

# AWS training: crew.aws.batch in R

[Install](#)

[Create job queue](#)

[Credentials](#)

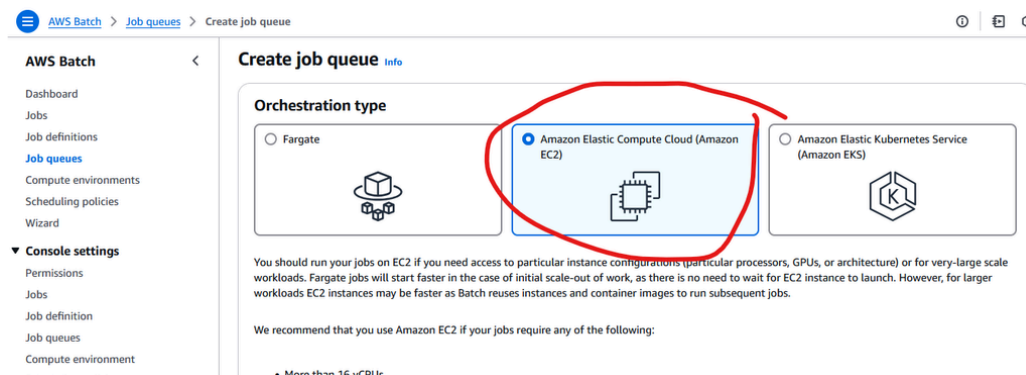
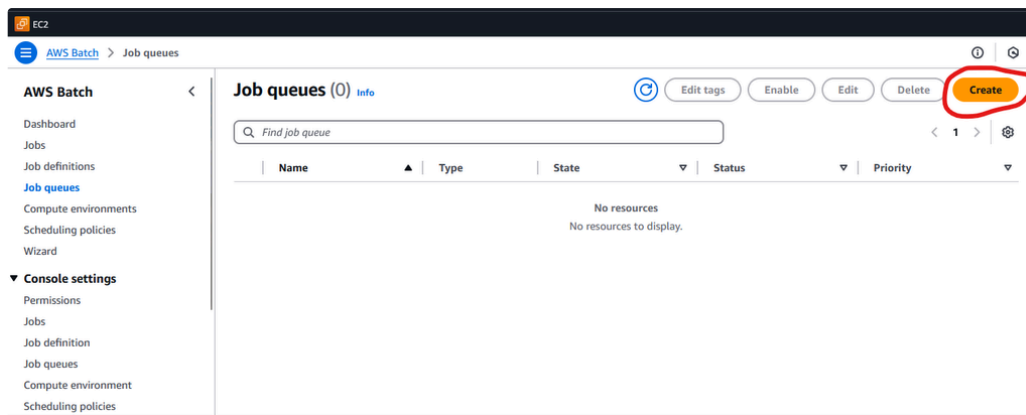
[Run jobs](#)

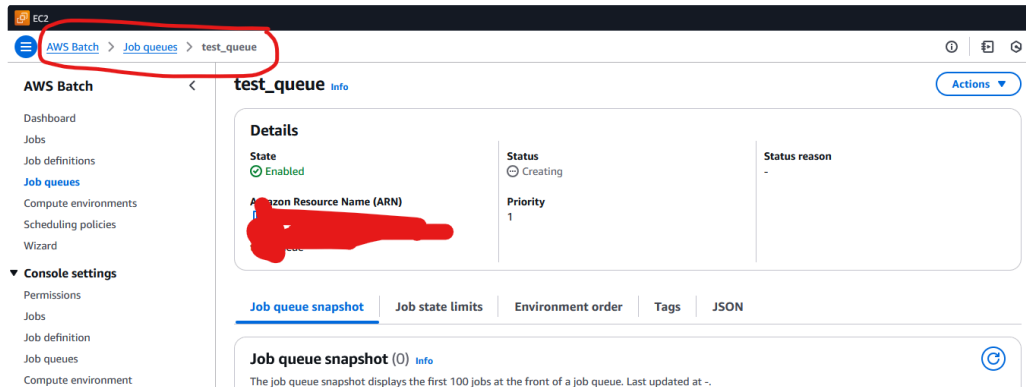
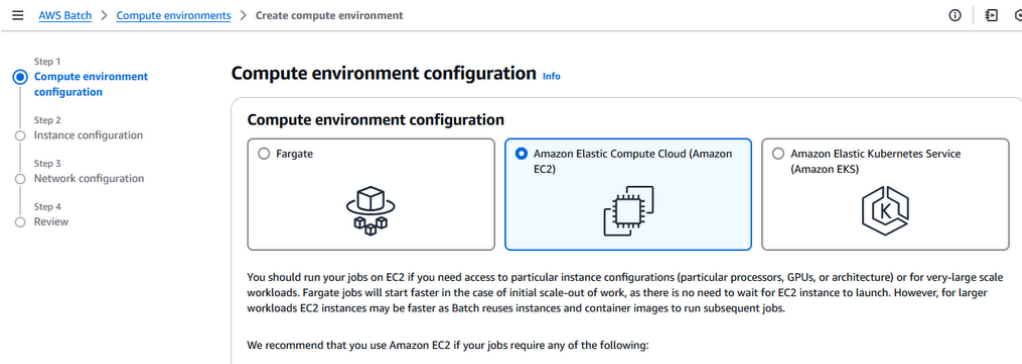
[AWS Batch workers](#)

## Install

```
1 renv::install(c("crew", "crew.aws.batch"))
```

## Create job queue





## Credentials

```
1 Sys.setenv("AWS_ACCESS_KEY_ID" = "foo")
2 Sys.setenv("AWS_SECRET_ACCESS_KEY" = "bar")
3 Sys.setenv("AWS_REGION" = "ap-southeast-2")
```

## Run jobs

```
1 library(crew.aws.batch)
2
3 definition <- crew_definition_aws_batch(
4   job_definition = "test_job",
5   job_queue = "test_queue"
6 )
7
8 definition$register(
9   image = "ghcr.io/jbris/stan-cmdstanr-gpu-docker:2.32.1",
10  platform_capabilities = "EC2",
11  memory_units = "gigabytes",
12  memory = 1,
13  cpus = 1
14 )
15
16 monitor <- crew_monitor_aws_batch(
17   job_definition = "test_job",
18   job_queue = "test_queue"
19 )
20
21 job1 <- definition$submit(name = "job1", command = c("echo", "hello\nworld"))
```

```
22 monitor$status(id = job1$id)
23
24 monitor$jobs()
25 monitor$succeeded()
26 monitor$inactive()
27 monitor$terminate(id = job1$id)
28 monitor$jobs()
```

## AWS Batch workers

```
1 library(crew.aws.batch)
2
3 controller <- crew_controller_aws_batch(
4   name = "my_workflow", # for informative job names
5   workers = 2,
6   tasks_max = 1, # to avoid reaching wall time limits (if any exist)
7   seconds_launch = 600, # to allow a 10-minute startup window
8   seconds_idle = 60, # to release resources when they are not needed
9   processes = NULL, # See the "Asynchronous worker management" section below.
10  options_aws_batch = crew_options_aws_batch(
11    job_definition = "test_job",
12    job_queue = "test_queue",
13    cpus = 1,
14    gpus = NULL,
15    memory = c(1, 2, 4),
16    memory_units = "gigabytes"
17  )
18 )
19 controller$start()
20 controller$push(name = "task", command = sqrt(4))
21 controller$wait()
22 controller$pop()$result
23 controller$terminate()
```