

Installing the Genomics Workflow Core and Cromwell

Summary

The purpose of this document is to demonstrate how an AWS user can provision the infrastructure necessary to run Cromwell versions 52 and beyond on AWS Batch using S3 as an object store using CloudFormation. The instructions cover deployment into an existing VPC. There are two main steps: deploying the genomics workflow core infrastructure which can be used with Cromwell, Nextflow and AWS Step Functions, and the deployment of the Cromwell server and related artifacts.

Assumptions

1. The instructions assume you have an existing AWS account with sufficient credentials to deploy the infrastructure or that you will use a role with CloudFormation that has sufficient privileges (admin role is recommended).
 2. You have an existing VPC to deploy artifacts into. This VPC should have a minimum of two subnets with routes to the public internet. Private subnet routes may be through a NAT Gateway.
-

Deployment of Genomics Workflow Core into an existing VPC.

Take note of the id of the VPC that you will use and the ids of the subnets of the VPC that you will use for the Batch worker nodes. We recommend using two or more private subnets.

1. Open the CloudFormation consoles and select **“Create stack”** with new resources. Enter `https://aws-genomics-workflows.s3.amazonaws.com/latest/templates/gwfc/core/gwfc-core-root.template.yaml` as the Amazon S3 URL.

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready

☐ Use a sample template

☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL

☐ Upload a template file

Amazon S3 URL

Amazon S3 template URL

S3 URL: Will be generated when URL is provided

[View in Designer](#)

Cancel

Next

2. Select appropriate values for your environment including the VPC and subnets you recorded above. It is recommended to leave the Default and High Priority Min vCPU values at 0 so that the AWS Batch cluster will not have any instances running when there are no workflows running. Max vCPU values may be increased if you expect to run large workloads utilizing many CPUs. Leave the Distribution Configuration values with the preset defaults.

Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Required

S3 Bucket Name

A S3 bucket name for storing analysis results. The bucket name must respect the S3 bucket naming conventions (can contain lowercase letters, numbers, periods and hyphens).

Existing Bucket?

Does this bucket already exist?

VPC ID

The VPC to create security groups and deploy AWS Batch to. NOTE: Must be the same VPC as the provided subnet IDs.

VPC Subnet IDs

Subnets you want your batch compute environment to launch in. We recommend private subnets. NOTE: Must be from the VPC provided.

subnet-0c5717ca74d12d238 (10.0.96.0/20) (CdkVpcStack/vpc/PrivateSubnet1) ✕

subnet-074c18061440d9a9d (10.0.112.0/20) (CdkVpcStack/vpc/PrivateSubnet2) ✕

subnet-03f0e9ffc54cb615d (10.0.144.0/20) (CdkVpcStack/vpc/PrivateSubnet4) ✕

subnet-053f099e5ea2e7a73 (10.0.16.0/20) (CdkVpcStack/vpc/PublicSubnet2) ✕

subnet-0ac961079ee2c4c41 (10.0.160.0/20) (CdkVpcStack/vpc/PrivateSubnet5) ✕

subnet-061c6a2271ea526a6 (10.0.128.0/20) (CdkVpcStack/vpc/PrivateSubnet3) ✕

subnet-04248a83ab8156878 (10.0.176.0/20) (CdkVpcStack/vpc/PrivateSubnet6) ✕

Optional

Namespace

Optional namespace (e.g. project name) to use to label resources. If not specified the stack-name will be used.

Default Min vCPU

The minimum number of CPUs for the default Batch Compute Environment

Default Max vCPU

The maximum number of CPUs for the default Batch Compute Environment

High Priority Min vCPU

The minimum number of CPUs for the high-priority Batch Compute Environment

High Priority Max vCPU

The maximum number of CPUs for the high-priority Batch Compute Environment

Distribution Configuration

Artifact S3 Bucket Name

S3 Bucket where distribution artifacts and additions scripts are stored

Artifact S3 Prefix

Prefix in ArtifactBucketName where distribution artifacts and additions scripts are stored

Template Root URL

Root URL for where nested templates are stored

Cancel

Previous

Next

3. Optionally add tags and click **Next**

4. Review the parameters, acknowledge the Capabilities notifications and click **“Create Stack”**

Capabilities

The following resource(s) require capabilities: [AWS::CloudFormation::Stack]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: CAPABILITY_AUTO_EXPAND. Check the capabilities of these resources.

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

☒ I acknowledge that AWS CloudFormation might require the following capability:
CAPABILITY_AUTO_EXPAND

Cancel

Previous

Create change set

Create stack

The template will now create several nested stacks to deploy the required resources. This step will take approximately 10 minutes to complete. When this is complete you can proceed with the “[Deploy Cromwell Resources](#)” section below.

Deploy Cromwell Resources

1. Ensure all steps of the CloudFormation deployment of the Genomics Workflow Core have successfully completed before proceeding any further.
2. From the CloudFormation console select “**Create Stack**” and if prompted select “**With new resources (Standard)**”
3. Fill in the Amazon S3 URL with `https://aws-genomics-workflows.s3.amazonaws.com/latest/templates/cromwell/cromwell-resources.template.yaml`

4. 

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready

☐ Use a sample template

☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL

☐ Upload a template file

Amazon S3 URL

`https://aws-genomics-workflows.s3.amazonaws.com/latest/templates/cromwell/cromwell-resources.template.yaml`

Amazon S3 template URL

S3 URL: `https://aws-genomics-workflows.s3.amazonaws.com/latest/templates/cromwell/cromwell-resources.template.yaml`

View in Designer

Cancel

Next

5. Fill in appropriate values for the template. For **GWFCoreNamespace** use the names space value you used in the section above. You should use the same VPC as you used in the previous step above. To secure your Cromwell server you should change the SSH Address Range and HTTP Address Range to trusted values, these will be used when creating the servers security group.
6. You may either use the latest version of Cromwell (recommended) or specify a version **52 or greater**.
7. Select a MySQL compliant Cromwell Database Password that will be used for Cromwell's metadata database. Select **"Next"**.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Stack Configuration

Namespace

Namespace (e.g. project name) to use to label resources.

GWFCoreNamespace

Namespace of the GWFCore deployment to use.

Network Configuration

VPC ID

Recommended to use the Default VPC here

Server Subnet ID

Subnet for the Cromwell server. For public access, use a public subnet.

Database Subnet IDs

Minimum 2 subnets for the Cromwell Database. Private subnets are recommended.

Instance Configuration

Latest Amazon Linux AMI

The latest Amazon Linux AMI

InstanceType

EC2 instance type. Cromwell itself does not require much compute power. A t3.medium should be sufficient. Larger instances are recommended for concurrent workflow scenarios. If you want to run this server on the free tier, use a t3.micro.

Instance Name

The name of the instance that is created

Key Pair Name

Name of an existing EC2 KeyPair to enable SSH access to the instance

SSH Address Range

The IP address range that can be used to SSH to the EC2 instances. In a production environment we recommend limiting this to a trusted range.

HTTP Address Range

The IP address range that has HTTP access to the EC2 instances. In a production environment we recommend limiting this to a trusted range.

0.0.0.0/0

Cromwell Configuration

Cromwell Version

Version of Cromwell to install. "latest" will retrieve the currently released version of Cromwell from Github.

latest

Cromwell Version Specified

Specific version of Cromwell to install. Must match a released version number. For example, 52, 52.1, etc. The minimum supported version is 52. Ignored if "Cromwell Version" is set to "latest".

CromwellJarUrl

URL to a pre-built cromwell-*.jar file. Example: <https://mycicdserver.com/build/cromwell-XX-SNAP.jar>. If this is specified, CromwellVersion is ignored.

S3 Open Data Bucket ARNs

Open datasets on AWS S3 for workflow inputs

arn:aws:s3:::gatk-test-data/*,arn:aws:s3:::broad-references/*

Cromwell Database Username

The master username for the Aurora MySQL RDS cluster that will be used as Cromwell's database

cromwell

Cromwell Database Password

The master password for the Aurora MySQL RDS cluster that will be used as Cromwell's database

- On the remaining two screens keep the defaults, acknowledge the IAM capabilities and then click **Create Stack**

Once the stack completes an EC2 will be deployed and it will be running an instance of the Cromwell server. You can now proceed with [Testing your deployment](#)

Testing your Deployment

The following WDL file is a very simple workflow that can be used to test that all the components of the deployment are working together. Add the code block below to a file named `workflow.wdl`

```
workflow helloWorld {
  call sayHello
}

task sayHello {
  command {
    echo "hello world"
  }
  output {
    String out = read_string(stdout())
  }

  runtime {
    docker: "ubuntu:latest"
    memory: "1 GB"
    cpu: 1
  }
}
```

```
}
}
```

This task can be submitted to the servers REST endpoint using `curl` either from a client that has access to the servers elastic IP or from within the server itself using `localhost`. The hostname of the server is also emitted as an output from the cromwell-resources CloudFormation template.

```
curl -X POST "http://localhost:8000/api/workflows/v1" \
-H "accept: application/json" \
-F "workflowSource=@workflow.wdl"
```

It can take a few minutes for AWS Batch to realize there is a job in the work queue and provision a worker to run it. You can monitor this in the AWS Batch console.

You can also monitor the Cromwell server logs in CloudWatch. There will be a log group called `cromwell-server`. Once the run is completed you will see output similar to:

▼ 2020-07-20T12:11:01.247-04:00	2020-07-20 16:11:01,233 cromwell-system-akka.dispatchers.backend-dispatcher-165913 INFO - AwsBatchAsyncBackendJobExecutionActor [UUID(d43c1ba1)helloWorld.sayHello:NA:1]: Status change from Initializing to Succeeded
▼ 2020-07-20T12:11:04.248-04:00	2020-07-20 16:11:03,862 cromwell-system-akka.dispatchers.engine-dispatcher-24 INFO - WorkflowExecutionActor-d43c1ba1-12b0-4d3b-b7f979a3ef06 [UUID(d43c1ba1)]: Workflow helloWorld complete. Final Outputs:
▼ 2020-07-20T12:11:04.249-04:00	{ "helloWorld.sayHello.out": "hello world"
▼ 2020-07-20T12:11:04.249-04:00	}
▼ 2020-07-20T12:11:04.249-04:00	2020-07-20 16:11:03,866 cromwell-system-akka.dispatchers.engine-dispatcher-28 INFO - WorkflowManagerActor WorkflowActor-d43c1ba1-12b0-4d3b-b7f979a3ef06 is in a terminal state: WorkflowSucceededState

If the run is successful subsequent runs will be “call cached” meaning that the results of the previous run will be copied for all successful steps. If you resubmit the job you will very quickly see the workflow success in the server logs and no additional jobs will be seen in the AWS Batch console. You can disable call caching for the job by adding an options file and submitting it with the run. This will cause the workflow to be re-executed in full.

```
{
  "write_to_cache": false,
  "read_from_cache": false
}
```

```
curl -X POST "http://localhost:8000/api/workflows/v1" \
-H "accept: application/json" \
-F "workflowSource=@workflow.wdl" \
-F "workflowOptions=@options.json"
```

For a more realistic workflow, here is a WDL for simple variant calling using `bwa-mem`, `samtools`, and `bcftools`: <https://github.com/wleepang/demo-genomics-workflow-wdl>

Clone the repo, and submit the WDL file to cromwell. The workflow uses default inputs from public data sources. If you want to override these inputs, modify the `inputs.json` file accordingly and submit it along with the workflow.