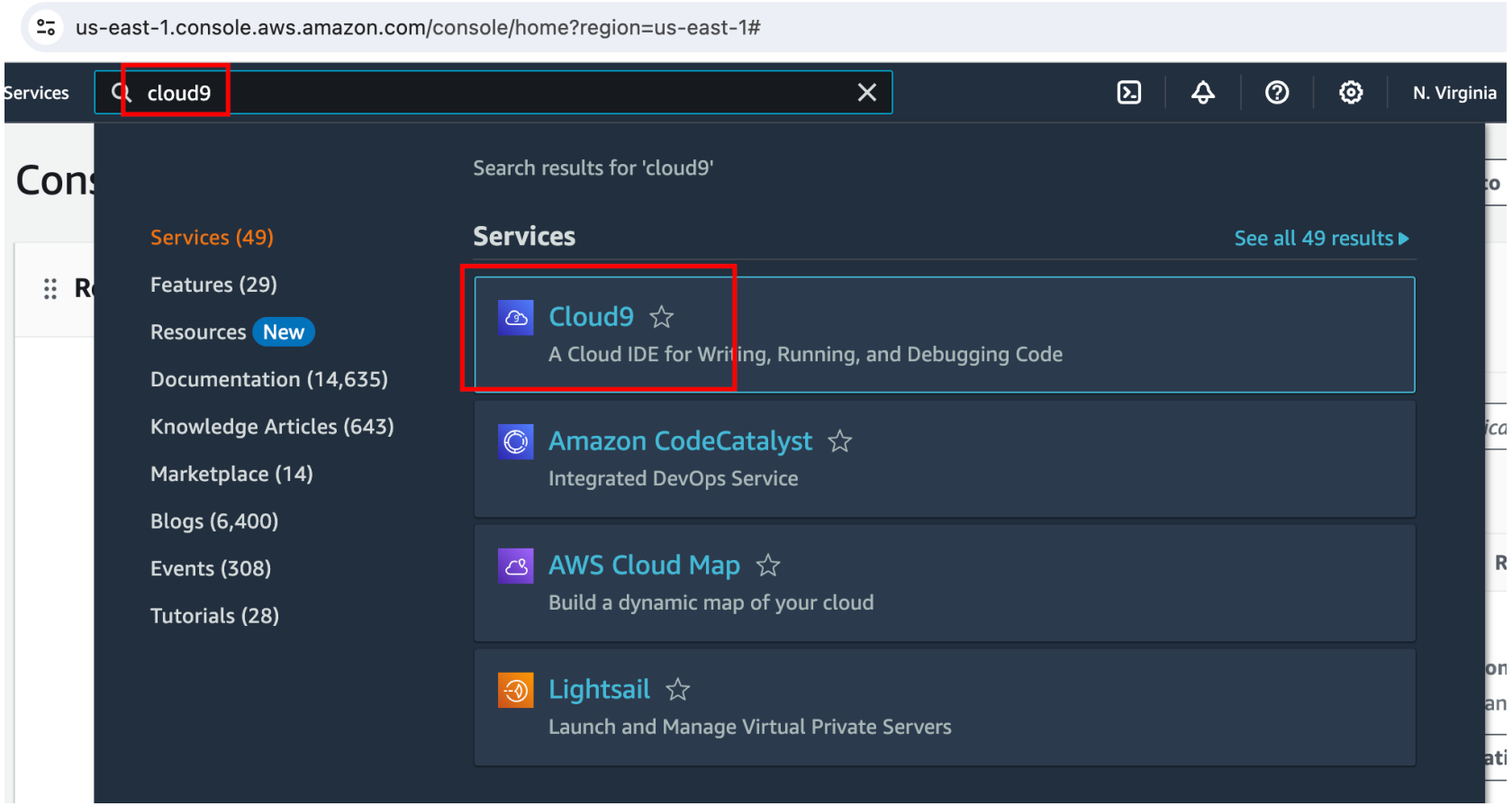
Start lab


awsacademy.instructure.com/courses/67880/modules/items/6035341
ALLv1EN-LTI... > Modules > AWS Acade... > Launch AWS Academy Learner Lab
AWS
Start Lab
End Lab
AWS Details
Readme
Reset
eee_W_2764216@runweb109510:~$
EN-US
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions

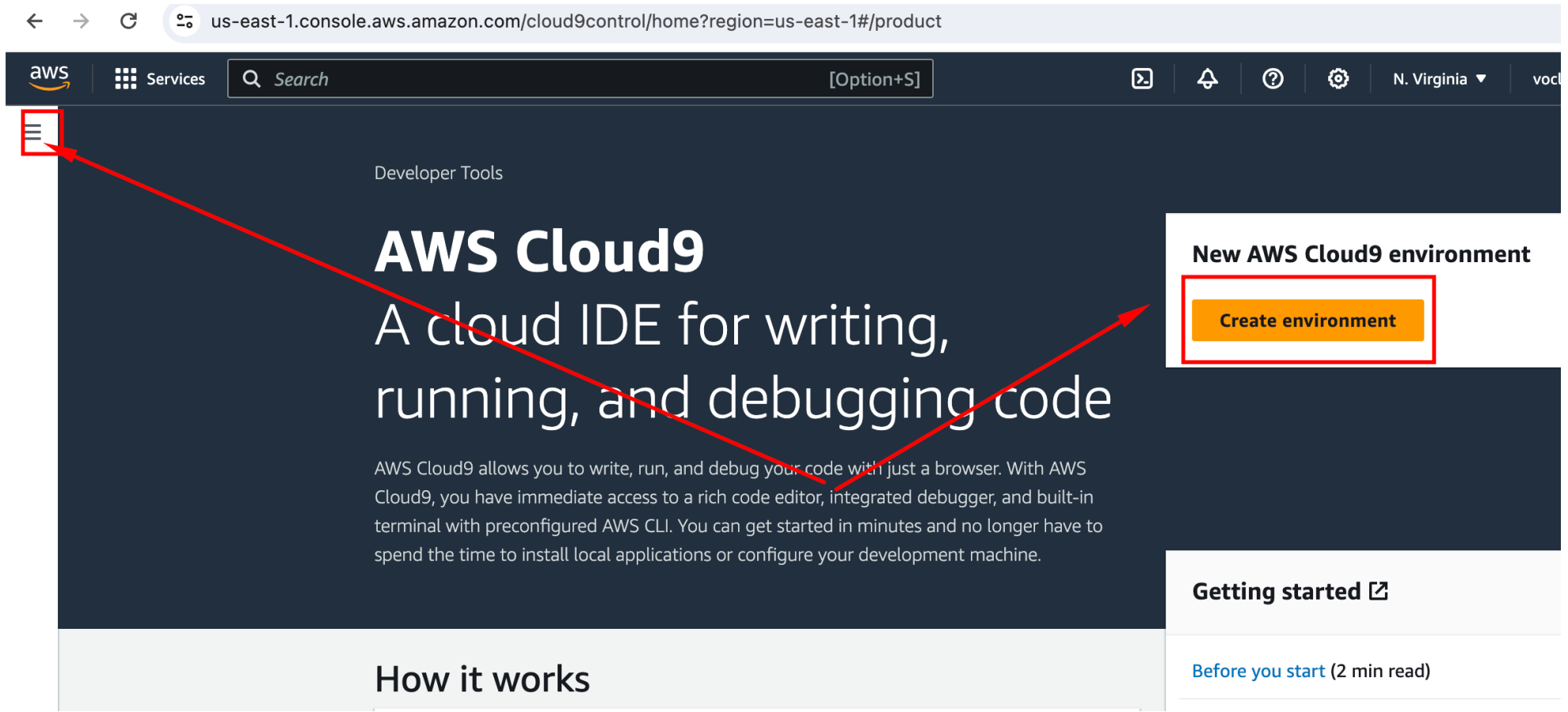
AWS green -> go to console



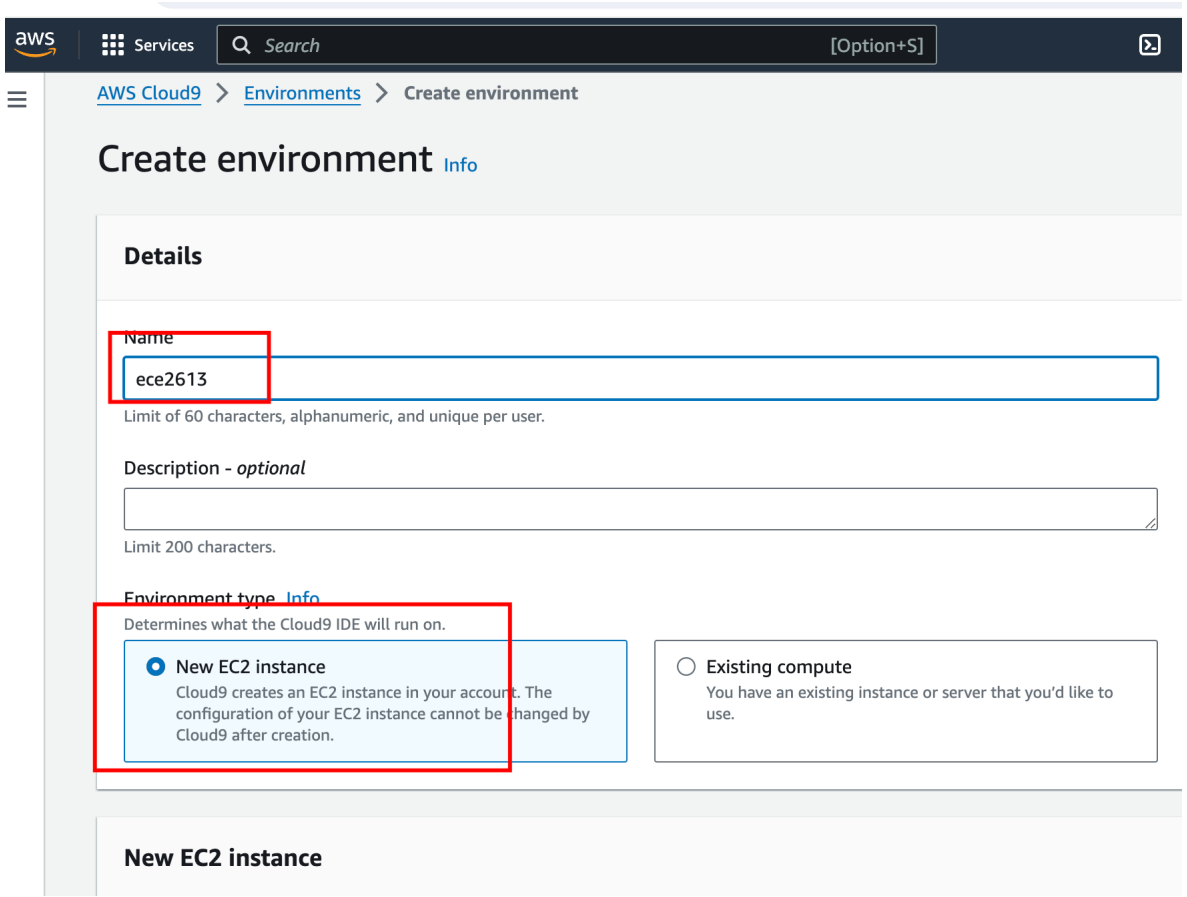
Find cloud9

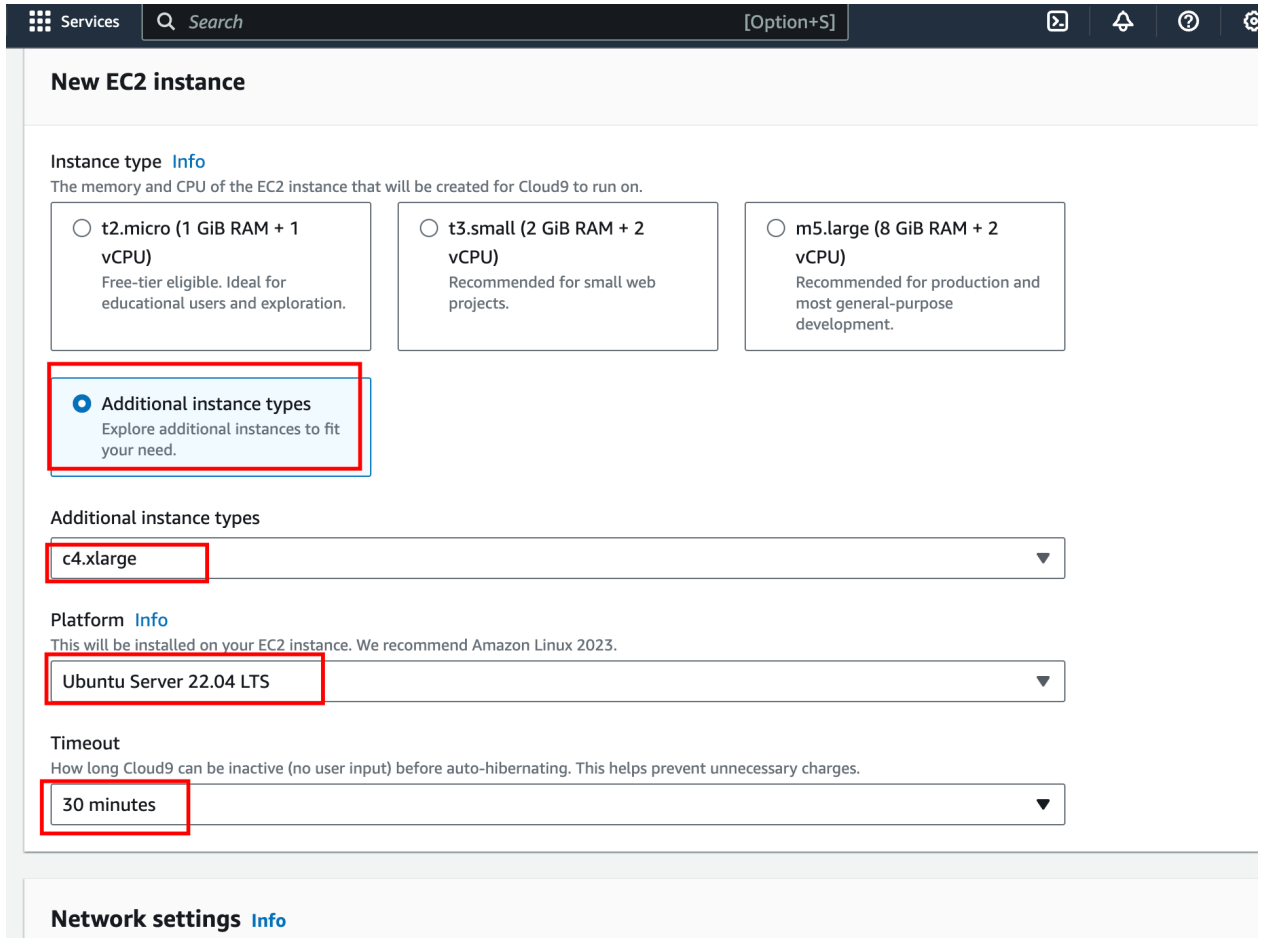

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#
Services
cloud9
N. Virginia
Search results for 'cloud9'
Services (49)
Features (29)
Resources New
Documentation (14,635)
Knowledge Articles (643)
Marketplace (14)
Blogs (6,400)
Events (308)
Tutorials (28)
Services
See all 49 results
Cloud9
A Cloud IDE for Writing, Running, and Debugging Code
Amazon CodeCatalyst
Integrated DevOps Service
AWS Cloud Map
Build a dynamic map of your cloud
Lightsail
Launch and Manage Virtual Private Servers

Create environment

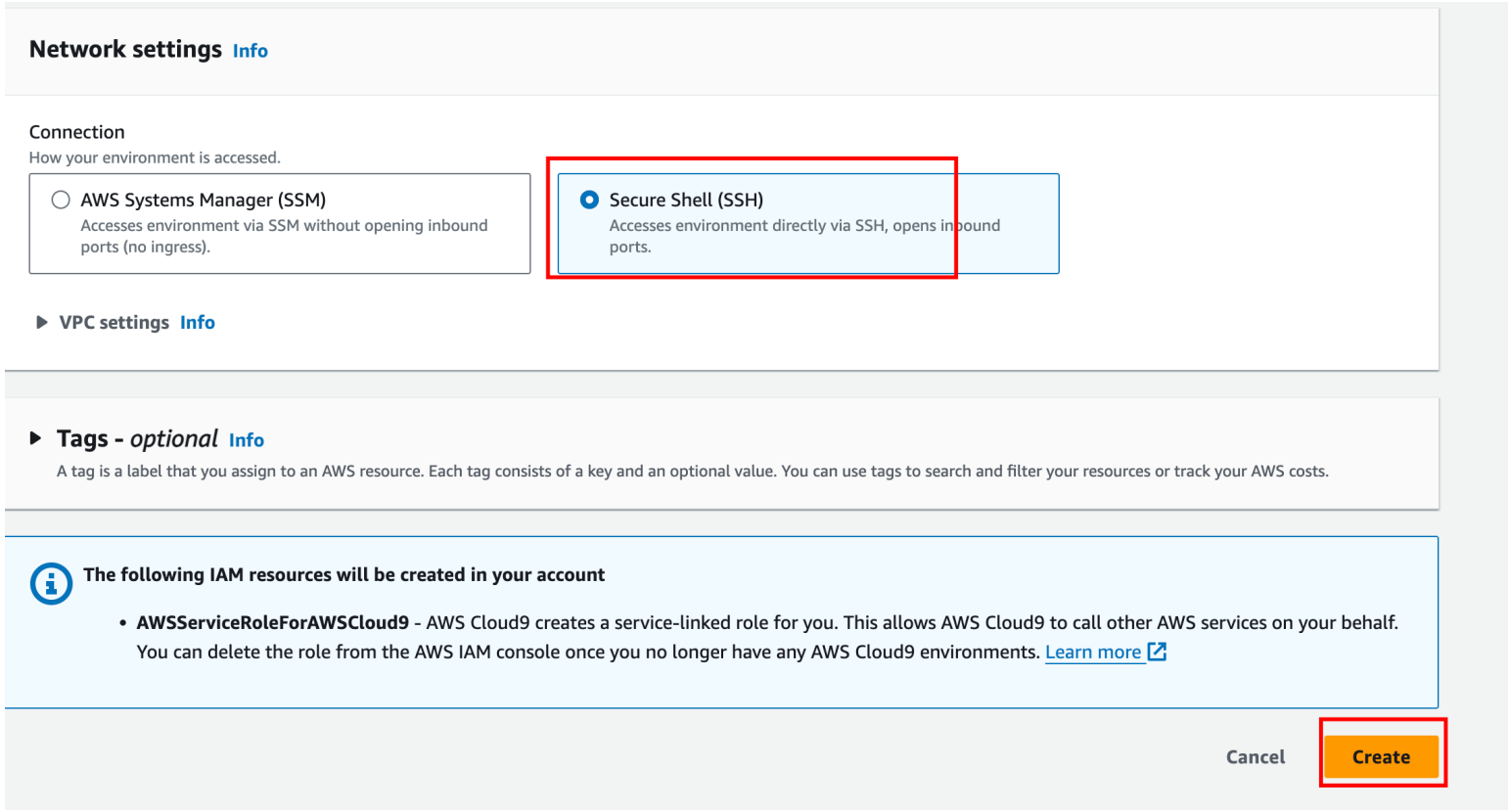


Ubuntu server cloud9 setup



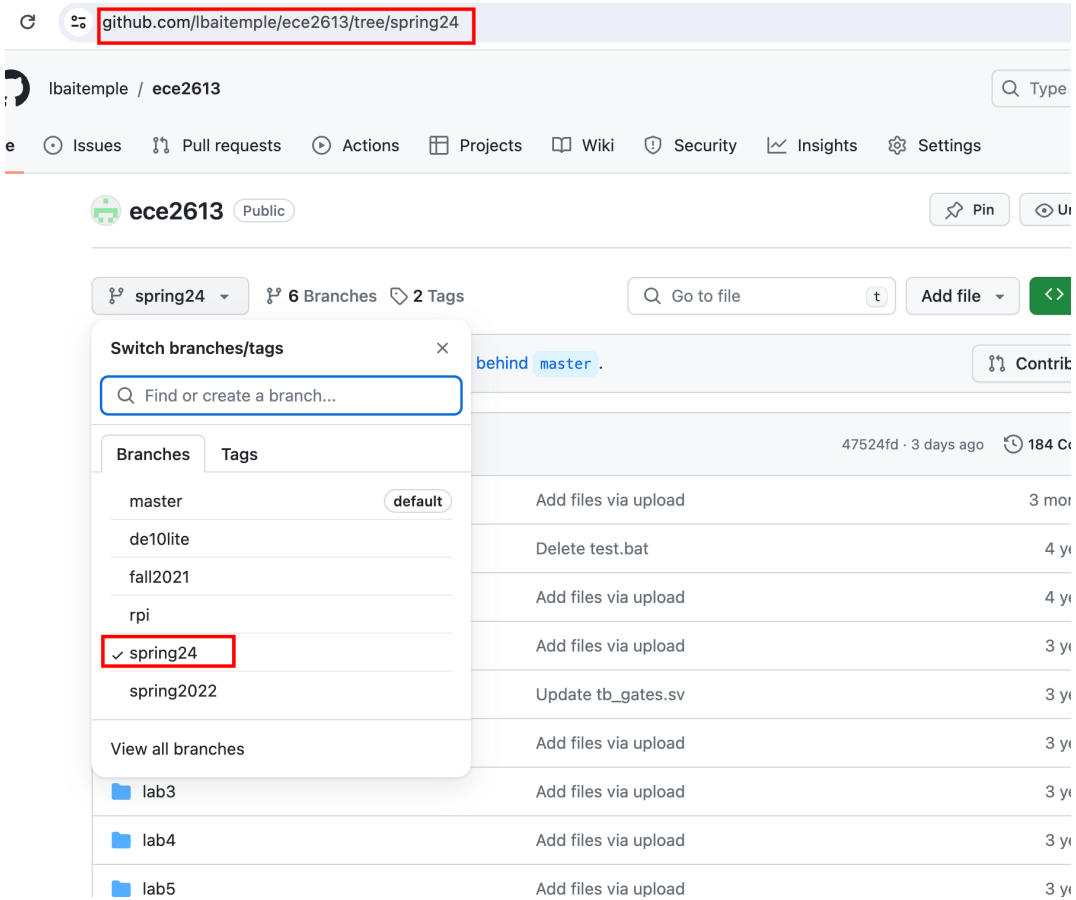


Finalize your cloud9 environment



- c4.xlarge
- Ubunttu22.04
- ssh

Instruction on github



2613_2024s

Instruction is provided at https://sites.google.com/a/temple.edu/ece2612/home/cloud9-setup

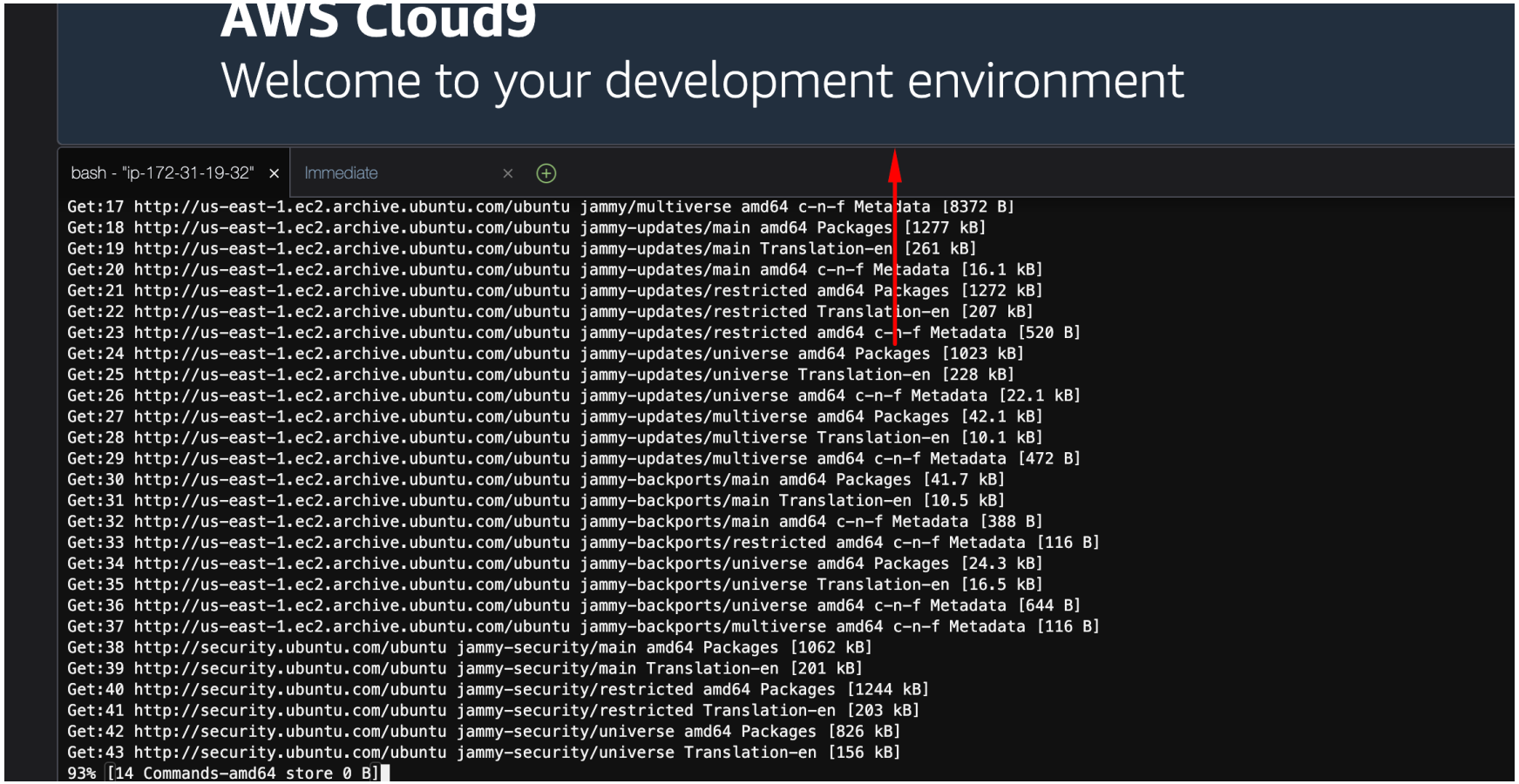
after you select the instance for cloud9, in the terminal

```
git clone -b spring24 https://github.com/lbaitemple/ece2613
cd ece2613
bash ./setup.bash
sudo reboot
```

Test the code

- right click on m_sim (extension file) and run
- right click on qsf (extension file) and run

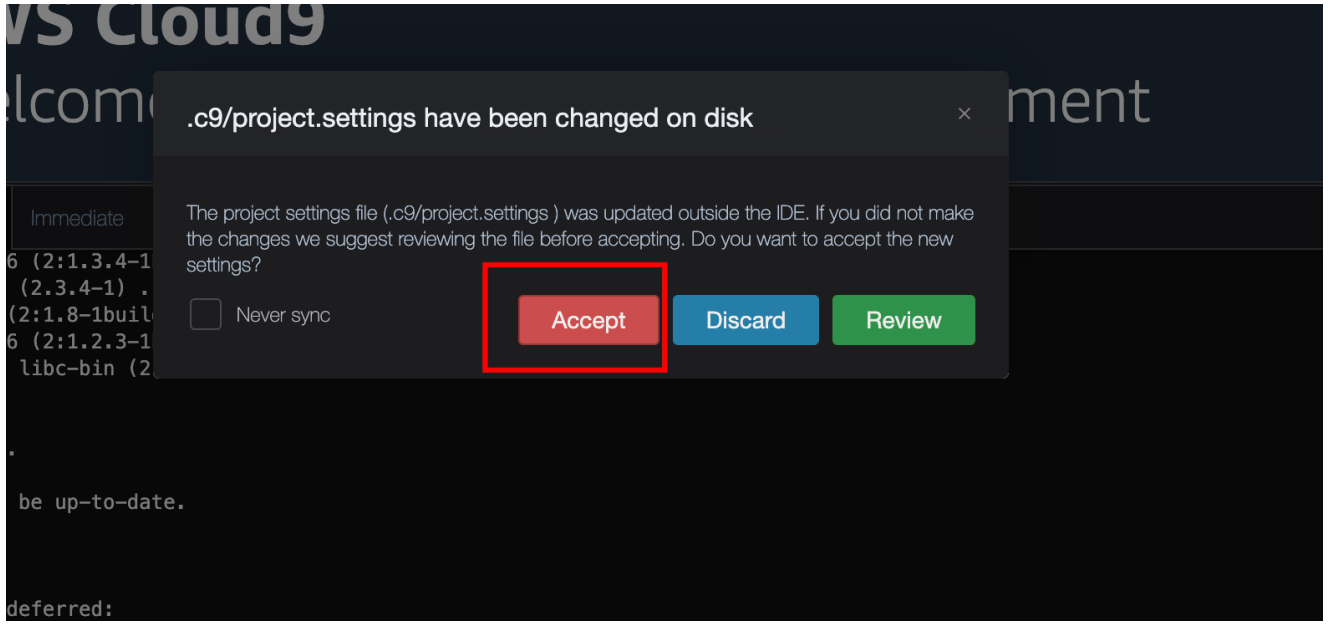
Resize cloud9 terminal windows

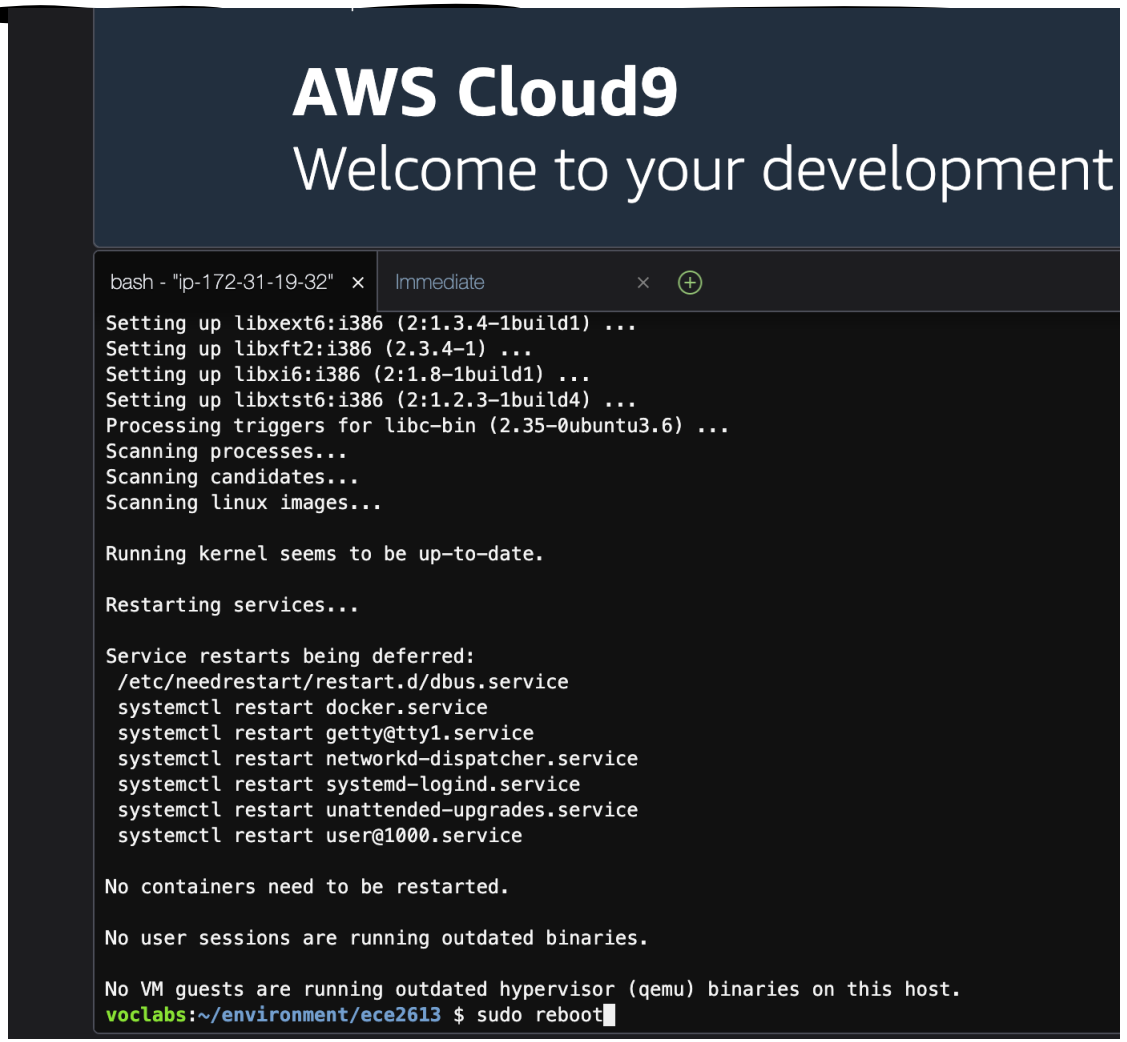


After about 15-20 minutes...

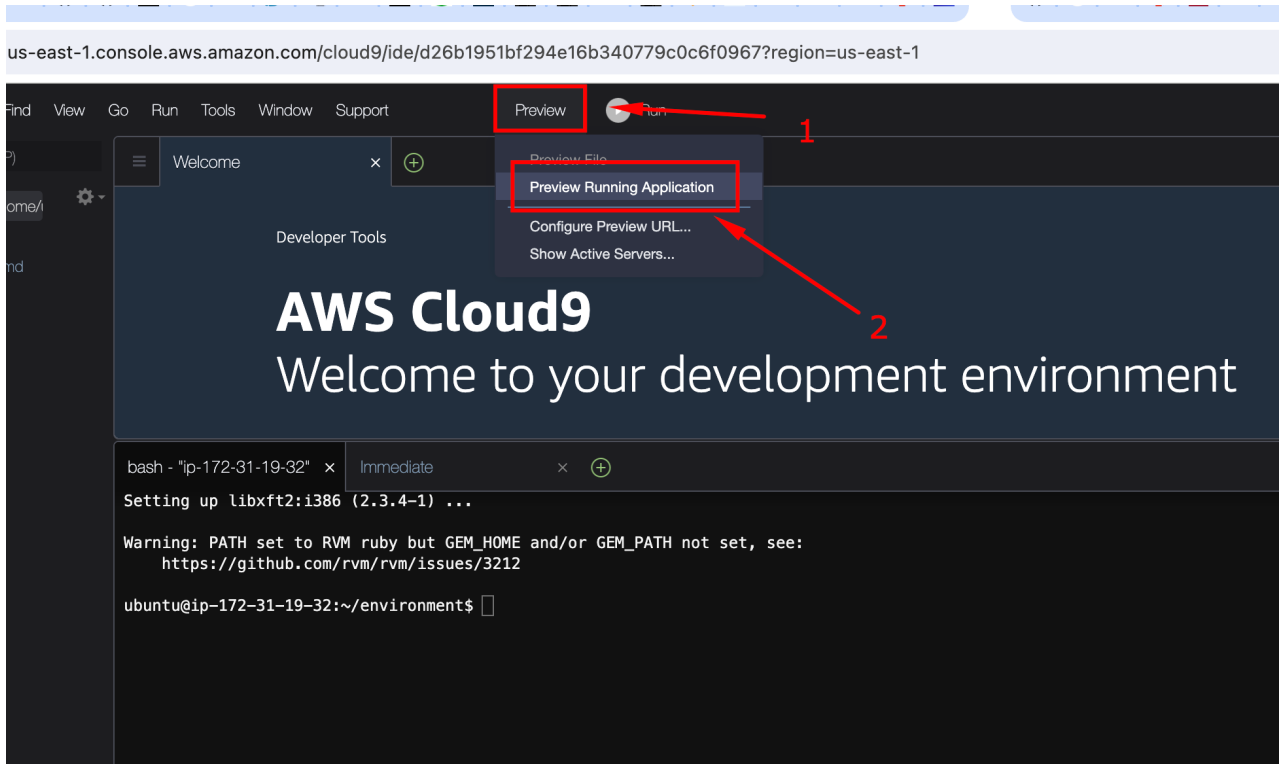
sudo reboot

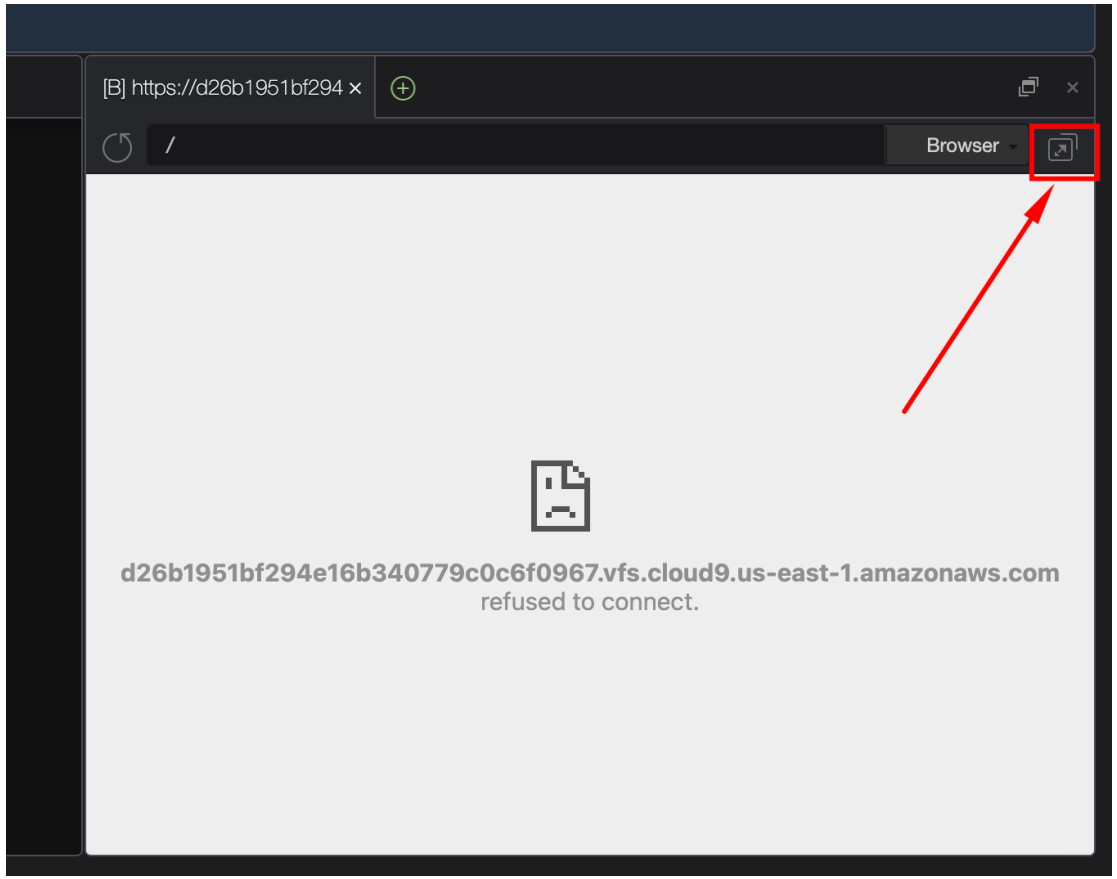
Reboot the system



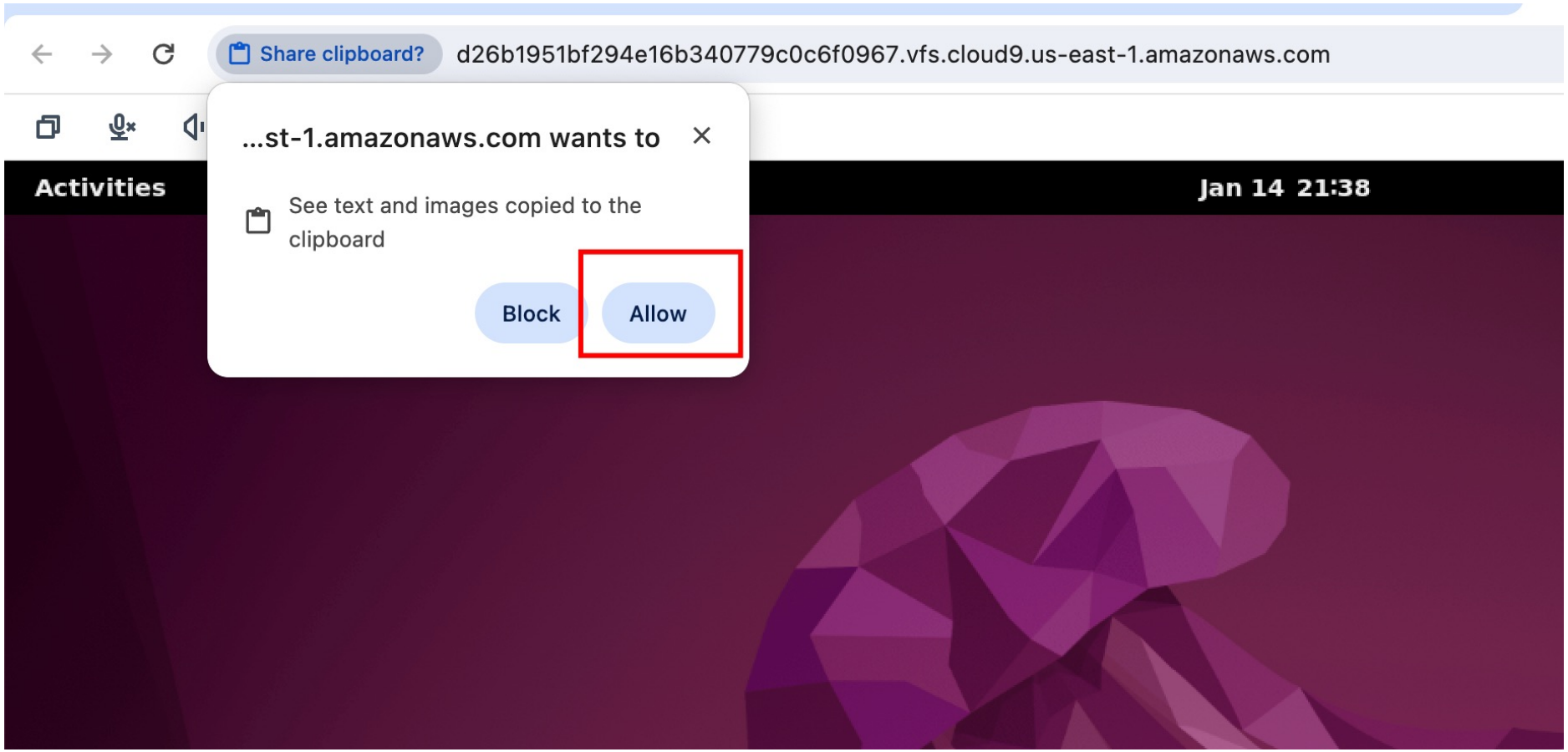


Xwindow access

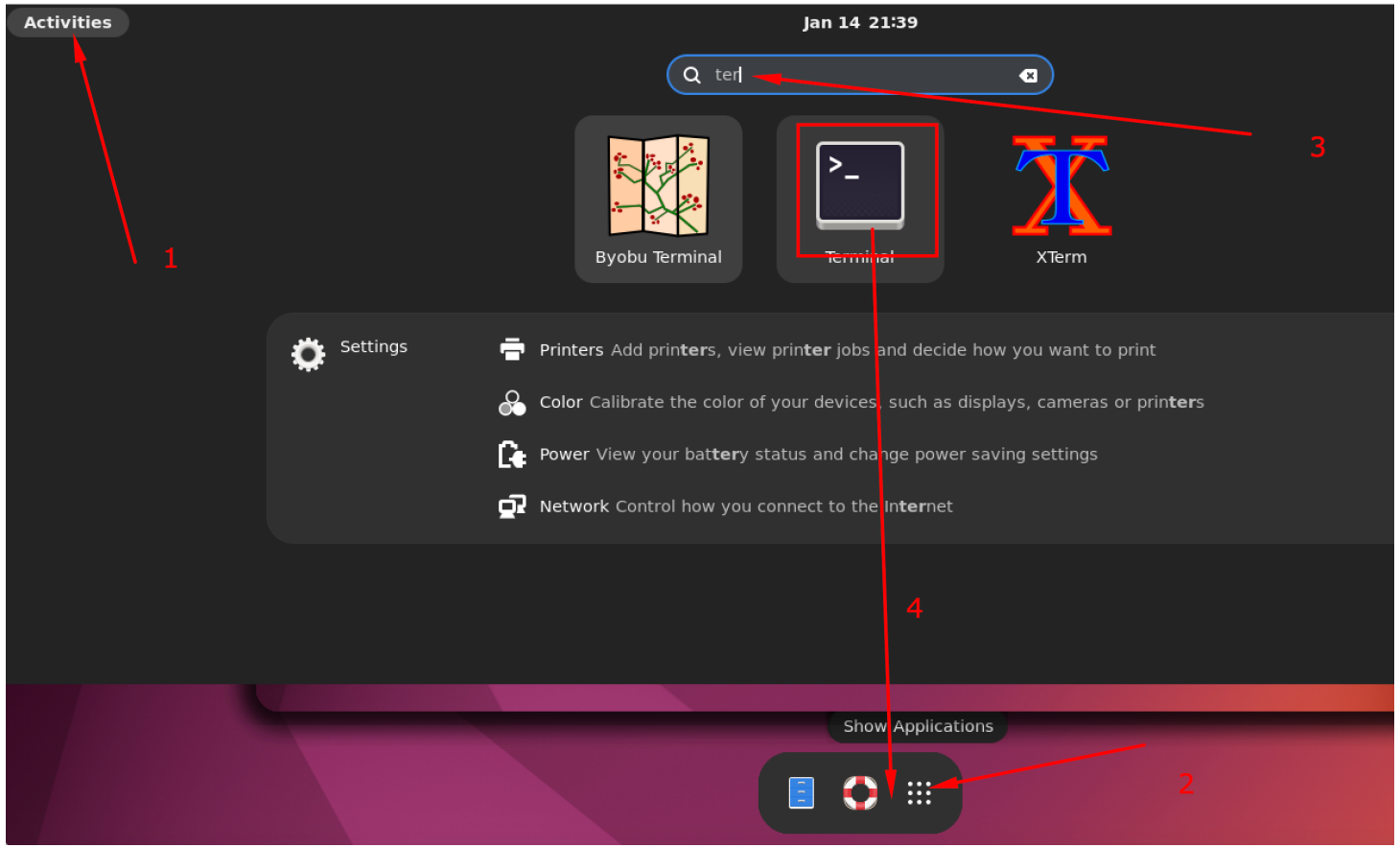




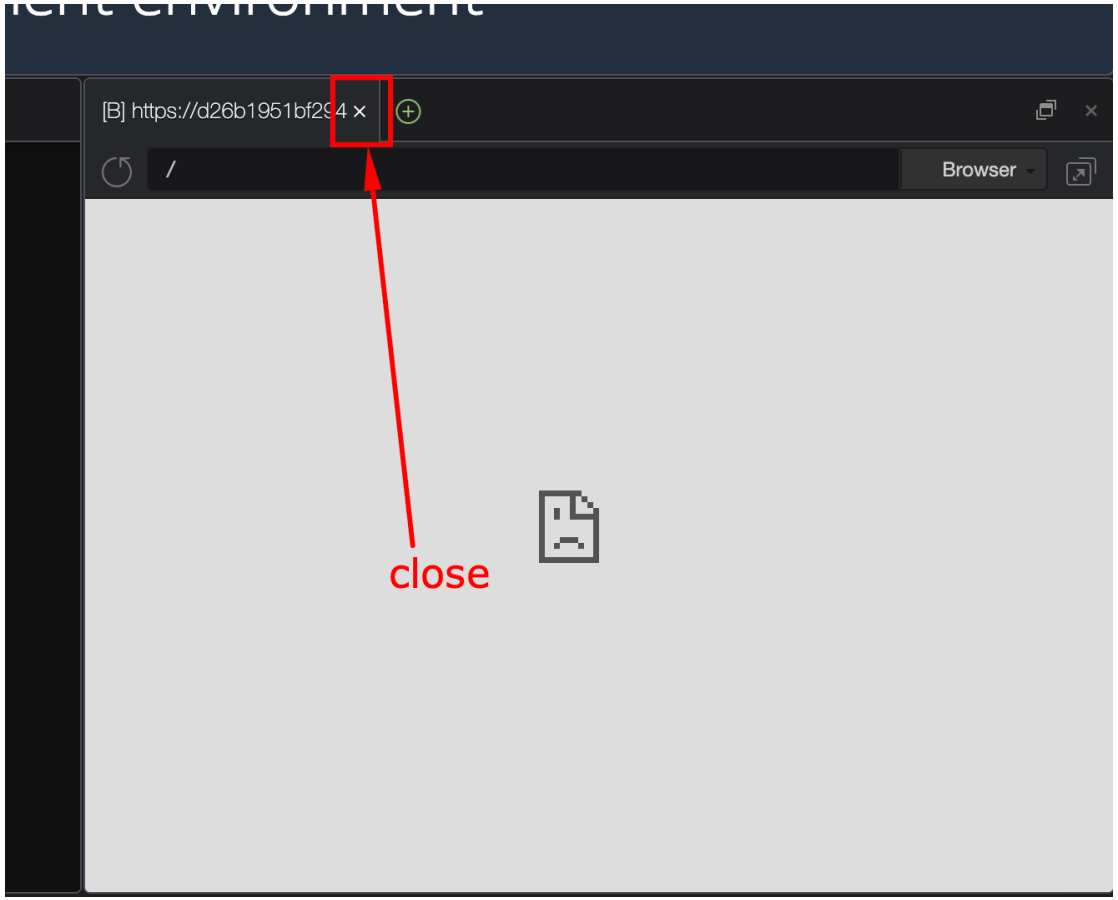
Allow xwindow



Place terminal in task bar

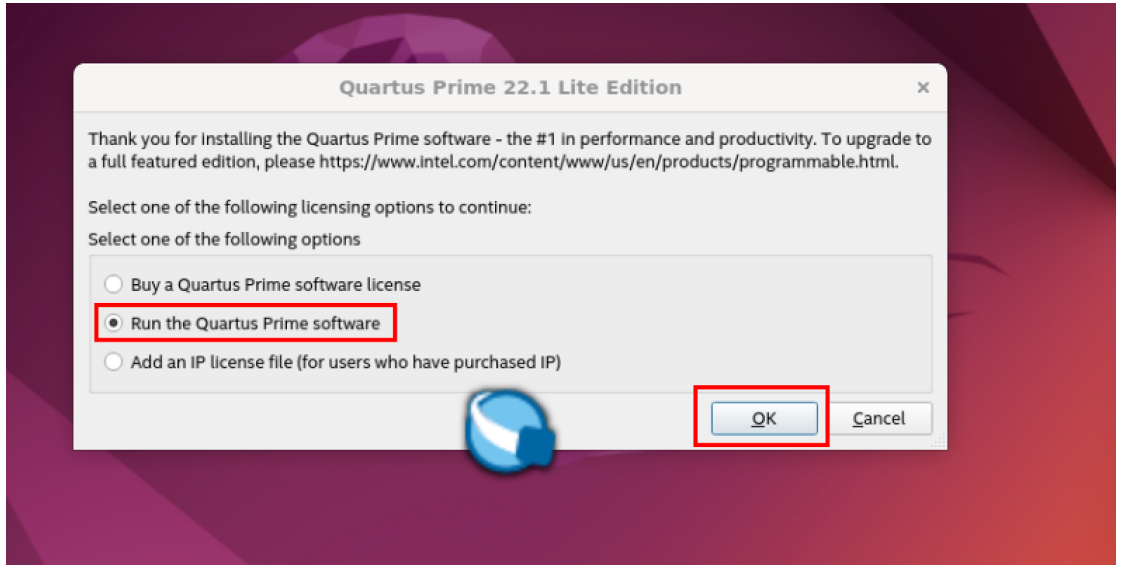


Close access to xwindow

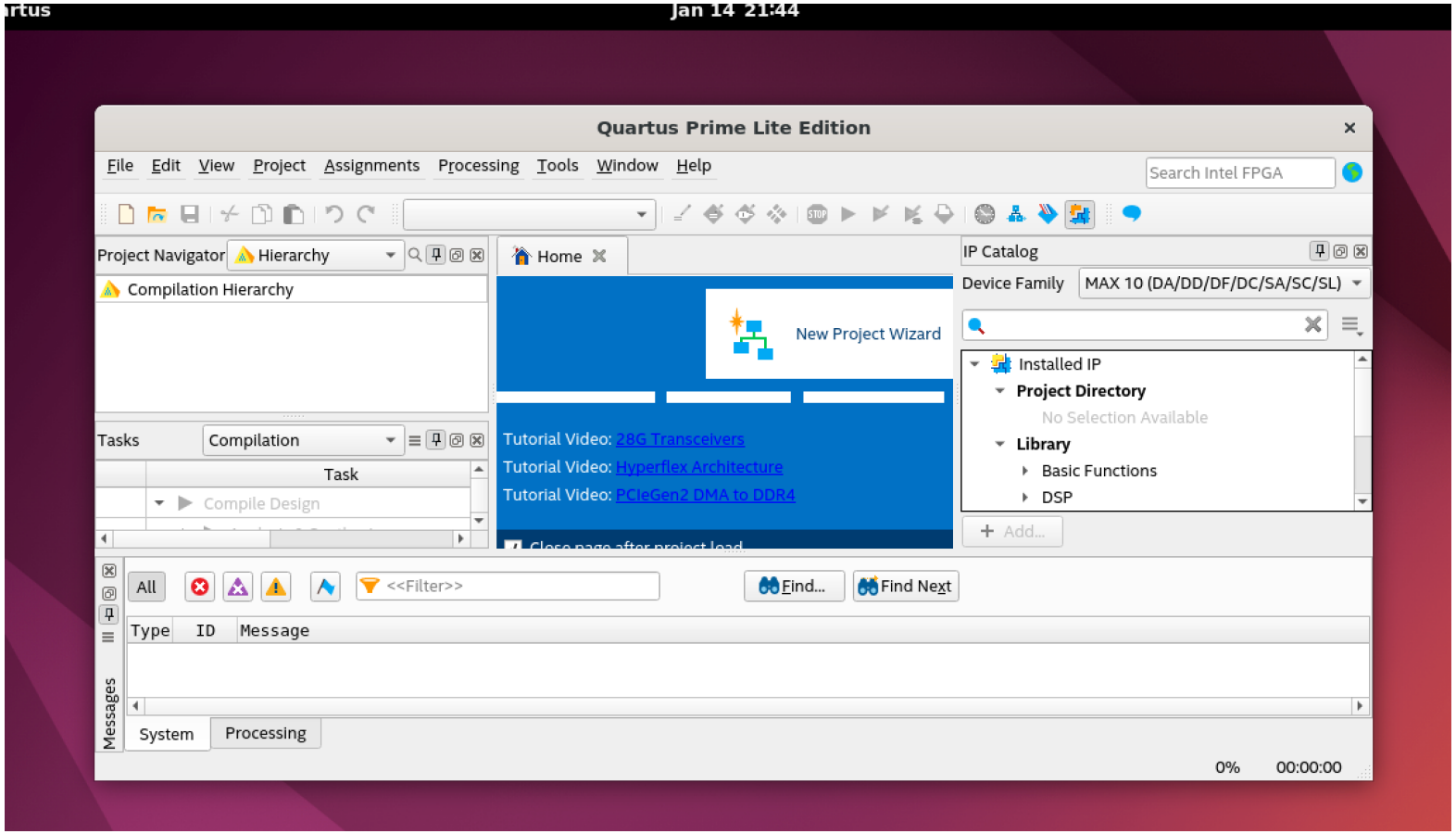


Open quartus in cloud 9 terminal

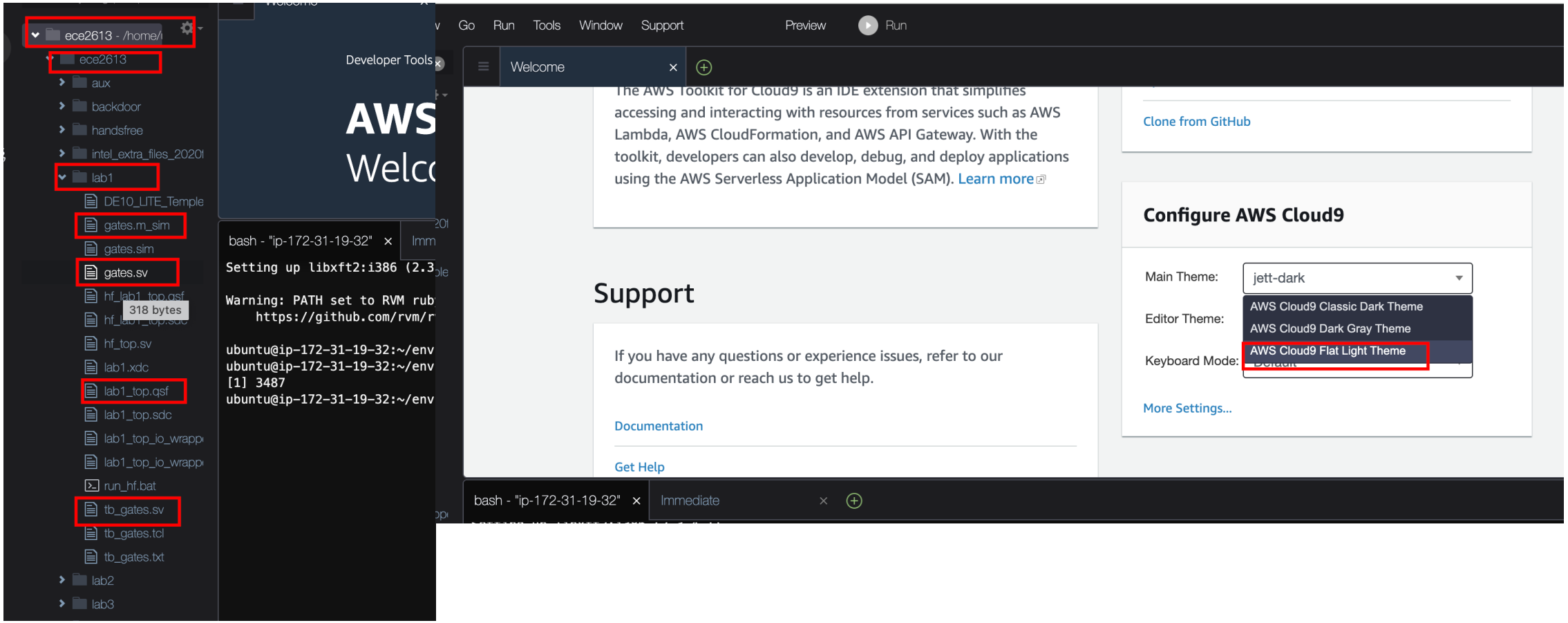
```
Setting up libxft2:i386 (2.3.4-1) ...

Warning: PATH set to RVM ruby but GEM_HOME and/or GEM_PATH not set, see:
    https://github.com/rvm/rvm/issues/3212

ubuntu@ip-172-31-19-32:~/environment$ export DISPLAY=:0
ubuntu@ip-172-31-19-32:~/environment$ quartus &
```



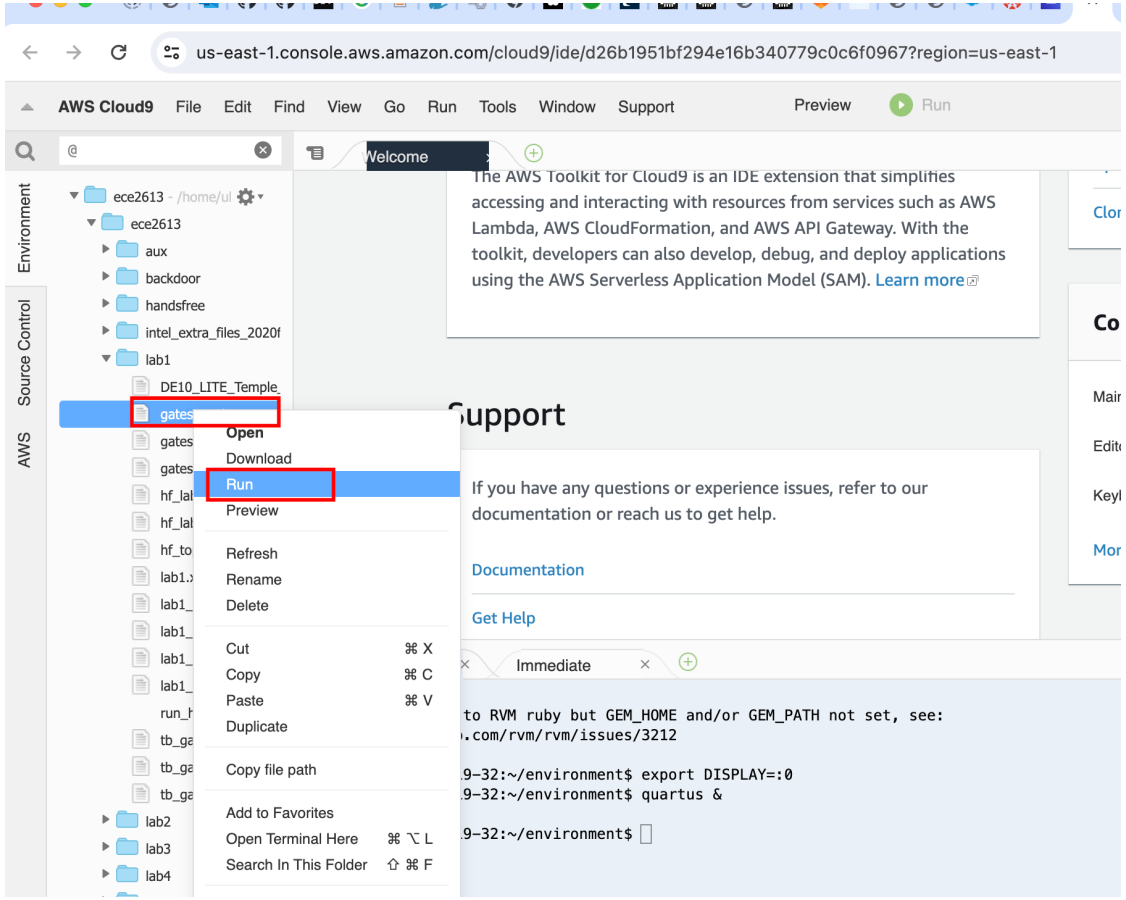
Quartus in xwindow



Cloud 9 file listing

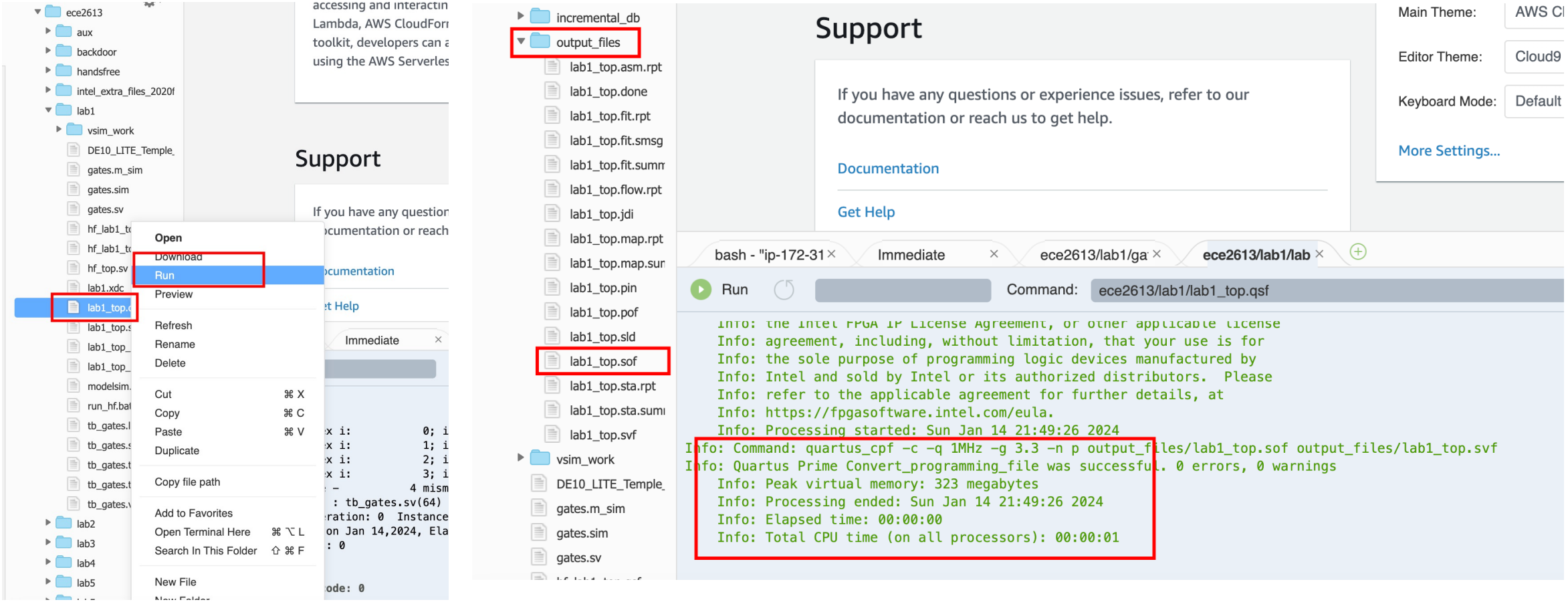


Check m_sim file




```
# Loading work.gates
# do gates.m_sim
# Mismatch--loop index i:           0; input: 00, expected: 1000, received: xxxx
# Mismatch--loop index i:           1; input: 01, expected: 1110, received: xxxx
# Mismatch--loop index i:           2; input: 10, expected: 1110, received: xxxx
# Mismatch--loop index i:           3; input: 11, expected: 0011, received: xxxx
# Simulation complete -           4 mismatches!!!
# ** Note: $finish    : tb_gates.sv(64)
#    Time: 80 ns  Iteration: 0  Instance: /tb_gates
# End time: 21:48:13 on Jan 14,2024, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0


Process exited with code: 0
```

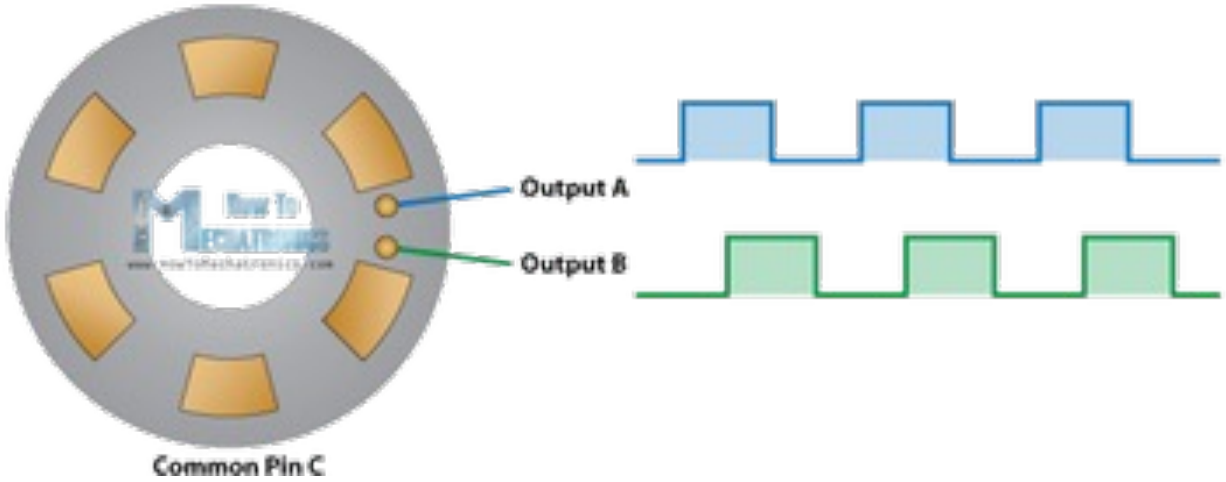
Check qsf file ...



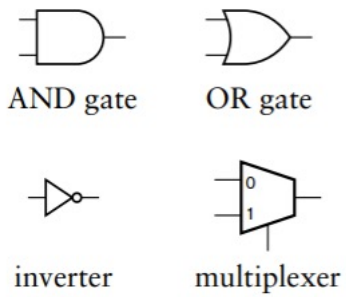
For the next class...

- Think about how you would design a vendor machine with instruction...

Rotatory encoder – example for FPGA



Gate symbols



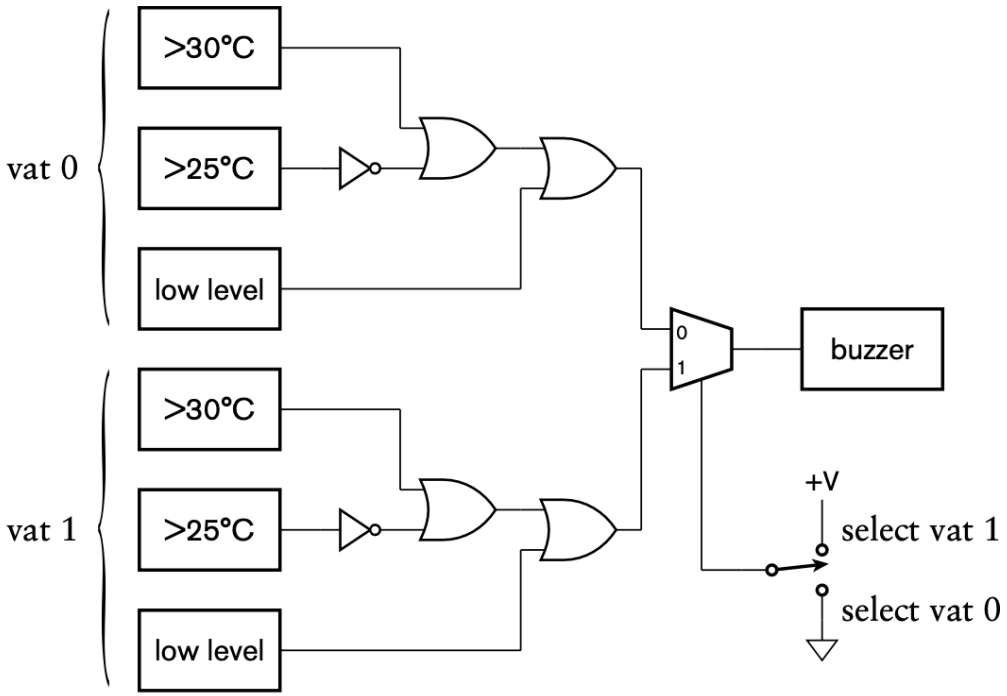
A	B	AND	OR	XOR	Inverter (A)
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Vat example

EXAMPLE 1.1 Suppose a factory has two vats, only one of which is used at a time. The liquid in the vat in use needs to be at the right temperature, between 25°C and 30°C. Each vat has two temperature sensors indicating whether the temperature is above 25°C and above 30°C, respectively. The vats also have low-level sensors. The supervisor needs to be woken up by a buzzer when the temperature is too high or too low or the vat level is too low. He has a switch to select which vat is in use. Design a circuit of gates to activate the buzzer as required.

SOLUTION For the selected vat, the condition for activating the buzzer is

Solution



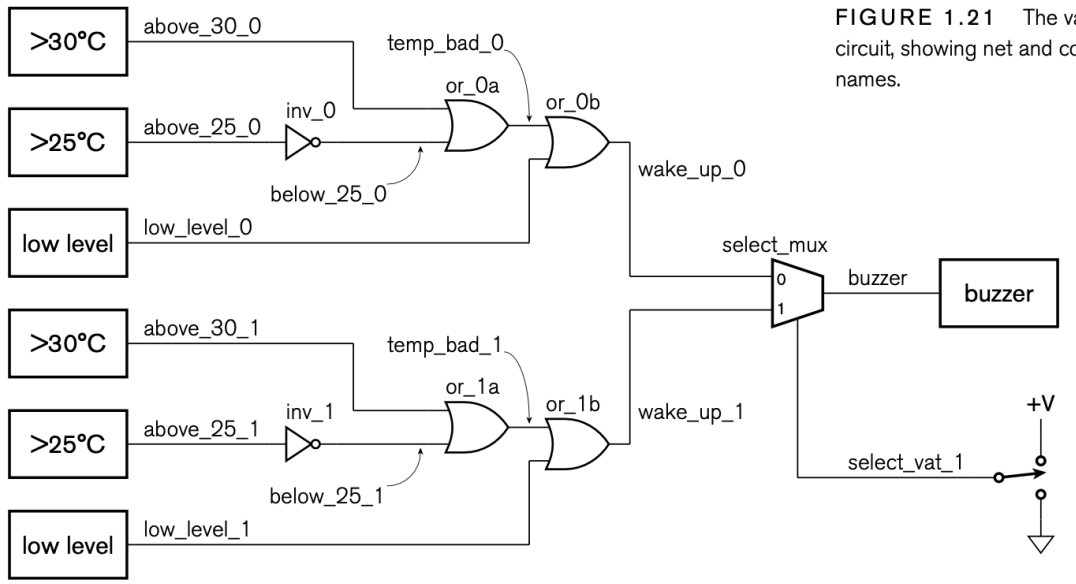


FIGURE 1.21 The vat buzzer circuit, showing net and component names.

Verilog

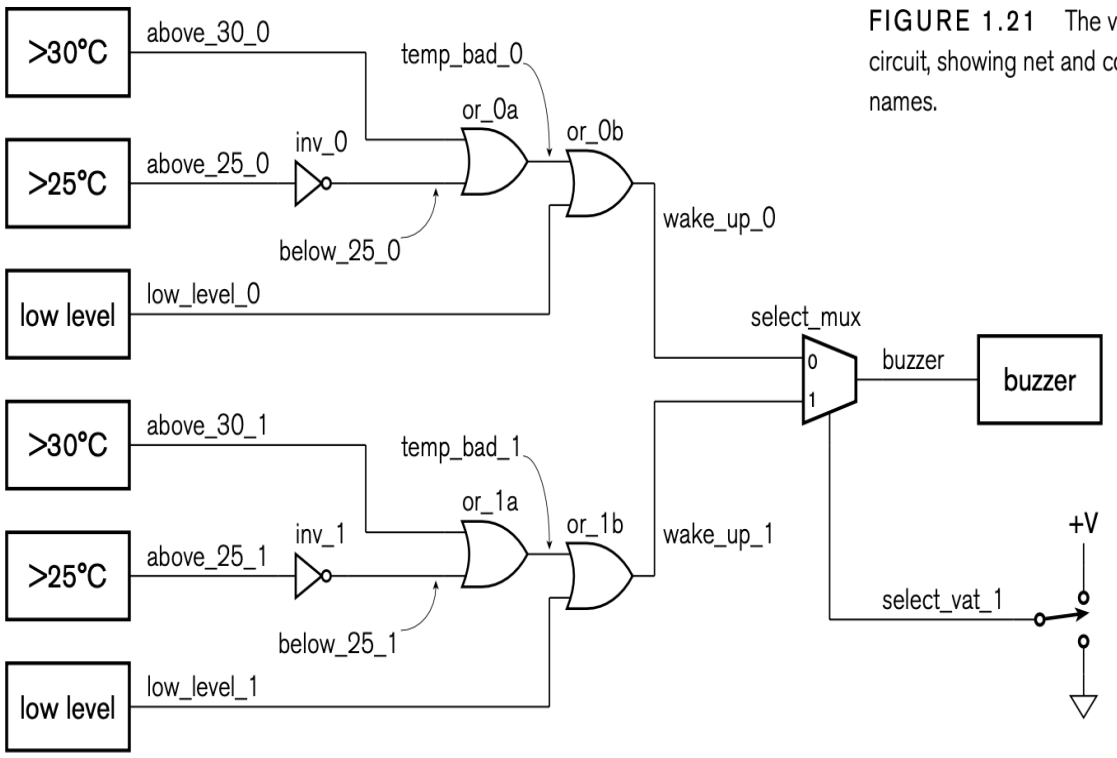


FIGURE 1.21 The vat buzzer circuit, showing net and component names.

```
module vat_buzzer_struct
  ( output buzzer,
    input above_25_0, above_30_0, low_level_0,
    input above_25_1, above_30_1, low_level_1,
    input select_vat_1 );

  wire below_25_0, temp_bad_0, wake_up_0;
  wire below_25_1, temp_bad_1, wake_up_1;

  // components for vat 0
  not inv_0 (below_25_0, above_25_0);
  or  or_0a (temp_bad_0, above_30_0, below_25_0);
  or  or_0b (wake_up_0, temp_bad_0, low_level_0);

  // components for vat 1
  not inv_1 (below_25_1, above_25_1);
  or  or_1a (temp_bad_1, above_30_1, below_25_1);
  or  or_1b (wake_up_1, temp_bad_1, low_level_1);

  mux2 select_mux (buzzer, select_vat_1, wake_up_0, wake_up_1);

endmodule
```

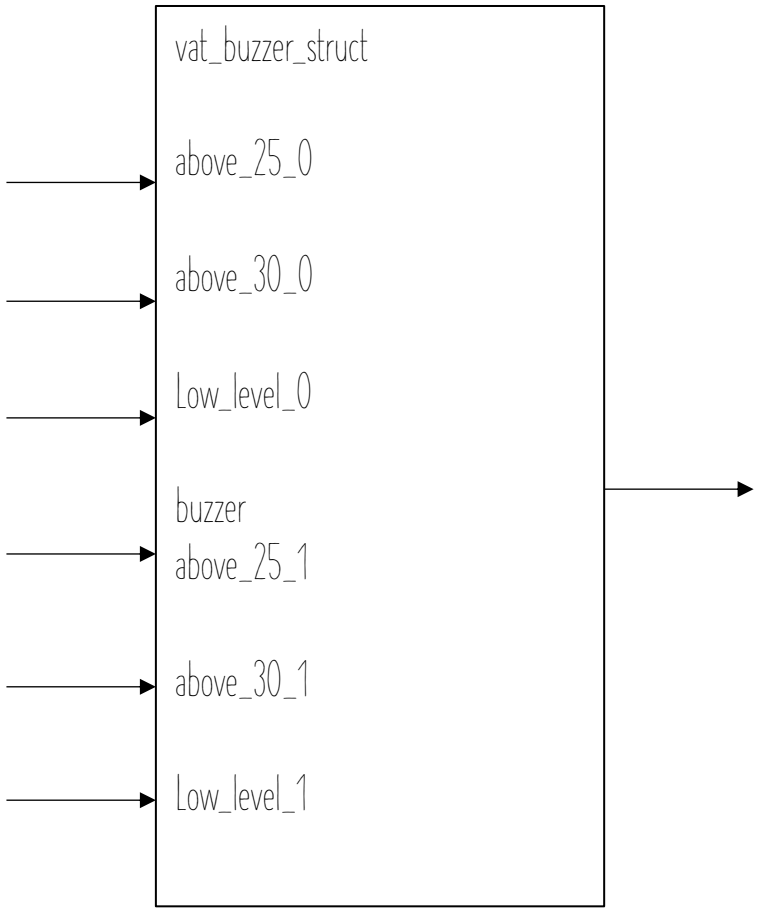
Block diagram and Verilog code

```
module vat_buzzer_struct
  ( output buzzer,
    input above_25_0, above_30_0, low_level_0,
    input above_25_1, above_30_1, low_level_1,
    input select_vat_1 );

  wire below_25_0, temp_bad_0, wake_up_0;
  wire below_25_1, temp_bad_1, wake_up_1;

  // components for vat 0
  not inv_0 (below_25_0, above_25_0);
  or  or_0a (temp_bad_0, above_30_0, below_25_0);
  or  or_0b (wake_up_0, temp_bad_0, low_level_0);

  // components for vat 1
  not inv_1 (below_25_1, above_25_1);
  or  or_1a (temp_bad_1, above_30_1, below_25_1);
  or  or_1b (wake_up_1, temp_bad_1, low_level_1);

  mux2 select_mux (buzzer, select_vat_1, wake_up_0, wake_up_1);

endmodule
```

```
module vat_buzzer_behavior
  ( output buzzer,
    input  above_25_0, above_30_0, low_level_0,
    input  above_25_1, above_30_1, low_level_1,
    input  select_vat_1 );

  assign buzzer =
    select_vat_1 ? low_level_1 | (above_30_1 | ~above_25_1)
                 : low_level_0 | (above_30_0 | ~above_25_0);

endmodule
```

use gate primitive

use logic behavior



Soda vending machine

- Suppose a soda vending machine has two choices of soda, only one of which is dispensed at a time. The soda in the chosen slot needs to have the right cost, between $2 and $3. Each slot has two cost sensors indicating whether the cost is above $3 and above $20, respectively. The slots also have zero-level sensors. The supervisor needs to be notified by a dispenser when the cost is too high or too low, or the soda level is too low. They have a switch to select which soda slot is in use. Design a circuit of gates to activate the dispenser as required.

Timing Diagram

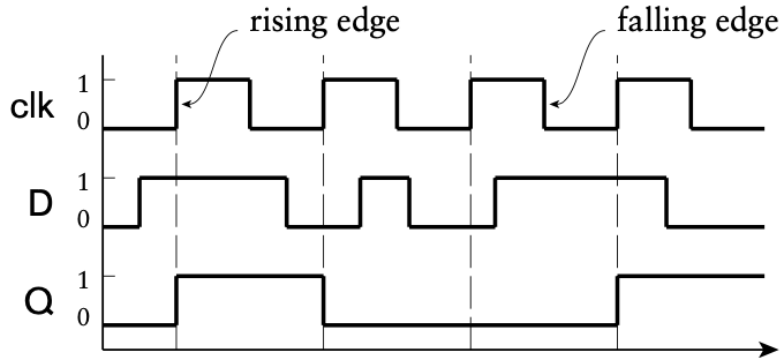


FIGURE 1.7 Timing diagram for a D flip-flop.

EXERCISE 1.6 Complete the timing diagram in Figure 1.25, showing the operation of a rising-edge-triggered D flip-flop.