LETTERKENNY INSTITUTE OF TECHNOLOGY

ASSIGNMENT COVER SHEET

To Be Completed By The Student

Lecturer’s Name: Ruth Lennon

Assessment Title: Cloud formation

Submission Date: 20/11/20222

Student’s Name: Ankush Ikhar Id. Number: L00171111

Course / Stage Devops

Subject/Module: Iac Devops Pipeline Word Limit: Actual Word Count: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I confirm that the work submitted has been produced solely through my own efforts.

Student’s signature: Ankush Ikhar Date: 20.11.2022

|  |
| --- |
| **Note**: **PENALTIES**   * The total marks available for an assessment will be reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. * Assessment work received more than two weeks late, without prior approval by the lecturer will receive a mark of zero. * Marks awarded will be reduced by 10 % if submitted work is greater than 10% above or below the assigned word limit. * A further hard or electronic copy of your submitted work may be requested, and therefore you must keep a copy on disc. * Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute’s Assessment Regulations   **Plagiarism:** Presenting the ideas, words of someone else without proper acknowledgement. Refer to the Institutes’ procedures and guidelines for the assessment of learners. |

Conclusion:

With the help of the configuration orchestration tool like AWS Cloud Formation, we can automate deployments by coding our architecture. Using yaml, we are building a cloud formation stack which will create all the resources needed for application.

We can create parameters and pass values to these parameters from the console. The parameters are created with necessary pattern check. This will make our template more dynamic and help us build architecture differently based on requirements. Default values are set which can be used in case no parameter value is provided in the console.

I have created necessary mapping which will provide me the correct image id based on the region the stack is running. This will make the stack more flexible and can be used to create architecture in many regions. I can also add mapping to decide the size of EC2 instances or RDS instances based on the application environment. This will help us in reducing the cost as we will not create unnecessary expensive configurations in development environment.

Ec2 instances are created in public and private subnet based on the values passed as parameter from the console. Make sure EC2 security groups don't have extensive port ranges open. Large port ranges open might expose vulnerabilities. Due to wide port ranges being exposed, an attacker can scan the ports and find flaws in hosted apps without being easily tracked. I Keep credentials out of my cloud formation templates: Storing credentials or other sensitive data in my AWS CloudFormation templates is not a wise choice. Instead, I make advantage of dynamic references to specify and access data stored in other services, such as AWS Secrets Manager, without ever saving the actual reference value in Cloud Formation. We can also create necessary configs in NACl to make our applications more secure.

Non-confidential parameters should be referred instead of being passed directly to CloudFormation as literal values. Pseudo-parameters are offered by AWS CloudFormation. I have tried to use these parameters like AWS::Region and AWS::Account Id wherever possible. These established criteria make it easier to develop templates that are independent of accounts and geographic locations and enable me to concentrate just on the Infrastructure as Code description.

In an ideal world, templates would be created such that they can be utilized across different AWS accounts or regions. Tagging resources is important for several reasons, and it should be viewed as a need rather than merely a best practice. It could improve cost allocations, security audits, and other tasks may be accomplished with better labelling. We need to ensure that all the resources are tagged as expected. It helps us to manage resources and not lose track of any resources which will save us cost.

The RDS instances are created in the private subnet with necessary db subnet group and security group. The security group only allows port 3306 from jumpbox and webserver. This will make the rds more secure because all the other ports are closed. We can enable multi-AZ deployment which will help us to make our database fault tolerant. Even If anything happens to DB in one AZ, since it is replicated in other AZs, we will not lose anything and our application will be fine.

Iac should always be created adhering best practices and security should always be high priority. We should always consider keeping our architecture up to date with necessary updates. We can update our stack to follow nested stack approach which will make the iac more flexible and encourage reusability. It will also be easy to make new changes to the code.

Appendices:

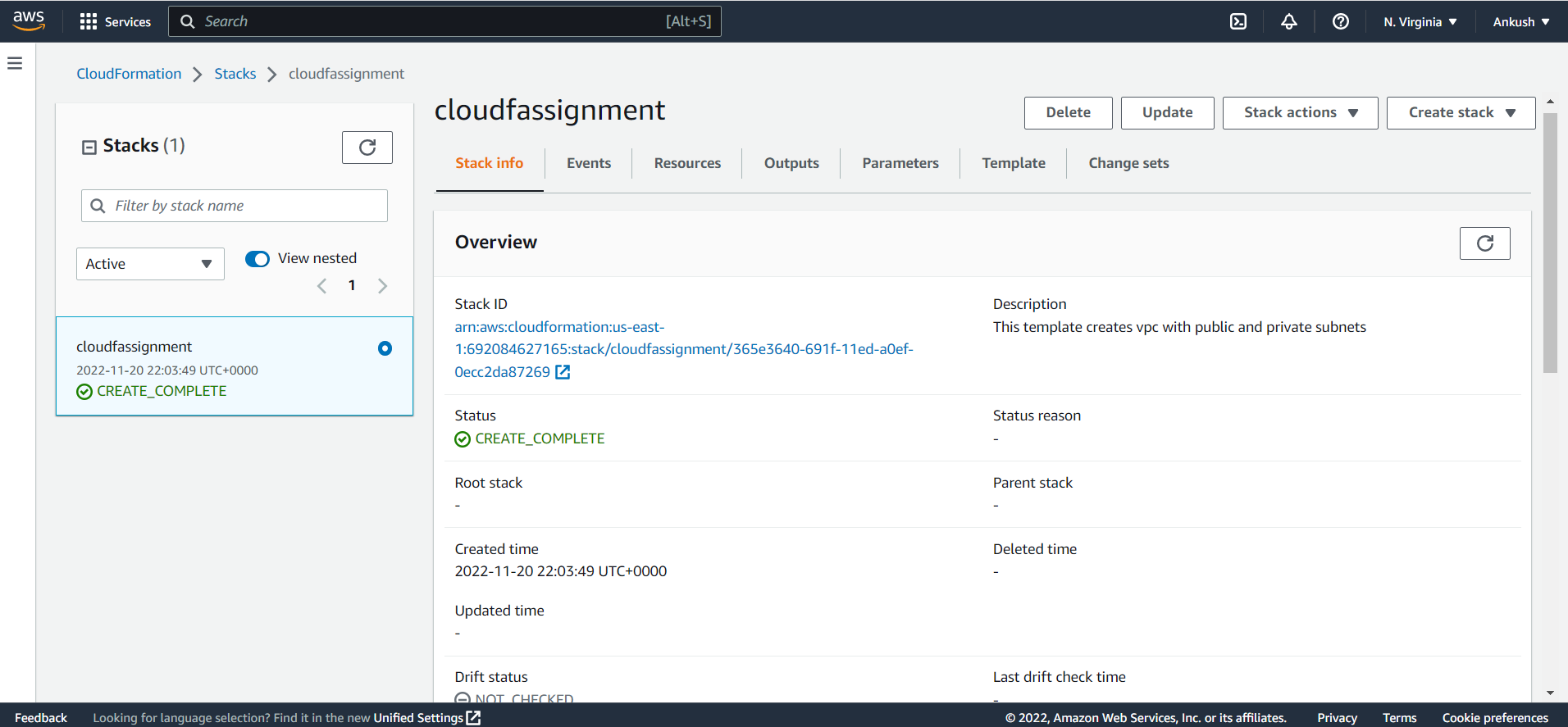


Figure 1: Stack Info

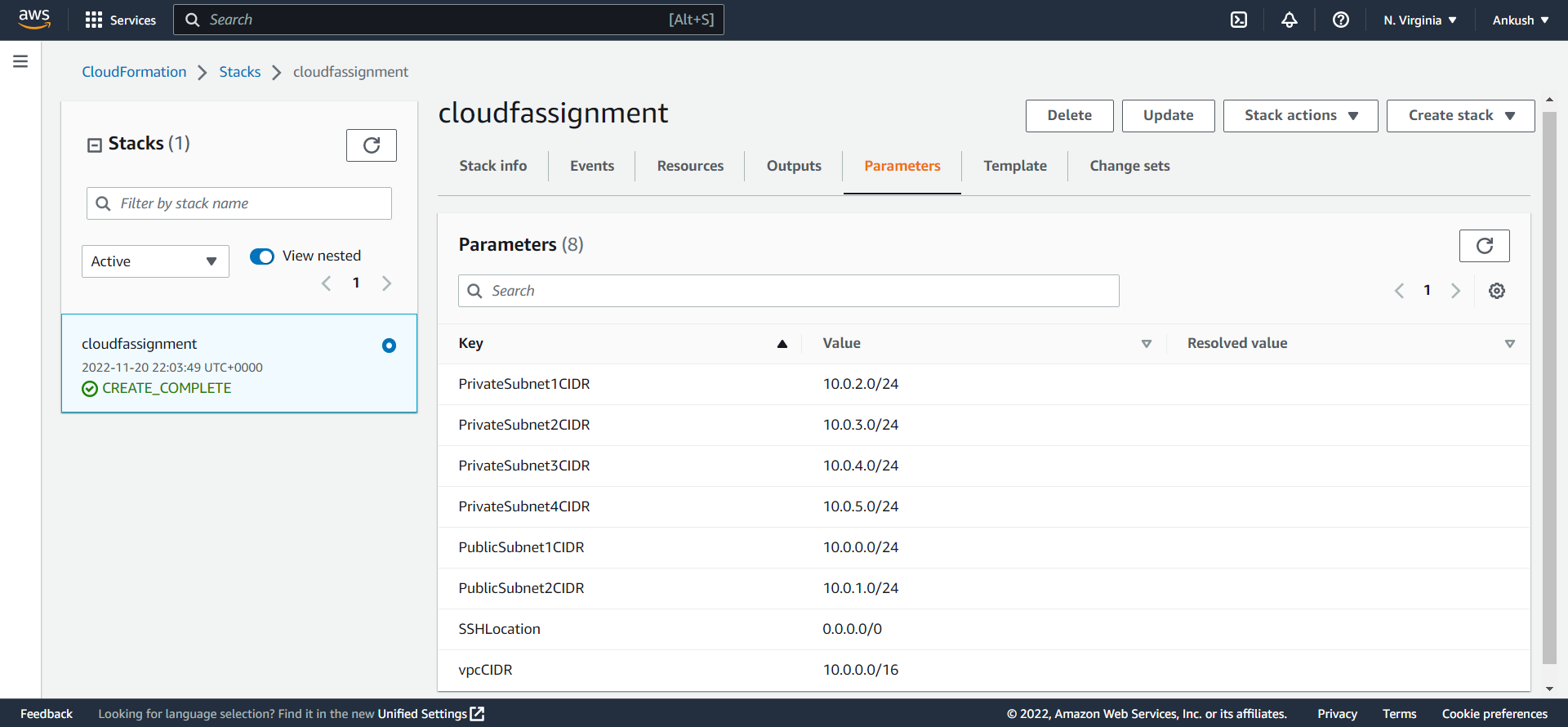


Figure 2: Parameter

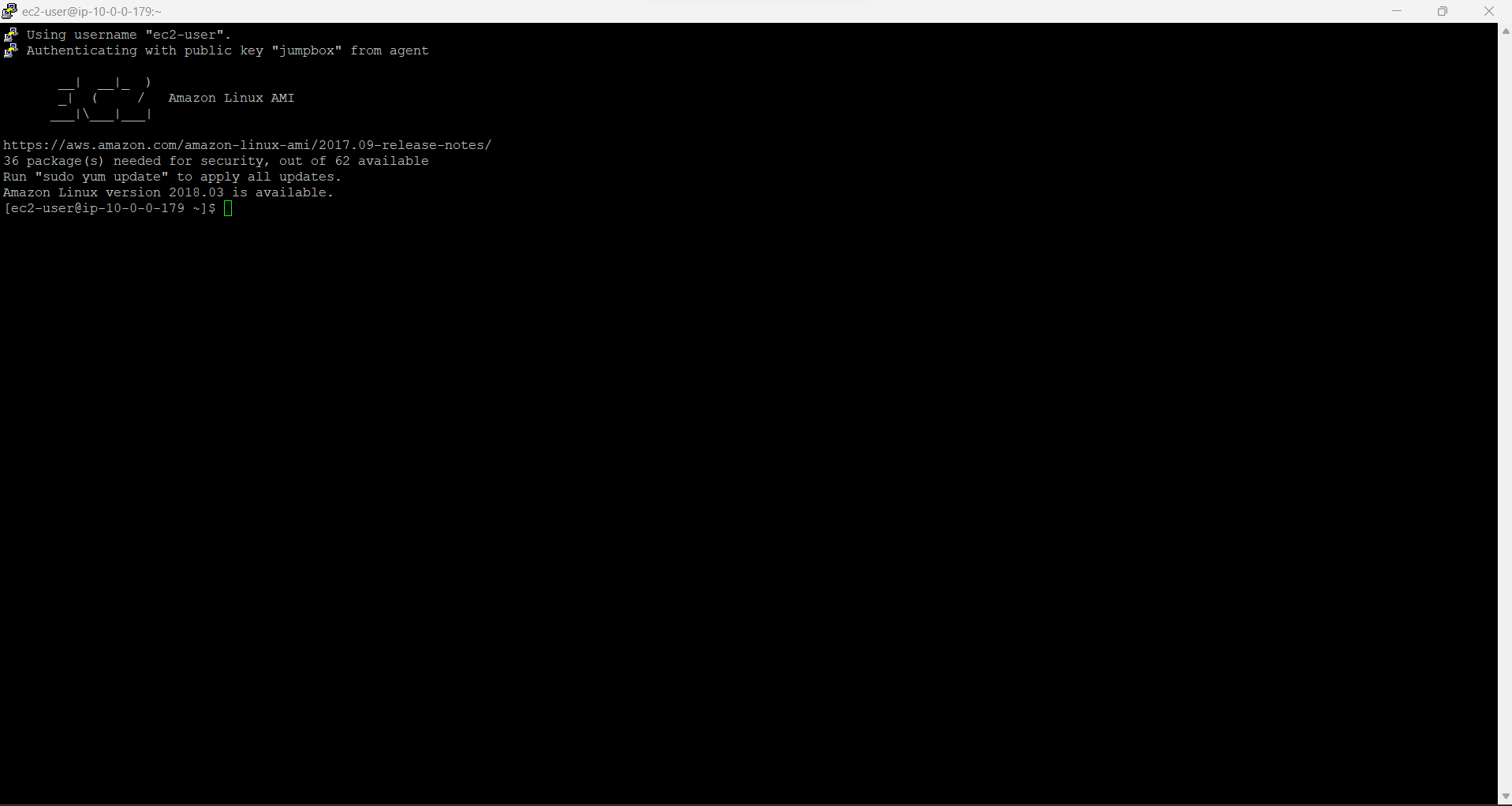


Figure 3: Linux Jumpbox

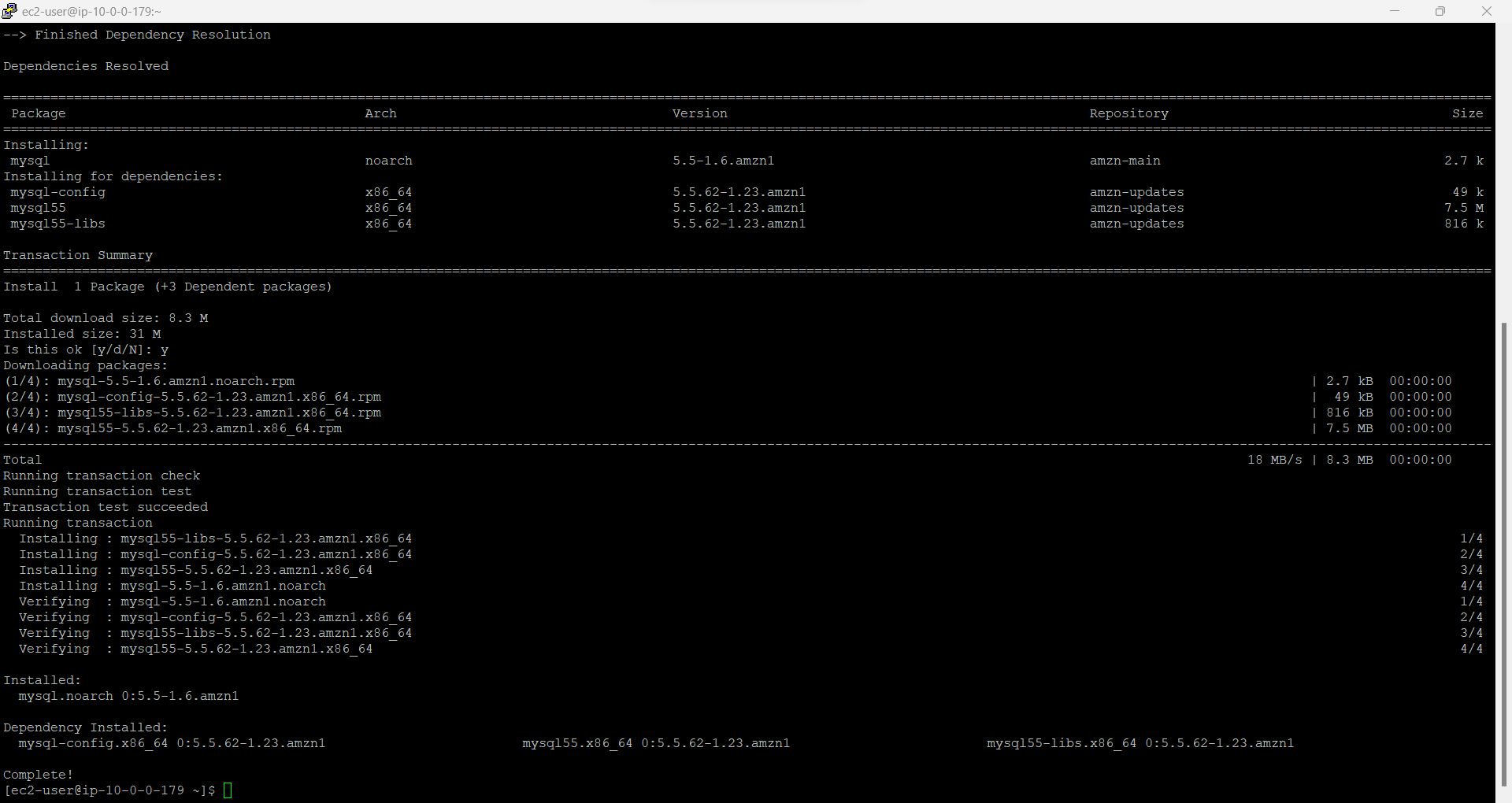


Figure 4: Install MySQL

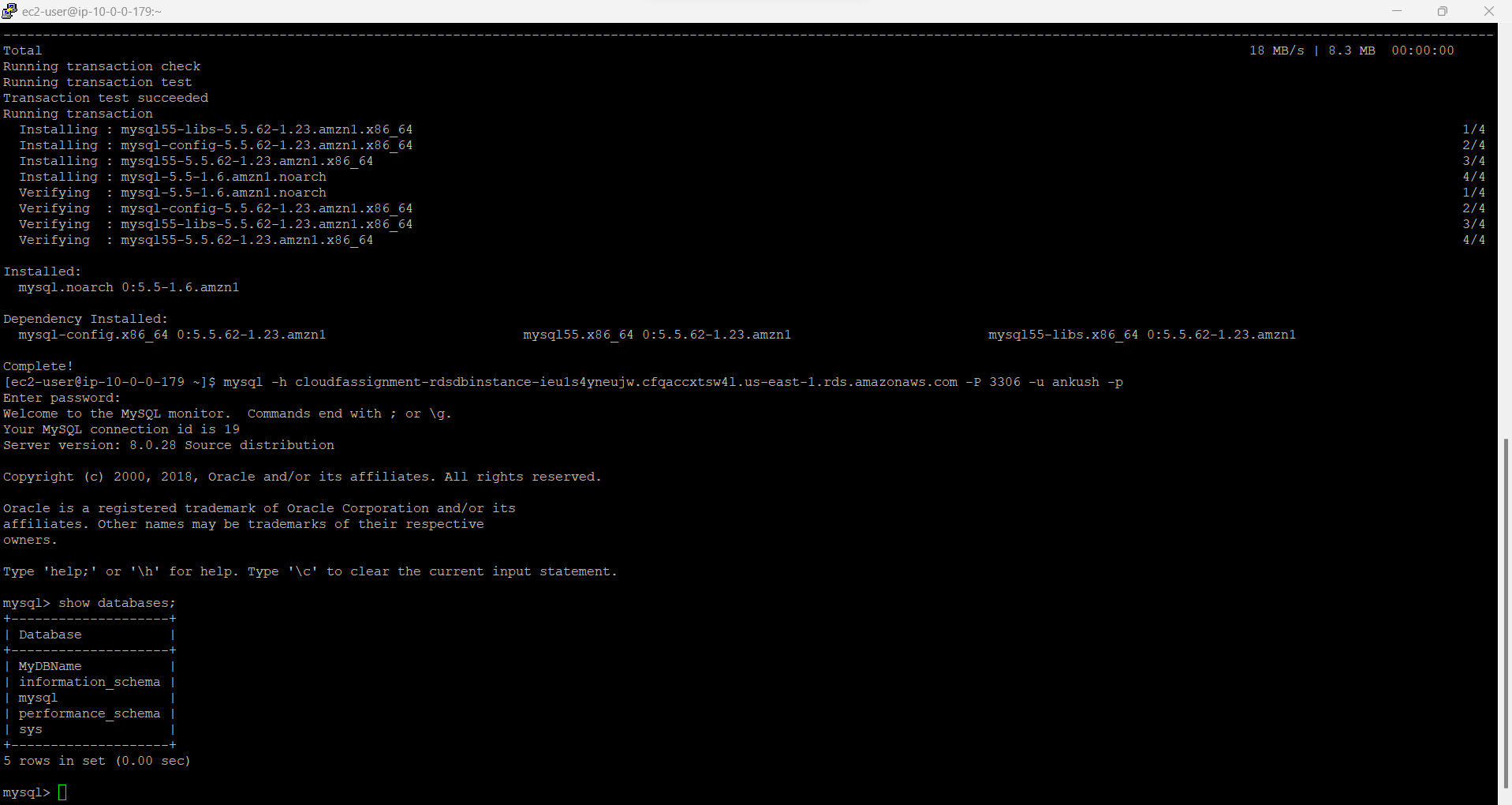


Figure 5: Database Table

References:

Reference 1:

<https://thorntech.com/whatisawscloudformation/#:~:text=AWS%20CloudFormation%20is%20a%20configuration%20orchestration%20tool%20that,templates%20and%20see%20the%20interdependencies%20of%20your%20resources>

Reference 2:

<https://www.techtarget.com/searchitoperations/tip/Building-an-infrastructure-as-code-pipeline-in-the-cloud>

Reference 3:

<https://www.cprime.com/resources/blog/what-is-infrastructure-as-code-in-ci-cd-pipeline/>

Reference 4:

<https://www.threatstack.com/blog/aws-security-groups-what-and-how-to-maximiz>

Reference 5:

<https://into-the-cloud.mechmann.com/2020/06/21/aws-cloudformation-best-practices/>

Reference 6: [21 AWS Security Groups https://www.mcafee.com/blogs/enterprise/cloud-security/21-aws-security-groups-best-practices/Best Practices | McAfee Blog](https://www.mcafee.com/blogs/enterprise/cloud-security/21-aws-security-groups-best-practices/)

Reference 7:

<https://www.checkpoint.com/cyber-hub/cloud-security/what-is-aws-security-groups/13-aws-security-best-practices/>

Reference 8:

<https://www.corestack.io/aws-security-best-practices/>

Reference 9:

<https://helecloud.com/blog/10-best-practices-for-ensuring-your-aws-cloud-security/>